

**SISTEMA COMPUTACIONAL PARA EL DIAGNÓSTICO ASISTIDO DE
DETERIORO COGNITIVO LEVE EN PERSONAS MAYORES DE 60 AÑOS**

JUAN CAMILO FLÓREZ RUIZ

UNIVERSIDAD AUTÓNOMA DE MANIZALES

FACULTAD DE INGENIERÍA

MAESTRÍA EN GESTION Y DESARROLLO DE PROYECTOS DE SOFTWARE

MANIZALES – 2016

**SISTEMA COMPUTACIONAL PARA EL DIAGNÓSTICO ASISTIDO DE
DETERIORO COGNITIVO LEVE EN PERSONAS MAYORES DE 60 AÑOS**

JUAN CAMILO FLÓREZ RUIZ

Ingeniero de Sistemas

**Proyecto de grado para optar al título de Magister en Gestión y Desarrollo
de Proyectos de Software**

TUTOR

SANTIAGO MURILLO RENDÓN

Magister en Ingeniería - Automatización Industrial

COTUTOR

BELARMINO SEGURA GIRALDO

Doctor en Ingeniería

UNIVERSIDAD AUTÓNOMA DE MANIZALES

FACULTAD DE INGENIERÍA

MAESTRÍA EN GESTION Y DESARROLLO DE PROYECTOS DE SOFTWARE

MANIZALES – 2016

CONTENIDO

RESUMEN	1
ABSTRACT	2
INTRODUCCIÓN	3
1 REFERENTE CONTEXTUAL.....	5
1.1 Descripción del área problemática.....	5
1.2 Antecedentes.....	6
1.3 Justificación	14
1.4 Formulación de la propuesta de investigación.....	16
1.5 Objetivo general	17
1.6 Objetivos específicos	17
2 REFERENTE TEÓRICO	18
2.1 Deterioro cognitivo.....	18
2.2 Deterioro cognitivo leve	18
2.3 Proceso de diagnóstico de Deterioro Cognitivo Leve.....	20
2.4 Pruebas de valoración cognitiva.....	23
2.4.1 <i>Lawton Instrumental Activities of Daily Living IADL</i>	23
2.4.2 <i>Mini Mental State Examination MMSE</i>	24
2.4.3 <i>Montreal Cognitive Assessment MoCA</i>	25
2.4.4 <i>Clinical Dementia Rating CDR</i>	26
2.4.5 <i>Global Deterioration Scale GDS</i>	27
2.5 Diagnóstico asistido por computadora	28
2.6 Inteligencia artificial	28
2.6.1 Inteligencia artificial simbólica	30

2.6.2	Inteligencia artificial subsimbólica	31
2.6.3	Sistemas híbridos.....	32
2.7	Aprendizaje de máquina.....	34
2.7.1	Aprendizaje supervisado	34
3	ESTRATEGIA METODOLÓGICA	35
3.1	Población	35
3.2	Obtención de la información	36
3.3	Descripción de las variables.....	38
3.4	Procedimientos (análisis del desempeño de las técnicas de ML)	41
3.5	Procedimientos (metodología de desarrollo de software)	43
3.6	Procedimientos (verificación del proceso de desarrollo de software)	46
3.7	Procedimientos (esquema de <i>cross-validation</i> o validación cruzada)	46
3.8	División del proyecto en fases y actividades	48
3.8.1	Análisis del desempeño de las técnicas de aprendizaje de máquina.....	48
3.8.2	Construcción del subsistema computacional tipo A	49
3.8.3	Construcción del subsistema computacional tipo B	49
3.8.4	Integración de los subsistemas computacionales de tipo A y B.....	50
3.8.5	Validación del sistema computacional híbrido	50
4	DESARROLLO Y RESULTADOS	51
4.1	Fase 1: análisis del desempeño de las técnicas de aprendizaje de máquina	51
4.1.1	Herramientas empleadas en el análisis de las técnicas	51
4.1.2	Técnica 1: <i>Passive-Aggressive Classifier</i>	52
4.1.3	Técnica 2: <i>Ridge Classifier</i>	59
4.1.4	Técnica 3: <i>Linear SVM</i>	65

4.1.5	Técnica 4: <i>Quadratic SVM</i>	72
4.1.6	Técnica 5: <i>Cubic SVM</i>	79
4.1.7	Técnica 6: <i>Coarse Gaussian SVM</i>	85
4.1.8	Técnica 7: <i>Medium Gaussian SVM</i>	91
4.1.9	Técnica 8: <i>Fine Gaussian SVM</i>	97
4.1.10	Técnica 9: <i>Decision Tree</i>	103
4.1.11	Técnica 10: <i>Decision Trees (Bagging)</i>	110
4.1.12	Técnica 11: <i>Decision Trees (Boosting)</i>	117
4.1.13	Técnica 12: <i>AdaBoost-SAMME.R</i>	123
4.1.14	Técnica 13: <i>Random Forest</i>	131
4.1.15	Técnica 14: <i>Extremely Randomized Trees</i>	138
4.1.16	Técnica 15: <i>RUSBoosted Trees</i>	145
4.1.17	Técnica 16: <i>Gradient Boosting</i>	151
4.1.18	Técnica 17: <i>ANN - ELU Softmax Adam</i>	157
4.1.19	Técnica 18: <i>ANN - Hard Sigmoid Softmax Adam</i>	169
4.1.20	Técnica 19: <i>ANN - PReLU Softmax Adam</i>	176
4.1.21	Técnica 20: <i>ANN - ReLU Softmax Adadelata</i>	183
4.1.22	Técnica 21: <i>ANN - ReLU Softmax Adagrad</i>	190
4.1.23	Técnica 22: <i>ANN - ReLU Softmax Adam</i>	197
4.1.24	Técnica 23: <i>ANN - ReLU Softmax Adam Deep</i>	204
4.1.25	Técnica 24: <i>ANN - ReLU Softmax Adamax</i>	211
4.1.26	Técnica 25: <i>ANN - Sigmoid Softmax Adam</i>	218
4.1.27	Técnica 26: <i>ANN - SReLU Softmax Adam</i>	225
4.1.28	Síntesis del desempeño de las técnicas de aprendizaje de máquina.....	232

4.1.29	Elección de las técnicas de aprendizaje de máquina	235
4.2	Fase 2: construcción del subsistema computacional tipo A	237
4.2.1	Extracción de los requerimientos del subsistema computacional tipo A	237
4.2.2	Análisis de los requerimientos del subsistema computacional tipo A	237
4.2.3	Especificación de los elementos del subsistema computacional tipo A	240
4.2.4	Prototipo base del subsistema computacional tipo A.....	242
4.2.5	Iteraciones sobre el prototipo base del subsistema computacional tipo A.	245
4.2.6	Código fuente del subsistema computacional tipo A.....	248
4.3	Fase 3: construcción del subsistema computacional tipo B.....	248
4.3.1	Extracción de los requerimientos del subsistema computacional tipo B	248
4.3.2	Análisis de los requerimientos del subsistema computacional tipo B	249
4.3.3	Especificación de los elementos del subsistema computacional tipo B	251
4.3.4	Prototipo base del subsistema computacional tipo B.....	253
4.3.5	Iteraciones sobre el prototipo base del subsistema computacional tipo B.	255
4.3.6	Código fuente del subsistema computacional tipo B.....	259
4.4	Fase 4: integración de los subsistemas computacionales de tipo A y B	259
4.4.1	Esquema de hibridación elegido	259
4.4.2	Integración de los subsistemas computacionales	261
4.4.3	Código fuente del sistema computacional híbrido	264
4.5	Fase 5: validación del sistema computacional híbrido.....	264
4.5.1	Desempeño del sistema mediante validación cruzada con 10 iteraciones .	264
4.5.2	Desempeño del sistema con nuevos datos	267
5.	DISCUSIÓN DE LOS RESULTADOS.....	268
6.	CONCLUSIONES.....	271

7. TRABAJO FUTURO.....	273
8. BIBLIOGRAFÍA.....	274
9. ANEXOS.....	290
1 Presupuesto del proyecto.....	290
2 Formato de plan de pruebas	291
3 Prueba cognitiva: IADL de Lawton.....	292
4 Prueba cognitiva: MMSE	293
5 Prueba cognitiva: MoCA	294
6 Prueba cognitiva: CDR	295
7 Prueba cognitiva: GDS	296
8 Plantilla N° 1 para la importación de variables del sistema	297
9 Plantilla N° 2 para la importación de variables del sistema	298
10 Plan de pruebas para el subsistema computacional tipo A	298
10.1 Resumen	298
10.2 Requerimientos	298
10.3 Calendario de pruebas.....	299
10.4 Bancos de pruebas.....	299
10.5 Casos de prueba	300
11 Plan de pruebas para el subsistema computacional tipo B	320
11.1 Resumen	320
11.2 Requerimientos	320
11.3 Calendario de pruebas.....	321
11.4 Bancos de pruebas.....	321
11.5 Casos de prueba	322

12	Plan de pruebas para el sistema computacional híbrido	342
12.1	Resumen	342
12.2	Requerimientos	342
12.3	Calendario de pruebas.....	342
12.4	Bancos de pruebas.....	343
12.5	Casos de prueba	343

LISTA DE TABLAS

Tabla 1 Síntesis de los métodos divisados en la revisión de antecedentes para el diagnóstico asistido de MCI.	14
Tabla 2 Descripción de las variables incluidas en el desarrollo del proyecto. La variable dependiente denominada «Diagnóstico» representa el dictamen brindado por los especialistas respecto a la condición mental de las personas. Dicha variable posee 3 clases: persona sana, persona con Deterioro Cognitivo Leve y persona con deterioro cognitivo moderado o severo. Las demás variables son independientes y provienen de la información socio-demográfica diligenciada en la primera consulta médica y de los resultados de las pruebas de valoración cognitiva aplicadas por los profesionales de la salud.....	40
Tabla 3 Fórmulas para calcular las métricas de evaluación del desempeño de un clasificador, obtenidas a partir de la matriz de confusión (Powers, 2007; Saito & Rehmsmeier, 2015).....	42
Tabla 4 Métricas de desempeño para el clasificador Pasivo-Agresivo sobre las bases de datos de la ADNI y Caldas.	58
Tabla 5 Métricas de desempeño para el clasificador contraído sobre las bases de datos de la ADNI y Caldas.	65
Tabla 6 Métricas de desempeño para la SVM lineal sobre las bases de datos de la ADNI y Caldas.	72
Tabla 7 Métricas de desempeño para la SVM cuadrática sobre las bases de datos de la ADNI y Caldas.	78
Tabla 8 Métricas de desempeño para la SVM cúbica sobre las bases de datos de la ADNI y Caldas.	84
Tabla 9 Métricas de desempeño para la SVM Gaussiana de gran anchura sobre las bases de datos de la ADNI y Caldas.	91

Tabla 10 Métricas de desempeño para la SVM Gaussiana de mediana anchura sobre las bases de datos de la ADNI y Caldas.	97
Tabla 11 Métricas de desempeño para la SVM Gaussiana de pequeña anchura sobre las bases de datos de la ADNI y Caldas.	103
Tabla 12 Métricas de desempeño para el árbol de decisión sobre las bases de datos de la ADNI y Caldas.	110
Tabla 13 Métricas de desempeño para el bagging de árboles de decisión sobre las bases de datos de la ADNI y Caldas.	116
Tabla 14 Métricas de desempeño para el boosting de árboles de decisión sobre las bases de datos de la ADNI y Caldas.	123
Tabla 15 Métricas de desempeño para el AdaBoost-SAMME.R sobre las bases de datos de la ADNI y Caldas.	130
Tabla 16 Métricas de desempeño para el bosque aleatorio sobre las bases de datos de la ADNI y Caldas.	136
Tabla 17 Métricas de desempeño para el clasificador Extra-Trees sobre las bases de datos de la ADNI y Caldas.	144
Tabla 18 Métricas de desempeño para el RUSBoost de árboles de decisión sobre las bases de datos de la ADNI y Caldas.	150
Tabla 19 Métricas de desempeño para el clasificador gradient boosting sobre las bases de datos de la ADNI y Caldas.	157
Tabla 20 Definición matemática de las funciones de activación comúnmente empleadas en la construcción de neuronas artificiales (Lasagne, 2015; Chainer, 2015; Jin et al., 2016).	158
Tabla 21 Métricas de desempeño para la ANN – ELU Softmax Adam sobre las bases de datos de la ADNI y Caldas.	169
Tabla 22 Métricas de desempeño para la ANN – Hard Sigmoid Softmax Adam sobre las bases de datos de la ADNI y Caldas.	175

Tabla 23 Métricas de desempeño para la ANN – PReLU Softmax Adam sobre las bases de datos de la ADNI y Caldas.	182
Tabla 24 Métricas de desempeño para la ANN – ReLU Softmax Adadelata sobre las bases de datos de la ADNI y Caldas.	189
Tabla 25 Métricas de desempeño para la ANN – ReLU Softmax Adagrad sobre las bases de datos de la ADNI y Caldas.	196
Tabla 26 Métricas de desempeño para la ANN – ReLU Softmax Adam sobre las bases de datos de la ADNI y Caldas.	203
Tabla 27 Métricas de desempeño para la ANN – ReLU Softmax Adam Deep sobre las bases de datos de la ADNI y Caldas.	210
Tabla 28 Métricas de desempeño para la ANN – ReLU Softmax Adamax sobre las bases de datos de la ADNI y Caldas.	217
Tabla 29 Métricas de desempeño para la ANN – Sigmoid Softmax Adam sobre las bases de datos de la ADNI y Caldas.	224
Tabla 30 Métricas de desempeño para la ANN – SReLU Softmax Adam sobre las bases de datos de la ADNI y Caldas.	231
Tabla 31 Consolidado de las métricas de desempeño de las técnicas de aprendizaje de máquina sobre la base de datos de la ADNI.	233
Tabla 32 Consolidado de las métricas de desempeño de las técnicas de aprendizaje de máquina sobre la base de datos de Caldas.	234
Tabla 33 Símil de las técnicas de aprendizaje de máquina de acuerdo con el desempeño obtenido en el diagnóstico asistido de Deterioro Cognitivo Leve para cada una de las bases de datos (ADNI y Caldas). El color azul indica las técnicas que presentaron un desempeño sobresaliente respecto de las demás en concordancia con las tablas 31 y 32.	235
Tabla 34 Conjunto reducido de técnicas de aprendizaje de máquina con un alto potencial para el diagnóstico asistido de Deterioro Cognitivo Leve. Las técnicas basadas	

en árboles de decisión son simbólicas mientras que las basadas en redes neuronales artificiales son subsimbólicas.....	236
Tabla 35 Requerimiento funcional N° 1 para el subsistema computacional tipo A.	237
Tabla 36 Requerimiento funcional N° 2 para el subsistema computacional tipo A.	238
Tabla 37 Requerimiento funcional N° 3 para el subsistema computacional tipo A.	238
Tabla 38 Requerimiento funcional N° 4 para el subsistema computacional tipo A.	239
Tabla 39 Requerimiento funcional N° 5 para el subsistema computacional tipo A.	239
Tabla 40 Requerimiento funcional N° 6 para el subsistema computacional tipo A.	239
Tabla 41 Comparación de las métricas resultantes de la validación cruzada con 10 iteraciones del árbol de decisión, antes y después de haber reducido el conjunto de variables por medio de bosques aleatorios.....	248
Tabla 42 Requerimiento funcional N° 1 para el subsistema computacional tipo B.	249
Tabla 43 Requerimiento funcional N° 2 para el subsistema computacional tipo B.	250
Tabla 44 Requerimiento funcional N° 3 para el subsistema computacional tipo B.	250
Tabla 45 Requerimiento funcional N° 4 para el subsistema computacional tipo B.	250
Tabla 46 Requerimiento funcional N° 5 para el subsistema computacional tipo B.	250
Tabla 47 Requerimiento funcional N° 6 para el subsistema computacional tipo B.	251
Tabla 48 Comparación de las métricas resultantes de la validación cruzada con 10 iteraciones del árbol de decisión, antes y después de haber reducido el conjunto de variables por medio de bosques aleatorios.....	258
Tabla 49 Requerimiento funcional N° 1 para el sistema computacional híbrido.	261
Tabla 50 Requerimiento funcional N° 2 para el sistema computacional híbrido.	262
Tabla 51 Requerimiento funcional N° 3 para el sistema computacional híbrido.....	262
Tabla 52 Requerimiento funcional N° 4 para el sistema computacional híbrido.	262
Tabla 53 Requerimiento funcional N° 5 para el sistema computacional híbrido.	263

Tabla 54 Comparación de las métricas resultantes de la validación cruzada con 10 iteraciones del sistema híbrido respecto a las obtenidas en la validación del subsistema computacional tipo A.....	265
Tabla 55 Comparación de las métricas resultantes de la validación cruzada con 10 iteraciones del sistema híbrido respecto a las obtenidas en la validación del subsistema computacional tipo B.	266
Tabla 56 Comparación de las métricas resultantes de la validación cruzada con 10 iteraciones del sistema híbrido respecto a las obtenidas por el árbol de decisión en el estudio comparativo de las técnicas de ML.....	266
Tabla 57 Comparación de las métricas resultantes de la validación cruzada con 10 iteraciones del sistema híbrido respecto a las obtenidas por la red neuronal artificial en el estudio comparativo de las técnicas de ML.	266
Tabla 58 Presupuesto asociado al proyecto.	290
Tabla 59 Costo unitario por recurso.	290
Tabla 60 Plantilla para registrar los requerimientos en el plan de pruebas.....	291
Tabla 61 Plantilla para registrar las fechas y actividades en el plan de pruebas.....	291
Tabla 62 Plantilla para registrar los bancos de pruebas en el plan de pruebas.	291
Tabla 63 Plantilla para especificar los casos de prueba en el plan de pruebas.	292
Tabla 64 Escala de IADL de Lawton modificada (Gómez, Curcio, & Gómez, 1995).....	292
Tabla 65 Escala Clínica de Demencia (CDR) adaptada al idioma español (Mapi Research Institute, 2006).....	295
Tabla 66 Escala de Deterioro Global (GDS) adaptada al idioma español (Ruiz, Pérez, & Medina, 2013).	297
Tabla 67 Plantilla N°1 en la cual se detallan las columnas y los tipos de datos que debe contener el archivo CSV para poder ser leído correctamente por el sistema. Los nombres de las columnas deberán ser iguales a los encabezados de la plantilla. Las filas de	

registros deberán cumplir con los tipos de datos y los rangos establecidos para cada variable.....	297
Tabla 68 Plantilla N°2 en la cual se detallan las columnas y los tipos de datos que debe contener el archivo CSV para poder ser leído correctamente por el sistema. Los nombres de las columnas deberán ser iguales a los encabezados de la plantilla. Las filas de registros deberán cumplir con los tipos de datos y los rangos establecidos para cada variable.....	298
Tabla 69 Requerimientos incluidos en el plan de pruebas para el subsistema computacional tipo A.....	299
Tabla 70 Calendario de pruebas para el subsistema computacional tipo A.	299
Tabla 71 Bancos de pruebas para verificar los requerimientos del subsistema computacional tipo A.....	300
Tabla 72 Ejecución del caso de prueba N° 1 para verificar el requerimiento REQ-A01 — importación de variables—.....	301
Tabla 73 Ejecución del caso de prueba N° 2 para verificar el requerimiento REQ-A01 — importación de variables—.....	301
Tabla 74 Ejecución del caso de prueba N° 3 para verificar el requerimiento REQ-A01 — importación de variables—.....	302
Tabla 75 Ejecución del caso de prueba N° 4 para verificar el requerimiento REQ-A01 — importación de variables—.....	302
Tabla 76 Ejecución del caso de prueba N° 5 para verificar el requerimiento REQ-A02 — entrenamiento de árbol de decisión—.....	303
Tabla 77 Ejecución del caso de prueba N° 6 para verificar el requerimiento REQ-A02 — entrenamiento de árbol de decisión—.....	304
Tabla 78 Ejecución del caso de prueba N° 7 para verificar el requerimiento REQ-A03 — asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión—..	305

Tabla 79 Ejecución del caso de prueba N° 8 para verificar el requerimiento REQ-A03 — asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión — . .	305
Tabla 80 Ejecución del caso de prueba N° 9 para verificar el requerimiento REQ-A03 — asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión — . .	306
Tabla 81 Ejecución del caso de prueba N° 10 para verificar el requerimiento REQ-A03 — asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión — . .	306
Tabla 82 Ejecución del caso de prueba N° 11 para verificar el requerimiento REQ-A03 — asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión — . .	307
Tabla 83 Ejecución del caso de prueba N° 12 para verificar el requerimiento REQ-A03 — asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión — . .	307
Tabla 84 Re-ejecución del caso de prueba N° 1 para verificar el requerimiento REQ-A01 —importación de variables—.	308
Tabla 85 Re-ejecución del caso de prueba N° 2 para verificar el requerimiento REQ-A01 —importación de variables—.	308
Tabla 86 Re-ejecución del caso de prueba N° 3 para verificar el requerimiento REQ-A01 —importación de variables—.	308
Tabla 87 Re-ejecución del caso de prueba N° 4 para verificar el requerimiento REQ-A01 —importación de variables—.	309
Tabla 88 Re-ejecución del caso de prueba N° 5 para verificar el requerimiento REQ-A02 —entrenamiento de árbol de decisión—.	309
Tabla 89 Re-ejecución del caso de prueba N° 6 para verificar el requerimiento REQ-A02 —entrenamiento de árbol de decisión—.	309
Tabla 90 Re-ejecución del caso de prueba N° 7 para verificar el requerimiento REQ-A03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión—.....	309

Tabla 91 Re-ejecución del caso de prueba N° 8 para verificar el requerimiento REQ-A03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión—.....	310
Tabla 92 Re-ejecución del caso de prueba N° 9 para verificar el requerimiento REQ-A03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión—.....	310
Tabla 93 Re-ejecución del caso de prueba N° 10 para verificar el requerimiento REQ-A03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión—.....	310
Tabla 94 Re-ejecución del caso de prueba N° 11 para verificar el requerimiento REQ-A03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión—.....	311
Tabla 95 Re-ejecución del caso de prueba N° 12 para verificar el requerimiento REQ-A03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión—.....	311
Tabla 96 Ejecución del caso de prueba N° 13 para verificar el requerimiento REQ-A04 —persistencia de árbol de decisión—.....	312
Tabla 97 Ejecución del caso de prueba N° 14 para verificar el requerimiento REQ-A04 —persistencia de árbol de decisión—.....	312
Tabla 98 Ejecución del caso de prueba N° 15 para verificar el requerimiento REQ-A05 —cargue de árbol de decisión—.....	313
Tabla 99 Ejecución del caso de prueba N° 16 para verificar el requerimiento REQ-A05 —cargue de árbol de decisión—.....	313
Tabla 100 Re-ejecución del caso de prueba N° 1 para verificar el requerimiento REQ-A01 —importación de variables—.....	314
Tabla 101 Re-ejecución del caso de prueba N° 2 para verificar el requerimiento REQ-A01 —importación de variables—.....	314

Tabla 102 Re-ejecución del caso de prueba N° 3 para verificar el requerimiento REQ-A01 —importación de variables—	314
Tabla 103 Re-ejecución del caso de prueba N° 4 para verificar el requerimiento REQ-A01 —importación de variables—	314
Tabla 104 Re-ejecución del caso de prueba N° 5 para verificar el requerimiento REQ-A02 —entrenamiento de árbol de decisión—	315
Tabla 105 Re-ejecución del caso de prueba N° 6 para verificar el requerimiento REQ-A02 —entrenamiento de árbol de decisión—	315
Tabla 106 Re-ejecución del caso de prueba N° 7 para verificar el requerimiento REQ-A03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión—	315
Tabla 107 Re-ejecución del caso de prueba N° 8 para verificar el requerimiento REQ-A03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión—	316
Tabla 108 Re-ejecución del caso de prueba N° 9 para verificar el requerimiento REQ-A03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión—	316
Tabla 109 Re-ejecución del caso de prueba N° 10 para verificar el requerimiento REQ-A03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión—	316
Tabla 110 Re-ejecución del caso de prueba N° 11 para verificar el requerimiento REQ-A03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión—	316
Tabla 111 Re-ejecución del caso de prueba N° 12 para verificar el requerimiento REQ-A03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión—	317

Tabla 112 Re-ejecución del caso de prueba N° 13 para verificar el requerimiento REQ-A04 —persistencia de árbol de decisión—.....	317
Tabla 113 Re-ejecución del caso de prueba N° 14 para verificar el requerimiento REQ-A04 —persistencia de árbol de decisión—.....	317
Tabla 114 Re-ejecución del caso de prueba N° 15 para verificar el requerimiento REQ-A05 —cargue de árbol de decisión—.....	318
Tabla 115 Re-ejecución del caso de prueba N° 16 para verificar el requerimiento REQ-A05 —cargue de árbol de decisión—.....	318
Tabla 116 Ejecución del caso de prueba N° 17 para verificar el requerimiento REQ-A06 —desempeño de árbol de decisión—.....	319
Tabla 117 Ejecución del caso de prueba N° 18 para verificar el requerimiento REQ-A06 —desempeño de árbol de decisión—.....	320
Tabla 118 Requerimientos incluidos en el plan de pruebas para el subsistema computacional tipo B.....	320
Tabla 119 Calendario de pruebas para el subsistema computacional tipo B.....	321
Tabla 120 Bancos de pruebas para verificar los requerimientos del subsistema computacional tipo B.....	322
Tabla 121 Ejecución del caso de prueba N° 1 para verificar el requerimiento REQ-B01 —importación de variables—.....	322
Tabla 122 Ejecución del caso de prueba N° 2 para verificar el requerimiento REQ-B01 —importación de variables—.....	323
Tabla 123 Ejecución del caso de prueba N° 3 para verificar el requerimiento REQ-B01 —importación de variables—.....	323
Tabla 124 Ejecución del caso de prueba N° 4 para verificar el requerimiento REQ-B01 —importación de variables—.....	324

Tabla 125 Ejecución del caso de prueba N° 5 para verificar el requerimiento REQ-B02 — entrenamiento de red neuronal artificial—.	325
Tabla 126 Ejecución del caso de prueba N° 6 para verificar el requerimiento REQ-B02 — entrenamiento de red neuronal artificial—.	326
Tabla 127 Ejecución del caso de prueba N° 7 para verificar el requerimiento REQ-B03 — asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—.	326
Tabla 128 Ejecución del caso de prueba N° 8 para verificar el requerimiento REQ-B03 — asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—.	327
Tabla 129 Ejecución del caso de prueba N° 9 para verificar el requerimiento REQ-B03 — asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—.	327
Tabla 130 Ejecución del caso de prueba N° 10 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—.	328
Tabla 131 Ejecución del caso de prueba N° 11 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—.	329
Tabla 132 Ejecución del caso de prueba N° 12 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—.	329
Tabla 133 Re-ejecución del caso de prueba N° 1 para verificar el requerimiento REQ-B01 —importación de variables—.	329
Tabla 134 Re-ejecución del caso de prueba N° 2 para verificar el requerimiento REQ-B01 —importación de variables—.	330

Tabla 135 Re-ejecución del caso de prueba N° 3 para verificar el requerimiento REQ-B01 —importación de variables—	330
Tabla 136 Re-ejecución del caso de prueba N° 4 para verificar el requerimiento REQ-B01 —importación de variables—	330
Tabla 137 Re-ejecución del caso de prueba N° 5 para verificar el requerimiento REQ-B02 —entrenamiento de red neuronal artificial—	331
Tabla 138 Re-ejecución del caso de prueba N° 6 para verificar el requerimiento REQ-B02 —entrenamiento de red neuronal artificial—	331
Tabla 139 Re-ejecución del caso de prueba N° 7 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—	331
Tabla 140 Re-ejecución del caso de prueba N° 8 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—	331
Tabla 141 Re-ejecución del caso de prueba N° 9 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—	332
Tabla 142 Re-ejecución del caso de prueba N° 10 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—	332
Tabla 143 Re-ejecución del caso de prueba N° 11 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—	332
Tabla 144 Re-ejecución del caso de prueba N° 12 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—	333

Tabla 145 Ejecución del caso de prueba N° 13 para verificar el requerimiento REQ-B04 —persistencia de red neuronal artificial—.....	333
Tabla 146 Ejecución del caso de prueba N° 14 para verificar el requerimiento REQ-B04 —persistencia de red neuronal artificial—.....	334
Tabla 147 Ejecución del caso de prueba N° 15 para verificar el requerimiento REQ-B05 —cargue de red neuronal artificial—.....	334
Tabla 148 Ejecución del caso de prueba N° 16 para verificar el requerimiento REQ-B05 —cargue de red neuronal artificial—.....	335
Tabla 149 Re-ejecución del caso de prueba N° 1 para verificar el requerimiento REQ-B01 —importación de variables—.....	335
Tabla 150 Re-ejecución del caso de prueba N° 2 para verificar el requerimiento REQ-B01 —importación de variables—.....	336
Tabla 151 Re-ejecución del caso de prueba N° 3 para verificar el requerimiento REQ-B01 —importación de variables—.....	336
Tabla 152 Re-ejecución del caso de prueba N° 4 para verificar el requerimiento REQ-B01 —importación de variables—.....	336
Tabla 153 Re-ejecución del caso de prueba N° 5 para verificar el requerimiento REQ-B02 —entrenamiento de red neuronal artificial—.....	336
Tabla 154 Re-ejecución del caso de prueba N° 6 para verificar el requerimiento REQ-B02 —entrenamiento de red neuronal artificial—.....	337
Tabla 155 Re-ejecución del caso de prueba N° 7 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—.....	337
Tabla 156 Re-ejecución del caso de prueba N° 8 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—.....	337

Tabla 157 Re-ejecución del caso de prueba N° 9 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—.....	338
Tabla 158 Re-ejecución del caso de prueba N° 10 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—.....	338
Tabla 159 Re-ejecución del caso de prueba N° 11 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—.....	338
Tabla 160 Re-ejecución del caso de prueba N° 12 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—.....	338
Tabla 161 Re-ejecución del caso de prueba N° 13 para verificar el requerimiento REQ-B04 —persistencia de red neuronal artificial—.....	339
Tabla 162 Re-ejecución del caso de prueba N° 14 para verificar el requerimiento REQ-B04 —persistencia de red neuronal artificial—.....	339
Tabla 163 Re-ejecución del caso de prueba N° 15 para verificar el requerimiento REQ-B05 —cargue de red neuronal artificial—.....	340
Tabla 164 Re-ejecución del caso de prueba N° 16 para verificar el requerimiento REQ-B05 —cargue de red neuronal artificial—.....	340
Tabla 165 Ejecución del caso de prueba N° 17 para verificar el requerimiento REQ-B06 —desempeño de red neuronal artificial—.....	341
Tabla 166 Ejecución del caso de prueba N° 18 para verificar el requerimiento REQ-B06 —desempeño de red neuronal artificial—.....	341
Tabla 167 Requerimientos incluidos en el plan de pruebas para la integración de los subsistemas computacionales de tipo A y B en el sistema computacional híbrido.....	342

Tabla 168 Calendario de pruebas para la integración de los subsistemas computacionales de tipo A y B en el sistema computacional híbrido.....	342
Tabla 169 Bancos de pruebas para verificar los requerimientos de la integración de los subsistemas computacionales de tipo A y B en el sistema computacional híbrido.	343
Tabla 170 Ejecución del caso de prueba N° 1 para verificar el requerimiento REQ-C01 — entrenamiento de ensamble basado en Bagging—.	344
Tabla 171 Ejecución del caso de prueba N° 2 para verificar el requerimiento REQ-C01 — entrenamiento de ensamble basado en Bagging—.	345
Tabla 172 Ejecución del caso de prueba N° 3 para verificar el requerimiento REQ-C02 — asistencia al diagnóstico de Deterioro Cognitivo Leve mediante ensamble basado en Bagging —.	346
Tabla 173 Ejecución del caso de prueba N° 4 para verificar el requerimiento REQ-C02 — asistencia al diagnóstico de Deterioro Cognitivo Leve mediante ensamble basado en Bagging —.	346
Tabla 174 Ejecución del caso de prueba N° 5 para verificar el requerimiento REQ-C02 — asistencia al diagnóstico de Deterioro Cognitivo Leve mediante ensamble basado en Bagging—.	347
Tabla 175 Ejecución del caso de prueba N° 6 para verificar el requerimiento REQ-C02 — asistencia al diagnóstico de Deterioro Cognitivo Leve mediante ensamble basado en Bagging—.	347
Tabla 176 Ejecución del caso de prueba N° 7 para verificar el requerimiento REQ-C02 — asistencia al diagnóstico de Deterioro Cognitivo Leve mediante ensamble basado en Bagging—.	348
Tabla 177 Ejecución del caso de prueba N° 8 para verificar el requerimiento REQ-C02 — asistencia al diagnóstico de Deterioro Cognitivo Leve mediante ensamble basado en Bagging—.	349

Tabla 178 Ejecución del caso de prueba N° 9 para verificar el requerimiento REQ-C03 — persistencia de ensamble basado en Bagging—.	349
Tabla 179 Ejecución del caso de prueba N° 10 para verificar el requerimiento REQ-C03 —persistencia de ensamble basado en Bagging—.	350
Tabla 180 Ejecución del caso de prueba N° 11 para verificar el requerimiento REQ-C04 —cargue de ensamble basado en Bagging—.	351
Tabla 181 Ejecución del caso de prueba N° 12 para verificar el requerimiento REQ-C04 —cargue de ensamble basado en Bagging—.	351
Tabla 182 Ejecución del caso de prueba N° 13 para verificar el requerimiento REQ-C05 —desempeño de ensamble basado en Bagging—.	352
Tabla 183 Ejecución del caso de prueba N° 14 para verificar el requerimiento REQ-C05 —desempeño de ensamble basado en Bagging—.	353

LISTA DE ILUSTRACIONES

Ilustración 1 Propuesta de desarrollo para la actual investigación que consta de tres momentos: primero se recopilan los resultados de diversas pruebas de valoración cognitiva; segundo se construye un sistema computacional basado en técnicas simbólicas y subsimbólicas de aprendizaje de máquina; tercero se apoya el diagnóstico de Deterioro Cognitivo Leve en personas mayores de 60 años.	16
Ilustración 2 Clasificación de los subtipos clínicos de MCI con su presunta etiología. AD indica enfermedad de Alzheimer; DLB indica demencia con cuerpos de Lewy; FTD indica demencia frontotemporal; MCI indica Deterioro Cognitivo Leve; y VAD indica demencia vascular (Petersen R. C., 2005).	20
Ilustración 3 Diagrama de flujo para el diagnóstico de los subtipos clínicos de Deterioro Cognitivo Leve. En última instancia, este esquema conducirá a la clasificación dispuesta en la ilustración 2 (Petersen et al., 2005).	21
Ilustración 4 Área de cobertura del proyecto que involucra a 14 municipios del departamento de Caldas.....	35
Ilustración 5 Matriz de confusión para un sistema clasificador binario con 2 clases (positivo y negativo) y 4 tipos de resultados (verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos) (Berzal, 2016).	41
Ilustración 6 Visualización de cuatro curvas ROC con diferentes valores de AUC (OpenEye Scientific, 2015). El clasificador binario representado por la curva azul posee un mejor desempeño que los demás, debido a que su área bajo la curva ROC se encuentra más cercana a 1.	43
Ilustración 7 Proceso de una metodología de desarrollo orientada a prototipos que consta de seis etapas: definición de las metas del sistema, extracción de los requerimientos del sistema, análisis de los requerimientos del sistema, especificación gráfica o textual del sistema, validación del sistema y construcción de los prototipos del sistema (Wood & Kang, 1992).	45

Ilustración 8 Esquema de cross-validation o validación cruzada. Primero se segmenta el conjunto de datos en 2 subconjuntos: entrenamiento y validación. Posteriormente, se entrena el sistema con el primer subconjunto de datos. Por último, se valida la efectividad del sistema con el segundo subconjunto de datos (Cohen et al., 2010). 47

Ilustración 9 Técnica de validación cruzada con 10 iteraciones. En cada iteración se selecciona un subconjunto de datos de entrenamiento y validación diferente. Finalmente, se promedia la predicción de las 10 iteraciones para estimar el desempeño del sistema (McCormick, 2013). 47

Ilustración 10 Matriz de confusión referente a los estimados del clasificador Pasivo-Agresivo sobre la base de datos de la ADNI. 54

Ilustración 11 Matriz de confusión referente a los estimados del clasificador Pasivo-Agresivo sobre la base de datos de Caldas. 54

Ilustración 12 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del clasificador Pasivo-Agresivo sobre la base de datos de la ADNI. 55

Ilustración 13 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del clasificador Pasivo-Agresivo sobre la base de datos de la ADNI. 56

Ilustración 14 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del clasificador Pasivo-Agresivo sobre la base de datos de la ADNI. 56

Ilustración 15 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del clasificador Pasivo-Agresivo sobre la base de datos de Caldas. 57

Ilustración 16 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del clasificador Pasivo-Agresivo sobre la base de datos de Caldas. 57

Ilustración 17 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del clasificador Pasivo-Agresivo sobre la base de datos de Caldas.....	58
Ilustración 18 Matriz de confusión referente a los estimados del clasificador contraído sobre la base de datos de la ADNI.	60
Ilustración 19 Matriz de confusión referente a los estimados del clasificador contraído sobre la base de datos de Caldas.....	61
Ilustración 20 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del clasificador contraído sobre la base de datos de la ADNI.	62
Ilustración 21 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del clasificador contraído sobre la base de datos de la ADNI.	62
Ilustración 22 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del clasificador contraído sobre la base de datos de la ADNI.	63
Ilustración 23 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del clasificador contraído sobre la base de datos de Caldas.....	63
Ilustración 24 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del clasificador contraído sobre la base de datos de Caldas.	64
Ilustración 25 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del clasificador contraído sobre la base de datos de Caldas.	64

Ilustración 26 Estructura de una máquina de soporte vectorial que separa las clases + y - por medio de un hiperplano. Los puntos más cercanos al hiperplano se denominan vectores soporte (MathWorks, 2016).....	66
Ilustración 27 Matriz de confusión referente a los estimados de la SVM lineal sobre la base de datos de la ADNI.	67
Ilustración 28 Matriz de confusión referente a los estimados de la SVM lineal sobre la base de datos de Caldas.....	68
Ilustración 29 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la SVM lineal sobre la base de datos de la ADNI.	69
Ilustración 30 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la SVM lineal sobre la base de datos de la ADNI.	69
Ilustración 31 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la SVM lineal sobre la base de datos de la ADNI.	70
Ilustración 32 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la SVM lineal sobre la base de datos de Caldas.	70
Ilustración 33 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la SVM lineal sobre la base de datos de Caldas.	71
Ilustración 34 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la SVM lineal sobre la base de datos de Caldas.	71
Ilustración 35 Matriz de confusión referente a los estimados de la SVM cuadrática sobre la base de datos de la ADNI.	74

Ilustración 36 Matriz de confusión referente a los estimados de la SVM cuadrática sobre la base de datos de Caldas.....	74
Ilustración 37 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la SVM cuadrática sobre la base de datos de la ADNI.	75
Ilustración 38 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la SVM cuadrática sobre la base de datos de la ADNI.	76
Ilustración 39 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la SVM cuadrática sobre la base de datos de la ADNI.	76
Ilustración 40 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la SVM cuadrática sobre la base de datos de Caldas.	77
Ilustración 41 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la SVM cuadrática sobre la base de datos de Caldas.	77
Ilustración 42 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la SVM cuadrática sobre la base de datos de Caldas.	78
Ilustración 43 Matriz de confusión referente a los estimados de la SVM cúbica sobre la base de datos de la ADNI.	80
Ilustración 44 Matriz de confusión referente a los estimados de la SVM cúbica sobre la base de datos de Caldas.....	80
Ilustración 45 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la SVM cúbica sobre la base de datos de la ADNI.	81

Ilustración 46 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la SVM cúbica sobre la base de datos de la ADNI.	82
Ilustración 47 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la SVM cúbica sobre la base de datos de la ADNI.	82
Ilustración 48 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la SVM cúbica sobre la base de datos de Caldas.	83
Ilustración 49 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la SVM cúbica sobre la base de datos de Caldas.	83
Ilustración 50 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la SVM cúbica sobre la base de datos de Caldas.	84
Ilustración 51 Matriz de confusión referente a los estimados de la SVM Gaussiana de gran anchura sobre la base de datos de la ADNI.	86
Ilustración 52 Matriz de confusión referente a los estimados de la SVM Gaussiana de gran anchura sobre la base de datos de Caldas.	87
Ilustración 53 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la SVM Gaussiana de gran anchura sobre la base de datos de la ADNI.	88
Ilustración 54 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la SVM Gaussiana de gran anchura sobre la base de datos de la ADNI.	88

Ilustración 55 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la SVM Gaussiana de gran anchura sobre la base de datos de la ADNI.	89
Ilustración 56 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la SVM Gaussiana de gran anchura sobre la base de datos de Caldas.....	89
Ilustración 57 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la SVM Gaussiana de gran anchura sobre la base de datos de Caldas.....	90
Ilustración 58 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la SVM Gaussiana de gran anchura sobre la base de datos de Caldas.....	90
Ilustración 59 Matriz de confusión referente a los estimados de la SVM Gaussiana de mediana anchura sobre la base de datos de la ADNI.	92
Ilustración 60 Matriz de confusión referente a los estimados de la SVM Gaussiana de mediana anchura sobre la base de datos de Caldas.....	93
Ilustración 61 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la SVM Gaussiana de mediana anchura sobre la base de datos de la ADNI.	94
Ilustración 62 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la SVM Gaussiana de mediana anchura sobre la base de datos de la ADNI.	94
Ilustración 63 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la SVM Gaussiana de mediana anchura sobre la base de datos de la ADNI.	95

Ilustración 64 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la SVM Gaussiana de mediana anchura sobre la base de datos de Caldas.....	95
Ilustración 65 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la SVM Gaussiana de mediana anchura sobre la base de datos de Caldas.....	96
Ilustración 66 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la SVM Gaussiana de mediana anchura sobre la base de datos de Caldas.....	96
Ilustración 67 Matriz de confusión referente a los estimados de la SVM Gaussiana de pequeña anchura sobre la base de datos de la ADNI.	98
Ilustración 68 Matriz de confusión referente a los estimados de la SVM Gaussiana de pequeña anchura sobre la base de datos de Caldas.	99
Ilustración 69 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la SVM Gaussiana de pequeña anchura sobre la base de datos de la ADNI.	100
Ilustración 70 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la SVM Gaussiana de pequeña anchura sobre la base de datos de la ADNI.	100
Ilustración 71 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la SVM Gaussiana de pequeña anchura sobre la base de datos de la ADNI.	101
Ilustración 72 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la SVM Gaussiana de pequeña anchura sobre la base de datos de Caldas.....	101

Ilustración 73 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la SVM Gaussiana de pequeña anchura sobre la base de datos de Caldas.....	102
Ilustración 74 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la SVM Gaussiana de pequeña anchura sobre la base de datos de Caldas.....	102
Ilustración 75 Matriz de confusión referente a los estimados del árbol de decisión sobre la base de datos de la ADNI.	105
Ilustración 76 Matriz de confusión referente a los estimados del árbol de decisión sobre la base de datos de Caldas.....	106
Ilustración 77 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del árbol de decisión sobre la base de datos de la ADNI.	107
Ilustración 78 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del árbol de decisión sobre la base de datos de la ADNI.	107
Ilustración 79 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del árbol de decisión sobre la base de datos de la ADNI.	108
Ilustración 80 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del árbol de decisión sobre la base de datos de Caldas.	108
Ilustración 81 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del árbol de decisión sobre la base de datos de Caldas.....	109

Ilustración 82 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del árbol de decisión sobre la base de datos de Caldas.	109
Ilustración 83 Matriz de confusión referente a los estimados del bagging de árboles de decisión sobre la base de datos de la ADNI.	112
Ilustración 84 Matriz de confusión referente a los estimados del bagging de árboles de decisión sobre la base de datos de Caldas.	112
Ilustración 85 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del bagging de árboles de decisión sobre la base de datos de la ADNI.	113
Ilustración 86 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del bagging de árboles de decisión sobre la base de datos de la ADNI.	114
Ilustración 87 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del bagging de árboles de decisión sobre la base de datos de la ADNI.	114
Ilustración 88 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del bagging de árboles de decisión sobre la base de datos de Caldas.	115
Ilustración 89 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del bagging de árboles de decisión sobre la base de datos de Caldas.	115
Ilustración 90 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del bagging de árboles de decisión sobre la base de datos de Caldas.	116
Ilustración 91 Matriz de confusión referente a los estimados del boosting de árboles de decisión sobre la base de datos de la ADNI.	118

Ilustración 92 Matriz de confusión referente a los estimados del boosting de árboles de decisión sobre la base de datos de Caldas.	119
Ilustración 93 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del boosting de árboles de decisión sobre la base de datos de la ADNI.	120
Ilustración 94 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del boosting de árboles de decisión sobre la base de datos de la ADNI.	120
Ilustración 95 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del boosting de árboles de decisión sobre la base de datos de la ADNI.	121
Ilustración 96 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del boosting de árboles de decisión sobre la base de datos de Caldas.	121
Ilustración 97 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del boosting de árboles de decisión sobre la base de datos de Caldas.	122
Ilustración 98 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del boosting de árboles de decisión sobre la base de datos de Caldas.	122
Ilustración 99 Matriz de confusión referente a los estimados del AdaBoost-SAMME.R sobre la base de datos de la ADNI.	126
Ilustración 100 Matriz de confusión referente a los estimados del AdaBoost-SAMME.R sobre la base de datos de Caldas.	126
Ilustración 101 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del AdaBoost-SAMME.R sobre la base de datos de la ADNI.	127

Ilustración 102 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del AdaBoost-SAMME.R sobre la base de datos de la ADNI.	128
Ilustración 103 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del AdaBoost-SAMME.R sobre la base de datos de la ADNI.	128
Ilustración 104 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del AdaBoost-SAMME.R sobre la base de datos de Caldas.....	129
Ilustración 105 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del AdaBoost-SAMME.R sobre la base de datos de Caldas.	129
Ilustración 106 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del AdaBoost-SAMME.R sobre la base de datos de Caldas.	130
Ilustración 107 Matriz de confusión referente a los estimados del bosque aleatorio sobre la base de datos de la ADNI.	132
Ilustración 108 Matriz de confusión referente a los estimados del bosque aleatorio sobre la base de datos de Caldas.....	132
Ilustración 109 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del bosque aleatorio sobre la base de datos de la ADNI.	133
Ilustración 110 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del bosque aleatorio sobre la base de datos de la ADNI.	134

Ilustración 111 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del bosque aleatorio sobre la base de datos de la ADNI.	134
Ilustración 112 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del bosque aleatorio sobre la base de datos de Caldas.	135
Ilustración 113 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del bosque aleatorio sobre la base de datos de Caldas.	135
Ilustración 114 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del bosque aleatorio sobre la base de datos de Caldas.	136
Ilustración 115 Importancia de las variables en la base de datos de la ADNI. Cuanto más alta es la barra, más significativa es la variable para la clasificación de los datos.	137
Ilustración 116 Importancia de las variables en la base de datos de Caldas. Cuanto más alta es la barra, más significativa es la variable para la clasificación de los datos.	138
Ilustración 117 Matriz de confusión referente a los estimados del clasificador Extra-Trees sobre la base de datos de la ADNI.	140
Ilustración 118 Matriz de confusión referente a los estimados del clasificador Extra-Trees sobre la base de datos de Caldas.	140
Ilustración 119 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del clasificador Extra-Trees sobre la base de datos de la ADNI.	141
Ilustración 120 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del clasificador Extra-Trees sobre la base de datos de la ADNI.	142

Ilustración 121 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del clasificador Extra-Trees sobre la base de datos de la ADNI.	142
Ilustración 122 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del clasificador Extra-Trees sobre la base de datos de Caldas.	143
Ilustración 123 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del clasificador Extra-Trees sobre la base de datos de Caldas.	143
Ilustración 124 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del clasificador Extra-Trees sobre la base de datos de Caldas.	144
Ilustración 125 Matriz de confusión referente a los estimados del RUSBoost de árboles de decisión sobre la base de datos de la ADNI.	146
Ilustración 126 Matriz de confusión referente a los estimados del RUSBoost de árboles de decisión sobre la base de datos de Caldas.	146
Ilustración 127 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del RUSBoost de árboles de decisión sobre la base de datos de la ADNI.	147
Ilustración 128 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del RUSBoost de árboles de decisión sobre la base de datos de la ADNI.	148
Ilustración 129 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del RUSBoost de árboles de decisión sobre la base de datos de la ADNI.	148

Ilustración 130 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del RUSBoost de árboles de decisión sobre la base de datos de Caldas.	149
Ilustración 131 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del RUSBoost de árboles de decisión sobre la base de datos de Caldas.....	149
Ilustración 132 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del RUSBoost de árboles de decisión sobre la base de datos de Caldas.....	150
Ilustración 133 Matriz de confusión referente a los estimados del clasificador gradient boosting sobre la base de datos de la ADNI.	152
Ilustración 134 Matriz de confusión referente a los estimados del clasificador gradient boosting sobre la base de datos de Caldas.....	153
Ilustración 135 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del clasificador gradient boosting sobre la base de datos de la ADNI.	154
Ilustración 136 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del clasificador gradient boosting sobre la base de datos de la ADNI.	154
Ilustración 137 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del clasificador gradient boosting sobre la base de datos de la ADNI.	155
Ilustración 138 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del clasificador gradient boosting sobre la base de datos de Caldas.....	155

Ilustración 139 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del clasificador gradient boosting sobre la base de datos de Caldas.	156
Ilustración 140 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del clasificador gradient boosting sobre la base de datos de Caldas.	156
Ilustración 141 Principio de funcionamiento de una neurona artificial. Esta se encarga de sumar cada una de las entradas ponderadas y posteriormente pasa dicho resultado por una función de transferencia (Krenker, Bešter, & Kos, 2011).	158
Ilustración 142 Ejemplo de una red neuronal artificial de cuatro entradas, una salida y tres capas de neuronas artificiales —dos capas ocultas y una capa de salida—. Cada neurona artificial de cierta capa está interconectada con todas las neuronas artificiales de la siguiente capa (Krenker, Bešter, & Kos, 2011).	159
Ilustración 143 Estructura de la ANN – ELU Softmax Adam de tres capas: dos capas ocultas de neuronas artificiales con funciones de activación ELU y una capa de salida de neuronas artificiales con funciones de activación Softmax. La cantidad de neuronas θ es equivalente a la cantidad de entradas de la ANN.	163
Ilustración 144 Matriz de confusión referente a los estimados de la ANN - ELU Softmax Adam sobre la base de datos de la ADNI.	164
Ilustración 145 Matriz de confusión referente a los estimados de la ANN - ELU Softmax Adam sobre la base de datos de Caldas.	165
Ilustración 146 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - ELU Softmax Adam sobre la base de datos de la ADNI.	166
Ilustración 147 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - ELU Softmax Adam sobre la base de datos de la ADNI.	166

Ilustración 148 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - ELU Softmax Adam sobre la base de datos de la ADNI.....	167
Ilustración 149 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - ELU Softmax Adam sobre la base de datos de Caldas.....	167
Ilustración 150 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - ELU Softmax Adam sobre la base de datos de Caldas.	168
Ilustración 151 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - ELU Softmax Adam sobre la base de datos de Caldas.....	168
Ilustración 152 Estructura de la ANN – Hard Sigmoid Softmax Adam de tres capas: dos capas ocultas de neuronas artificiales con funciones de activación Sigmoidea Dura y una capa de salida de neuronas artificiales con funciones de activación Softmax.	170
Ilustración 153 Matriz de confusión referente a los estimados de la ANN - Hard Sigmoid Softmax Adam sobre la base de datos de la ADNI.....	171
Ilustración 154 Matriz de confusión referente a los estimados de la ANN - Hard Sigmoid Softmax Adam sobre la base de datos de Caldas.	171
Ilustración 155 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - Hard Sigmoid Softmax Adam sobre la base de datos de la ADNI.....	172
Ilustración 156 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - Hard Sigmoid Softmax Adam sobre la base de datos de la ADNI.....	173

Ilustración 157 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - Hard Sigmoid Softmax Adam sobre la base de datos de la ADNI.	173
Ilustración 158 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - Hard Sigmoid Softmax Adam sobre la base de datos de Caldas.	174
Ilustración 159 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - Hard Sigmoid Softmax Adam sobre la base de datos de Caldas.....	174
Ilustración 160 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - Hard Sigmoid Softmax Adam sobre la base de datos de Caldas.....	175
Ilustración 161 Estructura de la ANN – PReLU Softmax Adam de tres capas: dos capas ocultas de neuronas artificiales con funciones de activación PReLU y una capa de salida de neuronas artificiales con funciones de activación Softmax. La cantidad de neuronas θ es equivalente a la cantidad de entradas de la ANN.....	177
Ilustración 162 Matriz de confusión referente a los estimados de la ANN - PReLU Softmax Adam sobre la base de datos de la ADNI.....	178
Ilustración 163 Matriz de confusión referente a los estimados de la ANN - PReLU Softmax Adam sobre la base de datos de Caldas.	178
Ilustración 164 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - PReLU Softmax Adam sobre la base de datos de la ADNI.	179
Ilustración 165 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - PReLU Softmax Adam sobre la base de datos de la ADNI.	180

Ilustración 166 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - PReLU Softmax Adam sobre la base de datos de la ADNI.	180
Ilustración 167 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - PReLU Softmax Adam sobre la base de datos de Caldas.	181
Ilustración 168 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - PReLU Softmax Adam sobre la base de datos de Caldas.	181
Ilustración 169 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - PReLU Softmax Adam sobre la base de datos de Caldas.	182
Ilustración 170 Estructura de la ANN – ReLU Softmax Adadelata de tres capas: dos capas ocultas de neuronas artificiales con funciones de activación ReLU y una capa de salida de neuronas artificiales con funciones de activación Softmax.	184
Ilustración 171 Matriz de confusión referente a los estimados de la ANN - ReLU Softmax Adadelata sobre la base de datos de la ADNI.	185
Ilustración 172 Matriz de confusión referente a los estimados de la ANN - ReLU Softmax Adadelata sobre la base de datos de Caldas.	185
Ilustración 173 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - ReLU Softmax Adadelata sobre la base de datos de la ADNI.	186
Ilustración 174 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - ReLU Softmax Adadelata sobre la base de datos de la ADNI.	187

Ilustración 175 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - ReLU Softmax Adadelata sobre la base de datos de la ADNI.	187
Ilustración 176 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - ReLU Softmax Adadelata sobre la base de datos de Caldas.	188
Ilustración 177 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - ReLU Softmax Adadelata sobre la base de datos de Caldas.....	188
Ilustración 178 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - ReLU Softmax Adadelata sobre la base de datos de Caldas.....	189
Ilustración 179 Estructura de la ANN – ReLU Softmax Adadelata de tres capas: dos capas ocultas de neuronas artificiales con funciones de activación ReLU y una capa de salida de neuronas artificiales con funciones de activación Softmax.....	191
Ilustración 180 Matriz de confusión referente a los estimados de la ANN - ReLU Softmax Adagrad sobre la base de datos de la ADNI.....	192
Ilustración 181 Matriz de confusión referente a los estimados de la ANN - ReLU Softmax Adagrad sobre la base de datos de Caldas.	192
Ilustración 182 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - ReLU Softmax Adagrad sobre la base de datos de la ADNI.	193
Ilustración 183 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - ReLU Softmax Adagrad sobre la base de datos de la ADNI.	194

Ilustración 184 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - ReLU Softmax Adagrad sobre la base de datos de la ADNI.	194
Ilustración 185 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - ReLU Softmax Adagrad sobre la base de datos de Caldas.	195
Ilustración 186 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - ReLU Softmax Adagrad sobre la base de datos de Caldas.	195
Ilustración 187 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - ReLU Softmax Adagrad sobre la base de datos de Caldas.	196
Ilustración 188 Estructura de la ANN – ReLU Softmax Adadelata de tres capas: dos capas ocultas de neuronas artificiales con funciones de activación ReLU y una capa de salida de neuronas artificiales con funciones de activación Softmax.	198
Ilustración 189 Matriz de confusión referente a los estimados de la ANN - ReLU Softmax Adam sobre la base de datos de la ADNI.	199
Ilustración 190 Matriz de confusión referente a los estimados de la ANN - ReLU Softmax Adam sobre la base de datos de Caldas.	199
Ilustración 191 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - ReLU Softmax Adam sobre la base de datos de la ADNI.	200
Ilustración 192 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - ReLU Softmax Adam sobre la base de datos de la ADNI.	201

Ilustración 193 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - ReLU Softmax Adam sobre la base de datos de la ADNI.	201
Ilustración 194 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - ReLU Softmax Adam sobre la base de datos de Caldas.	202
Ilustración 195 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - ReLU Softmax Adam sobre la base de datos de Caldas.	202
Ilustración 196 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - ReLU Softmax Adam sobre la base de datos de Caldas.	203
Ilustración 197 Estructura de la ANN – ReLU Softmax Adam Deep de cinco capas: cuatro capas ocultas de neuronas artificiales con funciones de activación ReLU y una capa de salida de neuronas artificiales con funciones de activación Softmax.	205
Ilustración 198 Matriz de confusión referente a los estimados de la ANN - ReLU Softmax Adam Deep sobre la base de datos de la ADNI.	206
Ilustración 199 Matriz de confusión referente a los estimados de la ANN - ReLU Softmax Adam Deep sobre la base de datos de Caldas.	206
Ilustración 200 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - ReLU Softmax Adam Deep sobre la base de datos de la ADNI.	207
Ilustración 201 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - ReLU Softmax Adam Deep sobre la base de datos de la ADNI.	208

Ilustración 202 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - ReLU Softmax Adam Deep sobre la base de datos de la ADNI.	208
Ilustración 203 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - ReLU Softmax Adam Deep sobre la base de datos de Caldas.	209
Ilustración 204 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - ReLU Softmax Adam Deep sobre la base de datos de Caldas.....	209
Ilustración 205 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - ReLU Softmax Adam Deep sobre la base de datos de Caldas.....	210
Ilustración 206 Estructura de la ANN – ReLU Softmax Adamax de tres capas: dos capas ocultas de neuronas artificiales con funciones de activación ReLU y una capa de salida de neuronas artificiales con funciones de activación Softmax.....	212
Ilustración 207 Matriz de confusión referente a los estimados de la ANN - ReLU Softmax Adamax sobre la base de datos de la ADNI.	213
Ilustración 208 Matriz de confusión referente a los estimados de la ANN - ReLU Softmax Adamax sobre la base de datos de Caldas.....	213
Ilustración 209 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - ReLU Softmax Adamax sobre la base de datos de la ADNI.	214
Ilustración 210 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - ReLU Softmax Adamax sobre la base de datos de la ADNI.	215

Ilustración 211 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - ReLU Softmax Adamax sobre la base de datos de la ADNI.	215
Ilustración 212 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - ReLU Softmax Adamax sobre la base de datos de Caldas.	216
Ilustración 213 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - ReLU Softmax Adamax sobre la base de datos de Caldas.	216
Ilustración 214 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - ReLU Softmax Adamax sobre la base de datos de Caldas.	217
Ilustración 215 Estructura de la ANN – Sigmoid Softmax Adam de tres capas: dos capas ocultas de neuronas artificiales con funciones de activación Sigmoidea Dura y una capa de salida de neuronas artificiales con funciones de activación Softmax.	219
Ilustración 216 Matriz de confusión referente a los estimados de la ANN - Sigmoid Softmax Adam sobre la base de datos de la ADNI.	220
Ilustración 217 Matriz de confusión referente a los estimados de la ANN - Sigmoid Softmax Adam sobre la base de datos de Caldas.	220
Ilustración 218 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - Sigmoid Softmax Adam sobre la base de datos de la ADNI.	221
Ilustración 219 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - Sigmoid Softmax Adam sobre la base de datos de la ADNI.	222

Ilustración 220 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - Sigmoid Softmax Adam sobre la base de datos de la ADNI.	222
Ilustración 221 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - Sigmoid Softmax Adam sobre la base de datos de Caldas.	223
Ilustración 222 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - Sigmoid Softmax Adam sobre la base de datos de Caldas.	223
Ilustración 223 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - Sigmoid Softmax Adam sobre la base de datos de Caldas.	224
Ilustración 224 Estructura de la ANN – SReLU Softmax Adam de tres capas: dos capas ocultas de neuronas artificiales con funciones de activación SReLU y una capa de salida de neuronas artificiales con funciones de activación Softmax. La cantidad de neuronas θ es equivalente a la cantidad de entradas de la ANN.	226
Ilustración 225 Matriz de confusión referente a los estimados de la ANN - SReLU Softmax Adam sobre la base de datos de la ADNI.	227
Ilustración 226 Matriz de confusión referente a los estimados de la ANN - SReLU Softmax Adam sobre la base de datos de Caldas.	227
Ilustración 227 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - SReLU Softmax Adam sobre la base de datos de la ADNI.	228
Ilustración 228 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - SReLU Softmax Adam sobre la base de datos de la ADNI.	229

Ilustración 229 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - SReLU Softmax Adam sobre la base de datos de la ADNI.	229
Ilustración 230 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - SReLU Softmax Adam sobre la base de datos de Caldas.	230
Ilustración 231 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - SReLU Softmax Adam sobre la base de datos de Caldas.	230
Ilustración 232 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - SReLU Softmax Adam sobre la base de datos de Caldas.	231
Ilustración 233 Arquitectura de dos capas: una capa para las interfaces gráficas de usuario y el controlador de flujo, y otra capa para las entidades y utilidades del subsistema.	240
Ilustración 234 Diagrama de clases para el subsistema computacional tipo A. Este cuenta con tres interfaces gráficas de usuario: una para la asistencia al diagnóstico de Deterioro Cognitivo Leve, una para realizar el entrenamiento del árbol de decisión, y una para realizar la validación del árbol de decisión. Adicionalmente, cuenta con un controlador de flujo para conectar las peticiones del usuario con las respectivas acciones de las entidades y una serie de utilidades que facilitan el tratamiento de los datos. Finalmente, se definen algunas operaciones comunes a todos los clasificadores por medio de una clase abstracta para facilitar los llamados a sus métodos.	241
Ilustración 235 Menú principal del prototipo base del subsistema computacional tipo A. Este permite establecer el modo de la aplicación entre asistencia, entrenamiento y validación. Cada modo posee su propia interfaz gráfica de usuario la cual permite desarrollar una tarea específica dentro del subsistema computacional.	243

Ilustración 236 Modo de entrenamiento del prototipo base del subsistema computacional tipo A. Este permite entrenar un árbol de decisión para el diagnóstico asistido de Deterioro Cognitivo Leve seleccionando el tipo de base de datos, el archivo de base de datos y la opción de entrenamiento del subsistema computacional.	243
Ilustración 237 Modo de asistencia del prototipo base del subsistema computacional tipo A. Este permite conocer la probabilidad de que una persona padezca de Deterioro Cognitivo Leve o grave, o esté sana seleccionando el tipo de base de datos, las características de la persona y la opción de diagnóstico asistido.	244
Ilustración 238 Modo de asistencia del prototipo base del subsistema computacional tipo A. Nótese el cambio en las características de la persona de acuerdo con el tipo de base de datos seleccionado.	244
Ilustración 239 Menú principal del subsistema computacional tipo A. Este posee una opción adicional para guardar los árboles de decisión entrenados para el diagnóstico asistido de Deterioro Cognitivo Leve. Los árboles persistidos se cargan automáticamente en los usos posteriores del subsistema computacional.	245
Ilustración 240 Archivos físicos resultantes de la persistencia de los árboles de decisión en el subsistema computacional tipo A.	246
Ilustración 241 Modo de validación del prototipo base del subsistema computacional tipo A. Este permite conocer las métricas de desempeño de un árbol de decisión entrenado para el diagnóstico asistido de Deterioro Cognitivo Leve seleccionando el tipo de base de datos, la cantidad de pliegues de validación, el archivo de base de datos y la opción de validación del subsistema computacional. La columna ubicada al lado derecho muestra las métricas de desempeño resultantes de la validación cruzada con 10 iteraciones del árbol de decisión entrenado con la base de datos de Caldas.	247
Ilustración 242 Modo de validación del prototipo base del subsistema computacional tipo A. La columna ubicada al lado derecho muestra las métricas de desempeño resultantes de la validación cruzada con 10 iteraciones del árbol de decisión entrenado con la base de datos de la ADNI.	247

Ilustración 243 Diagrama de clases para el subsistema computacional tipo B. Este cuenta con tres interfaces gráficas de usuario: una para la asistencia al diagnóstico de Deterioro Cognitivo Leve, una para realizar el entrenamiento de la red neuronal, y una para realizar la validación de la red neuronal. Adicionalmente, cuenta con un controlador de flujo para conectar las peticiones del usuario con las respectivas acciones de las entidades y una serie de utilidades que facilitan el tratamiento de los datos. Finalmente, se definen algunas operaciones comunes a todos los clasificadores por medio de una clase abstracta para facilitar los llamados a sus métodos.....	252
Ilustración 244 Modo de entrenamiento del prototipo base del subsistema computacional tipo B. Este permite entrenar una red neuronal para el diagnóstico asistido de Deterioro Cognitivo Leve y posee una interfaz gráfica de usuario equivalente a la del subsistema computacional tipo A.	254
Ilustración 245 Modo de asistencia del prototipo base del subsistema computacional tipo B. Este permite conocer la probabilidad de que una persona padezca de Deterioro Cognitivo Leve o grave, o esté sana seleccionando el tipo de base de datos, las características de la persona y la opción de diagnóstico asistido. Igualmente, posee una interfaz gráfica de usuario equivalente a la del subsistema computacional tipo A.	254
Ilustración 246 Modo de asistencia del prototipo base del subsistema computacional tipo B. Nótese el cambio en las características de la persona de acuerdo con el tipo de base de datos seleccionado. Igualmente, posee una interfaz gráfica de usuario equivalente a la del subsistema computacional tipo A.	255
Ilustración 247 Menú principal del subsistema computacional tipo B. Este posee una opción adicional para guardar las redes neuronales entrenadas para el diagnóstico asistido de Deterioro Cognitivo Leve. Las redes neuronales persistidas se cargan automáticamente en los usos posteriores del subsistema computacional.	256
Ilustración 248 Archivos físicos resultantes de la persistencia de las redes neuronales en el subsistema computacional tipo B.	256

Ilustración 249 Modo de validación del prototipo base del subsistema computacional tipo B. Este permite conocer las métricas de desempeño de una red neuronal artificial entrenada para el diagnóstico asistido de Deterioro Cognitivo Leve seleccionando el tipo de base de datos, la cantidad de pliegues de validación, el archivo de base de datos y la opción de validación del subsistema computacional. Igualmente, posee una interfaz gráfica de usuario equivalente a la del subsistema computacional tipo A. La columna ubicada al lado derecho muestra las métricas de desempeño resultantes de la validación cruzada con 10 iteraciones de la red neuronal artificial entrenada con la base de datos de Caldas.	257
Ilustración 250 Modo de validación del prototipo base del subsistema computacional tipo B. La columna ubicada al lado derecho muestra las métricas de desempeño resultantes de la validación cruzada con 10 iteraciones de la red neuronal artificial entrenada con la base de datos de la ADNI.	258
Ilustración 251 Algoritmo de producción del ensamble híbrido compuesto de árboles de decisión y redes neuronales artificiales. Los mejores resultados de clasificación se obtienen al emplear cinco clasificadores de cada tipo (Hsu, 2012).	259
Ilustración 252 Mecanismo de predicción para el ensamble híbrido. Este permite someter a votación las predicciones individuales de los clasificadores base como es sugerido en el Bagging. En caso de existir un empate, la decisión se realiza por el promedio de las probabilidades de pertenecer a cada una de las clases.	260
Ilustración 253 Diagrama de clases para la hibridación de los subsistemas computacionales de tipo A y B. El mayor cambio radica en la inclusión de un nuevo tipo de clasificador para el sistema denominado Bagging. Este define un modelo compuesto por clasificadores de tipo DecisionTree y NeuralNetwork los cuales provienen de los subsistemas computacionales de tipo A y B construidos con anterioridad. Las demás clases se mantienen iguales a las diseñadas para los subsistemas computacionales. ..	263

Ilustración 254 Resultados de la validación cruzada con 10 iteraciones del sistema computacional híbrido para el diagnóstico asistido de Deterioro Cognitivo Leve respecto a la base de datos de Caldas.	264
Ilustración 255 Resultados de la validación cruzada con 10 iteraciones del sistema computacional híbrido para el diagnóstico asistido de Deterioro Cognitivo Leve respecto a la base de datos de la ADNI.	265
Ilustración 256 Resultados de la validación del sistema computacional híbrido para el diagnóstico asistido de Deterioro Cognitivo Leve respecto a un conjunto de datos inéditos para el sistema.	267
Ilustración 257 Evaluación Cognitiva de Montreal (MoCA) adaptada al idioma español (Nasreddine Z., 2004).	294

LISTA DE ABREVIACIONES

- **AD:** *Alzheimer's Disease*, Enfermedad de Alzheimer.
- **ADNI:** *Alzheimer's Disease Neuroimaging Initiative*.
- **ADTree:** *Alternating Decision Tree*, Árbol de Decisión Alternativo.
- **AI:** *Artificial Intelligence*, Inteligencia Artificial.
- **aMCI:** *Amnesic MCI*, Deterioro Cognitivo Leve Amnésico.
- **ANN:** *Artificial Neural Network*, Red Neuronal Artificial.
- **AODEsr:** *Averaged One-Dependence Estimators with Subsumption Resolution*.
- **AUC:** *Area Under Curve*, Área Bajo la Curva.
- **BADL:** *Basic Activities of Daily Living*, Actividades Básicas de la Vida Diaria.
- **BN:** *Bayesian Network*, Red Bayesiana.
- **CAD:** *Computer-Aided Diagnosis*, Diagnóstico Asistido por Computadora.
- **CDR:** *Clinical Dementia Rating*, Escala Clínica de Demencia.
- **CERAD:** *Consortium to Establish a Registry for Alzheimer's Disease*.
- **CI:** *Cognitive Impairment*, Deterioro Cognitivo.
- **DAMIP:** *Discriminant Analysis via Mixed Integer Programming*.
- **DLB:** *Dementia with Lewy Bodies*, Demencia con Cuerpos de Lewy.
- **EBN:** *Evolutionary Bayesian Network learning method*.
- **FN:** *False Negatives*, Falsos Negativos.
- **FP:** *False Positives*, Falsos Positivos.
- **FPR:** *False Positive Rate*, Tasa de Falsos Positivos.
- **FT:** *Functional Tree*, Árbol Funcional.
- **FTD:** *Frontotemporal Dementia*, Demencia Frontotemporal.
- **GAs:** *Genetic Algorithms*, Algoritmos Genéticos.
- **GDS:** *Global Deterioration Scale*, Escala de Deterioro Global.
- **HIV:** *Human Immunodeficiency Virus*, Virus de Inmunodeficiencia Humana.
- **IADL:** *Instrumental Activities of Daily Living*, Actividades Instrumentales de la Vida Diaria.

- **IBM:** *International Business Machines Corporation.*
- **LADTree:** *Logical Alternating Decision Tree, Árbol de Decisión Alternativo Lógico.*
- **MBK:** *Multifold Bayesian Kernelization.*
- **MCI:** *Mild Cognitive Impairment, Deterioro Cognitivo Leve.*
- **ML:** *Machine Learning, Aprendizaje de Máquina.*
- **MMSE:** *Mini Mental State Examination, Mini Examen del Estado Mental.*
- **MoCA:** *Montreal Cognitive Assessment, Evaluación Cognitiva Montreal.*
- **MRI:** *Magnetic Resonance Imaging, Imágenes por Resonancia Magnética.*
- **naMCI:** *Non-Amnestic MCI, Deterioro Cognitivo Leve No Amnésico.*
- **NB:** *Naïve Bayes, Bayes Ingenuo.*
- **PDP:** *Parallel Distributed Programming.*
- **PET:** *Positron Emission Tomography, Tomografía de Emisión de Positrones.*
- **PSO:** *Particle Swarm Optimization, Optimización por Enjambre de Partículas.*
- **PSSH:** *Physical Symbol System Hypothesis, Hipótesis del Sistema de Símbolos Físicos.*
- **qEEG:** *Quantitive Electroencephalography, Electroencefalografía Cuantitativa.*
- **ReLU:** *Rectifier Linear Unit, Unidad Lineal Rectificadora.*
- **RF:** *Random Forests, Bosques Aleatorios.*
- **ROC:** *Receiver Operating Characteristic, Característica Operativa del Receptor*
- **SC:** *Simple Cart (Classifier).*
- **SL:** *Simple Logistics, Regresión Logística Simple.*
- **SMO:** *Sequential Minimal Optimization, Optimización Secuencial Mínima.*
- **SVM:** *Support Vector Machine, Máquina de Soporte Vectorial.*
- **TIC:** *Tecnologías de la Información y la Comunicación.*
- **TN:** *True Negatives, Verdaderos Negativos.*
- **TP:** *True Positives, Verdaderos Positivos.*
- **TPR:** *True Positive Rate, Tasa de Verdaderos Positivos.*
- **UAM:** *Universidad Autónoma de Manizales.*
- **VaD:** *Vascular Dementia, Demencia Vascular.*

RESUMEN

El Deterioro Cognitivo Leve es una entidad nosológica de suma importancia; fue introducida para definir la situación clínica de declive o pérdida de las capacidades cognitivas que supone una fase previa a un trastorno demencial grave y que se estima no estar originado por el envejecimiento, sino por una patología subyacente. No obstante, la detección de dicho deterioro presenta dificultades en términos de costo monetario, tiempo y personal calificado en el proceso de diagnóstico, dado que este deterioro puede involucrar la toma de exámenes médicos de alta complejidad (ej. resonancias magnéticas) que a su vez requieren equipos clínicos especializados (ej. escáneres de resonancia magnética).

En el presente trabajo, con el fin de asistir al diagnóstico de Deterioro Cognitivo Leve, se construye un sistema computacional híbrido basado en técnicas simbólicas y subsimbólicas de aprendizaje de máquina; éste permite analizar los resultados de diferentes pruebas cognitivas para apoyar el dictamen realizado por los profesionales de la salud respecto al estado mental de las personas. El proceso de desarrollo del sistema computacional se lleva a cabo mediante una metodología orientada a prototipos evolutivos. Finalmente, se valida la efectividad en el diagnóstico asistido de Deterioro Cognitivo Leve a través de un esquema de *cross-validation* o validación cruzada.

Palabras clave: aprendizaje de máquina, Deterioro Cognitivo Leve, diagnóstico asistido, y pruebas cognitivas.

ABSTRACT

Mild cognitive impairment is a crucial nosological entity. It was introduced to define the clinical state of decline or loss of cognitive abilities which represents a preliminary stage to severe dementia disorders and is thought not to be caused by aging, but by an underlying pathology. However, diagnosis of such impairment is a challenging task facing difficulties in terms of monetary costs, time as well as finding qualified experts on this topic, since this may involve taking medical tests of high complexity (e.g. magnetic resonance imaging), which in turn require specialized clinical equipment (e.g. magnetic resonance imaging scanners).

In this work, in order to assist the diagnosis of mild cognitive impairment, a hybrid system based on symbolic and subsymbolic machine learning techniques is built. The proposed system will be able to analyze the results of different cognitive tests to support the decisions-making by the health staff service regarding the mental state of patients. Particularly, the system development process is conducted by an evolutionary prototyping methodology. Finally, the computer-aided effectiveness is validated through a cross-validation scheme.

Keywords: *aided diagnosis, cognitive tests, machine learning, and mild cognitive impairment.*

INTRODUCCIÓN

En el ámbito clínico, el diagnóstico temprano de deterioro cognitivo es de suma importancia, debido a que los tratamientos para ciertos tipos de demencias son más eficaces en las etapas iniciales (Haller, 2014). No obstante, la detección manual de deterioro cognitivo es intensiva en tiempo y puede requerir múltiples piezas de información (ej. resultados de diversas pruebas cognitivas). Estos datos se articulan conjuntamente para crear una imagen coherente de la discapacidad de la persona donde la eficiencia y la precisión son gobernadas por la experticia de un profesional de la salud. Además, el gasto monetario de un diagnóstico médico es a menudo una preocupación primordial, haciendo valiosa la búsqueda de métodos de diagnóstico más económicos que los tradicionales (Williams, Weakley, Cook, & Schmitter-Edgecombe, 2013).

El concepto de deterioro cognitivo como una afección atípica al proceso de envejecimiento normal ha existido en la literatura por muchos años. Este refiere diversos problemas que poseen las personas con determinadas funciones cognitivas como el pensamiento, el razonamiento, la memoria o la atención (Roy, 2013). En la década de 1990, se planteó el constructo de Deterioro Cognitivo Leve (*Mild Cognitive Impairment*, MCI) para designar un temprano (pero anormal) estado de deterioro cognitivo (Flicker, Ferris, & Reisberg, 1991; Petersen et al., 1995). Desde aquel momento, la comprensión de dicha patología ha requerido de grandes esfuerzos de indagación tanto desde la perspectiva clínica como desde la perspectiva investigativa (Petersen R. C., 2004).

Originalmente, el concepto de MCI destacó únicamente el deterioro de la memoria y acentuó su estatus como un estado precursor de la enfermedad de Alzheimer (McDade & Petersen, 2015). Posteriormente, se reconoció que el MCI podía ser diverso en términos de etiología, pronóstico, presentación clínica y prevalencia (Voisin, Touchon, & Vellas, 2003). Lo anterior permitió expandir el concepto de Deterioro Cognitivo Leve para incluir distintos dominios cognitivos, extendiendo la detección temprana de otras demencias en sus fases prodrómicas (McDade & Petersen, 2015).

Al igual que el concepto de MCI, el proceso de diagnóstico de dicha patología ha evolucionado a través del tiempo. Inicialmente, los profesionales de la salud recurrían a la aplicación de múltiples pruebas de valoración cognitiva (Gauthier et al., 2006). No obstante, los procesos modernos contemplan además la toma de exámenes físicos, exámenes de laboratorio e imágenes cerebrales (Langa & Levine, 2014). Pese a los avances logrados, el MCI presenta múltiples fuentes de heterogeneidad, lo cual demanda mayores esfuerzos en el desarrollo de alternativas que permitan facilitar el diagnóstico de esta condición (Mariani, Monastero, & Mecocci, 2007).

Recientemente, el sector salud se ha visto beneficiado de los avances logrados en las ciencias de la computación. Los datos clínicos se reúnen en muchos lugares, desde hospitales hasta estudios médicos. Sin embargo, esta gran cantidad de información es virtualmente estéril hasta ser organizada y analizada (University of Illinois, 2014). En consecuencia, han adquirido gran importancia los sistemas de diagnóstico asistido por computadora y los sistemas de salud inteligentes (Bramer, 2009), que proporcionan medios para mejorar la interpretación y manipulación de datos clínicos a través de inteligencia artificial (*Artificial Intelligence*, AI) y técnicas de aprendizaje de máquina.

El presente estudio explora el beneficio de la aplicación conjunta de diferentes técnicas de aprendizaje de máquina (*Machine Learning*, ML) para asistir al diagnóstico de Deterioro Cognitivo Leve en personas mayores de 60 años del departamento de Caldas. En consecuencia, se realiza la construcción de un sistema computacional híbrido inteligente capaz de analizar los resultados de múltiples pruebas de valoración cognitiva con el propósito de apoyar el dictamen de MCI en la población objetivo.

El proceso de desarrollo del sistema híbrido parte del análisis del desempeño de diferentes técnicas simbólicas y subsimbólicas de aprendizaje de máquina respecto a la detección de Deterioro Cognitivo Leve. Posteriormente, se realiza la construcción de diversos subsistemas computacionales basados en técnicas de ML bajo una metodología orientada a prototipos evolutivos. Finalmente, los subsistemas se integran y se valida la efectividad en el diagnóstico asistido mediante un esquema de *cross-validation*.

1 REFERENTE CONTEXTUAL

1.1 Descripción del área problemática

En el año 2014, la Universidad de Caldas en conjunto con la Universidad Autónoma de Manizales (apoyados por el Sistema General de Regalías de Colombia) emprendieron la ejecución del proyecto denominado **«Implementación del programa para diagnóstico y control de enfermedades crónicas no trasmisibles y cáncer de cérvix y mama, con el apoyo de TIC en el departamento de Caldas»**.

Dicha iniciativa tiene por objetivo conformar un programa de diagnóstico y control de cáncer de cérvix y mama, diabetes, hipertensión y deterioro cognitivo en los habitantes de Caldas. Al interior del proyecto, la Universidad Autónoma de Manizales es responsable de ejecutar las tareas relacionadas con el dictamen y tratamiento de deterioro cognitivo en personas mayores de 60 años, esencialmente aquellas afecciones relacionadas con el Deterioro Cognitivo Leve.

El diagnóstico de Deterioro Cognitivo Leve se efectúa mediante la aplicación de diferentes pruebas de valoración cognitiva como: CDR (Petersen et al., 1999), MoCA (Nasreddine et al., 2005), MMSE (Petersen et al., 2001b), entre otras. No obstante, la evolución tecnológica ha permitido el hallazgo de múltiples métodos favorables para la detección de MCI como el estudio de imágenes cerebrales por resonancia magnética (Jack et al., 1999), o el análisis de tomografías de emisión de positrones (Chatelat et al., 2003).

El problema fundamental de las técnicas mencionadas radica en el costo monetario significativamente elevado que implica la aplicación de éstas. Los centros de salud requieren adquirir, instalar y conservar hardware especializado mientras que los pacientes (principalmente adultos mayores) deben solventar los costos asociados a los exámenes médicos y a su desplazamiento hacia centros de salud de alta complejidad (Glover, 2014). Además, es importante señalar que la cantidad de médicos expertos en neurología es escasa, y la demanda de los mismos incrementa cada año (Jeffrey, 2013).

A pesar de los enormes avances logrados, el Deterioro Cognitivo Leve aún parece ser una entidad heterogénea en cuanto a etiología, pronóstico y presentación clínica (Libon et al., 2010; Winder-Rhodes, Barker, & Williams-Gray, 2015; Ye et al., 2015). El diagnóstico temprano de esta condición seguirá siendo cada vez más importante, tanto para apoyar el tratamiento de los síntomas como para evitar o retrasar el progreso a otras enfermedades subyacentes (Krinsky-McHale & Silverman, 2013). Todavía se requieren considerables esfuerzos para el desarrollo de clasificaciones diagnósticas más uniformes y mejores criterios de operación (Mariani, Monastero, & Mecocci, 2007).

1.2 Antecedentes

A través del tiempo, diversos investigadores han diseñado múltiples escalas de valoración neuropsicológica que permiten inspeccionar las capacidades cognitivas de las personas. Herramientas como el Mini Examen del Estado Mental MMSE (Folstein, Folstein, & McHugh, 1975), la Evaluación Cognitiva Montreal MoCA (Nasreddine et al., 2005), la Escala Clínica de Demencia CDR (Hughes, Berg, & Danziger, 1982) y la Escala de Deterioro Global GDS (Yesavage et al., 1982), son consideradas hoy en día como instrumentos fiables y válidos para la detección de deterioro cognitivo en sus diferentes estadios (Yin, Zhao, Lu, & Duan, 2015).

Dado que la aplicación conjunta de las herramientas mencionadas involucra procesos de larga duración, varios investigadores han optado por proponer alternativas para agilizar el diagnóstico de MCI. En consecuencia, han prosperado diversos estudios que involucran el análisis de imágenes (MRI o PET) para determinar la prevalencia de dicha condición en los seres humanos (Gauthier et al., 2006). Algunos signos como el hipometabolismo y la atrofia del lóbulo temporal medial han sido registrados en personas con Deterioro Cognitivo Leve en comparación con individuos cognitivamente normales (Chatelat et al., 2003; Korf, Wahlund, Visser, & Scheltens, 2004), incluso la presencia de estos posee un alto valor predictivo de progreso a otras enfermedades subsecuentes como la demencia (Stoub et al., 2005).

En los últimos años, el diagnóstico asistido por computadora (*Computer-Aided Diagnosis*, CAD) ha adquirido mayor interés en las comunidades académicas y de investigación; este permite apoyar a los médicos en la toma de decisiones respecto al diagnóstico de una enfermedad específica (Doi, 2007). La mayoría de los estudios que involucran CAD y MCI se centran en la búsqueda de las diferencias estructurales utilizando análisis de imágenes mediante algoritmos de aprendizaje supervisado (Yin, Zhao, Lu, & Duan, 2015). Como muestra de ello, Liu et al. (2013) presentaron un algoritmo denominado MBK para la detección asistida de MCI y AD. Esta técnica permite modelar el proceso de diagnóstico como un análisis de síntesis de biomarcadores obtenidos mediante el procesamiento de imágenes médicas (MRI y PET). Por otro lado, Suk y Shen (2013) plantearon un modelo computacional para el dictamen de MCI a partir de la clasificación de componentes extraídos de imágenes médicas (MRI y PET), utilizando máquinas de soporte vectorial. Posteriormente, Liu et al. (2014) diseñaron un método para asistir al diagnóstico de MCI que involucra el análisis de imágenes cerebrales (MRI) por medio de redes neuronales multicapa. Este método posee gran potencial en el campo del diagnóstico asistido por computadora puesto que demostró ser más efectivo que las técnicas tradicionales de clasificación binaria.

Aunque el CAD representa una importante línea de investigación, aún es difícil aplicarlo en entornos clínicos primarios debido al acceso limitado que poseen los equipos de captura de imágenes médicas (Yin, Zhao, Lu, & Duan, 2015). Este obstáculo ha favorecido el desarrollo de un método diferente para la detección de MCI, el cual opera de forma semejante a las técnicas de diagnóstico asistido por computadora pero no requiere el uso de imágenes médicas. Debido a la gran similitud que presenta la investigación actual en relación con los estudios relativos a dicho método, se decide realizar una descripción más detallada de cada uno de ellos.

Título: *«Diagnose the mild cognitive impairment by constructing Bayesian network with missing data».*

Autores: Sun, Y., Tang, Y., Shuxue, D., & Cui, Y.

Descripción: Sun et al. (2011) proponen un algoritmo denominado MNBN para asistir al diagnóstico de Deterioro Cognitivo Leve. Este usa la información de diversas características como edad, sexo, nivel de educación, MMSE, CDR, IADL, entre otras, para encontrar casos similares que permitan orientar el dictamen realizado por los médicos.

Dado que una base de datos real podría no ser completa en un principio, los autores emplean la información mutua y la interpolación de Newton para estimar los valores de los datos faltantes. Posteriormente, utilizan un filtro que permite obtener el orden adecuado de las características y, en último lugar, aplican dicho ordenamiento en la construcción de una red bayesiana (Sun, Tang, Ding, Lv, & Cui, 2011).

Para adquirir el conjunto de datos, los autores reclutaron participantes en el hospital de Xinhua (China). Se eligieron 45 personas sanas (entre 60 y 70 años de edad) y 42 pacientes con MCI (entre 60 y 76 años de edad) diagnosticados por un especialista en neurología según los criterios de Petersen. Los grupos fueron equilibrados en términos de género; se contó con la participación de un 58% y 55% de mujeres en cada uno de los dos grupos respectivamente (Sun, Tang, Ding, Lv, & Cui, 2011).

Los resultados indican que el MNBN, en la mayoría de condiciones, logra mejores resultados que determinados métodos (LibB, EBN) pero la eficiencia en tiempo es menor que otros algoritmos (Sun, Tang, Ding, Lv, & Cui, 2011). Además, el proceso de encontrar los casos similares solo computa la información mutua entre dos características. Los autores señalan que la exactitud del método podría mejorar si se contemplara la información mutua multivariada, pero esto aumentaría la complejidad del algoritmo. Por último, el uso de la interpolación de Newton exige que los valores de las características difieran unos de otros. Con el fin de reducir la complejidad, solo se computan ciertas medias de valores, lo cual repercute en el desempeño del método¹.

¹ El desempeño del método se mide con base a 3 criterios: número de datos faltantes, error cuadrático medio y tiempo de ejecución promedio.

Título: «*Machine Learning Approaches for the Computer Aided Diagnosis and Prediction of Alzheimer's Disease based on Clinical Data*».

Autor: Umer, R.

Descripción: Umer (2011) divide su investigación en dos fases. La primera abarca el análisis del desempeño de diferentes métodos de aprendizaje de máquina en la tarea de clasificar el estado cognitivo de múltiples pacientes (normal, MCI o AD). Durante esta etapa se llevan a cabo las siguientes tareas: 1. Comparación de las características de rendimiento de diversos algoritmos con base en la exactitud de clasificación. 2. Eliminación de atributos redundantes mediante métodos de selección de factores. 3. Comparación de las características de rendimiento de los clasificadores con y sin reducción de atributos.

La segunda fase comprende la predicción de la probabilidad que un paciente progrese de MCI a AD. Durante esta etapa se llevan a cabo las siguientes tareas: 1. Comparación de las características de rendimiento de diversos clasificadores de predicción con base en sensibilidad y especificidad. 2. Análisis del costo generado por el balance entre especificidad y sensibilidad. A continuación, se eligen los clasificadores de predicción *Logical Alternating Decision Tree* (LADTree), *Alternating Decision Tree* (ADTree) y *Functional Tree* (FT) debido a que presentan mayor desempeño de acuerdo a un análisis de área bajo la curva ROC realizado por el autor.

La información empleada en esta investigación se obtuvo de la base de datos de la ADNI. En dicha iniciativa se reclutaron 800 personas entre 55 y 99 años de edad, de las cuales 200 eran cognitivamente sanas, 400 habían sido diagnosticadas con MCI y 200 presentaban principios de AD. Posteriormente, se usaron medidas de biomarcadores para realizar un seguimiento de los cambios cerebrales de los sujetos, a fin de detectar el AD en una etapa inicial (Umer, 2011).

El estudio de Umer muestra que es posible mejorar el diagnóstico de Deterioro Cognitivo Leve y Alzheimer usando algoritmos de aprendizaje de máquina (Umer, 2011). Para la predicción de la probabilidad de progreso de MCI a AD, se obtienen resultados

prometedores. Los investigadores consiguen un 87% de sensibilidad, pero aún existe un balance entre especificidad y sensibilidad que debe ser controlado usando clasificadores sensibles al costo. Finalmente, el investigador señala que dichos resultados son muy útiles para predecir el futuro diagnóstico de los pacientes con MCI que, o bien se mantendrán MCI o se convertirán a AD.

Título: «*Predictive Model for Early Detection of Mild Cognitive Impairment and Alzheimer's Disease*».

Autores: Lee, E., Wu, T., Goldstein, F., & Levey, A.

Descripción: Lee et al. (2012) plantean un modelo basado en técnicas de aprendizaje de máquina para la detección temprana de Deterioro Cognitivo Leve y enfermedad de Alzheimer. Las características contempladas en la construcción del modelo se obtienen a partir de cuatro pruebas cognitivas: MMSE, test del dibujo del reloj, memoria de lista de palabras del CERAD (*Consortium to Establish a Registry for Alzheimer's Disease*) y escala de depresión geriátrica.

El modelo consta de dos componentes principales: el primero se encarga de extraer un conjunto reducido de características discriminatorias mediante la técnica de optimización PSO (*Particle Swarm Optimization*); el segundo componente permite crear una regla de clasificación a través del método DAMIP (*Discriminant Analysis via Mixed Integer Programming*) para separar los pacientes en tres grupos: normal, MCI y AD. La validación del modelo se realiza por medio de la técnica de validación cruzada con 10 pliegues (Lee, Wu, Goldstein, & Levey, 2012).

Para desarrollar la investigación se recolectaron datos de dos estudios clínicos del Centro de Investigación de la Enfermedad de Alzheimer de la Universidad de Emory. El primero contó con la participación de 10 pacientes (5 con AD, 3 con MCI y 2 asignados al grupo control); el segundo contó con la participación de 25 pacientes (2 con AD, 13 con MCI y 10 asignados al grupo de control), para un total de 35 personas.

Empleando dos estudios clínicos de pacientes con Alzheimer, Deterioro Cognitivo Leve, y los grupos de control, los autores muestran que es posible desarrollar una regla de clasificación basada en los datos de diversas pruebas cognitivas para predecir AD y MCI. Además, la investigación sugiere fuertemente que los datos en bruto de las pruebas cognitivas tienen un mayor potencial para diferenciar los grupos de AD, MCI y control que los puntajes totales de las pruebas usados como características (Lee, Wu, Goldstein, & Levey, 2012).

Título: «*Machine Learning Techniques for Diagnostic Differentiation of Mild Cognitive Impairment and Dementia*».

Autores: Williams, J., Weakley, A., Cook, D., & Schmitter-Edgecombe, M.

Descripción: Williams et al. (2013) exploran el uso de datos demográficos y de pruebas cognitivas para predecir los puntajes de la Escala Clínica de Demencia (CDR) y los diagnósticos clínicos de los pacientes (cognitivamente sano, Deterioro Cognitivo Leve, demencia), a través de la implementación de cuatro algoritmos de aprendizaje de máquina: «*Naïve-Bayes*» (NB), árbol de decisión (DT), red neuronal artificial (ANN), y máquina de soporte vectorial (SVM).

En el estudio se emplean 3 conjuntos de datos diferentes; 1 conjunto de datos para predecir el diagnóstico clínico y dos conjuntos de datos para predecir los puntajes del CDR. El primer conjunto incluye 53 pacientes con demencia (52 participantes con sospecha de AD, 1 con sospecha de DLB), 97 pacientes con MCI, y 161 adultos mayores cognitivamente sanos (grupo de control). Los dos conjuntos restantes contienen 154 pacientes que recibieron un CDR igual a 0, 93 pacientes con CDR igual a 0,5, y 25 pacientes con CDR igual a 1 o a 2. Los participantes tenían edades comprendidas entre los 48 y 95 años (Williams, Weakley, Cook, & Schmitter-Edgecombe, 2013).

Por último, Williams et al. (2013) concluyen que las técnicas de aprendizaje de máquina pueden usarse para automatizar los aspectos del diagnóstico clínico de pacientes con

deterioro cognitivo. Sin embargo, también señalan que además de los cuatro modelos explorados en el estudio (NB, DT, ANN, y SVM), se deben explorar otros métodos conjuntos. Estos aprovecharían las ventajas de múltiples clasificadores y podrían obtener un mejor rendimiento que el logrado por un solo clasificador (Williams, Weakley, Cook, & Schmitter-Edgecombe, 2013).

Título: *«A Hybrid Intelligent Diagnosis Approach for Quick Screening of Alzheimer's Disease Based on Multiple Neuropsychological Rating Scales».*

Autores: Yin, Z., Zhao, Y., Lu, X., & Duan, H.

Descripción: Yin et al. (2015) proponen un enfoque híbrido inteligente para apoyar la detección de AD y MCI a partir del análisis computacional de los resultados de diversas pruebas cognitivas como MoCA, MMSE, entre otras. Este método involucra 2 etapas: la primera comprende una técnica de reducción de atributos basada en algoritmos genéticos. La segunda involucra la aplicación de técnicas de razonamiento incierto para pronosticar la probabilidad de sufrir Alzheimer. Igualmente, los investigadores ponen a prueba diferentes técnicas de clasificación como BN, C4.5, SMO y ANN (antes y después de la reducción de atributos) para determinar cuáles poseen mayor efectividad en el diagnóstico asistido de AD y MCI.

El conjunto de datos experimentales del estudio se compone de 500 casos históricos recogidos entre 2009 y 2014 por el departamento de neurología de un reconocido hospital de China. Cada caso comprende una serie de puntuaciones pertenecientes a las evaluaciones cognitivas de un sujeto; cada sujeto posee un solo caso. Todos los exámenes cognitivos fueron realizados por diversos especialistas entrenados en neuropsicología. Las edades de las personas valoradas se encuentran entre los 51 y 92 años y el 59,5% son mujeres (Yin, Zhao, Lu, & Duan, 2015).

Al finalizar el estudio los autores concluyen que aun cuando es factible realizar el diagnóstico asistido de MCI mediante el análisis computacional de las pruebas de

valoración cognitiva, todavía existen varios aspectos por mejorar. Respecto a dicha declaración Yin et al. (2015) señalan que es necesario realizar pruebas de efectividad con un conjunto más amplio de pruebas cognitivas, con el fin de mejorar la exactitud en el diagnóstico asistido de enfermedades específicas de deterioro cognitivo. Además, resaltan la necesidad de implementar diferentes estrategias de diagnóstico según el nivel de complejidad de cada hospital. Las herramientas más sencillas de valoración cognitiva ayudarían a los médicos en centros de salud de atención primaria a detectar los pacientes que sufren de deterioro cognitivo; las herramientas avanzadas permitirían a los médicos en clínicas de segunda clase a estimar la gravedad del deterioro; finalmente, las herramientas CAD podrían ser usadas en los hospitales más especializados para determinar el grado y tipo de deterioro cognitivo.

Si bien cada estudio plantea un paradigma particular para el diagnóstico asistido de MCI, la teoría que respalda los campos de investigación del CAD y de las técnicas de aprendizaje de máquina hace posible el hallazgo de regularidades que respaldan el devenir del proyecto. Puntualmente es de gran interés observar los mecanismos de elección e implementación de los algoritmos empleados en el análisis de datos (MBK, SVM, BN, otros), debido que la efectividad en el diagnóstico asistido de Deterioro Cognitivo Leve depende en su mayoría de estos componentes (Umer, 2011).

A continuación se presenta una síntesis de los métodos más relevantes observados en la revisión de antecedentes, con su respectiva finalidad y exactitud:

Autor	Método	Finalidad	Exactitud
Sun et al. (2011)	BN	Encontrar casos similares	82%
Umer (2011)	SL	Clasificar AD y MCI	88,78%
Umer (2011)	FURIA	Clasificar AD y MCI	88,41%
Umer (2011)	SC	Clasificar AD y MCI	87,35%
Umer (2011)	RF	Clasificar AD y MCI	86,96%
Umer (2011)	SMO	Clasificar AD y MCI	85,91%
Umer (2011)	AODEsr	Clasificar AD y MCI	84%

Umer (2011)	BN	Clasificar AD y MCI	82,79%
Lee et al. (2012)	PSO + DAMIP	Clasificar AD y MCI	69%
Williams et al. (2013)	NB	Predecir diagnóstico clínico	83,3%
Williams et al. (2013)	ANN	Predecir diagnóstico clínico	82%
Williams et al. (2013)	DT	Predecir diagnóstico clínico	78,5%
Williams et al. (2013)	SVM	Predecir diagnóstico clínico	74,3%
Yin et al. (2015)	GAs + BN	Clasificar AD y MCI	83,57%

Tabla 1 Síntesis de los métodos divisados en la revisión de antecedentes para el diagnóstico asistido de MCI.

1.3 Justificación

Se estima que para el año 2050 la población mundial mayor de 60 años será aproximadamente de 2 billones de personas. El envejecimiento de la población está ocurriendo rápidamente en los países de medios y bajos ingresos. En consecuencia, se evidencia un aumento significativo en el número de personas que padecen enfermedades de deterioro cognitivo como la demencia (World Health Organization, 2012).

La demencia es un síndrome, normalmente de naturaleza crónica o progresiva, causado por una gran variedad de enfermedades cerebrales que afectan la memoria, el pensamiento, el comportamiento y la capacidad de realizar actividades diarias (Duthey, 2013). Las deficiencias de las funciones cognitivas son acompañadas, y en ocasiones precedidas, por el deterioro del control emocional, la conducta social o la motivación (World Health Organization, 2012).

El Deterioro Cognitivo Leve es una entidad nosológica con múltiples fuentes de heterogeneidad, incluyendo etiología, pronosis y presentación clínica (Mariani, Monastero, & Mecocci, 2007). Esta condición se propuso como un estado intermedio entre el envejecimiento normal y las enfermedades graves de deterioro cognitivo. De hecho, más de la mitad de personas que poseen MCI progresan a la demencia al cabo de 5 años (Gauthier et al., 2006).

Algunas cuestiones relacionadas con el dictamen de Deterioro Cognitivo Leve todavía se encuentran latentes (Mariani, Monastero, & Mecocci, 2007). El proceso de diagnóstico requiere primero evaluar las funciones cognitivas de la persona. Esto usualmente se logra mediante la realización de un examen de estatus mental, posiblemente complementado por el uso de evaluaciones neuropsicológicas. Finalmente, la etiología de los síntomas debe ser elucidada sobre la base de la evaluación clínica, las pruebas de laboratorio y los estudios de imágenes neurológicas. Muchas de estas etapas son aún objeto de debate (Werner & Korczyn, 2008).

Debido a la falta de instalaciones médicas avanzadas (hardware para la captura de imágenes neurológicas), el diagnóstico de MCI y otras enfermedades de deterioro cognitivo depende principalmente del uso de escalas de valoración neuropsicológica en clínicas de atención primaria (Yin, Zhao, Lu, & Duan, 2015). Este proceso de dictamen basado en pruebas cognitivas, requiere personal altamente capacitado (Zadikoff et al., 2008). Sin embargo, la mayoría de los médicos no están calificados para llevar a cabo un examen completo del estatus mental o interpretar los puntajes de las escalas de valoración neuropsicológica; para ellos es difícil emitir juicios exactos acerca del estado cognitivo del paciente (Borson et al., 2007). Lo anterior respalda la necesidad de continuar explorando alternativas para asistir y facilitar el diagnóstico de Deterioro Cognitivo Leve, en el caso actual desde la perspectiva de los sistemas computacionales inteligentes.

Por último, es importante señalar que el proyecto financiado por el sistema general de regalías «**Implementación del programa para diagnóstico y control de enfermedades crónicas no transmisibles y cáncer de cérvix y mama, con el apoyo de TIC en el departamento de Caldas**» acoge a los habitantes de 14 municipios de Caldas que pertenecen al régimen subsidiado de salud y son atendidos en hospitales de baja complejidad, es decir que dicha iniciativa permite favorecer primordialmente a personas y centros de salud de escasos recursos.

1.4 Formulación de la propuesta de investigación

En concordancia con la evolución de las técnicas de apoyo a la detección de Deterioro Cognitivo Leve, se plantea la construcción de un sistema computacional para asistir al diagnóstico de dicha patología a partir del análisis de los resultados de diferentes pruebas de valoración cognitiva, mediante un enfoque híbrido inteligente.

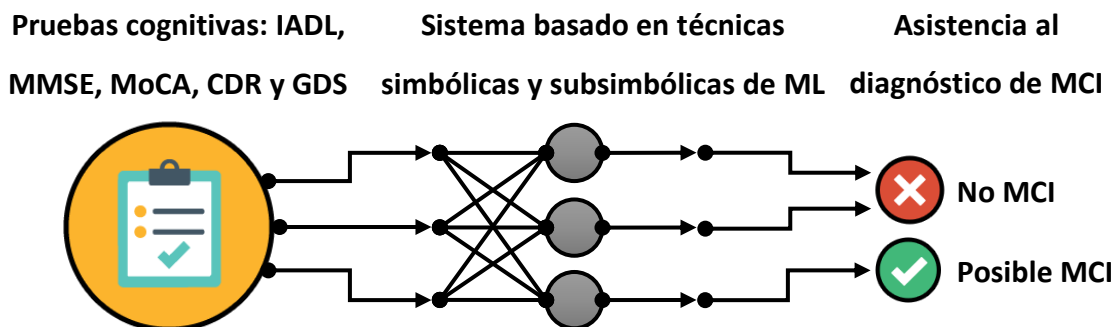


Ilustración 1 Propuesta de desarrollo para la actual investigación que consta de tres momentos: primero se recopilan los resultados de diversas pruebas de valoración cognitiva; segundo se construye un sistema computacional basado en técnicas simbólicas y subsimbólicas de aprendizaje de máquina; tercero se apoya el diagnóstico de Deterioro Cognitivo Leve en personas mayores de 60 años.

Para simplificar la consecución de la propuesta, se opta por segmentar el desarrollo de la misma en 3 momentos diferentes:

1. **Pruebas cognitivas:** comprende la recopilación de los resultados de cinco pruebas de valoración cognitiva aplicadas a personas mayores de 60 años. Asimismo, involucra el compendio de los datos personales (datos básicos, datos demográficos e información socio-económica) de dichas personas y los diagnósticos brindados por los especialistas respecto a su condición mental.
2. **Sistema basado en técnicas de ML:** comprende un estudio comparativo de diversas técnicas de aprendizaje de máquina y la construcción de un sistema computacional basado en técnicas de ML para asistir a los profesionales de la salud en el proceso de diagnóstico de Deterioro Cognitivo Leve. Los datos de entrada requeridos en el estudio comparativo y la construcción del sistema computacional provienen de la recopilación de los resultados de las pruebas de valoración cognitiva.

3. Asistencia al diagnóstico de MCI: comprende la puesta en producción del sistema computacional al interior del proyecto «**Implementación del programa para diagnóstico y control de enfermedades crónicas no transmisibles y cáncer de cérvix y mama, con el apoyo de TIC en el departamento de Caldas**». Este podrá ser utilizado por los profesionales de la salud para apoyar el proceso de diagnóstico de Deterioro Cognitivo Leve en personas mayores de 60 años del departamento de Caldas. No obstante, será posible extender su uso a otras poblaciones.

1.5 Objetivo general

Construir un sistema computacional basado en técnicas simbólicas y subsimbólicas de aprendizaje de máquina que permita el diagnóstico asistido de Deterioro Cognitivo Leve en personas mayores de 60 años.

1.6 Objetivos específicos

- Analizar diferentes técnicas simbólicas y subsimbólicas de aprendizaje de máquina mediante un estudio comparativo, para determinar aquellas que poseen mejor desempeño en la asistencia al diagnóstico de Deterioro Cognitivo Leve.
- Construir subsistemas computacionales basados en técnicas simbólicas y subsimbólicas de aprendizaje de máquina que permitan el diagnóstico asistido de Deterioro Cognitivo Leve en personas mayores de 60 años.
- Integrar los subsistemas computacionales basados en técnicas simbólicas y subsimbólicas de aprendizaje de máquina para obtener un sistema computacional híbrido.
- Establecer un esquema de *cross-validation* que permita validar la efectividad del sistema computacional híbrido en el diagnóstico asistido de Deterioro Cognitivo Leve.

2 REFERENTE TEÓRICO

2.1 Deterioro cognitivo

El deterioro cognitivo es un problema frecuente principalmente evidenciado en adultos mayores, el cual tiene un gran impacto en la calidad de vida, el funcionamiento diario y el cuidado de la salud (Lebedev, 2014). Esta condición se refleja cuando una persona posee problemas para recordar, aprender nuevas cosas, concentrarse o tomar decisiones que afectan su vida cotidiana (Centers for Disease Control and Prevention, 2011).

El CI varía de leve a severo. Con un deterioro leve las personas comienzan a notar cambios en sus funciones cognitivas, pero aún son capaces de realizar sus actividades cotidianas. Por el contrario, los niveles moderado y severo de deterioro pueden conducir a perder la capacidad para comprender el significado o importancia de algo y la habilidad de hablar o escribir que finalmente resulta en la incapacidad para vivir de forma independiente (Centers for Disease Control and Prevention, 2011).

2.2 Deterioro cognitivo leve

El Deterioro Cognitivo Leve se define como una pérdida cognitiva mayor a la esperada para la edad y el nivel de educación de un individuo, que no interfiere notablemente con las actividades cotidianas de la persona. Por lo tanto, el MCI es distinto de la demencia en la que los déficits cognitivos son más severos y tienen un efecto sustancial en la realización de las funciones diarias (Gauthier et al., 2006).

La Asociación de Alzheimer (2009) refiere:

El Deterioro Cognitivo Leve es un término general que comúnmente define un desorden sutil pero perceptible de la memoria. Una persona con MCI padece problemas de memoria que suelen ser peores que los que normalmente tienen otras personas de su edad, pero no muestra ningún otro síntoma de demencia, como puede ser la disminución del juicio o del razonamiento.

De igual forma, la Academia Americana de Psiquiatría (2013) sugiere considerar el MCI como una entidad caracterizada por el compromiso predominante, aunque no exclusivo, de la memoria en ausencia de un trastorno funcional. Además, el manual de diagnóstico y estadística de los trastornos mentales 5ª ed. (DSM-5) cuya autoría corresponde a dicha organización, incorpora la presencia de alteraciones en uno o varios dominios cognitivos tales como: atención compleja, función ejecutiva, aprendizaje y memoria, lenguaje, entre otros (American Psychiatric Association, 2013). Es importante señalar que el MCI no altera la función social u ocupacional, pero si puede llegar a comprometer el desempeño en tareas avanzadas (Ruiz, Pérez, & Medina, 2013).

En la actualidad, se han definido cuatro subtipos de Deterioro Cognitivo Leve: 1. MCI amnésico de único dominio, caracterizado por el deterioro aislado de la memoria; 2. MCI amnésico de múltiple dominio, caracterizado por el deterioro de varios dominios cognitivos incluyendo la memoria; 3. MCI no amnésico de único dominio, caracterizado por el deterioro aislado de un dominio cognitivo diferente de la memoria; 4. MCI no amnésico de múltiple dominio, caracterizado por el deterioro de varios dominios cognitivos, pero sin déficits de memoria. Los 2 primeros subtipos pertenecen al grupo denominado MCI amnésico (aMCI), mientras que los 2 últimos pertenecen al grupo denominado MCI no amnésico (naMCI) (Petersen R. C., 2004).

De acuerdo con los subtipos clínicos de MCI propuestos, es concebible que difieran en etiología y consecuencia (ver ilustración 2). De hecho, se considera que los subtipos del grupo aMCI poseen una etiología degenerativa, lo cual conlleva a que presenten una alta probabilidad de progresión a la enfermedad de Alzheimer, mientras que los subtipos del grupo naMCI tienden a progresar con mayor frecuencia a una demencia no-Alzheimer como la demencia de cuerpos de Lewy (Petersen et al., 2001a; Petersen R. C., 2004; Petersen & Morris, 2005).

Clasificación Clínica		Etiología			
		Degenerativa	Vascular	Psiquiátrica	Condiciones Médicas
MCI Amnésico	Único Dominio	AD		Depresión	
	Múltiple Dominio	AD	VaD	Depresión	
MCI No Amnésico	Único Dominio	FTD			
	Múltiple Dominio	DLB	VaD		

Ilustración 2 Clasificación de los subtipos clínicos de MCI con su presunta etiología. AD indica enfermedad de Alzheimer; DLB indica demencia con cuerpos de Lewy; FTD indica demencia frontotemporal; MCI indica Deterioro Cognitivo Leve; y VAD indica demencia vascular (Petersen R. C., 2005).

La prevalencia del MCI en estudios epidemiológicos oscila entre el 3% y el 19% en los adultos mayores de 65 años. Algunas personas que padecen Deterioro Cognitivo Leve pueden mantenerse estables o volver a la normalidad con el paso del tiempo; sin embargo, más de la mitad progresan a la demencia al cabo de 5 años. Por ende, el deterioro cognitivo puede considerarse como un estado de riesgo para la demencia y su identificación temprana podría conducir a la prevención secundaria de dicha enfermedad (Gauthier et al., 2006).

2.3 Proceso de diagnóstico de Deterioro Cognitivo Leve

Según Petersen et al. (2001b) existen cinco criterios fundamentales para diagnosticar Deterioro Cognitivo Leve. Estos son: 1. Queja de pérdida de memoria corroborada por un informante; 2. Deterioro de la memoria en relación con sujetos de la misma edad y nivel educativo; 3. Sin alteraciones en la funcionalidad; 4. Actividades cotidianas no afectadas; 5. Ausencia de demencia. Dichos criterios son incorporados a un diagrama de flujo que representa el proceso de diagnóstico de MCI (ver ilustración 3).

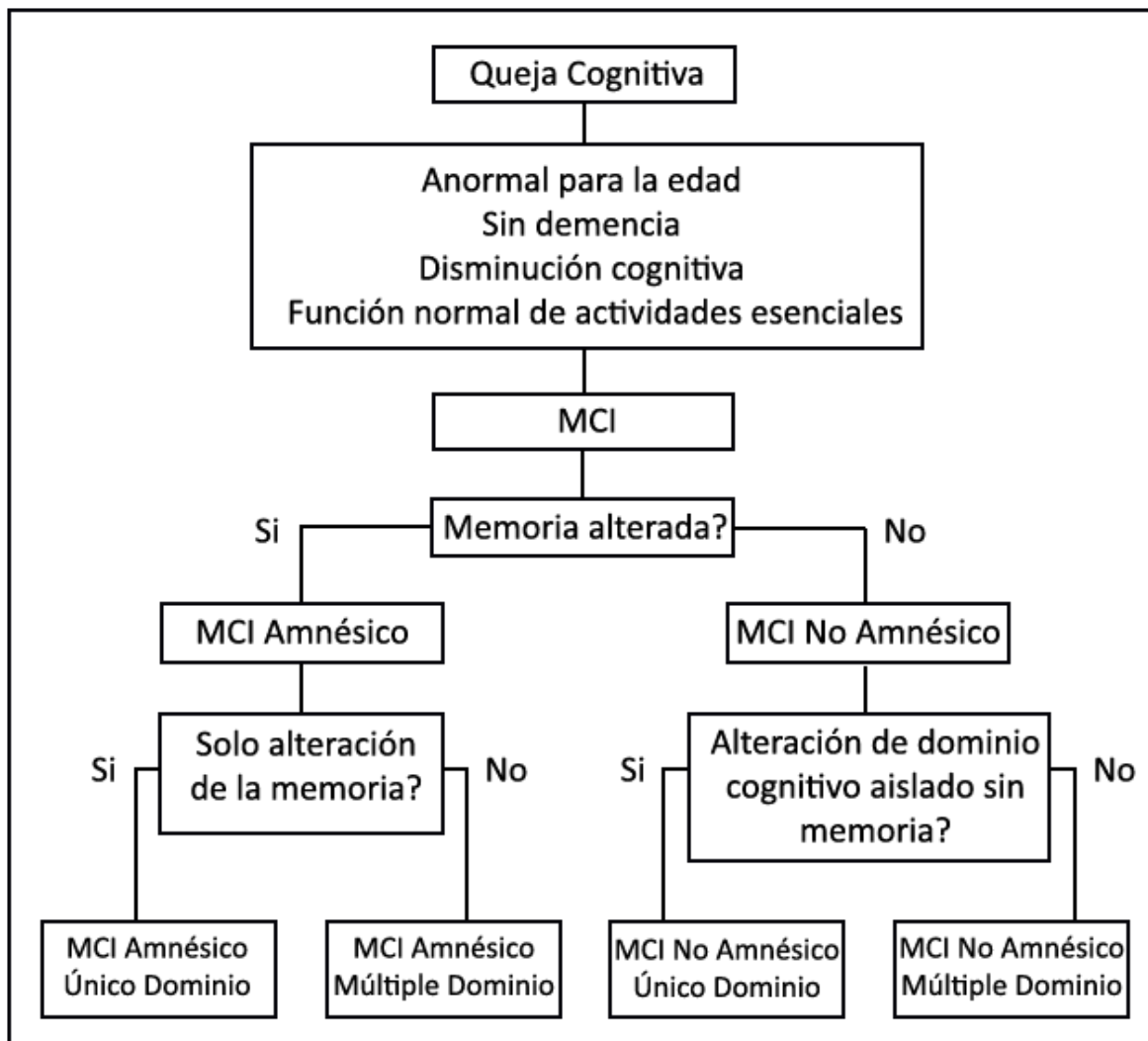


Ilustración 3 Diagrama de flujo para el diagnóstico de los subtipos clínicos de Deterioro Cognitivo Leve. En última instancia, este esquema conducirá a la clasificación dispuesta en la ilustración 2 (Petersen et al., 2005).

De acuerdo con la ilustración 3 cuando una persona expresa cierta preocupación por el estado actual de su función cognitiva, requiere ser valorada por un médico. Basado en la historia clínica y un examen del estatus mental, el doctor emite un juicio en términos de una cognición normal o una sospecha de demencia. No obstante, existen varias instancias en las que el profesional de la salud está incierto acerca del estado cognitivo de la persona; en estos casos es posible considerar el diagnóstico de Deterioro Cognitivo Leve (Petersen R. C., 2004).

Cuando el médico ha determinado que la persona no es normal ni demente, se prosigue a evaluar el deterioro mental. En caso de existir evidencia de deterioro, el doctor debe establecer si los cambios cognoscitivos constituyen un impedimento significativo en las actividades funcionales de tal manera que la persona sea catalogada por tener una demencia muy leve. Sin embargo, si el impedimento funcional no es significativo, el médico debe afianzar el diagnóstico de MCI y el siguiente paso es identificar el subtipo clínico (Petersen R. C., 2004).

Una vez el médico determina que existe un impedimento significativo de la memoria, entonces la condición del paciente se describe como aMCI, por el contrario, si no se detecta un deterioro de la memoria, la condición del paciente se define como naMCI (Petersen R. C., 2004).

El siguiente paso implica determinar si la persona posee un deterioro cognitivo de único dominio o no. En consecuencia, si la persona posee aMCI, el médico requiere evaluar otros dominios cognitivos como el lenguaje, la atención, la función ejecutiva o las habilidades visuales y espaciales para determinar si el impedimento sólo se presenta en la memoria o involucra otros dominios. Si la memoria es el único dominio deteriorado, entonces se cataloga como aMCI de dominio único. Si existen otros dominios deteriorados además de la memoria, la clasificación es aMCI de dominio múltiple. De forma similar, si la memoria no está alterada, la persona posee naMCI y nuevamente se realiza la determinación, ya sea de un único dominio o de múltiples dominios diferentes de la memoria, arrojando como resultado las clasificaciones naMCI de dominio único y naMCI de dominio múltiple respectivamente (Petersen R. C., 2004).

Finalmente, cuando el médico logra determinar el subtipo clínico de Deterioro Cognitivo Leve, se prosigue a puntualizar la etiología del síndrome (ver ilustración 2). Este enfoque es similar al usado en la delineación de la mayoría de diagnósticos neurológicos, incluyendo la demencia (Petersen & Morris, 2005).

2.4 Pruebas de valoración cognitiva

Las pruebas de valoración cognitiva buscan establecer el estado mental de un paciente en diversos dominios cognoscitivos (memoria, lenguaje, atención, percepción, función ejecutiva, entre otros) e identificar la presencia de posibles síndromes clínicos asociados al deterioro cognitivo (Alzheimer's Society, 2015).

Actualmente existe un gran número de pruebas de valoración cognitiva (Alzheimer's Association, 2013; Alzheimer's Society, 2015; National Institutes of Health, s.f.). Sin embargo, a continuación sólo se describen algunas de ellas, las cuales han arrojado resultados sobresalientes como herramientas de toma de decisiones durante el proceso de diagnóstico de MCI (Yin, Zhao, Lu, & Duan, 2015).

2.4.1 *Lawton Instrumental Activities of Daily Living* IADL

Las actividades de la vida diaria se clasifican en dos grupos: básicas (*Basic Activities of Daily Living*, BADL) e instrumentales (*Instrumental Activities of Daily Living*, IADL) (Lawton & Brody, 1969). Las BADL se componen de comportamientos básicos de autocuidado como vestirse, asearse, ir al baño, entre otros. En contraste, las IADL facilitan la vida independiente a través de comportamientos más complejos como transportarse, usar el teléfono, preparar alimentos, administrar medicamentos, manejar dinero, entre otros (Gold, 2012).

Los cuestionarios IADL juegan un papel importante en la evaluación de las capacidades funcionales y el impacto del deterioro cognitivo en las actividades de rutina de los adultos mayores. La independencia en IADL es uno de los rasgos más significativos que distinguen el envejecimiento normal del MCI y la demencia. Como parte de los criterios de diagnóstico de MCI, una persona debe ser independiente en BADL pero puede tener perturbaciones mínimas en el desempeño de IADL (Petersen R. C., 2004; Gold, 2012).

La escala de IADL de Lawton es un instrumento apropiado para evaluar las habilidades de vida independiente (Lawton & Brody, 1969). Esta escala permite medir ocho áreas de

función: capacidad de usar el teléfono, hacer las compras, preparación de la comida, cuidado de la casa, lavado de la ropa, uso de medios de transporte, responsabilidad respecto a su medicación y manejo de asuntos económicos. Las personas son calificadas de acuerdo al nivel más alto de funcionamiento en cada categoría. La puntuación global de la escala oscila entre 0 (dependencia total) y 8 (independencia) (The Hartford Institute for Geriatric Nursing, 2013).

El IADL de Lawton ha sido modificado y adaptado con diferentes enfoques respecto a puntuación. En Colombia la versión más utilizada es la validada por el grupo de Geriátrica de la Universidad de Caldas (Gómez, Curcio, & Gómez, 1995) (ver anexo 3). Con esta adaptación se tiene la ventaja de poder hacer la comparación de momentos antes y después en relación con el inicio del deterioro (Asociación Colombiana de Gerontología y Geriátrica, 2009).

2.4.2 *Mini Mental State Examination* MMSE

El mini examen del estatus mental es una herramienta práctica usada para evaluar el estado mental de manera sistemática y exhaustiva (The Hartford Institute for Geriatric Nursing, 1999). Es «mini» porque solo se concentra en los aspectos cognitivos de las funciones mentales, excluyendo las cuestiones relativas al estado de ánimo, experiencias mentales anormales y la forma de pensar (Folstein, Folstein, & McHugh, 1975).

El MMSE (ver anexo 4) evalúa cinco áreas de la función cognitiva: orientación, registro, atención y cálculo, memoria y lenguaje (The Hartford Institute for Geriatric Nursing, 1999). Dicha prueba está dividida en 2 secciones: la primera requiere solo respuestas vocales y cubre la orientación, la memoria y la atención. La segunda parte pone a prueba la capacidad de nombrar, seguir órdenes verbales y escritas, escribir oraciones de forma espontánea y copiar un polígono complejo. La prueba tiene una calificación máxima de 30 puntos y no posee límite de tiempo (Folstein, Folstein, & McHugh, 1975).

El MMSE es efectivo como instrumento de tamización para separar los pacientes sanos de aquellos que padecen MCI, además es capaz de medir los cambios en el estado cognitivo cuando se utiliza en repetidas ocasiones. Sin embargo, el examen depende en gran medida de la respuesta verbal, lectura y escritura. Por consiguiente, los pacientes que poseen impedimentos auditivos y visuales, bajo nivel de alfabetización, u otros trastornos de la comunicación pueden obtener un desempeño deficiente incluso estando cognitivamente intactos (The Hartford Institute for Geriatric Nursing, 1999).

2.4.3 *Montreal Cognitive Assessment MoCA*

La evaluación cognitiva de Montreal fue desarrollada para asistir a los médicos en la detección del Deterioro Cognitivo Leve. Esta permite evaluar diversos dominios cognitivos: atención y concentración, funciones ejecutivas, memoria, lenguaje, habilidades visuales y espaciales, pensamiento conceptual, cálculo y orientación (Nasreddine et al., 2005).

El MoCA (ver anexo 5) es un examen de rastreo que sirve para detectar disfunciones cognitivas leves. Este puede ser administrado en aproximadamente 10 minutos. La calificación máxima posible es de 30 puntos y una puntuación de 26 o más se considera normal (Nasreddine et al., 2005). Para ajustar el examen a personas con baja formación, se debe añadir un punto a la calificación total para aquellos con 0-12 años de educación. El rango de puntuación para MCI es 19-25,2 y para AD es 11,4-21. Aunque los rangos se superponen, la diferencia entre las condiciones depende del deterioro funcional asociado (The Hartford Institute for Geriatric Nursing, 2012).

El MoCA ha sido extensamente probado para su uso en una variedad de trastornos que afectan la cognición como el HIV, la corea de Huntington, esclerosis múltiple, enfermedad de Parkinson, apoplejía, demencia vascular, y abuso de sustancias. Igualmente ha sido probado en 14 idiomas y diferentes edades que van desde los 49 hasta los 85+ con distintos niveles de educación (The Hartford Institute for Geriatric Nursing, 2012).

2.4.4 *Clinical Dementia Rating CDR*

La Escala Clínica de Demencia (ver anexo 6) es una entrevista estructurada ampliamente utilizada para examinar la progresión de la demencia (Hughes, Berg, & Danziger, 1982). Se basa en las calificaciones del funcionamiento del paciente en seis dominios comúnmente afectados por la enfermedad de Alzheimer: memoria, orientación, juicio y resolución de problemas, asuntos de la comunidad, el hogar y pasatiempos y cuidado personal (Jin, Strauss, & Sherman, 2011).

Basándose únicamente en la información clínica obtenida del paciente y del informante, se genera un cuadro de puntuación entre 0 (sin impedimento) y 3 (impedimento severo) que describe el nivel de deterioro para cada dominio. Posteriormente, se emplea un algoritmo de puntuación para derivar uno de los cinco posibles escenarios: CDR = 0 (sin demencia); CDR = 0,5 (demencia cuestionable); CDR = 1 (demencia leve); CDR = 2 (demencia moderada); CDR = 3 (demencia severa). Aunque no es parte del protocolo original, el CDR = 4 (profunda) y el CDR = 5 (terminal) también se pueden utilizar para catalogar las últimas etapas de la demencia (Jin, Strauss, & Sherman, 2011).

Para ayudar en la calificación, el CDR provee descriptores para los distintos niveles de gravedad en cada cuadro de puntuación. No obstante, surgirán casos en los que un dominio individual cumple con algunos requerimientos para un cuadro de puntuación, y sin embargo, el resto de capacidades de la persona parecen encajar en otro cuadro de puntuación. En esta situación, el médico debe intentar distinguir cuál es la mejor representación de la gravedad de ese dominio en particular. En escenarios en los que el médico no puede decidir entre dos niveles de gravedad, la regla estándar requiere elegir el nivel de gravedad más alto (Alzheimer's Disease Research Center, 2001).

Un método alternativo que genera una puntuación total (rango de 0-18) de la suma de las cajas, también se ha utilizado para propósitos de cuantificación en estudios longitudinales (Cortes et al., 2008). El CDR toma alrededor de 90 minutos para administrar (Jin, Strauss, & Sherman, 2011).

2.4.5 *Global Deterioration Scale GDS*

La Escala de Deterioro Global fue desarrollada por el Dr. Barry Reisberg como instrumento para evaluar la gravedad de la demencia degenerativa primaria (Reisberg, Ferris, de Leon, & Crook, 1982). En el momento de la publicación original no había instrumentos de calificación global, generalmente reconocidos, que pudieran escenificar con claridad la progresión de la demencia (Belden, Burciu, & Sabbagh, 2014).

El GDS (ver anexo 7) provee una visión general de las etapas de la función cognitiva para aquellos que sufren de una demencia degenerativa primaria como la enfermedad de Alzheimer. Se divide en 7 etapas diferentes: 1-3 representan las etapas de pre-demencia; 4-7 corresponden a las etapas de demencia. A partir de la etapa 5, la persona ya no puede sobrevivir sin asistencia (Florida Health Care Association, 2006). La descripción detallada de cada escala es provista por Reisberg et al. (1982). En resumen, las etapas son:

- Etapa 1 – sin deterioro cognitivo: no hay quejas subjetivas o evidencia objetiva de déficit de memoria.
- Etapa 2 – deterioro cognitivo muy leve: quejas subjetivas de falta de memoria sin evidencia objetiva de déficit de memoria.
- Etapa 3 – Deterioro Cognitivo Leve: evidencia objetiva de déficit de memoria con impacto mínimo en el funcionamiento diario.
- Etapa 4 – deterioro cognitivo moderado: déficits evidentes en la memoria y la concentración. Función independiente notablemente afectada (incapacidad para viajar solo en lugares desconocidos o administrar las finanzas personales).
- Etapa 5 – deterioro cognitivo moderado-grave: asistencia regular requerida para las actividades diarias. Exhibe incapacidad para recordad información personal.
- Etapa 6 – deterioro cognitivo grave: déficit de memoria significativo. Asistencia sustancial requerida para las actividades de la vida diaria.
- Etapa 7 – deterioro cognitivo muy grave: se pierden las habilidades verbales. Las habilidades psicomotoras pueden perderse (ej. inhabilidad para caminar) (Reisberg, Ferris, de Leon, & Crook, 1982).

2.5 Diagnóstico asistido por computadora

El diagnóstico asistido por computadora (CAD) es un término atribuido a los sistemas que usan inteligencia artificial para ayudar en la detección y el diagnóstico de una enfermedad (Simmers, Simmers-Nartker, & Simmers-Kobelak, 2014). Con el paso de los años, las ayudas informáticas han evolucionado para asistir con el procesamiento de información y así mejorar aún más el proceso de diagnóstico de diversas patologías (Armitage & Colton, 2005).

El origen de los sistemas CAD se le atribuye con frecuencia a Ledley y Lusted. Su artículo de 1959 describe la lógica simbólica y la teoría de la probabilidad que condujo a diagnósticos similares a los producidos por el razonamiento complejo de los médicos, aunque probablemente sin reproducir dicho razonamiento exactamente (Ledley & Lusted, 1959). Desde entonces, han existido múltiples refinamientos a los alcances, métodos, y capacidades de los sistemas de diagnóstico asistido por computadora (Armitage & Colton, 2005).

2.6 Inteligencia artificial

La inteligencia artificial se define como el estudio de sistemas artificiales que exhiben comportamientos inteligentes (Armitage & Colton, 2005). Los dos principales paradigmas que compiten en AI pueden resumirse de la siguiente forma:

- El paradigma simbólico tradicional (Newell & Simon, 1976). El campo de la AI, desde su origen, ha sido concebido principalmente como el desarrollo de modelos que utilizan la manipulación de símbolos. El cómputo en dichos modelos se basa en representaciones explícitas que contienen símbolos organizados en formas específicas. Asimismo, la información agregada se representa explícitamente con estructuras conjuntas que se construyen a partir de símbolos y combinaciones sintácticas de estos mismos (Sun R., 2000).

- El paradigma conexionista de más reciente creación (Rumelheart & McClelland, 1986; Smolensky P., 1988). La aparición del enfoque conexionista (también llamado subsimbólico) se dio como resultado de varias insatisfacciones con los modelos de manipulación de símbolos, especialmente por su incapacidad para manejar el procesamiento flexible y robusto de una manera eficiente. El paradigma conexionista apunta a modelos masivamente paralelos que constan de un gran número de simples y uniformes elementos de procesamiento interconectados con enlaces extensivos (Sun R., 2000).

Cabe destacar que hasta el momento, ningún paradigma (simbólico o conexionista) puede abordar plenamente todos los grandes problemas de la AI. Cada paradigma tiene sus fortalezas y debilidades, y se destaca en cierta tarea mientras que se queda corto en otras. Esta situación, en cierto modo, indica la necesidad de integrar ambos paradigmas existentes de alguna manera (Sun R., 2000).

La investigación en inteligencia artificial es fundamentalmente interdisciplinaria. Esta se superpone con la informática, la psicología, la estadística, las matemáticas, la ingeniería, la biología, la lingüística, entre otras disciplinas (Armitage & Colton, 2005). El desarrollo de sistemas inteligentes ha sido uno de los temas más ambiciosos al interior del campo de la medicina. Desde plataformas capaces de «minar» conocimiento de las historias clínicas (Ollier, 2015), pasando por sistemas que permiten detectar anomalías en imágenes médicas (Forrest, 2015), hasta avatares digitales aptos para responder preguntas específicas de los pacientes (CB Insights, 2016). Estos son algunos ejemplos que muestran el potencial que posee la AI en la transformación de la industria sanitaria.

En el ámbito clínico, Los sistemas inteligentes están destinados a asistir a los profesionales de la salud en el curso normal de sus funciones, ayudando con las tareas basadas en manipulación de datos (ej. CAD). En particular, algunos de estos sistemas tienen la capacidad de aprender, lo que conlleva al descubrimiento de nuevos fenómenos y la creación de conocimiento médico (EPIA, 2015).

2.6.1 Inteligencia artificial simbólica

La PSSH (*Physical Symbol System Hypothesis*) introducida por Newell y Simon (1976) articula claramente los principios de la AI simbólica sosteniendo que «un sistema de símbolos físicos posee los medios necesarios y suficientes para realizar una acción inteligente genérica». Ellos definieron un sistema de símbolos físicos de la siguiente forma:

Un sistema de símbolos físicos consta de un conjunto de entidades, llamadas símbolos, los cuales son patrones físicos que pueden funcionar como componentes de otro tipo de entidad llamada expresión (estructura de símbolos). Por lo tanto, una estructura de símbolos está compuesta por un número de instancias de símbolos (o *tokens*) relacionadas de alguna forma física (ej. un *token* que debe seguir a otro) (Newell & Simon, 1976).

Newell y Simon (1976) señalan que un símbolo puede ser usado para designar absolutamente cualquier expresión. Esto es, dado un símbolo, no se prescribe a priori qué expresiones puede designar. Esta arbitrariedad pertenece solo a los símbolos: los ejemplares de símbolos y sus mutuas relaciones determinan qué objetos son designados por una expresión compleja. Además, existen procesos para crear cualquier expresión y modificar cualquier expresión de maneras arbitrarias.

La estructura de un sistema simbólico se divide en dos componentes: un conjunto de conocimientos representados mediante símbolos (ej. palabras, números, frases o imágenes) y un algoritmo de razonamiento encargado de manipular dicho conjunto. Lo anterior permite crear nuevas combinaciones de símbolos que posiblemente representan soluciones a problemas o nuevos conocimientos (Millington & Funge, 2009).

Frecuentemente, los algoritmos de razonamiento se basan en búsquedas —probar diferentes alternativas para obtener el mejor resultado—. Esto conlleva a la regla de oro de la AI: la búsqueda y el conocimiento están intrínsecamente ligados. Cuanto más conocimiento se tiene, menos búsqueda se requiere para encontrar la solución; cuanta

más búsqueda se pueda realizar (es decir, más rápido se pueda buscar), se necesita menos conocimiento (Millington & Funge, 2009).

La PSSE ha dado lugar a un enorme esfuerzo de investigación en AI (y en ciencia cognitiva). El enfoque tradicional suele utilizar símbolos discretos como primitivas, y lleva a cabo la manipulación de símbolos de una manera secuencial y discreta (Sun R., 2000). Un sistema experto, una de las expresiones más puras de dicho enfoque, es el instrumento más famoso de AI simbólica. Cuenta con una gran base de datos de conocimiento y aplica reglas para descubrir nuevos elementos. No obstante, existen muchos otros instrumentos de AI simbólica incluyendo arquitecturas en pizarra, búsqueda de caminos, árboles de decisión, y máquinas de estado (Millington & Funge, 2009).

2.6.2 Inteligencia artificial subsimbólica

En la década de 1980, la publicación del libro *Parallel Distributed Processing* (PDP) comenzó la denominada «revolución conexionista» en la AI y la ciencia cognitiva (Rumelheart & McClelland, 1986). La idea básica de usar una amplia red de unidades simples para hacer frente a la computación compleja, parecía completamente antitética a los principios de la AI simbólica (Sun R., 2000).

A través de la década de 1980 y en la década de 1990, hubo un aumento de la frustración con los enfoques simbólicos que provino desde varias direcciones. Desde el punto de vista de la ingeniería, el éxito temprano en problemas simples no parecía escalar a problemas más difíciles o manejar la incertidumbre y complejidad del mundo real. Aparentemente, era simple desarrollar modelos que entendieran oraciones simples, pero el desarrollo de la comprensión del lenguaje humano completo no parecía estar cerca. Lo anterior impulsó el desarrollo de la AI conexionista o subsimbólica (Millington & Funge, 2009).

Se creía que los modelos subsimbólicos eran un primer paso en la dirección hacia la captura de las propiedades intrínsecas del sustrato biológico de la inteligencia, en tanto habían sido inspirados por las redes neuronales biológicas y parecían estar más cercanos a

los procesos biológicos. También, que eran capaces de hacer frente a la generalización, así como a la información incompleta, aproximada e inconsistente (Sun R., 2000). Dichos modelos eran basados en redes neuronales, algoritmos genéticos y reconocimiento simulado (Millington & Funge, 2009).

Poco a poco, los principales investigadores en AI se dieron cuenta de que el ingrediente clave de este nuevo enfoque no era la conexión con el mundo natural, sino la capacidad de manejar la incertidumbre para resolver problemas del mundo real. Comprendieron que varios instrumentos tales como las redes neuronales podían explicarse matemáticamente en términos de un marco de trabajo probabilístico y estadístico riguroso. Libre de la necesidad de realizar cualquier interpretación natural, el marco de trabajo pudo extenderse para fundar el núcleo de la AI moderna que incluye instrumentos como las redes bayesianas (BN), los procesos de Gauss, las máquinas de soporte vectorial (SVM), entre otras (Millington & Funge, 2009).

2.6.3 Sistemas híbridos

La aplicación de un solo instrumento de AI (con sus fortalezas, capacidades y supuestos) para resolver un problema del mundo real puede dar resultados insatisfactorios. La integración de dos o más instrumentos es necesaria en algunos casos. Una combinación de diferentes instrumentos de AI contribuye al refuerzo mutuo de las ventajas y reducción de las debilidades que exhibe cada uno por separado (Rebizant, Szafran, & Wiszniewski, 2011).

Teniendo en cuenta los diferentes énfasis y fortalezas de los sistemas simbólicos y conexionistas, parece plausible que la combinación de ellos sería una vía prometedora para el desarrollo de sistemas más robustos, más potentes y más versátiles. La necesidad de este tipo de sistemas viene creciendo lentamente pero de manera constante. Lo anterior ha desencadenado un gran esfuerzo de investigación, que conduce a los actualmente llamados sistemas híbridos (Sun R., 2000).

Los sistemas híbridos tienen por objeto proporcionar la capacidad de procesamiento de información para la representación y evaluación de diversas situaciones ambiguas e inciertas de la vida real. Dichos sistemas cumplen las siguientes características:

- Combinan al menos dos instrumentos de AI (comúnmente uno simbólico y uno conexionista).
- Son altamente capaces de razonar y aprender en un entorno incierto e impreciso.
- Permiten la obtención de una mayor flexibilidad y robustez, y reducen el costo de las soluciones (Rebizant, Szafran, & Wiszniewski, 2011).

Cualquier combinación de instrumentos de AI es concebible; los ejemplos pueden incluir híbridos neuro-expertos, sistemas expertos difusos, redes neuronales difusas, algoritmos genéticos difusos, entre otros. El estudio de la literatura confirma que la hibridación neuro-difusa es la integración más viable realizada hasta el momento (Rebizant, Szafran, & Wiszniewski, 2011).

En general, los esquemas híbridos son fusionados (fundidos) o cooperativos. La primera categoría incluye aquellos en los que uno de los esquemas se «funde» en el otro para realizar cierta función parcial, o cuándo un instrumento de AI se representa por medio de otro para explotar las virtudes del instrumento (ej. entrenamiento de un esquema difuso). La segunda categoría comprende los esquemas híbridos en los que cada uno de los instrumentos de AI trabaja de forma independiente de los demás. Sus entradas y salidas pueden intercambiarse, lo cual resulta comúnmente en conexiones en cascada entre los instrumentos (ej. los sistemas neuro-expertos) (Rebizant, Szafran, & Wiszniewski, 2011).

Un sistema híbrido puede ser bueno o malo —depende de los componentes que lo conforman—. Metafóricamente, es posible decir que un buen híbrido sería «la policía británica, los mecánicos alemanes, la cocina francesa y la banca suiza», pero «la policía suiza, los mecánicos franceses, la cocina alemana y la banca británica» sería uno malo. Por ende, se requiere de un análisis cuidadoso para seleccionar los componentes correctos de forma que puedan producir un buen sistema híbrido (Rebizant, Szafran, & Wiszniewski, 2011).

2.7 Aprendizaje de máquina

Es un tipo de inteligencia artificial que provee a las computadoras la habilidad de aprender sin ser explícitamente programadas (Rouse, 2016). Se dice que el aprendizaje de máquina ocurre en un programa que puede modificar algún aspecto de sí mismo, a menudo referido como su estado, de modo que las ejecuciones posteriores con el mismo conjunto de entradas produzcan una diferente (ojalá mejor) salida (Wilson, 2012).

El proceso de aprendizaje de la máquina es similar al llevado a cabo en minería de datos. Ambos sistemas buscan a través de los datos para encontrar patrones. No obstante, en lugar de extraer datos para la comprensión humana —como es el caso en las aplicaciones de minería de datos— el aprendizaje de máquina emplea esos datos para ajustar las acciones del programa como corresponde (Rouse, 2016).

Las técnicas de aprendizaje de máquina pueden ser simbólicas o subsimbólicas y su aprendizaje es comúnmente catalogado como supervisado o no supervisado (Rouse, 2016). En el presente estudio se empleará únicamente el aprendizaje supervisado.

2.7.1 Aprendizaje supervisado

Representa la tarea de aprendizaje de máquina de inferir una función a partir de datos de entrenamiento. Los datos de entrenamiento comprenden un conjunto de ejemplos u observaciones del mundo real. En el aprendizaje supervisado, cada ejemplo es un par que consta de un objeto de entrada (típicamente un vector) y un valor de salida deseado (también llamado señal de supervisión) (Russel, 2013).

Un algoritmo de aprendizaje supervisado analiza los datos de entrenamiento y produce una función inferida, que se denomina un clasificador (si la salida es discreta) o una función de regresión (si la salida es continua). La función inferida debe poder predecir el valor de salida correcto para cualquier objeto de entrada válido. Esto requiere que el algoritmo de aprendizaje generalice a partir de los datos de entrenamiento a situaciones desconocidas —de una manera «razonable»— (Russel, 2013).

3 ESTRATEGIA METODOLÓGICA

3.1 Población

La población objetivo se enfoca en las personas mayores de 60 años pertenecientes al régimen subsidiado de salud, que residen en los siguientes municipios del departamento de Caldas: Aguadas, Anserma, Belalcázar, Chinchiná, La Dorada, Manizales, Neira, Pensilvania, Riosucio, Salamina, Samaná, Supía, Villamaría y Viterbo. Los municipios escogidos son equivalentes a los beneficiados por el actual proyecto de regalías; esto favoreció el desarrollo de la presente investigación. A continuación se ilustra el área geográfica correspondiente a la ubicación de la población objetivo:

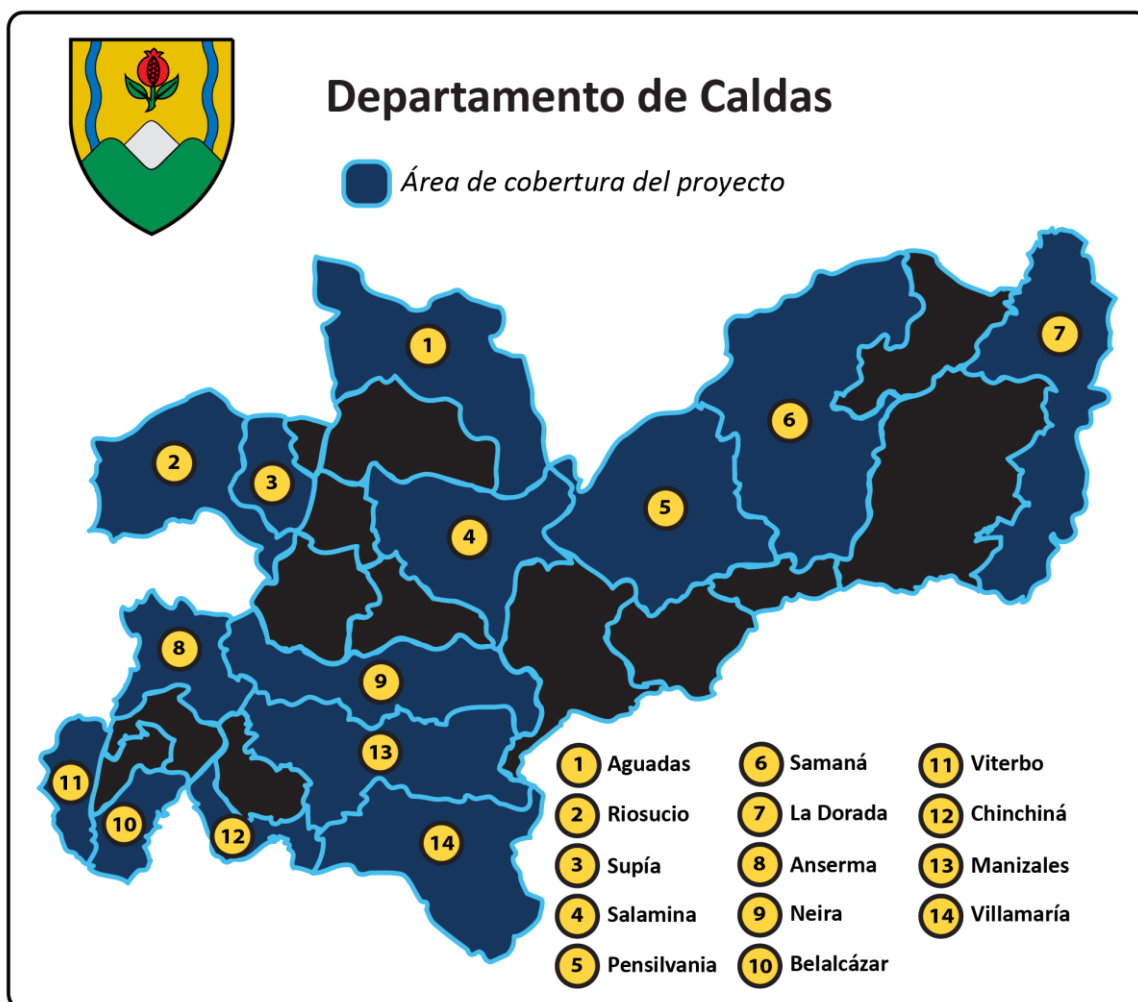


Ilustración 4 Área de cobertura del proyecto que involucra a 14 municipios del departamento de Caldas.

3.2 Obtención de la información

En el marco de la iniciativa «**Implementación del programa para diagnóstico y control de enfermedades crónicas no transmisibles y Cáncer de Cérvix y Mama, con el apoyo de TIC en el departamento de Caldas**» la Universidad Autónoma de Manizales desea evaluar la prevalencia de deterioro cognitivo en la población mayor de 60 años de 14 municipios del departamento de Caldas. Para lograr este propósito, se ha construido un sistema de información denominado «**Repositorio Clínico UAM**» el cual permite almacenar de forma digital la historia clínica de los pacientes beneficiados por dicho proyecto de regalías.

La Historia Clínica es un documento privado, obligatorio y sometido a reserva, en el cual se registran cronológicamente las condiciones de salud del paciente, los actos médicos y los demás procedimientos ejecutados por el equipo de salud que interviene en su atención (Ministerio de Salud de Colombia, 1999).

Dicho documento posee gran importancia para los profesionales de la salud debido a que permite: 1. Documentar el estado de salud o enfermedad del paciente; 2. Establecer el diagnóstico exacto del proceso patológico que está produciendo manifestaciones clínicas en el paciente; 3. Establecer el pronóstico de la enfermedad por la que cursa el paciente y definir un plan de acción concerniente a prevenir que el proceso patológico continúe avanzando; 4. Diseñar un programa que permita la rehabilitación del paciente; 5. Educar al paciente sobre la entidad que lo afecta y la manera como él puede disminuir los efectos de la enfermedad sobre sí mismo, la familia y el ambiente de amigos o compañeros de trabajo más cercano (Ramírez, 1995).

Como se indica previamente, el sistema de información construido por la UAM permite diligenciar y almacenar digitalmente la historia clínica de los pacientes adscritos al actual proyecto de regalías. Esta comprende los siguientes elementos:

- Datos básicos: corresponden con el primer paso en la relación médico-paciente y permiten conocer de forma general quién es la persona y cuál es su procedencia (Rodríguez Silva, 2010). Estos datos incluyen el documento de identidad, el nombre completo, el género, el grupo sanguíneo, la fecha de nacimiento, la edad y el lugar de

nacimiento. Igualmente, contienen información relevante para el estudio de la prevalencia de deterioro cognitivo como la lateralidad, la autopercepción de la salud y el dominio de una segunda lengua.

- Datos de contacto: contienen la información de localización de las personas allegadas al paciente, permitiendo que el profesional de la salud pueda acudir a ellas en caso de ocurrir algún acontecimiento. Estos datos incluyen el nombre completo del contacto sumado a su ubicación, dirección, teléfono y celular.
- Información socio-económica: permite conocer el entorno físico y emocional en el que se desenvuelve el paciente. Dicha información incluye la situación laboral y familiar de la persona, así como su nivel de escolaridad, estrato socio-económico, tipo de vivienda (propia, rentada o familiar) y régimen de salud (contributivo o subsidiado).
- Antecedentes: proporcionan una visión clara del estado de salud general del paciente antes de la enfermedad actual y de la salud de sus familiares; su estilo de vida, ambiente familiar y social, ocupación, pasatiempos, hábitos y costumbres alimenticias, higiénicas y sexuales potencialmente riesgosas para la salud (Ramírez, 1995). El objetivo principal de conocer los antecedentes radica en la necesidad de identificar factores de riesgo de la persona, es decir situaciones o aspectos del estilo de vida que pueden iniciar, promover o fortalecer la posibilidad de desarrollar cierta patología.
- Registros clínicos: comprenden la información de las consultas médicas realizadas por los profesionales de la salud. Un registro de consulta incluye el lugar y la fecha donde se realiza la atención médica. Igualmente, contiene los resultados obtenidos por el paciente en diferentes pruebas de tamizaje cognitivo (IADL, MMSE, MoCA, CDR y GDS), las cuales permiten determinar su estado de salud mental.

Actualmente, el repositorio se encuentra en <http://teleasistencia.autonoma.edu.co>. Los datos registrados en este pueden ser exportados a libros de Excel por medio de diversos reportes configurados en la aplicación. Dichos datos también pueden ser consultados vía Internet a través de los servicios web implementados en el software. Por consiguiente, la información requerida para realizar el presente estudio se obtiene a través de los métodos antes descritos.

Adicional a la información recopilada en los 14 municipios del departamento de Caldas, en el presente estudio se emplea igualmente la base de datos de la *ADNI (Alzheimer's Disease Neuroimaging Initiative)*, obtenida en junio del 2016 de <http://adni.loni.usc.edu>. Dicha iniciativa fue fundada en el año 2003 por el *National Institute on Aging* de Estados Unidos de América, 13 compañías farmacéuticas y 2 fundaciones dedicadas a la investigación de la enfermedad de Alzheimer (Weiner et al., 2012).

La *ADNI* ofrece una amplia información sobre exámenes clínicos, biomarcadores y pruebas neuropsicológicas de los participantes del proyecto. Actualmente, la base de datos cuenta con información de 877 pacientes provenientes de Estados Unidos de América y Canadá, catalogados en tres grupos diferentes: 426 sujetos de control (sanos), 254 sujetos con MCI y 197 sujetos con AD. Los procedimientos de selección de los participantes y los protocolos utilizados en la *ADNI*, fueron descritos por Jack et al. (2008); Mueller et al. (2005) y se pueden encontrar en línea en <http://adni.loni.usc.edu/methods>. Asimismo, una completa revisión de los principales resultados obtenidos hasta el momento en la *ADNI* fue presentada por los investigadores del proyecto en el año 2012 (Gavidia, Soudah, & Oñate, 2012). Recientemente, dicha revisión fue actualizada para incluir la descripción de la segunda fase del proyecto denominada *ADNI-2* (Weiner et al., 2015). Para mayor información visitar: <http://www.adni-info.org/>.

3.3 Descripción de las variables

Investigaciones dedicadas al progreso de enfermedades del tipo MCI, AD y otros tipos de demencias, destacan un especial interés por analizar la influencia de las características socio-demográficas en el deterioro cognitivo (Gavidia, Soudah, & Oñate, 2012). Los estudios realizados por Stern et al. (1994); Anderson et al. (2007); Wu et al. (2011) han demostrado que variables como la edad, el nivel socio-económico, la ocupación, el nivel educativo, el género, el dominio de una segunda lengua, el estado civil, entre otras, repercuten en el desempeño de las pruebas de valoración cognitiva.

En el presente estudio se incluyen inicialmente las variables socio-demográficas de edad, sexo, lateralidad, estado civil y nivel educativo. Dichas variables también han sido consideradas en diferentes investigaciones realizadas sobre la base de datos de la *ADNI* y otras bases de datos relativas a AD (Gavidia, Soudah, & Oñate, 2012).

Respecto a las variables provenientes de las pruebas de valoración cognitiva se incluyen inicialmente los resultados de los exámenes MMSE, CDR, GDS, IADL y MoCA, sumado al diagnóstico otorgado por los especialistas en las consultas preliminares. Las tres últimas pruebas cognitivas (GDS, IADL y MoCA) no están totalmente documentadas en los registros de la *ADNI* por lo cual no fueron consideradas durante el uso de dicha base de datos. En la tabla 2 se muestran con más detalle las variables que se incluyeron en el desarrollo del actual proyecto.

Nombre	Tipo	Valores	<i>ADNI</i>	<i>Caldas</i>
N° sujetos	Numérica	370 - 877	877	370
N° hombres	Numérica	113 - 480	480	113
N° mujeres	Numérica	257 - 397	397	257
Edad (Años)	Numérica	54 - 99	75,13 ± 13,54	72,61 ± 16,58
Sexo	Nominal	Femenino	397	257
		Masculino	480	113
Lateralidad	Nominal	Diestro(a)	811	362
		Zurdo(a)	66	7
		Ambidiestro(a)	0	1
Estado Civil	Nominal	Soltero(a)	38	76
		Viudo(a)	115	109
		Casado(a)	655	136
		Separado(a)	68	36
		Unión Libre	0	13
		Desconocido	1	0
Educación (Años)	Numérica	0-20	15,81 ± 5,94	-

Educación (Categoría)	Nominal	Analfabeta	-	53
		Lee y Escribe	-	69
		Primaria	-	230
		Secundaria	-	17
		Profesional	-	0
		Especialización	-	1
IADL (Previo Sin Dificultad)	Numérica	0 - 14	-	13,48 ± 2,28
IADL (Previo Con Dificultad)	Numérica	0 - 14	-	0,05 ± 0,6
IADL (Previo Con Ayuda)	Numérica	0 - 14	-	0,02 ± 0,32
IADL (Previo No Realiza)	Numérica	0 - 14	-	0,45 ± 2,08
IADL (Actual Sin Dificultad)	Numérica	0 - 14	-	12,00 ± 6,02
IADL (Actual Con Dificultad)	Numérica	0 - 14	-	0,79 ± 3,04
IADL (Actual Con Ayuda)	Numérica	0 - 14	-	0,48 ± 2,34
IADL (Actual No Realiza)	Numérica	0 - 14	-	0,72 ± 2,74
MMSE	Numérica	0 - 30	27,29 ± 5,02	24,97 ± 9,06
MoCA	Numérica	0 - 30	-	18,55 ± 13,18
CDR	Numérica	0 - 3	0,31 ± 0,66	0,30 ± 0,94
GDS	Numérica	1 - 7	-	2,31 ± 1,92
Diagnóstico	Nominal	Sano	426	226
		Leve	254	115
		Moderado o Severo	197	29

Tabla 2 Descripción de las variables incluidas en el desarrollo del proyecto. La variable dependiente denominada «Diagnóstico» representa el dictamen brindado por los especialistas respecto a la condición mental de las personas. Dicha variable posee 3 clases: persona sana, persona con Deterioro Cognitivo Leve y persona con deterioro cognitivo moderado o severo. Las demás variables son independientes y provienen de la información socio-demográfica diligenciada en la primera consulta médica y de los resultados de las pruebas de valoración cognitiva aplicadas por los profesionales de la salud.

3.4 Procedimientos (análisis del desempeño de las técnicas de ML)

Con el propósito de determinar cuáles técnicas de aprendizaje de máquina poseen un mejor desempeño en la asistencia al diagnóstico de Deterioro Cognitivo Leve, se decide establecer un estudio comparativo que involucra las métricas de exactitud, precisión, sensibilidad, especificidad y área bajo la curva (AUC, *Area Under Curve*) «característica operativa del receptor» (ROC, *Receiver Operating Characteristic*). Dichas métricas han sido usadas comúnmente para valorar el performance de las técnicas de ML en métodos comparativos y construcción de modelos (Saito & Rehmsmeier, 2015).

La curva ROC es una representación gráfica que ilustra el desempeño de un sistema clasificador binario según se varía el umbral de discriminación (Onieva, Santos, Osaba, Quintián, & Corchado, 2015). En una clasificación binaria, los datos se dividen en dos clases diferentes: positivos (P) y negativos (N). Posteriormente, el clasificador binario cataloga todas las instancias de datos como positivas o negativas lo cual produce cuatro tipos de resultados: verdaderos positivos (TP, *True Positives*), verdaderos negativos (TN, *True Negatives*), falsos positivos (FP, *False Positives*) y falsos negativos (FN, *False Negatives*) (Saito & Rehmsmeier, 2015). Una tabla de 2x2 que contiene estos tipos de resultados se denomina una «matriz de confusión» (ver ilustración 5).

Matriz de Confusión (*Confusion Matrix*)

		Predicción	
		C _p	C _N
Clase real	C _p	TP (P_{11})	FN (P_{12})
	C _N	FP (P_{21})	TN (P_{22})

***TP: True Positives**

***TN: True Negatives**

***FP: False Positives**

***FN: False Negatives**

Ilustración 5 Matriz de confusión para un sistema clasificador binario con 2 clases (positivo y negativo) y 4 tipos de resultados (verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos) (Berzal, 2016).

Adicional a los cuatro tipos de resultados (TP, TN, FP y FN), la matriz de confusión permite visualizar diferentes métricas de desempeño como la tasa de verdaderos positivos (TPR, *True Positive Rate*), la tasa de falsos positivos (FPR, *False Positive Rate*), la exactitud, la precisión, la especificidad, entre otras (ver tabla 3) (Davis & Goadrich, 2006).

Métrica	Fórmula
Exactitud	$Exactitud = \frac{TP + TN}{TP + TN + FN + FP}$
Tasa de Error	$Tasa\ de\ Error = \frac{FP + FN}{TP + TN + FN + FP}$
TPR (Sensibilidad)	$TPR = \frac{TP}{TP + FN}$
Especificidad	$Especificidad = \frac{TN}{TN + FP}$
FPR	$FPR = \frac{FP}{TN + FP}$
Precisión	$Precisión = \frac{TP}{TP + FP}$
F_1 -Score	$f_1 = \frac{2 * Precisión * Sensibilidad}{Precisión + Sensibilidad}$
Youden's J	$J = Sensibilidad + Especificidad - 1$
Cohen's Kappa	$Kappa = \frac{p_o - p_e}{1 - p_e} \quad p_o = \sum_{i=1}^k p_{ii} \quad p_e = \sum_{i=1}^k p_{i.} * p_{.i}$

Tabla 3 Fórmulas para calcular las métricas de evaluación del desempeño de un clasificador, obtenidas a partir de la matriz de confusión (Powers, 2007; Saito & Rehmsmeier, 2015).

En definitiva, una curva ROC es un gráfico bidimensional en el cual la TPR se representa en el eje Y y la FPR se representa en el eje X (ver ilustración 6). Su área bajo la curva indica una sola medida de desempeño que varía entre 0,5 y 1. Mientras más cerca se encuentre del valor 1, la precisión del sistema o modelo será mayor (Fawcett, 2006).

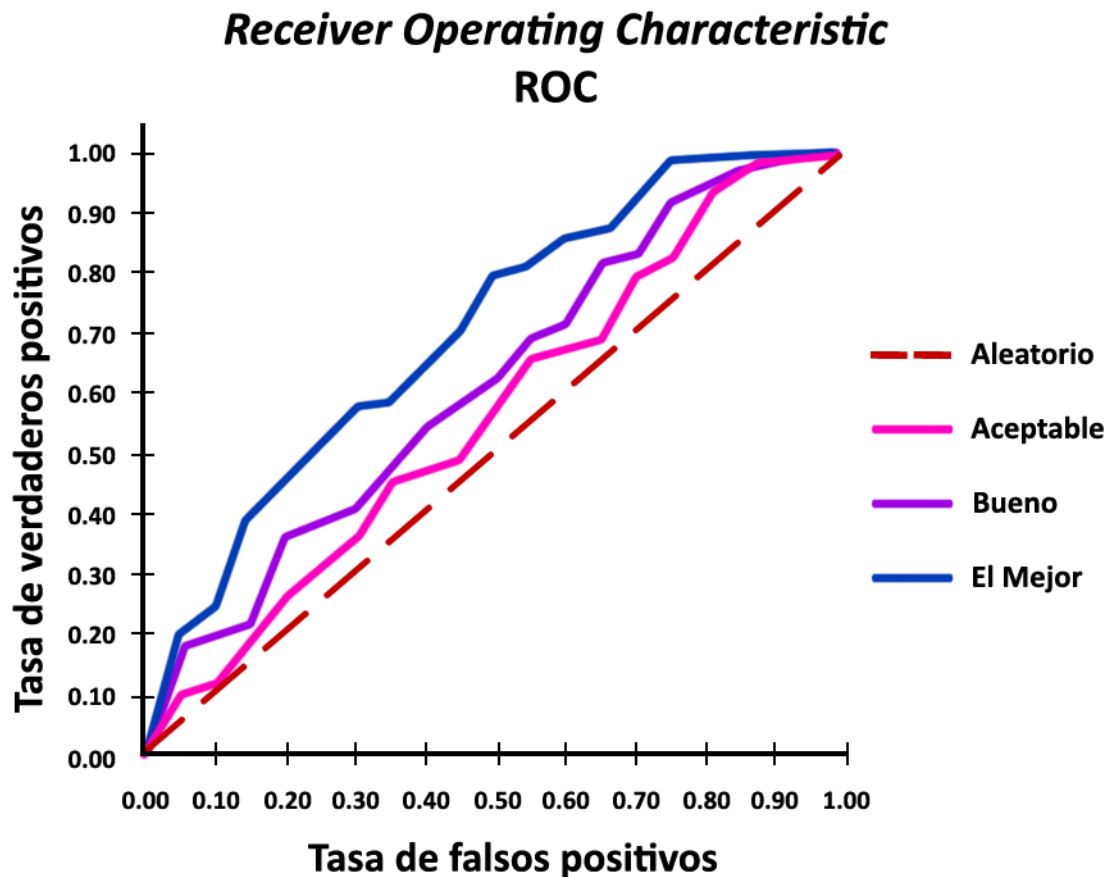


Ilustración 6 Visualización de cuatro curvas ROC con diferentes valores de AUC (OpenEye Scientific, 2015). El clasificador binario representado por la curva azul posee un mejor desempeño que los demás, debido a que su área bajo la curva ROC se encuentra más cercana a 1.

Las puntuaciones AUC-ROC son convenientes para comparar los resultados de diferentes sistemas o modelos (Saito & Rehmsmeier, 2015). Además, se han elaborado extensiones de ambos instrumentos (curva ROC y matriz de confusión) para abordar situaciones con un mayor número de clases (Hand & Till, 2001; Beleites, Salzer, & Sergio, 2013).

3.5 Procedimientos (metodología de desarrollo de software)

En el presente estudio se ha optado por trabajar con una metodología de desarrollo de software orientada a prototipos. Estas han sido discutidas en la literatura como un

acercamiento importante para la validación temprana de requisitos. Un prototipo es una maqueta de un sistema que permite la evaluación de características a través de la interacción de usuarios y desarrolladores con escenarios operacionales. La orientación a prototipos expone varios aspectos funcionales del sistema, lo cual aumenta la precisión de los requerimientos y ayuda a controlar su volatilidad durante el desarrollo (Wood & Kang, 1992).

Al interior del ámbito de los prototipos existen dos enfoques diferentes: aquellos que implican la creación de una serie de ejemplares desplegados, y los que intentan explorar ideas sin recurrir al entorno de despliegue. Los primeros son usualmente conocidos como prototipos de campo o evolutivos, mientras que los últimos admiten diversos nombres, entre ellos prototipos rápidos, experimentales, desechables, exploratorios, entre otros (Budde, Kuhlenkamp, Mathiassen, & Züllighoven, 1984).

Si bien algunos detalles difieren entre cada perspectiva, existe un proceso común que abarca ambos enfoques (ver ilustración 7). De este modo, se denota una secuencia de pasos por los cuales un conjunto de metas del sistema se transforman en un modelo realizable: 1. Los objetivos del sistema, en forma de requerimientos, deben ser extraídos de fuentes primarias (ej. clientes, usuarios, documentación). 2. La información obtenida debe ser analizada frente a distintas propiedades como coherencia e integridad, y posteriormente debe ser codificada en alguna forma de especificación (ej. textual, gráfica, matemática). 3. Finalmente, esta especificación puede constituir la base para la etapa de validación mediante la creación de prototipos (Wood & Kang, 1992).

El desarrollo orientado a prototipos representa un enfoque iterativo, en el que cada entrega mejora desde el punto de vista del usuario (Parbst, 1984). La secuencia continua de pasos aplica igualmente bien a cualquier tipo de necesidades de sistema, incluyendo requerimientos de usuario, funcionales, de comportamiento y de interfaz de usuario, así como restricciones de diseño o aspectos de arquitectura (Wood & Kang, 1992).

Metodología de desarrollo orientada a prototipos

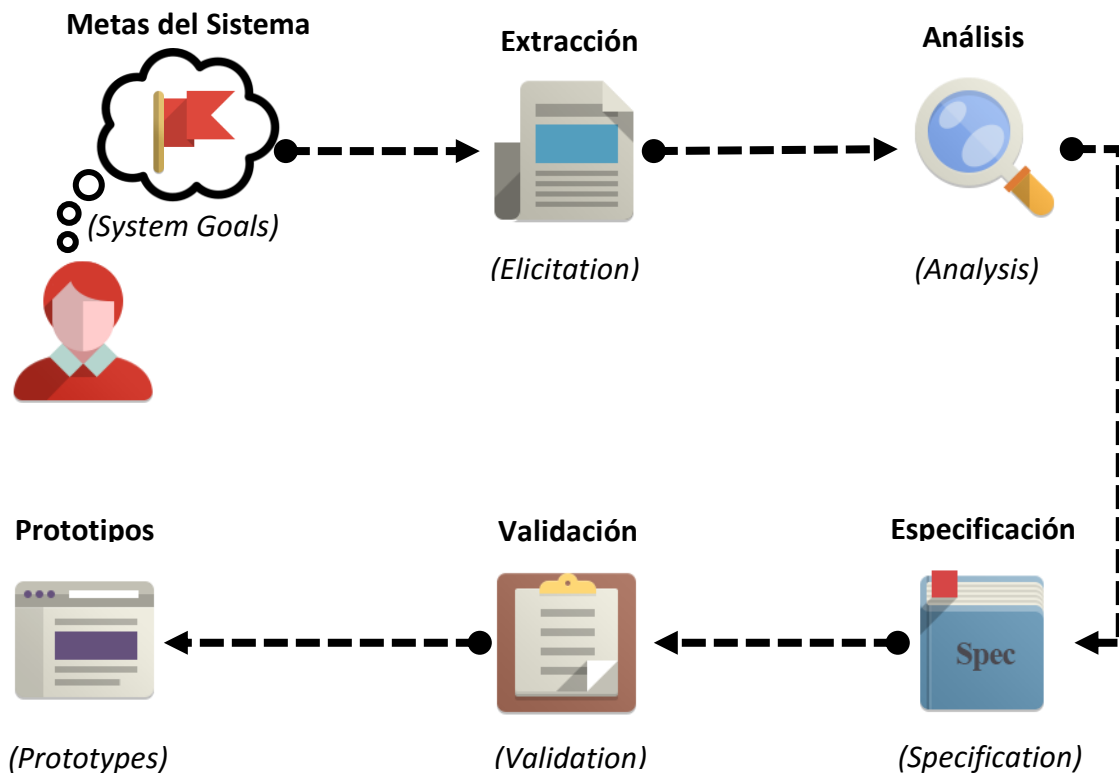


Ilustración 7 Proceso de una metodología de desarrollo orientada a prototipos que consta de seis etapas: definición de las metas del sistema, extracción de los requerimientos del sistema, análisis de los requerimientos del sistema, especificación gráfica o textual del sistema, validación del sistema y construcción de los prototipos del sistema (Wood & Kang, 1992).

Para la construcción de los subsistemas computacionales se decide usar específicamente la orientación a prototipos evolutivos. Esta es una técnica de desarrollo de software que implica la elaboración inicial de un ejemplar. Posteriormente, se reciben sugerencias por parte del cliente y se producen nuevos prototipos, cada uno con funcionalidad o mejoras adicionales, hasta que el producto final emerge. Es importante resaltar que la orientación a prototipos evolutivos es similar al desarrollo incremental en cuanto a que las partes del sistema pueden ser inspeccionadas o entregadas al cliente a lo largo del ciclo de vida del software (Sherrell, 2013).

3.6 Procedimientos (verificación del proceso de desarrollo de software)

La alternativa más adecuada para verificar la construcción de un sistema computacional durante un desarrollo de software orientado a prototipos evolutivos implica elaborar un plan de pruebas en paralelo con las iteraciones propias del método (Shimeall, 1990). Inicialmente, sólo se deben considerar los casos de prueba para la especificación básica del sistema contenida en la primera iteración por prototipos. A medida que las iteraciones avanzan, el plan de pruebas se amplía para cubrir la funcionalidad añadida o mejorada.

Es importante resaltar que una iteración puede generar cambios o adiciones a las especificaciones actuales. En consecuencia, siempre que un cambio altere un elemento probado con anterioridad, se debe repetir la totalidad de las pruebas para determinar si las modificaciones introdujeron nuevos errores (Shimeall, 1990).

Lo anterior indica que los casos de prueba y los resultados de las ejecución de los mismos requieren ser documentados y supervisados a lo largo de todo el proceso de desarrollo de software orientado a prototipos evolutivos. Por consiguiente, se decide hacer uso del formato básico (ver anexo 2) de planes de prueba propuesto por la corporación IBM para registrar la totalidad de los aspectos relacionados con el proceso de verificación.

3.7 Procedimientos (esquema de *cross-validation* o validación cruzada)

Un esquema de *cross-validation* o validación cruzada es un método estadístico para evaluar el desempeño de un sistema o modelo mediante la división del conjunto de datos en dos segmentos: uno para entrenar y otro para validar (ver ilustración 8). En la validación cruzada tradicional, los conjuntos de entrenamiento y validación deben cruzarse en rondas sucesivas de tal forma que cada dato adquiera la posibilidad de ser validado (Refaeilzadeh, Tang, & Liu, 2009).

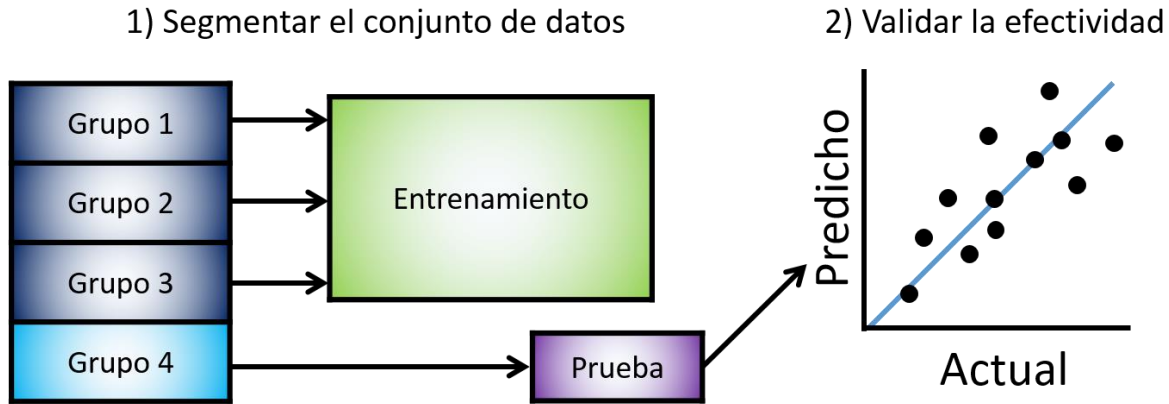


Ilustración 8 Esquema de cross-validation o validación cruzada. Primero se segmenta el conjunto de datos en 2 subconjuntos: entrenamiento y validación. Posteriormente, se entrena el sistema con el primer subconjunto de datos. Por último, se valida la efectividad del sistema con el segundo subconjunto de datos (Cohen et al., 2010).

La forma principal del método de *cross-validation* se denomina *k-fold cross-validation* (validación cruzada con k iteraciones). Ésta implica primero la partición de los datos en k segmentos o pliegues del mismo tamaño. A continuación, se realizan k iteraciones de entrenamiento y validación. En cada iteración se elige un pliegue diferente para validar, mientras que los k-1 pliegues restantes se usan para entrenar. Finalmente, se computa el error promedio de todos los ensayos para estimar el desempeño del sistema o modelo (ver ilustración 9) (Schneider, 1977).

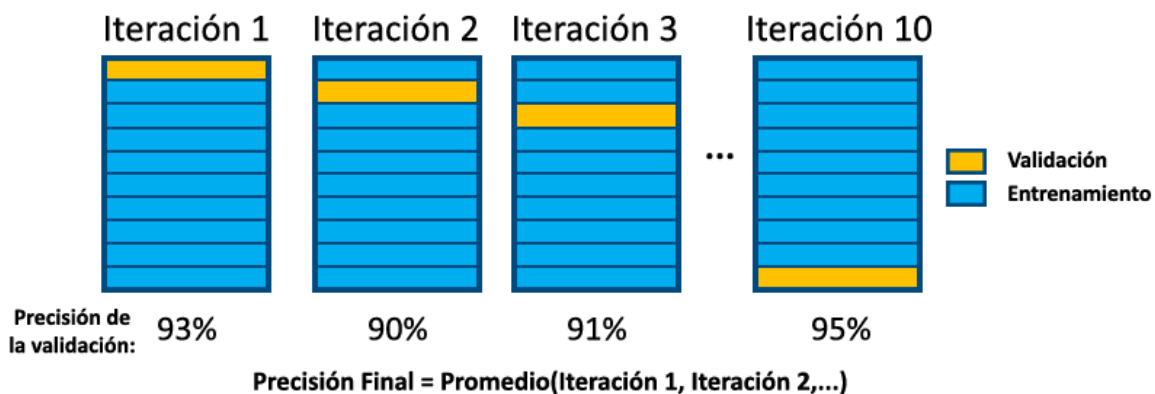


Ilustración 9 Técnica de validación cruzada con 10 iteraciones. En cada iteración se selecciona un subconjunto de datos de entrenamiento y validación diferente. Finalmente, se promedia la predicción de las 10 iteraciones para estimar el desempeño del sistema (Mccormick, 2013).

3.8 División del proyecto en fases y actividades

Para facilitar la consecución de los objetivos específicos, se opta por dividir el proyecto en 5 fases: 1. Análisis del desempeño de las técnicas de aprendizaje de máquina 2. Construcción del subsistema computacional tipo A; 3. Construcción del subsistema computacional tipo B; 4. Integración de los subsistemas computacionales de tipo A y B; 5. Validación del sistema computacional híbrido. A continuación se describe con más detalle cada una de las fases planteadas.

3.8.1 Análisis del desempeño de las técnicas de aprendizaje de máquina

Comprende la elaboración de un estudio comparativo basado en los criterios de exactitud, precisión, sensibilidad, especificidad, área bajo la curva ROC, *Youden's J* y *Cohen's Kappa* para observar el desempeño de las técnicas simbólicas y subsimbólicas de aprendizaje de máquina en la asistencia al diagnóstico de Deterioro Cognitivo Leve. Esta fase contempla la realización de las siguientes actividades:

- Describir el funcionamiento de cada técnica de aprendizaje de máquina considerada en el estudio comparativo.
- Entrenar cada técnica de aprendizaje de máquina con los registros de la base de datos de la *ADNI* bajo un esquema de validación cruzada.
- Entrenar cada técnica de aprendizaje de máquina con los registros de la base de datos del departamento de *Caldas* bajo un esquema de validación cruzada.
- Calcular la matriz de confusión y las curvas ROC para cada técnica de aprendizaje de máquina considerada en el estudio comparativo.
- Analizar el desempeño de cada técnica de aprendizaje de máquina con base en los criterios de exactitud, precisión, sensibilidad, especificidad, área bajo la curva ROC, *Youden's J* y *Cohen's Kappa*.
- Elegir las técnicas simbólicas y subsimbólicas de aprendizaje de máquina a usar en la construcción de los subsistemas computacionales.

3.8.2 Construcción del subsistema computacional tipo A

Comprende la construcción del subsistema computacional basado en una técnica simbólica de aprendizaje de máquina a través de la metodología de desarrollo de software orientada a prototipos evolutivos. Esta fase contempla la realización de las siguientes actividades:

- Extraer los requerimientos del subsistema computacional tipo A.
- Analizar los requerimientos del subsistema computacional tipo A.
- Especificar gráfica y textualmente los elementos del subsistema tipo A.
- Elaborar un prototipo base que satisface los requerimientos más importantes del subsistema tipo A.
- Iterar sobre el prototipo base corrigiendo o adicionando funcionalidad, hasta obtener el subsistema tipo A final.

3.8.3 Construcción del subsistema computacional tipo B

Comprende la construcción del subsistema computacional basado en una técnica subsimbólica de aprendizaje de máquina a través de la metodología de desarrollo de software orientada a prototipos evolutivos. Esta fase contempla la realización de las siguientes actividades:

- Extraer los requerimientos del subsistema computacional tipo B.
- Analizar los requerimientos del subsistema computacional tipo B.
- Especificar gráfica y textualmente los elementos del subsistema tipo B.
- Elaborar un prototipo base que satisface los requerimientos más importantes del subsistema tipo B.
- Iterar sobre el prototipo base corrigiendo o adicionando funcionalidad, hasta obtener el subsistema tipo B final.

3.8.4 Integración de los subsistemas computacionales de tipo A y B

Comprende la integración de los subsistemas computacionales basados en técnicas simbólicas y subsimbólicas de aprendizaje de máquina para obtener el sistema computacional híbrido. Esta fase contempla la realización de las siguientes actividades:

- Elegir el esquema de hibridación para integrar los subsistemas computacionales de tipo A y B.
- Integrar los subsistemas computacionales de tipo A y B siguiendo los lineamientos del esquema de hibridación elegido.
- Realizar pruebas de software para verificar el funcionamiento conjunto de los subsistemas computacionales.

3.8.5 Validación del sistema computacional híbrido

Comprende la validación del funcionamiento del sistema computacional híbrido en el diagnóstico asistido de Deterioro Cognitivo Leve. Lo anterior se lleva a cabo a través de un esquema de *cross-validation* empleando la técnica denominada *10-fold cross-validation* o validación cruzada con 10 iteraciones. Esta fase contempla la realización de las siguientes actividades:

- Segmentar el conjunto de datos en dos clases: entrenamiento y prueba.
- Entrenar el sistema computacional híbrido con el primer subconjunto de datos.
- Validar la efectividad del sistema computacional híbrido con el segundo subconjunto de datos.
- Repetir el procedimiento mencionado de acuerdo con el número de iteraciones
- Observar el funcionamiento al introducir un conjunto de datos desconocido para el sistema computacional híbrido.

4 DESARROLLO Y RESULTADOS

4.1 Fase 1: análisis del desempeño de las técnicas de aprendizaje de máquina

4.1.1 Herramientas empleadas en el análisis de las técnicas

4.1.1.1 *MATLAB: Statistics and Machine Learning Toolbox™*

Statistics and Machine Learning Toolbox™ Es un conjunto de funciones y aplicaciones adicionales para el software MATLAB® que permiten describir, analizar y modelar datos usando la estadística y el aprendizaje de máquina. Estas funciones y aplicaciones permiten realizar estadísticas descriptivas y gráficas para el análisis de datos, ajustar distribuciones de probabilidad, y efectuar pruebas de hipótesis. Los algoritmos designados para regresión y clasificación facilitan la extracción de inferencias a partir de los datos y posibilitan la construcción de modelos predictivos (MathWorks, 2016).

Adicionalmente, dicho *toolbox* proporciona mecanismos para identificar las variables o características claves que impactan a los modelos. Por último, provee diversos algoritmos supervisados y no supervisados de aprendizaje de máquina, incluyendo máquinas de soporte vectorial, árboles de decisión, k-vecinos más cercanos, k-medias, agrupamiento jerárquico, modelos gaussianos, y modelos ocultos de Márkov (MathWorks, 2016).

4.1.1.2 *Scikit-learn: Machine Learning in Python*

Scikit-learn es un marco de trabajo o *framework* para el lenguaje de programación Python que integra una amplia gama de modernos algoritmos de aprendizaje de máquina a fin de resolver problemas supervisados y no supervisados de mediana escala. Este marco se centra en llevar el ML a personas inexpertas en el tema mediante el uso de un lenguaje de programación de alto nivel de propósito general. Asimismo, cuenta con mínimas dependencias de otros paquetes de software y es distribuido bajo la licencia BSD simplificada, lo cual favorece su uso tanto en el ámbito académico como comercial (Pedregosa et al., 2011).

Scikit-learn está construido sobre SciPy (*Scientific Computing Tools for Python*) y provee diversos algoritmos supervisados y no supervisados de aprendizaje de máquina, incluyendo modelos lineales generalizados, análisis discriminante, regresión Ridge, vecinos más cercanos, procesos gaussianos, máquinas de soporte vectorial, «*Naïve-Bayes*», árboles de decisión, métodos de ensamble, y agrupamiento (Scikit-learn, 2015).

4.1.1.3 Keras: Deep Learning Library for Theano and TensorFlow

Keras es una librería minimalista y altamente modular para la construcción de redes neuronales de aprendizaje profundo. Está escrita en el lenguaje de programación Python y puede ejecutarse sobre *TensorFlow* o *Theano* —librerías para realizar operaciones matemáticas en Python— (Keras, 2016).

Keras permite la construcción de prototipos de manera ágil y soporta tanto redes neuronales convolucionales como recurrentes. Igualmente, soporta la creación de esquemas de conectividad arbitrarios (incluyendo su respectivo entrenamiento con múltiples entradas y salidas) y funciona sin inconvenientes tanto en la CPU (*Central processing unit*) como en la GPU (*Graphics processing unit*) (Keras, 2016).

4.1.2 Técnica 1: *Passive-Aggressive Classifier*

El marco de trabajo Pasivo-Agresivo es un planteamiento basado en los principios de la clasificación lineal que aboga por las actualizaciones mínimas de los pesos, es decir, lo que menos se requiera para que la instancia de entrenamiento actual sea clasificada de forma correcta. Una vez el algoritmo ha hecho un pronóstico, recibe realimentación indicándole la salida esperada. Posteriormente, el algoritmo puede modificar su mecanismo de predicción, presumiblemente mejorando las posibilidades de hacer un pronóstico exacto en iteraciones subsecuentes (Crammer, Dekel, Keshet, Shalev-Shwartz, & Singer, 2006).

El clasificador Pasivo-Agresivo es una técnica de entrenamiento *online* que puede ser aplicada a diferentes tareas incluyendo clasificación de múltiples clases, ranking, y

aprendizaje multitarea una vez le sea brindada una función de pérdida. Además, el algoritmo garantiza la terminación en el caso separable y un límite superior para la suma de la función de pérdida en el transcurso del procesamiento de todo el conjunto de datos (Crammer, Dekel, Keshet, Shalev-Shwartz, & Singer, 2006).

4.1.2.1 Construcción, entrenamiento y clasificación

El clasificador pasivo-agresivo se construye empleando la clase *PassiveAggressiveClassifier* perteneciente a los modelos lineales de *Scikit-learn*. El entrenamiento se realiza bajo la técnica de validación cruzada con 10 iteraciones empleando la función *cross_val_score* perteneciente al módulo *cross-validation* de *Scikit-learn*. Finalmente, se generan los estimados para cada registro de ambas bases de datos (*ADNI* y *Caldas*) de acuerdo con las 3 clases de diagnóstico definidas: Sano, Leve y Moderado o Severo. Dichos estimados se obtienen empleando la función *cross_val_predict* perteneciente al módulo *cross-validation* de *Scikit-learn*.

4.1.2.2 Matrices de confusión

En las ilustraciones 10 y 11 se muestran las matrices de confusión obtenidas para el clasificador pasivo-agresivo sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se deduce que el clasificador logró un porcentaje de acierto del 0.52% y 0.68% en las respectivas predicciones. No obstante, se puede observar que existe un alto grado de confusión en las estimaciones de la técnica de aprendizaje de máquina respecto a las tres clases de predicción.

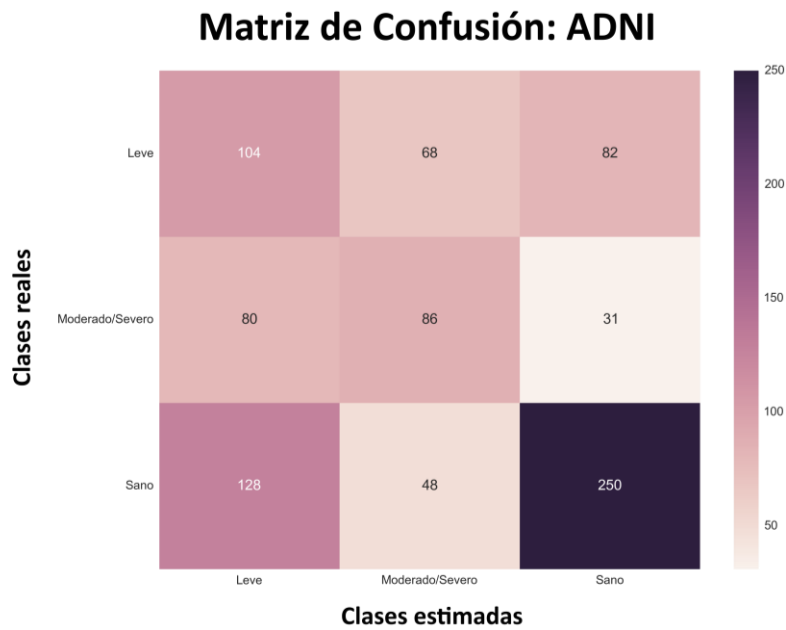


Ilustración 10 Matriz de confusión referente a los estimados del clasificador Pasivo-Agresivo sobre la base de datos de la ADNI.

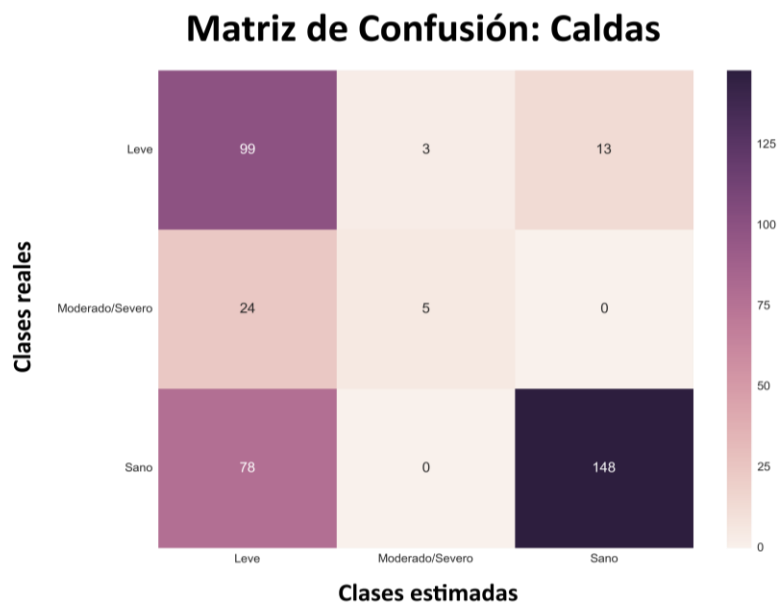


Ilustración 11 Matriz de confusión referente a los estimados del clasificador Pasivo-Agresivo sobre la base de datos de Caldas.

4.1.2.3 Curvas ROC

En las ilustraciones 12 a 17 se muestran las curvas ROC obtenidas para el clasificador pasivo-agresivo sobre las bases de datos de la *ADNI* y *Caldas*. Dado que los estimados se agrupan en 3 clases diferentes, se requiere la misma cantidad de curvas ROC para contrastar el desempeño por cada clase (clase positiva) versus las demás clases restantes (clases negativas). A partir de estas se observa que el AUC para cada una de las clases se encuentra bastante alejada de 1, lo cual indica que la técnica de aprendizaje de máquina presenta grandes dificultades para elegir la clase correcta al momento de realizar las predicciones.

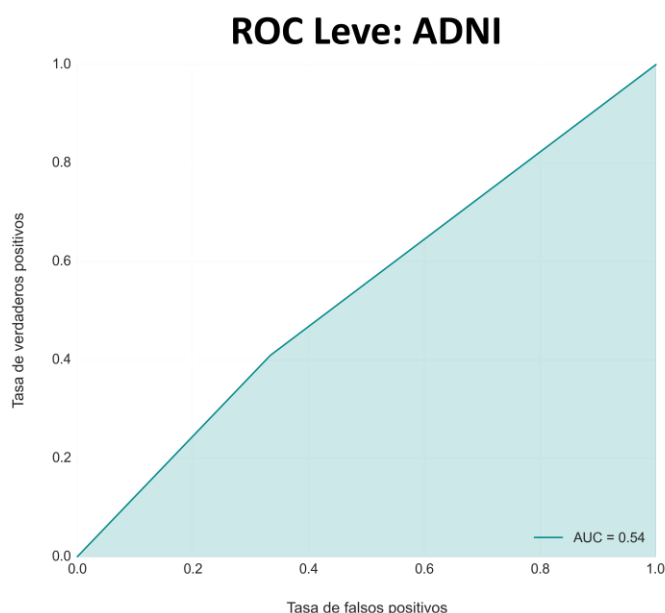


Ilustración 12 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del clasificador Pasivo-Agresivo sobre la base de datos de la ADNI.

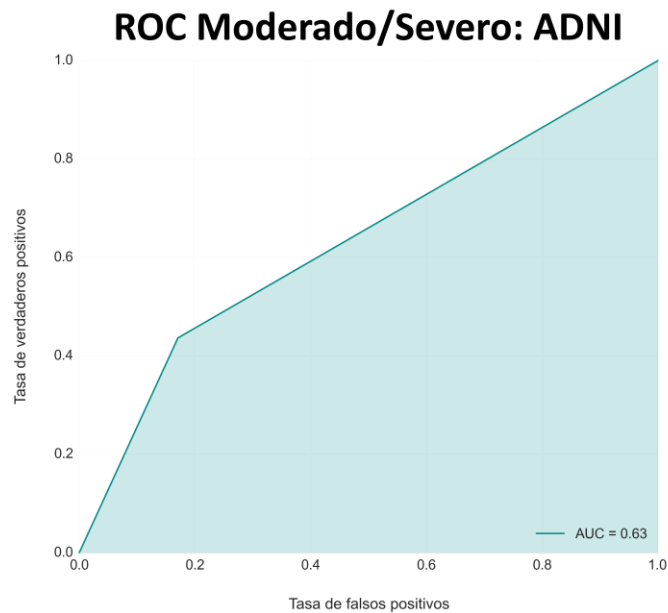


Ilustración 13 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del clasificador Pasivo-Agresivo sobre la base de datos de la ADNI.

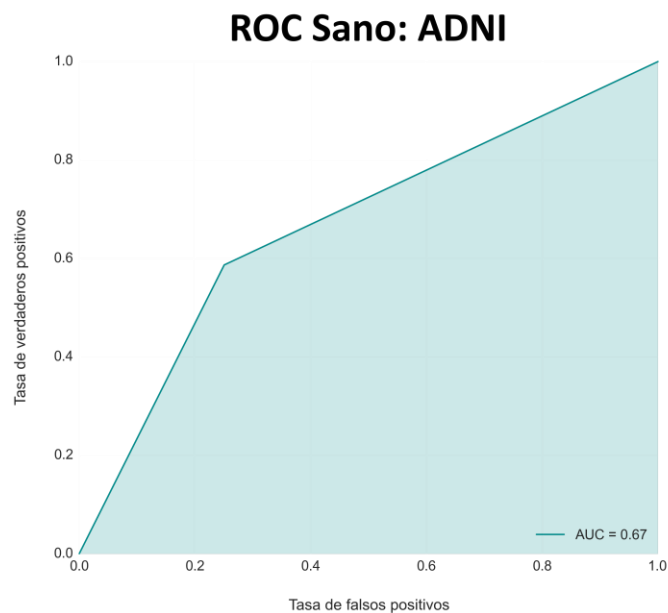


Ilustración 14 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del clasificador Pasivo-Agresivo sobre la base de datos de la ADNI.

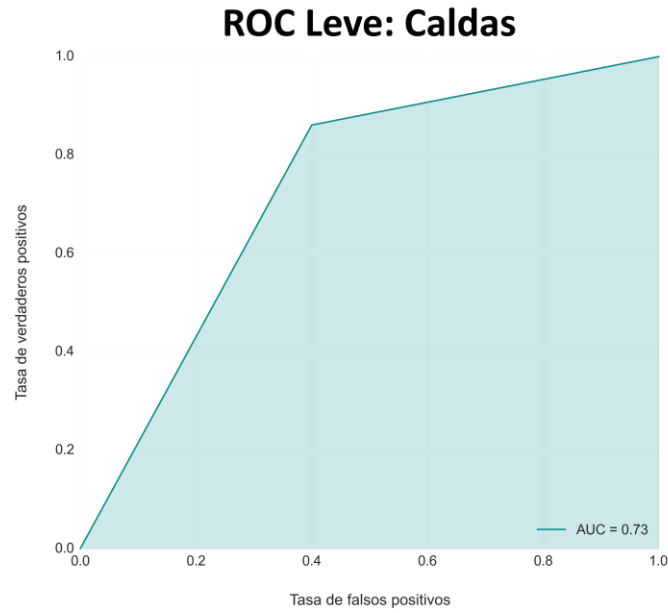


Ilustración 15 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del clasificador Pasivo-Agresivo sobre la base de datos de Caldas.

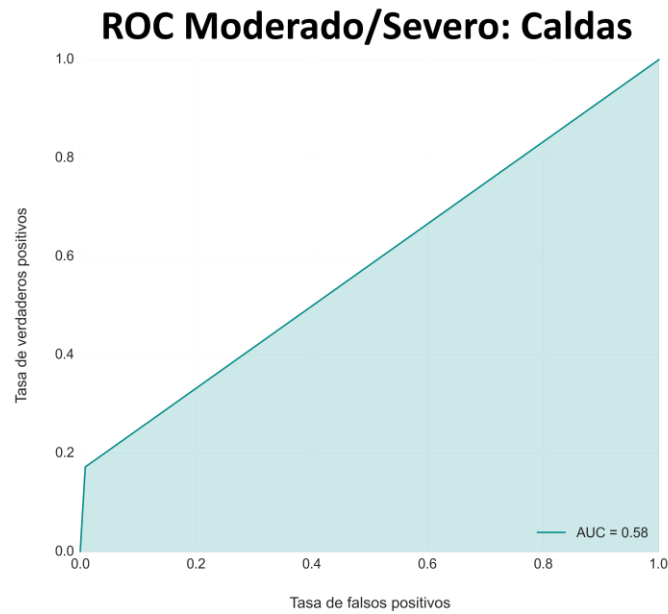


Ilustración 16 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del clasificador Pasivo-Agresivo sobre la base de datos de Caldas.

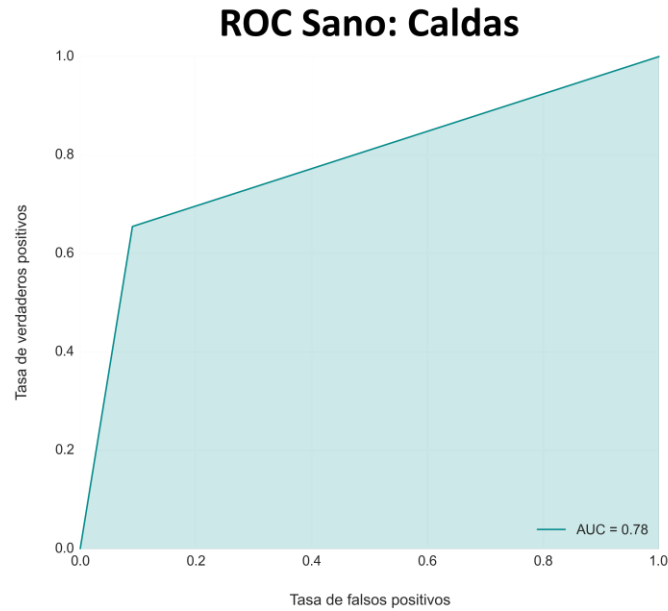


Ilustración 17 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del clasificador Pasivo-Agresivo sobre la base de datos de Caldas.

4.1.2.4 Métricas de desempeño

A continuación se muestran las métricas de desempeño calculadas a partir de las matrices de confusión y de las curvas ROC para el clasificador pasivo-agresivo.

ADNI		Caldas	
Exactitud	0,5017	Exactitud	0,6811
Precisión	0,5267	Precisión	0,7636
TPR (Sensibilidad)	0,5017	TPR (Sensibilidad)	0,6811
Especificidad	0,7433	Especificidad	0,8198
AUC Leve	0,5378	AUC Leve	0,7304
AUC Moderado/Severo	0,6330	AUC Moderado/Severo	0,5818
AUC Sano	0,6682	AUC Sano	0,7823
J Leve	0,0756	J Leve	0,4609
J Moderado/Severo	0,2660	J Moderado/Severo	0,1636
J Sano	0,3363	J Sano	0,5646
Kappa	0,2265	Kappa	0,4342

Tabla 4 Métricas de desempeño para el clasificador Pasivo-Agresivo sobre las bases de datos de la ADNI y Caldas.

4.1.3 Técnica 2: *Ridge Classifier*

Ridge regression es un método para analizar múltiples datos de regresión que sufren de multicolinealidad. Cuando esto último ocurre, los estimados obtenidos por el método de mínimos cuadrados son imparciales pero sus varianzas son grandes, así que pueden estar muy lejos del valor real (NCSS Statistical Software, 2016). La *regression contraída* aborda algunos de los problemas de la regresión por mínimos cuadrados ordinarios mediante la imposición de una penalización en el tamaño de los coeficientes:

$$\min_{\omega} \|X_{\omega} - y\|_2^2 + \alpha \|\omega\|_2^2$$

Aquí, X son las variables dependientes, y es la variable dependiente, ω son los coeficientes de la regresión lineal y $\alpha \geq 0$ es un parámetro de complejidad que controla la cantidad de contracción: cuanto mayor sea el valor de α , mayor es la cantidad de contracción y por lo tanto los coeficientes se vuelven más robustos —frente al problema de colinealidad— (Scikit-learn, 2015).

Ridge classifier es una técnica de aprendizaje de máquina que permite construir un clasificador basado en el método de regresión contraída. Para la estimación con múltiples clases se entrenan n clasificadores en un enfoque de uno-versus-todos, donde n representa la cantidad de clases existentes. Concretamente, lo anterior se lleva a cabo mediante el aprovechamiento del soporte de respuesta multivariado en la regresión contraída (Scikit-learn, 2015).

4.1.3.1 Construcción, entrenamiento y clasificación

El clasificador contraído se construye empleando la clase *RidgeClassifier* perteneciente a los modelos lineales de *Scikit-learn*. El entrenamiento se realiza bajo la técnica de validación cruzada con 10 iteraciones empleando la función *cross_val_score* perteneciente al módulo *cross-validation* de *Scikit-learn*. Finalmente, se generan los estimados para cada registro de ambas bases de datos (*ADNI* y *Caldas*) empleando la función *cross_val_predict* perteneciente al módulo *cross-validation* de *Scikit-learn*.

4.1.3.2 Matrices de confusión

En las ilustraciones 18 y 19 se muestran las matrices de confusión obtenidas para el clasificador contraído sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se deduce que el clasificador logró un porcentaje de acierto del 0.83% y 0.86% en las respectivas predicciones. No obstante, se puede observar que existe un alto grado de confusión en las estimaciones de la técnica de aprendizaje de máquina respecto a las clases de predicción «Leve» y «Moderado/Severo».

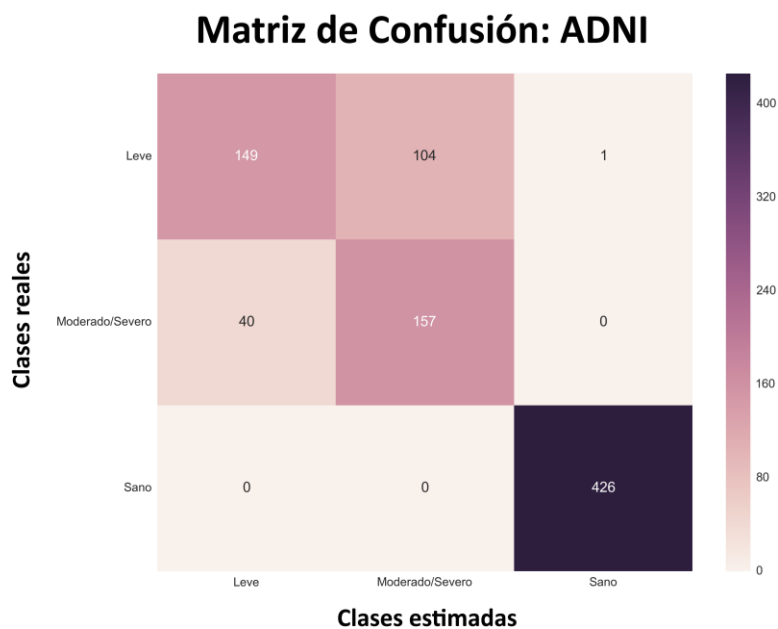


Ilustración 18 Matriz de confusión referente a los estimados del clasificador contraído sobre la base de datos de la ADNI.

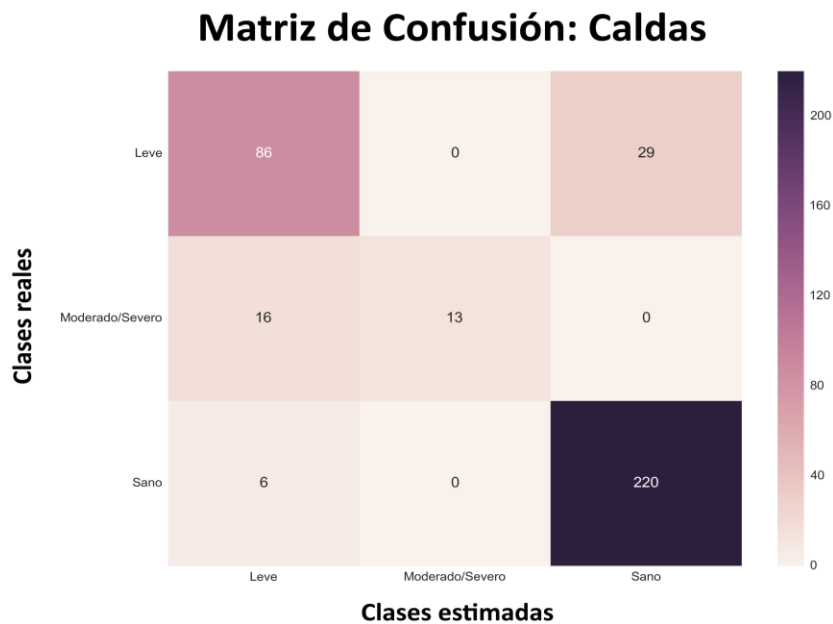


Ilustración 19 Matriz de confusión referente a los estimados del clasificador contraído sobre la base de datos de Caldas.

4.1.3.3 Curvas ROC

En las ilustraciones 20 a 25 se muestran las curvas ROC obtenidas para el clasificador contraído sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se observa que las AUC para las clases «Leve» y «Moderado/Severo» se encuentran medianamente alejadas de 1, lo cual indica que la técnica de aprendizaje de máquina presenta algunas dificultades para elegir la clase correcta entre las dos mencionadas, al momento de realizar las predicciones. No obstante, el AUC para la clase «Sano» se encuentra bastante cercana a 1, es decir que la técnica es capaz de diferenciar dicha clase de las demás con suma fidelidad.

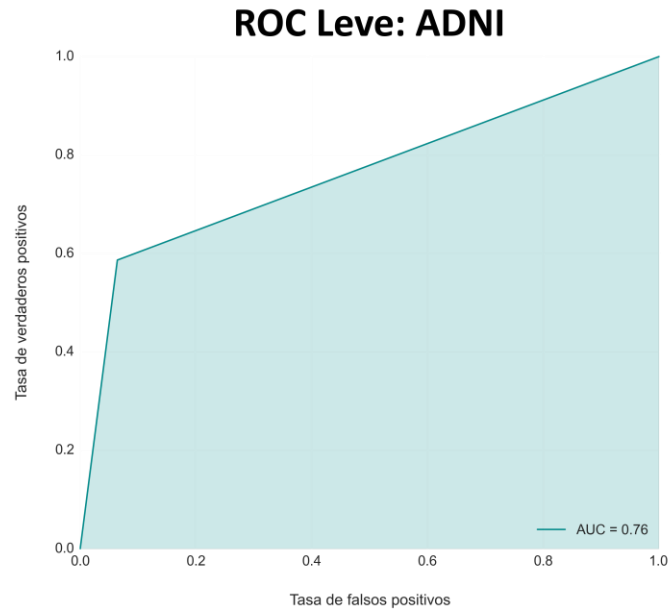


Ilustración 20 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del clasificador contraído sobre la base de datos de la ADNI.

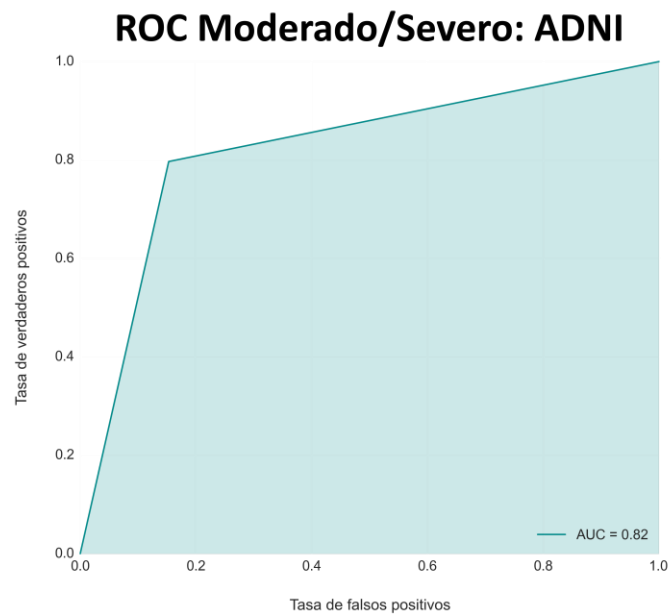


Ilustración 21 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del clasificador contraído sobre la base de datos de la ADNI.

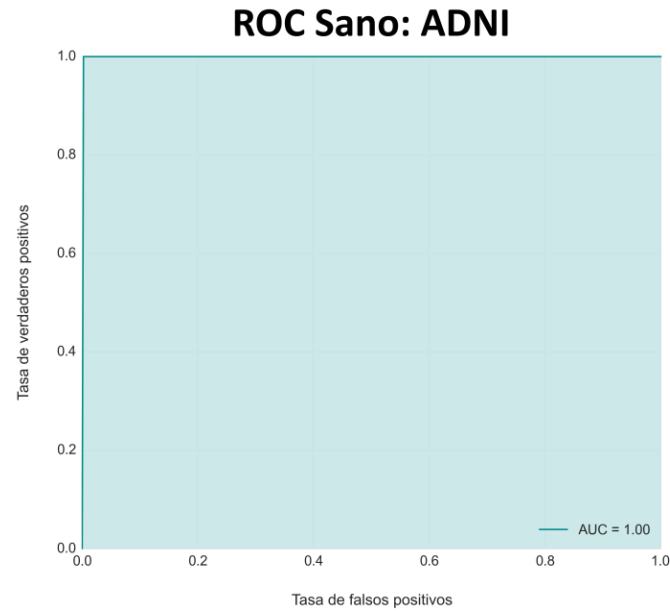


Ilustración 22 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo— referente a los estimados del clasificador contraído sobre la base de datos de la ADNI.

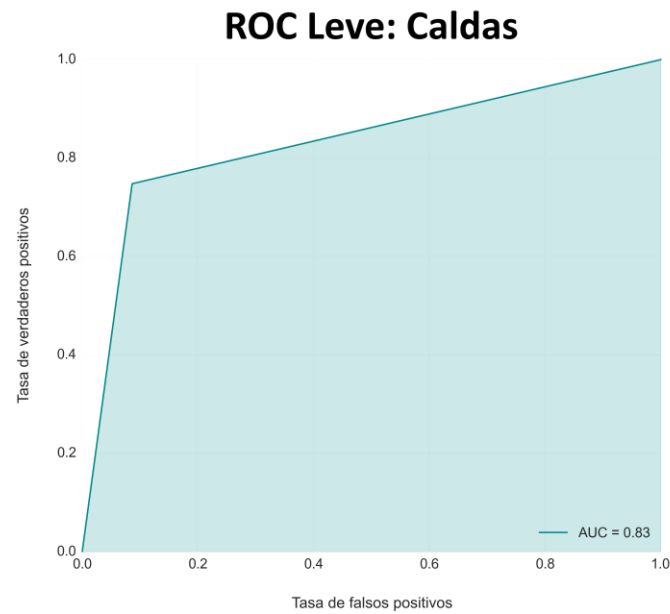


Ilustración 23 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del clasificador contraído sobre la base de datos de Caldas.

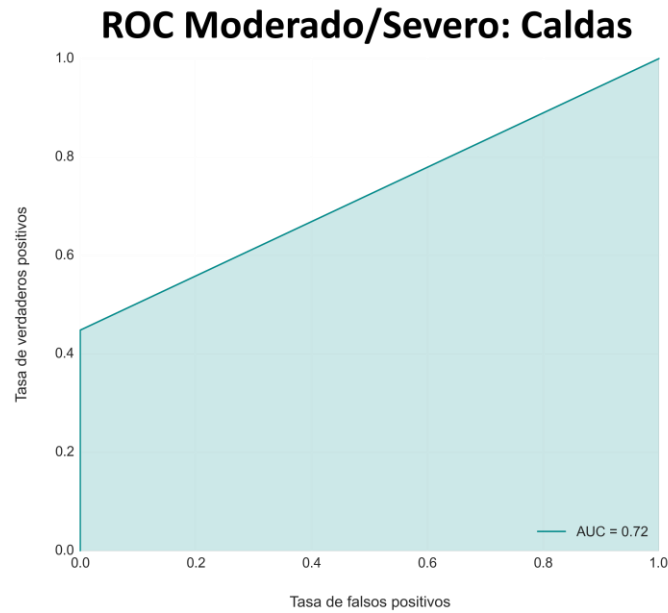


Ilustración 24 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del clasificador contraído sobre la base de datos de Caldas.

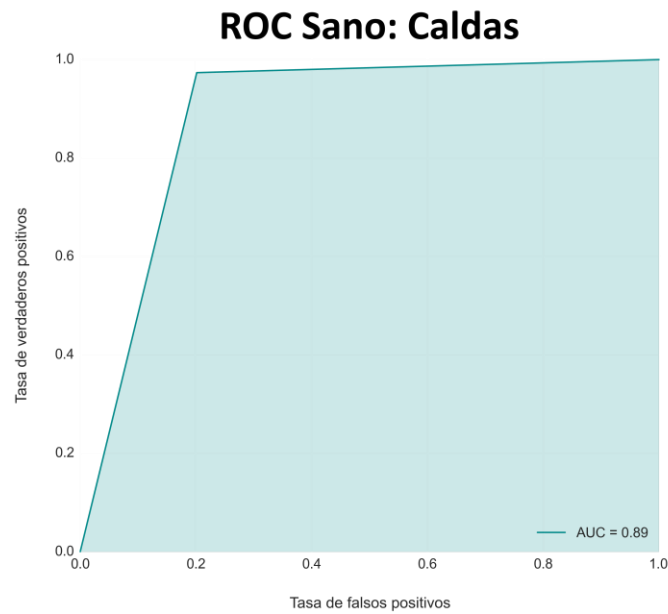


Ilustración 25 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del clasificador contraído sobre la base de datos de Caldas.

4.1.3.4 Métricas de desempeño

A continuación se muestran las métricas de desempeño calculadas a partir de las matrices de confusión y de las curvas ROC para el clasificador contraído.

ADNI		Caldas	
Exactitud	0,8347	Exactitud	0,8622
Precisión	0,8481	Precisión	0,8655
TPR (Sensibilidad)	0,8347	TPR (Sensibilidad)	0,8622
Especificidad	0,9460	Especificidad	0,8502
AUC Leve	0,7612	AUC Leve	0,8308
AUC Moderado/Severo	0,8220	AUC Moderado/Severo	0,7241
AUC Sano	0,9989	AUC Sano	0,8860
J Leve	0,5224	J Leve	0,6616
J Moderado/Severo	0,6440	J Moderado/Severo	0,4483
J Sano	0,9978	J Sano	0,7721
Kappa	0,7393	Kappa	0,7218

Tabla 5 Métricas de desempeño para el clasificador contraído sobre las bases de datos de la ADNI y Caldas.

4.1.4 Técnica 3: *Linear SVM*

Una máquina de soporte vectorial es una técnica para clasificar datos mediante el hallazgo del mejor hiperplano que separa todos los puntos de datos de una clase de los de la otra clase. El mejor hiperplano para una SVM significa aquel que posee el mayor margen entre las clases. El margen refiere el ancho máximo del bloque paralelo al hiperplano que no contiene puntos de datos en su interior (MathWorks, 2016).

Se denominan vectores soporte a los puntos de datos que se encuentran más cercanos del hiperplano de separación; estos puntos están en el límite del bloque paralelo al hiperplano (MathWorks, 2016). La siguiente ilustración muestra las definiciones mencionadas anteriormente, con símbolos + indicando los puntos de datos de tipo 1, y símbolos – indicando los puntos de datos de tipo -1.

Máquina de soporte vectorial

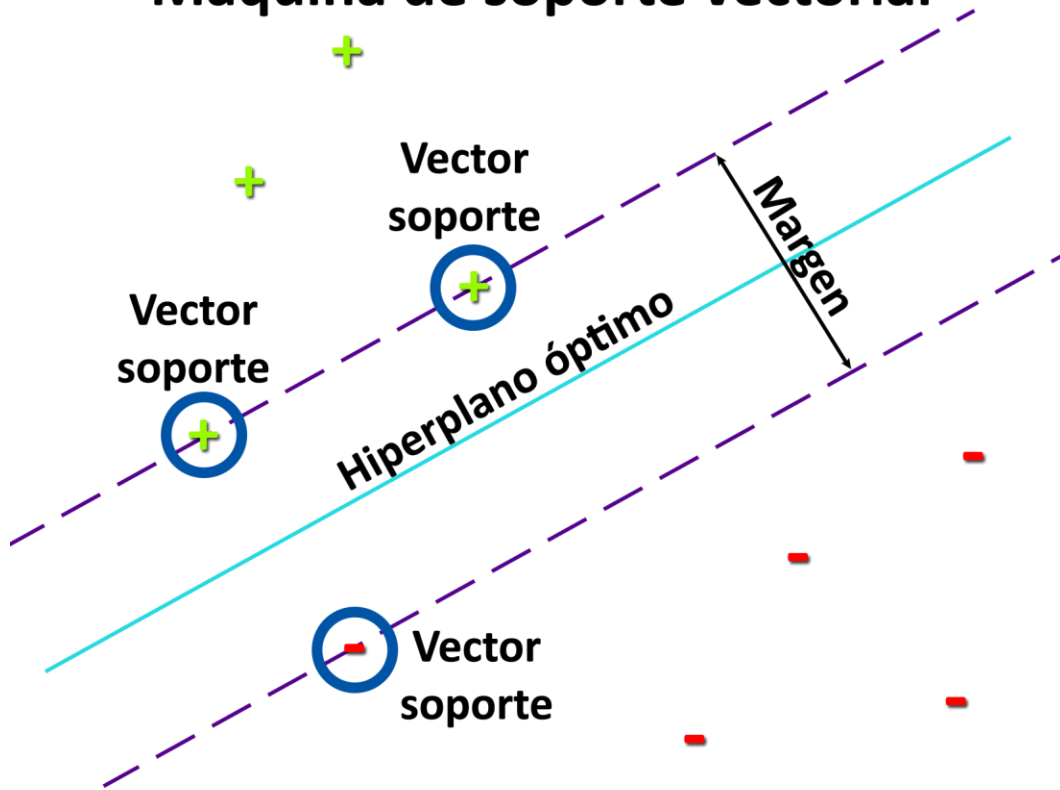


Ilustración 26 Estructura de una máquina de soporte vectorial que separa las clases + y - por medio de un hiperplano. Los puntos más cercanos al hiperplano se denominan vectores soporte (MathWorks, 2016).

Dado un conjunto de datos de entrenamiento (x_i, y_i) para $i = 1 \dots N$, con $x_i \in \mathbb{R}^d$ y $y_i \in \{-1, 1\}$, se puede formular una SVM para aprender un clasificador $f(x)$ mediante la resolución de un problema de optimización sobre los pesos α_i :

$$f(x) = \sum_{i=1}^N \alpha_i y_i k(x, x_i) + b$$

El término $k(x, x_i)$ se denomina el *kernel* o núcleo de la máquina y puede ser sustituido por diversas funciones dependiendo del tipo de SVM (Zisserman, 2015). Finalmente, las *linear SVMs* son aquellas que poseen un núcleo lineal $k(x, x_i) = x^\top x_i$, esto es:

$$f(x) = \sum_{i=1}^N \alpha_i y_i (x^\top x_i) + b$$

4.1.4.1 Construcción, entrenamiento y clasificación

La SVM lineal se construye empleando la función `templateSVM('KernelFunction', 'linear',...)` perteneciente a los algoritmos de clasificación de *Statistics and Machine Learning Toolbox*TM. El entrenamiento se realiza bajo la técnica de validación cruzada con 10 iteraciones empleando las funciones `fitcecoc` y `crossval` pertenecientes a los algoritmos de regresión y clasificación de *Statistics and Machine Learning Toolbox*TM. Finalmente, se generan los estimados para cada registro de ambas bases de datos (*ADNI* y *Caldas*) empleando la función `kfoldPredict` perteneciente a los algoritmos de clasificación de *Statistics and Machine Learning Toolbox*TM.

4.1.4.2 Matrices de confusión

En las ilustraciones 27 y 28 se muestran las matrices de confusión obtenidas para la SVM lineal sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se deduce que el clasificador logró un porcentaje de acierto del 0.89% y 0.96% en las respectivas predicciones. Asimismo, se puede observar que existe un bajo grado de confusión en las estimaciones de la técnica de aprendizaje de máquina respecto a las clases de predicción.

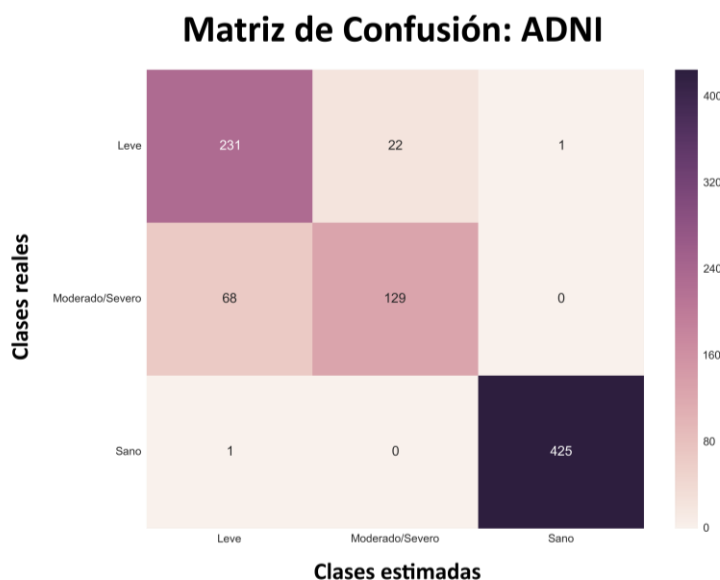


Ilustración 27 Matriz de confusión referente a los estimados de la SVM lineal sobre la base de datos de la *ADNI*.

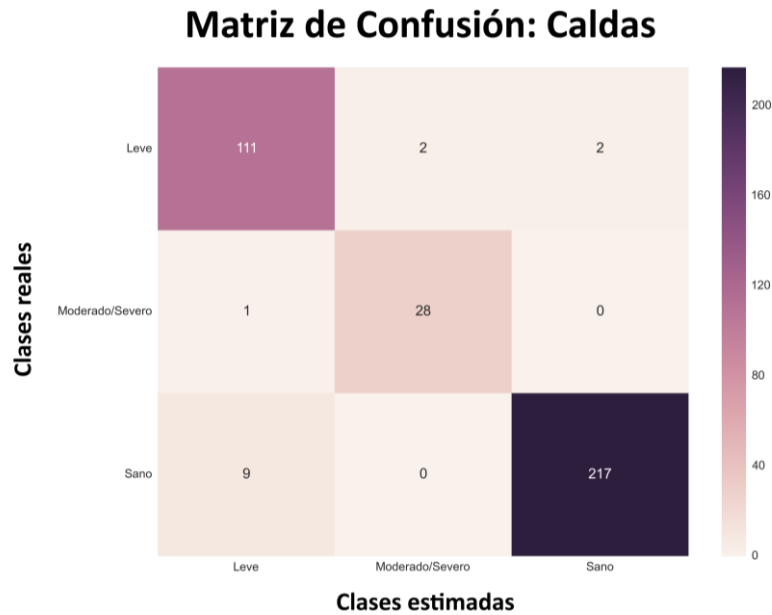


Ilustración 28 Matriz de confusión referente a los estimados de la SVM lineal sobre la base de datos de Caldas.

4.1.4.3 Curvas ROC

En las ilustraciones 29 a 34 se muestran las curvas ROC obtenidas para la SVM lineal sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se observa que el AUC para cada una de las clases se encuentra bastante cercana de 1, lo cual indica que la técnica de aprendizaje de máquina no presenta mayores dificultades para elegir la clase correcta al momento de realizar las predicciones. Solamente se presentan algunos inconvenientes menores para diferenciar las clases «Leve» y «Moderado/Severo» de las demás en la base de datos de la *ADNI*.

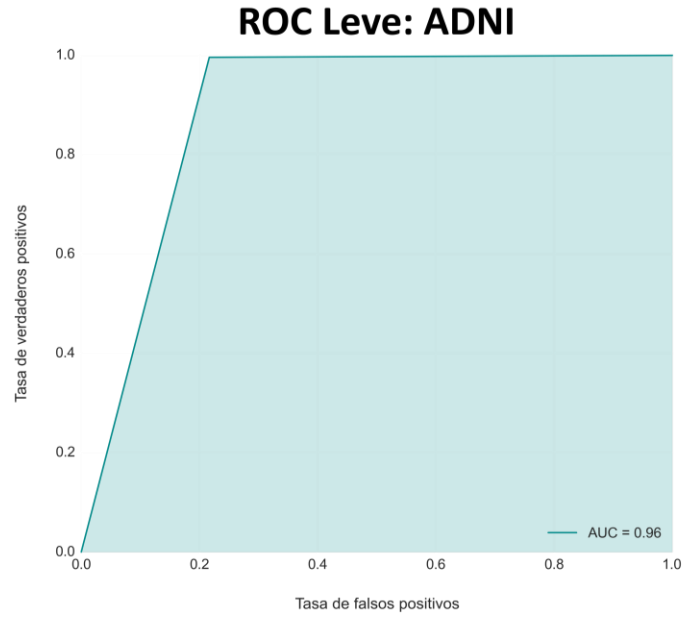


Ilustración 29 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la SVM lineal sobre la base de datos de la ADNI.

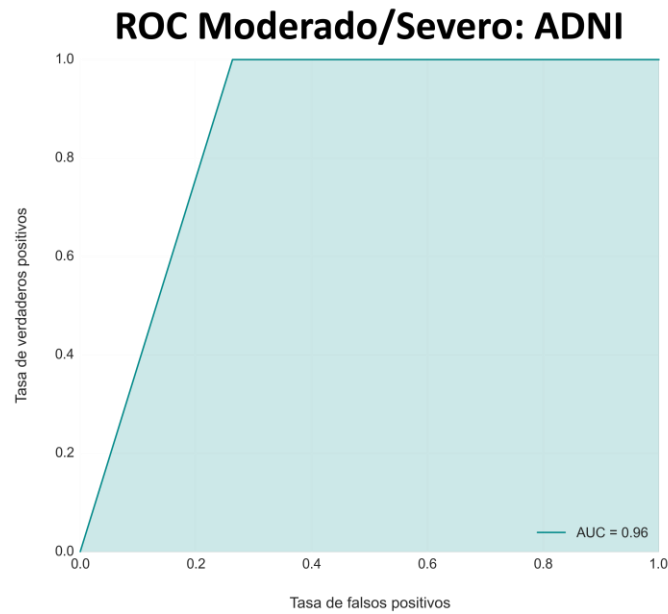


Ilustración 30 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la SVM lineal sobre la base de datos de la ADNI.

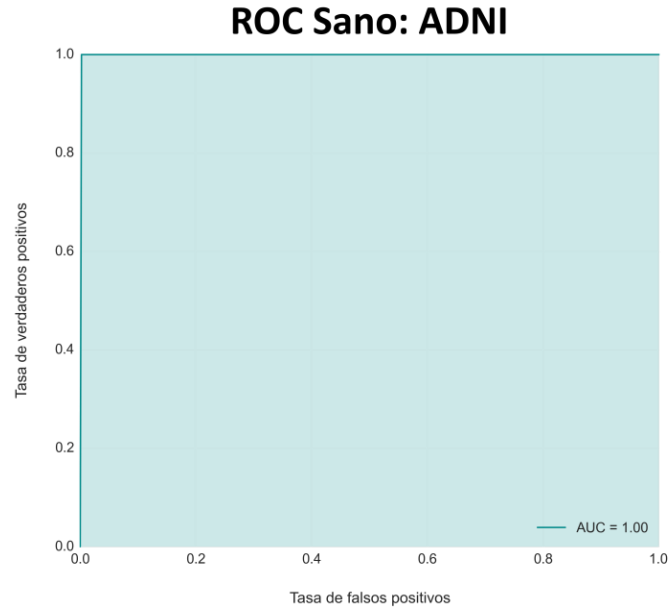


Ilustración 31 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo— referente a los estimados de la SVM lineal sobre la base de datos de la ADNI.

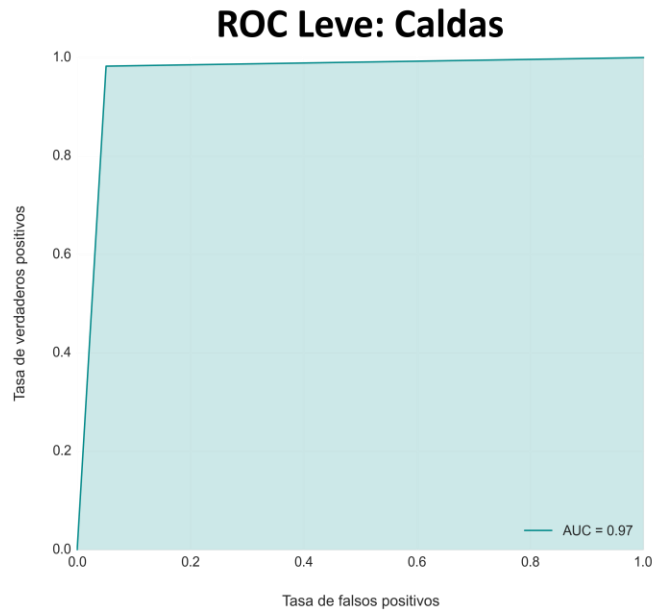


Ilustración 32 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la SVM lineal sobre la base de datos de Caldas.

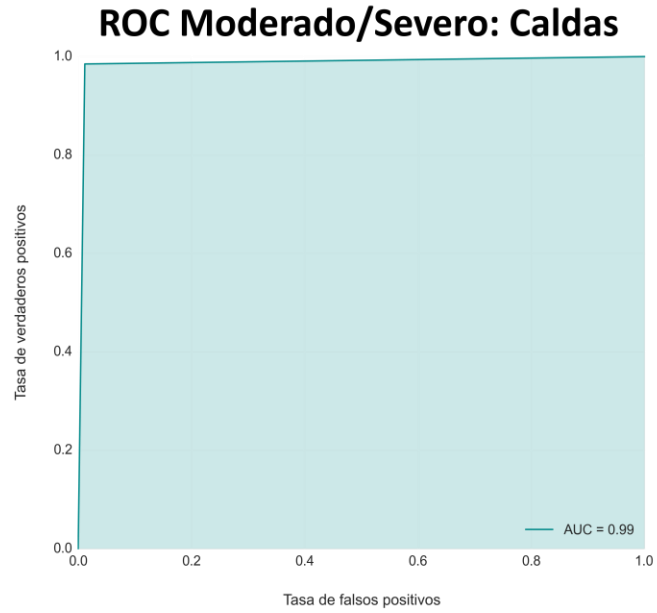


Ilustración 33 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la SVM lineal sobre la base de datos de Caldas.

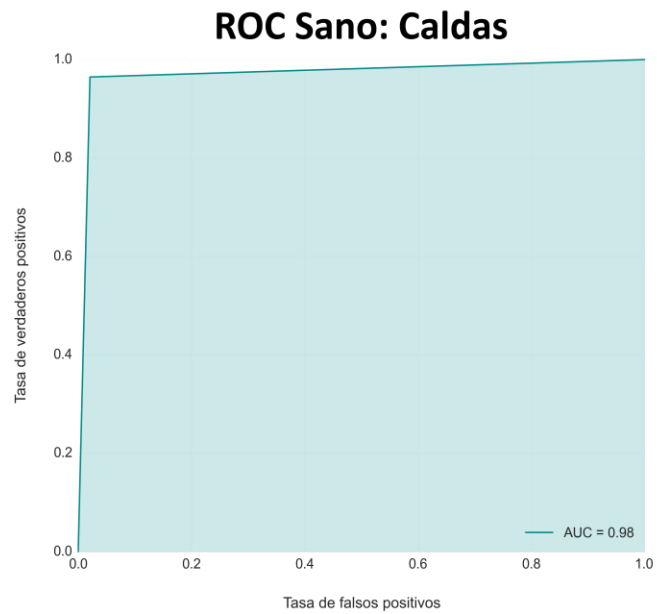


Ilustración 34 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la SVM lineal sobre la base de datos de Caldas.

4.1.4.4 Métricas de desempeño

A continuación se muestran las métricas de desempeño calculadas a partir de las matrices de confusión y de las curvas ROC para la SVM lineal.

ADNI		Caldas	
Exactitud	0,8951	Exactitud	0,9622
Precisión	0,8995	Precisión	0,9635
TPR (Sensibilidad)	0,8951	TPR (Sensibilidad)	0,9622
Especificidad	0,9596	Especificidad	0,9789
AUC Leve	0,9614	AUC Leve	0,9716
AUC Moderado/Severo	0,9643	AUC Moderado/Severo	0,9864
AUC Sano	0,9987	AUC Sano	0,9798
J Leve	0,7987	J Leve	0,9260
J Moderado/Severo	0,6225	J Moderado/Severo	0,9597
J Sano	0,9954	J Sano	0,9463
Kappa	0,8325	Kappa	0,9287

Tabla 6 Métricas de desempeño para la SVM lineal sobre las bases de datos de la ADNI y Caldas.

4.1.5 Técnica 4: Quadratic SVM

Algunos problemas de clasificación no poseen hiperplanos simples como un criterio de separación útil. Para estos, existe una variante del enfoque matemático que conserva casi toda la simplicidad de un hiperplano de separación de SVM; dicho enfoque se basa en la teoría de los *kernels* (MathWorks, 2016). Un ejemplo práctico son las SVM polinomiales cuya clasificación se encuentra descrita por un núcleo polinomial $k(x, x_i) = (1 + x^\top x_i)^d$ para algún entero positivo d , esto es:

$$f(x) = \sum_{i=1}^N \alpha_i y_i (1 + x^\top x_i)^d + b$$

El enfoque matemático que emplea los *kernels* se basa en los métodos de cómputo de hiperplanos. La totalidad de los cálculos para la clasificación usan solamente productos punto. Por ende, los núcleos polinomiales pueden utilizar los mismos cálculos y algoritmos para obtener soluciones no lineales. Los clasificadores resultantes son hipersuperficies en

algún espacio s , pero este espacio no requiere ser identificado o examinado (MathWorks, 2016).

Según lo anterior, es posible definir las *Quadratic SVMs* en términos de un núcleo polinomial de segundo grado $k(x, x_i) = (1 + x^T x_i)^2$, esto es:

$$f(x) = \sum_{i=1}^N \alpha_i y_i (1 + x^T x_i)^2 + b$$

4.1.5.1 Construcción, entrenamiento y clasificación

La SVM cuadrática se construye empleando la función `templateSVM('KernelFunction', 'polynomial', 'PolynomialOrder', 2,...)` perteneciente a los algoritmos de clasificación de *Statistics and Machine Learning Toolbox*TM. El entrenamiento se realiza bajo la técnica de validación cruzada con 10 iteraciones empleando las funciones `fitcecoc` y `crossval` pertenecientes a los algoritmos de regresión y clasificación de *Statistics and Machine Learning Toolbox*TM. Finalmente, se generan los estimados para cada registro de ambas bases de datos (*ADNI* y *Caldas*) empleando la función `kfoldPredict` perteneciente a los algoritmos de clasificación de *Statistics and Machine Learning Toolbox*TM.

4.1.5.2 Matrices de confusión

En las ilustraciones 35 y 36 se muestran las matrices de confusión obtenidas para la SVM cuadrática sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se deduce que el clasificador logró un porcentaje de acierto del 0.88% y 0.96% en las respectivas predicciones. Asimismo, se puede observar que existe un bajo grado de confusión en las estimaciones de la técnica de aprendizaje de máquina respecto a las clases de predicción.

Matriz de Confusión: ADNI

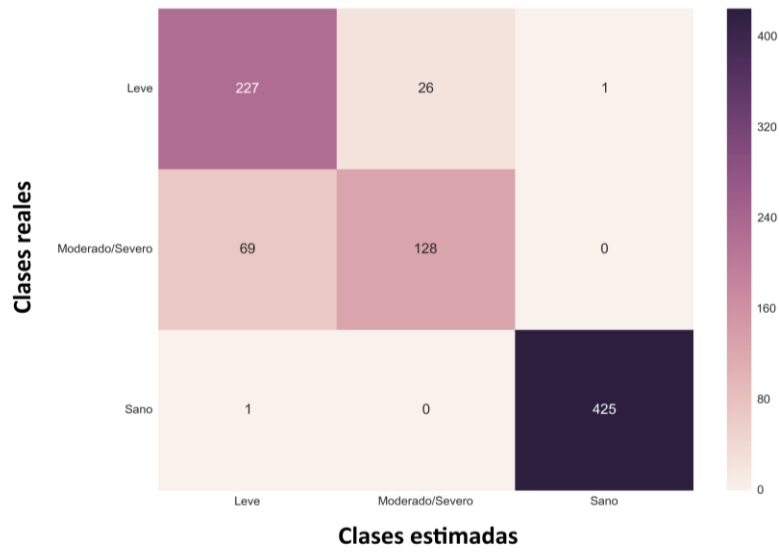


Ilustración 35 Matriz de confusión referente a los estimados de la SVM cuadrática sobre la base de datos de la ADNI.

Matriz de Confusión: Caldas

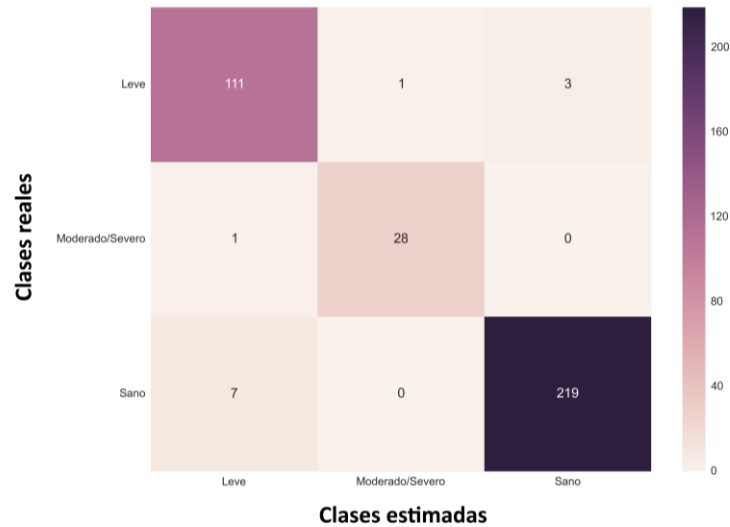


Ilustración 36 Matriz de confusión referente a los estimados de la SVM cuadrática sobre la base de datos de Caldas.

4.1.5.3 Curvas ROC

En las ilustraciones 37 a 42 se muestran las curvas ROC obtenidas para la SVM cuadrática sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se observa que el AUC para cada una de las clases se encuentra bastante cercana de 1, lo cual indica que la técnica de aprendizaje de máquina no presenta mayores dificultades para elegir la clase correcta al momento de realizar las predicciones. Solamente se presentan algunos inconvenientes menores para diferenciar las clases «Leve» y «Moderado/Severo» de las demás en la base de datos de la *ADNI*.

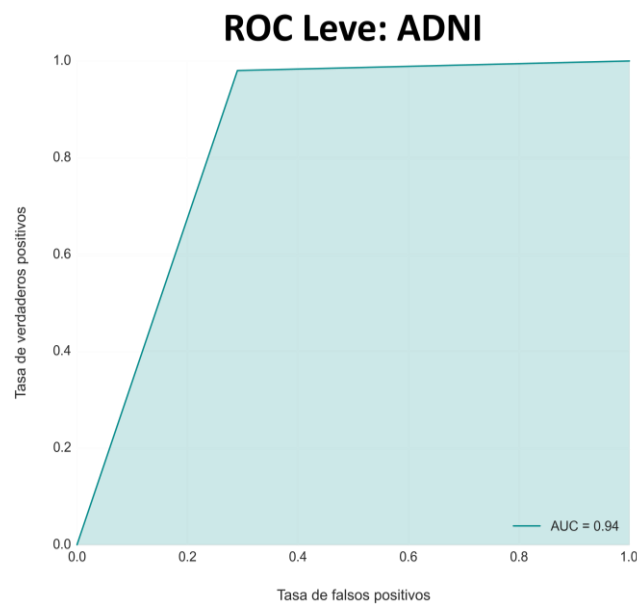


Ilustración 37 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la SVM cuadrática sobre la base de datos de la ADNI.

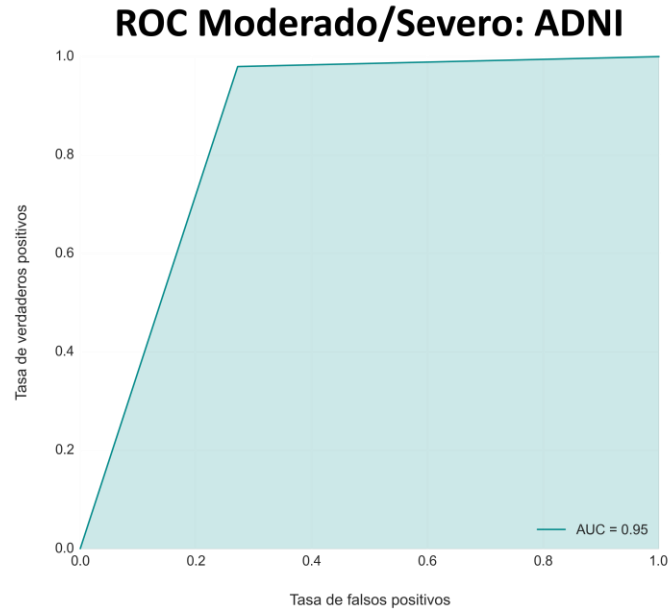


Ilustración 38 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la SVM cuadrática sobre la base de datos de la ADNI.

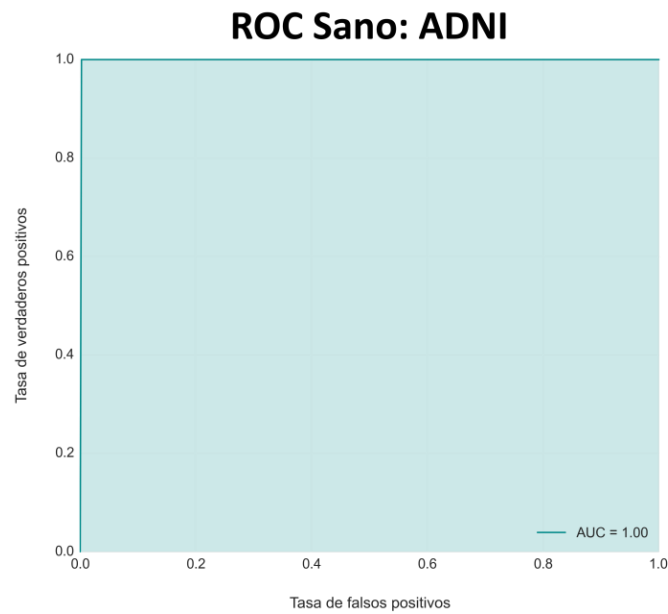


Ilustración 39 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la SVM cuadrática sobre la base de datos de la ADNI.

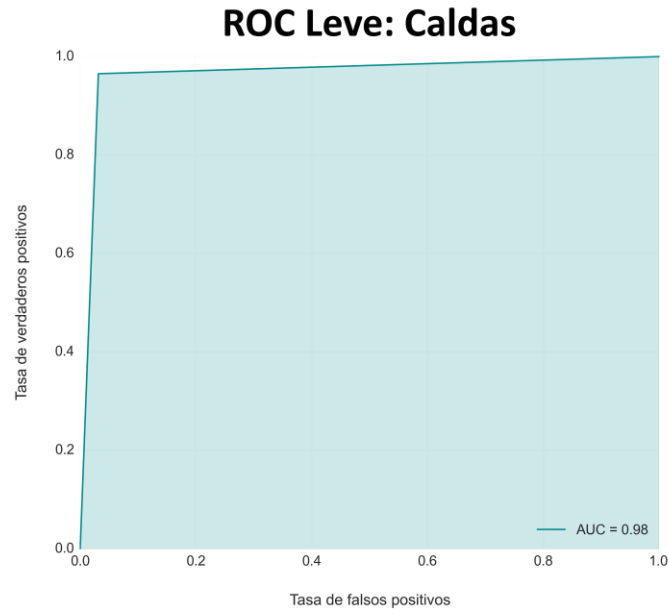


Ilustración 40 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la SVM cuadrática sobre la base de datos de Caldas.

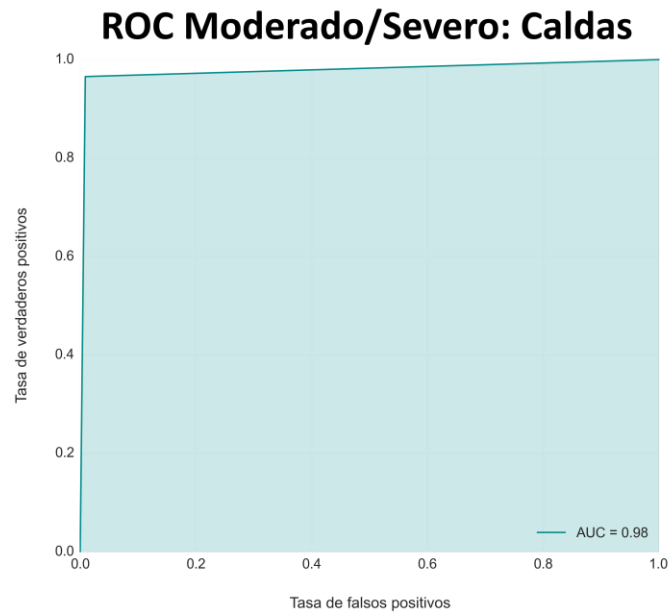


Ilustración 41 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la SVM cuadrática sobre la base de datos de Caldas.

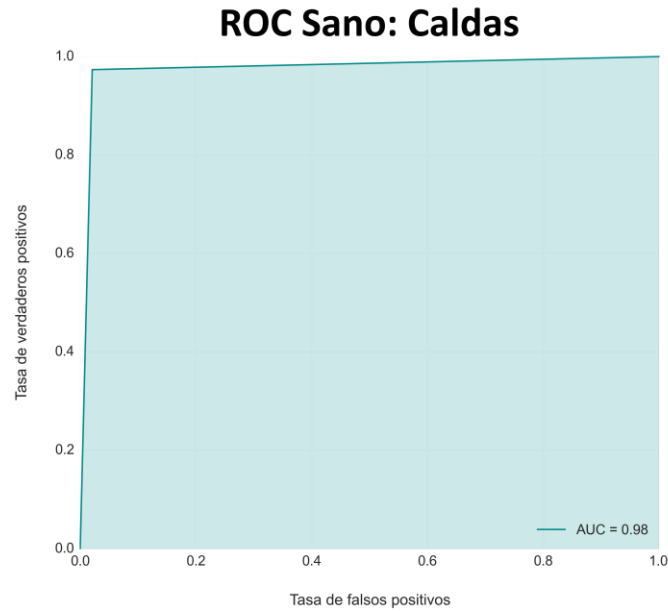


Ilustración 42 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la SVM cuadrática sobre la base de datos de Caldas.

4.1.5.4 Métricas de desempeño

A continuación se muestran las métricas de desempeño calculadas a partir de las matrices de confusión y de las curvas ROC para la SVM cuadrática.

ADNI		Caldas	
Exactitud	0,8894	Exactitud	0,9676
Precisión	0,8927	Precisión	0,9681
TPR (Sensibilidad)	0,8894	TPR (Sensibilidad)	0,9676
Especificidad	0,9578	Especificidad	0,9773
AUC Leve	0,9394	AUC Leve	0,9765
AUC Moderado/Severo	0,9517	AUC Moderado/Severo	0,9825
AUC Sano	0,9987	AUC Sano	0,9823
J Leve	0,7813	J Leve	0,9338
J Moderado/Severo	0,6115	J Moderado/Severo	0,9626
J Sano	0,9954	J Sano	0,9482
Kappa	0,8235	Kappa	0,9385

Tabla 7 Métricas de desempeño para la SVM cuadrática sobre las bases de datos de la ADNI y Caldas.

4.1.6 Técnica 5: *Cubic SVM*

De acuerdo con la sección previa, es posible definir las *Cubic SVMs* en términos de un núcleo polinomial de tercer grado $k(x, x_i) = (1 + x^T x_i)^3$, esto es:

$$f(x) = \sum_{i=1}^N \alpha_i y_i (1 + x^T x_i)^3 + b$$

4.1.6.1 Construcción, entrenamiento y clasificación

La SVM cúbica se construye empleando la función `templateSVM('KernelFunction', 'polynomial', 'PolynomialOrder', 3,...)` perteneciente a los algoritmos de clasificación de *Statistics and Machine Learning Toolbox*TM. El entrenamiento se realiza bajo la técnica de validación cruzada con 10 iteraciones empleando las funciones `fitcecoc` y `crossval` pertenecientes a los algoritmos de regresión y clasificación de *Statistics and Machine Learning Toolbox*TM. Finalmente, se generan los estimados para cada registro de ambas bases de datos (*ADNI* y *Caldas*) empleando la función `kfoldPredict` perteneciente a los algoritmos de clasificación de *Statistics and Machine Learning Toolbox*TM.

4.1.6.2 Matrices de confusión

En las ilustraciones 43 y 44 se muestran las matrices de confusión obtenidas para la SVM cúbica sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se deduce que el clasificador logró un porcentaje de acierto del 0.88% y 0.94% en las respectivas predicciones. Asimismo, se puede observar que existe un bajo grado de confusión en las estimaciones de la técnica de aprendizaje de máquina respecto a las clases de predicción.

Matriz de Confusión: ADNI

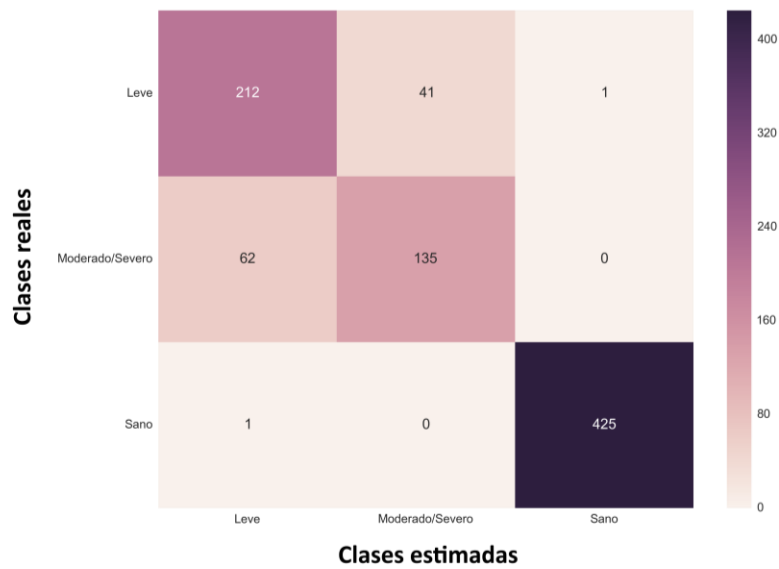


Ilustración 43 Matriz de confusión referente a los estimados de la SVM cúbica sobre la base de datos de la ADNI.

Matriz de Confusión: Caldas

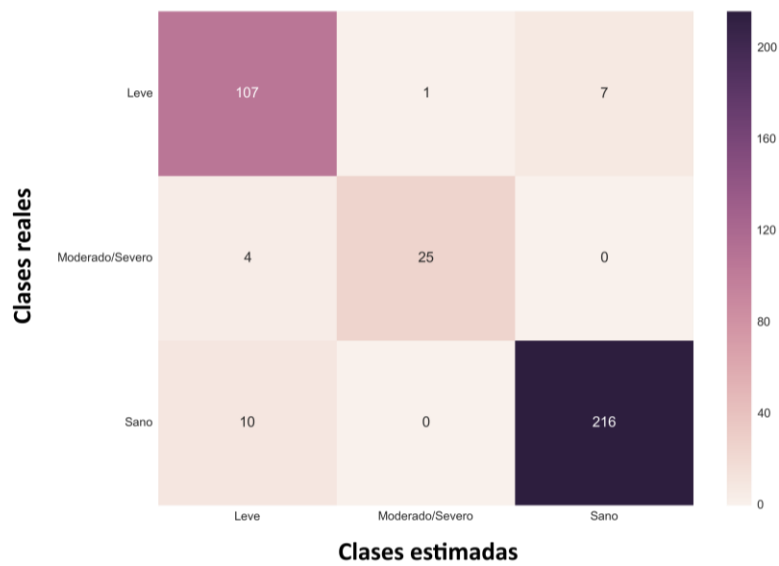


Ilustración 44 Matriz de confusión referente a los estimados de la SVM cúbica sobre la base de datos de Caldas.

4.1.6.3 Curvas ROC

En las ilustraciones 45 a 50 se muestran las curvas ROC obtenidas para la SVM cúbica sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se observa que el AUC para cada una de las clases se encuentra bastante cercana de 1, lo cual indica que la técnica de aprendizaje de máquina no presenta mayores dificultades para elegir la clase correcta al momento de realizar las predicciones. Solamente se presentan algunos inconvenientes menores para diferenciar las clases «Leve» y «Moderado/Severo» de las demás en la base de datos de la *ADNI*.

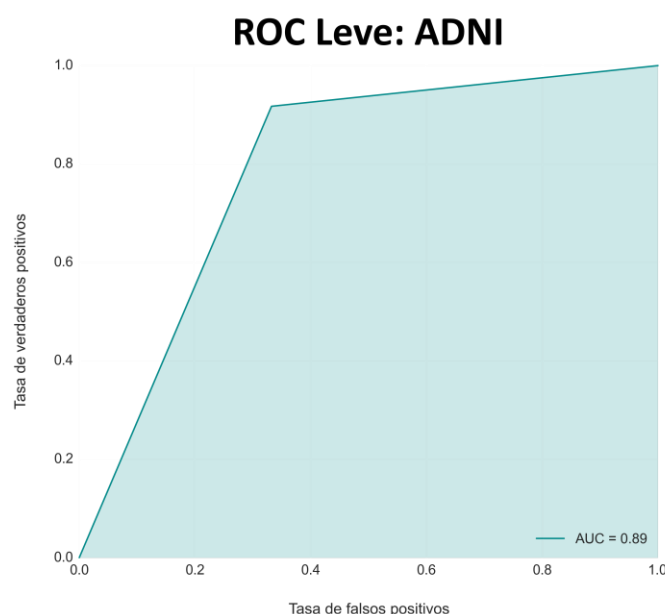


Ilustración 45 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la SVM cúbica sobre la base de datos de la ADNI.

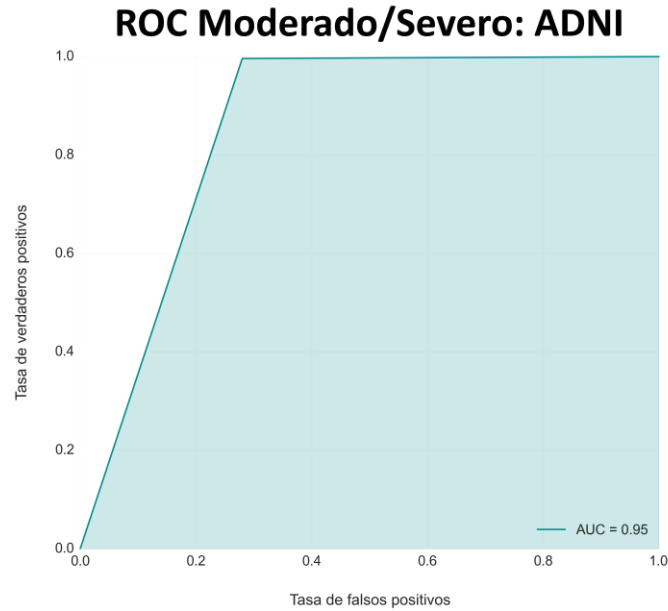


Ilustración 46 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la SVM cúbica sobre la base de datos de la ADNI.

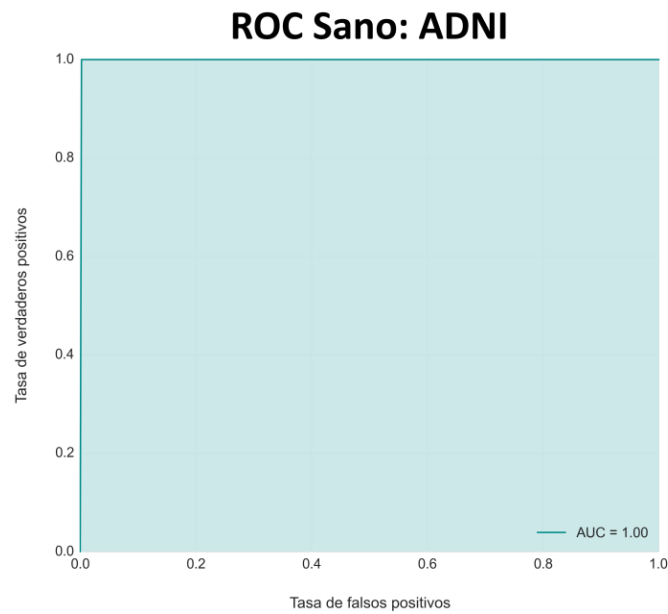


Ilustración 47 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo— referente a los estimados de la SVM cúbica sobre la base de datos de la ADNI.

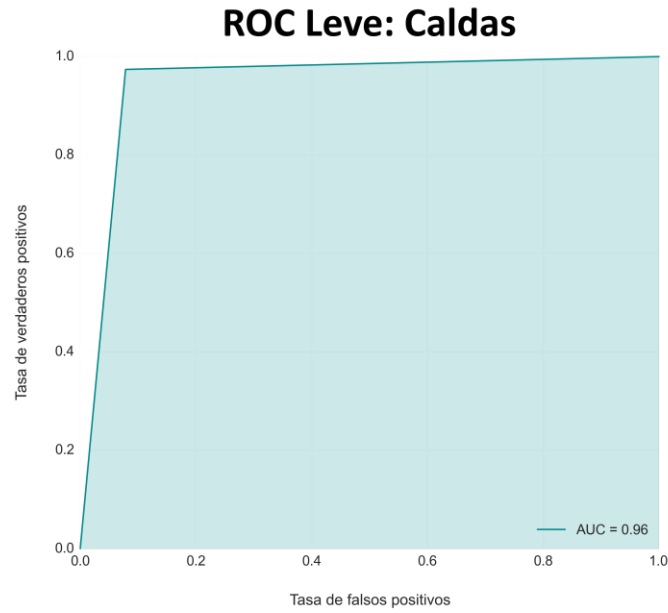


Ilustración 48 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la SVM cúbica sobre la base de datos de Caldas.

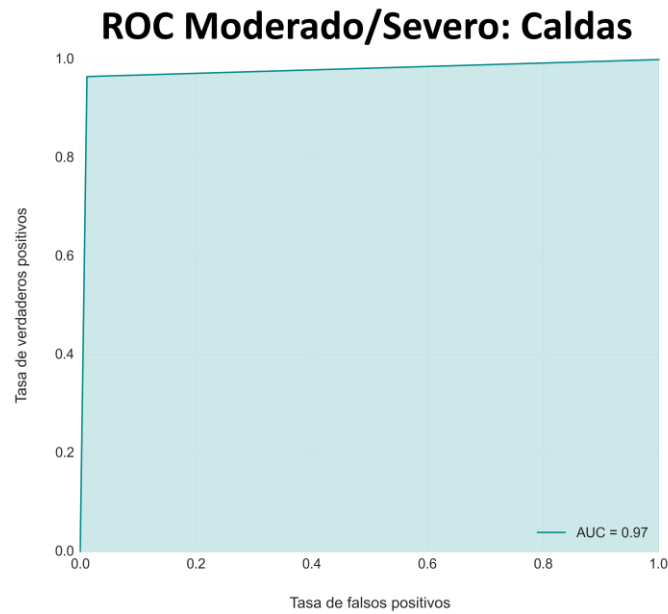


Ilustración 49 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la SVM cúbica sobre la base de datos de Caldas.

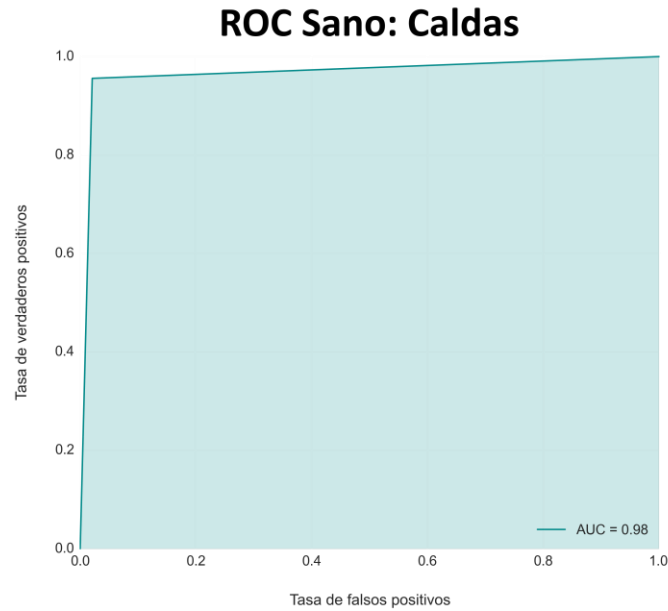


Ilustración 50 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la SVM cúbica sobre la base de datos de Caldas.

4.1.6.4 Métricas de desempeño

A continuación se muestran las métricas de desempeño calculadas a partir de las matrices de confusión y de las curvas ROC para la SVM cúbica.

ADNI		Caldas	
Exactitud	0,8803	Exactitud	0,9405
Precisión	0,8802	Precisión	0,9419
TPR (Sensibilidad)	0,8803	TPR (Sensibilidad)	0,9405
Especificidad	0,9561	Especificidad	0,9530
AUC Leve	0,8943	AUC Leve	0,9632
AUC Moderado/Severo	0,9495	AUC Moderado/Severo	0,9711
AUC Sano	0,9988	AUC Sano	0,9796
J Leve	0,7335	J Leve	0,8755
J Moderado/Severo	0,6250	J Moderado/Severo	0,8591
J Sano	0,9954	J Sano	0,9071
Kappa	0,8094	Kappa	0,8867

Tabla 8 Métricas de desempeño para la SVM cúbica sobre las bases de datos de la ADNI y Caldas.

4.1.7 Técnica 6: *Coarse Gaussian SVM*

Continuando con el enfoque basado en la teoría de los *kernels*, existe un núcleo que ha sido comúnmente usado en las SVMs aparte de los núcleos lineales y polinomiales; este se denomina el *kernel* Gaussiano. La clasificación por máquinas de soporte vectorial con *kernel* Gaussiano es sensible a la anchura del núcleo. Una anchura pequeña puede producir *overfitting*, mientras que una anchura grande puede producir *underfitting* (Chang, Chen, & Wang, 2005).

Las *Gaussian SVMs* se encuentran definidas en términos de un núcleo Gaussiano $k(x, x_i) = e^{-\lambda \|x - x_i\|^2}$ siendo $\lambda = \frac{1}{2\sigma^2}$, esto es:

$$f(x) = \sum_{i=1}^N \alpha_i y_i (e^{-\lambda \|x - x_i\|^2}) + b$$

En la ecuación anterior el coeficiente λ representa la anchura del *kernel* (Chang, Chen, & Wang, 2005).

Las *Coarse Gaussian SVMs* son aquellas que poseen una gran anchura del *kernel*. Comúnmente, esta anchura está dada por $\lambda = \sqrt{P} * 4$ donde P representa el número de variables dependientes (MathWorks, 2016). Esto es:

$$f(x) = \sum_{i=1}^N \alpha_i y_i (e^{-\sqrt{P} * 4 \|x - x_i\|^2}) + b$$

4.1.7.1 Construcción, entrenamiento y clasificación

La SVM Gaussiana de gran anchura se construye empleando la función `templateSVM('KernelFunction', 'gaussian', 'KernelScale', sqrt(P)*4,...)` perteneciente a los algoritmos de clasificación de *Statistics and Machine Learning Toolbox*TM. El entrenamiento se realiza bajo la técnica de validación cruzada con 10 iteraciones empleando las funciones `fitcecoc` y `crossval` pertenecientes a los algoritmos de regresión y clasificación de *Statistics and Machine Learning Toolbox*TM. Finalmente, se generan los

estimados para cada registro de ambas bases de datos (*ADNI* y *Caldas*) empleando la función *kfoldPredict* perteneciente a los algoritmos de clasificación de *Statistics and Machine Learning Toolbox*TM.

4.1.7.2 Matrices de confusión

En las ilustraciones 51 y 52 se muestran las matrices de confusión obtenidas para la SVM Gaussiana de gran anchura sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se deduce que el clasificador logró un porcentaje de acierto del 0.90% y 0.93% en las respectivas predicciones. Asimismo, se puede observar que existe un bajo grado de confusión en las estimaciones de la técnica de aprendizaje de máquina respecto a las clases de predicción.

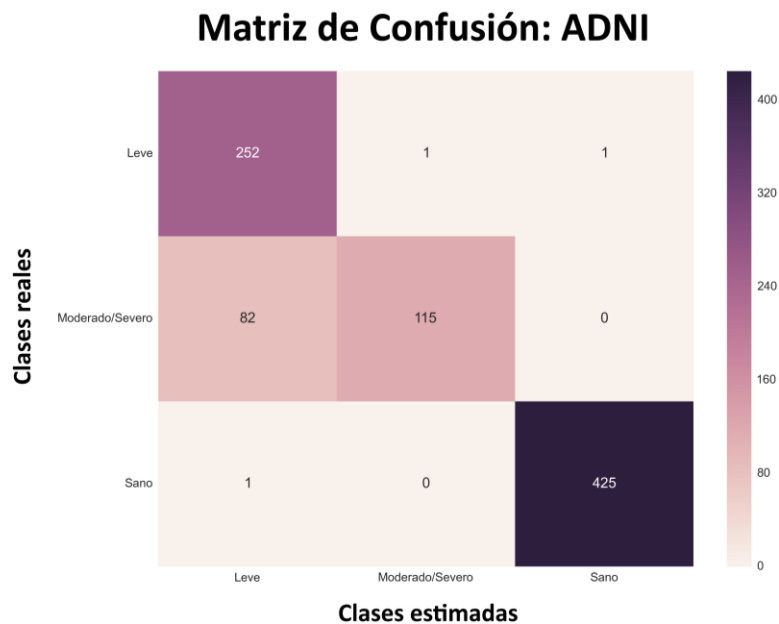


Ilustración 51 Matriz de confusión referente a los estimados de la SVM Gaussiana de gran anchura sobre la base de datos de la ADNI.

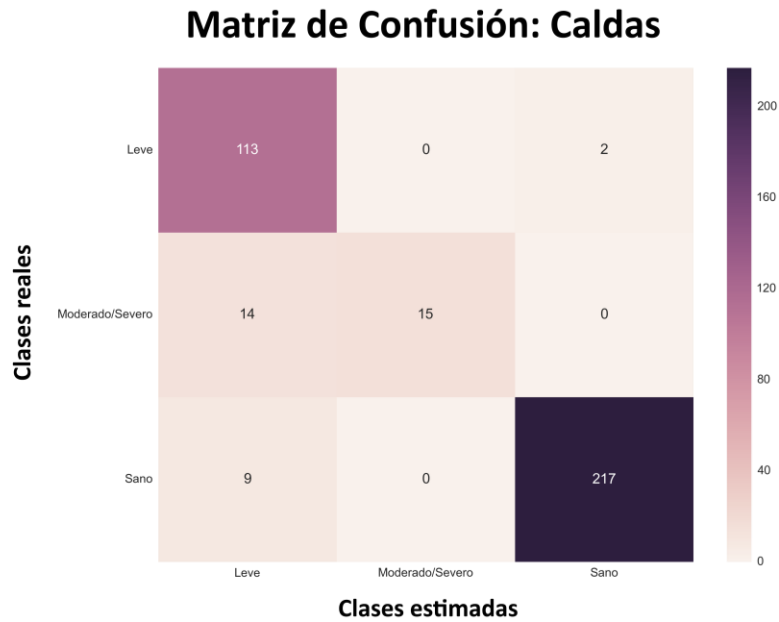


Ilustración 52 Matriz de confusión referente a los estimados de la SVM Gaussiana de gran anchura sobre la base de datos de Caldas.

4.1.7.3 Curvas ROC

En las ilustraciones 53 a 58 se muestran las curvas ROC obtenidas para la SVM Gaussiana de gran anchura sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se observa que el AUC para cada una de las clases se encuentra bastante cercana de 1, lo cual indica que la técnica de aprendizaje de máquina no presenta mayores dificultades para elegir la clase correcta al momento de realizar las predicciones. Solamente se presentan algunos inconvenientes menores para diferenciar las clases «Leve» y «Moderado/Severo» de las demás en la base de datos de la *ADNI*.

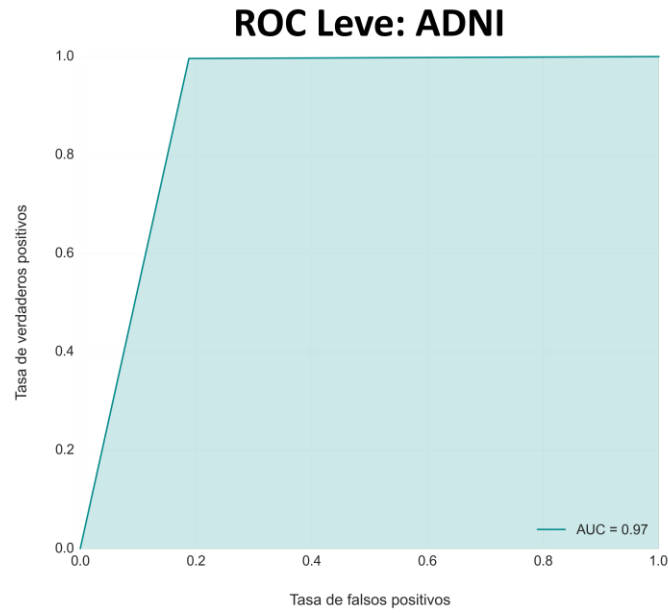


Ilustración 53 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la SVM Gaussiana de gran anchura sobre la base de datos de la ADNI.

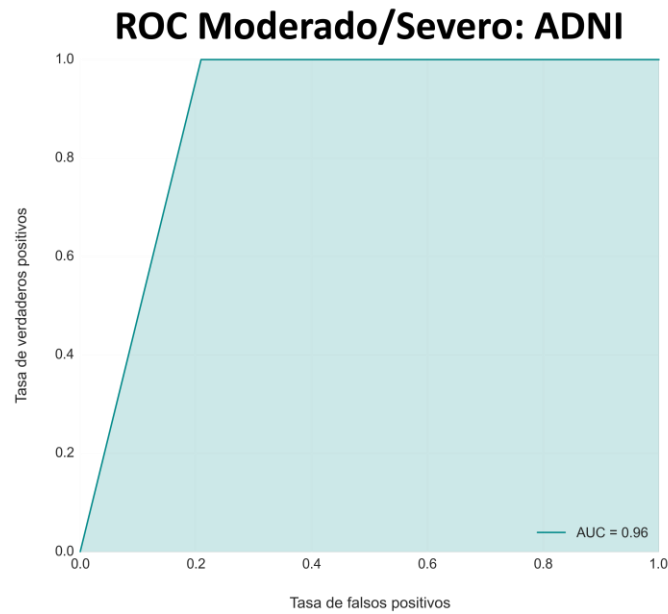


Ilustración 54 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la SVM Gaussiana de gran anchura sobre la base de datos de la ADNI.

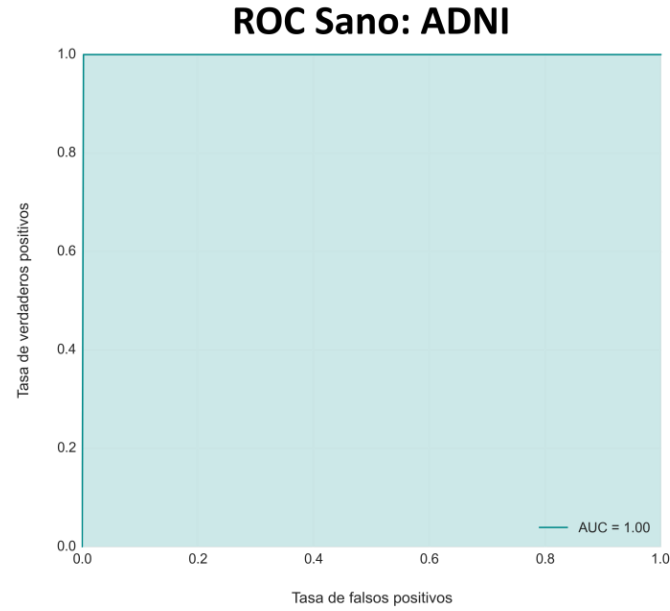


Ilustración 55 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo— referente a los estimados de la SVM Gaussiana de gran anchura sobre la base de datos de la ADNI.

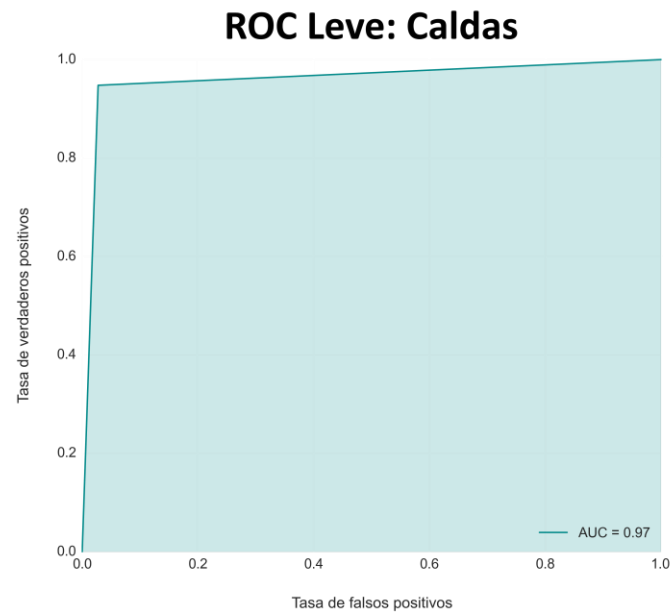


Ilustración 56 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la SVM Gaussiana de gran anchura sobre la base de datos de Caldas.

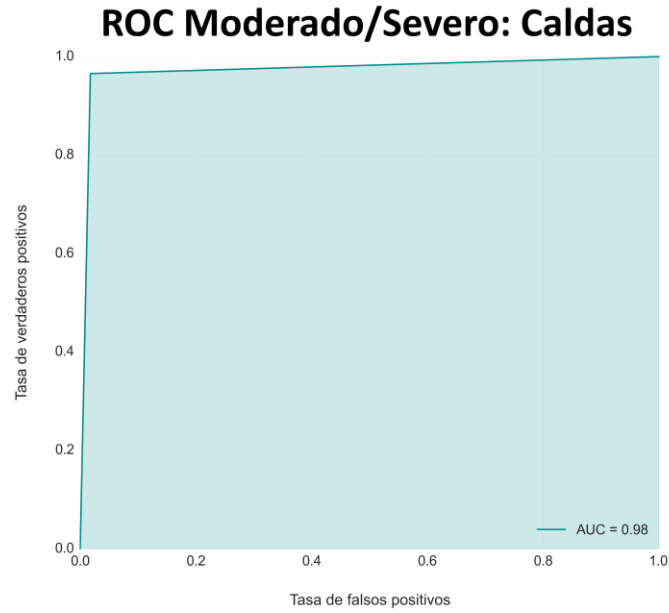


Ilustración 57 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la SVM Gaussiana de gran anchura sobre la base de datos de Caldas.

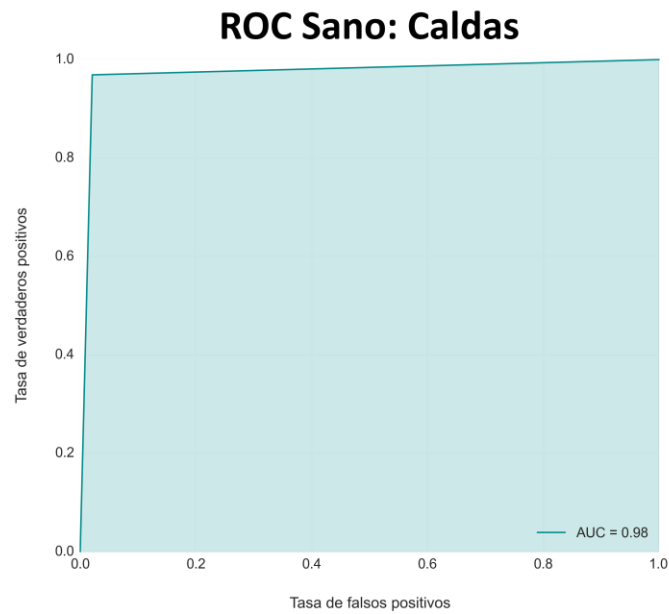


Ilustración 58 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo— referente a los estimados de la SVM Gaussiana de gran anchura sobre la base de datos de Caldas.

4.1.7.4 Métricas de desempeño

A continuación se muestran las métricas de desempeño calculadas a partir de las matrices de confusión y de las curvas ROC para la SVM Gaussiana de gran anchura.

ADNI		Caldas	
Exactitud	0,9031	Exactitud	0,9324
Precisión	0,9252	Precisión	0,9419
TPR (Sensibilidad)	0,9031	TPR (Sensibilidad)	0,9324
Especificidad	0,9600	Especificidad	0,9635
AUC Leve	0,9669	AUC Leve	0,9739
AUC Moderado/Severo	0,9639	AUC Moderado/Severo	0,9813
AUC Sano	0,9987	AUC Sano	0,9799
J Leve	0,8589	J Leve	0,8924
J Moderado/Severo	0,5823	J Moderado/Severo	0,5172
J Sano	0,9954	J Sano	0,9463
Kappa	0,8446	Kappa	0,8703

Tabla 9 Métricas de desempeño para la SVM Gaussiana de gran anchura sobre las bases de datos de la ADNI y Caldas.

4.1.8 Técnica 7: Medium Gaussian SVM

De acuerdo con la sección previa, las *Medium Gaussian SVMs* son aquellas que poseen una mediana anchura del *kernel*. Comúnmente, esta anchura está dada por $\lambda = \sqrt{P}$ donde P representa el número de variables dependientes (MathWorks, 2016). Esto es:

$$f(x) = \sum_{i=1}^N \alpha_i y_i \left(e^{-\sqrt{P} \|x - x_i\|^2} \right) + b$$

4.1.8.1 Construcción, entrenamiento y clasificación

La SVM Gaussiana de mediana anchura se construye empleando la función `templateSVM('KernelFunction', 'gaussian', 'KernelScale', sqrt(P),...)` perteneciente a los algoritmos de clasificación de *Statistics and Machine Learning Toolbox™*. El

entrenamiento se realiza bajo la técnica de validación cruzada con 10 iteraciones empleando las funciones *fitcecoc* y *crossval* pertenecientes a los algoritmos de regresión y clasificación de *Statistics and Machine Learning Toolbox*™. Finalmente, se generan los estimados para cada registro de ambas bases de datos (*ADNI* y *Caldas*) empleando la función *kfoldPredict* perteneciente a los algoritmos de clasificación de *Statistics and Machine Learning Toolbox*™.

4.1.8.2 Matrices de confusión

En las ilustraciones 59 y 60 se muestran las matrices de confusión obtenidas para la SVM Gaussiana de mediana anchura sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se deduce que el clasificador logró un porcentaje de acierto del 0.89% y 0.94% en las respectivas predicciones. Asimismo, se puede observar que existe un bajo grado de confusión en las estimaciones de la técnica de aprendizaje de máquina respecto a las clases de predicción.

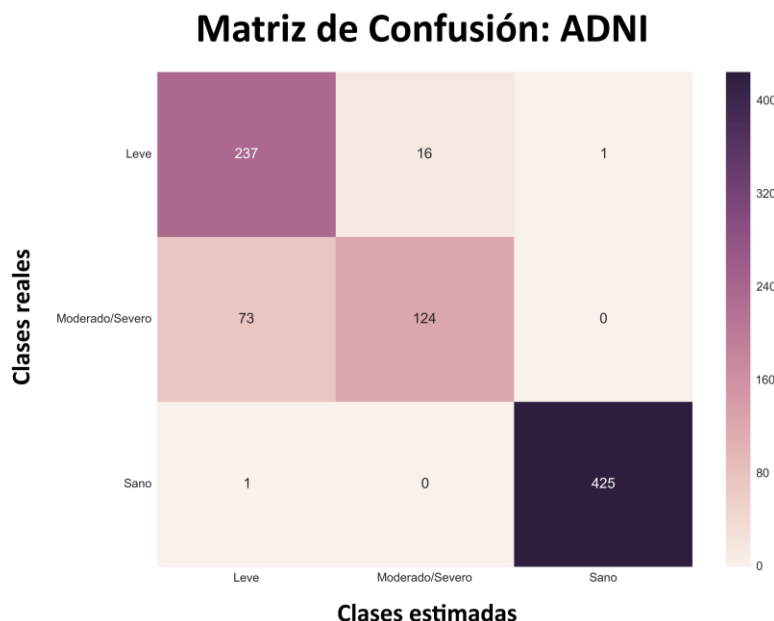


Ilustración 59 Matriz de confusión referente a los estimados de la SVM Gaussiana de mediana anchura sobre la base de datos de la ADNI.

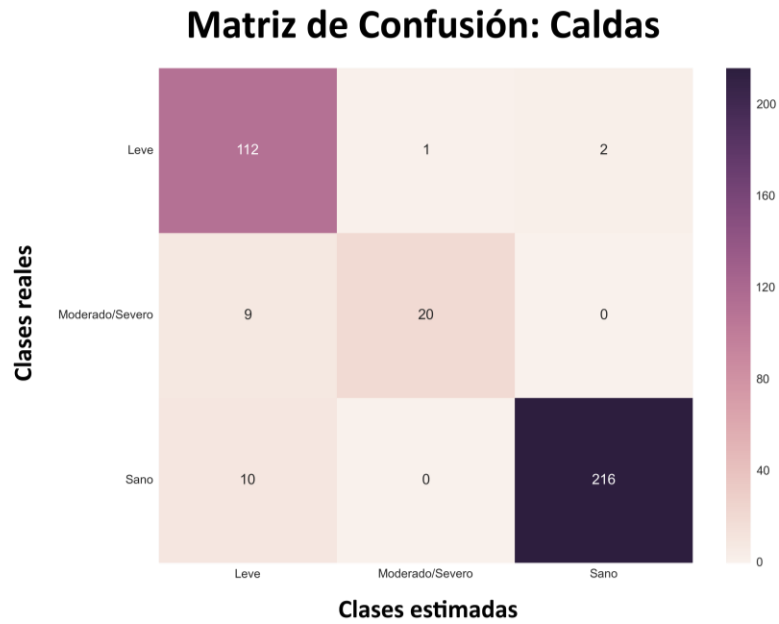


Ilustración 60 Matriz de confusión referente a los estimados de la SVM Gaussiana de mediana anchura sobre la base de datos de Caldas.

4.1.8.3 Curvas ROC

En las ilustraciones 61 a 66 se muestran las curvas ROC obtenidas para la SVM Gaussiana de mediana anchura sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se observa que el AUC para cada una de las clases se encuentra bastante cercana de 1, lo cual indica que la técnica de aprendizaje de máquina no presenta mayores dificultades para elegir la clase correcta al momento de realizar las predicciones. Solamente se presentan algunos inconvenientes menores para diferenciar las clases «Leve» y «Moderado/Severo» de las demás en la base de datos de la *ADNI*.

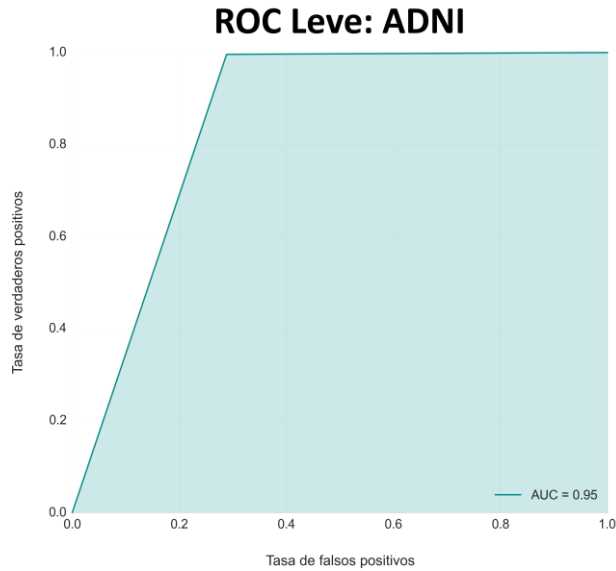


Ilustración 61 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la SVM Gaussiana de mediana anchura sobre la base de datos de la ADNI.

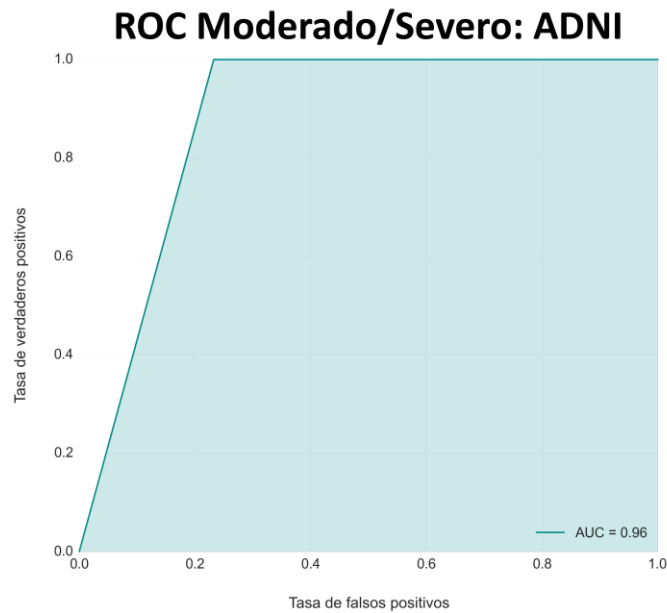


Ilustración 62 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la SVM Gaussiana de mediana anchura sobre la base de datos de la ADNI.

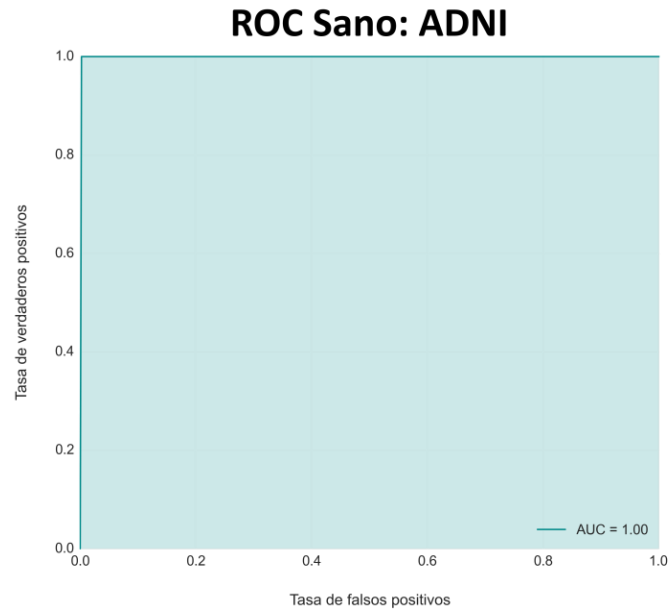


Ilustración 63 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo— referente a los estimados de la SVM Gaussiana de mediana anchura sobre la base de datos de la ADNI.

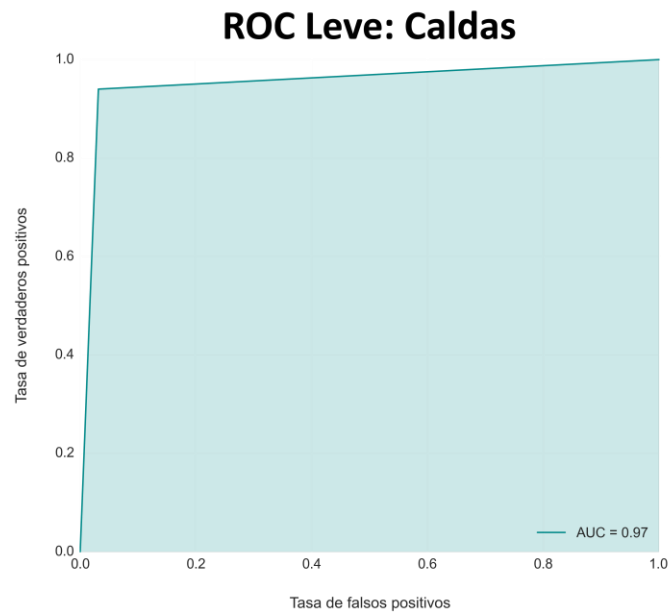


Ilustración 64 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la SVM Gaussiana de mediana anchura sobre la base de datos de Caldas.

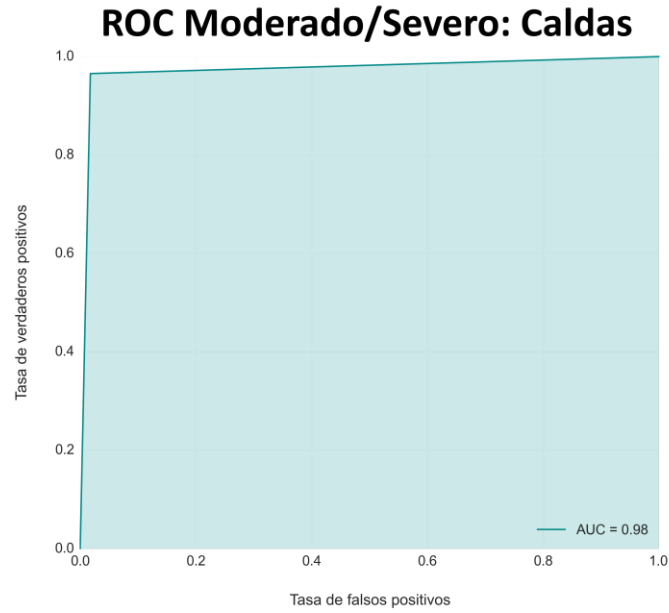


Ilustración 65 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la SVM Gaussiana de mediana anchura sobre la base de datos de Caldas.

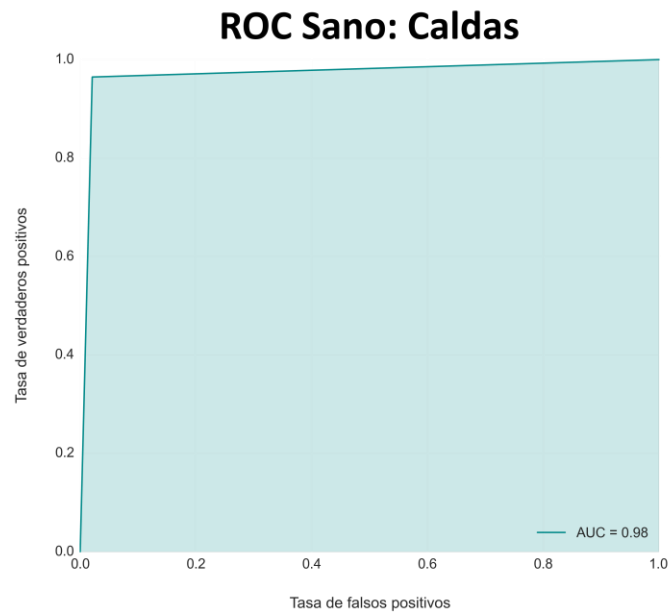


Ilustración 66 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo— referente a los estimados de la SVM Gaussiana de mediana anchura sobre la base de datos de Caldas.

4.1.8.4 Métricas de desempeño

A continuación se muestran las métricas de desempeño calculadas a partir de las matrices de confusión y de las curvas ROC para la SVM Gaussiana de mediana anchura.

ADNI		Caldas	
Exactitud	0,8962	Exactitud	0,9405
Precisión	0,9043	Precisión	0,9456
TPR (Sensibilidad)	0,8962	TPR (Sensibilidad)	0,9405
Especificidad	0,9592	Especificidad	0,9681
AUC Leve	0,9515	AUC Leve	0,9734
AUC Moderado/Severo	0,9581	AUC Moderado/Severo	0,9774
AUC Sano	0,9988	AUC Sano	0,9778
J Leve	0,8143	J Leve	0,8994
J Moderado/Severo	0,6059	J Moderado/Severo	0,6867
J Sano	0,9954	J Sano	0,9419
Kappa	0,8341	Kappa	0,8869

Tabla 10 Métricas de desempeño para la SVM Gaussiana de mediana anchura sobre las bases de datos de la ADNI y Caldas.

4.1.9 Técnica 8: *Fine Gaussian SVM*

De acuerdo con la sección previa, las *Fine Gaussian SVMs* son aquellas que poseen una pequeña anchura del *kernel*. Comúnmente, esta anchura está dada por $\lambda = \frac{\sqrt{P}}{4}$ donde P representa el número de variables dependientes (MathWorks, 2016). Esto es:

$$f(x) = \sum_{i=1}^N \alpha_i y_i \left(e^{-\frac{\sqrt{P}}{4} \|x - x_i\|^2} \right) + b$$

4.1.9.1 Construcción, entrenamiento y clasificación

La SVM Gaussiana de pequeña anchura se construye empleando la función `templateSVM('KernelFunction', 'gaussian', 'KernelScale', sqrt(P)/4,...)` perteneciente a los algoritmos de clasificación de *Statistics and Machine Learning Toolbox™*. El

entrenamiento se realiza bajo la técnica de validación cruzada con 10 iteraciones empleando las funciones *fitcecoc* y *crossval* pertenecientes a los algoritmos de regresión y clasificación de *Statistics and Machine Learning Toolbox*TM. Finalmente, se generan los estimados para cada registro de ambas bases de datos (*ADNI* y *Caldas*) empleando la función *kfoldPredict* perteneciente a los algoritmos de clasificación de *Statistics and Machine Learning Toolbox*TM.

4.1.9.2 Matrices de confusión

En las ilustraciones 67 y 68 se muestran las matrices de confusión obtenidas para la SVM Gaussiana de pequeña anchura sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se deduce que el clasificador logró un porcentaje de acierto del 0.81% y 0.71% en las respectivas predicciones. No obstante, se puede observar que existe un alto grado de confusión en las estimaciones de la técnica de aprendizaje de máquina respecto a las tres clases de predicción.

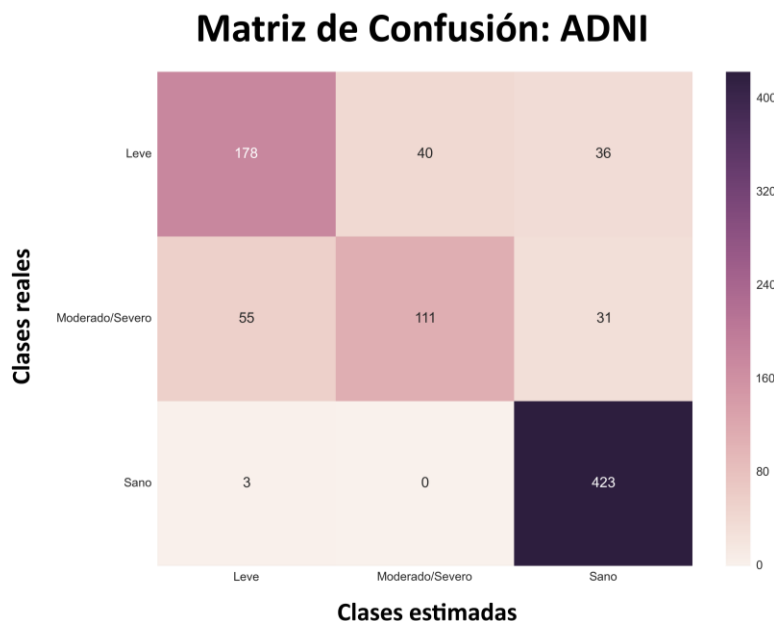


Ilustración 67 Matriz de confusión referente a los estimados de la SVM Gaussiana de pequeña anchura sobre la base de datos de la ADNI.

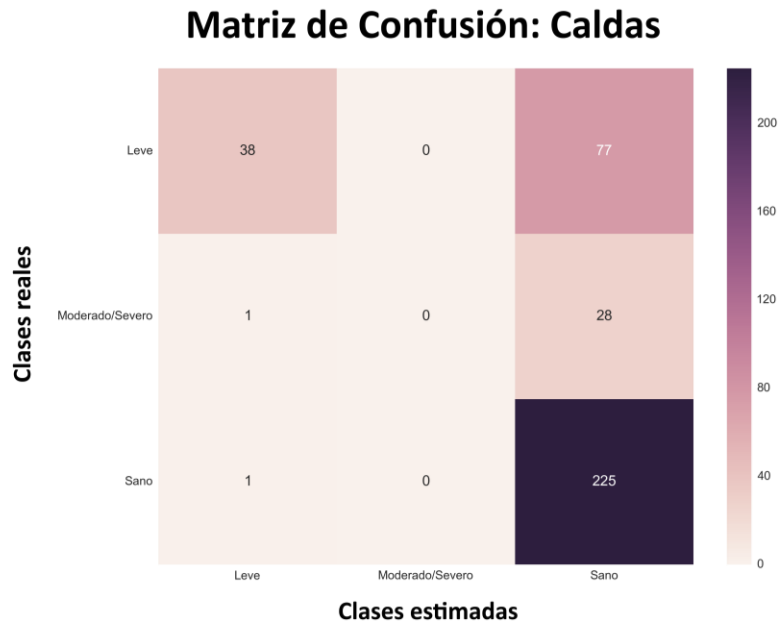


Ilustración 68 Matriz de confusión referente a los estimados de la SVM Gaussiana de pequeña anchura sobre la base de datos de Caldas.

4.1.9.3 Curvas ROC

En las ilustraciones 69 a 74 se muestran las curvas ROC obtenidas para la SVM Gaussiana de pequeña anchura sobre las bases de datos de la ADNI y Caldas. A partir de estas se observa que las AUC para las clases «Leve» y «Moderado/Severo» se encuentran medianamente alejadas de 1, lo cual indica que la técnica de aprendizaje de máquina presenta algunas dificultades para elegir la clase correcta entre las dos mencionadas, al momento de realizar las predicciones. No obstante, el AUC para la clase «Sano» se encuentra bastante cercana a 1, es decir que la técnica es capaz de diferenciar dicha clase de las demás con suma fidelidad.

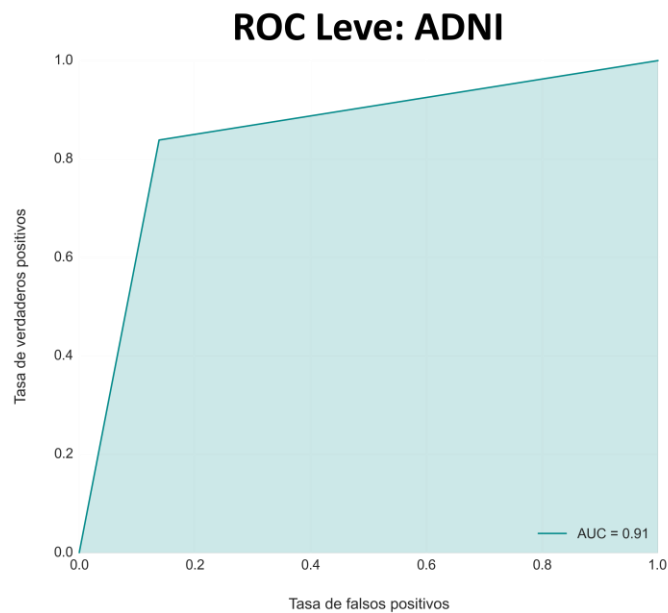


Ilustración 69 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la SVM Gaussiana de pequeña anchura sobre la base de datos de la ADNI.

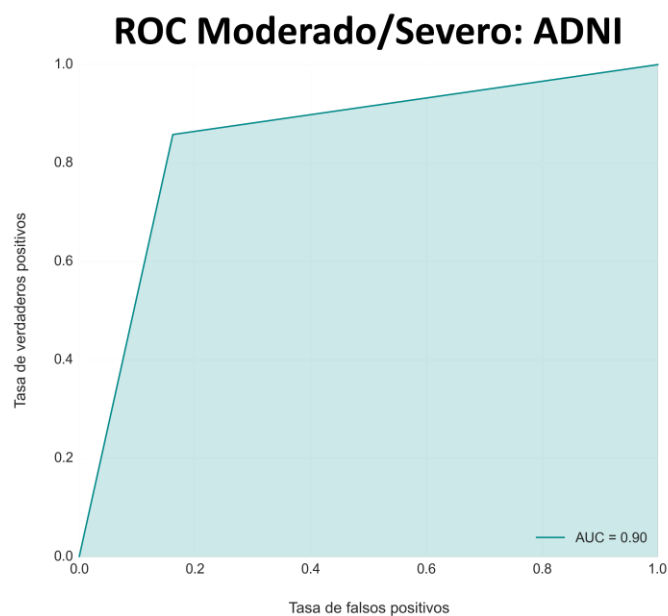


Ilustración 70 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la SVM Gaussiana de pequeña anchura sobre la base de datos de la ADNI.

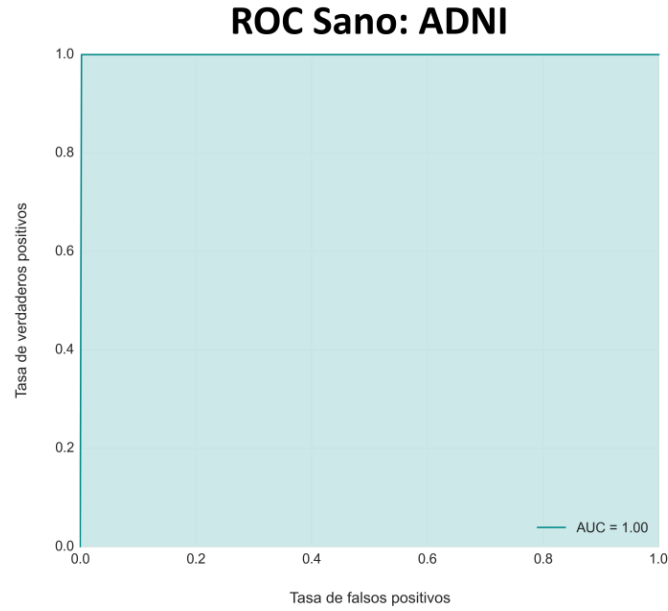


Ilustración 71 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo— referente a los estimados de la SVM Gaussiana de pequeña anchura sobre la base de datos de la ADNI.

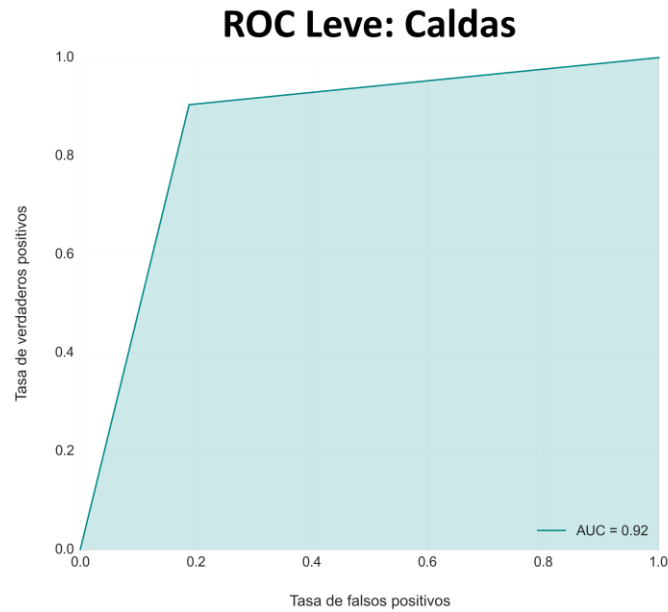


Ilustración 72 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la SVM Gaussiana de pequeña anchura sobre la base de datos de Caldas.

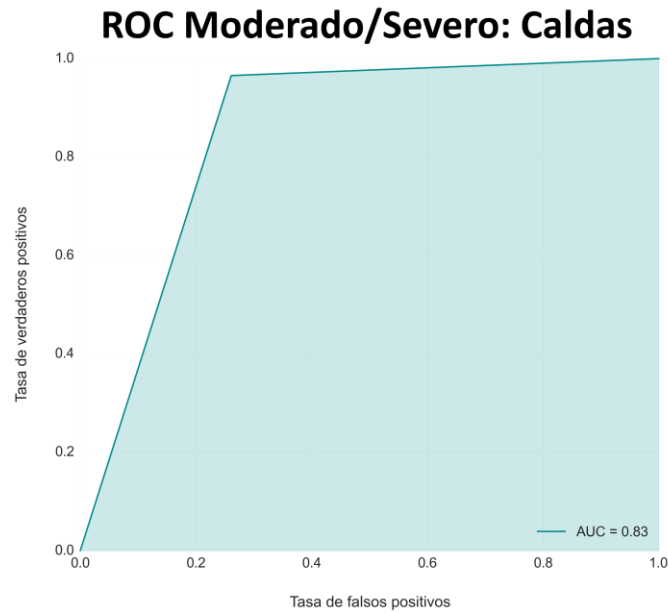


Ilustración 73 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la SVM Gaussiana de pequeña anchura sobre la base de datos de Caldas.

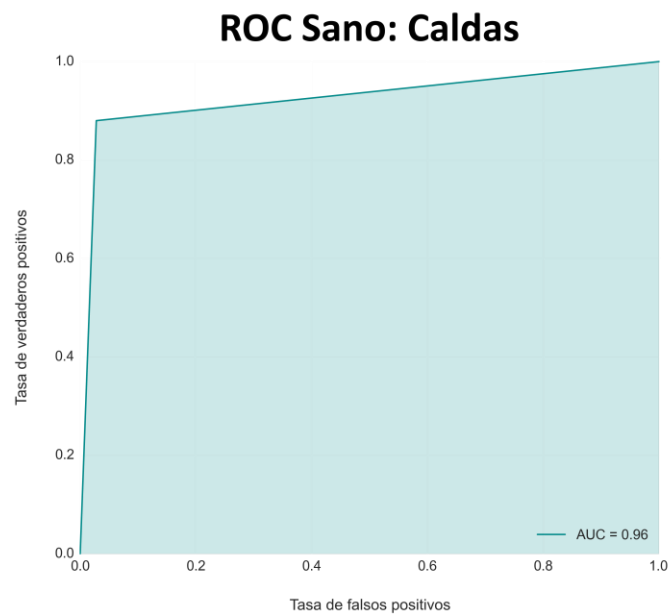


Ilustración 74 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la SVM Gaussiana de pequeña anchura sobre la base de datos de Caldas.

4.1.9.4 Métricas de desempeño

A continuación se muestran las métricas de desempeño calculadas a partir de las matrices de confusión y de las curvas ROC para la SVM Gaussiana de pequeña anchura.

ADNI		Caldas	
Exactitud	0,8119	Exactitud	0,7108
Precisión	0,8029	Precisión	0,7901
TPR (Sensibilidad)	0,8119	TPR (Sensibilidad)	0,7108
Especificidad	0,8877	Especificidad	0,5522
AUC Leve	0,9083	AUC Leve	0,9209
AUC Moderado/Severo	0,9036	AUC Moderado/Severo	0,8296
AUC Sano	0,9959	AUC Sano	0,9583
J Leve	0,6077	J Leve	0,3226
J Moderado/Severo	0,5046	J Moderado/Severo	0,0000
J Sano	0,8444	J Sano	0,2664
Kappa	0,6926	Kappa	0,3141

Tabla 11 Métricas de desempeño para la SVM Gaussiana de pequeña anchura sobre las bases de datos de la ADNI y Caldas.

4.1.10 Técnica 9: *Decision Tree*

Los árboles de decisión son técnicas no parametrizadas de aprendizaje de máquina que se usan tanto para tareas de clasificación como de regresión. La meta es crear un modelo que predice el valor de una variable objetivo mediante el aprendizaje de reglas de decisión simples inferidas a partir de las características de los datos (Scikit-learn, 2015).

Un árbol de decisión representa una estructura jerárquica formada por nodos y aristas dirigidas. Estos árboles poseen tres tipos de nodos: un nodo raíz que no tiene aristas entrantes y cero o más aristas salientes; varios nodos internos, cada uno de los cuales tiene exactamente una arista entrante y dos o más aristas salientes; y varios nodos «hoja» o terminales, cada uno de los cuales tiene exactamente una arista entrante y cero aristas salientes (Tan, Steinbach, & Kumar, 2006).

En un árbol de decisión, cada nodo terminal se asocia a una etiqueta de clase. Los nodos no terminales —que incluyen la raíz y otros nodos internos— contienen las condiciones de prueba de atributos para separar los registros que poseen características diferentes (Tan, Steinbach, & Kumar, 2006).

Las medidas concebidas para seleccionar la mejor separación de los registros en los nodos no terminales se basan usualmente en el grado de impureza de los nodos descendientes. Cuanto menos sea el grado de impureza, más sesgada será la distribución de las clases (Tan, Steinbach, & Kumar, 2006). Las dos medidas de impureza más comunes son la entropía y el índice Gini (Kingsford & Salzberg, 2008):

$$Entropía(t) = - \sum_{i=0}^{c-1} p(i|t) \log_2 p(i|t)$$
$$Gini(t) = 1 - \sum_{i=0}^{c-1} [p(i|t)]^2$$

Donde c indica el número de clases existentes y $p(i|t)$ denota la fracción de los registros pertenecientes a la clase i en un nodo t específico.

4.1.10.1 Construcción, entrenamiento y clasificación

El árbol de decisión se construye empleando la función `fitctree('SplitCriterion', 'gdi',...)` perteneciente a los algoritmos de clasificación de *Statistics and Machine Learning Toolbox*TM. El entrenamiento se realiza bajo la técnica de validación cruzada con 10 iteraciones empleando las funciones `fitctree` y `crossval` pertenecientes a los algoritmos de clasificación de *Statistics and Machine Learning Toolbox*TM. Finalmente, se generan los estimados para cada registro de ambas bases de datos (*ADNI* y *Caldas*) empleando la función `kfoldPredict` perteneciente a los algoritmos de clasificación de *Statistics and Machine Learning Toolbox*TM.

4.1.10.2 Matrices de confusión

En las ilustraciones 75 y 76 se muestran las matrices de confusión obtenidas para el árbol de decisión sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se deduce que el clasificador logró un porcentaje de acierto del 0.90% y 0.96% en las respectivas predicciones. Asimismo, se puede observar que existe un bajo grado de confusión en las estimaciones de la técnica de aprendizaje de máquina respecto a las clases de predicción.

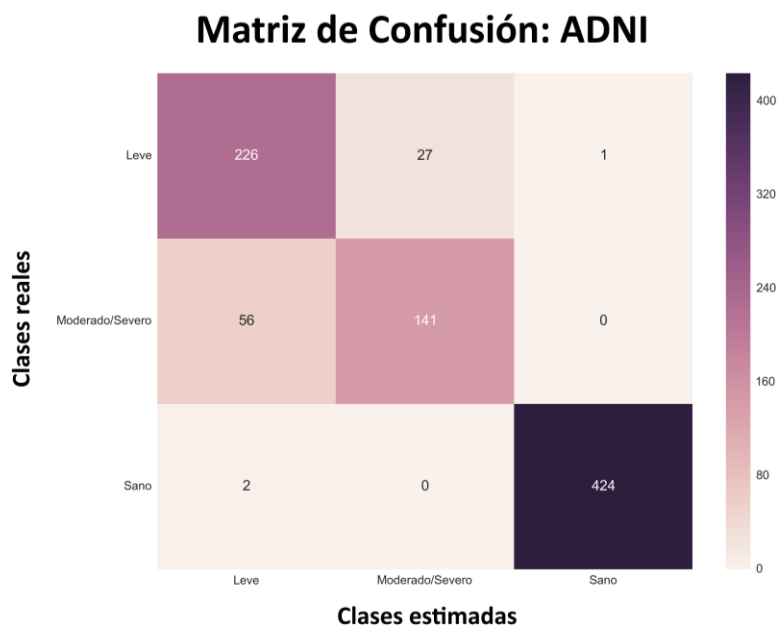


Ilustración 75 Matriz de confusión referente a los estimados del árbol de decisión sobre la base de datos de la ADNI.

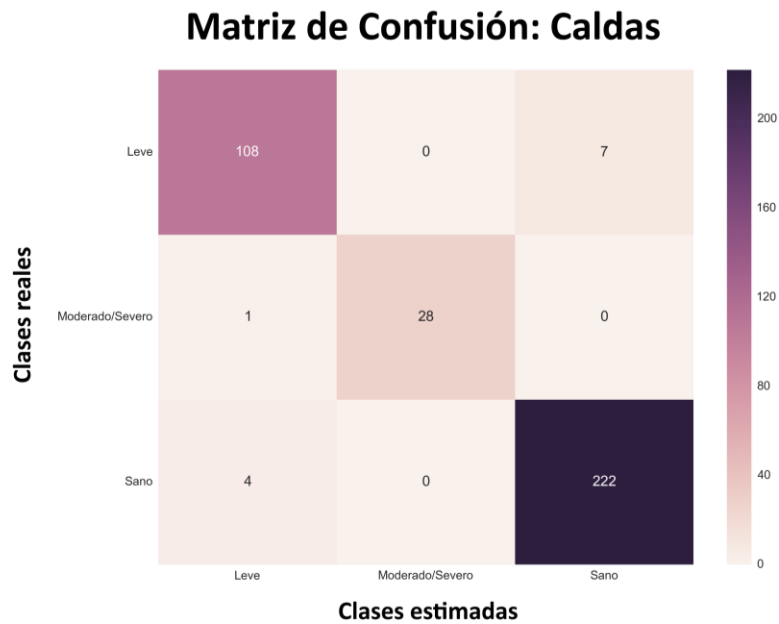


Ilustración 76 Matriz de confusión referente a los estimados del árbol de decisión sobre la base de datos de Caldas.

4.1.10.3 Curvas ROC

En las ilustraciones 77 a 82 se muestran las curvas ROC obtenidas para el árbol de decisión sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se observa que el AUC para cada una de las clases se encuentra bastante cercana de 1, lo cual indica que la técnica de aprendizaje de máquina no presenta mayores dificultades para elegir la clase correcta al momento de realizar las predicciones. Solamente se presentan algunos inconvenientes menores para diferenciar las clases «Leve» y «Moderado/Severo» de las demás en la base de datos de la *ADNI*.

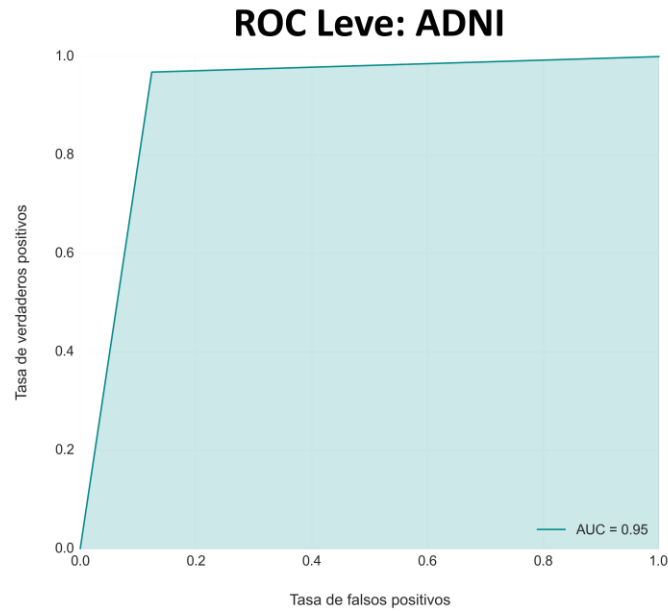


Ilustración 77 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del árbol de decisión sobre la base de datos de la ADNI.

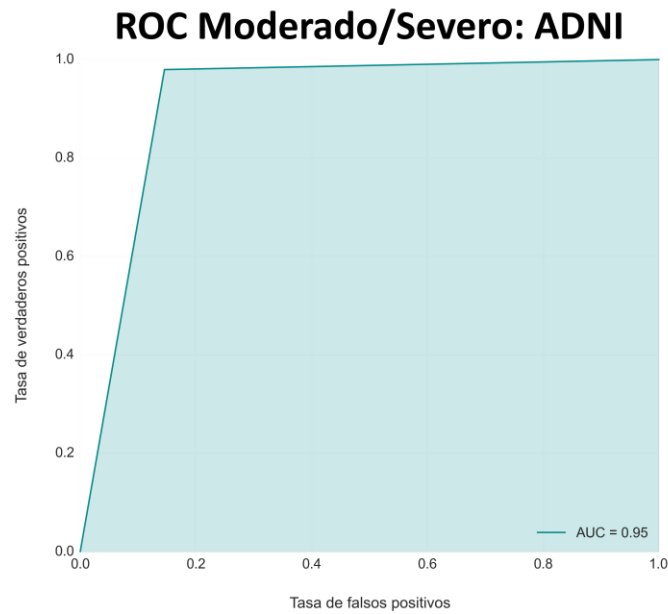


Ilustración 78 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del árbol de decisión sobre la base de datos de la ADNI.

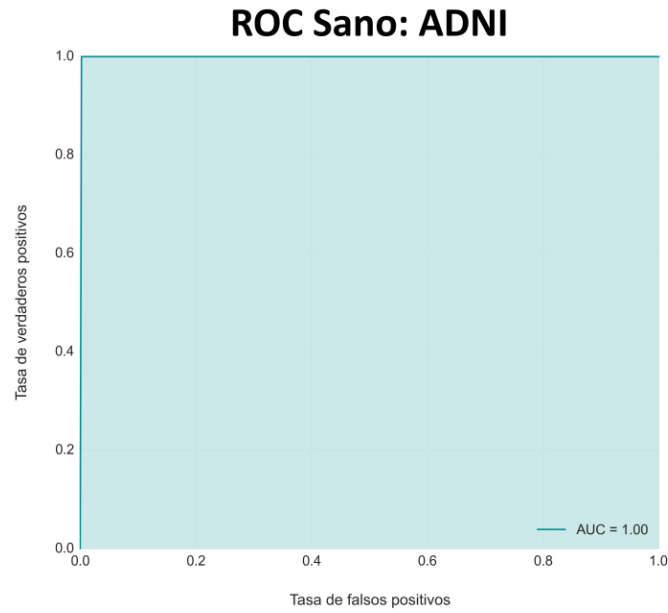


Ilustración 79 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo— referente a los estimados del árbol de decisión sobre la base de datos de la ADNI.

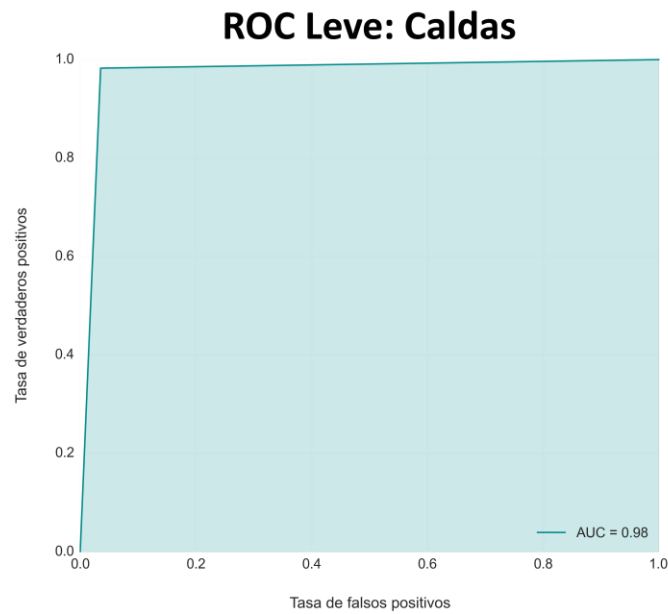


Ilustración 80 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del árbol de decisión sobre la base de datos de Caldas.

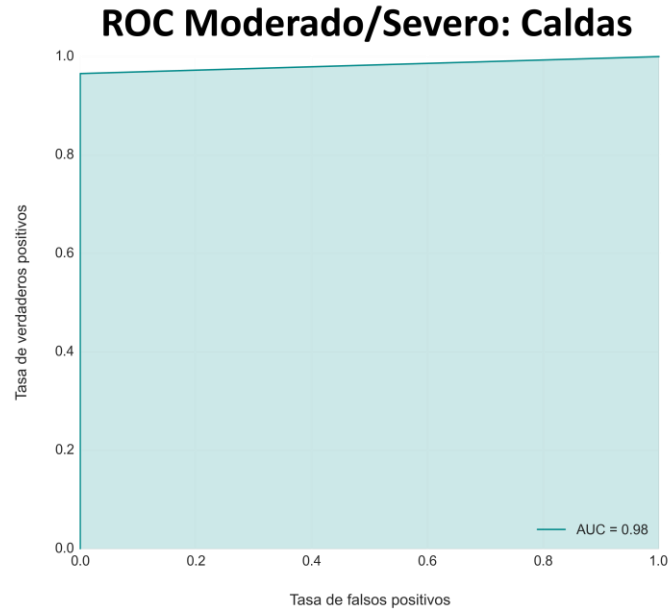


Ilustración 81 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del árbol de decisión sobre la base de datos de Caldas.

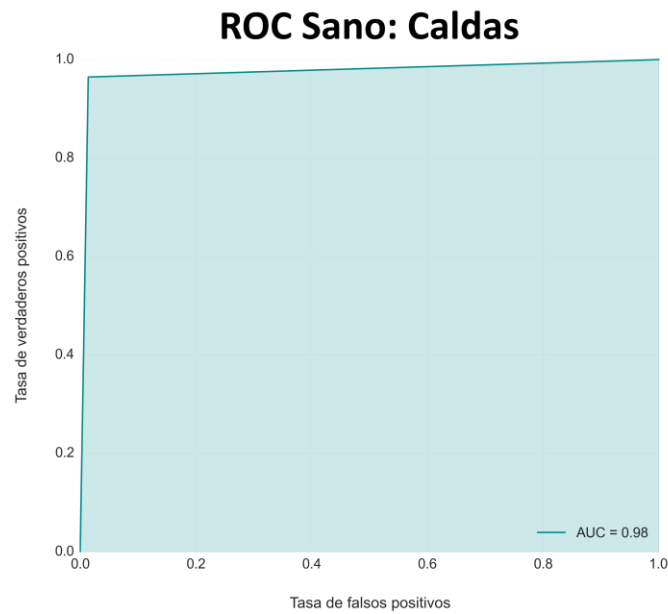


Ilustración 82 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del árbol de decisión sobre la base de datos de Caldas.

4.1.10.4 Métricas de desempeño

A continuación se muestran las métricas de desempeño calculadas a partir de las matrices de confusión y de las curvas ROC para el árbol de decisión.

ADNI		Caldas	
Exactitud	0,9019	Exactitud	0,9676
Precisión	0,9036	Precisión	0,9676
TPR (Sensibilidad)	0,9019	TPR (Sensibilidad)	0,9676
Especificidad	0,9630	Especificidad	0,9642
AUC Leve	0,9525	AUC Leve	0,9816
AUC Moderado/Severo	0,9540	AUC Moderado/Severo	0,9779
AUC Sano	0,9965	AUC Sano	0,9850
J Leve	0,7967	J Leve	0,9195
J Moderado/Severo	0,6760	J Moderado/Severo	0,9655
J Sano	0,9931	J Sano	0,9337
Kappa	0,8438	Kappa	0,9378

Tabla 12 Métricas de desempeño para el árbol de decisión sobre las bases de datos de la ADNI y Caldas.

4.1.11 Técnica 10: *Decision Trees (Bagging)*

El *bagging* o *bootstrap aggregating* es una meta-técnica de ensamble que ajusta diferentes clasificadores base —cada uno en subconjuntos aleatorios del conjunto de datos original— y posteriormente agrega sus predicciones individuales, ya sea por votación o por promedio, para formar una predicción final. Dicha meta-técnica es frecuentemente usada como una vía para reducir la varianza de un estimador de caja negra (ej. un árbol de decisión), mediante la inclusión de la aleatoriedad en su proceso de construcción y después la realización de un ensamble a partir de este (Scikit-learn, 2015).

El procedimiento para realizar el *bagging* de árboles de decisión se concreta en cuatro pasos:

1. Extraer una muestra aleatoria del conjunto de datos original.
2. Construir un árbol de decisión a partir de la muestra.
3. Repetir los pasos 1 y 2 múltiples veces, de forma independiente.

4. Calcular un clasificador agregado usando los árboles de decisión formados en los pasos 1 a 3. Para determinar el resultado de predicción, se debe tomar el voto mayoritario de las predicciones de cada árbol de decisión (Breiman, 1994).

4.1.11.1 Construcción, entrenamiento y clasificación

El *bagging* de árboles de decisión se construye empleando la función *fitensemble('Bag', 30, 'Tree', 'Type', 'Classification',...)* perteneciente a los algoritmos de regresión de *Statistics and Machine Learning Toolbox*TM. El entrenamiento se realiza bajo la técnica de validación cruzada con 10 iteraciones empleando las funciones *fitensemble* y *crossval* pertenecientes a los algoritmos de regresión y clasificación de *Statistics and Machine Learning Toolbox*TM. Finalmente, se generan los estimados para cada registro de ambas bases de datos (*ADNI* y *Caldas*) empleando la función *kfoldPredict* perteneciente a los algoritmos de clasificación de *Statistics and Machine Learning Toolbox*TM.

4.1.11.2 Matrices de confusión

En las ilustraciones 83 y 84 se muestran las matrices de confusión obtenidas para el *bagging* de árboles de decisión sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se deduce que el clasificador logró un porcentaje de acierto del 0.90% y 0.97% en las respectivas predicciones. Asimismo, se puede observar que existe un bajo grado de confusión en las estimaciones de la técnica de aprendizaje de máquina respecto a las clases de predicción.

Matriz de Confusión: ADNI

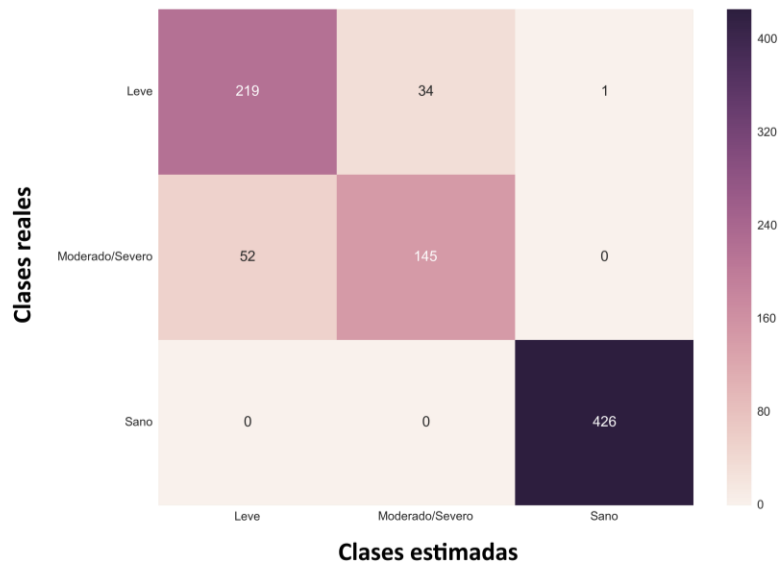


Ilustración 83 Matriz de confusión referente a los estimados del bagging de árboles de decisión sobre la base de datos de la ADNI.

Matriz de Confusión: Caldas

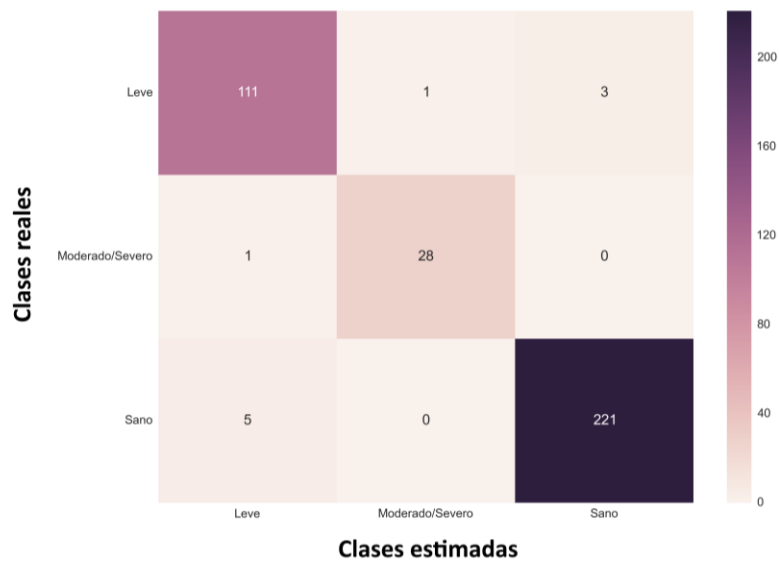


Ilustración 84 Matriz de confusión referente a los estimados del bagging de árboles de decisión sobre la base de datos de Caldas.

4.1.11.3 Curvas ROC

En las ilustraciones 85 a 90 se muestran las curvas ROC obtenidas para el *bagging* de árboles de decisión sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se observa que el AUC para cada una de las clases se encuentra bastante cercana de 1, lo cual indica que la técnica de aprendizaje de máquina no presenta mayores dificultades para elegir la clase correcta al momento de realizar las predicciones. Solamente se presentan algunos inconvenientes menores para diferenciar las clases «Leve» y «Moderado/Severo» de las demás en la base de datos de la ADNI.

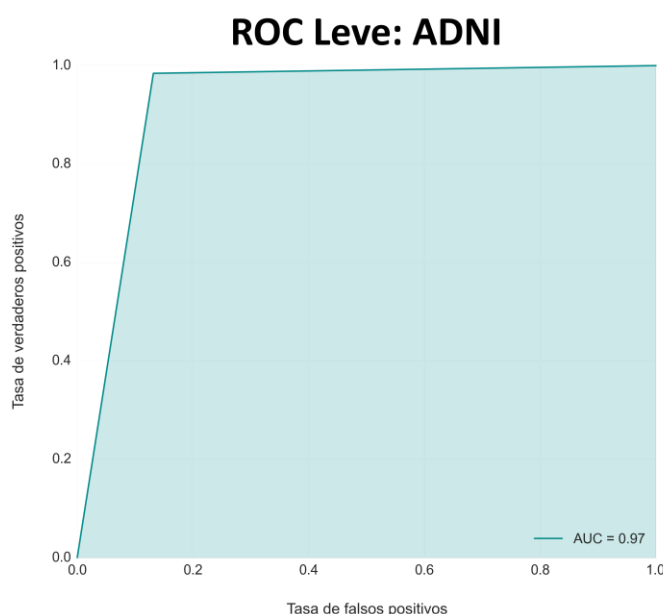


Ilustración 85 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del bagging de árboles de decisión sobre la base de datos de la ADNI.

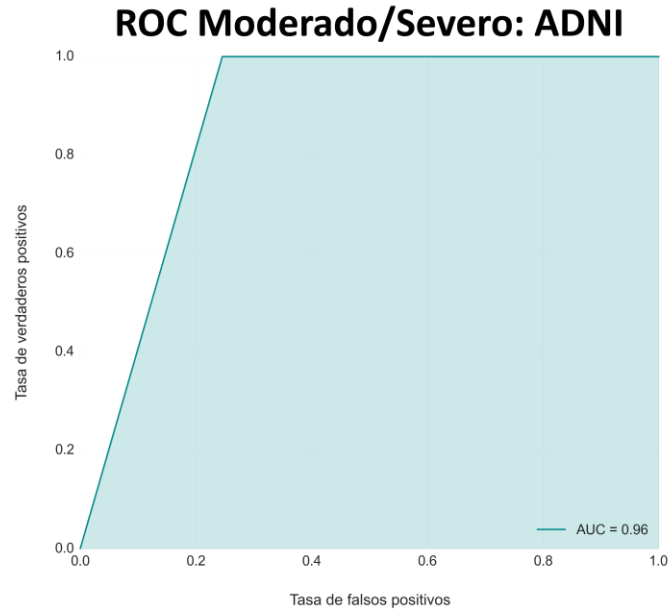


Ilustración 86 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del bagging de árboles de decisión sobre la base de datos de la ADNI.

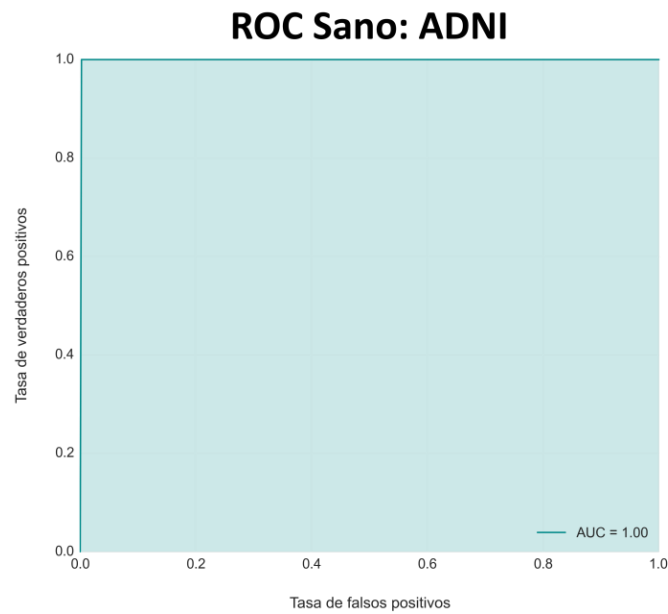


Ilustración 87 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del bagging de árboles de decisión sobre la base de datos de la ADNI.

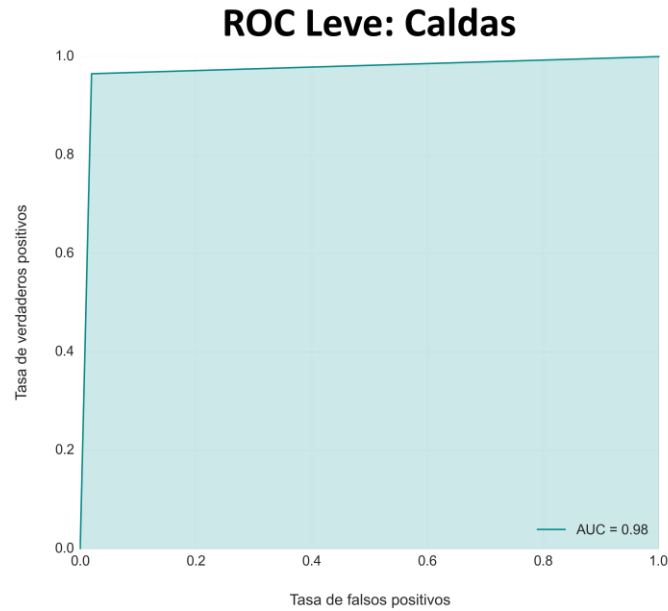


Ilustración 88 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del bagging de árboles de decisión sobre la base de datos de Caldas.

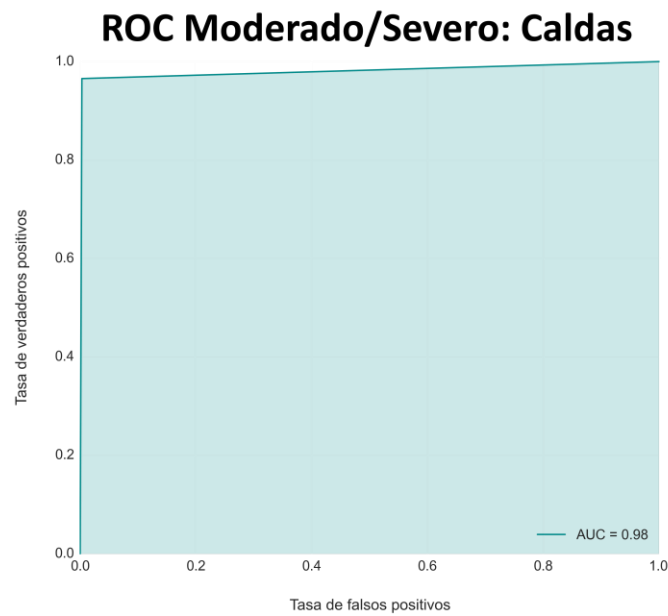


Ilustración 89 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del bagging de árboles de decisión sobre la base de datos de Caldas.

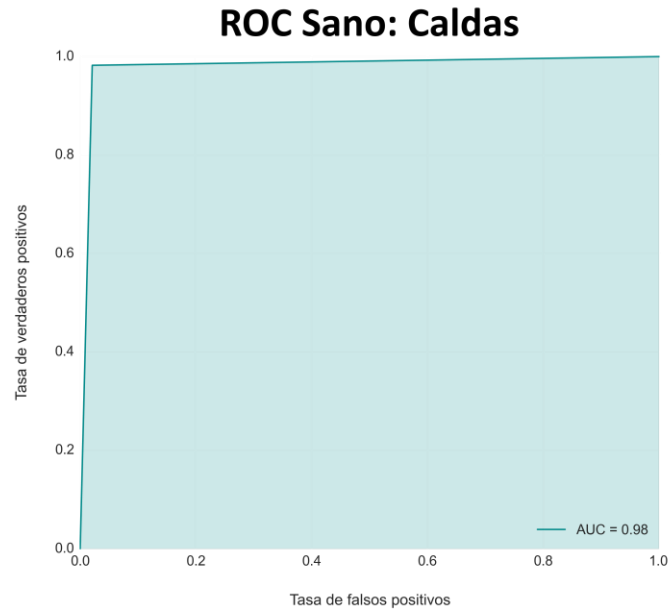


Ilustración 90 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del bagging de árboles de decisión sobre la base de datos de Caldas.

4.1.11.4 Métricas de desempeño

A continuación se muestran las métricas de desempeño calculadas a partir de las matrices de confusión y de las curvas ROC para el *bagging* de árboles de decisión.

ADNI		Caldas	
Exactitud	0,9008	Exactitud	0,9730
Precisión	0,9006	Precisión	0,9732
TPR (Sensibilidad)	0,9008	TPR (Sensibilidad)	0,9730
Especificidad	0,9635	Especificidad	0,9797
AUC Leve	0,9665	AUC Leve	0,9755
AUC Moderado/Severo	0,9643	AUC Moderado/Severo	0,9786
AUC Sano	0,9987	AUC Sano	0,9823
J Leve	0,7787	J Leve	0,9417
J Moderado/Severo	0,6860	J Moderado/Severo	0,9626
J Sano	0,9978	J Sano	0,9570
Kappa	0,8421	Kappa	0,9486

Tabla 13 Métricas de desempeño para el bagging de árboles de decisión sobre las bases de datos de la ADNI y Caldas.

4.1.12 Técnica 11: *Decision Trees (Boosting)*

El *boosting* es una meta-técnica de ensamble que ajusta diferentes clasificadores base — cada uno en la totalidad del conjunto de datos original— pero las instancias que fueron mal etiquetadas por los clasificadores previos se les da más peso, de manera que los clasificadores subsecuentes presten mayor atención a estos durante la etapa de entrenamiento. Dicha meta-técnica es frecuentemente usada como una vía para reducir el sesgo de un estimador de caja negra (ej. un árbol de decisión), mediante la combinación lineal de múltiples clasificadores «débiles» para formar uno con mayor exactitud (Schapire, 2003).

El procedimiento para realizar el *boosting* de árboles de decisión se concreta en cinco pasos:

1. Construir un árbol de decisión a partir del conjunto de datos original.
2. Incrementar el peso de las instancias de entrenamiento que fueron mal clasificadas.
3. Construir un nuevo árbol de decisión teniendo en cuenta los pesos actualizados.
4. Repetir los pasos 2 a 3 múltiples veces.
5. Calcular un clasificador agregado usando los árboles de decisión formados en los pasos 1 a 4. Para determinar el resultado de predicción, se debe tomar una combinación lineal de las predicciones de cada árbol de decisión (Roe, Yang, & Zhu, 2005).

4.1.12.1 Construcción, entrenamiento y clasificación

El *boosting* de árboles de decisión se construye empleando las funciones *templateTree* y *fitensemble('AdaBoostM2', 30, templateTree, 'Type', 'Classification', 'LearnRate', 0.1,...)* perteneciente a los algoritmos de regresión de *Statistics and Machine Learning Toolbox™*. El entrenamiento se realiza bajo la técnica de validación cruzada con 10 iteraciones empleando las funciones *fitensemble* y *crossval* pertenecientes a los algoritmos de regresión y clasificación de *Statistics and Machine Learning Toolbox™*. Finalmente, se generan los estimados para cada registro de ambas bases de datos (*ADNI* y *Caldas*)

empleando la función *kfoldPredict* perteneciente a los algoritmos de clasificación de *Statistics and Machine Learning Toolbox™*.

4.1.12.2 Matrices de confusión

En las ilustraciones 91 y 92 se muestran las matrices de confusión obtenidas para el *boosting* de árboles de decisión sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se deduce que el clasificador logró un porcentaje de acierto del 0.90% y 0.31% en las respectivas predicciones. Se puede observar que existe un bajo grado de confusión en las estimaciones de la técnica de aprendizaje de máquina respecto a las clases de predicción en la base de datos de la *ADNI*. No obstante, existe un alto grado de confusión en las estimaciones de la técnica respecto a las clases de predicción en la base de datos de *Caldas*.

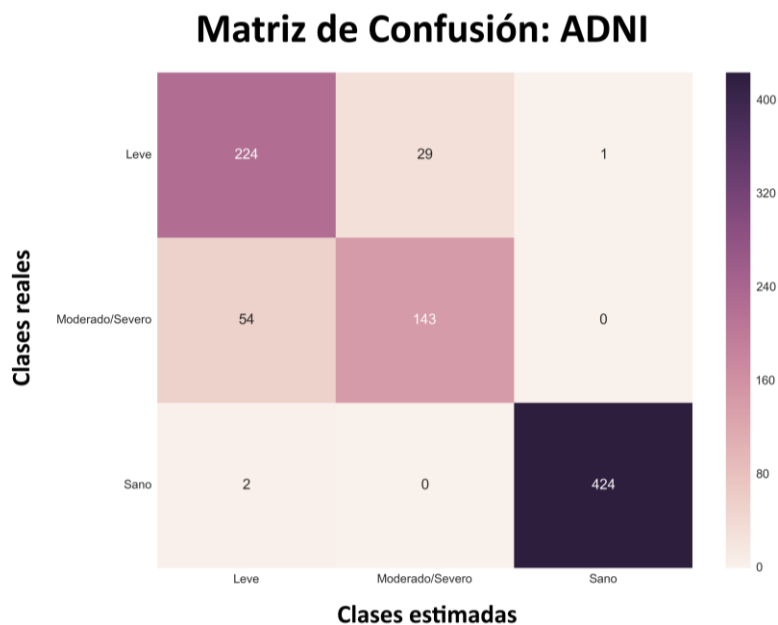


Ilustración 91 Matriz de confusión referente a los estimados del boosting de árboles de decisión sobre la base de datos de la ADNI.

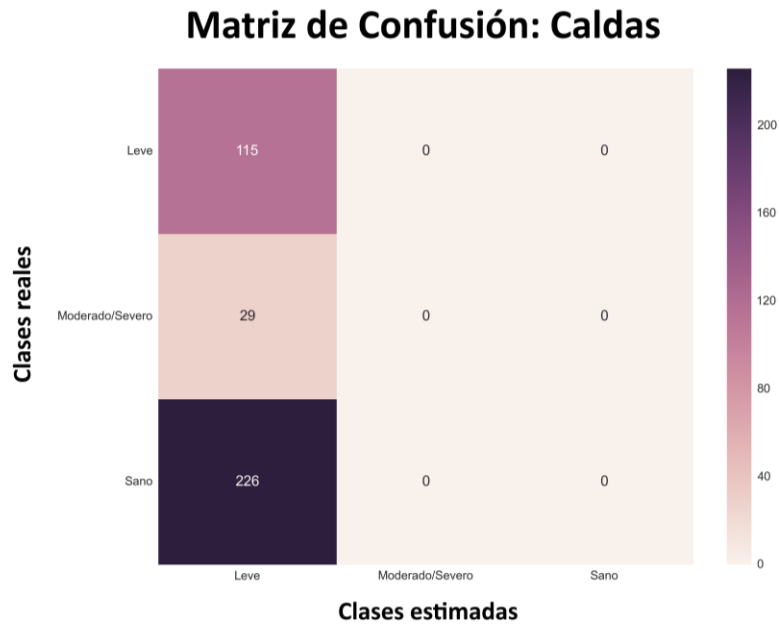


Ilustración 92 Matriz de confusión referente a los estimados del boosting de árboles de decisión sobre la base de datos de Caldas.

4.1.12.3 Curvas ROC

En las ilustraciones 93 a 98 se muestran las curvas ROC obtenidas para el *boosting* de árboles de decisión sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se observa que el AUC para cada una de las clases en la base de datos de la *ADNI* se encuentra bastante cercana de 1, lo cual indica que la técnica de aprendizaje de máquina no presenta mayores dificultades para elegir la clase correcta al momento de realizar las predicciones sobre dicha base de datos. Por otro lado, el AUC para cada una de las clases en la base de datos de *Caldas* se encuentra bastante alejada de 1, lo cual indica que la técnica de aprendizaje de máquina presenta grandes dificultades para elegir la clase correcta al momento de realizar las predicciones sobre dicha base de datos.

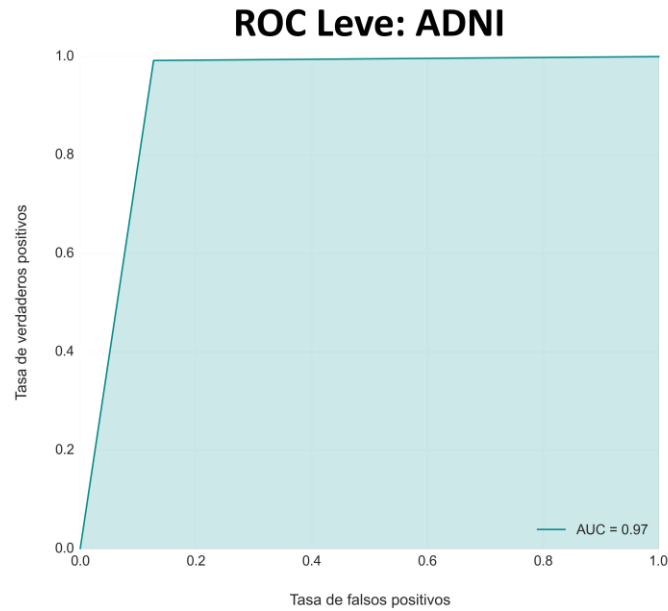


Ilustración 93 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del boosting de árboles de decisión sobre la base de datos de la ADNI.

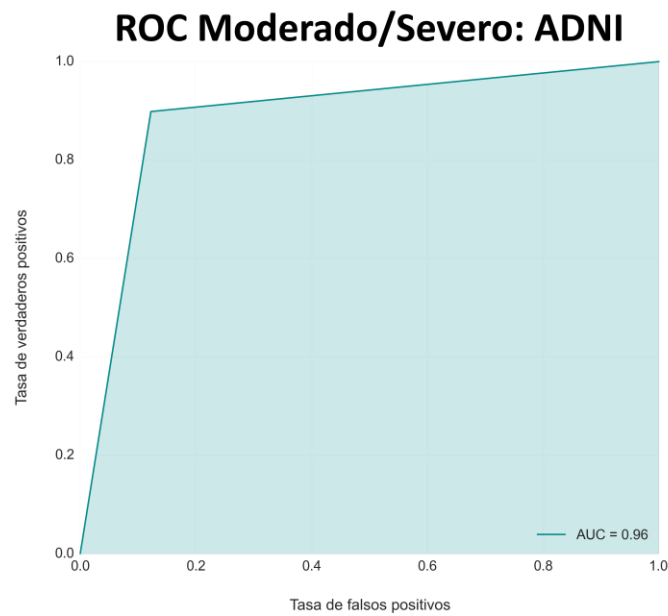


Ilustración 94 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del boosting de árboles de decisión sobre la base de datos de la ADNI.

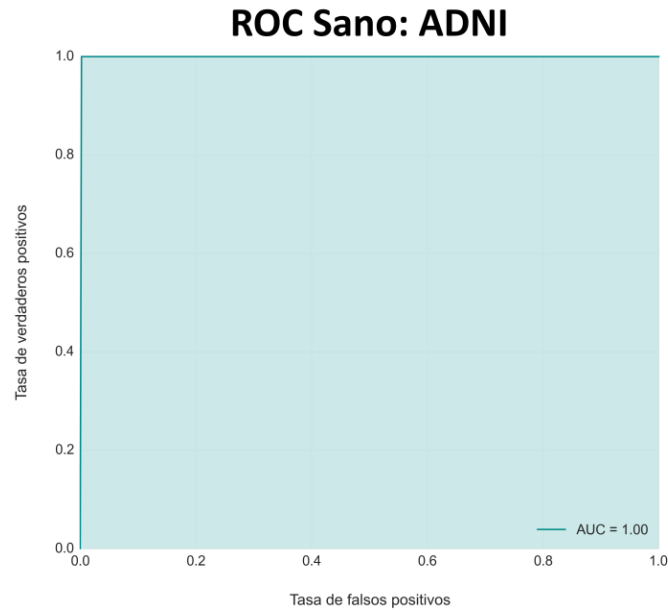


Ilustración 95 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo— referente a los estimados del boosting de árboles de decisión sobre la base de datos de la ADNI.

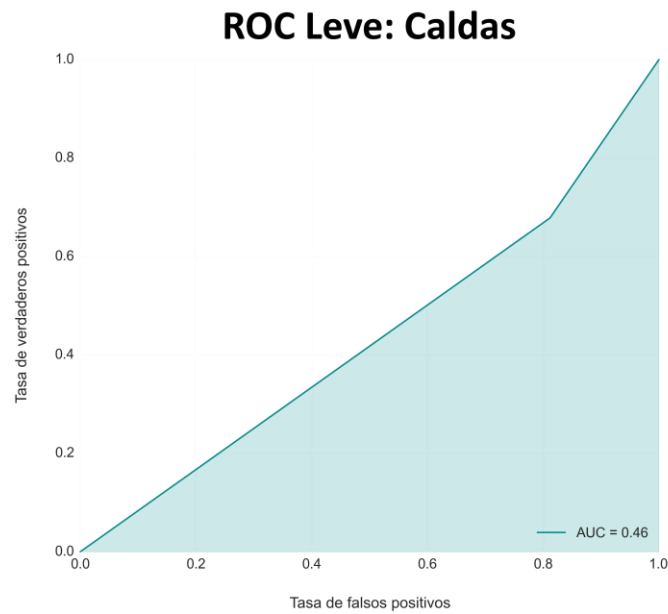


Ilustración 96 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del boosting de árboles de decisión sobre la base de datos de Caldas.

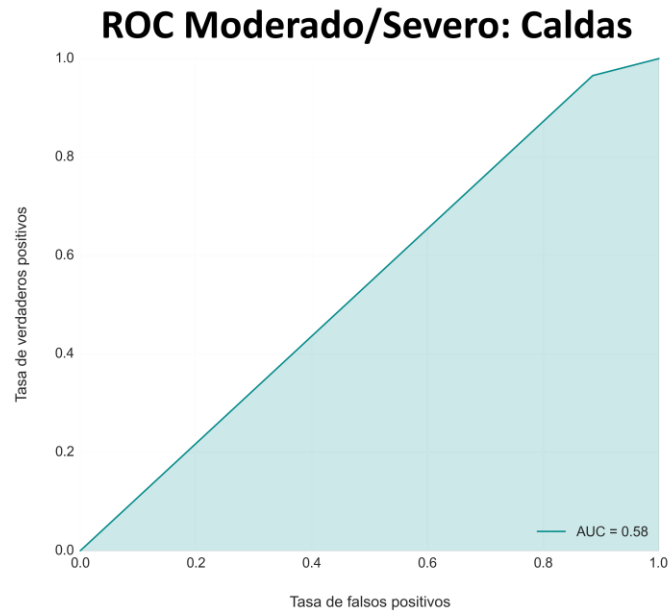


Ilustración 97 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del boosting de árboles de decisión sobre la base de datos de Caldas.

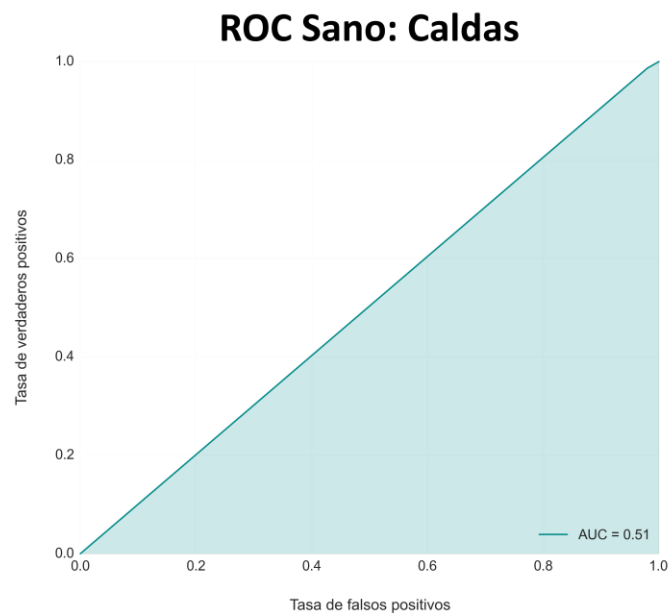


Ilustración 98 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del boosting de árboles de decisión sobre la base de datos de Caldas.

4.1.12.4 Métricas de desempeño

A continuación se muestran las métricas de desempeño calculadas a partir de las matrices de confusión y de las curvas ROC para el *boosting* de árboles de decisión.

ADNI		Caldas	
Exactitud	0,9019	Exactitud	0,3108
Precisión	0,9031	Precisión	0,7858
TPR (Sensibilidad)	0,9019	TPR (Sensibilidad)	0,3108
Especificidad	0,9633	Especificidad	0,6892
AUC Leve	0,9680	AUC Leve	0,4611
AUC Moderado/Severo	0,9646	AUC Moderado/Severo	0,5756
AUC Sano	0,9955	AUC Sano	0,5121
J Leve	0,7920	J Leve	0,0000
J Moderado/Severo	0,6832	J Moderado/Severo	0,0000
J Sano	0,9931	J Sano	0,0000
Kappa	0,8439	Kappa	0,0000

Tabla 14 Métricas de desempeño para el *boosting* de árboles de decisión sobre las bases de datos de la ADNI y Caldas.

4.1.13 Técnica 12: *AdaBoost-SAMME.R*

AdaBoost es una técnica de aprendizaje de máquina basada en el *boosting* que implica un procedimiento iterativo donde se combinan múltiples clasificadores «débiles» para aproximar el clasificador de Bayes. Cuando es aplicada a problemas de clasificación binaria, esta técnica ha demostrado ser bastante exitosa en la producción de clasificadores precisos. Sin embargo, no es el caso para problemas con múltiples clases, aun cuando *AdaBoost* fue propuesto para ser usado también en dichos problemas (Zhu, Rosset, Zou, & Hastie, 2006).

A fin de abordar situaciones con múltiples clases mediante *AdaBoost*, Zhu et al. (2006) propusieron un algoritmo fundamentado en esta última técnica al cual denominaron *SAMME* —*Stagewise Additive Modeling using a Multi-class Exponential loss function*—.

Siendo $T(x)$ un clasificador «débil» de múltiples clases que asigna una etiqueta de clase a x , el algoritmo SAMME procede de la siguiente manera:

1. Inicializar los pesos de observación $w_i = \frac{1}{n}, i = 1, 2, \dots, n$.
2. Desde $m = 1$ hasta M :
 - a. Ajustar un clasificador $T^{(m)}(x)$ a los datos de entrenamiento usando los pesos w_i .
 - b. Computar

$$err^{(m)} = \sum_{i=1}^n w_i \mathbb{I}[(c_i \neq T^{(m)}(x_i))] / \sum_{i=1}^n w_i.$$

- c. Computar

$$\alpha^{(m)} = \log \frac{1 - err^{(m)}}{err^{(m)}} + \log(K - 1).$$

- d. Fijar

$$w_i \leftarrow w_i \cdot e^{\alpha^{(m)} \cdot \mathbb{I}[(c_i \neq T^{(m)}(x_i))]}, i = 1, \dots, n.$$

- e. Normalizar w_i .

3. Salida

$$C(x) = \arg \max_k \sum_{m=1}^M \alpha^{(m)} \cdot \mathbb{I}[(T^{(m)}(x) = k)].$$

Aquí, los datos de entrenamiento están dados por $(x_1, c_1), \dots, (x_n, c_n)$ donde $x_i \in \mathbb{R}^d$ son las variables dependientes y c_i son las etiquetas de clase que asumen valores en un conjunto finito, ej. $\{1, 2, \dots, K\}$. El término K representa el número de clases.

El algoritmo SAMME espera que el aprendizaje proporcione un clasificador $T(x) \in \{1, \dots, K\}$. Una alternativa es el uso de predicciones de confianza valoradas en tiempo real —tales como las estimaciones de probabilidad ponderada— para actualizar el modelo aditivo, en lugar de actualizar las propias clasificaciones. Esta variación del algoritmo se denomina SAMME.R (Zhu, Rosset, Zou, & Hastie, 2006).

4.1.13.1 Construcción, entrenamiento y clasificación

El clasificador *AdaBoost-SAMME.R* se construye empleando la clase *AdaBoostClassifier* perteneciente a los modelos de ensamble de *Scikit-learn*. Por defecto, la clase *AdaBoostClassifier* instancia un ensamble de árboles de decisión y trabaja con el algoritmo *SAMME.R*. El entrenamiento se realiza bajo la técnica de validación cruzada con 10 iteraciones empleando la función *cross_val_score* perteneciente al módulo *cross-validation* de *Scikit-learn*. Finalmente, se generan los estimados para cada registro de ambas bases de datos (*ADNI* y *Caldas*) empleando la función *cross_val_predict* perteneciente al módulo *cross-validation* de *Scikit-learn*.

4.1.13.2 Matrices de confusión

En las ilustraciones 99 y 100 se muestran las matrices de confusión obtenidas para el clasificador *AdaBoost-SAMME.R* sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se deduce que el clasificador logró un porcentaje de acierto del 0.88% y 0.81% en las respectivas predicciones. No obstante, se puede observar que existe un mediano grado de confusión en las estimaciones de la técnica de aprendizaje de máquina respecto a las clases de predicción «Leve» y «Moderado/Severo».

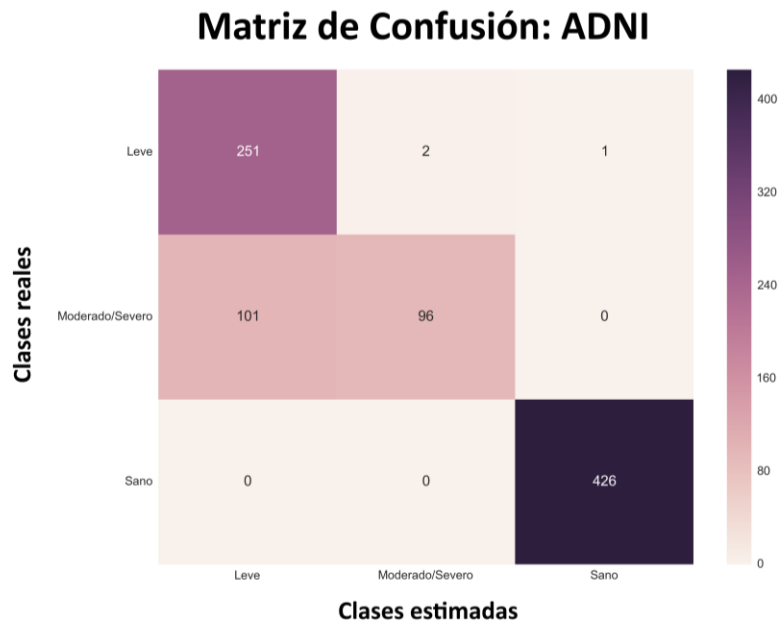


Ilustración 99 Matriz de confusión referente a los estimados del AdaBoost-SAMME.R sobre la base de datos de la ADNI.

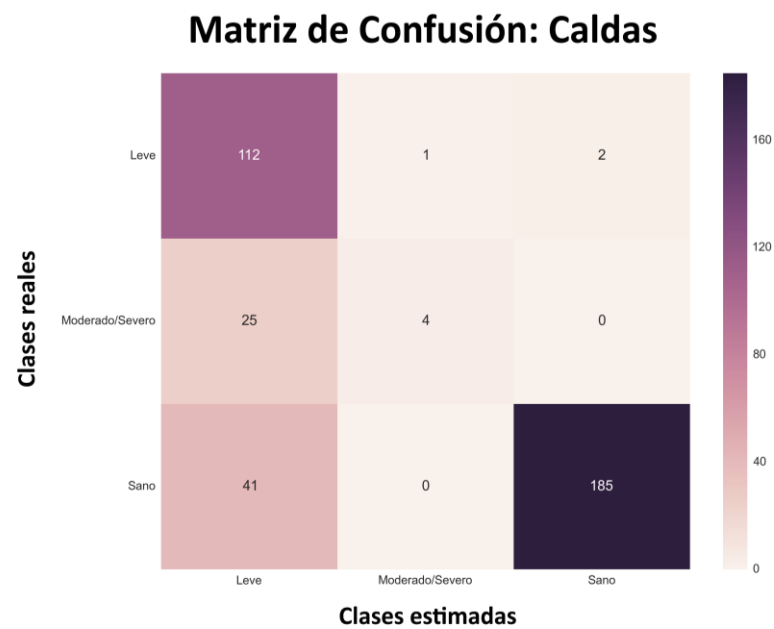


Ilustración 100 Matriz de confusión referente a los estimados del AdaBoost-SAMME.R sobre la base de datos de Caldas.

4.1.13.3 Curvas ROC

En las ilustraciones 101 a 106 se muestran las curvas ROC obtenidas para el clasificador *AdaBoost-SAMME.R* sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se observa que las AUC para la clase «Moderado/Severo» se encuentran bastante alejadas de 1, lo cual indica que la técnica de aprendizaje de máquina presenta grandes dificultades para diferenciar la clase mencionada de las demás, al momento de realizar las predicciones. No obstante, las AUC para las clases «Sano» y «Leve» se encuentra medianamente cercanas a 1, es decir que la técnica es capaz de diferenciar dichas clases de las demás sin mayores dificultades.

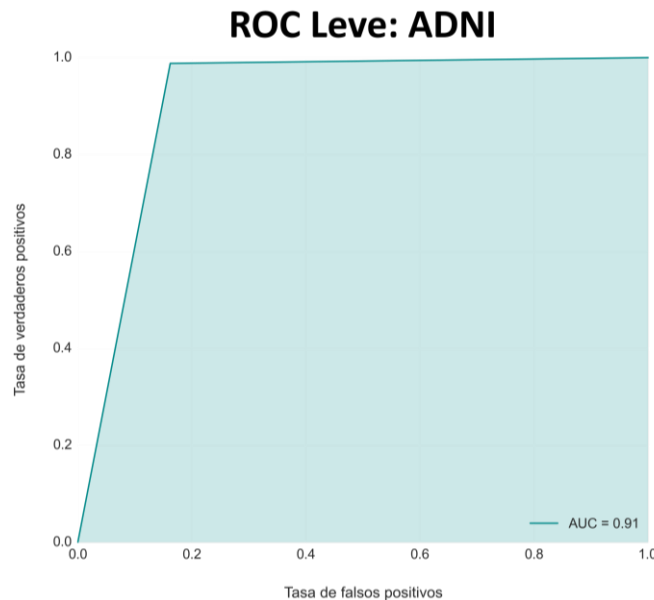


Ilustración 101 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del AdaBoost-SAMME.R sobre la base de datos de la ADNI.

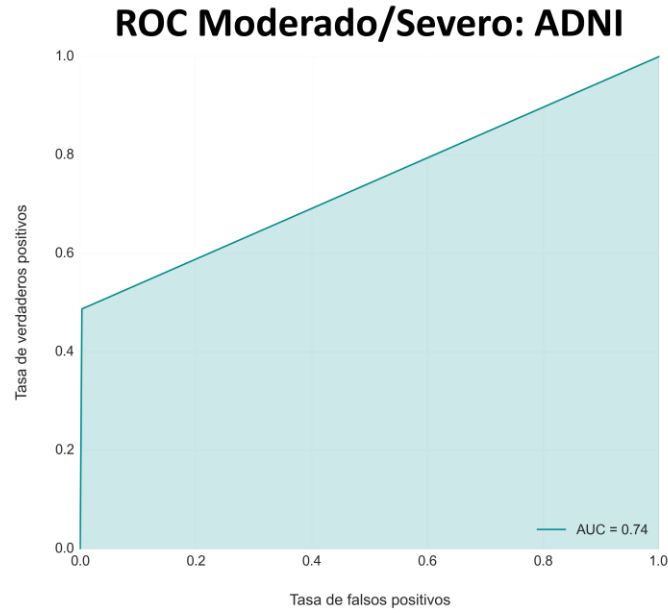


Ilustración 102 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del AdaBoost-SAMME.R sobre la base de datos de la ADNI.

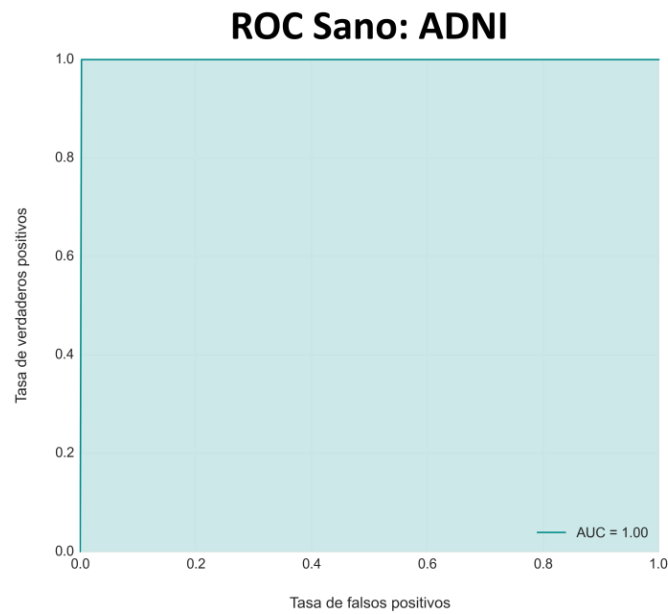


Ilustración 103 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del AdaBoost-SAMME.R sobre la base de datos de la ADNI.

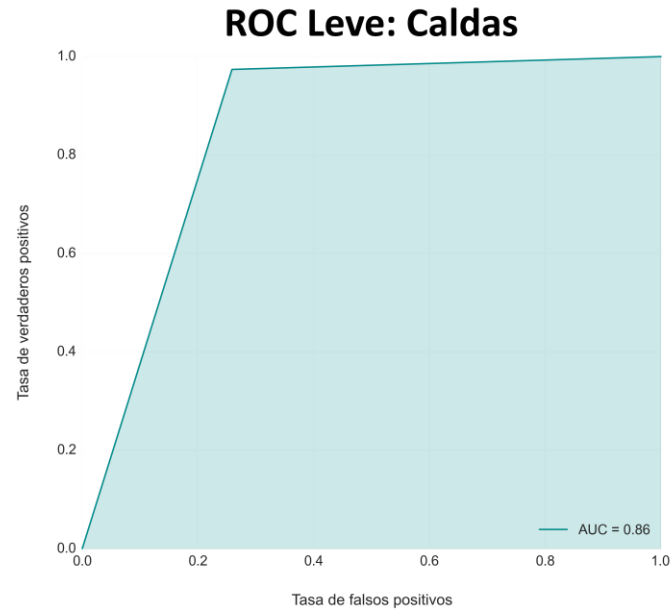


Ilustración 104 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del AdaBoost-SAMME.R sobre la base de datos de Caldas.

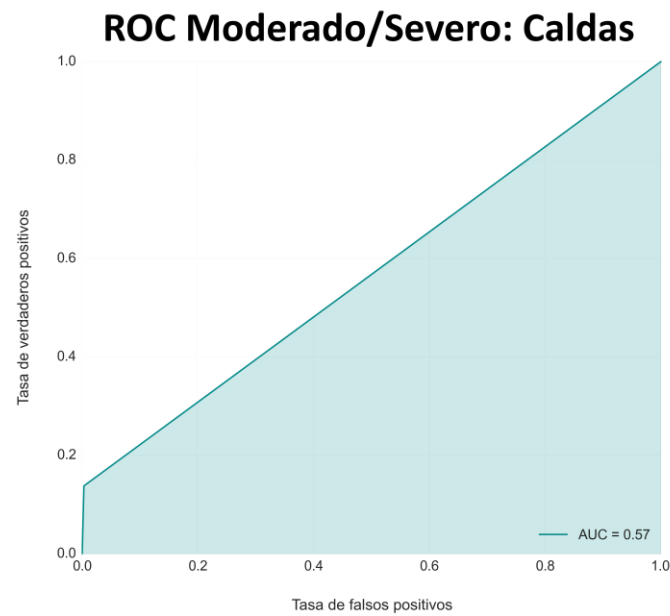


Ilustración 105 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del AdaBoost-SAMME.R sobre la base de datos de Caldas.

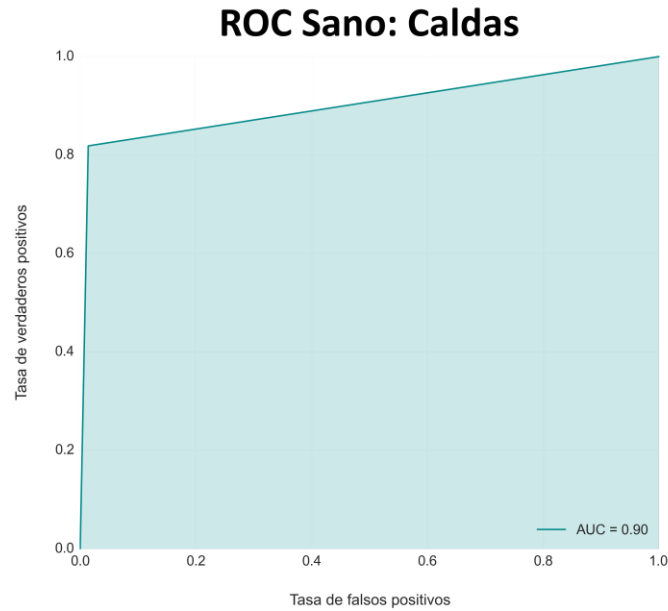


Ilustración 106 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del AdaBoost-SAMME.R sobre la base de datos de Caldas.

4.1.13.4 Métricas de desempeño

A continuación se muestran las métricas de desempeño calculadas a partir de las matrices de confusión y de las curvas ROC para el *AdaBoost-SAMME.R*.

ADNI		Caldas	
Exactitud	0,8814	Exactitud	0,8135
Precisión	0,9112	Precisión	0,8625
TPR (Sensibilidad)	0,8814	TPR (Sensibilidad)	0,8135
Especificidad	0,9513	Especificidad	0,9108
AUC Leve	0,9130	AUC Leve	0,8575
AUC Moderado/Severo	0,7422	AUC Moderado/Severo	0,5675
AUC Sano	0,9989	AUC Sano	0,9023
J Leve	0,8261	J Leve	0,7151
J Moderado/Severo	0,4844	J Moderado/Severo	0,1350
J Sano	0,9978	J Sano	0,8047
Kappa	0,8094	Kappa	0,6551

Tabla 15 Métricas de desempeño para el AdaBoost-SAMME.R sobre las bases de datos de la ADNI y Caldas.

4.1.14 Técnica 13: *Random Forest*

Un *Random Forest* (RF) o bosque Aleatorio es una técnica de ensamble de árboles predictivos tales que cada árbol depende de los valores de un vector muestreado aleatoriamente —con la misma distribución para todos los árboles del bosque—. RF es similar al *bagging* de árboles de decisión en cuanto a que ambas técnicas construyen múltiples árboles y combinan sus predicciones por mayoría de votos (Breiman, 2001). No obstante, el mecanismo de segmentación en los bosques aleatorios solo considera m características seleccionadas al azar en cada uno de los *splits*, usualmente $m = \sqrt{(\text{número de características})}$ (Scikit-learn, 2015). En contraste, el *bagging* de árboles de decisión considera todo el conjunto de características en cada uno de los *splits* (Breiman, 1994).

4.1.14.1 Construcción, entrenamiento y clasificación

El bosque aleatorio se construye empleando la clase *RandomForestClassifier* perteneciente a los modelos de ensamble de *Scikit-learn*. El entrenamiento se realiza bajo la técnica de validación cruzada con 10 iteraciones empleando la función *cross_val_score* perteneciente al módulo *cross-validation* de *Scikit-learn*. Finalmente, se generan los estimados para cada registro de ambas bases de datos (*ADNI* y *Caldas*) empleando la función *cross_val_predict* perteneciente al módulo *cross-validation* de *Scikit-learn*.

4.1.14.2 Matrices de confusión

En las ilustraciones 107 y 108 se muestran las matrices de confusión obtenidas para el bosque aleatorio sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se deduce que el clasificador logró un porcentaje de acierto del 0.87% y 0.95% en las respectivas predicciones. Asimismo, se puede observar que existe un bajo grado de confusión en las estimaciones de la técnica de aprendizaje de máquina respecto a las clases de predicción.

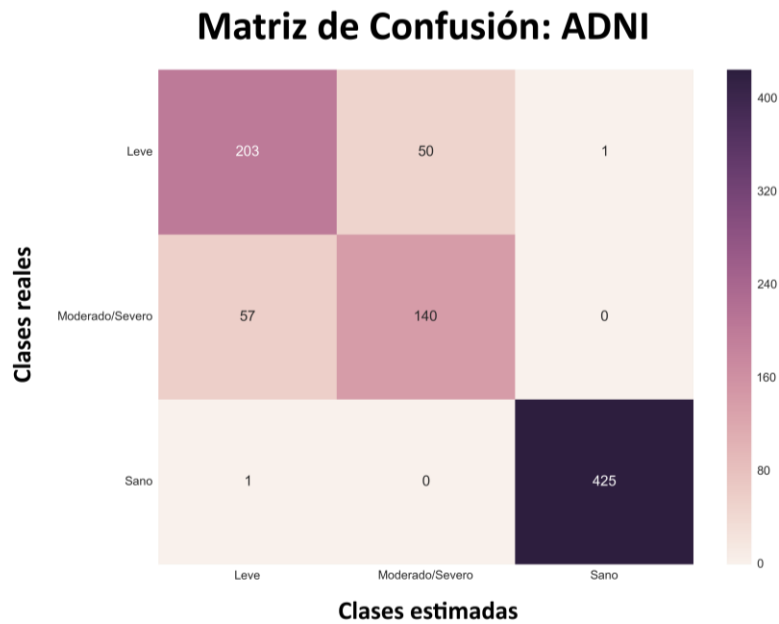


Ilustración 107 Matriz de confusión referente a los estimados del bosque aleatorio sobre la base de datos de la ADNI.

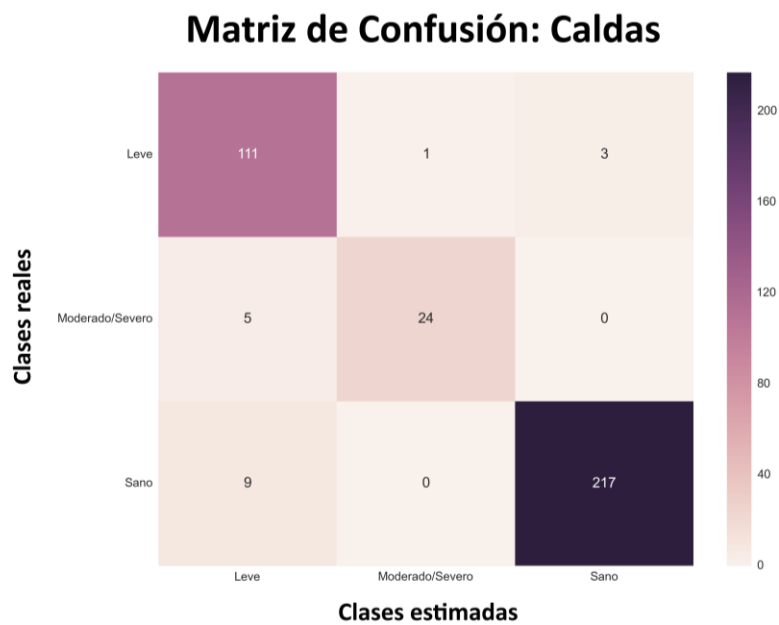


Ilustración 108 Matriz de confusión referente a los estimados del bosque aleatorio sobre la base de datos de Caldas.

4.1.14.3 Curvas ROC

En las ilustraciones 109 a 114 se muestran las curvas ROC obtenidas para el bosque aleatorio sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se observa que el AUC para cada una de las clases se encuentra bastante cercana de 1, lo cual indica que la técnica de aprendizaje de máquina no presenta mayores dificultades para elegir la clase correcta al momento de realizar las predicciones. Solamente se presentan algunos inconvenientes menores para diferenciar las clases «Leve» y «Moderado/Severo» de las demás en la base de datos de la *ADNI*.

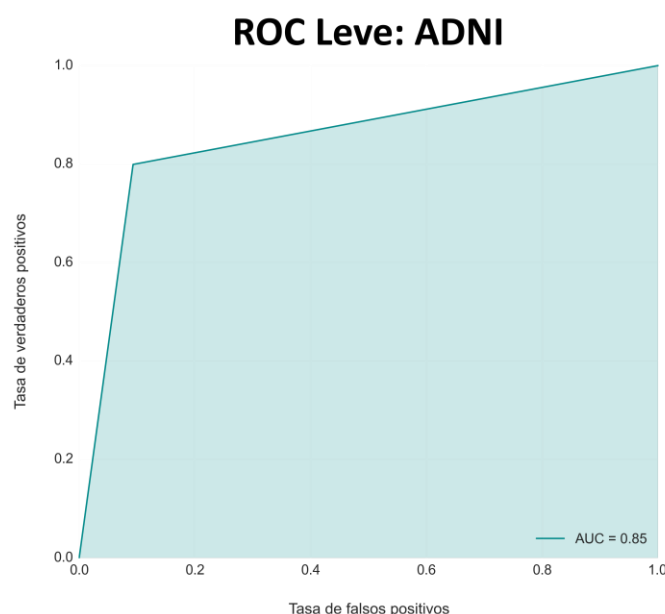


Ilustración 109 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del bosque aleatorio sobre la base de datos de la *ADNI*.

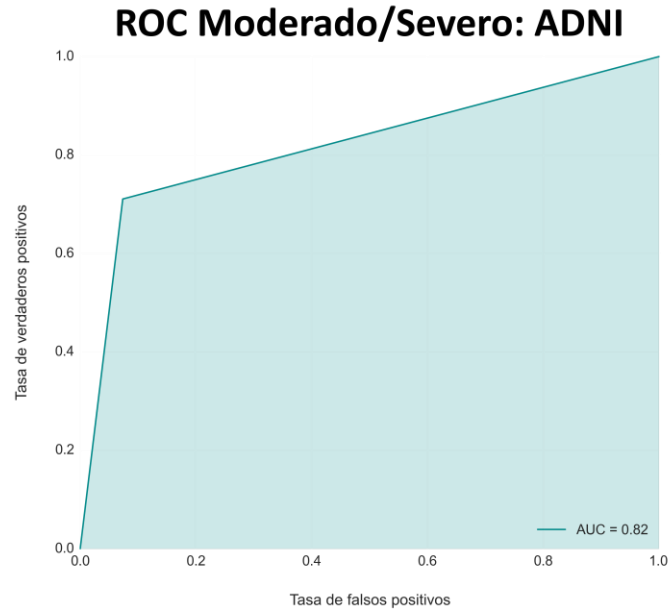


Ilustración 110 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del bosque aleatorio sobre la base de datos de la ADNI.

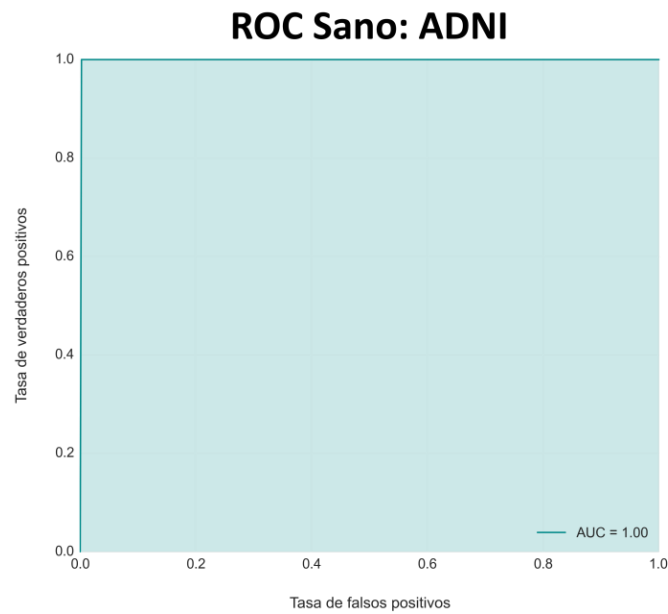


Ilustración 111 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del bosque aleatorio sobre la base de datos de la ADNI.

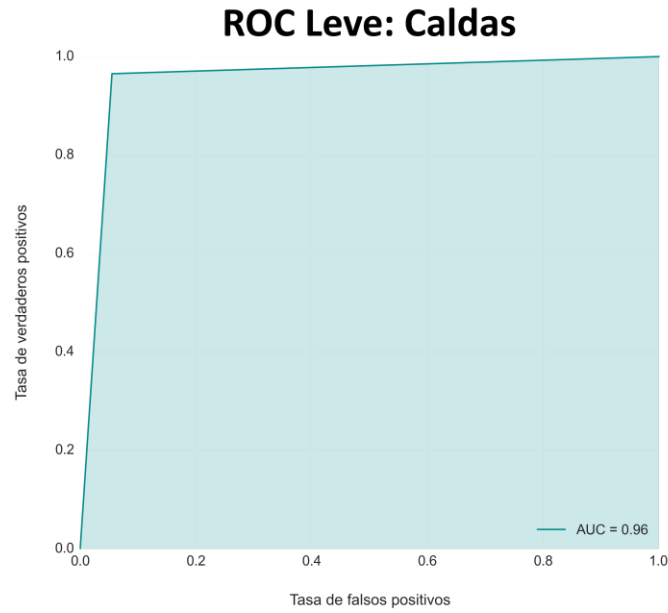


Ilustración 112 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del bosque aleatorio sobre la base de datos de Caldas.

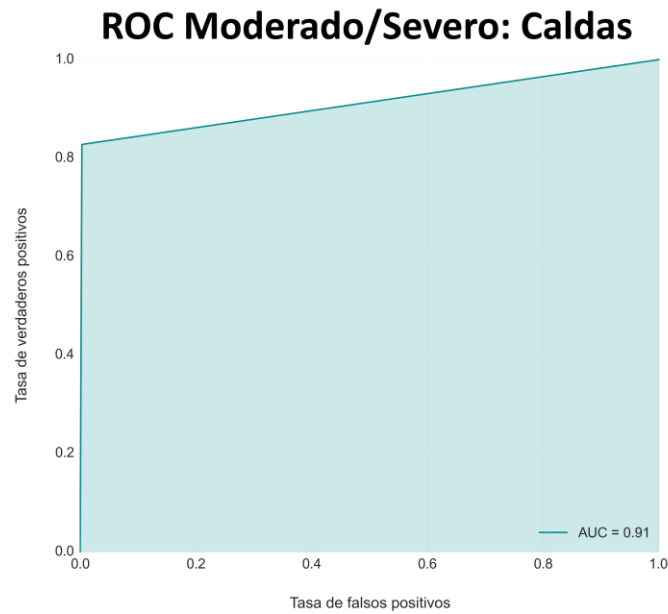


Ilustración 113 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del bosque aleatorio sobre la base de datos de Caldas.

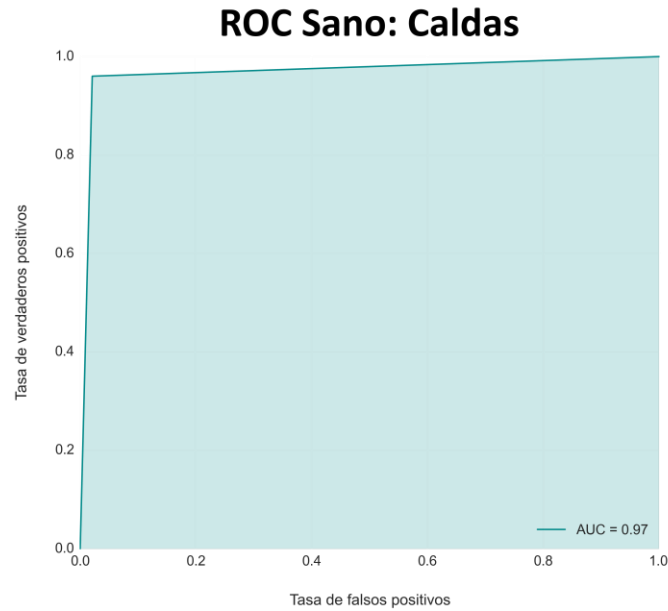


Ilustración 114 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del bosque aleatorio sobre la base de datos de Caldas.

4.1.14.4 Métricas de desempeño

A continuación se muestran las métricas de desempeño calculadas a partir de las matrices de confusión y de las curvas ROC para el bosque aleatorio.

ADNI		Caldas	
Exactitud	0,8757	Exactitud	0,9514
Precisión	0,8754	Precisión	0,9537
TPR (Sensibilidad)	0,8757	TPR (Sensibilidad)	0,9514
Especificidad	0,9554	Especificidad	0,9700
AUC Leve	0,8531	AUC Leve	0,9552
AUC Moderado/Severo	0,8186	AUC Moderado/Severo	0,9123
AUC Sano	0,9977	AUC Sano	0,9697
J Leve	0,7061	J Leve	0,9103
J Moderado/Severo	0,6371	J Moderado/Severo	0,8247
J Sano	0,9954	J Sano	0,9393
Kappa	0,8025	Kappa	0,9076

Tabla 16 Métricas de desempeño para el bosque aleatorio sobre las bases de datos de la ADNI y Caldas.

4.1.14.5 Importancia de las variables en la base de datos de la ADNI

Adicional a la capacidad de clasificación, los bosques aleatorios ofrecen otra información valiosa: la importancia de las variables. La medida base para elegir los *splits* en los árboles de decisión del bosque se denomina impureza. Así, cuando se entrena un árbol, se puede calcular cuánto influye cada variable en la impureza de este último (Saabas, 2014).

Para un bosque aleatorio la disminución de la impureza de cada variable se puede promediar, lo cual permite establecer un ranking y observar cuáles son más significativas para la clasificación de los datos (Saabas, 2014). Esta medida es extremadamente útil al seleccionar un conjunto reducido de variables para disminuir la complejidad y tiempo de procesamiento de la etapa de entrenamiento sin comprometer en gran medida la exactitud de un clasificador.

En la ilustración 115 se muestran las importancias de las variables para la base de datos de la ADNI. A partir de estas se deduce que las variables CRD, MMSE, Edad y Educación son las que más contribuyen al momento de realizar las predicciones correctas sobre los registros de dicha base de datos.

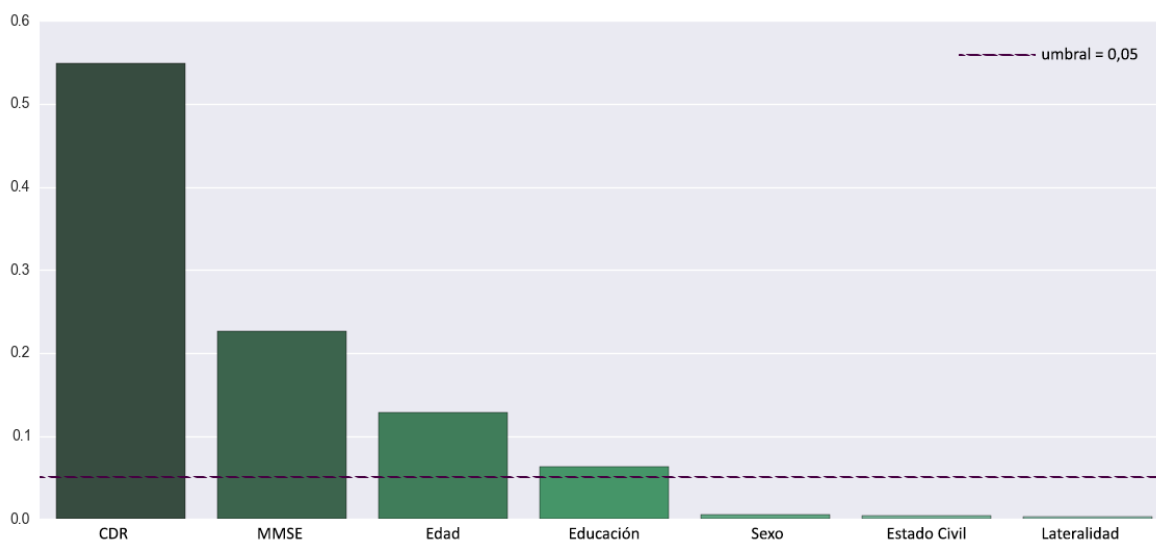


Ilustración 115 Importancia de las variables en la base de datos de la ADNI. Cuanto más alta es la barra, más significativa es la variable para la clasificación de los datos.

4.1.14.6 Importancia de las variables en la base de datos de Caldas

En la ilustración 116 se muestran las importancias de las variables para la base de datos de Caldas. A partir de estas se deduce que las variables CRD, GDS, MoCA y MMSE son las que más contribuyen al momento de realizar las predicciones correctas sobre los registros de dicha base de datos.

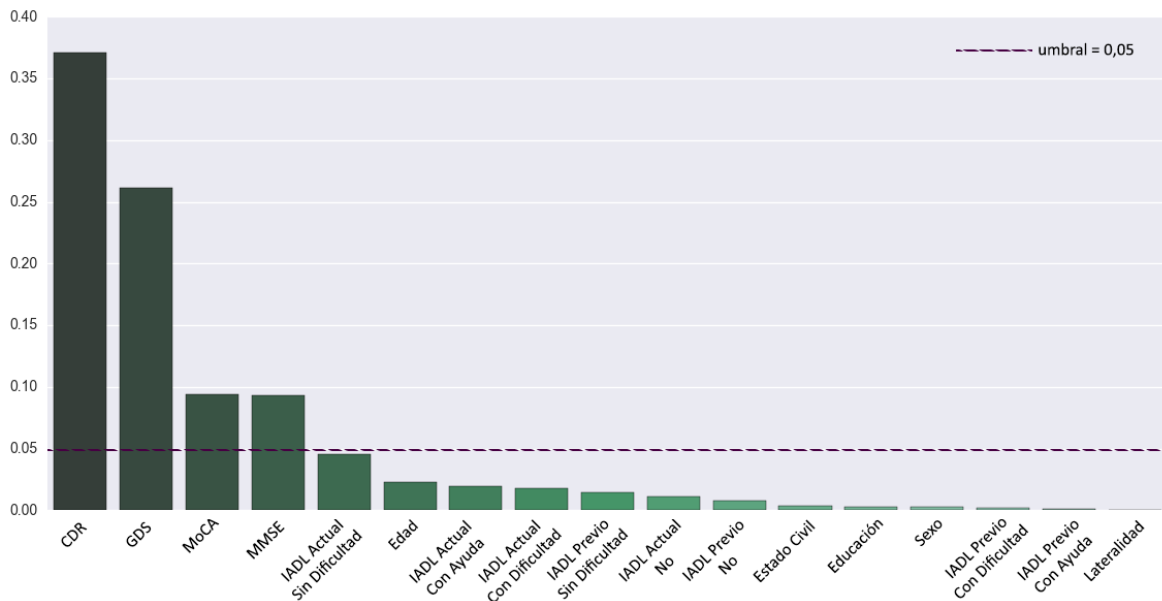


Ilustración 116 Importancia de las variables en la base de datos de Caldas. Cuanto más alta es la barra, más significativa es la variable para la clasificación de los datos.

4.1.15 Técnica 14: *Extremely Randomized Trees*

Extremely Randomized Trees o *Extra-Trees* es una técnica de ensamble basada en árboles, para abordar problemas de clasificación y regresión. En esencia, consiste en aleatorizar fuertemente el proceso de selección del atributo y los puntos de corte durante la realización de cada *split*. En el caso extremo, se construyen árboles totalmente aleatorios cuyas estructuras son independientes de los valores de salida de los datos de entrenamiento (Geurts, Ernst, & Wehenkel, 2006).

El algoritmo de *Extra-Trees* construye un ensamble de árboles de clasificación o regresión de acuerdo al enfoque clásico *Top-Down*. Sus dos principales diferencias con las demás

técnicas basadas en ensambles de árboles radican en la aleatoriedad involucrada en cada uno de los *splits* y el uso de la totalidad del conjunto de datos para la fase de entrenamiento (en lugar de una muestra aleatoria). Por último, las predicciones de los árboles son agregadas para obtener la predicción final —por mayoría de votos en problemas de clasificación y por media aritmética en problemas de regresión— (Geurts, Ernst, & Wehenkel, 2006).

4.1.15.1 Construcción, entrenamiento y clasificación

El clasificador *Extra-Trees* se construye empleando la clase *ExtraTreesClassifier* perteneciente a los modelos de ensamble de *Scikit-learn*. El entrenamiento se realiza bajo la técnica de validación cruzada con 10 iteraciones empleando la función *cross_val_score* perteneciente al módulo *cross-validation* de *Scikit-learn*. Finalmente, se generan los estimados para cada registro de ambas bases de datos (*ADNI* y *Caldas*) empleando la función *cross_val_predict* perteneciente al módulo *cross-validation* de *Scikit-learn*.

4.1.15.2 Matrices de confusión

En las ilustraciones 117 y 118 se muestran las matrices de confusión obtenidas para el clasificador *Extra-Trees* sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se deduce que el clasificador logró un porcentaje de acierto del 0.87% y 0.95% en las respectivas predicciones. Asimismo, se puede observar que existe un bajo grado de confusión en las estimaciones de la técnica de aprendizaje de máquina respecto a las clases de predicción.

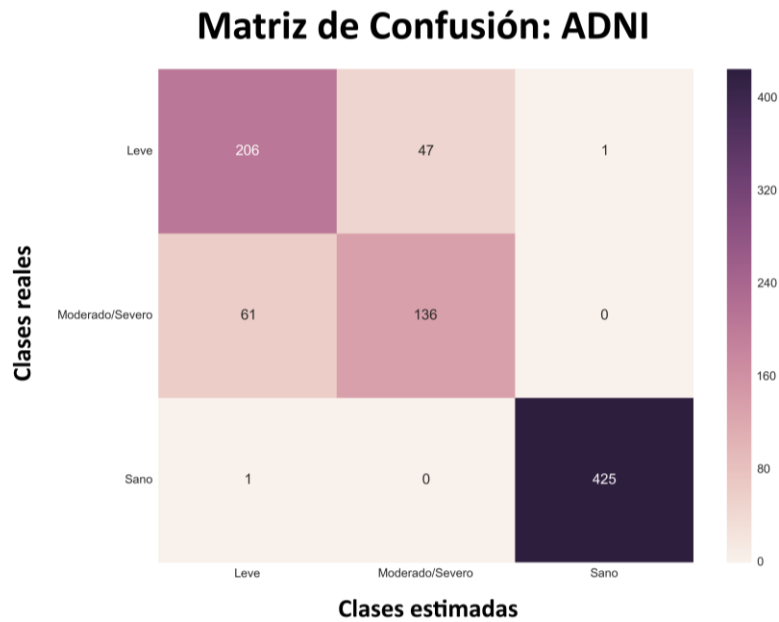


Ilustración 117 Matriz de confusión referente a los estimados del clasificador Extra-Trees sobre la base de datos de la ADNI.

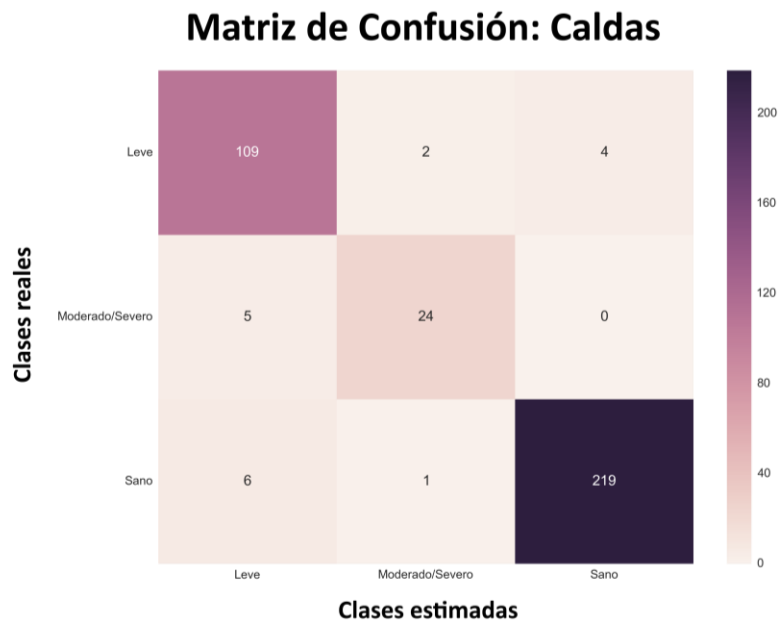


Ilustración 118 Matriz de confusión referente a los estimados del clasificador Extra-Trees sobre la base de datos de Caldas.

4.1.15.3 Curvas ROC

En las ilustraciones 119 a 124 se muestran las curvas ROC obtenidas para el clasificador *Extra-Trees* sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se observa que el AUC para cada una de las clases se encuentra bastante cercana de 1, lo cual indica que la técnica de aprendizaje de máquina no presenta mayores dificultades para elegir la clase correcta al momento de realizar las predicciones. Solamente se presentan algunos inconvenientes menores para diferenciar las clases «Leve» y «Moderado/Severo» de las demás en la base de datos de la *ADNI*.

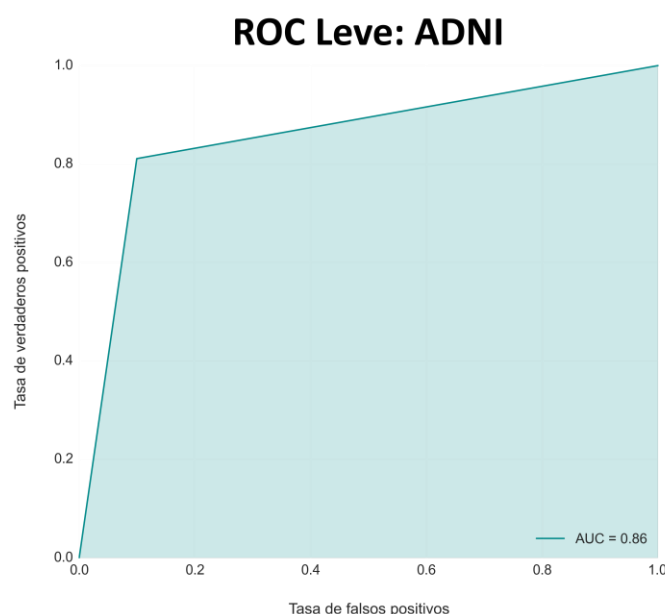


Ilustración 119 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del clasificador Extra-Trees sobre la base de datos de la ADNI.

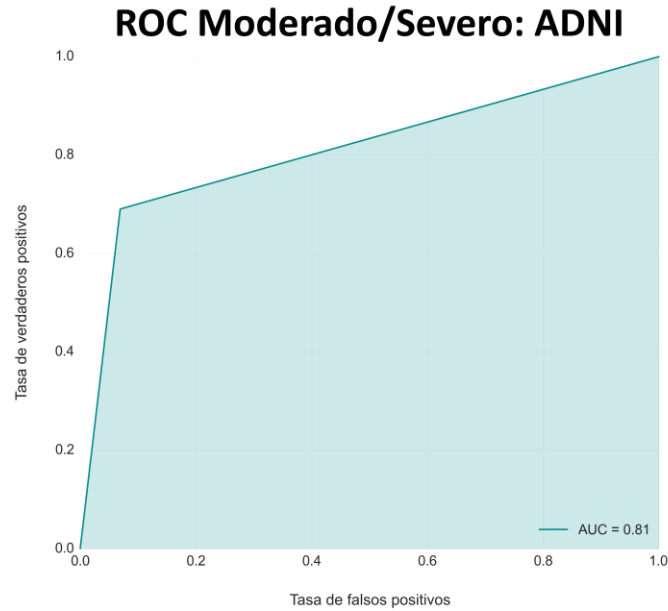


Ilustración 120 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del clasificador Extra-Trees sobre la base de datos de la ADNI.

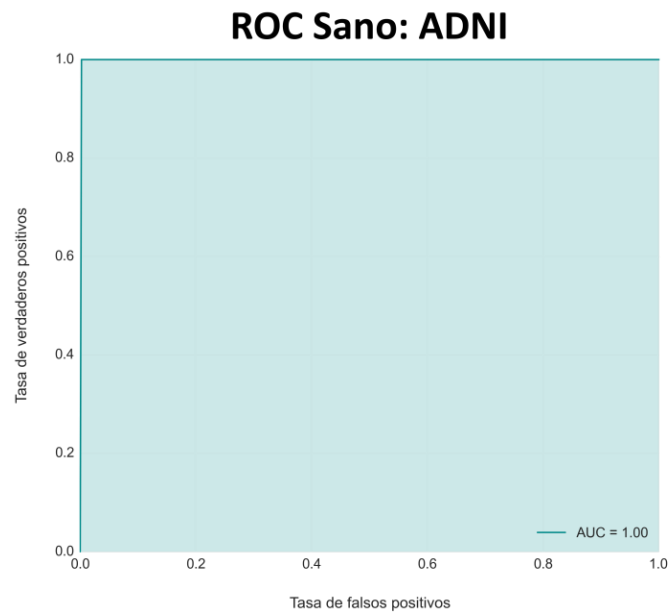


Ilustración 121 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del clasificador Extra-Trees sobre la base de datos de la ADNI.

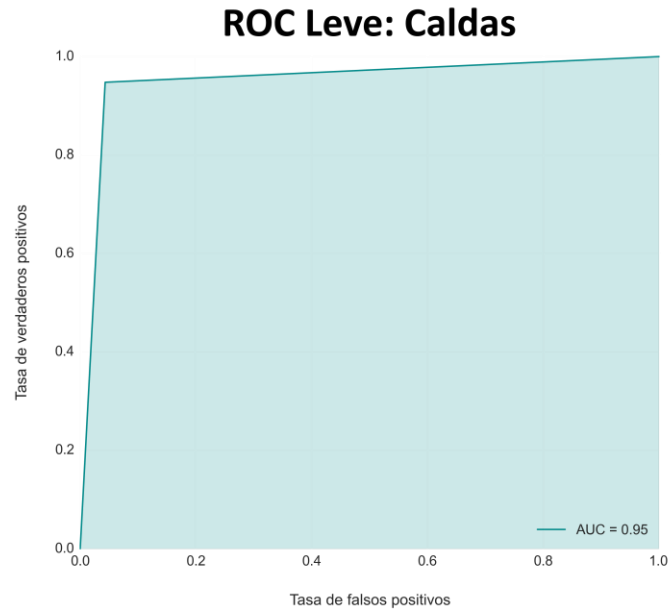


Ilustración 122 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del clasificador Extra-Trees sobre la base de datos de Caldas.

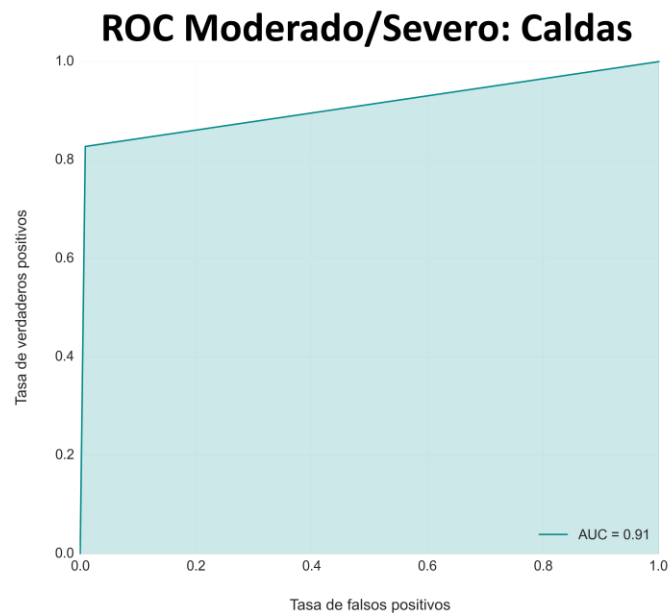


Ilustración 123 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del clasificador Extra-Trees sobre la base de datos de Caldas.

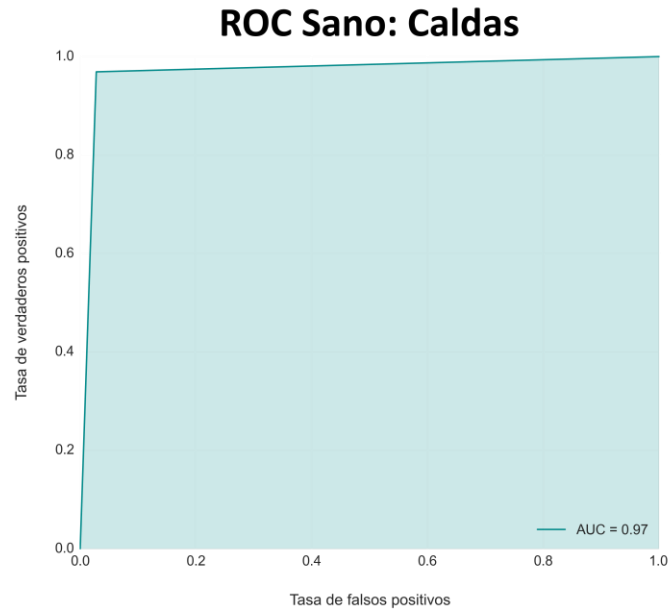


Ilustración 124 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del clasificador Extra-Trees sobre la base de datos de Caldas.

4.1.15.4 Métricas de desempeño

A continuación se muestran las métricas de desempeño calculadas a partir de las matrices de confusión y de las curvas ROC para el clasificador *Extra-Trees*.

ADNI		Caldas	
Exactitud	0,8746	Exactitud	0,9514
Precisión	0,8742	Precisión	0,9518
TPR (Sensibilidad)	0,8746	TPR (Sensibilidad)	0,9514
Especificidad	0,9546	Especificidad	0,9689
AUC Leve	0,8558	AUC Leve	0,9523
AUC Moderado/Severo	0,8106	AUC Moderado/Severo	0,9094
AUC Sano	0,9977	AUC Sano	0,9706
J Leve	0,7115	J Leve	0,9047
J Moderado/Severo	0,6212	J Moderado/Severo	0,8188
J Sano	0,9954	J Sano	0,9412
Kappa	0,8005	Kappa	0,9074

Tabla 17 Métricas de desempeño para el clasificador Extra-Trees sobre las bases de datos de la ADNI y Caldas.

4.1.16 Técnica 15: *RUSBoosted Trees*

RUSBoost es una técnica de ensamble especialmente efectiva en la clasificación de datos desbalanceados, lo cual significa que alguna de las clases en los datos de entrenamiento tiene menos miembros que las demás clases (Seiffert, Khoshgoftaar, Hulse, & Napolitano, 2009).

El algoritmo RUS (*Random Under Sampling*) toma N (el número de miembros de la clase con la menor cantidad de miembros), como la unidad básica de muestreo. Las clases con mayor cantidad de miembros son muestreadas tomando únicamente N observaciones de cada clase. En otras palabras, si hay K clases, por cada clasificador «débil» en el ensamble RUS toma un subconjunto de los datos con N observaciones de cada una de las K clases. Posteriormente, el procedimiento de *boosting* continúa mediante la técnica *AdaBoost* para construir el ensamble (MathWorks, 2016).

4.1.16.1 Construcción, entrenamiento y clasificación

El *RUSBoost* de árboles de decisión se construye empleando las funciones *templateTree* y *fitensemble('RUSBoost', 30, templateTree, 'Type', 'Classification', 'LearnRate', 0.1,...)* perteneciente a los algoritmos de regresión de *Statistics and Machine Learning Toolbox*TM. El entrenamiento se realiza bajo la técnica de validación cruzada con 10 iteraciones empleando las funciones *fitensemble* y *crossval* pertenecientes a los algoritmos de regresión y clasificación de *Statistics and Machine Learning Toolbox*TM. Finalmente, se generan los estimados para cada registro de ambas bases de datos (*ADNI* y *Caldas*) empleando la función *kfoldPredict* perteneciente a los algoritmos de clasificación de *Statistics and Machine Learning Toolbox*TM.

4.1.16.2 Matrices de confusión

En las ilustraciones 125 y 126 se muestran las matrices de confusión obtenidas para el clasificador *RUSBoost* sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se

deduce que el clasificador logró un porcentaje de acierto del 0.88% y 0.97% en las respectivas predicciones. Asimismo, se puede observar que existe un bajo grado de confusión en las estimaciones de la técnica de aprendizaje de máquina respecto a las clases de predicción.

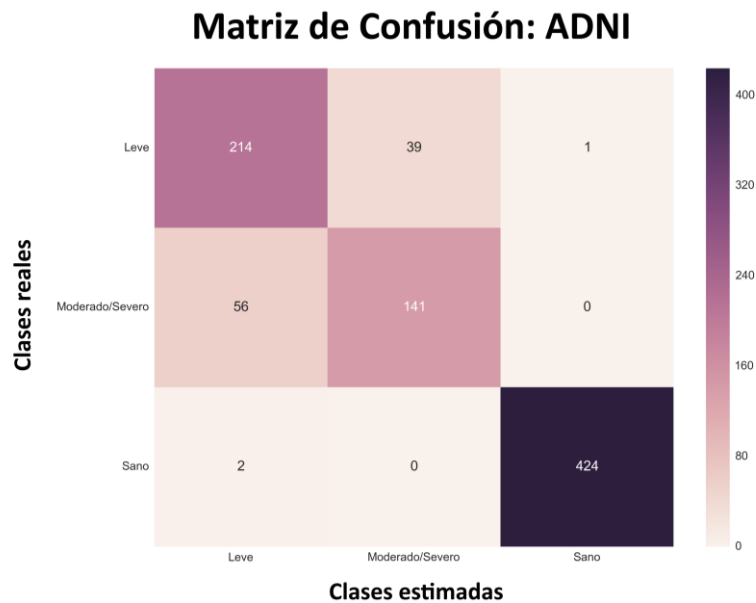


Ilustración 125 Matriz de confusión referente a los estimados del RUSBoost de árboles de decisión sobre la base de datos de la ADNI.

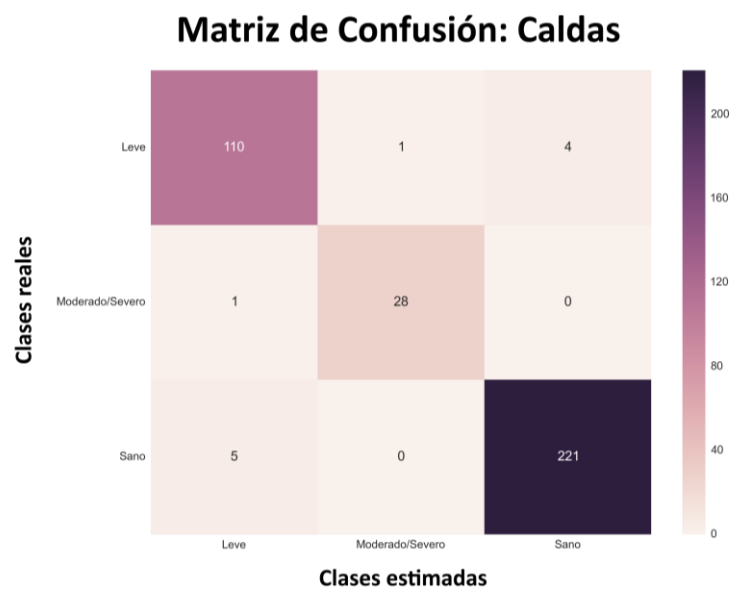


Ilustración 126 Matriz de confusión referente a los estimados del RUSBoost de árboles de decisión sobre la base de datos de Caldas.

4.1.16.3 Curvas ROC

En las ilustraciones 127 a 132 se muestran las curvas ROC obtenidas para el clasificador *RUSBoost* sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se observa que el AUC para cada una de las clases se encuentra bastante cercana de 1, lo cual indica que la técnica de aprendizaje de máquina no presenta mayores dificultades para elegir la clase correcta al momento de realizar las predicciones.

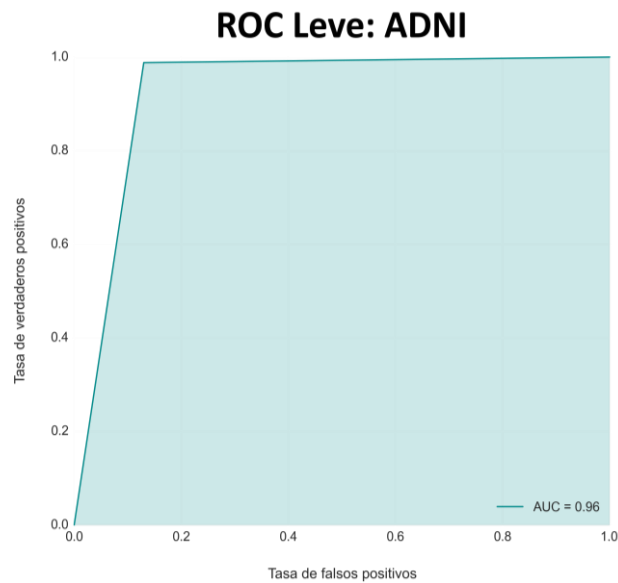


Ilustración 127 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del RUSBoost de árboles de decisión sobre la base de datos de la ADNI.

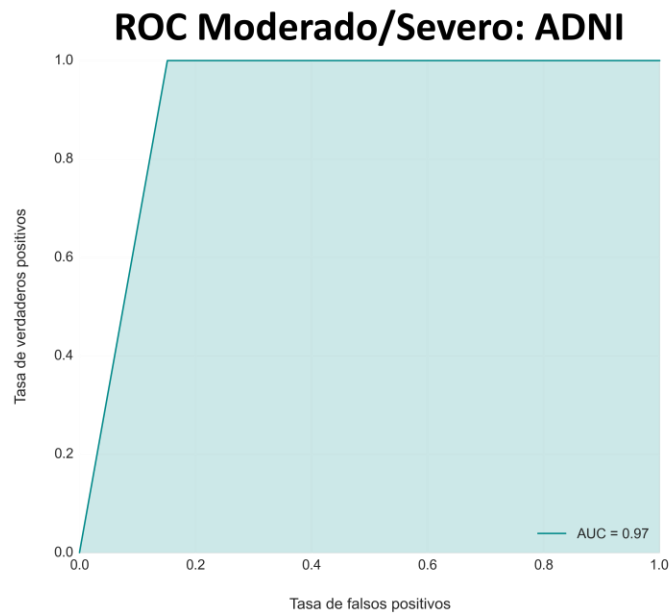


Ilustración 128 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del RUSBoost de árboles de decisión sobre la base de datos de la ADNI.

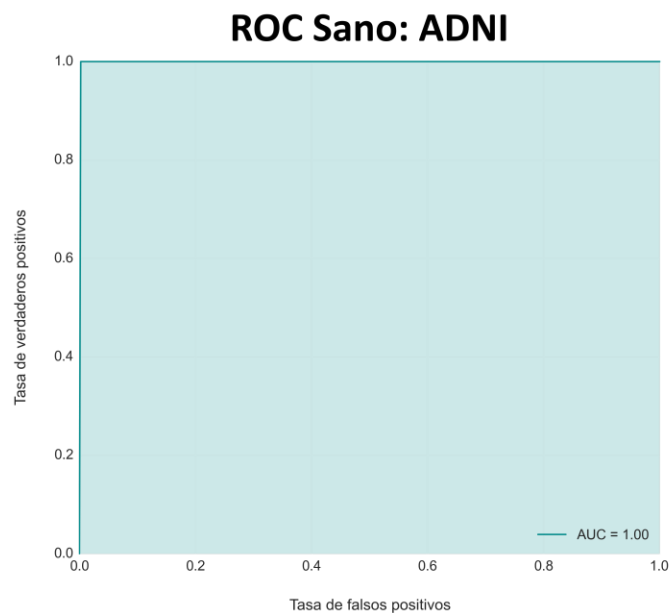


Ilustración 129 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del RUSBoost de árboles de decisión sobre la base de datos de la ADNI.

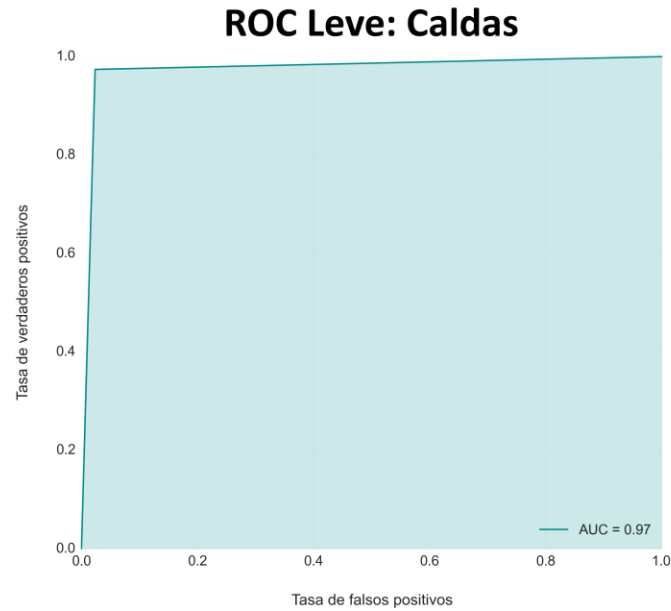


Ilustración 130 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del RUSBoost de árboles de decisión sobre la base de datos de Caldas.

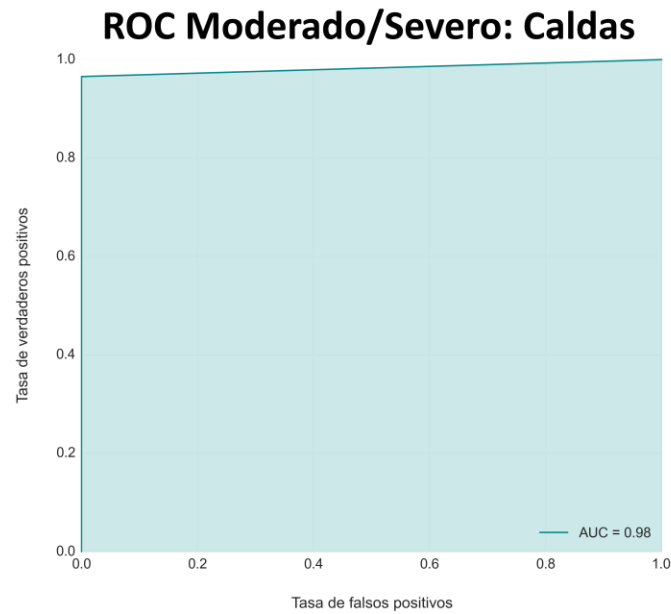


Ilustración 131 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del RUSBoost de árboles de decisión sobre la base de datos de Caldas.

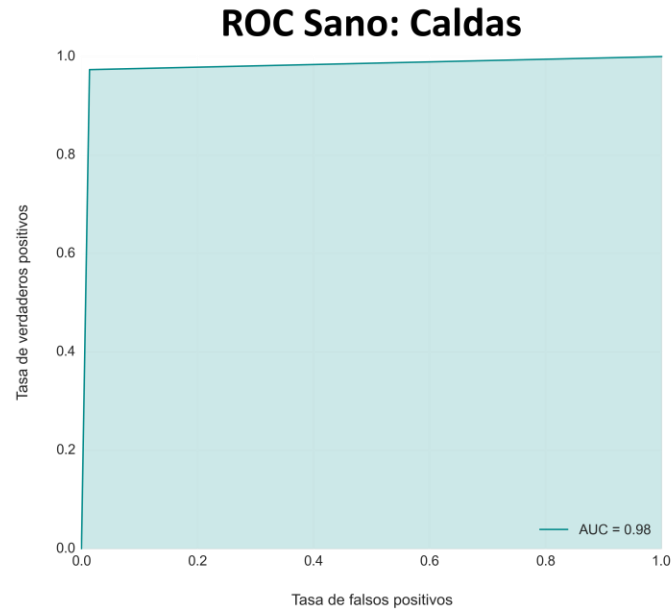


Ilustración 132 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del RUSBoost de árboles de decisión sobre la base de datos de Caldas.

4.1.16.4 Métricas de desempeño

A continuación se muestran las métricas de desempeño calculadas a partir de las matrices de confusión y de las curvas ROC para el *RUSBoost* de árboles de decisión.

ADNI		Caldas	
Exactitud	0,8883	Exactitud	0,9703
Precisión	0,8884	Precisión	0,9704
TPR (Sensibilidad)	0,8883	TPR (Sensibilidad)	0,9703
Especificidad	0,9591	Especificidad	0,9755
AUC Leve	0,9647	AUC Leve	0,9728
AUC Moderado/Severo	0,9659	AUC Moderado/Severo	0,9787
AUC Sano	0,9981	AUC Sano	0,9797
J Leve	0,7494	J Leve	0,9330
J Moderado/Severo	0,6584	J Moderado/Severo	0,9626
J Sano	0,9931	J Sano	0,9501
Kappa	0,8223	Kappa	0,9434

Tabla 18 Métricas de desempeño para el RUSBoost de árboles de decisión sobre las bases de datos de la ADNI y Caldas.

4.1.17 Técnica 16: *Gradient Boosting*

Gradient boosting es una generalización de la técnica de *boosting* a funciones de pérdida arbitrarias. Es un procedimiento preciso y efectivo que puede ser usado tanto para problemas de regresión como de clasificación (Scikit-learn, 2015).

El algoritmo de *gradient boosting* se fundamenta en la minimización de una función de pérdida esperada E_y respecto a un conjunto de datos $(x, y)_{i=1}^N$. En la práctica, dada una función de pérdida $\Psi(y, f)$ y/o un modelo personalizado $h(x, \theta)$, la solución a las estimaciones de los parámetros puede ser difícil de obtener. Para lidiar con esto, se propuso la elección de una nueva función $h(x, \theta_t)$ que fuera la más paralela al gradiente negativo $\{g_t(x_i)\}_{i=1}^N$ a lo largo de los datos observados (Natekin & Knoll, 2013), esto es:

$$g_t(x) = E_y \left[\frac{\partial \Psi(y, f)}{\partial f(x)} | x \right]_{f(x) = \hat{f}^{t-1}(x)}$$

En lugar de buscar una solución general para el incremento del impulso en el espacio funcional, simplemente se puede elegir el incremento que sea el más correlacionado con $-g_t(x)$. Esto permite el reemplazo de una tarea de optimización potencialmente compleja con una minimización de mínimos cuadrados (Natekin & Knoll, 2013):

$$(p_t, \theta_t) = \arg \min_{\rho, \theta} \sum_{i=1}^N [-g_t(x_i) + \rho h(x_i, \theta)]^2$$

4.1.17.1 Construcción, entrenamiento y clasificación

El clasificador *gradient boosting* se construye empleando la clase *GradientBoostingClassifier* perteneciente a los modelos de ensamble de *Scikit-learn*. El entrenamiento se realiza bajo la técnica de validación cruzada con 10 iteraciones empleando la función *cross_val_score* perteneciente al módulo *cross-validation* de *Scikit-learn*. Finalmente, se generan los estimados para cada registro de ambas bases de datos (ADNI y Caldas) empleando la función *cross_val_predict* perteneciente al módulo *cross-validation* de *Scikit-learn*.

4.1.17.2 Matrices de confusión

En las ilustraciones 133 y 134 se muestran las matrices de confusión obtenidas para el clasificador *gradient boosting* sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se deduce que el clasificador logró un porcentaje de acierto del 0.89% y 0.97% en las respectivas predicciones. Asimismo, se puede observar que existe un bajo grado de confusión en las estimaciones de la técnica de aprendizaje de máquina respecto a las clases de predicción.

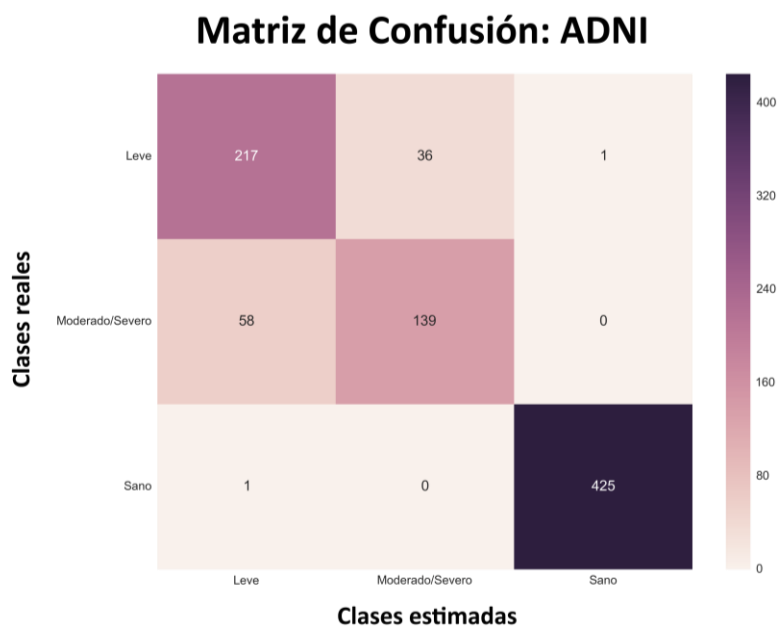


Ilustración 133 Matriz de confusión referente a los estimados del clasificador gradient boosting sobre la base de datos de la ADNI.

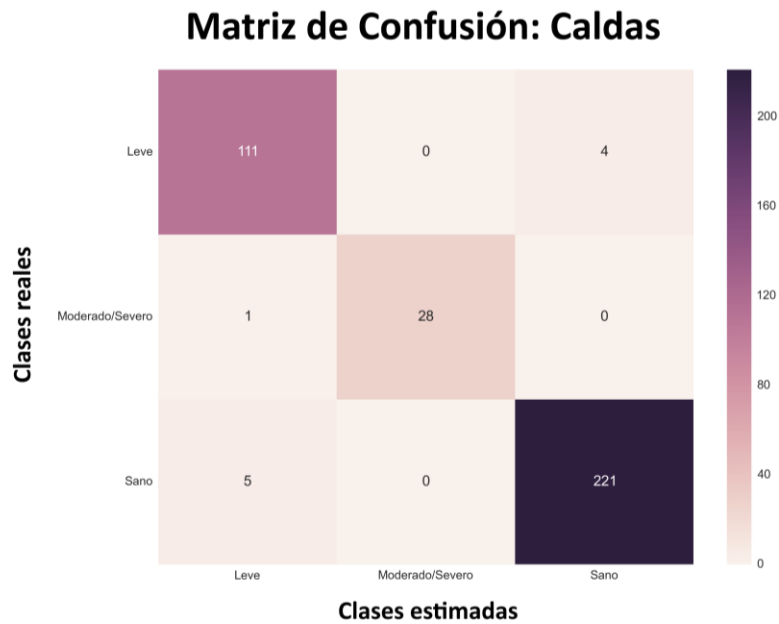


Ilustración 134 Matriz de confusión referente a los estimados del clasificador gradient boosting sobre la base de datos de Caldas.

4.1.17.3 Curvas ROC

En las ilustraciones 135 a 140 se muestran las curvas ROC obtenidas para el clasificador *gradient boosting* sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se observa que el AUC para cada una de las clases se encuentra bastante cercana de 1, lo cual indica que la técnica de aprendizaje de máquina no presenta mayores dificultades para elegir la clase correcta al momento de realizar las predicciones. Solamente se presentan algunos inconvenientes menores para diferenciar las clases «Leve» y «Moderado/Severo» de las demás en la base de datos de la *ADNI*.

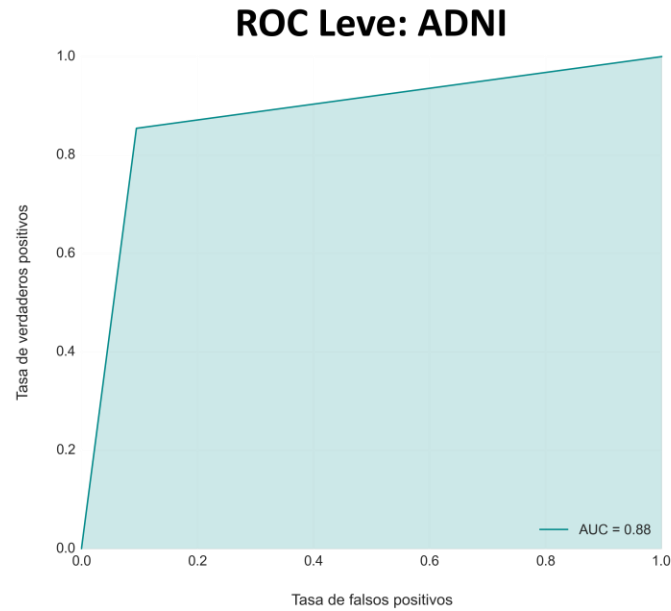


Ilustración 135 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del clasificador gradient boosting sobre la base de datos de la ADNI.

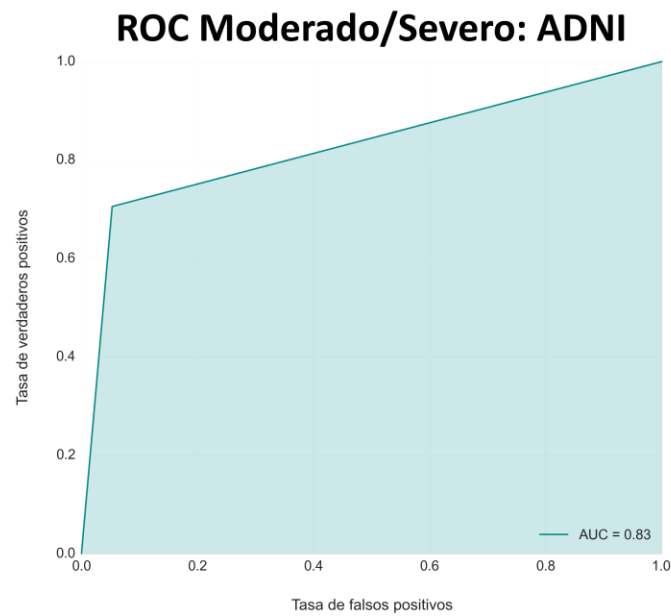


Ilustración 136 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del clasificador gradient boosting sobre la base de datos de la ADNI.

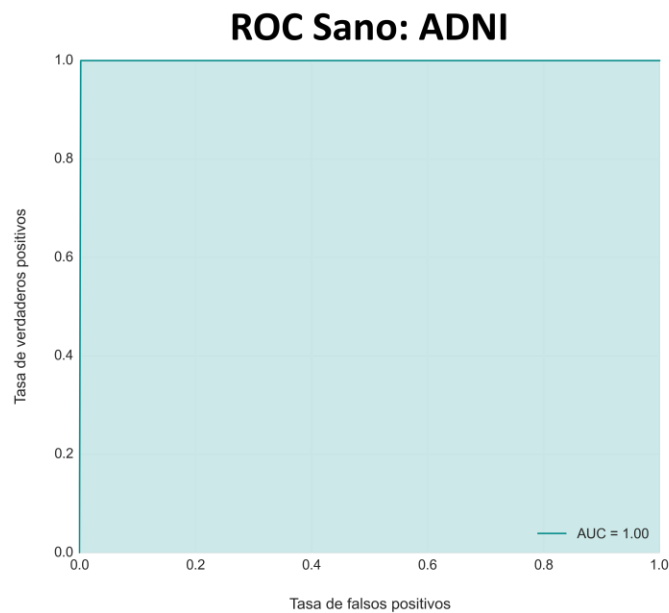


Ilustración 137 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del clasificador gradient boosting sobre la base de datos de la ADNI.

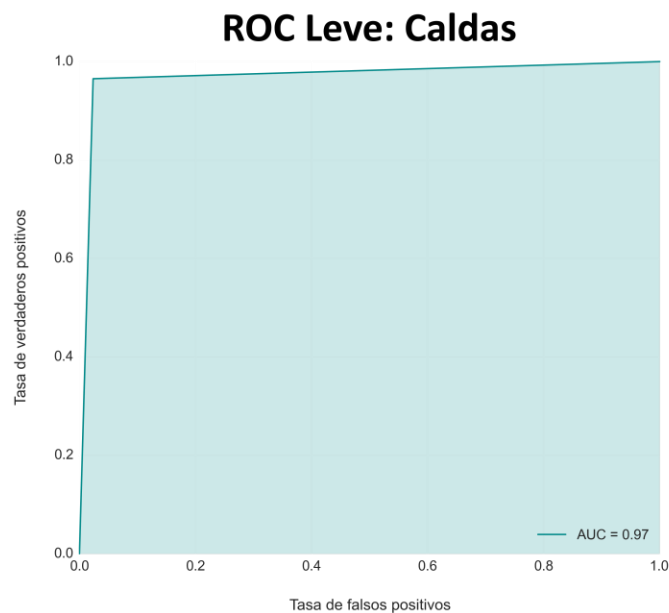


Ilustración 138 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados del clasificador gradient boosting sobre la base de datos de Caldas.

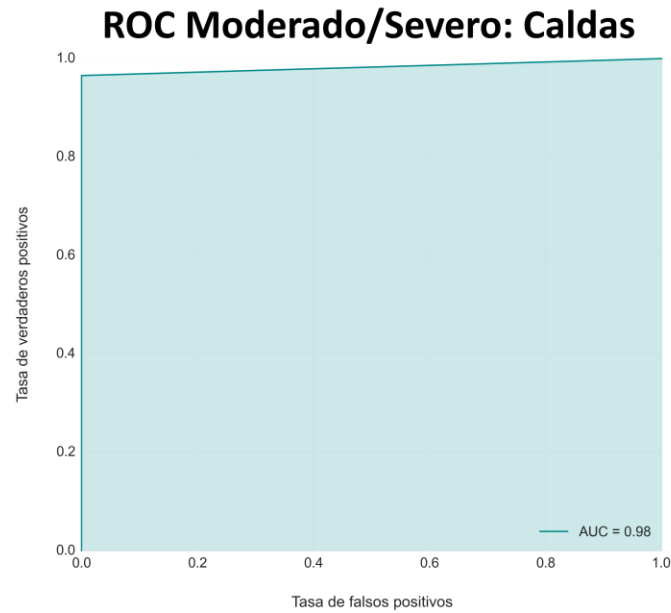


Ilustración 139 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados del clasificador gradient boosting sobre la base de datos de Caldas.

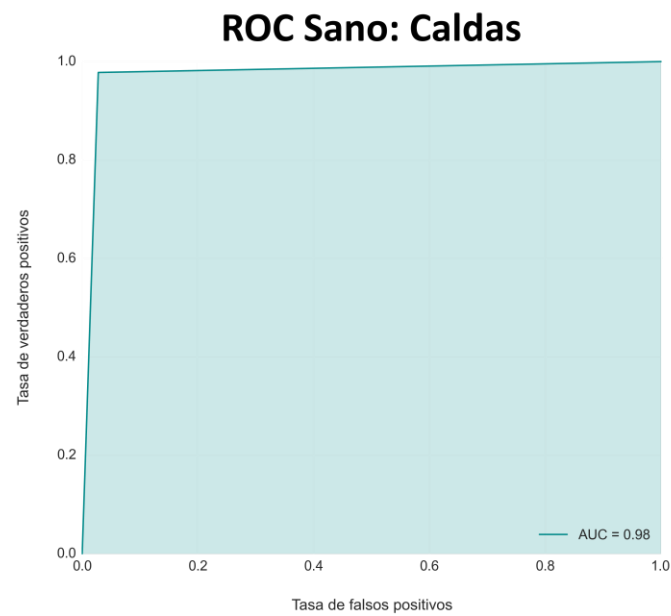


Ilustración 140 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados del clasificador gradient boosting sobre la base de datos de Caldas.

4.1.17.4 Métricas de desempeño

A continuación se muestran las métricas de desempeño calculadas a partir de las matrices de confusión y de las curvas ROC para el clasificador *gradient boosting*.

ADNI		Caldas	
Exactitud	0,8905	Exactitud	0,9730
Precisión	0,8907	Precisión	0,9732
TPR (Sensibilidad)	0,8905	TPR (Sensibilidad)	0,9730
Especificidad	0,9596	Especificidad	0,9757
AUC Leve	0,8798	AUC Leve	0,9708
AUC Moderado/Severo	0,8263	AUC Moderado/Severo	0,9828
AUC Sano	0,9977	AUC Sano	0,9750
J Leve	0,7596	J Leve	0,9417
J Moderado/Severo	0,6526	J Moderado/Severo	0,9655
J Sano	0,9954	J Sano	0,9501
Kappa	0,8257	Kappa	0,9485

Tabla 19 Métricas de desempeño para el clasificador *gradient boosting* sobre las bases de datos de la ADNI y Caldas.

4.1.18 Técnica 17: ANN - ELU Softmax Adam

Una red neuronal artificial (ANN) es un modelo matemático que pretende simular la estructura y las funcionalidades de las redes neuronales biológicas (Abraham, 2005). El bloque de construcción básico para cualquier ANN es la neurona artificial, esto es, una función matemática simple. Dicha función posee tres conjuntos de reglas básicas: multiplicación, suma y activación (ver ilustración 141). Al inicio de la neurona artificial las entradas se ponderan, lo cual significa que cada valor de entrada es multiplicado por un peso individual. En la sección central se suman todas las entradas ponderadas y los sesgos de la neurona artificial. A la salida, las entradas ponderadas y los sesgos sumados previamente se pasan a través de la función de activación —también llamada función de transferencia— (Krenker, Bešter, & Kos, 2011).

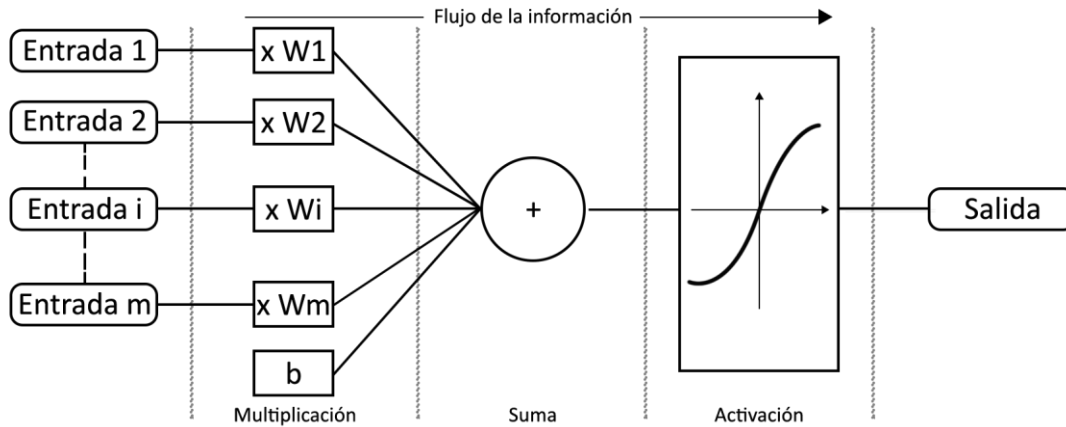


Ilustración 141 Principio de funcionamiento de una neurona artificial. Esta se encarga de sumar cada una de las entradas ponderadas y posteriormente pasa dicho resultado por una función de transferencia (Krenker, Bešter, & Kos, 2011).

A continuación se describen algunas funciones de activación que han sido empleadas comúnmente en la construcción de neuronas artificiales:

Función	Definición
$linear(x)$	$\varphi(x) = x$
$tanh(x)$	$\varphi(x) = \tanh(x)$
$sigmoid(x)$	$\varphi(x) = \frac{1}{1 + e^{-x}}$
$hard_sigmoid(x)$	$\varphi(x) = \max(\min(0.25x + 0.5, 1), 0)$
$ReLU(x)$	$\varphi(x) = \max(0, x)$
$PReLU(x, W)$	$\varphi(x) = \max(x, W * x)$
$SReLU(x)$	$\varphi(x) = \begin{cases} t_i^r + a_i^r(x_i - t_i^r), & x_i \geq t_i^r \\ x_i, & t_i^r > x_i > t_i^l \\ t_i^l + a_i^l(x_i - t_i^l), & x_i \leq t_i^l \end{cases}$
$ELU(x, \alpha)$	$\varphi(x) = \begin{cases} x, & x \geq 0 \\ \alpha(e^x - 1), & x < 0 \end{cases}$
$softmax(x, K)$	$\varphi(x)_j = \frac{e^{x_j}}{\sum_{k=1}^K e^{x_k}}$
$softplus(x)$	$\varphi(x) = \log(1 + e^x)$

Tabla 20 Definición matemática de las funciones de activación comúnmente empleadas en la construcción de neuronas artificiales (Lasagne, 2015; Chainer, 2015; Jin et al., 2016).

Aunque los principios de funcionamiento de una neurona artificial son bastante simples, el verdadero potencial y poder de cálculo de estos modelos cobran vida cuando se empiezan a interconectar en capas de redes artificiales (ver ilustración 142). Dichas redes se fundamentan en el hecho de que la complejidad puede surgir simplemente a partir de reglas básicas y sencillas (Krenker, Bešter, & Kos, 2011).

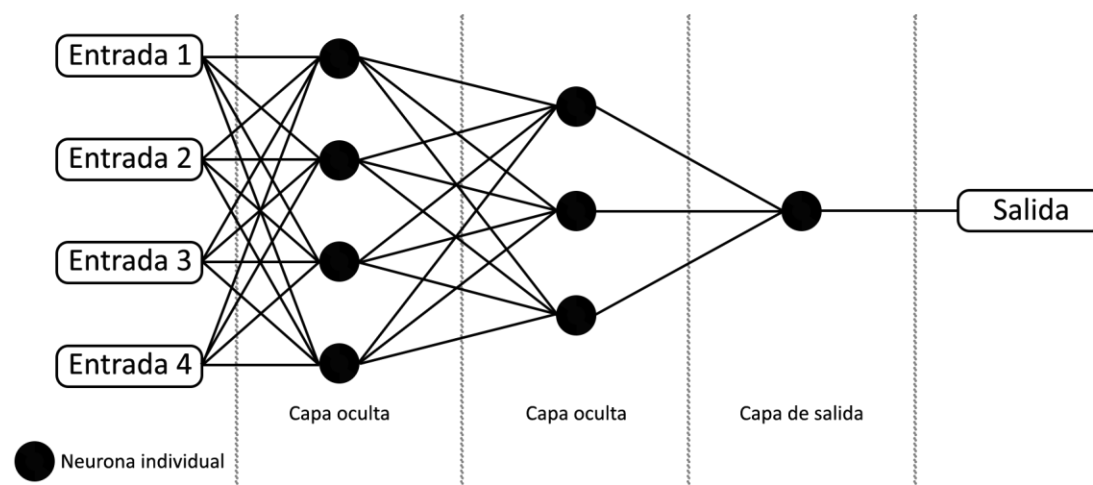


Ilustración 142 Ejemplo de una red neuronal artificial de cuatro entradas, una salida y tres capas de neuronas artificiales —dos capas ocultas y una capa de salida—. Cada neurona artificial de cierta capa está interconectada con todas las neuronas artificiales de la siguiente capa (Krenker, Bešter, & Kos, 2011).

El entrenamiento de una red neuronal artificial consta de dos pasadas: una pasada hacia adelante y una pasada hacia atrás. En la primera, se computan las salidas de la red y se comparan con las salidas esperadas. Asimismo, se calcula el error para todas las salidas. En la segunda, el error es utilizado para alterar los pesos de la red neuronal artificial con el fin de reducir el tamaño del mismo. Las pasadas hacia adelante y hacia atrás son repetidas hasta que el error sea lo suficientemente bajo. Este método de entrenamiento es frecuentemente llamado *backpropagation* o propagación hacia atrás (Cilimkovic, 2015).

La alteración de los pesos durante la propagación hacia atrás se efectúa mediante la aplicación de un algoritmo de gradiente descendente. Este último permite minimizar una función objetivo $J(\theta)$ por medio de la actualización de los parámetros $\theta \in \mathbb{R}^d$ en la dirección opuesta al gradiente de la función objetivo $\nabla_{\theta} J(\theta)$ (Ruder, 2016).

Existen tres variantes del gradiente descendiente (Ruder, 2016):

- *Batch gradient descendant*: computa el gradiente de la función objetivo a lo largo de todo el set de datos. Esto es $\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta)$.
- *Stochastic gradient descendant*: realiza una actualización de parámetros para cada instancia $x^{(i)}$ con etiqueta $y^{(i)}$. Esto es $\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta; x^{(i)}; y^{(i)})$.
- *Mini-batch gradient descendant*: realiza una actualización para cada mini-lote de n instancias. Esto es $\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta; x^{(i:i+1)}; y^{(i:i+1)})$.

En las tres variantes del gradiente descendiente la tasa de aprendizaje η determina el tamaño de los pasos que se siguen para llegar a un mínimo (local). En otras palabras, se sigue cuesta abajo la dirección de la pendiente de la superficie creada por la función objetivo hasta llegar a un valle (Ruder, 2016).

Aun así, el gradient descendiente posee diversos inconvenientes en términos de parametrización, tiempo de procesamiento y convergencia. (Darken, Chang, & Moody, 1992; Dauphin et al., 2014; Ruder, 2016). Para superar dichos problemas, algunos investigadores han propuesto alternativas que permiten optimizar su funcionamiento:

- *Adagrad*: es un algoritmo de optimización basado en gradientes que adapta la tasa de aprendizaje a los parámetros, realizando actualizaciones más grandes para los parámetros infrecuentes y más pequeñas para los parámetros frecuentes. Por esta razón, es bastante efectivo en el tratamiento de datos dispersos (Duchi, Hazan, & Singer, 2011).

Adagrad modifica la tasa de aprendizaje η en un paso de tiempo t para cada parámetro θ_i , basado en los gradientes pasados que han sido calculados para θ_i :

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} \cdot g_{t,i}$$

$G_t \in \mathbb{R}^{d \times d}$ es una matriz donde cada elemento de la diagonal i, i contiene la suma de los cuadrados de los gradientes con respecto a θ_i hasta un paso de tiempo t ; $g_{t,i}$ es el gradiente de la función objetivo; y ϵ es un término de suavizado que evita la división por cero (Ruder, 2016).

Como G_t contiene en su diagonal la suma de los cuadrados de los gradientes pasados con respecto a todos los parámetros θ , se puede expresar la regla de actualización *Adagrad* en términos de un producto Hadamard $\odot$ entre G_t y g_t :

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t + \epsilon}} \odot g_t$$

- *Adadelta*: es una extensión de *Adagrad* que busca reducir su agresividad, decreciendo monótonamente la tasa de aprendizaje. En lugar de almacenar de manera ineficiente los cuadrados de los gradientes previos, la suma de los gradientes es definida recursivamente como un promedio atenuado de todos los cuadrados de los gradientes anteriores (Zeiler, 2012).

Para *Adadelta*, primero se reescribe la fórmula de *Adagrad* en términos del vector de actualización de parámetros $\Delta\theta_t$:

$$\Delta\theta_t = -\frac{\eta}{\sqrt{G_t + \epsilon}} \odot g_t$$

A continuación, se reemplaza la matriz diagonal G_t con el promedio atenuado de todos los cuadrados de los gradientes anteriores $E[g^2]_t = \gamma E[g^2]_{t-1} + (1 - \gamma)g_t^2$ donde γ es una constante de atenuación:

$$\Delta\theta_t = -\frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} \odot g_t$$

Como el denominador cumple con el criterio de valor cuadrático medio (RMS, *Root Mean Square*) del gradiente, se puede realizar el respectivo reemplazo de forma inmediata:

$$\Delta\theta_t = -\frac{\eta}{RMS[g]_t} \odot g_t$$

Finalmente, la regla de actualización *Adadelta* es escrita en términos de una aproximación Hessiana a fin de coincidir las unidades de los parámetros con las unidades de la actualización. Lo anterior se realiza mediante la sustitución de la tasa de aprendizaje η por el RMS exponencialmente atenuado de los $\Delta\theta$ previos:

$$\Delta\theta_t = -\frac{RMS[\Delta\theta]_{t-1}}{RMS[g]_t} \odot g_t$$

- *Adam*: es un método que computa tasas de aprendizaje adaptables para cada parámetro. Además de almacenar un promedio exponencialmente atenuado de los cuadrados de los gradientes anteriores como en *Adadelta*, *Adam* también guarda un promedio exponencialmente atenuado de los gradientes anteriores (Kingma & Ba, 2015):

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

m_t y v_t son estimaciones del primero momento (la media) y del segundo momento de los gradientes respectivamente. En cuanto m_t y v_t se inicializan como vectores de ceros se observa que están sesgados hacia cero, especialmente durante los primeros pasos de tiempo y sobre todo, cuando las tasas de atenuación son pequeñas (Kingma & Ba, 2015).

Dicho sesgo es contrarrestado mediante el cómputo de las estimaciones de primer y segundo orden con corrección del mismo:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

Estas estimaciones se utilizan para actualizar los parámetros tal y como se observa en *Adagrad*, lo cual da origen a la regla de actualización *Adam*:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \odot \hat{m}_t$$

- *Adamax*: es una variante del algoritmo *Adam* basada en la norma infinita (Kingma & Ba, 2015). Empleando $v_t = \max(\beta_2 \cdot v_{t-1}, |g_t|)$, se establece la regla de actualización *Adamax* como:

$$\theta_{t+1} = \theta_t - \frac{\eta}{1 - \beta_1^t} \odot \frac{m_t}{v_t}$$

Aquí, $\eta / (1 - \beta_1^t)$ representa la tasa de aprendizaje con el término de corrección del sesgo para el primer momento (Kingma & Ba, 2015).

4.1.18.1 Construcción, entrenamiento y clasificación

La ANN se fundamenta en la estructura dispuesta en la ilustración 143. Esta se construye empleando la clase *Sequential* perteneciente a los modelos de *Keras*. Asimismo, se emplean las funciones de activación *ELU* y *Softmax* —pertenecientes a las funciones de activación de *Keras*— para construir las neuronas artificiales de cada capa. A continuación, se compila la red neuronal empleando la función *compile(loss='categorical_crossentropy', optimizer='adam',...)* perteneciente a la clase *Sequential* de los modelos de *Keras*.

El entrenamiento de la ANN se realiza bajo la técnica de validación cruzada con 10 iteraciones empleando las funciones *StratifiedKFold(n_folds=10,...)* y *fit* pertenecientes al módulo *cross-validation* de *Scikit-learn* y a la clase *Sequential* de los modelos de *Keras* respectivamente. Finalmente, se generan los estimados para cada registro de ambas bases de datos (*ADNI* y *Caldas*) empleando la función *predict_classes* perteneciente a la clase *Sequential* de los modelos de *Keras*.

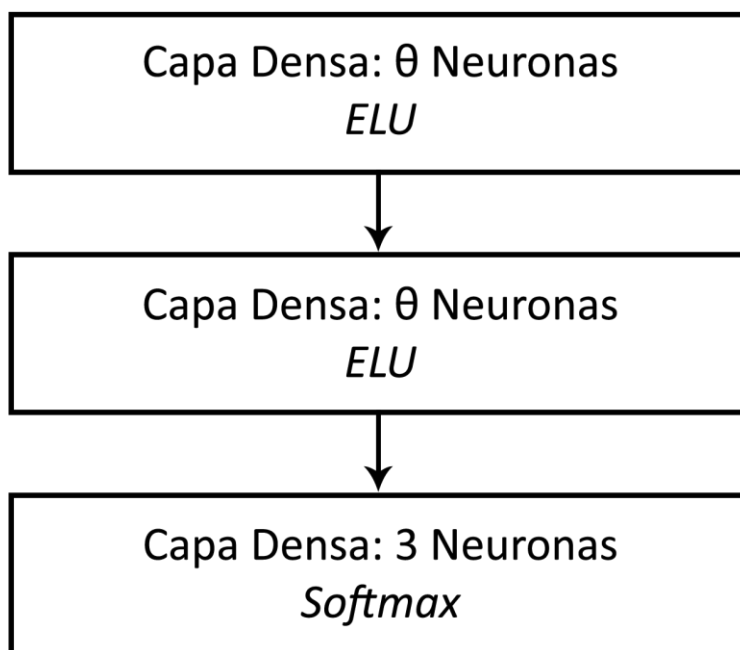


Ilustración 143 Estructura de la ANN – ELU Softmax Adam de tres capas: dos capas ocultas de neuronas artificiales con funciones de activación ELU y una capa de salida de neuronas artificiales con funciones de activación Softmax. La cantidad de neuronas θ es equivalente a la cantidad de entradas de la ANN.

4.1.18.2 Matrices de confusión

En las ilustraciones 144 y 145 se muestran las matrices de confusión obtenidas para la ANN – ELU Softmax Adam sobre las bases de datos de la ADNI y Caldas. A partir de estas se deduce que el clasificador logró un porcentaje de acierto del 0.90% y 0.92% en las respectivas predicciones. Asimismo, se puede observar que existe un bajo grado de confusión en las estimaciones de la técnica de aprendizaje de máquina respecto a las clases de predicción.

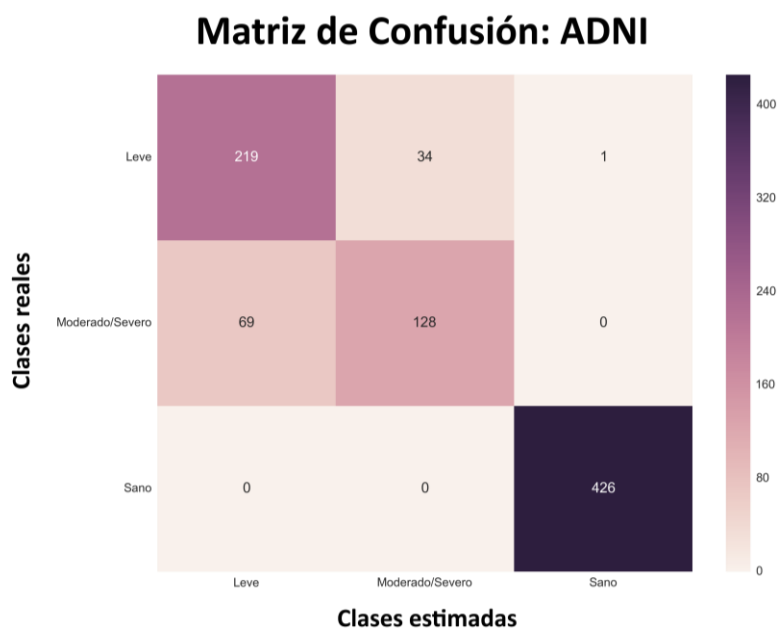


Ilustración 144 Matriz de confusión referente a los estimados de la ANN - ELU Softmax Adam sobre la base de datos de la ADNI.

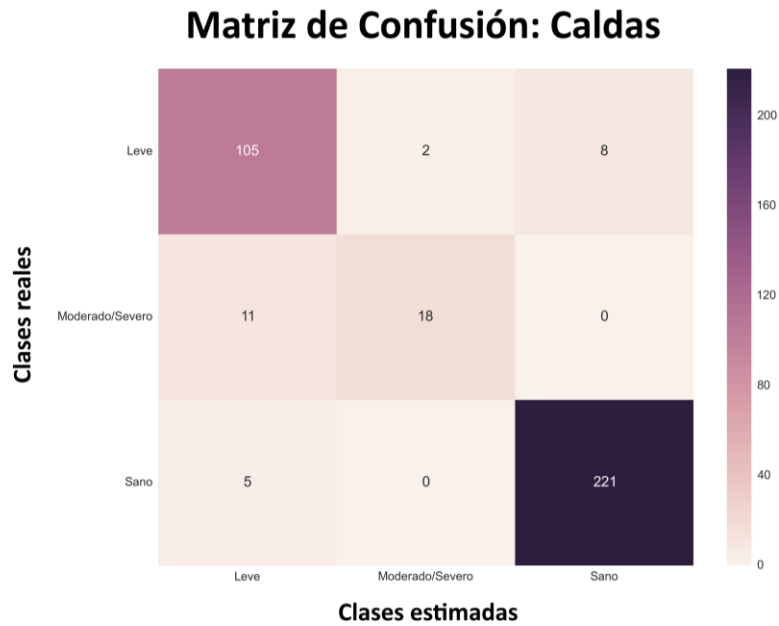


Ilustración 145 Matriz de confusión referente a los estimados de la ANN - ELU Softmax Adam sobre la base de datos de Caldas.

4.1.18.3 Curvas ROC

En las ilustraciones 146 a 151 se muestran las curvas ROC obtenidas para la ANN – ELU Softmax Adam sobre las bases de datos de la ADNI y Caldas. A partir de estas se observa que el AUC para cada una de las clases se encuentra bastante cercana de 1, lo cual indica que la técnica de aprendizaje de máquina no presenta mayores dificultades para elegir la clase correcta al momento de realizar las predicciones. Solamente se presentan algunos inconvenientes menores para diferenciar las clases «Leve» y «Moderado/Severo» de las demás en las bases de datos de la ADNI y Caldas.

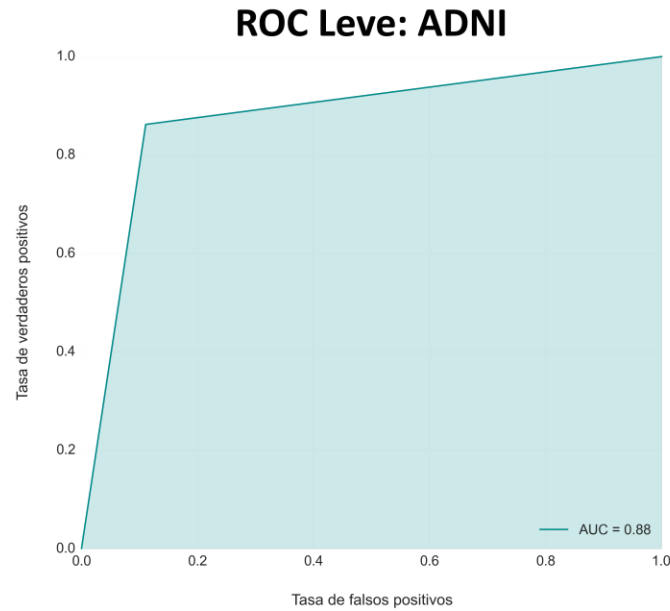


Ilustración 146 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - ELU Softmax Adam sobre la base de datos de la ADNI.

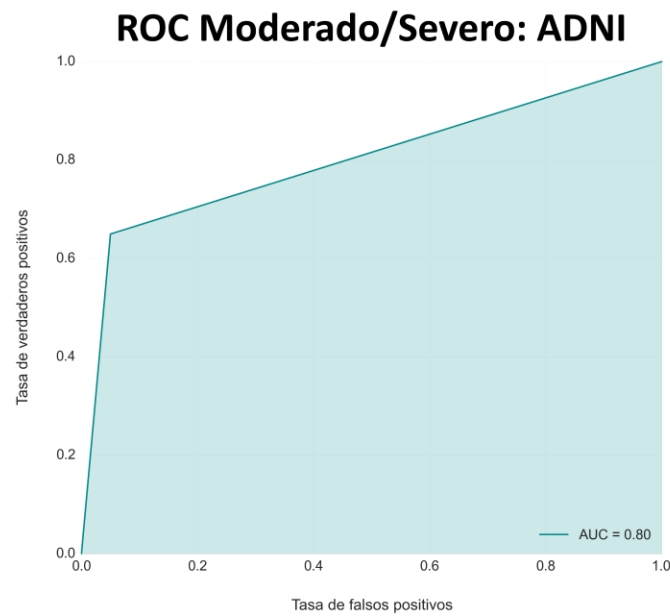


Ilustración 147 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - ELU Softmax Adam sobre la base de datos de la ADNI.

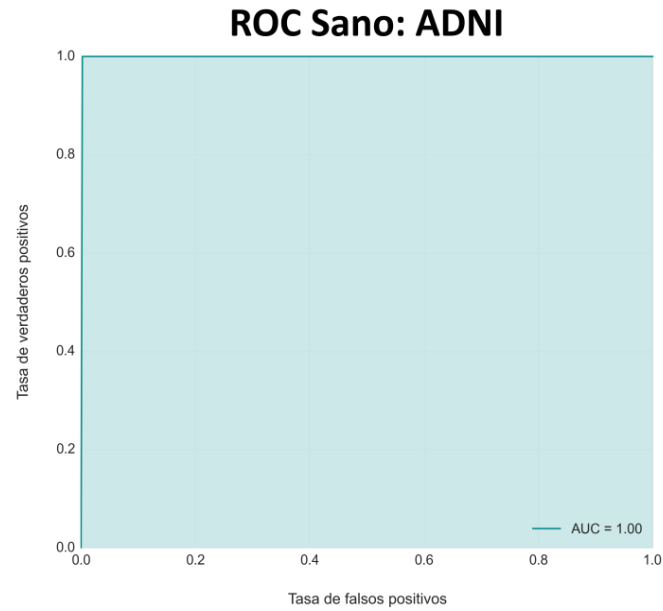


Ilustración 148 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - ELU Softmax Adam sobre la base de datos de la ADNI.

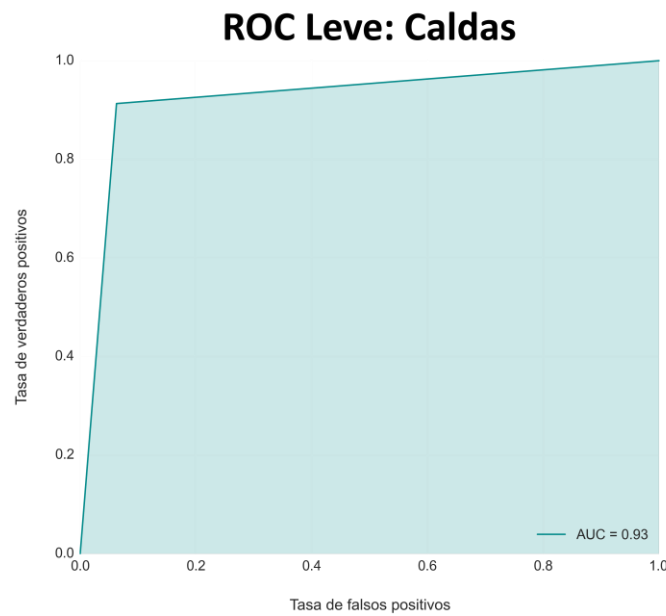


Ilustración 149 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - ELU Softmax Adam sobre la base de datos de Caldas.

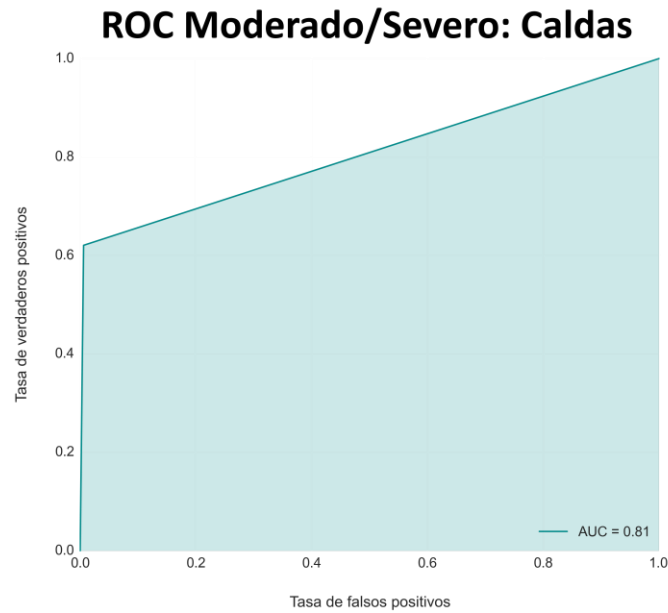


Ilustración 150 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - ELU Softmax Adam sobre la base de datos de Caldas.

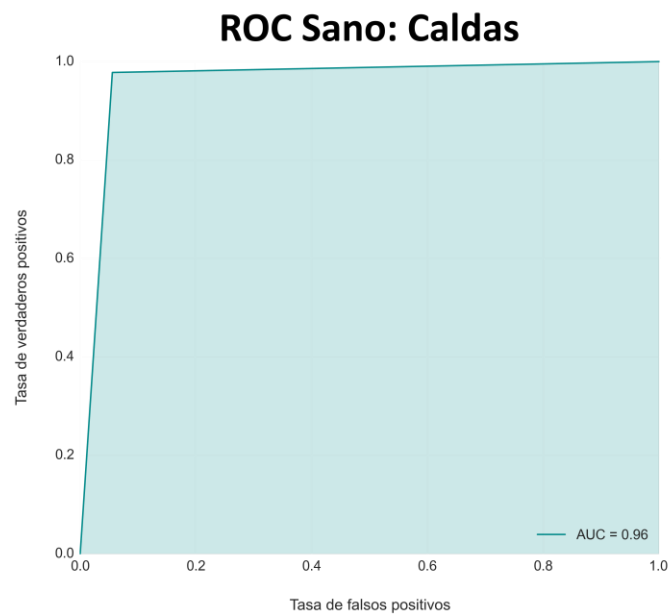


Ilustración 151 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - ELU Softmax Adam sobre la base de datos de Caldas.

4.1.18.4 Métricas de desempeño

A continuación se muestran las métricas de desempeño calculadas a partir de las matrices de confusión y de las curvas ROC para la ANN – ELU Softmax Adam.

ADNI		Caldas	
Exactitud	0,8814	Exactitud	0,9297
Precisión	0,8823	Precisión	0,9297
TPR (Sensibilidad)	0,8814	TPR (Sensibilidad)	0,9297
Especificidad	0,9556	Especificidad	0,9461
AUC Leve	0,8757	AUC Leve	0,9251
AUC Moderado/Severo	0,7999	AUC Moderado/Severo	0,8074
AUC Sano	0,9989	AUC Sano	0,9612
J Leve	0,7515	J Leve	0,8503
J Moderado/Severo	0,5997	J Moderado/Severo	0,6148
J Sano	0,9978	J Sano	0,9223
Kappa	0,8108	Kappa	0,8638

Tabla 21 Métricas de desempeño para la ANN – ELU Softmax Adam sobre las bases de datos de la ADNI y Caldas.

4.1.19 Técnica 18: ANN - Hard Sigmoid Softmax Adam

La explicación de los temas relacionados con redes neuronales artificiales se encuentra dispuesta en la sección «4.1.18 Técnica 17: ANN – ELU Softmax Adam».

4.1.19.1 Construcción, entrenamiento y clasificación

La ANN se fundamenta en la estructura dispuesta en la ilustración 152. Esta se construye empleando la clase *Sequential* perteneciente a los modelos de *Keras*. Asimismo, se emplean las funciones de activación *Hard_Sigmoid* y *Softmax* —pertenecientes a las funciones de activación de *Keras*— para construir las neuronas artificiales de cada capa. A continuación, se compila la red neuronal empleando la función

compile(loss='categorical_crossentropy', optimizer='adam',...) perteneciente a la clase *Sequential* de los modelos de *Keras*.

El entrenamiento de la ANN se realiza bajo la técnica de validación cruzada con 10 iteraciones empleando las funciones *StratifiedKFold(n_folds=10,...)* y *fit* pertenecientes al módulo *cross-validation* de *Scikit-learn* y a la clase *Sequential* de los modelos de *Keras* respectivamente. Finalmente, se generan los estimados para cada registro de ambas bases de datos (*ADNI* y *Caldas*) empleando la función *predict_classes* perteneciente a la clase *Sequential* de los modelos de *Keras*.

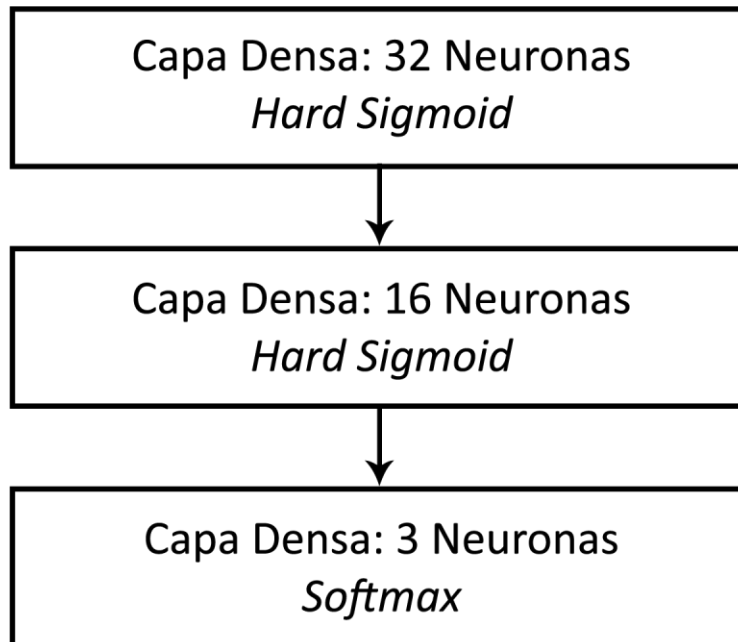


Ilustración 152 Estructura de la ANN – Hard Sigmoid Softmax Adam de tres capas: dos capas ocultas de neuronas artificiales con funciones de activación Sigmoidea Dura y una capa de salida de neuronas artificiales con funciones de activación Softmax.

4.1.19.2 Matrices de confusión

En las ilustraciones 153 y 154 se muestran las matrices de confusión obtenidas para la ANN – *Hard Sigmoid Softmax Adam* sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se deduce que el clasificador logró un porcentaje de acierto del 0.90% y 0.96% en

las respectivas predicciones. Asimismo, se puede observar que existe un bajo grado de confusión en las estimaciones de la técnica de aprendizaje de máquina respecto a las clases de predicción.

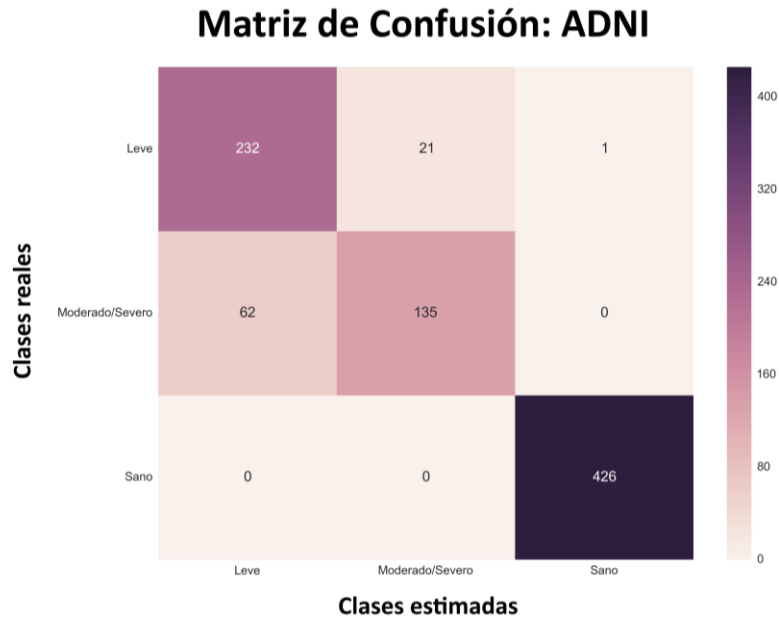


Ilustración 153 Matriz de confusión referente a los estimados de la ANN - Hard Sigmoid Softmax Adam sobre la base de datos de la ADNI.

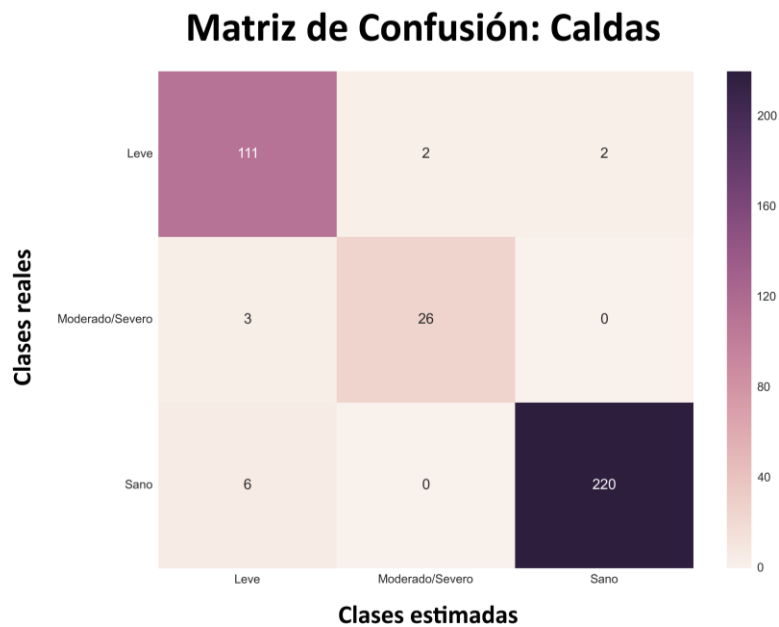


Ilustración 154 Matriz de confusión referente a los estimados de la ANN - Hard Sigmoid Softmax Adam sobre la base de datos de Caldas.

4.1.19.3 Curvas ROC

En las ilustraciones 155 a 160 se muestran las curvas ROC obtenidas para la *ANN – Hard Sigmoid Softmax Adam* sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se observa que el AUC para cada una de las clases se encuentra bastante cercana de 1, lo cual indica que la técnica de aprendizaje de máquina no presenta mayores dificultades para elegir la clase correcta al momento de realizar las predicciones. Solamente se presentan algunos inconvenientes menores para diferenciar las clases «Leve» y «Moderado/Severo» de las demás en la base de datos de la *ADNI*.

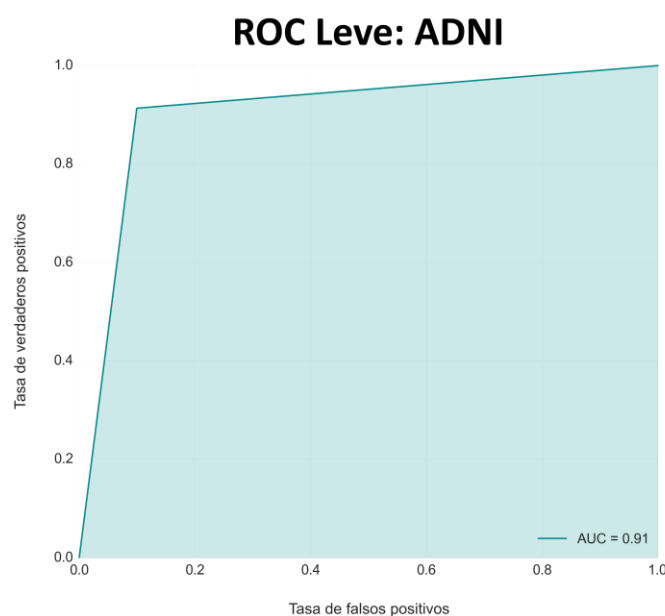


Ilustración 155 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - Hard Sigmoid Softmax Adam sobre la base de datos de la ADNI.

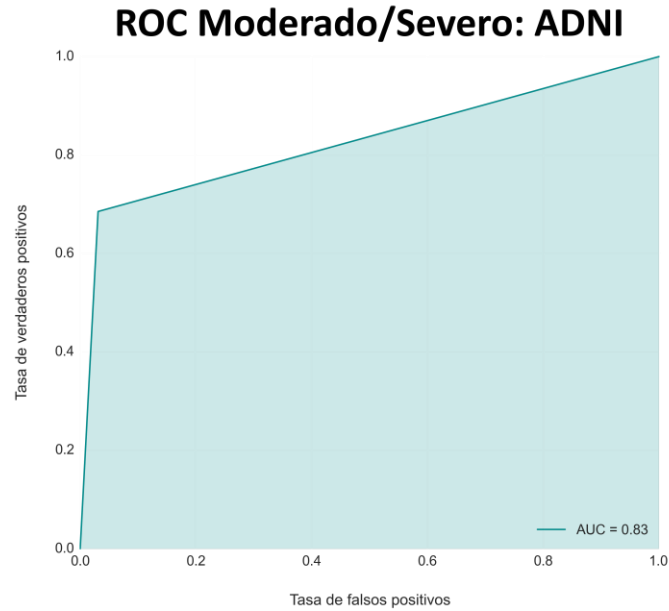


Ilustración 156 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - Hard Sigmoid Softmax Adam sobre la base de datos de la ADNI.

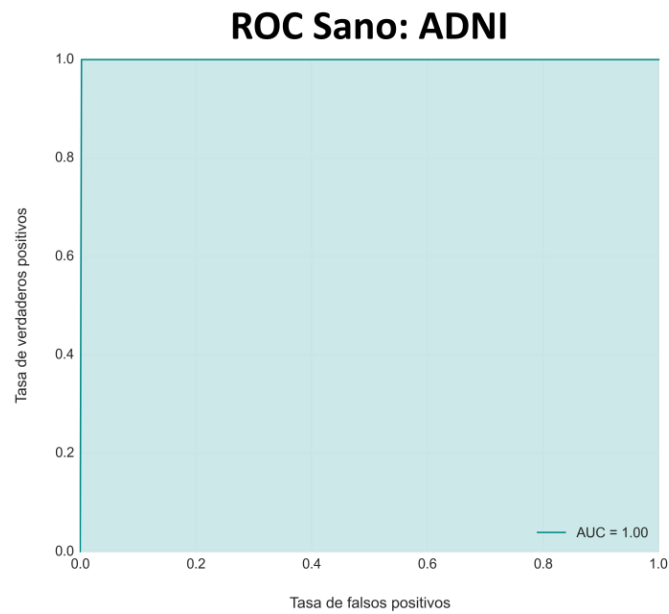


Ilustración 157 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - Hard Sigmoid Softmax Adam sobre la base de datos de la ADNI.

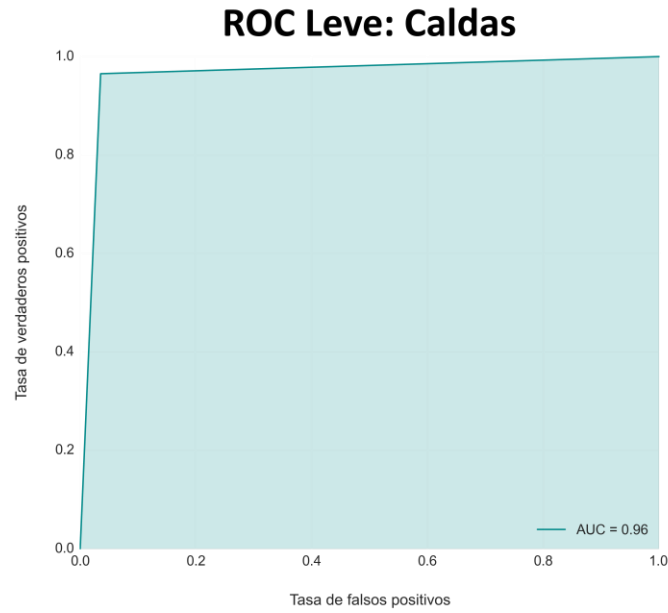


Ilustración 158 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - Hard Sigmoid Softmax Adam sobre la base de datos de Caldas.

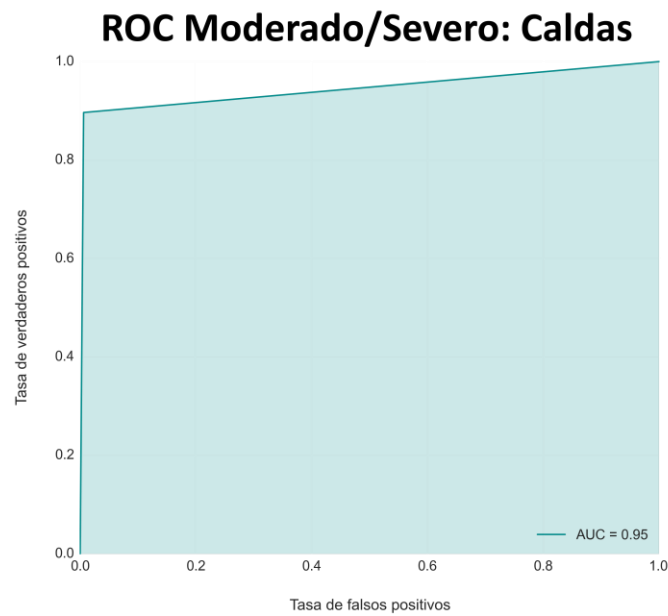


Ilustración 159 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - Hard Sigmoid Softmax Adam sobre la base de datos de Caldas.

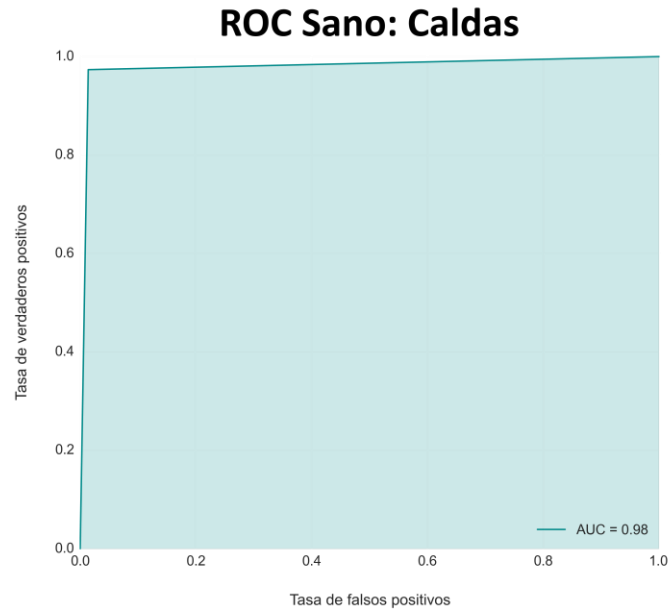


Ilustración 160 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - Hard Sigmoid Softmax Adam sobre la base de datos de Caldas.

4.1.19.4 Métricas de desempeño

A continuación se muestran las métricas de desempeño calculadas a partir de las matrices de confusión y de las curvas ROC para la ANN – *Hard Sigmoid Softmax Adam*.

ADNI		Caldas	
Exactitud	0,9042	Exactitud	0,9649
Precisión	0,9075	Precisión	0,9656
TPR (Sensibilidad)	0,9042	TPR (Sensibilidad)	0,9649
Especificidad	0,9632	Especificidad	0,9801
AUC Leve	0,9069	AUC Leve	0,9650
AUC Moderado/Severo	0,8272	AUC Moderado/Severo	0,9453
AUC Sano	0,9989	AUC Sano	0,9798
J Leve	0,8139	J Leve	0,9299
J Moderado/Severo	0,6544	J Moderado/Severo	0,8907
J Sano	0,9978	J Sano	0,9596
Kappa	0,8471	Kappa	0,9333

Tabla 22 Métricas de desempeño para la ANN – Hard Sigmoid Softmax Adam sobre las bases de datos de la ADNI y Caldas.

4.1.20 Técnica 19: *ANN - PReLU Softmax Adam*

La explicación de los temas relacionados con redes neuronales artificiales se encuentra dispuesta en la sección «4.1.18 Técnica 17: ANN – ELU Softmax Adam».

4.1.20.1 *Construcción, entrenamiento y clasificación*

La ANN se fundamenta en la estructura dispuesta en la ilustración 161. Esta se construye empleando la clase *Sequential* perteneciente a los modelos de *Keras*. Asimismo, se emplean las funciones de activación *PReLU* y *Softmax* —pertenecientes a las funciones de activación de *Keras*— para construir las neuronas artificiales de cada capa. A continuación, se compila la red neuronal empleando la función *compile(loss='categorical_crossentropy', optimizer='adam',...)* perteneciente a la clase *Sequential* de los modelos de *Keras*.

El entrenamiento de la ANN se realiza bajo la técnica de validación cruzada con 10 iteraciones empleando las funciones *StratifiedKFold(n_folds=10,...)* y *fit* pertenecientes al módulo *cross-validation* de *Scikit-learn* y a la clase *Sequential* de los modelos de *Keras* respectivamente. Finalmente, se generan los estimados para cada registro de ambas bases de datos (*ADNI* y *Caldas*) empleando la función *predict_classes* perteneciente a la clase *Sequential* de los modelos de *Keras*.

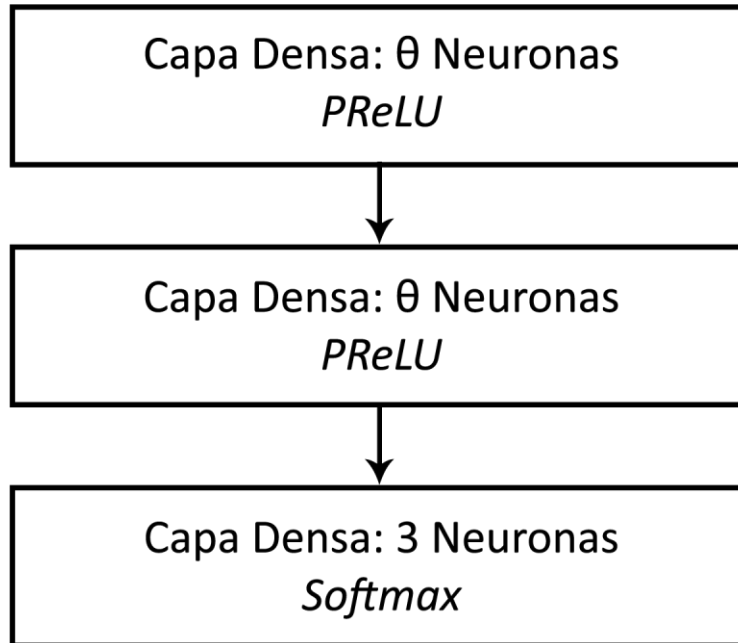


Ilustración 161 Estructura de la ANN – PReLU Softmax Adam de tres capas: dos capas ocultas de neuronas artificiales con funciones de activación PReLU y una capa de salida de neuronas artificiales con funciones de activación Softmax. La cantidad de neuronas θ es equivalente a la cantidad de entradas de la ANN.

4.1.20.2 Matrices de confusión

En las ilustraciones 162 y 163 se muestran las matrices de confusión obtenidas para la ANN – PReLU Softmax Adam sobre las bases de datos de la ADNI y Caldas. A partir de estas se deduce que el clasificador logró un porcentaje de acierto del 0.88% y 0.92% en las respectivas predicciones. Asimismo, se puede observar que existe un bajo grado de confusión en las estimaciones de la técnica de aprendizaje de máquina respecto a las clases de predicción.

Matriz de Confusión: Caldas

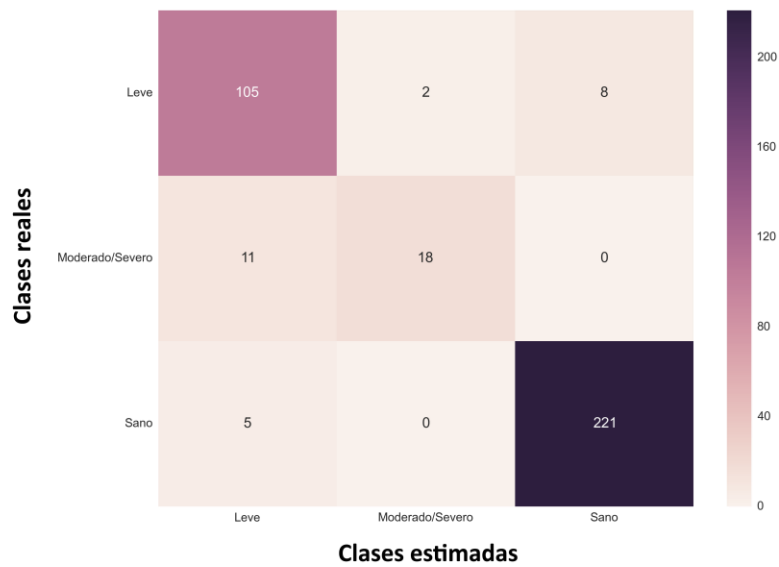


Ilustración 162 Matriz de confusión referente a los estimados de la ANN - PReLU Softmax Adam sobre la base de datos de la ADNI.

Matriz de Confusión: Caldas

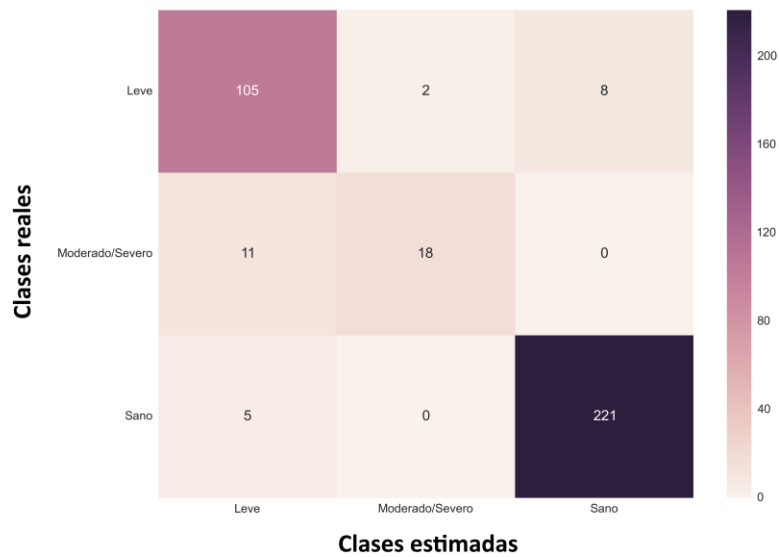


Ilustración 163 Matriz de confusión referente a los estimados de la ANN - PReLU Softmax Adam sobre la base de datos de Caldas.

4.1.20.3 Curvas ROC

En las ilustraciones 164 a 169 se muestran las curvas ROC obtenidas para la *ANN – PReLU Softmax Adam* sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se observa que el AUC para cada una de las clases se encuentra bastante cercana de 1, lo cual indica que la técnica de aprendizaje de máquina no presenta mayores dificultades para elegir la clase correcta al momento de realizar las predicciones. Solamente se presentan algunos inconvenientes menores para diferenciar las clases «Leve» y «Moderado/Severo» de las demás en las bases de datos de la *ADNI* y *Caldas*.

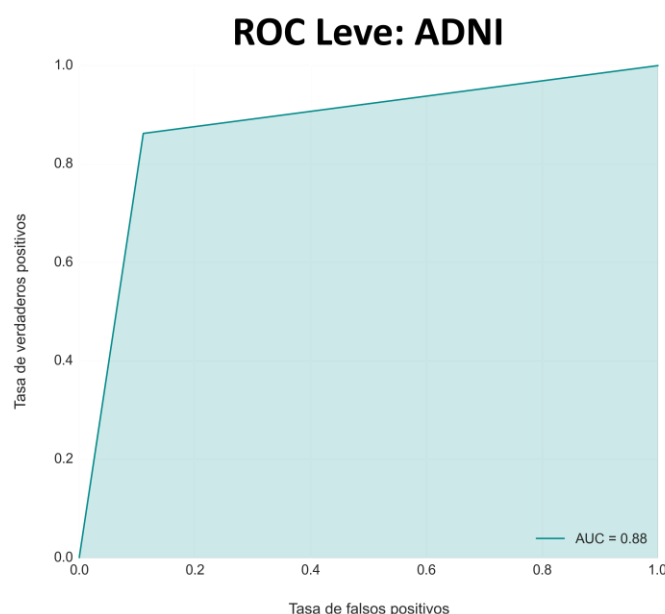


Ilustración 164 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - PReLU Softmax Adam sobre la base de datos de la ADNI.

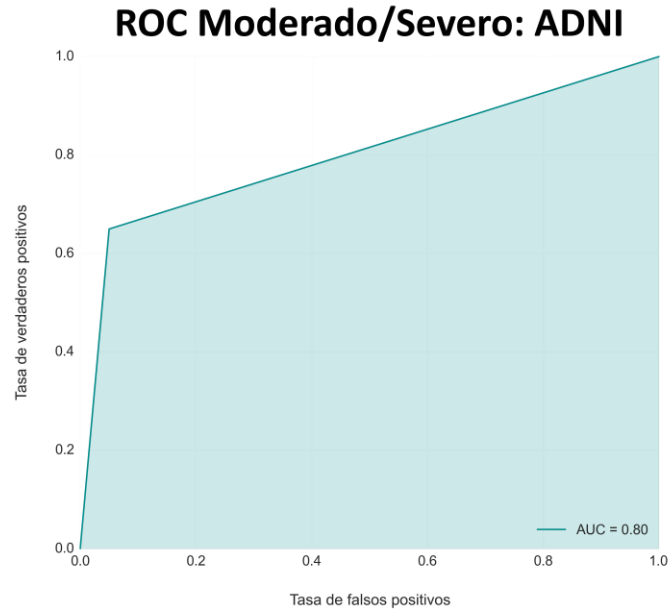


Ilustración 165 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - PReLU Softmax Adam sobre la base de datos de la ADNI.

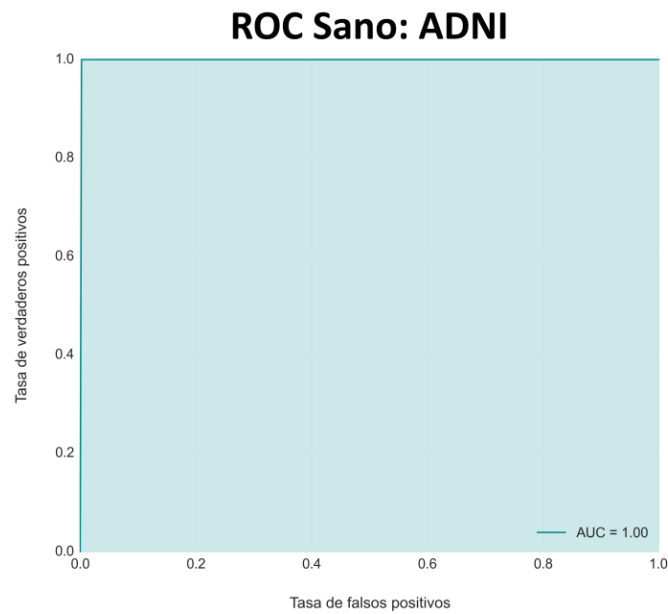


Ilustración 166 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - PReLU Softmax Adam sobre la base de datos de la ADNI.

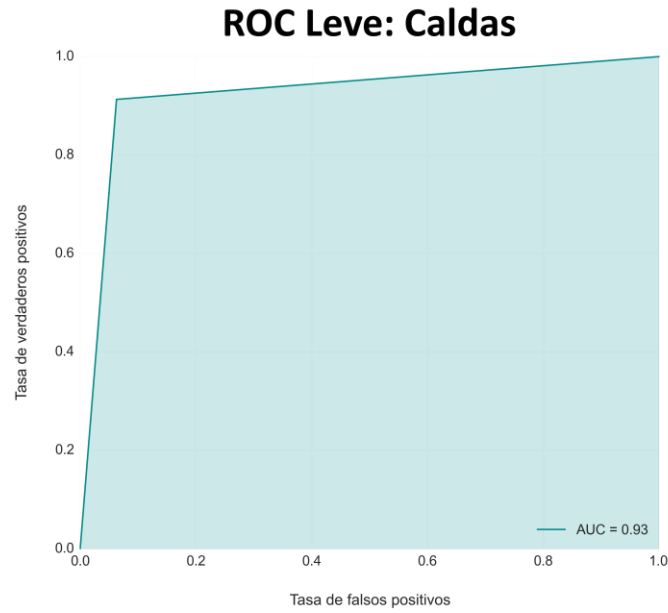


Ilustración 167 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - PReLU Softmax Adam sobre la base de datos de Caldas.

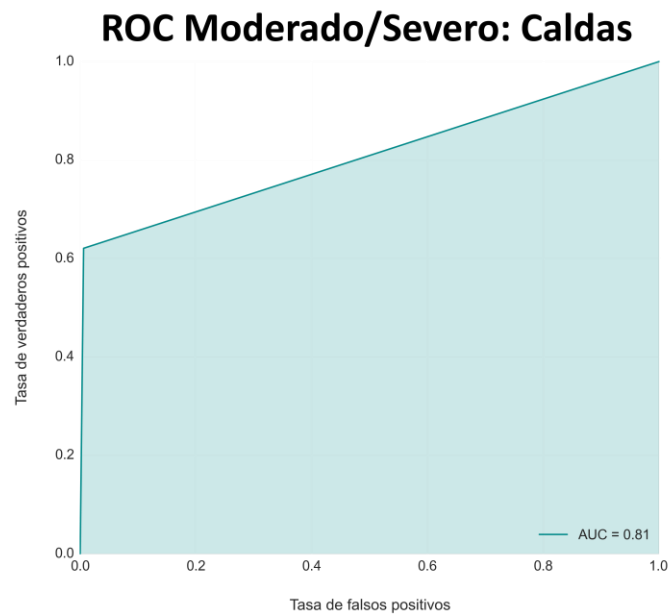


Ilustración 168 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - PReLU Softmax Adam sobre la base de datos de Caldas.

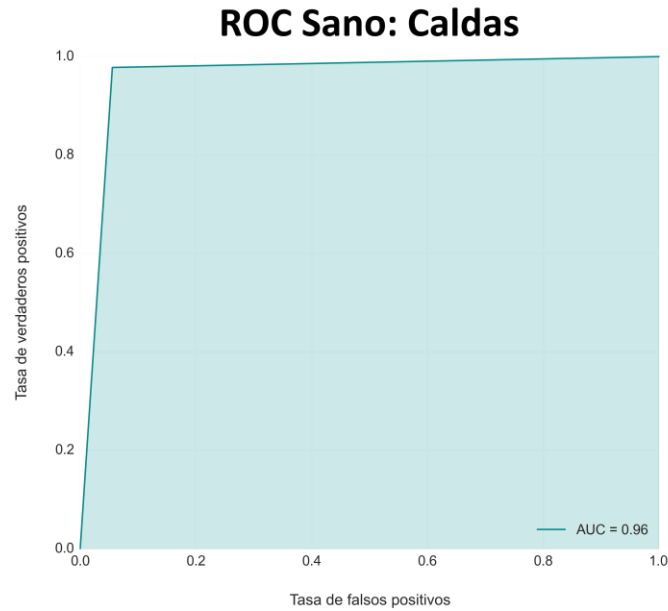


Ilustración 169 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - PReLU Softmax Adam sobre la base de datos de Caldas.

4.1.20.4 Métricas de desempeño

A continuación se muestran las métricas de desempeño calculadas a partir de las matrices de confusión y de las curvas ROC para la ANN – PReLU Softmax Adam.

ADNI		Caldas	
Exactitud	0,8814	Exactitud	0,9297
Precisión	0,8823	Precisión	0,9297
TPR (Sensibilidad)	0,8814	TPR (Sensibilidad)	0,9297
Especificidad	0,9556	Especificidad	0,9461
AUC Leve	0,8757	AUC Leve	0,9251
AUC Moderado/Severo	0,7999	AUC Moderado/Severo	0,8074
AUC Sano	0,9989	AUC Sano	0,9612
J Leve	0,7515	J Leve	0,8503
J Moderado/Severo	0,5997	J Moderado/Severo	0,6148
J Sano	0,9978	J Sano	0,9223
Kappa	0,8108	Kappa	0,8638

Tabla 23 Métricas de desempeño para la ANN – PReLU Softmax Adam sobre las bases de datos de la ADNI y Caldas.

4.1.21 Técnica 20: ANN - ReLU Softmax Adadelta

La explicación de los temas relacionados con redes neuronales artificiales se encuentra dispuesta en la sección «4.1.18 Técnica 17: ANN – ELU Softmax Adam».

4.1.21.1 Construcción, entrenamiento y clasificación

La ANN se fundamenta en la estructura dispuesta en la ilustración 170. Esta se construye empleando la clase *Sequential* perteneciente a los modelos de *Keras*. Asimismo, se emplean las funciones de activación *ReLU* y *Softmax* —pertenecientes a las funciones de activación de *Keras*— para construir las neuronas artificiales de cada capa. A continuación, se compila la red neuronal empleando la función *compile(loss='categorical_crossentropy', optimizer='adadelta',...)* perteneciente a la clase *Sequential* de los modelos de *Keras*.

El entrenamiento de la ANN se realiza bajo la técnica de validación cruzada con 10 iteraciones empleando las funciones *StratifiedKFold(n_folds=10,...)* y *fit* pertenecientes al módulo *cross-validation* de *Scikit-learn* y a la clase *Sequential* de los modelos de *Keras* respectivamente. Finalmente, se generan los estimados para cada registro de ambas bases de datos (*ADNI* y *Caldas*) empleando la función *predict_classes* perteneciente a la clase *Sequential* de los modelos de *Keras*.

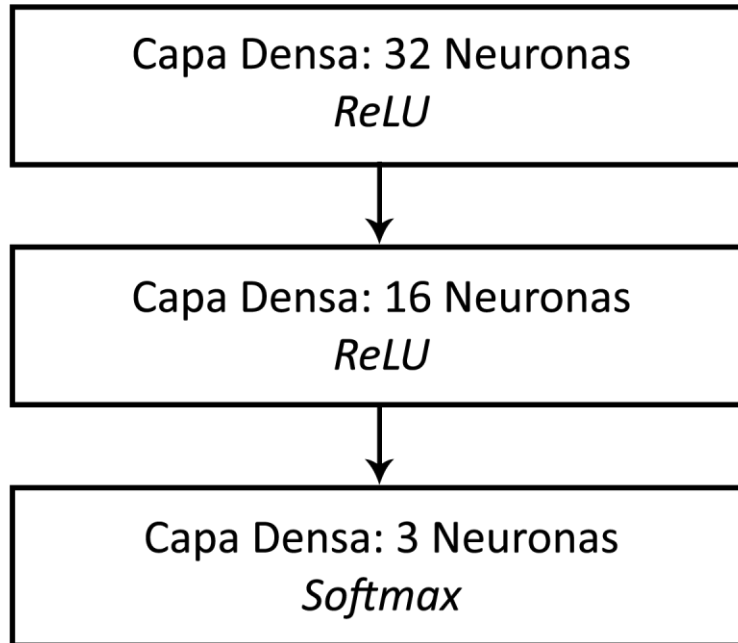


Ilustración 170 Estructura de la ANN – ReLU Softmax Adadelata de tres capas: dos capas ocultas de neuronas artificiales con funciones de activación ReLU y una capa de salida de neuronas artificiales con funciones de activación Softmax.

4.1.21.2 Matrices de confusión

En las ilustraciones 171 y 172 se muestran las matrices de confusión obtenidas para la ANN – ReLU Softmax Adadelata sobre las bases de datos de la ADNI y Caldas. A partir de estas se deduce que el clasificador logró un porcentaje de acierto del 0.90% y 0.95% en las respectivas predicciones. Asimismo, se puede observar que existe un bajo grado de confusión en las estimaciones de la técnica de aprendizaje de máquina respecto a las clases de predicción.

Matriz de Confusión: ADNI

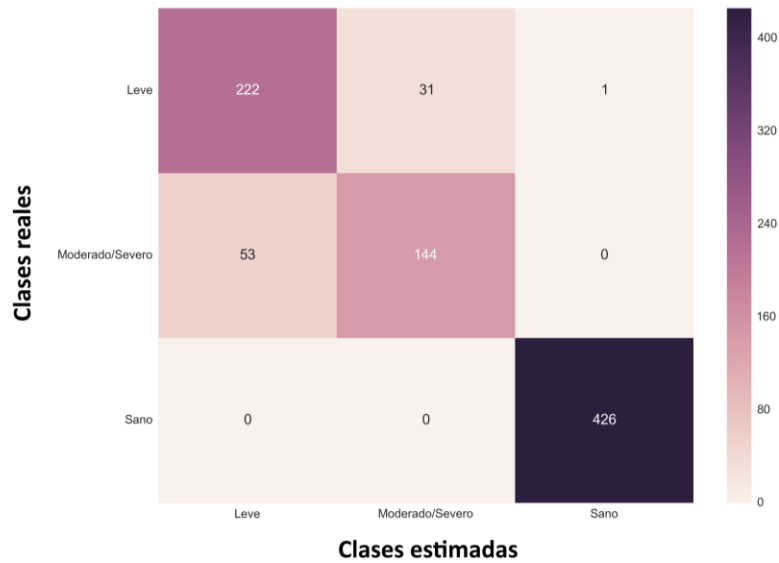


Ilustración 171 Matriz de confusión referente a los estimados de la ANN - ReLU Softmax Adadelata sobre la base de datos de la ADNI.

Matriz de Confusión: Caldas

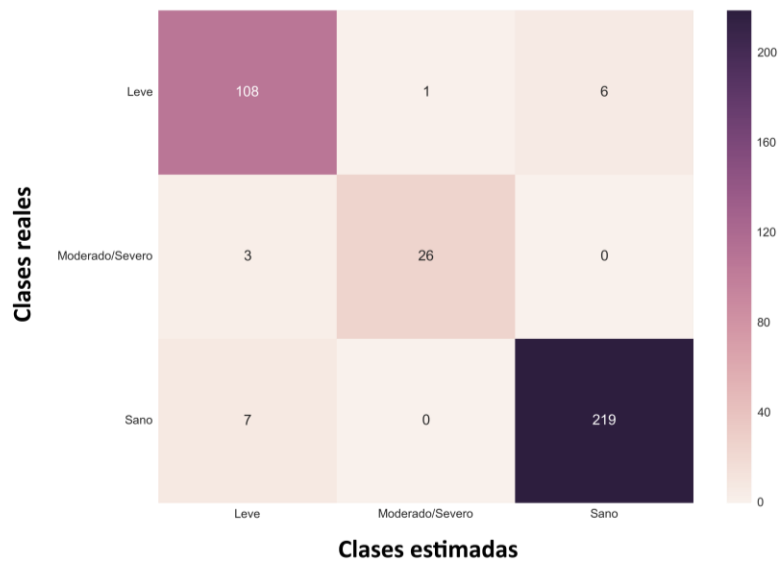


Ilustración 172 Matriz de confusión referente a los estimados de la ANN - ReLU Softmax Adadelata sobre la base de datos de Caldas.

4.1.21.3 Curvas ROC

En las ilustraciones 173 a 178 se muestran las curvas ROC obtenidas para la *ANN – ReLU Softmax Adadelta* sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se observa que el AUC para cada una de las clases se encuentra bastante cercana de 1, lo cual indica que la técnica de aprendizaje de máquina no presenta mayores dificultades para elegir la clase correcta al momento de realizar las predicciones. Solamente se presentan algunos inconvenientes menores para diferenciar las clases «Leve» y «Moderado/Severo» de las demás en la base de datos de la *ADNI*.

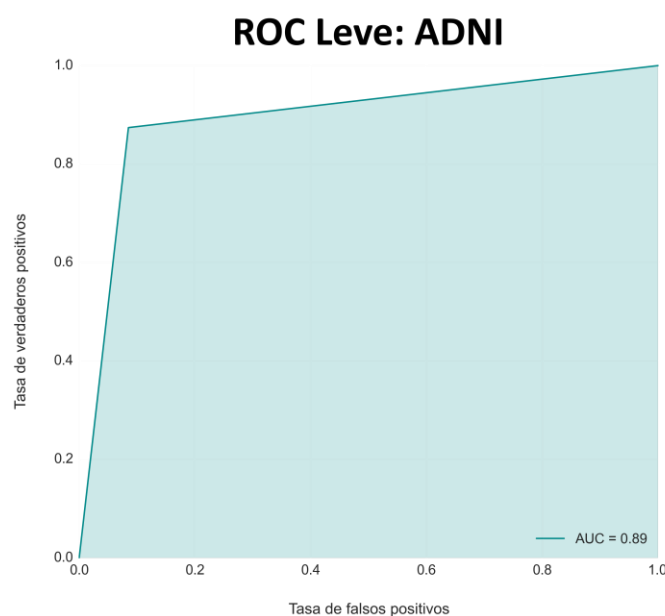


Ilustración 173 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - ReLU Softmax Adadelta sobre la base de datos de la ADNI.

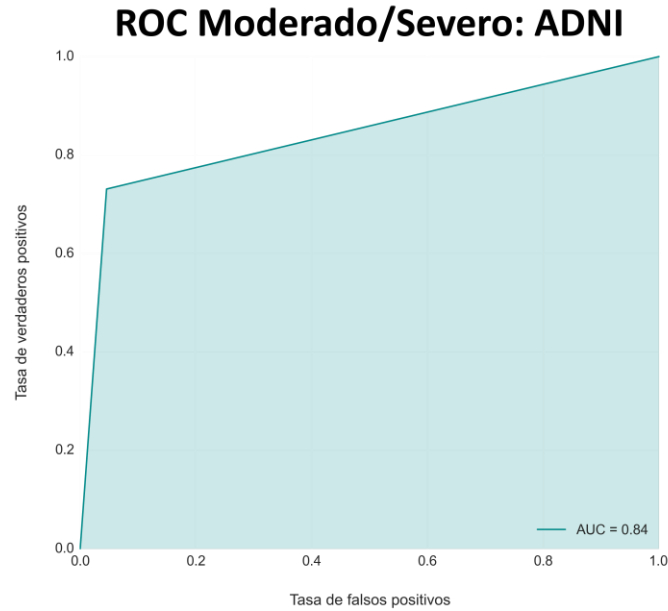


Ilustración 174 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - ReLU Softmax Adadelta sobre la base de datos de la ADNI.

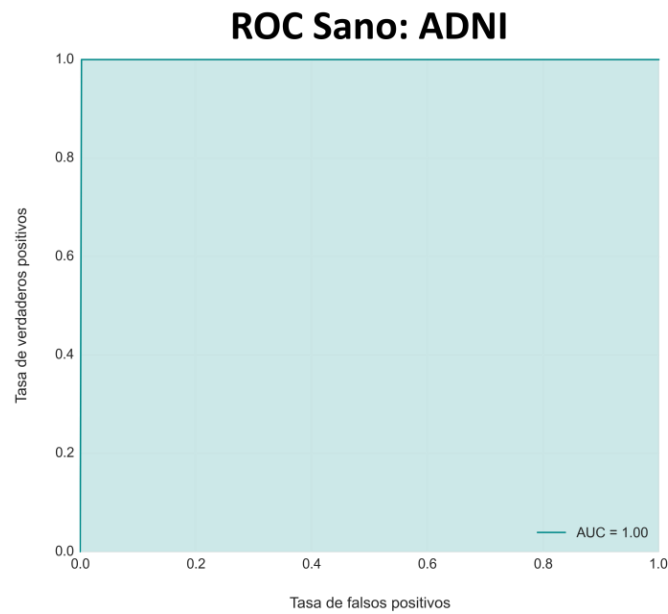


Ilustración 175 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - ReLU Softmax Adadelta sobre la base de datos de la ADNI.

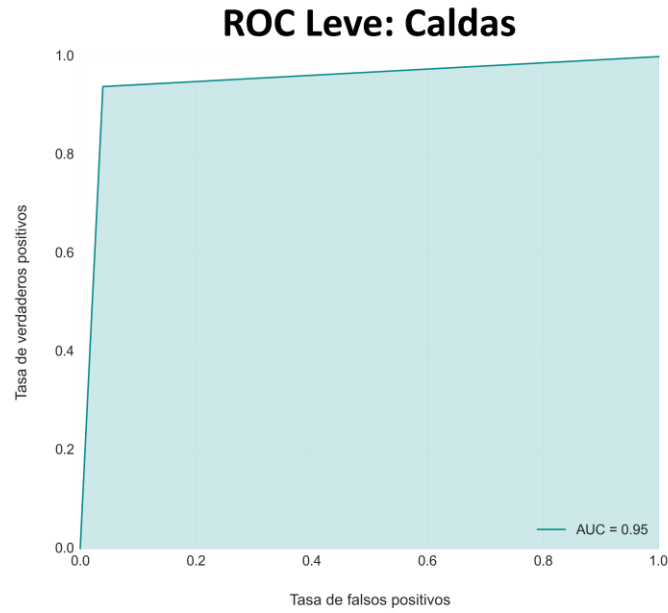


Ilustración 176 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - ReLU Softmax Adadelata sobre la base de datos de Caldas.

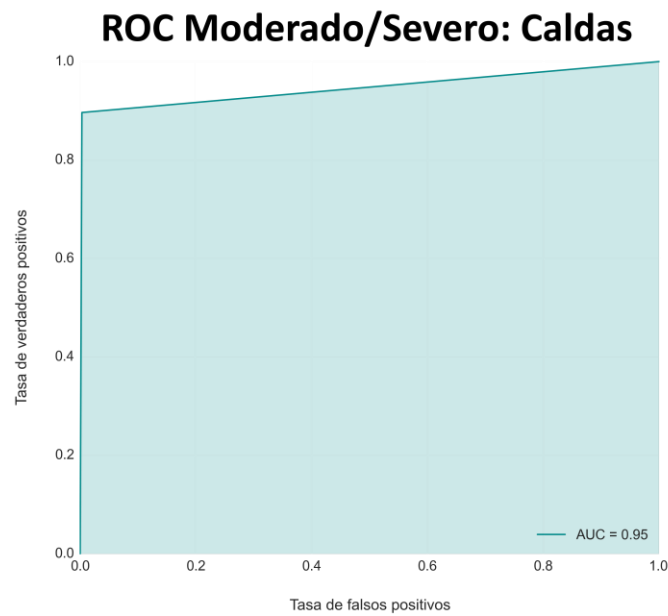


Ilustración 177 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - ReLU Softmax Adadelata sobre la base de datos de Caldas.

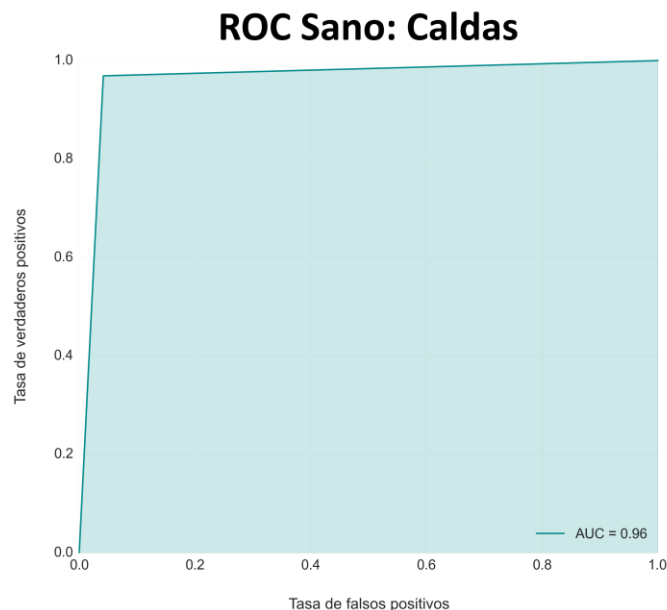


Ilustración 178 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - ReLU Softmax Adadelata sobre la base de datos de Caldas.

4.1.21.4 Métricas de desempeño

A continuación se muestran las métricas de desempeño calculadas a partir de las matrices de confusión y de las curvas ROC para la ANN – ReLU Softmax Adadelata.

ADNI		Caldas	
Exactitud	0,9031	Exactitud	0,9541
Precisión	0,9033	Precisión	0,9545
TPR (Sensibilidad)	0,9031	TPR (Sensibilidad)	0,9541
Especificidad	0,9640	Especificidad	0,9621
AUC Leve	0,8945	AUC Leve	0,9500
AUC Moderado/Severo	0,8427	AUC Moderado/Severo	0,9468
AUC Sano	0,9989	AUC Sano	0,9637
J Leve	0,7889	J Leve	0,8999
J Moderado/Severo	0,6854	J Moderado/Severo	0,8936
J Sano	0,9978	J Sano	0,9274
Kappa	0,8456	Kappa	0,9123

Tabla 24 Métricas de desempeño para la ANN – ReLU Softmax Adadelata sobre las bases de datos de la ADNI y Caldas.

4.1.22 Técnica 21: ANN - ReLU Softmax Adagrad

La explicación de los temas relacionados con redes neuronales artificiales se encuentra dispuesta en la sección «4.1.18 Técnica 17: ANN – ELU Softmax Adam».

4.1.22.1 Construcción, entrenamiento y clasificación

La ANN se fundamenta en la estructura dispuesta en la ilustración 179. Esta se construye empleando la clase *Sequential* perteneciente a los modelos de *Keras*. Asimismo, se emplean las funciones de activación *ReLU* y *Softmax* —pertenecientes a las funciones de activación de *Keras*— para construir las neuronas artificiales de cada capa. A continuación, se compila la red neuronal empleando la función *compile(loss='categorical_crossentropy', optimizer='adagrad',...)* perteneciente a la clase *Sequential* de los modelos de *Keras*.

El entrenamiento de la ANN se realiza bajo la técnica de validación cruzada con 10 iteraciones empleando las funciones *StratifiedKFold(n_folds=10,...)* y *fit* pertenecientes al módulo *cross-validation* de *Scikit-learn* y a la clase *Sequential* de los modelos de *Keras* respectivamente. Finalmente, se generan los estimados para cada registro de ambas bases de datos (*ADNI* y *Caldas*) empleando la función *predict_classes* perteneciente a la clase *Sequential* de los modelos de *Keras*.

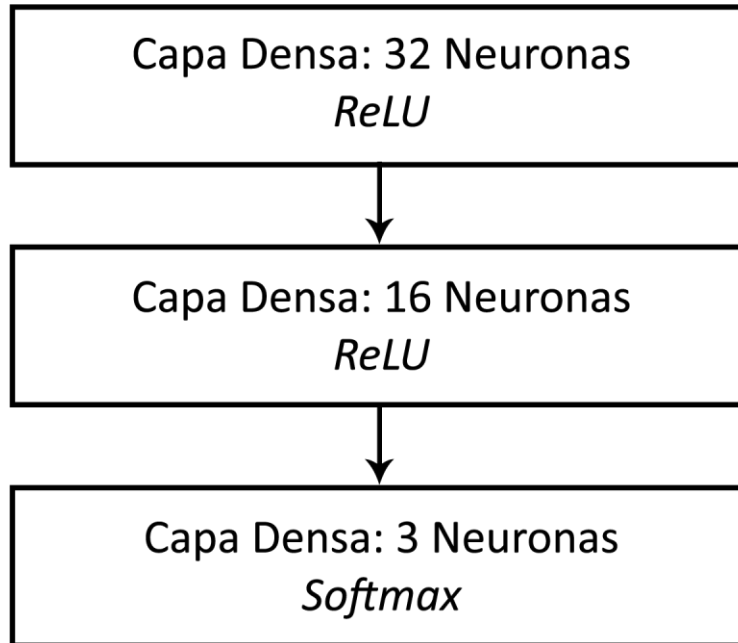


Ilustración 179 Estructura de la ANN – ReLU Softmax Adadelata de tres capas: dos capas ocultas de neuronas artificiales con funciones de activación ReLU y una capa de salida de neuronas artificiales con funciones de activación Softmax.

4.1.22.2 Matrices de confusión

En las ilustraciones 180 y 181 se muestran las matrices de confusión obtenidas para la ANN – ReLU Softmax Adagrad sobre las bases de datos de la ADNI y Caldas. A partir de estas se deduce que el clasificador logró un porcentaje de acierto del 0.90% y 0.95% en las respectivas predicciones. Asimismo, se puede observar que existe un bajo grado de confusión en las estimaciones de la técnica de aprendizaje de máquina respecto a las clases de predicción.

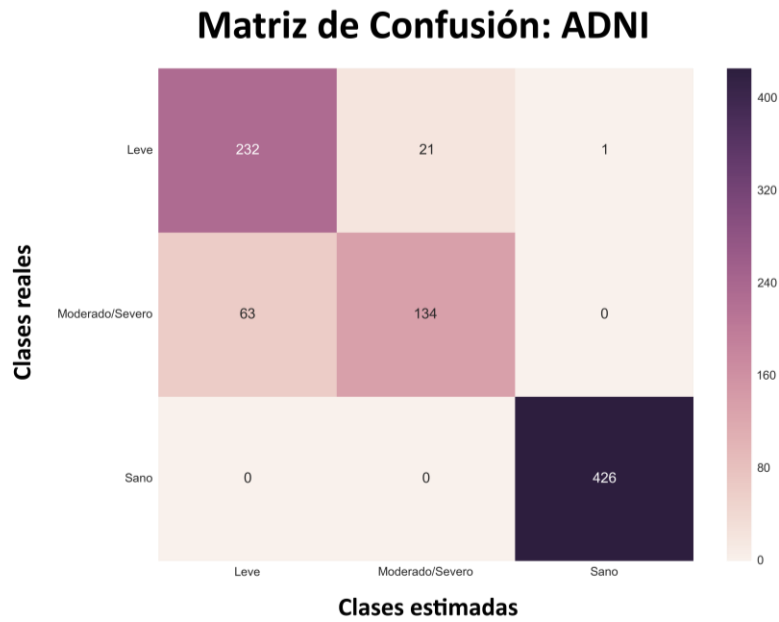


Ilustración 180 Matriz de confusión referente a los estimados de la ANN - ReLU Softmax Adagrad sobre la base de datos de la ADNI.

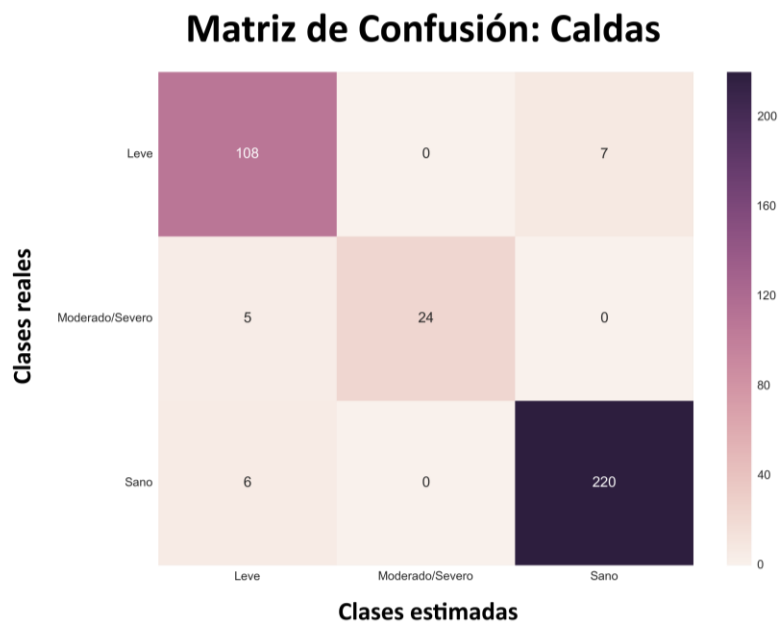


Ilustración 181 Matriz de confusión referente a los estimados de la ANN - ReLU Softmax Adagrad sobre la base de datos de Caldas.

4.1.22.3 Curvas ROC

En las ilustraciones 182 a 187 se muestran las curvas ROC obtenidas para la *ANN – ReLU Softmax Adagrad* sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se observa que el AUC para cada una de las clases se encuentra bastante cercana de 1, lo cual indica que la técnica de aprendizaje de máquina no presenta mayores dificultades para elegir la clase correcta al momento de realizar las predicciones. Solamente se presentan algunos inconvenientes menores para diferenciar las clases «Leve» y «Moderado/Severo» de las demás en la base de datos de la *ADNI*.

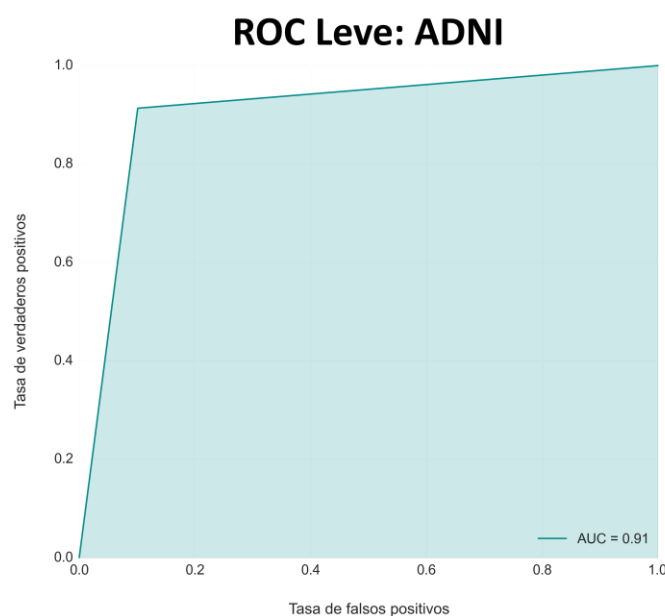


Ilustración 182 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - ReLU Softmax Adagrad sobre la base de datos de la ADNI.

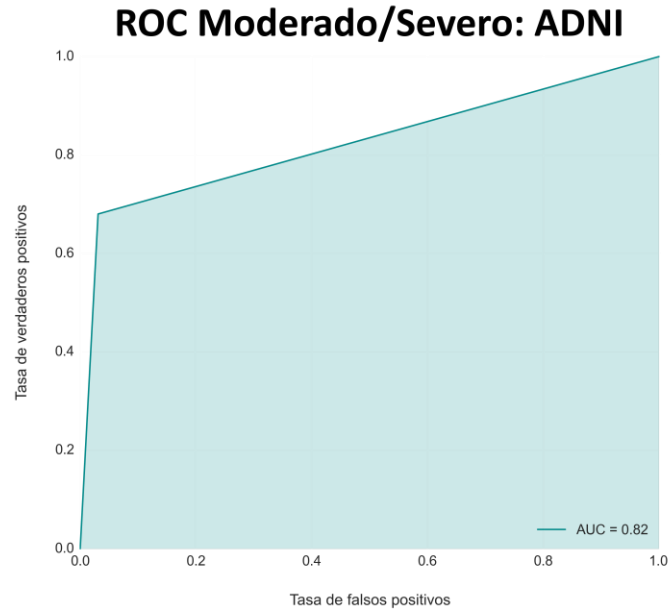


Ilustración 183 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - ReLU Softmax Adagrad sobre la base de datos de la ADNI.

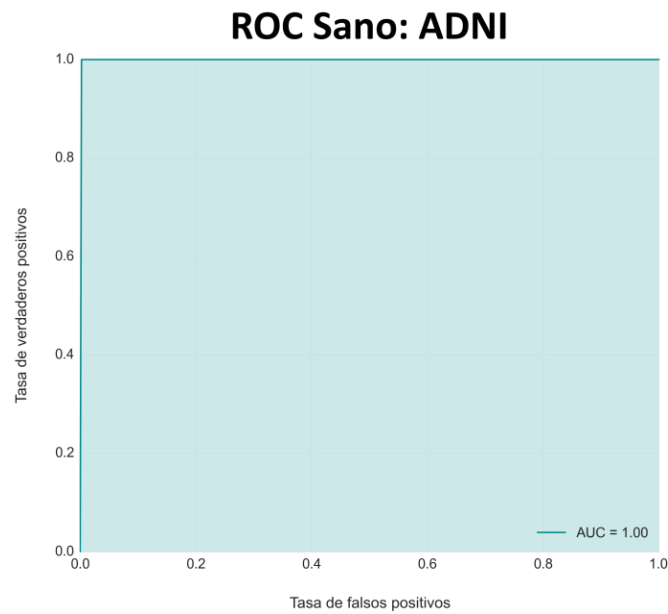


Ilustración 184 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - ReLU Softmax Adagrad sobre la base de datos de la ADNI.

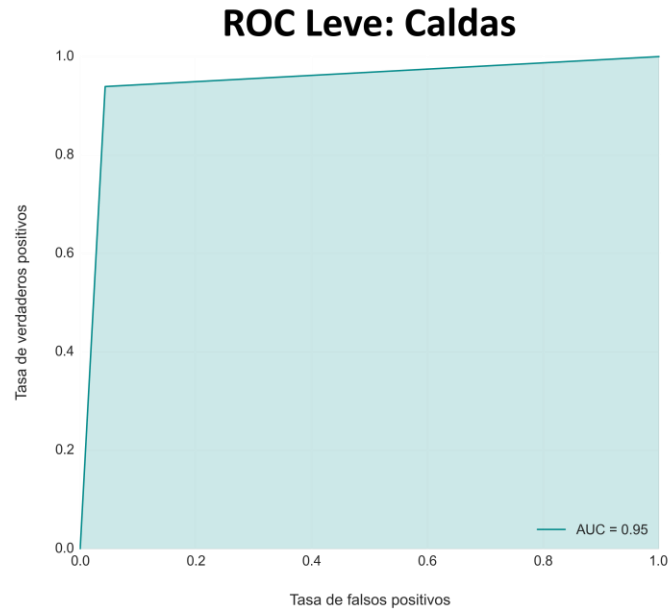


Ilustración 185 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - ReLU Softmax Adagrad sobre la base de datos de Caldas.

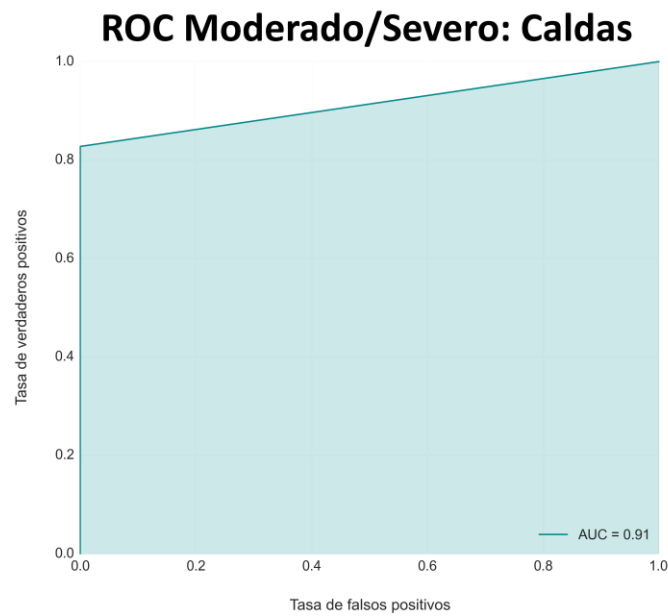


Ilustración 186 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - ReLU Softmax Adagrad sobre la base de datos de Caldas.

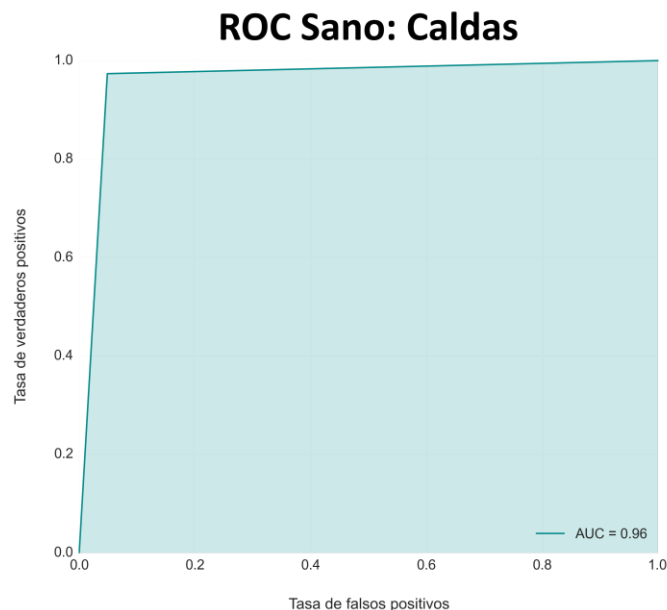


Ilustración 187 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - ReLU Softmax Adagrad sobre la base de datos de Caldas.

4.1.22.4 Métricas de desempeño

A continuación se muestran las métricas de desempeño calculadas a partir de las matrices de confusión y de las curvas ROC para la ANN – ReLU Softmax Adagrad.

ADNI		Caldas	
Exactitud	0,9031	Exactitud	0,9514
Precisión	0,9066	Precisión	0,9524
TPR (Sensibilidad)	0,9031	TPR (Sensibilidad)	0,9514
Especificidad	0,9627	Especificidad	0,9569
AUC Leve	0,9061	AUC Leve	0,9480
AUC Moderado/Severo	0,8247	AUC Moderado/Severo	0,9138
AUC Sano	0,9989	AUC Sano	0,9624
J Leve	0,8123	J Leve	0,8960
J Moderado/Severo	0,6493	J Moderado/Severo	0,8276
J Sano	0,9978	J Sano	0,9248
Kappa	0,8453	Kappa	0,9065

Tabla 25 Métricas de desempeño para la ANN – ReLU Softmax Adagrad sobre las bases de datos de la ADNI y Caldas.

4.1.23 Técnica 22: ANN - ReLU Softmax Adam

La explicación de los temas relacionados con redes neuronales artificiales se encuentra dispuesta en la sección «4.1.18 Técnica 17: ANN – ELU Softmax Adam».

4.1.23.1 Construcción, entrenamiento y clasificación

La ANN se fundamenta en la estructura dispuesta en la ilustración 188. Esta se construye empleando la clase *Sequential* perteneciente a los modelos de *Keras*. Asimismo, se emplean las funciones de activación *ReLU* y *Softmax* —pertenecientes a las funciones de activación de *Keras*— para construir las neuronas artificiales de cada capa. A continuación, se compila la red neuronal empleando la función *compile(loss='categorical_crossentropy', optimizer='adam',...)* perteneciente a la clase *Sequential* de los modelos de *Keras*.

El entrenamiento de la ANN se realiza bajo la técnica de validación cruzada con 10 iteraciones empleando las funciones *StratifiedKFold(n_folds=10,...)* y *fit* pertenecientes al módulo *cross-validation* de *Scikit-learn* y a la clase *Sequential* de los modelos de *Keras* respectivamente. Finalmente, se generan los estimados para cada registro de ambas bases de datos (*ADNI* y *Caldas*) empleando la función *predict_classes* perteneciente a la clase *Sequential* de los modelos de *Keras*.

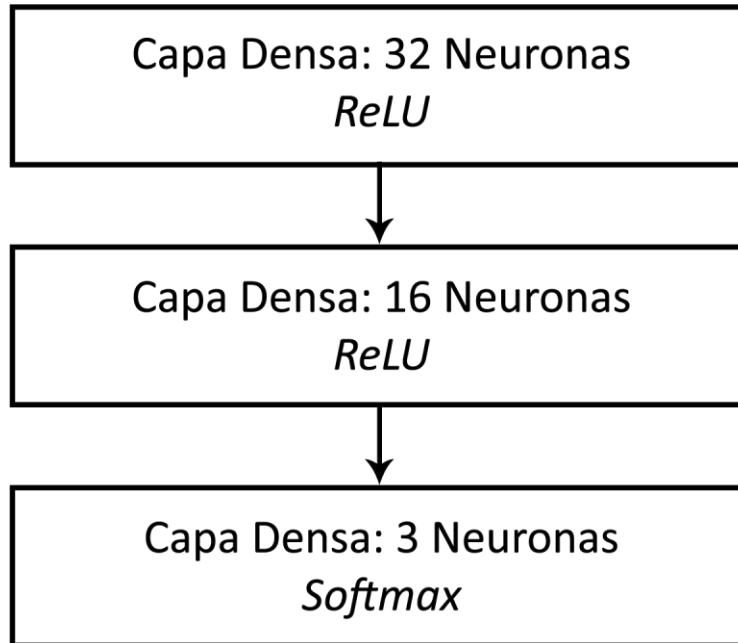


Ilustración 188 Estructura de la ANN – ReLU Softmax Adadelta de tres capas: dos capas ocultas de neuronas artificiales con funciones de activación ReLU y una capa de salida de neuronas artificiales con funciones de activación Softmax.

4.1.23.2 Matrices de confusión

En las ilustraciones 189 y 190 se muestran las matrices de confusión obtenidas para la ANN – ReLU Softmax Adam sobre las bases de datos de la ADNI y Caldas. A partir de estas se deduce que el clasificador logró un porcentaje de acierto del 0.89% y 0.95% en las respectivas predicciones. Asimismo, se puede observar que existe un bajo grado de confusión en las estimaciones de la técnica de aprendizaje de máquina respecto a las clases de predicción.

Matriz de Confusión: ADNI

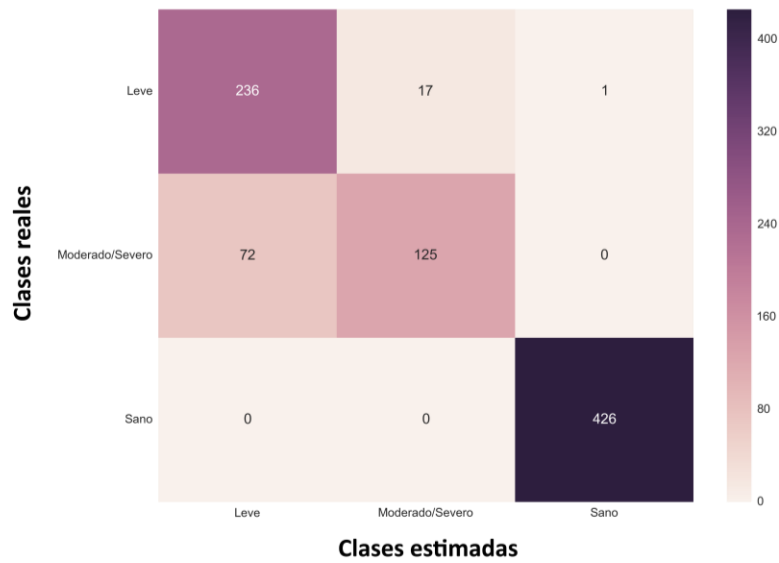


Ilustración 189 Matriz de confusión referente a los estimados de la ANN - ReLU Softmax Adam sobre la base de datos de la ADNI.

Matriz de Confusión: Caldas

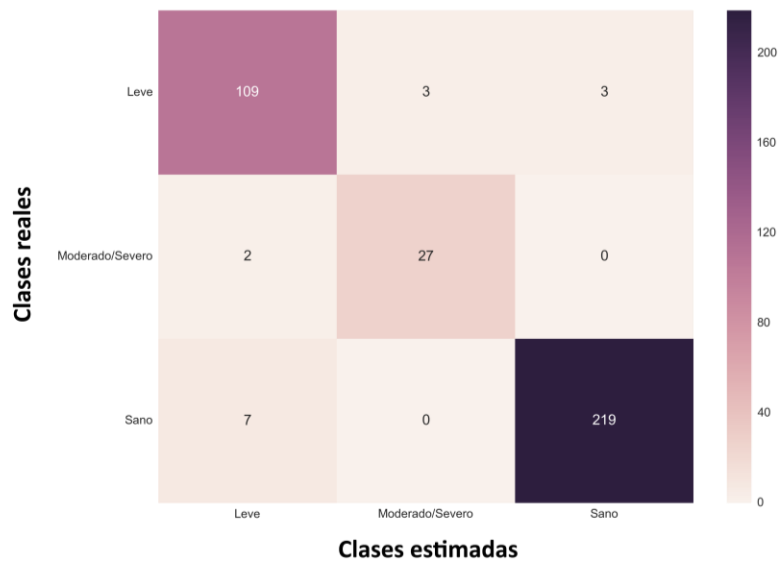


Ilustración 190 Matriz de confusión referente a los estimados de la ANN - ReLU Softmax Adam sobre la base de datos de Caldas.

4.1.23.3 Curvas ROC

En las ilustraciones 191 a 196 se muestran las curvas ROC obtenidas para la *ANN – ReLU Softmax Adam* sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se observa que el AUC para cada una de las clases se encuentra bastante cercana de 1, lo cual indica que la técnica de aprendizaje de máquina no presenta mayores dificultades para elegir la clase correcta al momento de realizar las predicciones. Solamente se presentan algunos inconvenientes menores para diferenciar las clases «Leve» y «Moderado/Severo» de las demás en la base de datos de la *ADNI*.

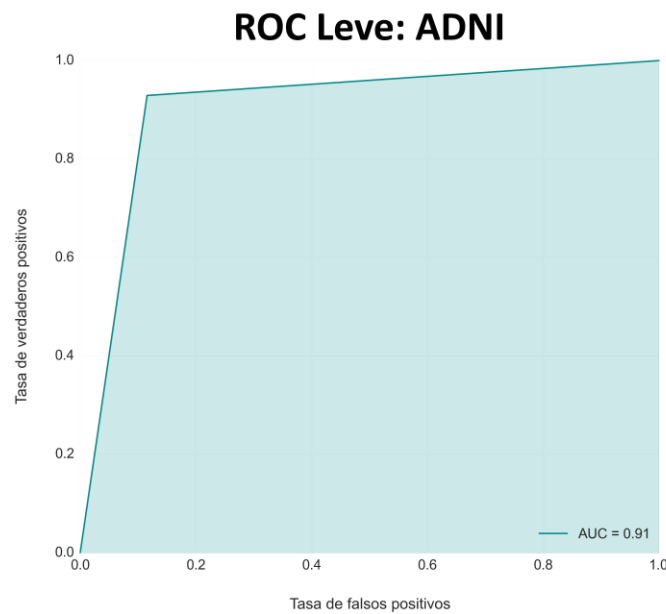


Ilustración 191 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - ReLU Softmax Adam sobre la base de datos de la ADNI.

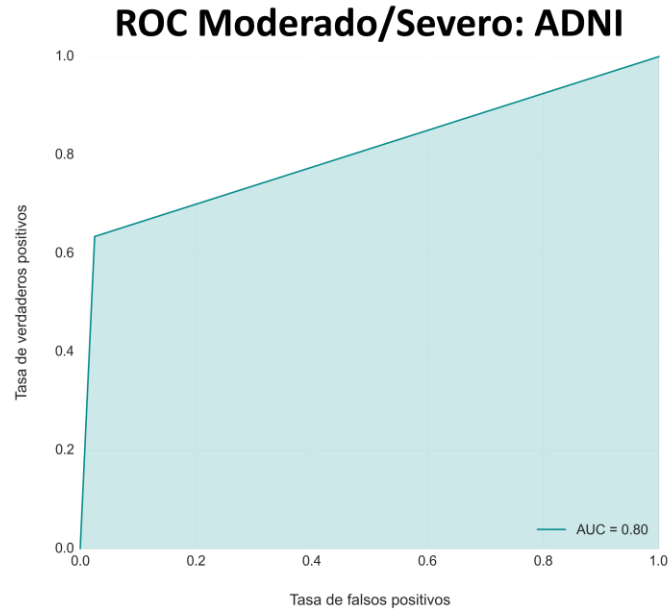


Ilustración 192 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - ReLU Softmax Adam sobre la base de datos de la ADNI.

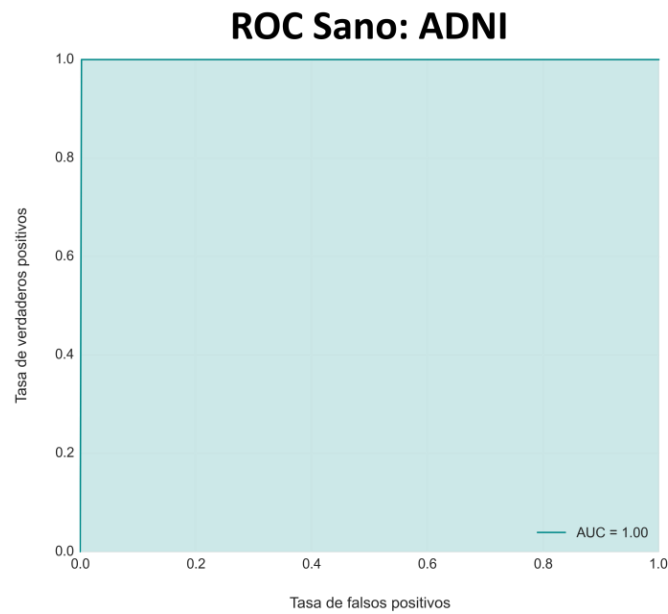


Ilustración 193 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - ReLU Softmax Adam sobre la base de datos de la ADNI.

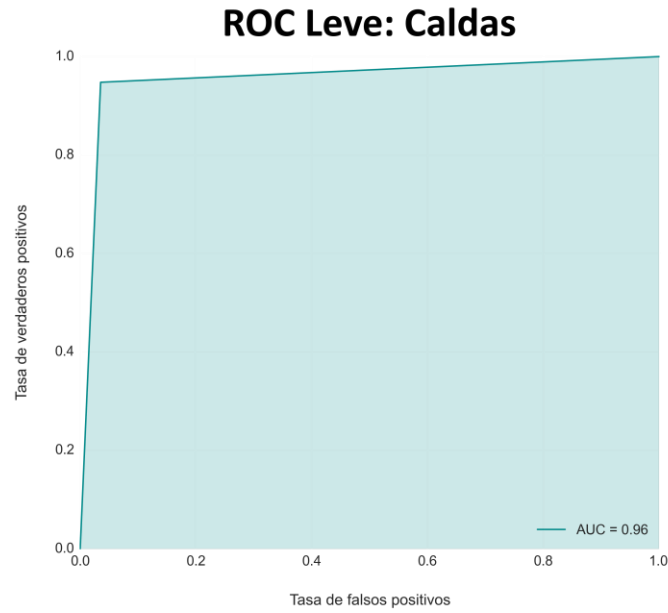


Ilustración 194 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - ReLU Softmax Adam sobre la base de datos de Caldas.

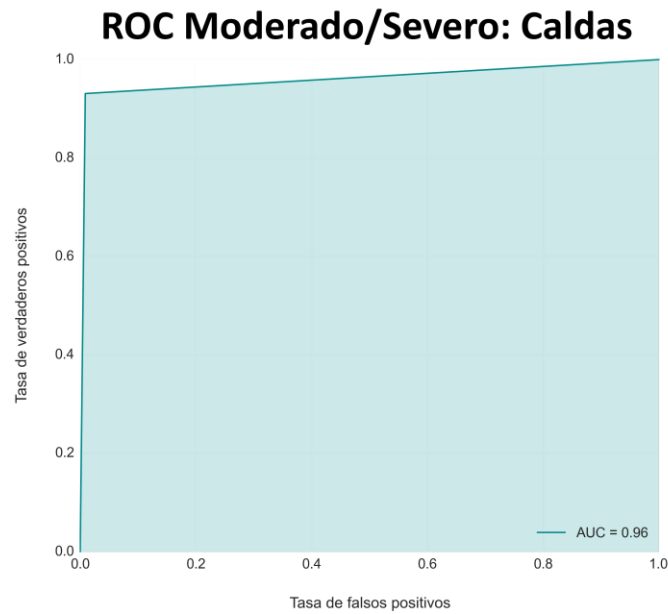


Ilustración 195 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - ReLU Softmax Adam sobre la base de datos de Caldas.

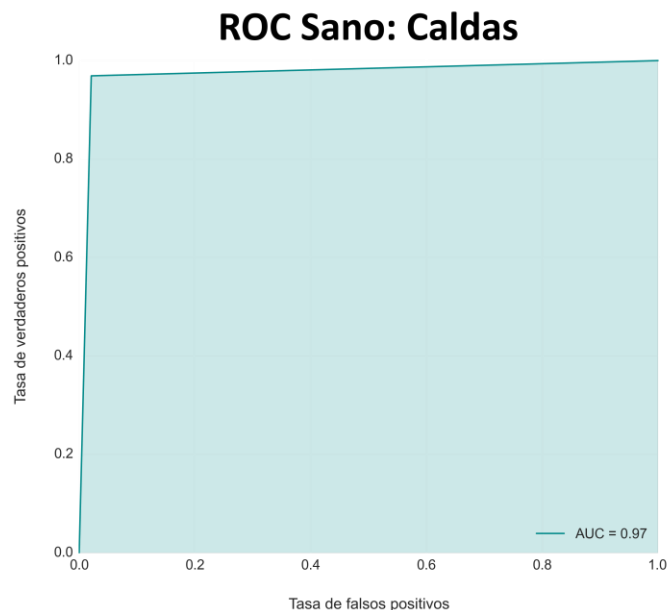


Ilustración 196 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - ReLU Softmax Adam sobre la base de datos de Caldas.

4.1.23.4 Métricas de desempeño

A continuación se muestran las métricas de desempeño calculadas a partir de las matrices de confusión y de las curvas ROC para la ANN – ReLU Softmax Adam.

ADNI		Caldas	
Exactitud	0,8974	Exactitud	0,9595
Precisión	0,9043	Precisión	0,9602
TPR (Sensibilidad)	0,8974	TPR (Sensibilidad)	0,9595
Especificidad	0,9598	Especificidad	0,9756
AUC Leve	0,9068	AUC Leve	0,9563
AUC Moderado/Severo	0,8048	AUC Moderado/Severo	0,9611
AUC Sano	0,9989	AUC Sano	0,9741
J Leve	0,8136	J Leve	0,9125
J Moderado/Severo	0,6095	J Moderado/Severo	0,9222
J Sano	0,9978	J Sano	0,9482
Kappa	0,8359	Kappa	0,9232

Tabla 26 Métricas de desempeño para la ANN – ReLU Softmax Adam sobre las bases de datos de la ADNI y Caldas.

4.1.24 Técnica 23: *ANN - ReLU Softmax Adam Deep*

La explicación de los temas relacionados con redes neuronales artificiales se encuentra dispuesta en la sección «4.1.18 Técnica 17: ANN – ELU Softmax Adam».

4.1.24.1 *Construcción, entrenamiento y clasificación*

La ANN se fundamenta en la estructura dispuesta en la ilustración 197. Esta se construye empleando la clase *Sequential* perteneciente a los modelos de *Keras*. Asimismo, se emplean las funciones de activación *ReLU* y *Softmax* —pertenecientes a las funciones de activación de *Keras*— para construir las neuronas artificiales de cada capa. A continuación, se compila la red neuronal empleando la función *compile(loss='categorical_crossentropy', optimizer='adam',...)* perteneciente a la clase *Sequential* de los modelos de *Keras*.

El entrenamiento de la ANN se realiza bajo la técnica de validación cruzada con 10 iteraciones empleando las funciones *StratifiedKFold(n_folds=10,...)* y *fit* pertenecientes al módulo *cross-validation* de *Scikit-learn* y a la clase *Sequential* de los modelos de *Keras* respectivamente. Finalmente, se generan los estimados para cada registro de ambas bases de datos (*ADNI* y *Caldas*) empleando la función *predict_classes* perteneciente a la clase *Sequential* de los modelos de *Keras*.

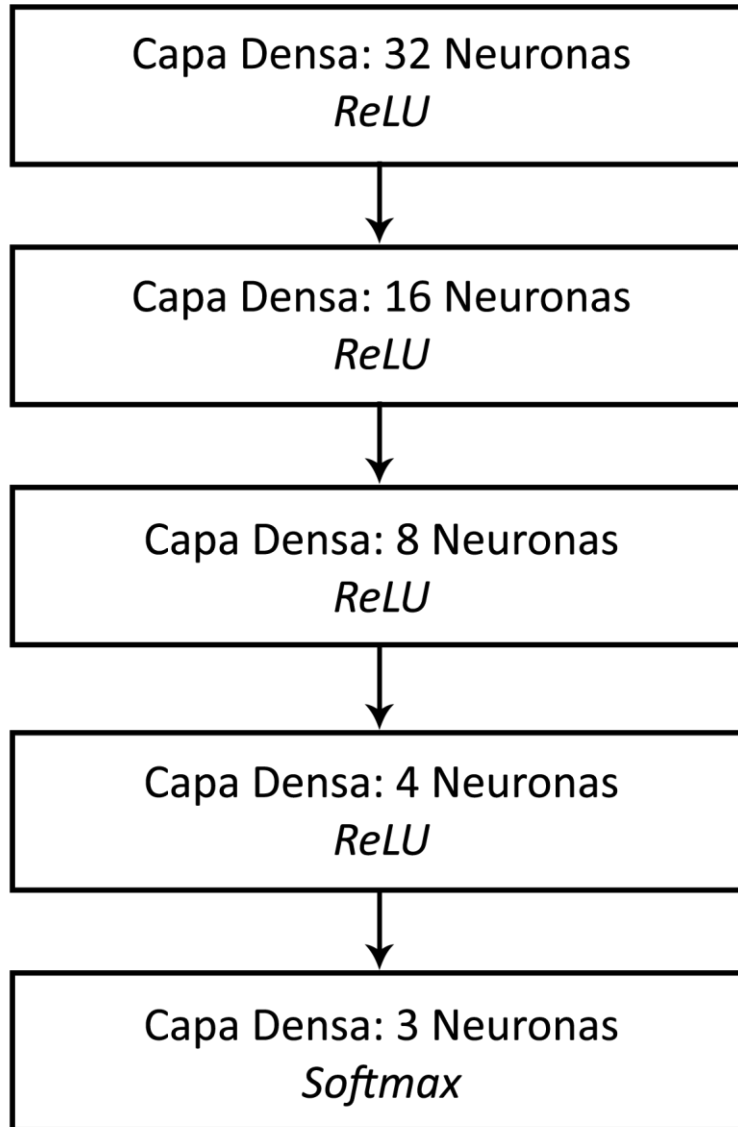


Ilustración 197 Estructura de la ANN – ReLU Softmax Adam Deep de cinco capas: cuatro capas ocultas de neuronas artificiales con funciones de activación ReLU y una capa de salida de neuronas artificiales con funciones de activación Softmax.

4.1.24.2 Matrices de confusión

En las ilustraciones 198 y 199 se muestran las matrices de confusión obtenidas para la ANN – ReLU Softmax Adam Deep sobre las bases de datos de la ADNI y Caldas. A partir de estas se deduce que el clasificador logró un porcentaje de acierto del 0.69% y 0.85% en las

respectivas predicciones. No obstante, se puede observar que existe un alto grado de confusión en las estimaciones de la técnica de aprendizaje de máquina respecto a las tres clases de predicción.

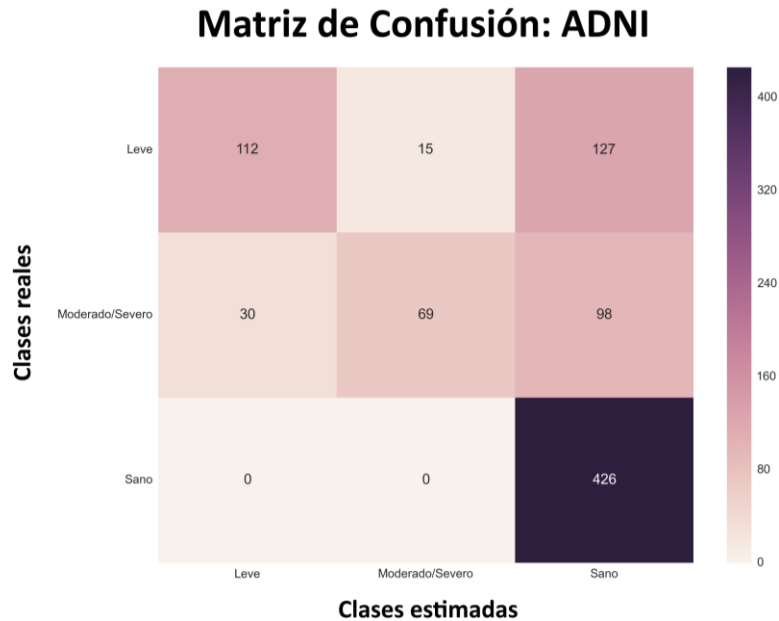


Ilustración 198 Matriz de confusión referente a los estimados de la ANN - ReLU Softmax Adam Deep sobre la base de datos de la ADNI.

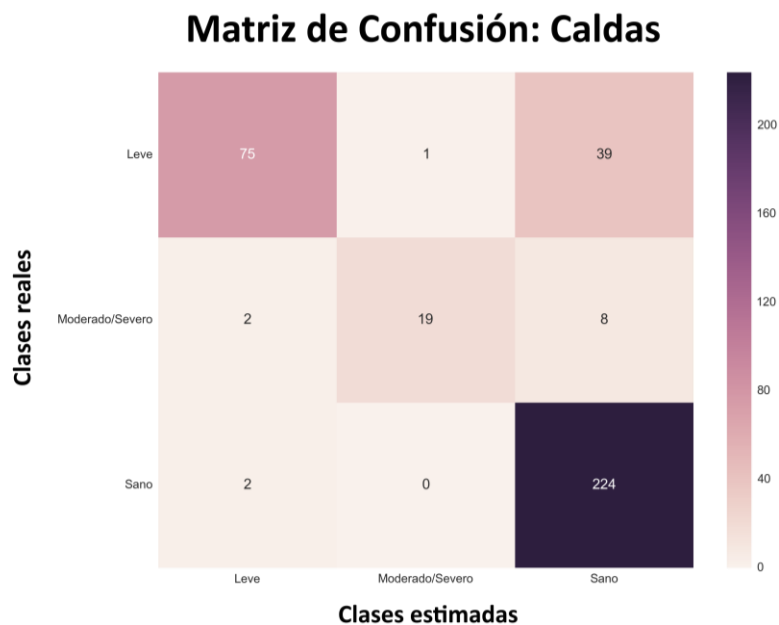


Ilustración 199 Matriz de confusión referente a los estimados de la ANN - ReLU Softmax Adam Deep sobre la base de datos de Caldas.

4.1.24.3 Curvas ROC

En las ilustraciones 200 a 205 se muestran las curvas ROC obtenidas para la *ANN – ReLU Softmax Adam Deep* sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se observa que el AUC para cada una de las clases se encuentra bastante alejada de 1, lo cual indica que la técnica de aprendizaje de máquina presenta grandes dificultades para elegir la clase correcta al momento de realizar las predicciones.

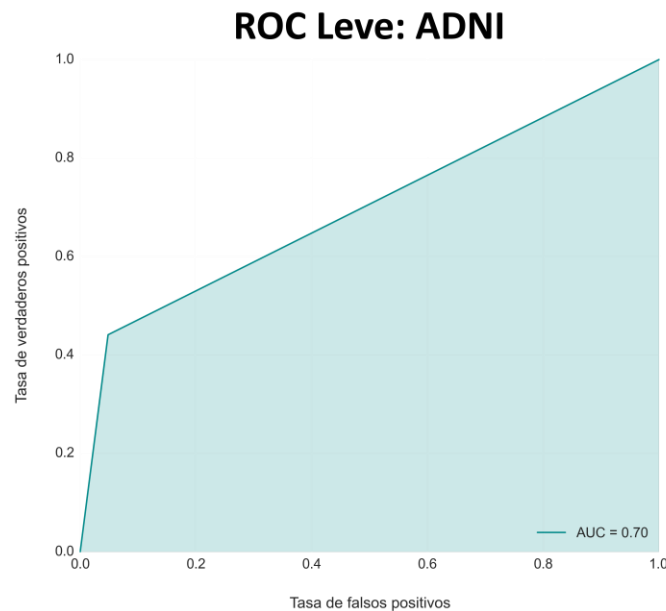


Ilustración 200 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la *ANN - ReLU Softmax Adam Deep* sobre la base de datos de la *ADNI*.

ROC Moderado/Severo: ADNI

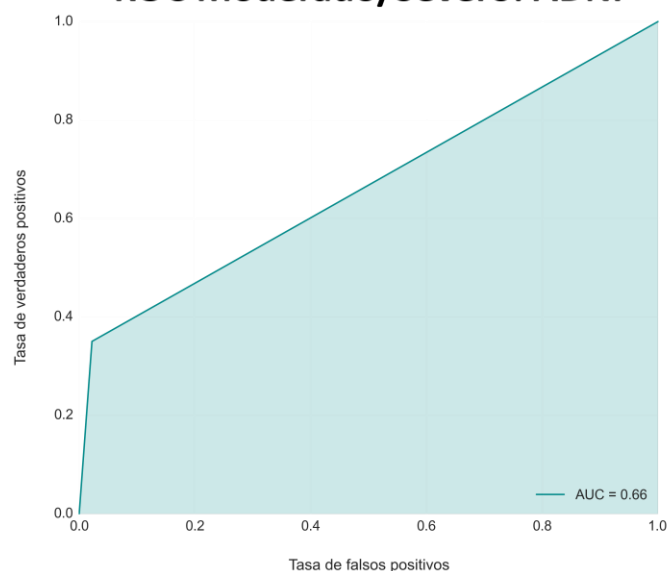


Ilustración 201 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - ReLU Softmax Adam Deep sobre la base de datos de la ADNI.

ROC Sano: ADNI

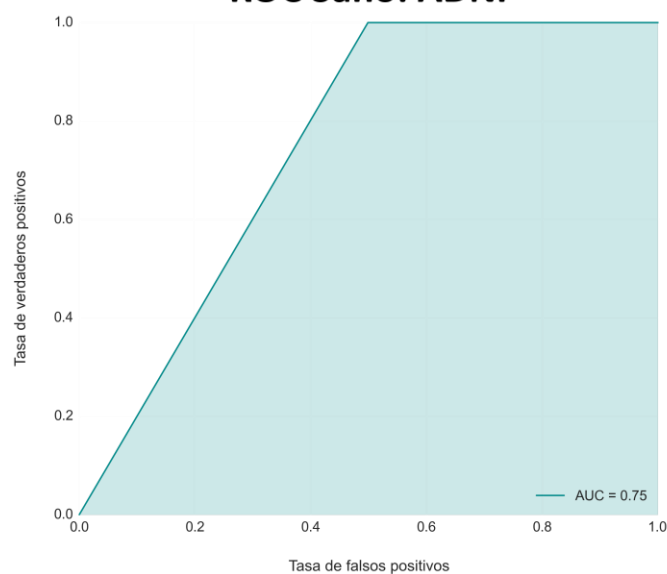


Ilustración 202 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - ReLU Softmax Adam Deep sobre la base de datos de la ADNI.

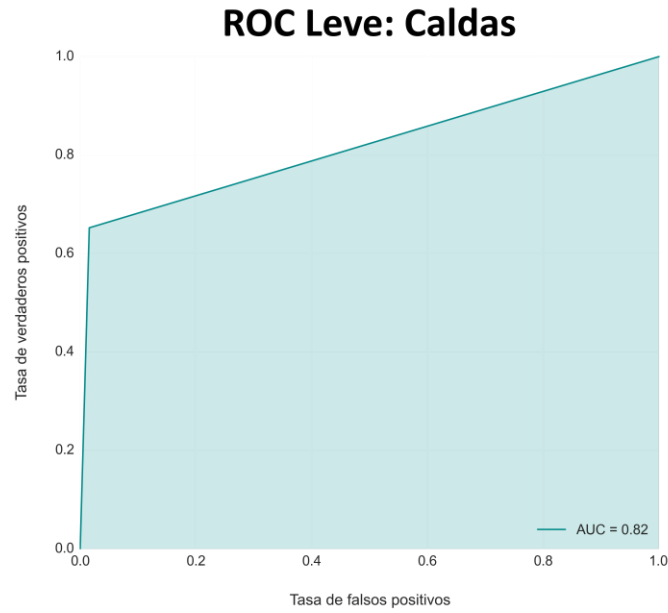


Ilustración 203 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - ReLU Softmax Adam Deep sobre la base de datos de Caldas.

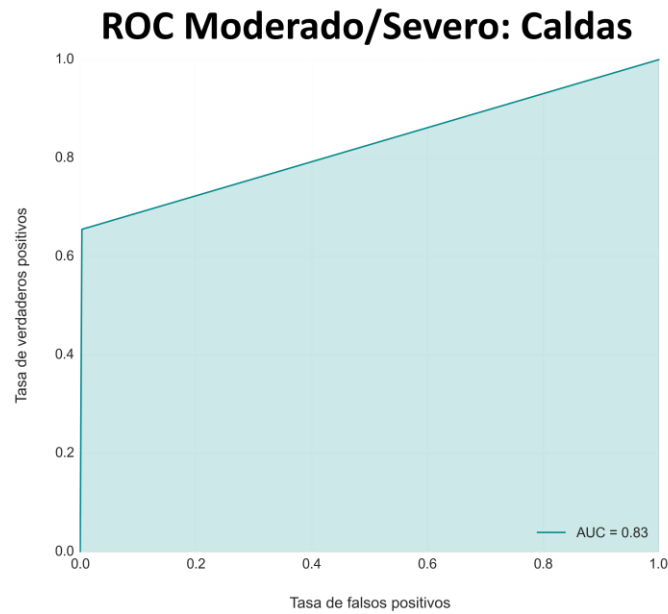


Ilustración 204 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - ReLU Softmax Adam Deep sobre la base de datos de Caldas.

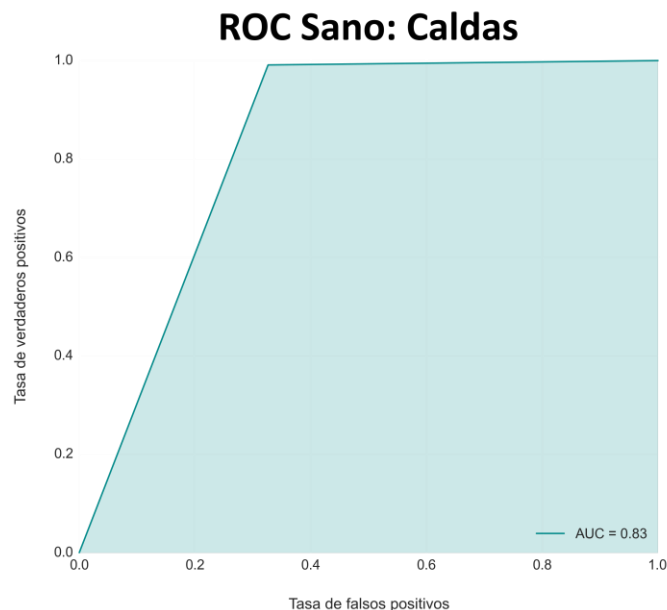


Ilustración 205 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - ReLU Softmax Adam Deep sobre la base de datos de Caldas.

4.1.24.4 Métricas de desempeño

A continuación se muestran las métricas de desempeño calculadas a partir de las matrices de confusión y de las curvas ROC para la ANN – ReLU Softmax Adam Deep.

ADNI		Caldas	
Exactitud	0,6921	Exactitud	0,8595
Precisión	0,7308	Precisión	0,8744
TPR (Sensibilidad)	0,6921	TPR (Sensibilidad)	0,8595
Especificidad	0,7388	Especificidad	0,7955
AUC Leve	0,6964	AUC Leve	0,8182
AUC Moderado/Severo	0,6641	AUC Moderado/Severo	0,8261
AUC Sano	0,7506	AUC Sano	0,8324
J Leve	0,3928	J Leve	0,6365
J Moderado/Severo	0,3282	J Moderado/Severo	0,6522
J Sano	0,5011	J Sano	0,6648
Kappa	0,4608	Kappa	0,7084

Tabla 27 Métricas de desempeño para la ANN – ReLU Softmax Adam Deep sobre las bases de datos de la ADNI y Caldas.

4.1.25 Técnica 24: ANN - ReLU Softmax Adamax

La explicación de los temas relacionados con redes neuronales artificiales se encuentra dispuesta en la sección «4.1.18 Técnica 17: ANN – ELU Softmax Adam».

4.1.25.1 Construcción, entrenamiento y clasificación

La ANN se fundamenta en la estructura dispuesta en la ilustración 206. Esta se construye empleando la clase *Sequential* perteneciente a los modelos de *Keras*. Asimismo, se emplean las funciones de activación *ReLU* y *Softmax* —pertenecientes a las funciones de activación de *Keras*— para construir las neuronas artificiales de cada capa. A continuación, se compila la red neuronal empleando la función *compile(loss='categorical_crossentropy', optimizer='adamax',...)* perteneciente a la clase *Sequential* de los modelos de *Keras*.

El entrenamiento de la ANN se realiza bajo la técnica de validación cruzada con 10 iteraciones empleando las funciones *StratifiedKFold(n_folds=10,...)* y *fit* pertenecientes al módulo *cross-validation* de *Scikit-learn* y a la clase *Sequential* de los modelos de *Keras* respectivamente. Finalmente, se generan los estimados para cada registro de ambas bases de datos (*ADNI* y *Caldas*) empleando la función *predict_classes* perteneciente a la clase *Sequential* de los modelos de *Keras*.

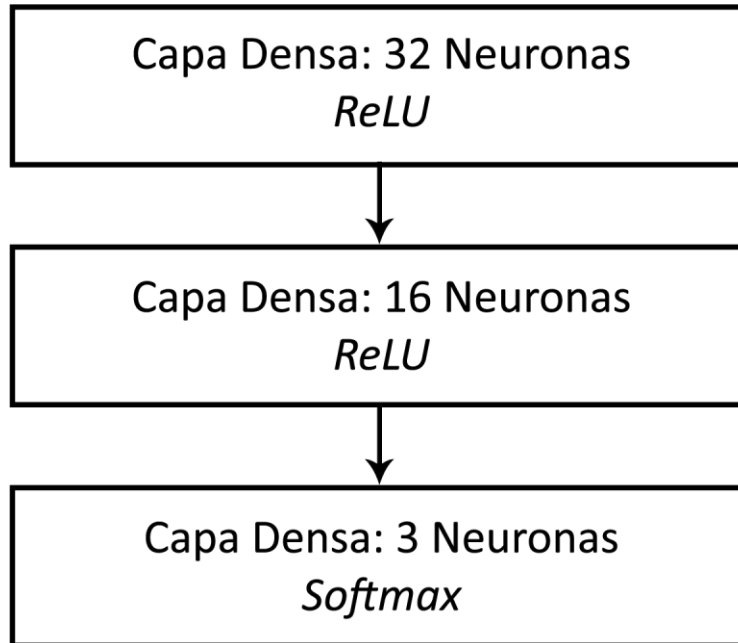


Ilustración 206 Estructura de la ANN – ReLU Softmax Adamax de tres capas: dos capas ocultas de neuronas artificiales con funciones de activación ReLU y una capa de salida de neuronas artificiales con funciones de activación Softmax.

4.1.25.2 Matrices de confusión

En las ilustraciones 207 y 208 se muestran las matrices de confusión obtenidas para la ANN – ReLU Softmax Adamax sobre las bases de datos de la ADNI y Caldas. A partir de estas se deduce que el clasificador logró un porcentaje de acierto del 0.90% y 0.96% en las respectivas predicciones. Asimismo, se puede observar que existe un bajo grado de confusión en las estimaciones de la técnica de aprendizaje de máquina respecto a las clases de predicción.

Matriz de Confusión: ADNI

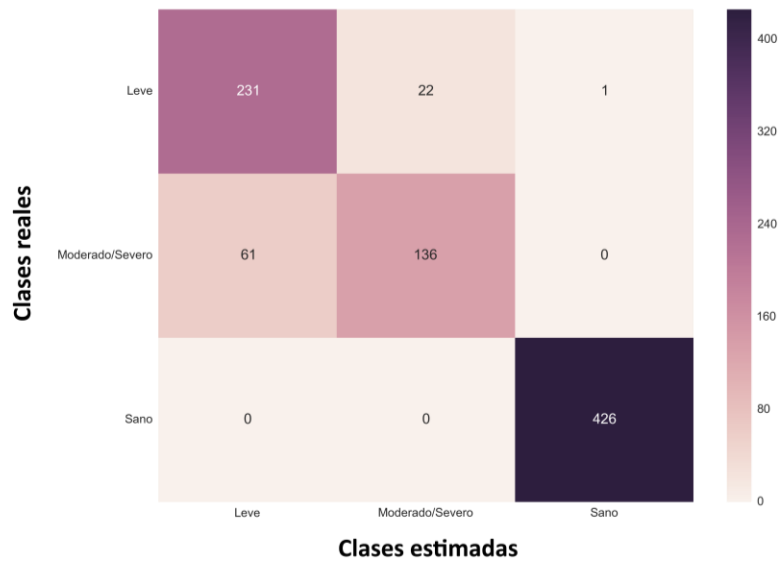


Ilustración 207 Matriz de confusión referente a los estimados de la ANN - ReLU Softmax Adamax sobre la base de datos de la ADNI.

Matriz de Confusión: Caldas

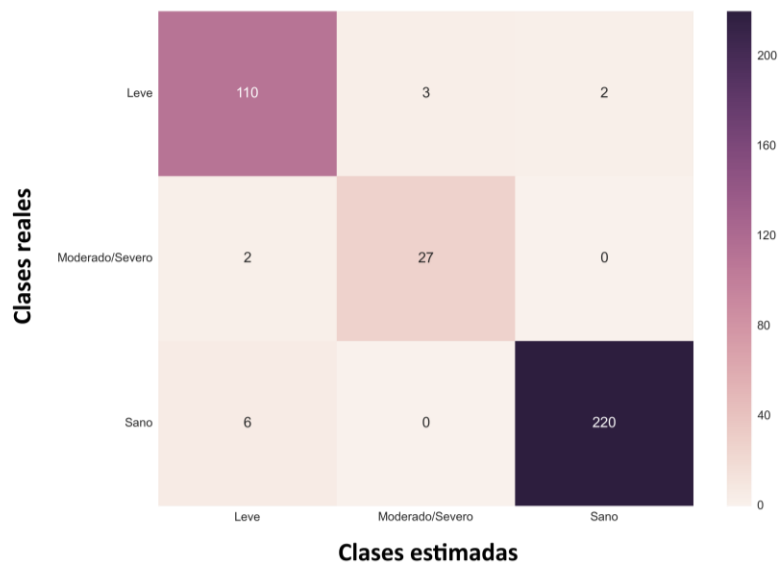


Ilustración 208 Matriz de confusión referente a los estimados de la ANN - ReLU Softmax Adamax sobre la base de datos de Caldas.

4.1.25.3 Curvas ROC

En las ilustraciones 209 a 214 se muestran las curvas ROC obtenidas para la *ANN – ReLU Softmax Adamax* sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se observa que el AUC para cada una de las clases se encuentra bastante cercana de 1, lo cual indica que la técnica de aprendizaje de máquina no presenta mayores dificultades para elegir la clase correcta al momento de realizar las predicciones. Solamente se presentan algunos inconvenientes menores para diferenciar las clases «Leve» y «Moderado/Severo» de las demás en la base de datos de la *ADNI*.

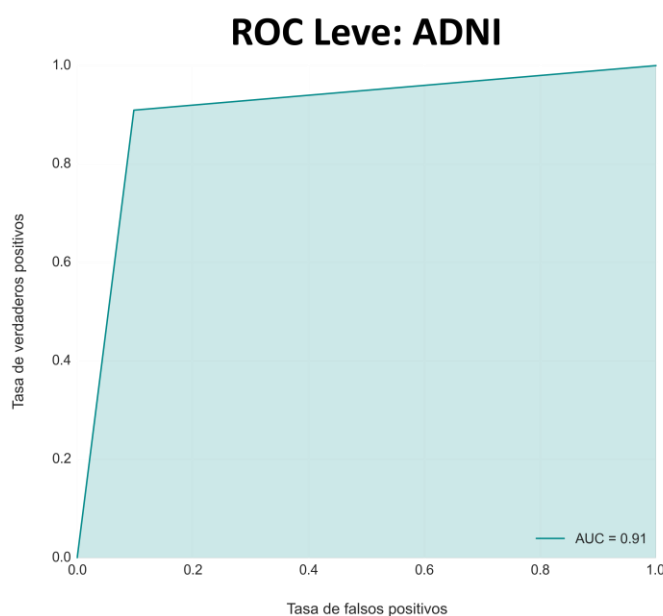


Ilustración 209 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - ReLU Softmax Adamax sobre la base de datos de la ADNI.

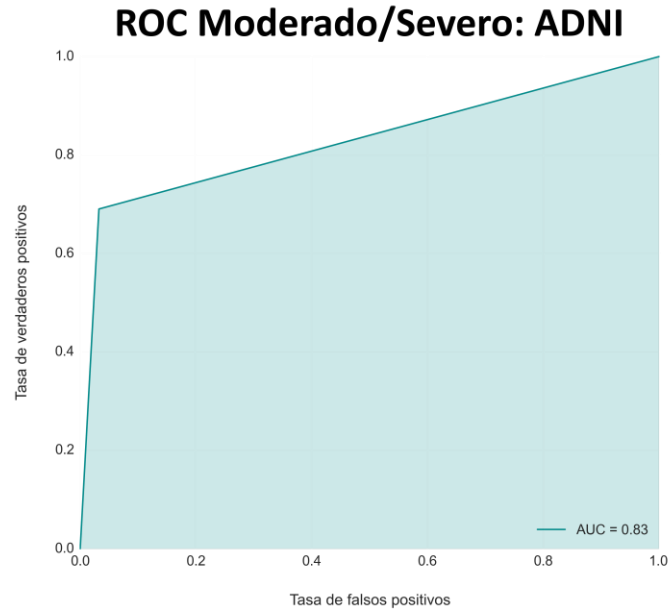


Ilustración 210 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - ReLU Softmax Adamax sobre la base de datos de la ADNI.

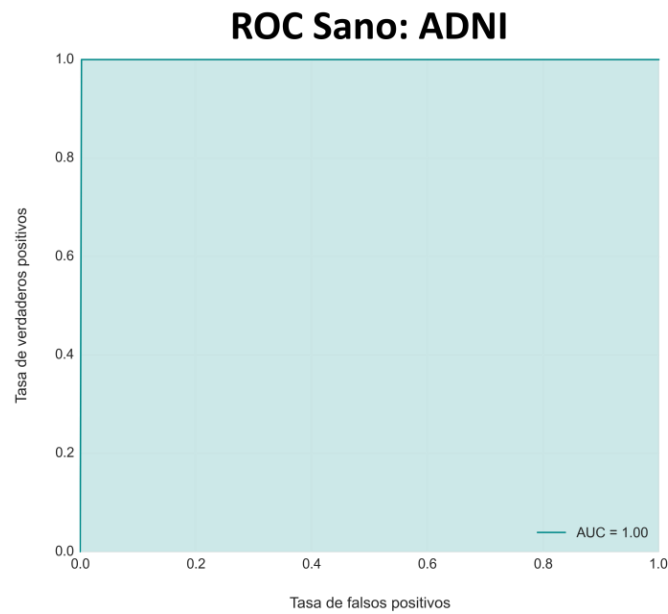


Ilustración 211 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - ReLU Softmax Adamax sobre la base de datos de la ADNI.

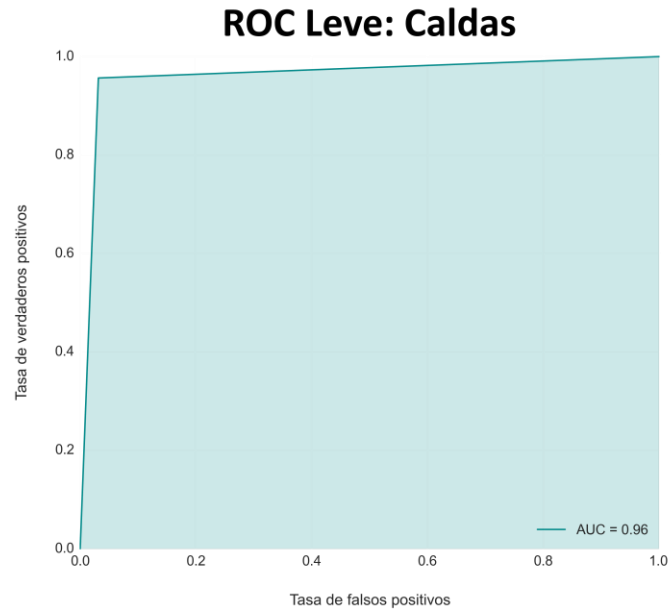


Ilustración 212 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - ReLU Softmax Adamax sobre la base de datos de Caldas.

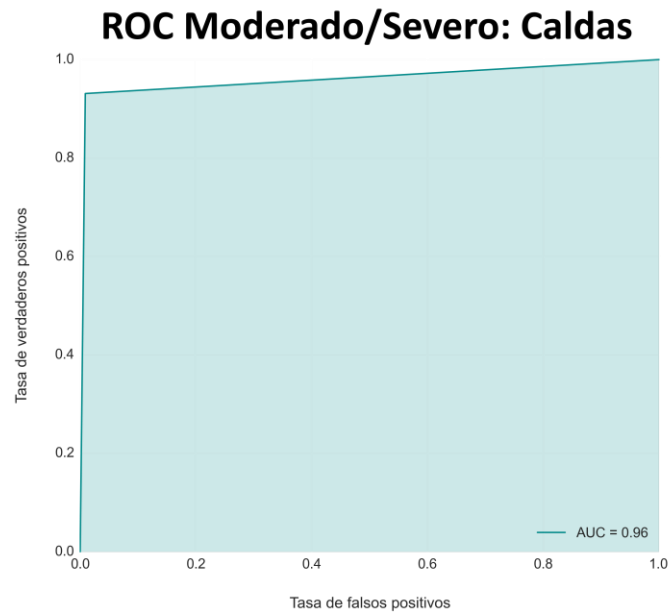


Ilustración 213 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - ReLU Softmax Adamax sobre la base de datos de Caldas.

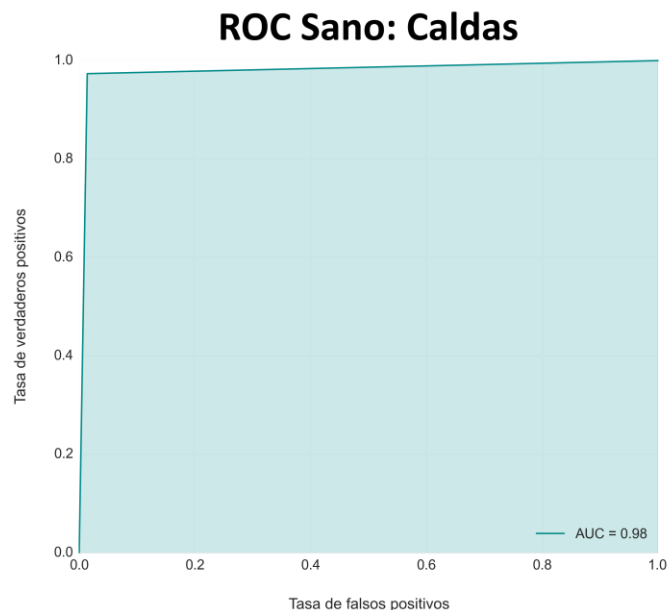


Ilustración 214 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - ReLU Softmax Adamax sobre la base de datos de Caldas.

4.1.25.4 Métricas de desempeño

A continuación se muestran las métricas de desempeño calculadas a partir de las matrices de confusión y de las curvas ROC para la ANN – ReLU Softmax Adamax.

ADNI		Caldas	
Exactitud	0,9042	Exactitud	0,9649
Precisión	0,9071	Precisión	0,9656
TPR (Sensibilidad)	0,9042	TPR (Sensibilidad)	0,9649
Especificidad	0,9633	Especificidad	0,9811
AUC Leve	0,9058	AUC Leve	0,9626
AUC Moderado/Severo	0,8290	AUC Moderado/Severo	0,9611
AUC Sano	0,9989	AUC Sano	0,9798
J Leve	0,8115	J Leve	0,9251
J Moderado/Severo	0,6580	J Moderado/Severo	0,9222
J Sano	0,9978	J Sano	0,9596
Kappa	0,8471	Kappa	0,9335

Tabla 28 Métricas de desempeño para la ANN – ReLU Softmax Adamax sobre las bases de datos de la ADNI y Caldas.

4.1.26 Técnica 25: *ANN - Sigmoid Softmax Adam*

La explicación de los temas relacionados con redes neuronales artificiales se encuentra dispuesta en la sección «4.1.18 Técnica 17: ANN – ELU Softmax Adam».

4.1.26.1 *Construcción, entrenamiento y clasificación*

La ANN se fundamenta en la estructura dispuesta en la ilustración 215. Esta se construye empleando la clase *Sequential* perteneciente a los modelos de *Keras*. Asimismo, se emplean las funciones de activación *Sigmoid* y *Softmax* —pertenecientes a las funciones de activación de *Keras*— para construir las neuronas artificiales de cada capa. A continuación, se compila la red neuronal empleando la función *compile(loss='categorical_crossentropy', optimizer='adam',...)* perteneciente a la clase *Sequential* de los modelos de *Keras*.

El entrenamiento de la ANN se realiza bajo la técnica de validación cruzada con 10 iteraciones empleando las funciones *StratifiedKFold(n_folds=10,...)* y *fit* pertenecientes al módulo *cross-validation* de *Scikit-learn* y a la clase *Sequential* de los modelos de *Keras* respectivamente. Finalmente, se generan los estimados para cada registro de ambas bases de datos (*ADNI* y *Caldas*) empleando la función *predict_classes* perteneciente a la clase *Sequential* de los modelos de *Keras*.

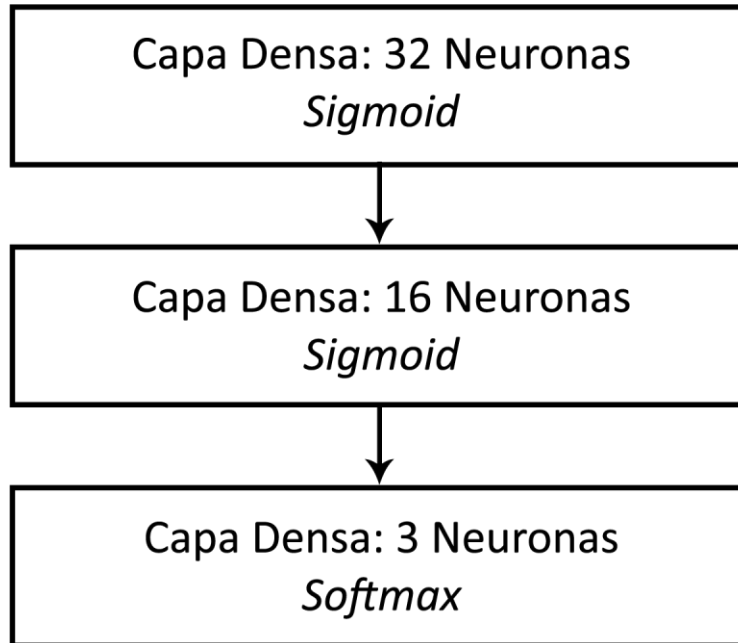


Ilustración 215 Estructura de la ANN – Sigmoid Softmax Adam de tres capas: dos capas ocultas de neuronas artificiales con funciones de activación Sigmoidea Dura y una capa de salida de neuronas artificiales con funciones de activación Softmax.

4.1.26.2 Matrices de confusión

En las ilustraciones 216 y 217 se muestran las matrices de confusión obtenidas para la ANN – Sigmoid Softmax Adam sobre las bases de datos de la ADNI y Caldas. A partir de estas se deduce que el clasificador logró un porcentaje de acierto del 0.90% y 0.97% en las respectivas predicciones. Asimismo, se puede observar que existe un bajo grado de confusión en las estimaciones de la técnica de aprendizaje de máquina respecto a las clases de predicción.

Matriz de Confusión: ADNI

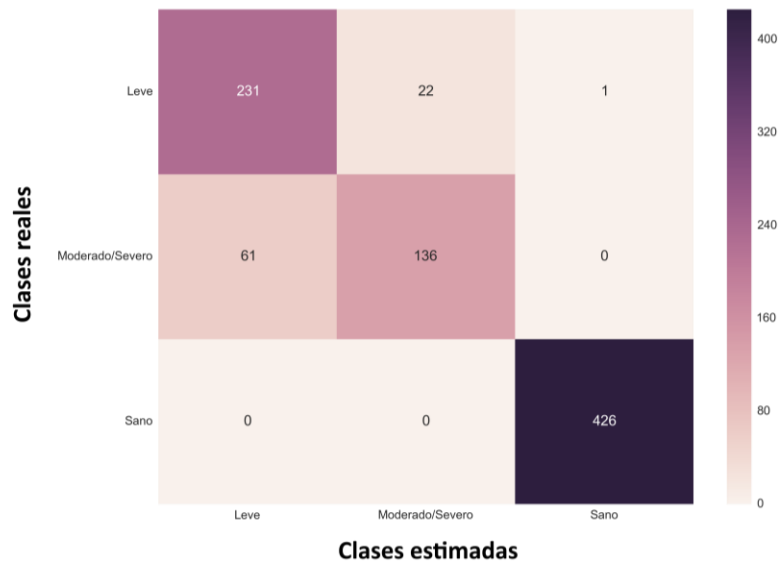


Ilustración 216 Matriz de confusión referente a los estimados de la ANN - Sigmoid Softmax Adam sobre la base de datos de la ADNI.

Matriz de Confusión: Caldas

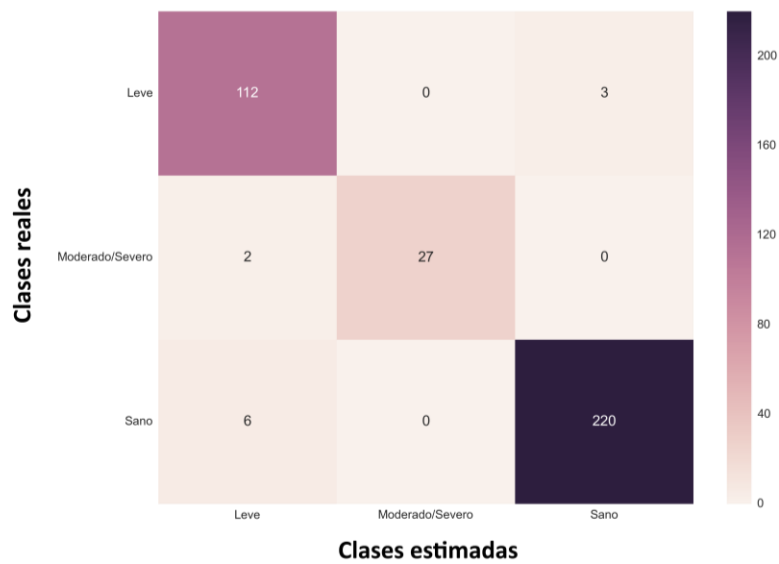


Ilustración 217 Matriz de confusión referente a los estimados de la ANN - Sigmoid Softmax Adam sobre la base de datos de Caldas.

4.1.26.3 Curvas ROC

En las ilustraciones 218 a 223 se muestran las curvas ROC obtenidas para la *ANN – Sigmoid Softmax Adam* sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se observa que el AUC para cada una de las clases se encuentra bastante cercana de 1, lo cual indica que la técnica de aprendizaje de máquina no presenta mayores dificultades para elegir la clase correcta al momento de realizar las predicciones. Solamente se presentan algunos inconvenientes menores para diferenciar las clases «Leve» y «Moderado/Severo» de las demás en la base de datos de la *ADNI*.

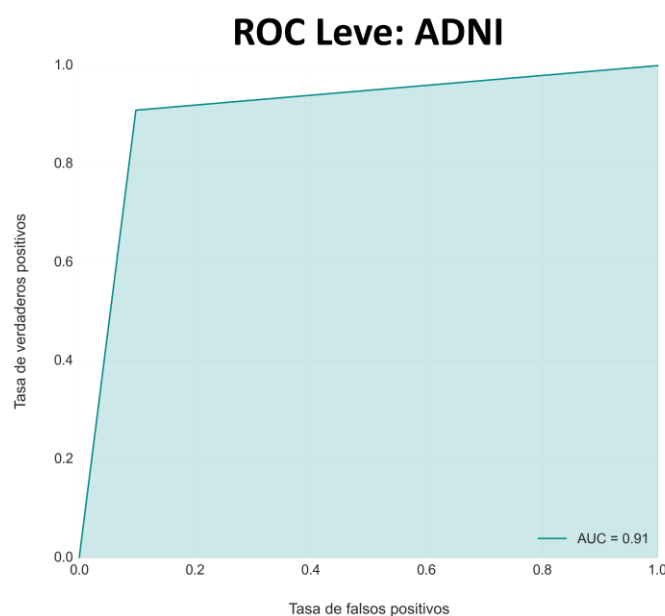


Ilustración 218 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - Sigmoid Softmax Adam sobre la base de datos de la ADNI.

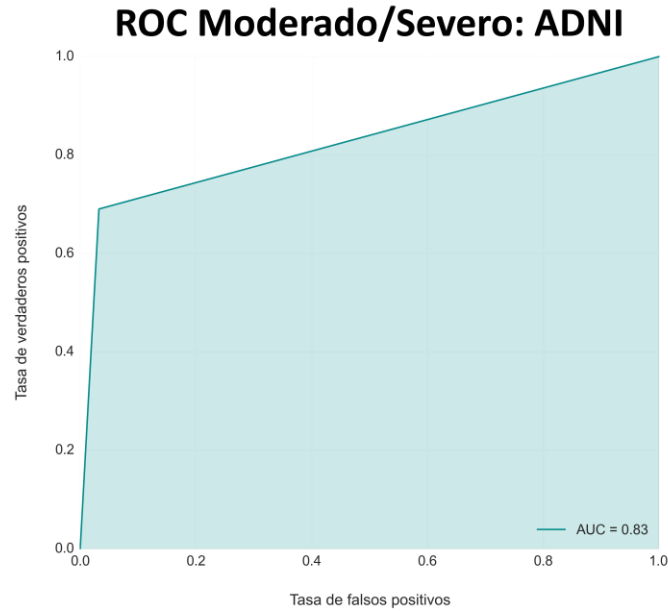


Ilustración 219 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - Sigmoid Softmax Adam sobre la base de datos de la ADNI.

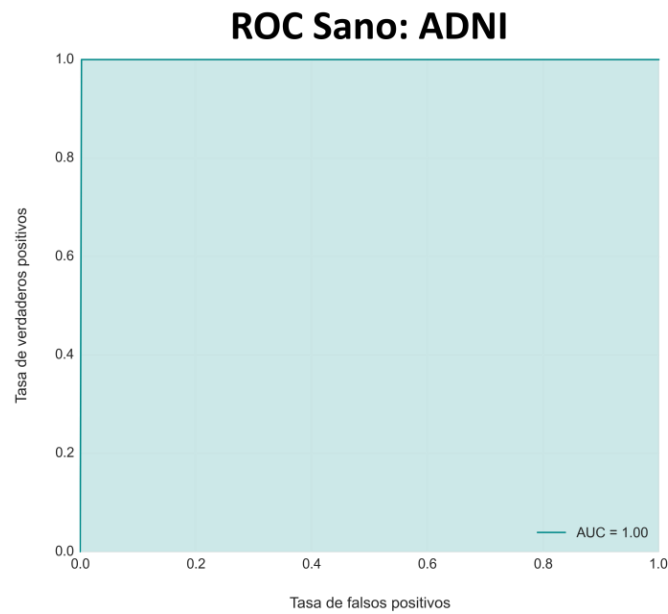


Ilustración 220 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - Sigmoid Softmax Adam sobre la base de datos de la ADNI.

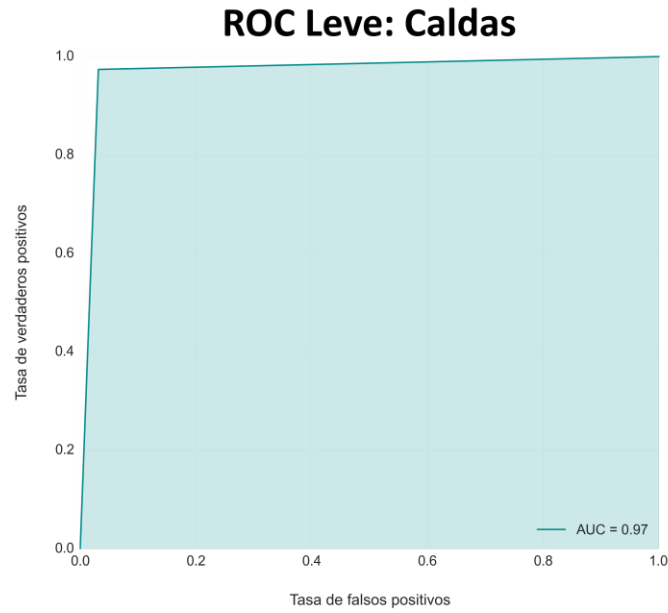


Ilustración 221 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - Sigmoid Softmax Adam sobre la base de datos de Caldas.

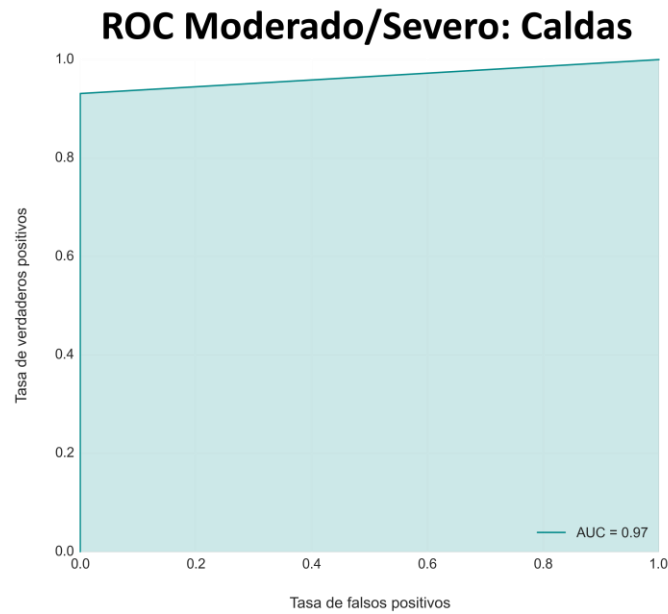


Ilustración 222 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - Sigmoid Softmax Adam sobre la base de datos de Caldas.

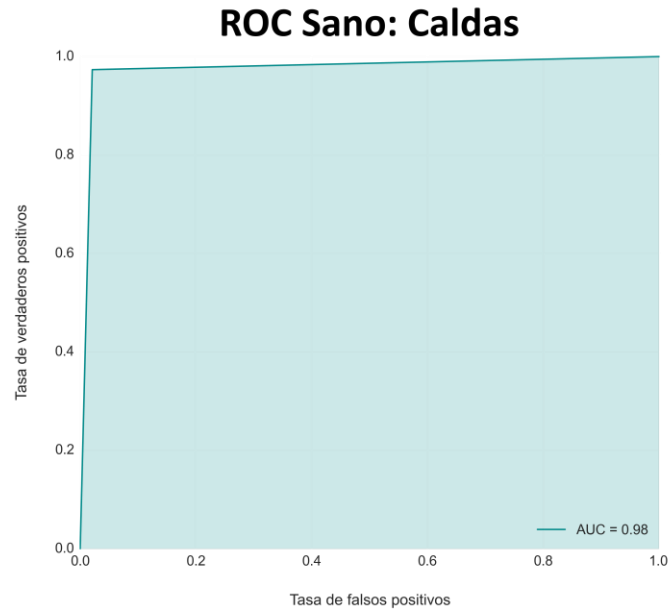


Ilustración 223 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - Sigmoid Softmax Adam sobre la base de datos de Caldas.

4.1.26.4 Métricas de desempeño

A continuación se muestran las métricas de desempeño calculadas a partir de las matrices de confusión y de las curvas ROC para la ANN – Sigmoid Softmax Adam.

ADNI		Caldas	
Exactitud	0,9042	Exactitud	0,9703
Precisión	0,9071	Precisión	0,9711
TPR (Sensibilidad)	0,9042	TPR (Sensibilidad)	0,9703
Especificidad	0,9633	Especificidad	0,9775
AUC Leve	0,9058	AUC Leve	0,9713
AUC Moderado/Severo	0,8290	AUC Moderado/Severo	0,9655
AUC Sano	0,9989	AUC Sano	0,9763
J Leve	0,8115	J Leve	0,9425
J Moderado/Severo	0,6580	J Moderado/Severo	0,9310
J Sano	0,9978	J Sano	0,9526
Kappa	0,8471	Kappa	0,9434

Tabla 29 Métricas de desempeño para la ANN – Sigmoid Softmax Adam sobre las bases de datos de la ADNI y Caldas.

4.1.27 Técnica 26: ANN - SReLU Softmax Adam

La explicación de los temas relacionados con redes neuronales artificiales se encuentra dispuesta en la sección «4.1.18 Técnica 17: ANN – ELU Softmax Adam».

4.1.27.1 Construcción, entrenamiento y clasificación

La ANN se fundamenta en la estructura dispuesta en la ilustración 224. Esta se construye empleando la clase *Sequential* perteneciente a los modelos de *Keras*. Asimismo, se emplean las funciones de activación *SReLU* y *Softmax* —pertenecientes a las funciones de activación de *Keras*— para construir las neuronas artificiales de cada capa. A continuación, se compila la red neuronal empleando la función *compile(loss='categorical_crossentropy', optimizer='adam',...)* perteneciente a la clase *Sequential* de los modelos de *Keras*.

El entrenamiento de la ANN se realiza bajo la técnica de validación cruzada con 10 iteraciones empleando las funciones *StratifiedKFold(n_folds=10,...)* y *fit* pertenecientes al módulo *cross-validation* de *Scikit-learn* y a la clase *Sequential* de los modelos de *Keras* respectivamente. Finalmente, se generan los estimados para cada registro de ambas bases de datos (*ADNI* y *Caldas*) empleando la función *predict_classes* perteneciente a la clase *Sequential* de los modelos de *Keras*.

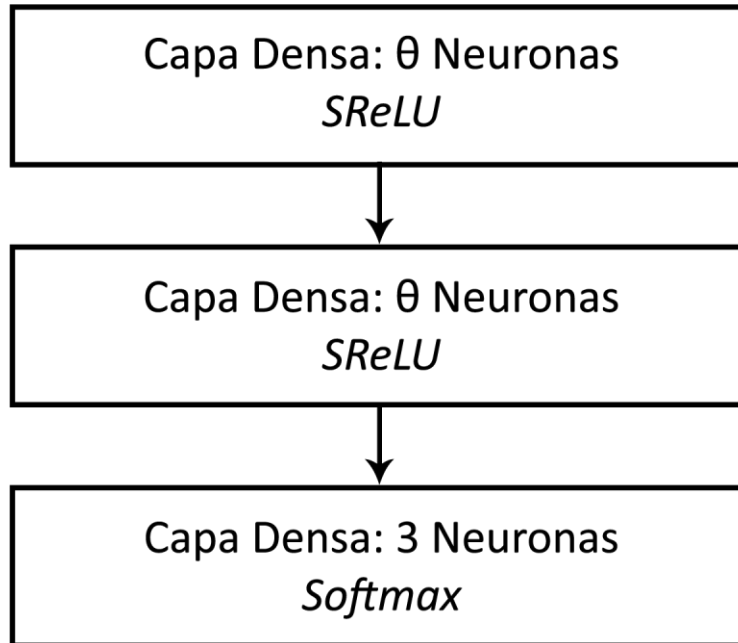


Ilustración 224 Estructura de la ANN – SReLU Softmax Adam de tres capas: dos capas ocultas de neuronas artificiales con funciones de activación SReLU y una capa de salida de neuronas artificiales con funciones de activación Softmax. La cantidad de neuronas θ es equivalente a la cantidad de entradas de la ANN.

4.1.27.2 Matrices de confusión

En las ilustraciones 225 y 226 se muestran las matrices de confusión obtenidas para la ANN – SReLU Softmax Adam sobre las bases de datos de la ADNI y Caldas. A partir de estas se deduce que el clasificador logró un porcentaje de acierto del 0.89% y 0.95% en las respectivas predicciones. Asimismo, se puede observar que existe un bajo grado de confusión en las estimaciones de la técnica de aprendizaje de máquina respecto a las clases de predicción.

Matriz de Confusión: ADNI

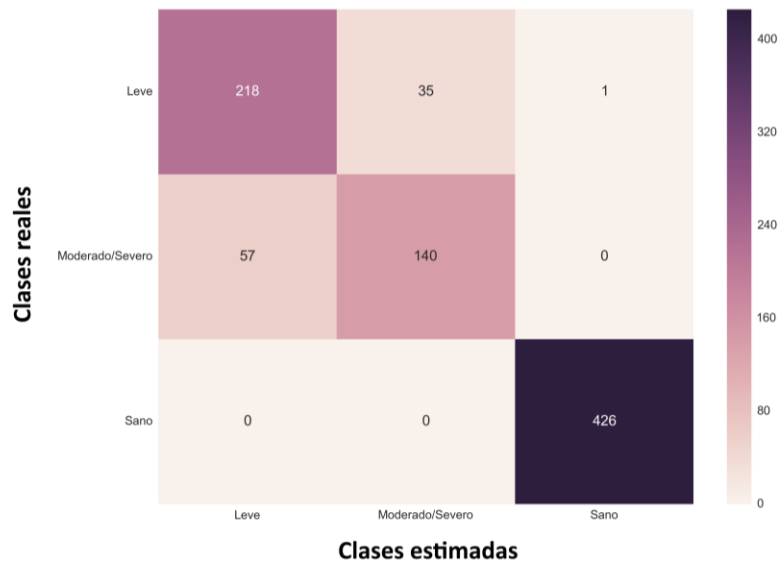


Ilustración 225 Matriz de confusión referente a los estimados de la ANN - SReLU Softmax Adam sobre la base de datos de la ADNI.

Matriz de Confusión: Caldas

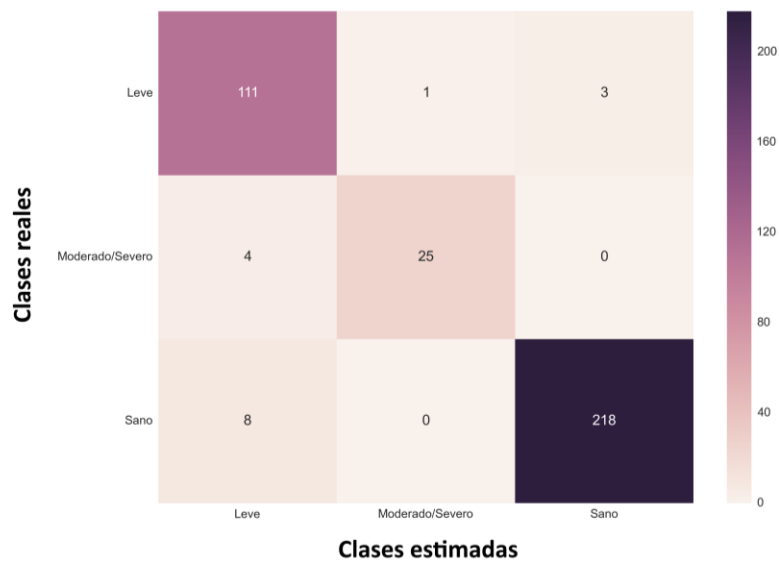


Ilustración 226 Matriz de confusión referente a los estimados de la ANN - SReLU Softmax Adam sobre la base de datos de Caldas.

4.1.27.3 Curvas ROC

En las ilustraciones 227 a 232 se muestran las curvas ROC obtenidas para la *ANN – SReLU Softmax Adam* sobre las bases de datos de la *ADNI* y *Caldas*. A partir de estas se observa que el AUC para cada una de las clases se encuentra bastante cercana de 1, lo cual indica que la técnica de aprendizaje de máquina no presenta mayores dificultades para elegir la clase correcta al momento de realizar las predicciones. Solamente se presentan algunos inconvenientes menores para diferenciar las clases «Leve» y «Moderado/Severo» de las demás en la base de datos de la *ADNI*.

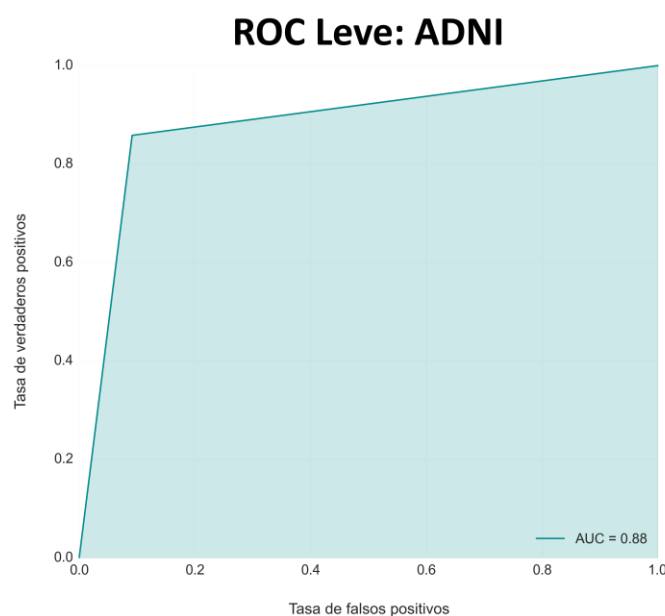


Ilustración 227 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - SReLU Softmax Adam sobre la base de datos de la ADNI.

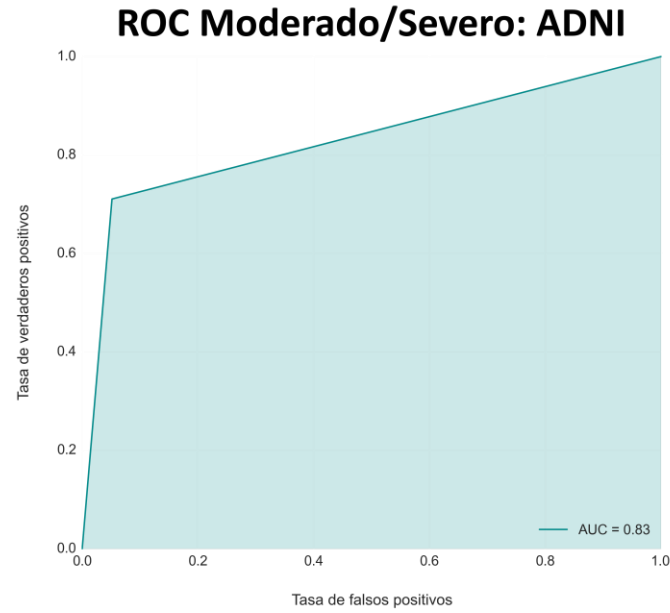


Ilustración 228 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - SReLU Softmax Adam sobre la base de datos de la ADNI.

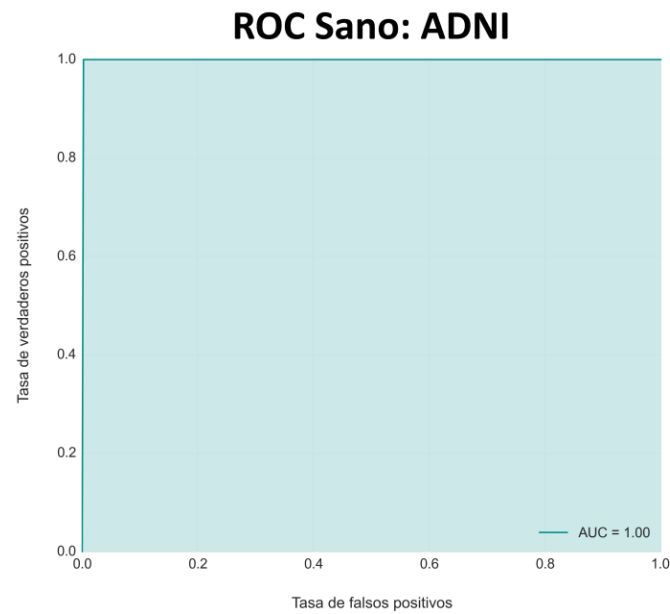


Ilustración 229 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - SReLU Softmax Adam sobre la base de datos de la ADNI.

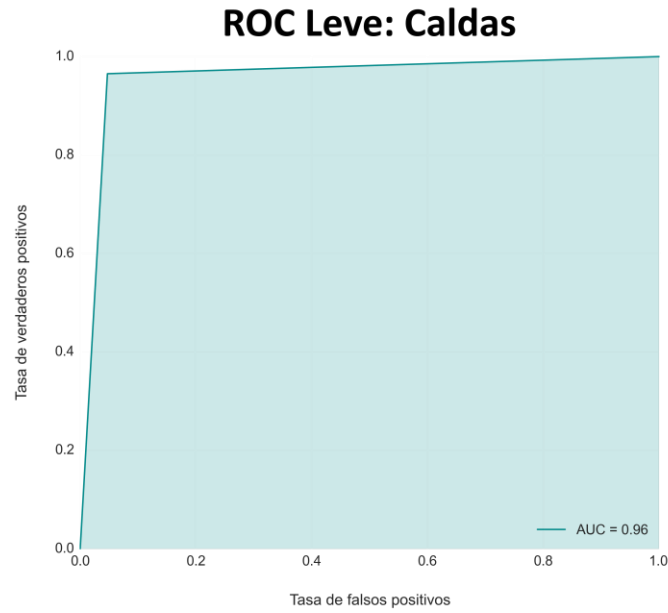


Ilustración 230 Curva ROC —de clase positiva = Leve y clases negativas = Moderado/Severo, Sano— referente a los estimados de la ANN - SReLU Softmax Adam sobre la base de datos de Caldas.

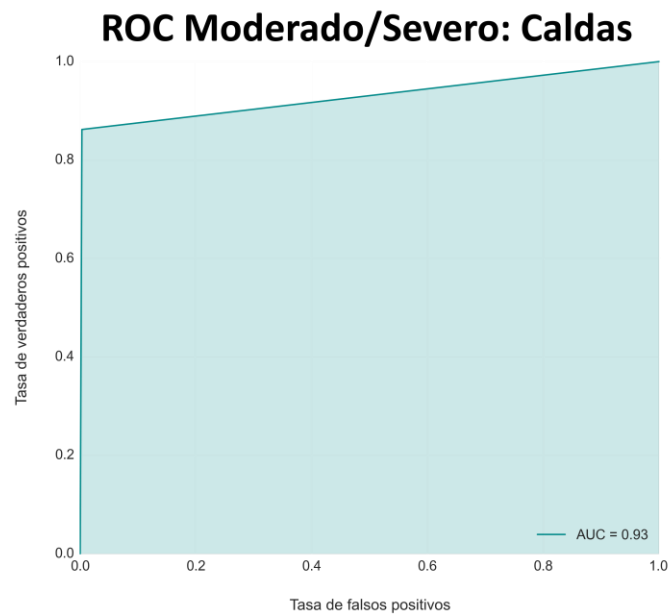


Ilustración 231 Curva ROC —de clase positiva = Moderado/Severo y clases negativas = Leve, Sano— referente a los estimados de la ANN - SReLU Softmax Adam sobre la base de datos de Caldas.

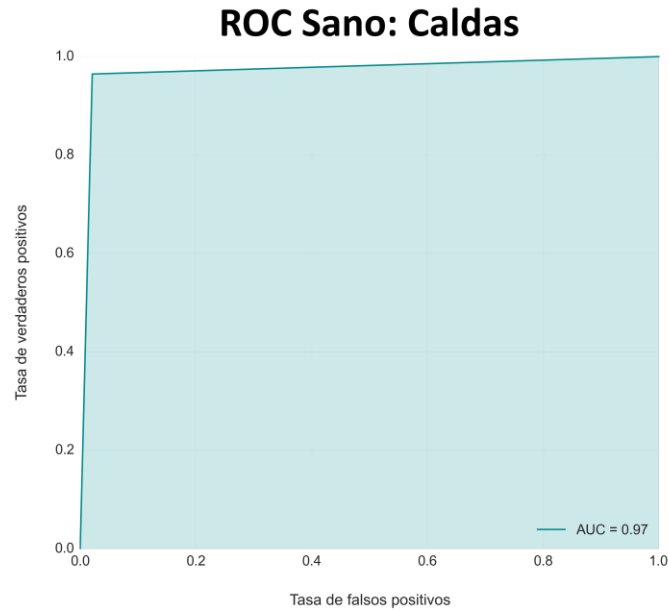


Ilustración 232 Curva ROC —de clase positiva = Sano y clases negativas = Leve, Moderado/Severo — referente a los estimados de la ANN - SReLU Softmax Adam sobre la base de datos de Caldas.

4.1.27.4 Métricas de desempeño

A continuación se muestran las métricas de desempeño calculadas a partir de las matrices de confusión y de las curvas ROC para la ANN – SReLU Softmax Adam.

ADNI		Caldas	
Exactitud	0,8940	Exactitud	0,9568
Precisión	0,8940	Precisión	0,9584
TPR (Sensibilidad)	0,8940	TPR (Sensibilidad)	0,9568
Especificidad	0,8940	Especificidad	0,9724
AUC Leve	0,8940	AUC Leve	0,9591
AUC Moderado/Severo	0,8940	AUC Moderado/Severo	0,9296
AUC Sano	0,8940	AUC Sano	0,9719
J Leve	0,8940	J Leve	0,9182
J Moderado/Severo	0,8940	J Moderado/Severo	0,8591
J Sano	0,8940	J Sano	0,9438
Kappa	0,8940	Kappa	0,9178

Tabla 30 Métricas de desempeño para la ANN – SReLU Softmax Adam sobre las bases de datos de la ADNI y Caldas.

4.1.28 Síntesis del desempeño de las técnicas de aprendizaje de máquina

En el estudio comparativo se incluyeron diversas técnicas simbólicas y subsimbólicas de aprendizaje de máquina entre clasificadores lineales, máquinas de soporte vectorial, árboles de decisión, clasificadores en ensambles y redes neuronales artificiales. Cada una fue entrenada con los registros de ambas bases de datos (*ADNI* y *Caldas*). Posteriormente, se extrajeron las respectivas matrices de confusión y las curvas ROC para cada experimento. Por último, se calcularon las siguientes métricas de desempeño: exactitud, precisión, sensibilidad, especificidad, área bajo la curva ROC, *Youden's J* y *Cohen's Kappa*.

Las tablas 31 y 32 muestran el consolidado de las métricas de desempeño de las técnicas de aprendizaje de máquina sobre las bases de datos de la *ADNI* y *Caldas* respectivamente. En estas, cada columna refiere una métrica de desempeño diferente y se encuentra resaltada en una escala de colores para facilitar su interpretación: el color rojo está reservado para los valores más bajos de la respectiva columna; el color verde está reservado para los valores más altos de la respectiva columna; el color blanco está reservado para los valores intermedios de la respectiva columna. Por otra parte, el color azul en la primera columna indica las técnicas que presentaron un desempeño sobresaliente respecto de las demás —de acuerdo al panorama brindado por todas las métricas de desempeño—.

De acuerdo con el consolidado de las métricas de desempeño se puede observar que la mayoría de las técnicas de aprendizaje de máquina son bastante efectivas para diferenciar las personas sanas de las que padecen algún nivel de deterioro cognitivo. Por otro lado, las técnicas que presentaron un desempeño sobresaliente también distinguen con facilidad las personas que padecen de Deterioro Cognitivo Leve de las demás.

4.1.28.1 Síntesis del desempeño de las técnicas de aprendizaje de máquina para la base de datos de la ADNl

Técnica \ Métricas	Exactitud	Precisión	Sensibilidad	Especificidad	AUC Leve	AUC Moderado/Severo	AUC Sano	J Leve	J Moderado/Severo	J Sano	Kappa
<i>Passive-Aggressive Classifier</i>	0,5017	0,5267	0,5017	0,7433	0,5378	0,6330	0,6682	0,0756	0,2660	0,3363	0,2265
<i>Ridge Classifier</i>	0,8347	0,8481	0,8347	0,9460	0,7612	0,8220	0,9989	0,5224	0,6440	0,9978	0,7393
<i>Linear SVM</i>	0,8951	0,8995	0,8951	0,9596	0,9614	0,9643	0,9987	0,7987	0,6225	0,9954	0,8325
<i>Quadratic SVM</i>	0,8894	0,8927	0,8894	0,9578	0,9394	0,9517	0,9987	0,7813	0,6115	0,9954	0,8235
<i>Cubic SVM</i>	0,8803	0,8802	0,8803	0,9561	0,8943	0,9495	0,9988	0,7335	0,6250	0,9954	0,8094
<i>Coarse Gaussian SVM</i>	0,9031	0,9252	0,9031	0,9600	0,9669	0,9639	0,9987	0,8589	0,5823	0,9954	0,8446
<i>Medium Gaussian SVM</i>	0,8962	0,9043	0,8962	0,9592	0,9515	0,9581	0,9988	0,8143	0,6059	0,9954	0,8341
<i>Fine Gaussian SVM</i>	0,8119	0,8029	0,8119	0,8877	0,9083	0,9036	0,9959	0,6077	0,5046	0,8444	0,6926
<i>Decision Tree</i>	0,9019	0,9036	0,9019	0,9630	0,9525	0,9540	0,9965	0,7967	0,6760	0,9931	0,8438
<i>Decision Trees (Bagging)</i>	0,9008	0,9006	0,9008	0,9635	0,9665	0,9643	0,9987	0,7787	0,6860	0,9978	0,8421
<i>Decision Trees (Boosting)</i>	0,9019	0,9031	0,9019	0,9633	0,9680	0,9646	0,9955	0,7920	0,6832	0,9931	0,8439
<i>AdaBoost-SAMME.R</i>	0,8814	0,9112	0,8814	0,9513	0,9130	0,7422	0,9989	0,8261	0,4844	0,9978	0,8094
<i>Random Forest</i>	0,8757	0,8754	0,8757	0,9554	0,8531	0,8186	0,9977	0,7061	0,6371	0,9954	0,8025
<i>Extra Trees</i>	0,8746	0,8742	0,8746	0,9546	0,8558	0,8106	0,9977	0,7115	0,6212	0,9954	0,8005
<i>RUSBoosted Trees</i>	0,8883	0,8884	0,8883	0,9591	0,9647	0,9659	0,9981	0,7494	0,6584	0,9931	0,8223
<i>Gradient Boosting</i>	0,8905	0,8907	0,8905	0,9596	0,8798	0,8263	0,9977	0,7596	0,6526	0,9954	0,8257
<i>ELU Softmax Adam ANN</i>	0,8814	0,8823	0,8814	0,9556	0,8757	0,7999	0,9989	0,7515	0,5997	0,9978	0,8108
<i>Hard Sigmoid Softmax Adam ANN</i>	0,9042	0,9075	0,9042	0,9632	0,9069	0,8272	0,9989	0,8139	0,6544	0,9978	0,8471
<i>PReLU Softmax Adam ANN</i>	0,8814	0,8823	0,8814	0,9556	0,8757	0,7999	0,9989	0,7515	0,5997	0,9978	0,8108
<i>ReLU Softmax Adadelta ANN</i>	0,9031	0,9033	0,9031	0,9640	0,8945	0,8427	0,9989	0,7889	0,6854	0,9978	0,8456
<i>ReLU Softmax Adagrad ANN</i>	0,9031	0,9066	0,9031	0,9627	0,9061	0,8247	0,9989	0,8123	0,6493	0,9978	0,8453
<i>ReLU Softmax Adam ANN</i>	0,8974	0,9043	0,8974	0,9598	0,9068	0,8048	0,9989	0,8136	0,6095	0,9978	0,8359
<i>ReLU Softmax Adam Deep ANN</i>	0,6921	0,7308	0,6921	0,7388	0,6964	0,6641	0,7506	0,3928	0,3282	0,5011	0,4608
<i>ReLU Softmax Adamax ANN</i>	0,9042	0,9071	0,9042	0,9633	0,9058	0,8290	0,9989	0,8115	0,6580	0,9978	0,8471
<i>Sigmoid Softmax Adam ANN</i>	0,9042	0,9071	0,9042	0,9633	0,9058	0,8290	0,9989	0,8115	0,6580	0,9978	0,8471
<i>SReLU Softmax Adam ANN</i>	0,8940	0,8939	0,8940	0,9609	0,8834	0,8296	0,9989	0,7668	0,6592	0,9978	0,8311

Tabla 31 Consolidado de las métricas de desempeño de las técnicas de aprendizaje de máquina sobre la base de datos de la ADNl.

4.1.28.2 Síntesis del desempeño de las técnicas de aprendizaje de máquina para la base de datos de Caldas

Técnica \ Métricas	Exactitud	Precisión	Sensibilidad	Especificidad	AUC Leve	AUC Moderado/Severo	AUC Sano	J Leve	J Moderado/Severo	J Sano	Kappa
<i>Passive-Aggressive Classifier</i>	0,6811	0,7636	0,6811	0,8198	0,7304	0,5818	0,7823	0,4609	0,1636	0,5646	0,4342
<i>Ridge Classifier</i>	0,8622	0,8655	0,8622	0,8502	0,8308	0,7241	0,8860	0,6616	0,4483	0,7721	0,7218
<i>Linear SVM</i>	0,9622	0,9635	0,9622	0,9789	0,9716	0,9864	0,9798	0,9260	0,9597	0,9463	0,9287
<i>Quadratic SVM</i>	0,9676	0,9681	0,9676	0,9773	0,9765	0,9825	0,9823	0,9338	0,9626	0,9482	0,9385
<i>Cubic SVM</i>	0,9405	0,9419	0,9405	0,9530	0,9632	0,9711	0,9796	0,8755	0,8591	0,9071	0,8867
<i>Coarse Gaussian SVM</i>	0,9324	0,9419	0,9324	0,9635	0,9739	0,9813	0,9799	0,8924	0,5172	0,9463	0,8703
<i>Medium Gaussian SVM</i>	0,9405	0,9456	0,9405	0,9681	0,9734	0,9774	0,9778	0,8994	0,6867	0,9419	0,8869
<i>Fine Gaussian SVM</i>	0,7108	0,7901	0,7108	0,5522	0,9209	0,8296	0,9583	0,3226	0,0000	0,2664	0,3141
<i>Decision Tree</i>	0,9676	0,9676	0,9676	0,9642	0,9816	0,9779	0,9850	0,9195	0,9655	0,9337	0,9378
<i>Decision Trees (Bagging)</i>	0,9730	0,9732	0,9730	0,9797	0,9755	0,9786	0,9823	0,9417	0,9626	0,9570	0,9486
<i>Decision Trees (Boosting)</i>	0,3108	0,7858	0,3108	0,6892	0,4611	0,5756	0,5121	0,0000	0,0000	0,0000	0,0000
<i>AdaBoost-SAMME.R</i>	0,8135	0,8625	0,8135	0,9108	0,8575	0,5675	0,9023	0,7151	0,1350	0,8047	0,6551
<i>Random Forest</i>	0,9514	0,9537	0,9514	0,9700	0,9552	0,9123	0,9697	0,9103	0,8247	0,9393	0,9076
<i>Extra Trees</i>	0,9514	0,9518	0,9514	0,9689	0,9523	0,9094	0,9706	0,9047	0,8188	0,9412	0,9074
<i>RUSBoosted Trees</i>	0,9703	0,9704	0,9703	0,9755	0,9728	0,9787	0,9797	0,9330	0,9626	0,9501	0,9434
<i>Gradient Boosting</i>	0,9730	0,9732	0,9730	0,9757	0,9708	0,9828	0,9750	0,9417	0,9655	0,9501	0,9485
<i>ELU Softmax Adam ANN</i>	0,9297	0,9297	0,9297	0,9461	0,9251	0,8074	0,9612	0,8503	0,6148	0,9223	0,8638
<i>Hard Sigmoid Softmax Adam ANN</i>	0,9649	0,9656	0,9649	0,9801	0,9650	0,9453	0,9798	0,9299	0,8907	0,9596	0,9333
<i>PReLU Softmax Adam ANN</i>	0,9297	0,9297	0,9297	0,9461	0,9251	0,8074	0,9612	0,8503	0,6148	0,9223	0,8638
<i>ReLU Softmax Adadelta ANN</i>	0,9541	0,9545	0,9541	0,9621	0,9500	0,9468	0,9637	0,8999	0,8936	0,9274	0,9123
<i>ReLU Softmax Adagrad ANN</i>	0,9514	0,9524	0,9514	0,9569	0,9480	0,9138	0,9624	0,8960	0,8276	0,9248	0,9065
<i>ReLU Softmax Adam ANN</i>	0,9595	0,9602	0,9595	0,9756	0,9563	0,9611	0,9741	0,9125	0,9222	0,9482	0,9232
<i>ReLU Softmax Adam Deep ANN</i>	0,8595	0,8744	0,8595	0,7955	0,8182	0,8261	0,8324	0,6365	0,6522	0,6648	0,7084
<i>ReLU Softmax Adamax ANN</i>	0,9649	0,9656	0,9649	0,9811	0,9626	0,9611	0,9798	0,9251	0,9222	0,9596	0,9335
<i>Sigmoid Softmax Adam ANN</i>	0,9703	0,9711	0,9703	0,9775	0,9713	0,9655	0,9763	0,9425	0,9310	0,9526	0,9434
<i>SReLU Softmax Adam ANN</i>	0,9568	0,9584	0,9568	0,9724	0,9591	0,9296	0,9719	0,9182	0,8591	0,9438	0,9178

Tabla 32 Consolidado de las métricas de desempeño de las técnicas de aprendizaje de máquina sobre la base de datos de Caldas.

4.1.29 Elección de las técnicas de aprendizaje de máquina

El primer paso para elegir las técnicas de aprendizaje de máquina implicó extraer aquellas que presentaron un excelente desempeño en el diagnóstico asistido de Deterioro Cognitivo Leve. Esto se llevó a cabo al observar y contrastar la primera columna de las tablas 31 y 32 que indican cuáles técnicas tuvieron un desempeño sobresaliente en dicha tarea respecto de las demás:

ADNI	Caldas
<i>Passive-Aggressive Classifier</i>	<i>Passive-Aggressive Classifier</i>
<i>Ridge Classifier</i>	<i>Ridge Classifier</i>
<i>Linear SVM</i>	<i>Linear SVM</i>
<i>Quadratic SVM</i>	<i>Quadratic SVM</i>
<i>Cubic SVM</i>	<i>Cubic SVM</i>
<i>Coarse Gaussian SVM</i>	<i>Coarse Gaussian SVM</i>
<i>Medium Gaussian SVM</i>	<i>Medium Gaussian SVM</i>
<i>Fine Gaussian SVM</i>	<i>Fine Gaussian SVM</i>
<i>Decision Tree</i>	<i>Decision Tree</i>
<i>Decision Trees (Bagging)</i>	<i>Decision Trees (Bagging)</i>
<i>Decision Trees (Boosting)</i>	<i>Decision Trees (Boosting)</i>
<i>AdaBoost-SAMME.R</i>	<i>AdaBoost-SAMME.R</i>
<i>Random Forest</i>	<i>Random Forest</i>
<i>Extra Trees</i>	<i>Extra Trees</i>
<i>RUSBoosted Trees</i>	<i>RUSBoosted Trees</i>
<i>Gradient Boosting</i>	<i>Gradient Boosting</i>
<i>ELU Softmax Adam ANN</i>	<i>ELU Softmax Adam ANN</i>
<i>Hard Sigmoid Softmax Adam ANN</i>	<i>Hard Sigmoid Softmax Adam ANN</i>
<i>PReLU Softmax Adam ANN</i>	<i>PReLU Softmax Adam ANN</i>
<i>ReLU Softmax Adadelta ANN</i>	<i>ReLU Softmax Adadelta ANN</i>
<i>ReLU Softmax Adagrad ANN</i>	<i>ReLU Softmax Adagrad ANN</i>
<i>ReLU Softmax Adam ANN</i>	<i>ReLU Softmax Adam ANN</i>
<i>ReLU Softmax Adam Deep ANN</i>	<i>ReLU Softmax Adam Deep ANN</i>
<i>ReLU Softmax Adamax ANN</i>	<i>ReLU Softmax Adamax ANN</i>
<i>Sigmoid Softmax Adam ANN</i>	<i>Sigmoid Softmax Adam ANN</i>
<i>SReLU Softmax Adam ANN</i>	<i>SReLU Softmax Adam ANN</i>

Tabla 33 Símil de las técnicas de aprendizaje de máquina de acuerdo con el desempeño obtenido en el diagnóstico asistido de Deterioro Cognitivo Leve para cada una de las bases de datos (ADNI y Caldas). El color azul indica las técnicas que presentaron un desempeño sobresaliente respecto de las demás en concordancia con las tablas 31 y 32.

Lo anterior permitió obtener un conjunto mucho más reducido de técnicas de aprendizaje de máquina las cuales poseen un alto potencial para el diagnóstico asistido de Deterioro Cognitivo Leve:

Técnica de aprendizaje de máquina	Simbólica	Subsimbólica
<i>Decision Tree</i>	✓	
<i>Decision Trees (Bagging)</i>	✓	
<i>Hard Sigmoid Softmax Adam ANN</i>		✓
<i>ReLU Softmax Adamax ANN</i>		✓
<i>Sigmoid Softmax Adam ANN</i>		✓

Tabla 34 Conjunto reducido de técnicas de aprendizaje de máquina con un alto potencial para el diagnóstico asistido de Deterioro Cognitivo Leve. Las técnicas basadas en árboles de decisión son simbólicas mientras que las basadas en redes neuronales artificiales son subsimbólicas.

La pretensión era elegir una sola técnica de cada categoría (simbólica y subsimbólica) para integrarlas en un enfoque híbrido. No obstante, las cinco técnicas mencionadas en la tabla 34 presentaron métricas de desempeño bastante similares, incluso en aquellas mediciones que no sufren de sesgo debido al desbalance de las clases (*Youden's J* y *Cohen's Kappa*) (Powers, 2015). Por consiguiente, se incorporó un último criterio de decisión basado en el costo computacional de las técnicas de aprendizaje de máquina.

En relación con las técnicas simbólicas Blockeel & Struyf (2002) refieren el alto costo computacional del *Bagging*; se hace evidente que entrenar n clasificadores de cualquier tipo es más costoso que entrenar un clasificador individual. En relación con las técnicas subsimbólicas Glorot, Bordes, & Bengio (2011) y Krizhevsk, Sutskever, & Hinton (2012) manifiestan que los cálculos con unidades de rectificadores lineales (ReLU) son más eficientes que los cálculos con unidades sigmoideas. De acuerdo con lo anterior, se eligieron finalmente las técnicas *Decision Tree* y *ReLU Softmax Adamax ANN* teniendo en cuenta que las redes neuronales confrontadas poseían la misma cantidad de neuronas artificiales en cada capa.

4.2 Fase 2: construcción del subsistema computacional tipo A

El subsistema computacional tipo A se encuentra basado en la técnica simbólica de aprendizaje de máquina denominada árbol de decisión.

4.2.1 Extracción de los requerimientos del subsistema computacional tipo A

Los requerimientos del subsistema computacional tipo A se extrajeron en las siguientes historias de usuario:

- Como médico quiero saber si una persona sufre de Deterioro Cognitivo Leve para poder iniciar el tratamiento adecuado lo antes posible.
- Como administrador quiero entrenar un árbol de decisión para brindar una herramienta de asistencia al diagnóstico de Deterioro Cognitivo Leve.
- Como administrador quiero guardar y cargar un árbol de decisión entrenado, para evitar la pérdida de información al momento de cerrar la herramienta.
- Como administrador quiero observar el desempeño de un árbol de decisión entrenado, para determinar su efectividad en la asistencia al diagnóstico de Deterioro Cognitivo Leve.

4.2.2 Análisis de los requerimientos del subsistema computacional tipo A

Identificador	REQ-A01
Nombre	Importación de variables
Descripción	El subsistema deberá leer e importar un conjunto de datos desde un archivo CSV. Cada columna del archivo (separada por el carácter «,») representa una variable diferente. El archivo debe cumplir con la plantilla descrita en el anexo 8 o la plantilla descrita en el anexo 9
Entradas	- <i>R</i> : ruta física del archivo CSV a importar
Salidas	- <i>M</i> : arreglo cargado con la totalidad de los datos del archivo CSV

Tabla 35 Requerimiento funcional N° 1 para el subsistema computacional tipo A.



Identificador	REQ-A02
Nombre	Entrenamiento de árbol de decisión
Descripción	El subsistema deberá entrenar un árbol de decisión para asistir al diagnóstico de Deterioro Cognitivo Leve empleando un conjunto de características y etiquetas
Entradas	- X: arreglo cargado con todas las características - Y: arreglo cargado con todas las etiquetas - Z: variable que designa las posibles clases de predicción
Salidas	- A: árbol de decisión entrenado para asistir al diagnóstico de Deterioro Cognitivo Leve

Tabla 36 Requerimiento funcional N° 2 para el subsistema computacional tipo A.

Identificador	REQ-A03
Nombre	Asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión
Descripción	El subsistema deberá indicar la probabilidad de que una persona padezca de Deterioro Cognitivo Leve por medio de un árbol de decisión
Entradas	- A: árbol de decisión entrenado para asistir al diagnóstico de Deterioro Cognitivo Leve - X: arreglo cargado con todas las características de la persona - Z: variable que designa las posibles clases de predicción
Salidas	- Pr: arreglo cargado con las probabilidades de que la persona pertenezca a cada una de las clases designadas por Z

Tabla 37 Requerimiento funcional N° 3 para el subsistema computacional tipo A.

Identificador	REQ-A04
Nombre	Persistencia de árbol de decisión
Descripción	El subsistema deberá persistir un árbol de decisión
Entradas	- A: árbol de decisión a persistir - R: ruta física donde se guarda el árbol de decisión

Salidas	- <i>W</i> : archivo que contiene el resultado de la persistencia de <i>A</i>
----------------	---

Tabla 38 Requerimiento funcional N° 4 para el subsistema computacional tipo A.

Identificador	REQ-A05
Nombre	Cargue de árbol de decisión
Descripción	El subsistema deberá cargar un árbol de decisión
Entradas	- <i>R</i> : ruta física donde se encuentra guardado el árbol de decisión
Salidas	- <i>A</i> : árbol de decisión cargado a partir de la información contenida en <i>R</i>

Tabla 39 Requerimiento funcional N° 5 para el subsistema computacional tipo A.

Identificador	REQ-A06
Nombre	Desempeño de árbol de decisión
Descripción	El subsistema deberá medir el desempeño de un árbol de decisión en el diagnóstico asistido de Deterioro Cognitivo Leve mediante la técnica <i>10-fold cross-validation</i>
Entradas	- <i>A</i>: árbol de decisión entrenado para asistir al diagnóstico de Deterioro Cognitivo Leve - <i>X</i>: arreglo cargado con todas las características - <i>Y</i>: arreglo cargado con todas las etiquetas - <i>Z</i>: variable que designa las posibles clases de predicción
Salidas	- <i>Acc</i>: exactitud del árbol de decisión - <i>Pr</i>: precisión del árbol de decisión - <i>Se</i>: sensibilidad del árbol de decisión - <i>Sp</i>: especificidad del árbol de decisión - <i>K</i>: coeficiente Kappa del árbol de decisión

Tabla 40 Requerimiento funcional N° 6 para el subsistema computacional tipo A.

4.2.3 Especificación de los elementos del subsistema computacional tipo A

4.2.3.1 Arquitectura del subsistema computacional tipo A

Dado que los requerimientos del subsistema computacional tipo A no son excesivamente extensos o complejos, se decide optar por una arquitectura tradicional en capas. Los elementos que componen dicho patrón de arquitectura se organizan en segmentos horizontales, los cuales llevan a cabo una función específica dentro del sistema (ej. la lógica de presentación o la lógica de negocio). Aunque el patrón no especifica el número o tipos de capas que deben existir, la mayoría de estas arquitecturas constan de cuatro capas: presentación, negocio, persistencia y base de datos (Richards, 2015).

En algunos casos la capa de negocio y la capa de persistencia se combinan en una sola, particularmente cuando la lógica de la persistencia (ej. SQL o HSQL) está embebida dentro de los componentes de la capa de negocio. Por ende, las aplicaciones más pequeñas pueden tener dos o tres capas, mientras que las aplicaciones empresariales más grandes y complejas pueden tener cinco o más capas (Richards, 2015).

De acuerdo con lo anterior, se definen solamente dos capas para el subsistema computacional tipo A: capa de presentación y capa de negocio (ver ilustración 233). En la primera se alojan las interfaces gráficas de usuario y los controladores del subsistema. En la segunda, se alojan las entidades que modelan las técnicas de aprendizaje de máquina. En el caso actual, la capa de persistencia se encuentra inmersa en la capa de negocio.

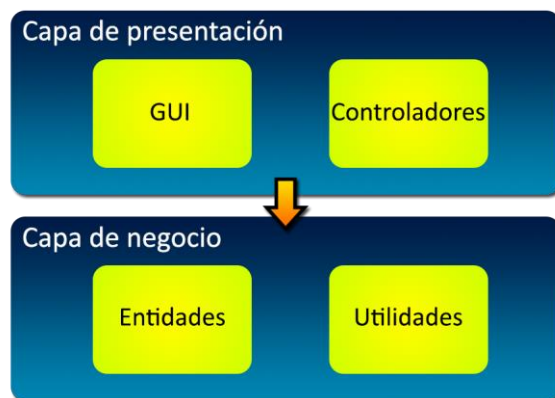


Ilustración 233 Arquitectura de dos capas: una capa para las interfaces gráficas de usuario y el controlador de flujo, y otra capa para las entidades y utilidades del subsistema.

4.2.3.2 Diagrama de clases

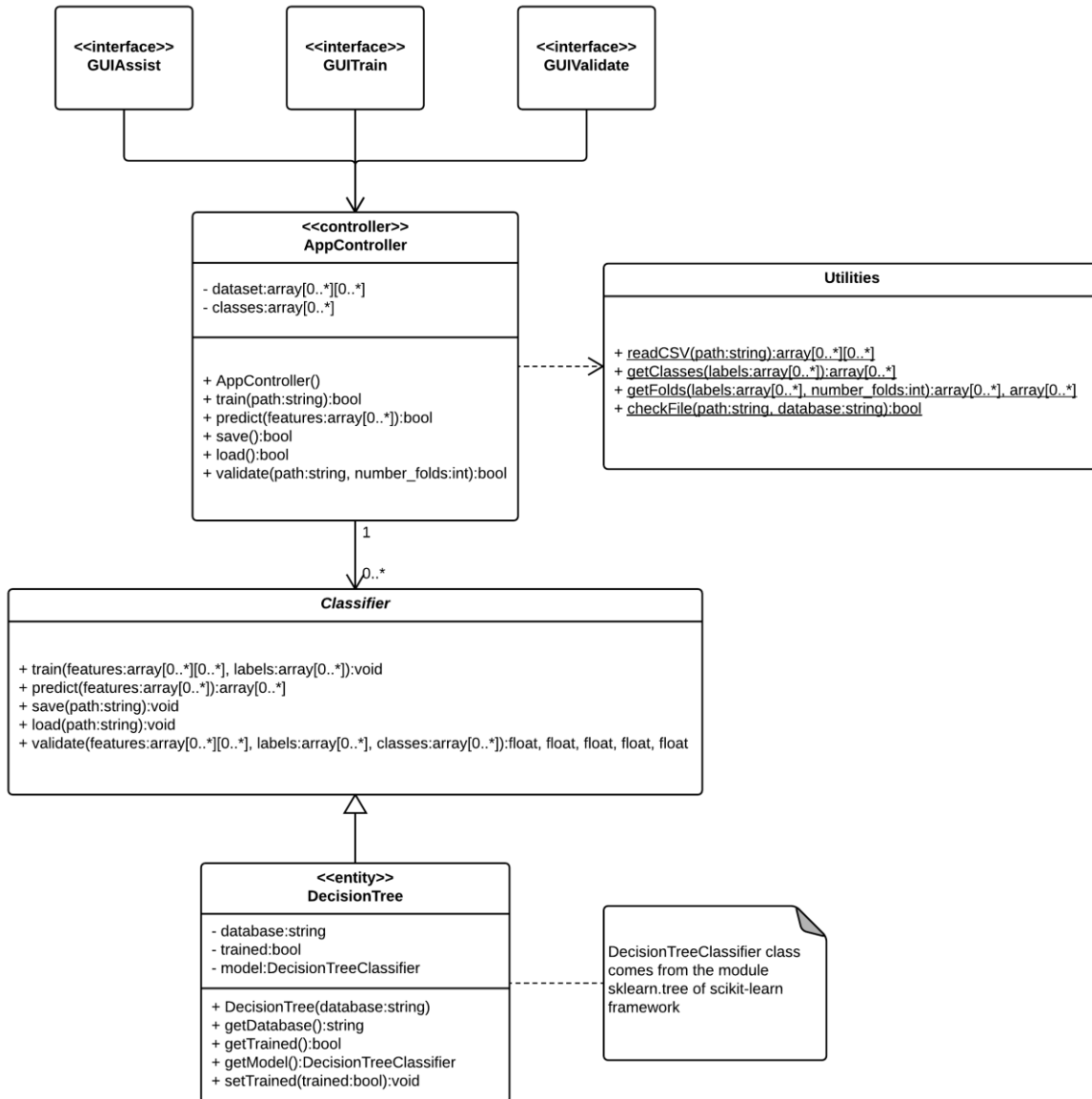


Ilustración 234 Diagrama de clases para el subsistema computacional tipo A. Este cuenta con tres interfaces gráficas de usuario: una para la asistencia al diagnóstico de Deterioro Cognitivo Leve, una para realizar el entrenamiento del árbol de decisión, y una para realizar la validación del árbol de decisión. Adicionalmente, cuenta con un controlador de flujo para conectar las peticiones del usuario con las respectivas acciones de las entidades y una serie de utilidades que facilitan el tratamiento de los datos. Finalmente, se definen algunas operaciones comunes a todos los clasificadores por medio de una clase abstracta para facilitar los llamados a sus métodos.

4.2.3.3 Decisiones de diseño para la implementación del subsistema computacional tipo A

- **Lenguaje de programación:** para la implementación del subsistema computacional tipo A se decide emplear el lenguaje de programación Python dado que este posee gran cantidad de librerías y herramientas que facilitan el trabajo con todos los temas relacionados a aprendizaje de máquina (Atabay, 2016). Además, Python viene listo para desplegar en ambientes de producción, a diferencia de otros lenguajes ampliamente usados en ML como R o Matlab (Owen, 2015).
- **Frameworks y herramientas:** para la implementación del árbol de decisión se decide emplear el marco de trabajo *Scikit-learn*. Este *framework* fue tenido en cuenta en el estudio comparativo de las técnicas de ML y es ampliamente reconocido por diversas organizaciones por su modernidad, versatilidad y consistencia (Scikit-learn, 2014). Adicionalmente, se decide emplear la herramienta PyQt para implementar la interfaz gráfica de usuario. Esta provee una conexión con el *framework* Qt que es reconocido a nivel mundial por brindar agilidad, interactividad y portabilidad en el desarrollo de aplicaciones y GUIs (Qt Company, 2016).
- **Persistencia:** para guardar y cargar los árboles de decisión se decide emplear las funciones provistas por *Scikit-learn* que admiten persistir modelos implementados con dicho *framework*. Estas funciones están basadas en el módulo *pickle* de Python el cual permite serializar y de-serializar objetos en archivos de extensión *.pkl* (Scikit-learn, 2014).

4.2.4 Prototipo base del subsistema computacional tipo A

El prototipo base del subsistema computacional tipo A comprende tres requerimientos cruciales: REQ-A01 —importación de variables—, REQ-A02 —entrenamiento de árbol de decisión—, y REQ-A03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión—. Estos requerimientos definen la funcionalidad base esperada por el usuario final por lo cual son implementados y probados (ver anexo 10) antes que los demás.

A continuación, se presentan algunas capturas de pantalla relativas al prototipo base del subsistema computacional tipo A.

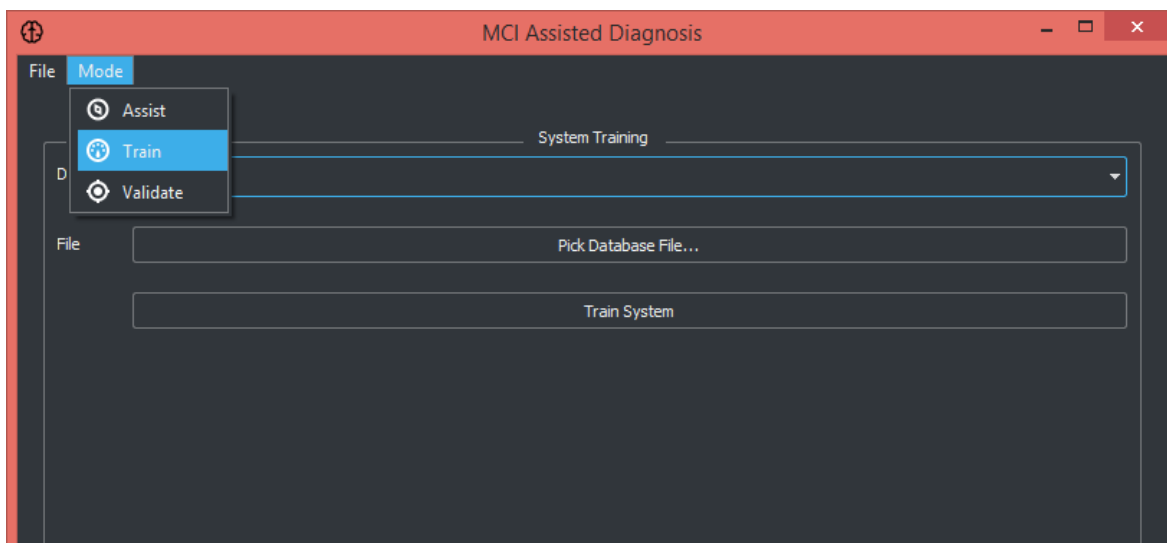


Ilustración 235 Menú principal del prototipo base del subsistema computacional tipo A. Este permite establecer el modo de la aplicación entre asistencia, entrenamiento y validación. Cada modo posee su propia interfaz gráfica de usuario la cual permite desarrollar una tarea específica dentro del subsistema computacional.

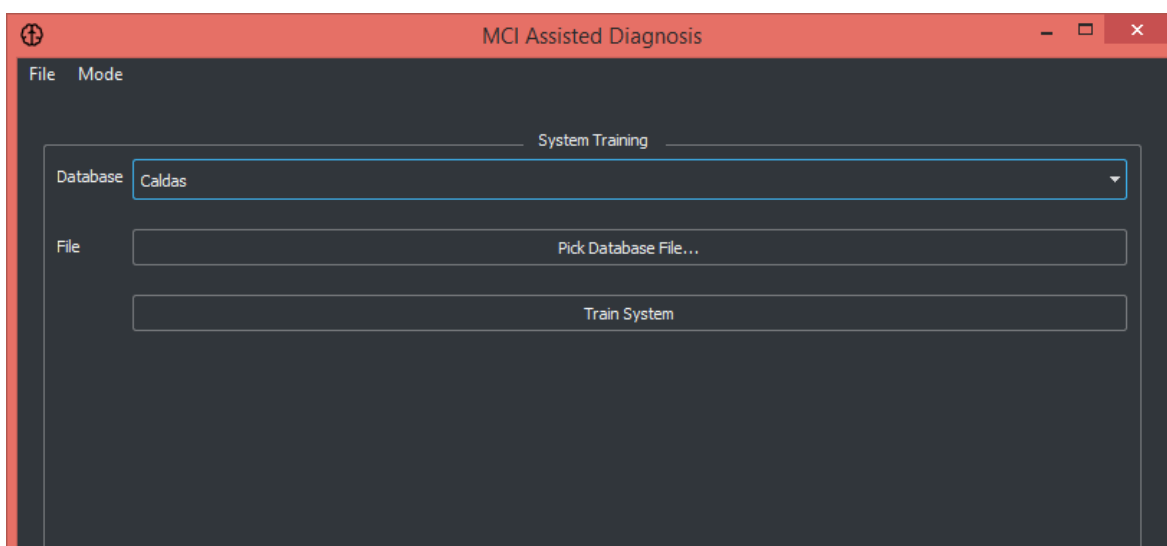
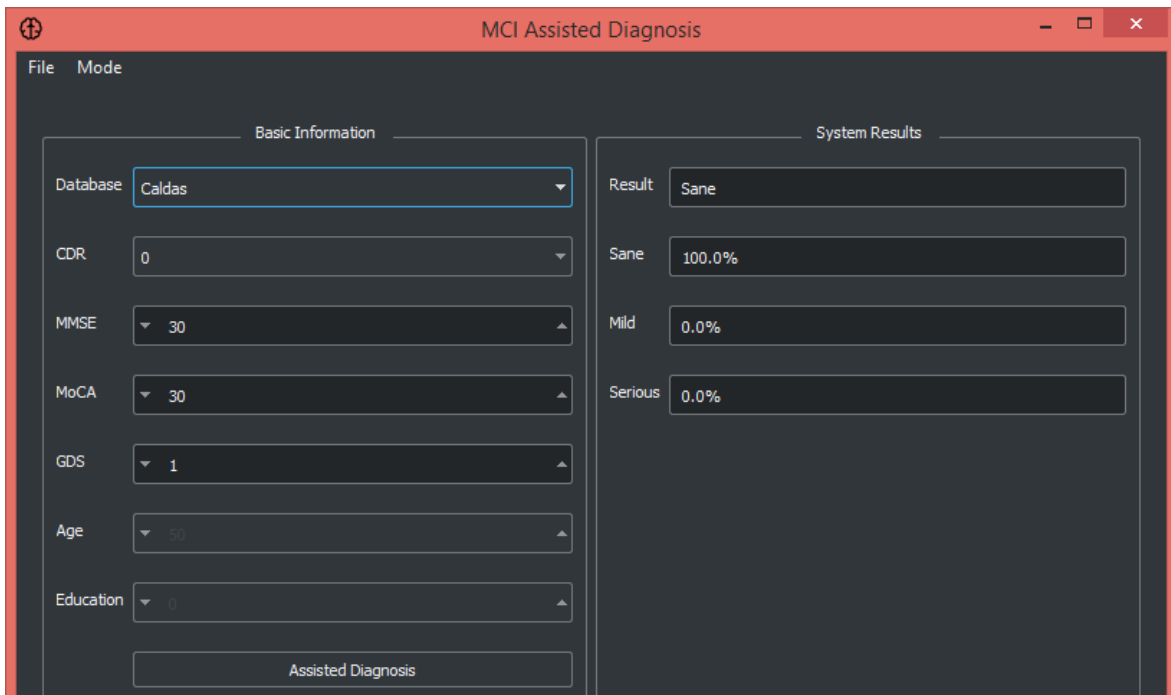


Ilustración 236 Modo de entrenamiento del prototipo base del subsistema computacional tipo A. Este permite entrenar un árbol de decisión para el diagnóstico asistido de Deterioro Cognitivo Leve seleccionando el tipo de base de datos, el archivo de base de datos y la opción de entrenamiento del subsistema computacional.



The screenshot shows the 'MCI Assisted Diagnosis' application window. The 'Basic Information' panel on the left contains the following fields:

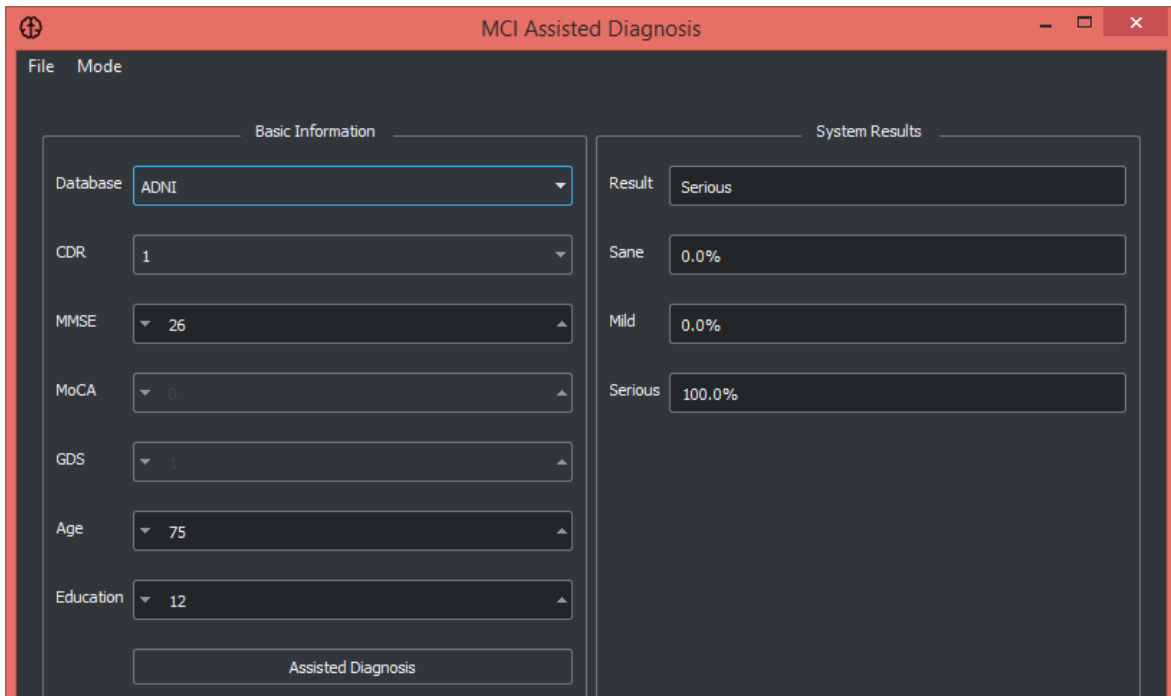
- Database: Caldas
- CDR: 0
- MMSE: 30
- MoCA: 30
- GDS: 1
- Age: 50
- Education: 0

The 'System Results' panel on the right shows the following results:

- Result: Sane
- Sane: 100.0%
- Mild: 0.0%
- Serious: 0.0%

An 'Assisted Diagnosis' button is located at the bottom of the 'Basic Information' panel.

Ilustración 237 Modo de asistencia del prototipo base del subsistema computacional tipo A. Este permite conocer la probabilidad de que una persona padezca de Deterioro Cognitivo Leve o grave, o esté sana seleccionando el tipo de base de datos, las características de la persona y la opción de diagnóstico asistido.



The screenshot shows the 'MCI Assisted Diagnosis' application window with different input values. The 'Basic Information' panel on the left contains the following fields:

- Database: ADNI
- CDR: 1
- MMSE: 26
- MoCA: 0
- GDS: 1
- Age: 75
- Education: 12

The 'System Results' panel on the right shows the following results:

- Result: Serious
- Sane: 0.0%
- Mild: 0.0%
- Serious: 100.0%

An 'Assisted Diagnosis' button is located at the bottom of the 'Basic Information' panel.

Ilustración 238 Modo de asistencia del prototipo base del subsistema computacional tipo A. Nótese el cambio en las características de la persona de acuerdo con el tipo de base de datos seleccionado.

4.2.5 Iteraciones sobre el prototipo base del subsistema computacional tipo A

4.2.5.1 Iteración N° 1

Con el objeto de expandir la funcionalidad del subsistema computacional tipo A se realiza una primera iteración que comprende dos requerimientos secundarios: REQ-A04 —persistencia de árbol de decisión—, y REQ-A05 —cargue de árbol de decisión—. Estos requerimientos no son cruciales en la asistencia al diagnóstico de Deterioro Cognitivo Leve, pero permiten agilizar el uso del sistema y evitan tener que volver a entrenar las técnicas de aprendizaje de máquina cada vez que se desee utilizar la herramienta. Las pruebas ejecutadas para los nuevos requerimientos se encuentran descritas en el ítem «10.5.2 Iteración N° 1» de la sección «10.5 Casos de prueba»

A continuación, se presentan algunas capturas de pantalla relativas a la implementación de los requerimientos adicionados en la primera iteración sobre el prototipo base del subsistema computacional tipo A.

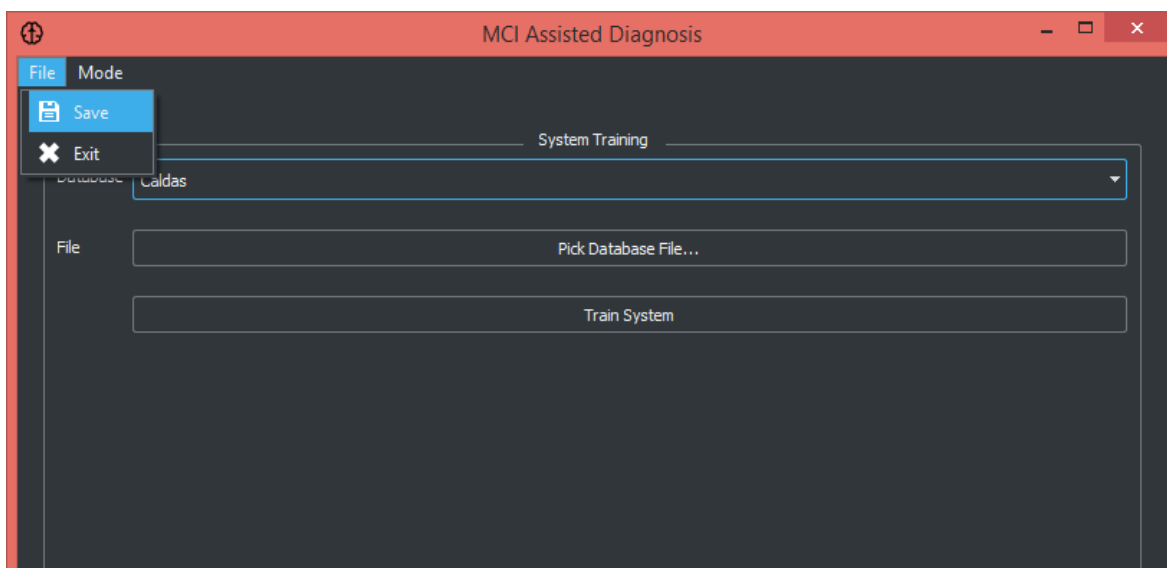


Ilustración 239 Menú principal del subsistema computacional tipo A. Este posee una opción adicional para guardar los árboles de decisión entrenados para el diagnóstico asistido de Deterioro Cognitivo Leve. Los árboles persistidos se cargan automáticamente en los usos posteriores del subsistema computacional.

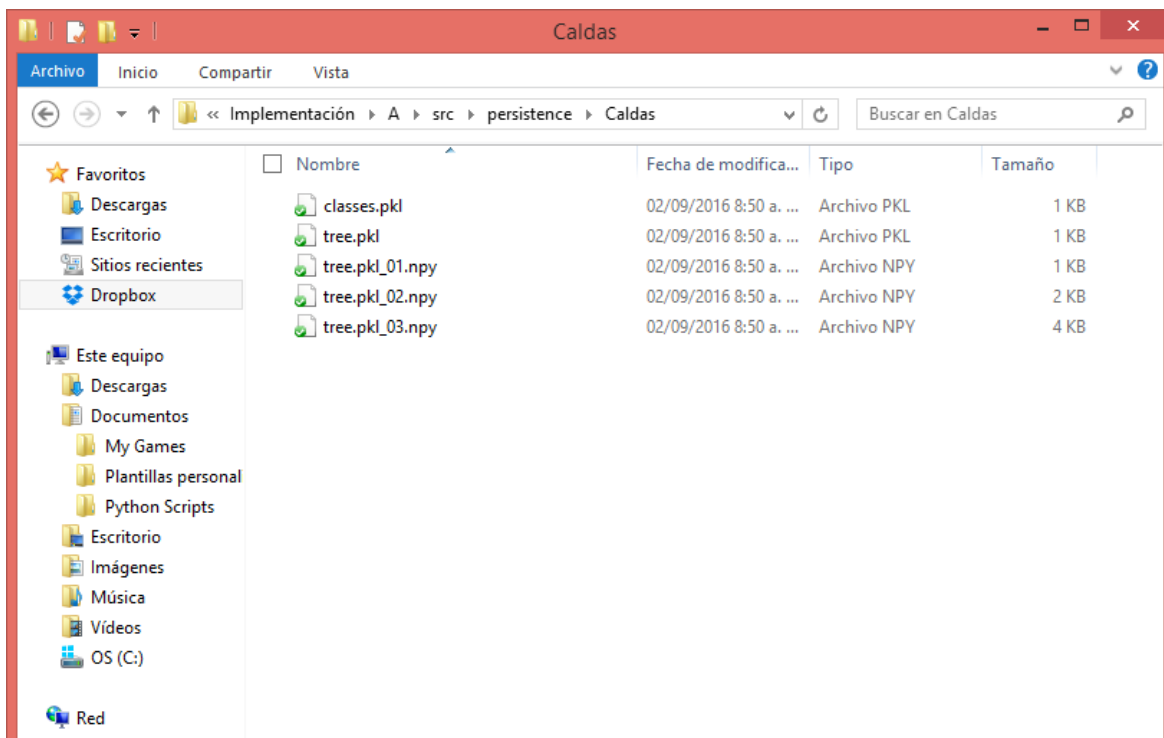
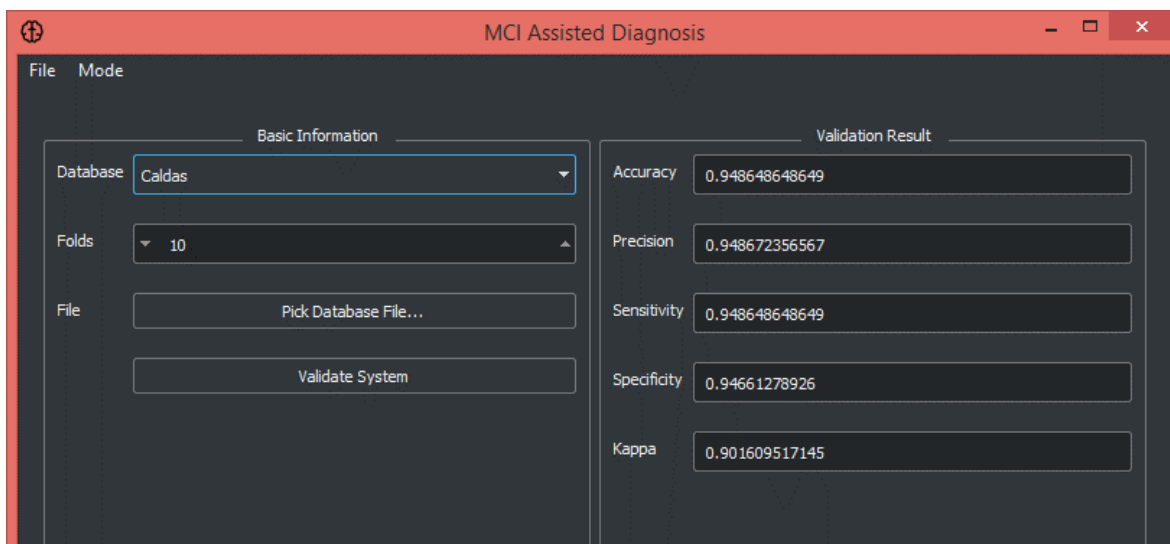


Ilustración 240 Archivos físicos resultantes de la persistencia de los árboles de decisión en el subsistema computacional tipo A.

4.2.5.2 Iteración N° 2

Por último, se realiza una segunda iteración que comprende un único requerimiento: REQ-A06 —Desempeño de árbol de decisión—. Este requerimiento no es crucial en la asistencia al diagnóstico de Deterioro Cognitivo Leve, pero permite observar cuán adecuada es la técnica de aprendizaje de máquina para dicha tarea. Las pruebas ejecutadas para el nuevo requerimiento se encuentran descritas en el ítem «10.5.3 Iteración N° 2» de la sección «10.5 Casos de prueba». Lo anterior faculta liberar el subsistema computacional tipo A para el ambiente de producción.

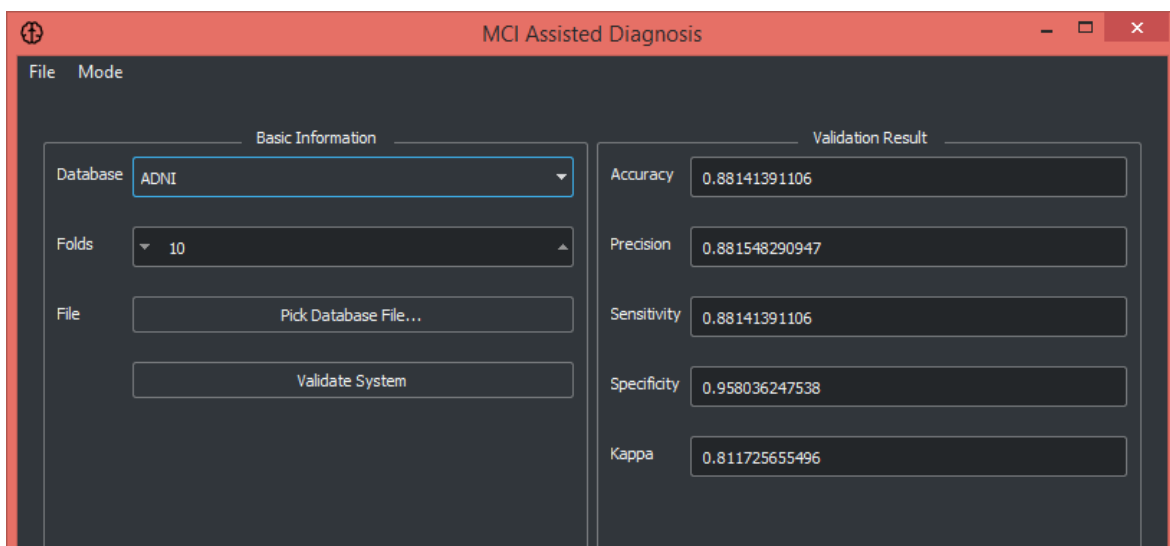
A continuación, se presenta una captura de pantalla relativa a la implementación del requerimiento adicionado en la segunda iteración sobre el prototipo base del subsistema computacional tipo A.



The screenshot shows the 'MCI Assisted Diagnosis' application window. The 'Basic Information' section on the left has 'Database' set to 'Caldas', 'Folds' set to '10', and the 'Validate System' button is highlighted. The 'Validation Result' section on the right displays the following metrics:

Metric	Value
Accuracy	0.948648648649
Precision	0.948672356567
Sensitivity	0.948648648649
Specificity	0.94661278926
Kappa	0.901609517145

Ilustración 241 Modo de validación del prototipo base del subsistema computacional tipo A. Este permite conocer las métricas de desempeño de un árbol de decisión entrenado para el diagnóstico asistido de Deterioro Cognitivo Leve seleccionando el tipo de base de datos, la cantidad de pliegues de validación, el archivo de base de datos y la opción de validación del subsistema computacional. La columna ubicada al lado derecho muestra las métricas de desempeño resultantes de la validación cruzada con 10 iteraciones del árbol de decisión entrenado con la base de datos de Caldas.



The screenshot shows the 'MCI Assisted Diagnosis' application window. The 'Basic Information' section on the left has 'Database' set to 'ADNI', 'Folds' set to '10', and the 'Validate System' button is highlighted. The 'Validation Result' section on the right displays the following metrics:

Metric	Value
Accuracy	0.88141391106
Precision	0.881548290947
Sensitivity	0.88141391106
Specificity	0.958036247538
Kappa	0.811725655496

Ilustración 242 Modo de validación del prototipo base del subsistema computacional tipo A. La columna ubicada al lado derecho muestra las métricas de desempeño resultantes de la validación cruzada con 10 iteraciones del árbol de decisión entrenado con la base de datos de la ADNI.

En las dos ilustraciones anteriores se puede observar que las métricas resultantes de la validación cruzada con 10 pliegues del árbol de decisión, son inferiores a las obtenidas en el estudio comparativo de las técnicas de ML (ver tabla 12). Esto se debe a la reducción de variables realizada por medio de bosques aleatorios sobre las bases de datos de la *ADNI* y *Caldas* (ver anexos 8 y 9). A continuación, se muestra la cuantificación de dicha reducción.

Métricas	ADNI		Caldas	
	Original	Actual	Original	Actual
Exactitud	0,9019	0,8818 ± 0,0018	0,9676	0,9526 ± 0,0127
Precisión	0,9036	0,8818 ± 0,0020	0,9676	0,9526 ± 0,0128
Sensibilidad	0,9019	0,8818 ± 0,0019	0,9676	0,9526 ± 0,0127
Especificidad	0,9630	0,9479 ± 0,0010	0,9642	0,9509 ± 0,0137
Kappa	0,8438	0,8123 ± 0,0030	0,9378	0,9091 ± 0,0237

Tabla 41 Comparación de las métricas resultantes de la validación cruzada con 10 iteraciones del árbol de decisión, antes y después de haber reducido el conjunto de variables por medio de bosques aleatorios.

4.2.6 Código fuente del subsistema computacional tipo A

El código fuente del subsistema computacional tipo A puede ser consultado en línea en el repositorio <https://github.com/florez87/a-cadmci>

4.3 Fase 3: construcción del subsistema computacional tipo B

El subsistema computacional tipo B se encuentra basado en la técnica subsimbólica de aprendizaje de máquina denominada red neuronal artificial —*ReLU Softmax Adamax*—.

4.3.1 Extracción de los requerimientos del subsistema computacional tipo B

Los requerimientos del subsistema computacional tipo B se extrajeron en las siguientes historias de usuario:

- Como médico quiero saber si una persona sufre de Deterioro Cognitivo Leve para poder iniciar el tratamiento adecuado lo antes posible.

- Como administrador quiero entrenar una red neuronal artificial para brindar una herramienta de asistencia al diagnóstico de Deterioro Cognitivo Leve.
- Como administrador quiero guardar y cargar una red neuronal artificial entrenada, para evitar la pérdida de información al momento de cerrar la herramienta.
- Como administrador quiero observar el desempeño de una red neuronal artificial entrenada, para determinar su efectividad en la asistencia al diagnóstico de Deterioro Cognitivo Leve.

4.3.2 Análisis de los requerimientos del subsistema computacional tipo B

Identificador	REQ-B01
Nombre	Importación de variables
Descripción	El subsistema deberá leer e importar un conjunto de datos desde un archivo CSV. Cada columna del archivo (separada por el carácter «,») representa una variable diferente. El archivo debe cumplir con la plantilla descrita en el anexo 8 o la plantilla descrita en el anexo 9
Entradas	- <i>R</i> : ruta física del archivo CSV a importar
Salidas	- <i>M</i> : arreglo cargado con la totalidad de los datos del archivo CSV

Tabla 42 Requerimiento funcional N° 1 para el subsistema computacional tipo B.

Identificador	REQ-B02
Nombre	Entrenamiento de red neuronal artificial
Descripción	El subsistema deberá entrenar una red neuronal artificial para asistir al diagnóstico de Deterioro Cognitivo Leve empleando un conjunto de características y etiquetas
Entradas	- <i>X</i> : arreglo cargado con todas las características - <i>Y</i> : arreglo cargado con todas las etiquetas - <i>Z</i> : variable que designa las posibles clases de predicción
Salidas	- <i>A</i> : red neuronal artificial entrenada para asistir al diagnóstico de

	Deterioro Cognitivo Leve
--	--------------------------

Tabla 43 Requerimiento funcional N° 2 para el subsistema computacional tipo B.

Identificador	REQ-B03
Nombre	Asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial
Descripción	El subsistema deberá indicar la probabilidad de que una persona padezca de Deterioro Cognitivo Leve por medio de una red neuronal artificial
Entradas	- A: red neuronal artificial entrenada para asistir al diagnóstico de Deterioro Cognitivo Leve - X: arreglo cargado con todas las características de la persona - Z: variable que designa las posibles clases de predicción
Salidas	- Pr: arreglo cargado con las probabilidades de que la persona pertenezca a cada una de las clases designadas por Z

Tabla 44 Requerimiento funcional N° 3 para el subsistema computacional tipo B.

Identificador	REQ-B04
Nombre	Persistencia de red neuronal artificial
Descripción	El sistema deberá persistir una red neuronal artificial
Entradas	- A: red neuronal artificial a persistir - R: ruta física donde se guarda la red neuronal artificial
Salidas	- W: archivo que contiene el resultado de la persistencia de A

Tabla 45 Requerimiento funcional N° 4 para el subsistema computacional tipo B.

Identificador	REQ-B05
Nombre	Cargue de red neuronal artificial
Descripción	El sistema deberá cargar una red neuronal artificial
Entradas	- R: ruta física donde se encuentra guardada la red neuronal artificial
Salidas	- A: red neuronal artificial cargada a partir de la información contenida en R

Tabla 46 Requerimiento funcional N° 5 para el subsistema computacional tipo B.

Identificador	REQ-B06
Nombre	Desempeño de red neuronal artificial
Descripción	El sistema deberá medir el desempeño de una red neuronal artificial en el diagnóstico asistido de Deterioro Cognitivo Leve mediante la técnica <i>10-fold cross-validation</i>
Entradas	- A: red neuronal artificial entrenada para asistir al diagnóstico de Deterioro Cognitivo Leve - X: arreglo cargado con todas las características - Y: arreglo cargado con todas las etiquetas - Z: variable que designa las posibles clases de predicción
Salidas	- Acc: exactitud de la red neuronal artificial - Pr: precisión de la red neuronal artificial - Se: sensibilidad de la red neuronal artificial - Sp: especificidad de la red neuronal artificial - K: coeficiente Kappa de la red neuronal artificial

Tabla 47 Requerimiento funcional N° 6 para el subsistema computacional tipo B.

4.3.3 Especificación de los elementos del subsistema computacional tipo B

4.3.3.1 Arquitectura del subsistema computacional tipo B

Dado que los requerimientos del subsistema computacional tipo B no son excesivamente extensos o complejos, se decide optar por una arquitectura equivalente a la descrita en la sección «4.2.3.1 Arquitectura del subsistema computacional tipo A».

4.3.3.2 Diagrama de clases

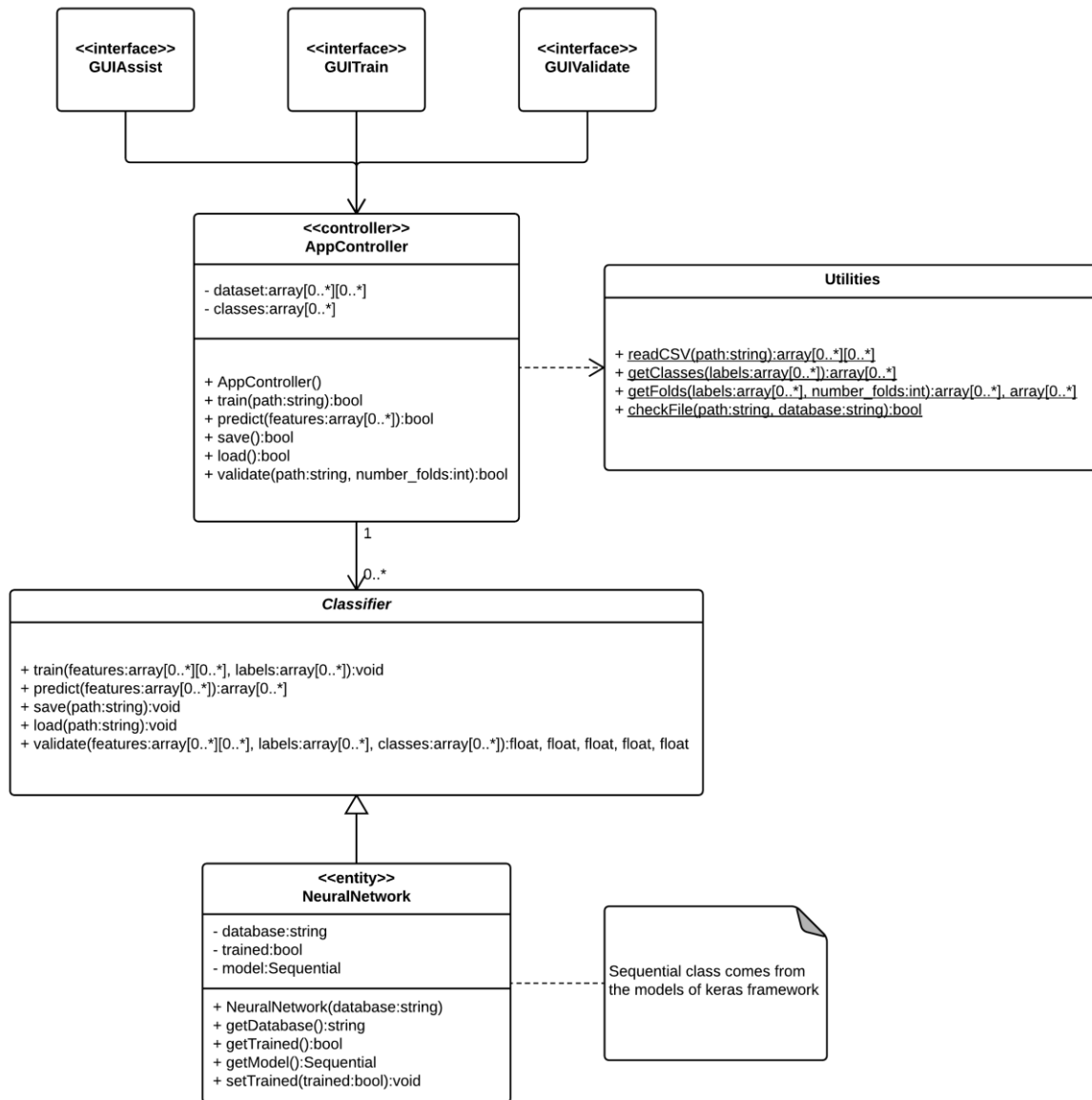


Ilustración 243 Diagrama de clases para el subsistema computacional tipo B. Este cuenta con tres interfaces gráficas de usuario: una para la asistencia al diagnóstico de Deterioro Cognitivo Leve, una para realizar el entrenamiento de la red neuronal, y una para realizar la validación de la red neuronal. Adicionalmente, cuenta con un controlador de flujo para conectar las peticiones del usuario con las respectivas acciones de las entidades y una serie de utilidades que facilitan el tratamiento de los datos. Finalmente, se definen algunas operaciones comunes a todos los clasificadores por medio de una clase abstracta para facilitar los llamados a sus métodos.

4.3.3.3 Decisiones de diseño para la implementación del subsistema computacional tipo B

- Lenguaje de programación: para la implementación del subsistema computacional tipo B se decide emplear el lenguaje de programación Python a fin de obtener uniformidad con los demás subsistemas computacionales.
- *Frameworks* y herramientas: para la implementación de la red neuronal artificial — *ReLU Softmax Adamax*— se decide emplear el marco de trabajo *Keras*. Este *framework* fue tenido en cuenta en el estudio comparativo de las técnicas de ML y es reconocido por su simpleza y modularidad (Keras, 2016). Adicionalmente, se decide emplear la herramienta PyQt para implementar la interfaz gráfica de usuario a fin de obtener uniformidad con los demás subsistemas computacionales.
- Persistencia: para guardar y cargar las redes neuronales artificiales se decide emplear las funciones provistas por *Keras* que admiten persistir modelos implementados en dicho *framework*. Estas funciones sirven para almacenar y cargar modelos de *Keras* a partir de archivos HDF5 (*Hierarchical Data Format*) (Keras, 2016).

4.3.4 Prototipo base del subsistema computacional tipo B

El prototipo base del subsistema computacional tipo B comprende 3 requerimientos cruciales: REQ-B01 —Importación de variables—, REQ-B02 —Entrenamiento de red neuronal artificial—, y REQ-B03 —Asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—. Estos requerimientos definen la funcionalidad base esperada por el usuario final por lo cual son implementados y probados (ver anexo 11) antes que los demás.

A continuación se presentan algunas capturas de pantalla relativas al prototipo base del subsistema computacional tipo B.

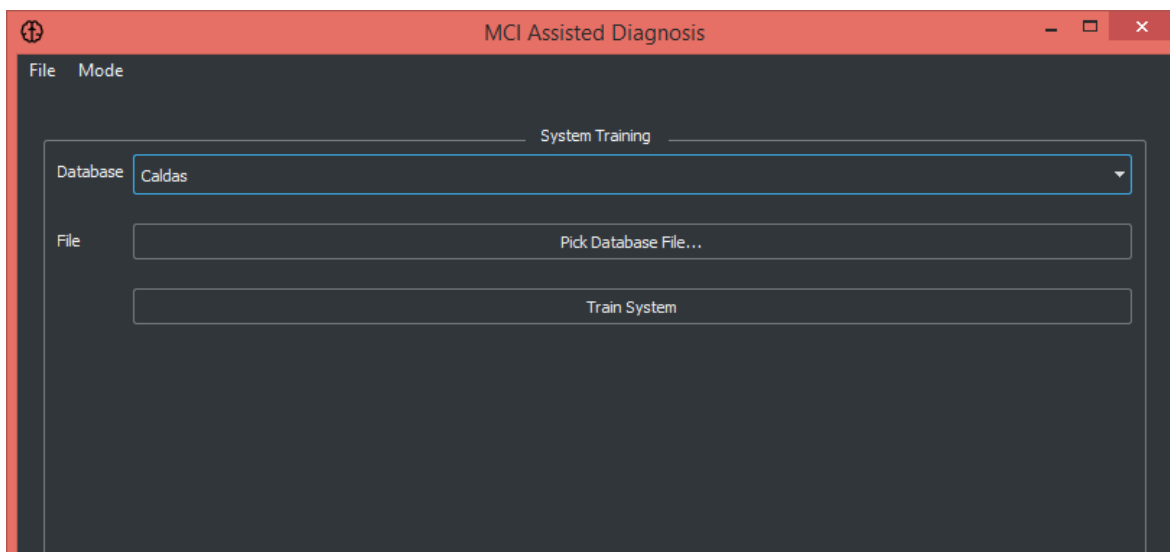


Ilustración 244 Modo de entrenamiento del prototipo base del subsistema computacional tipo B. Este permite entrenar una red neuronal para el diagnóstico asistido de Deterioro Cognitivo Leve y posee una interfaz gráfica de usuario equivalente a la del subsistema computacional tipo A.

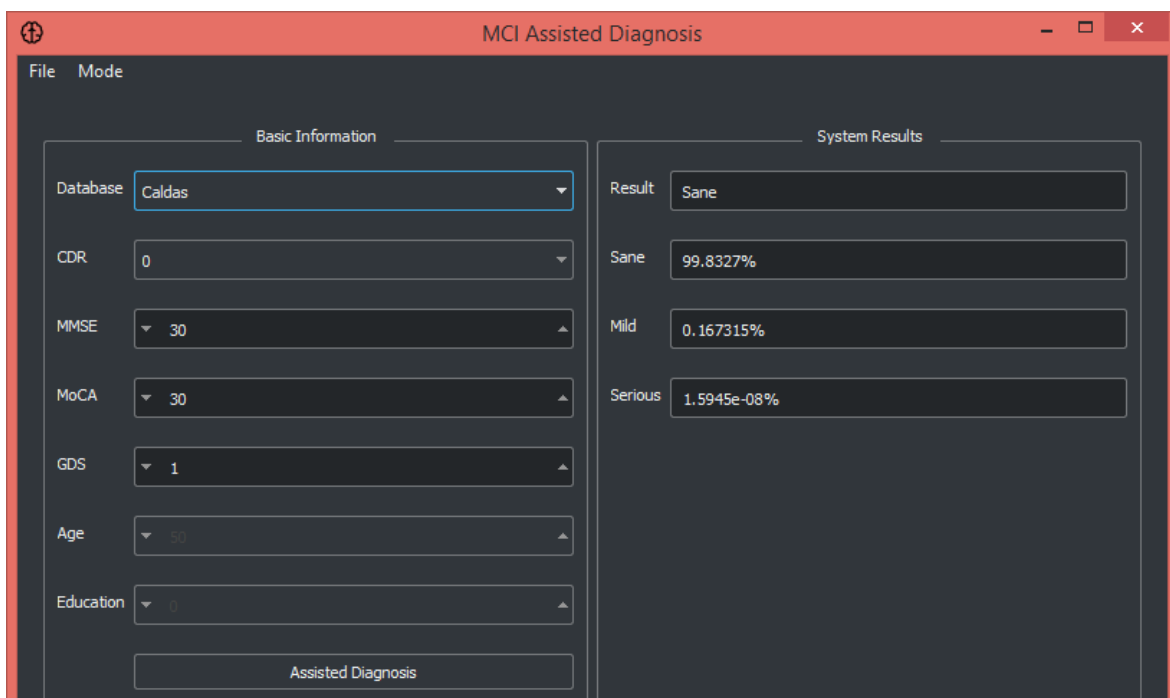
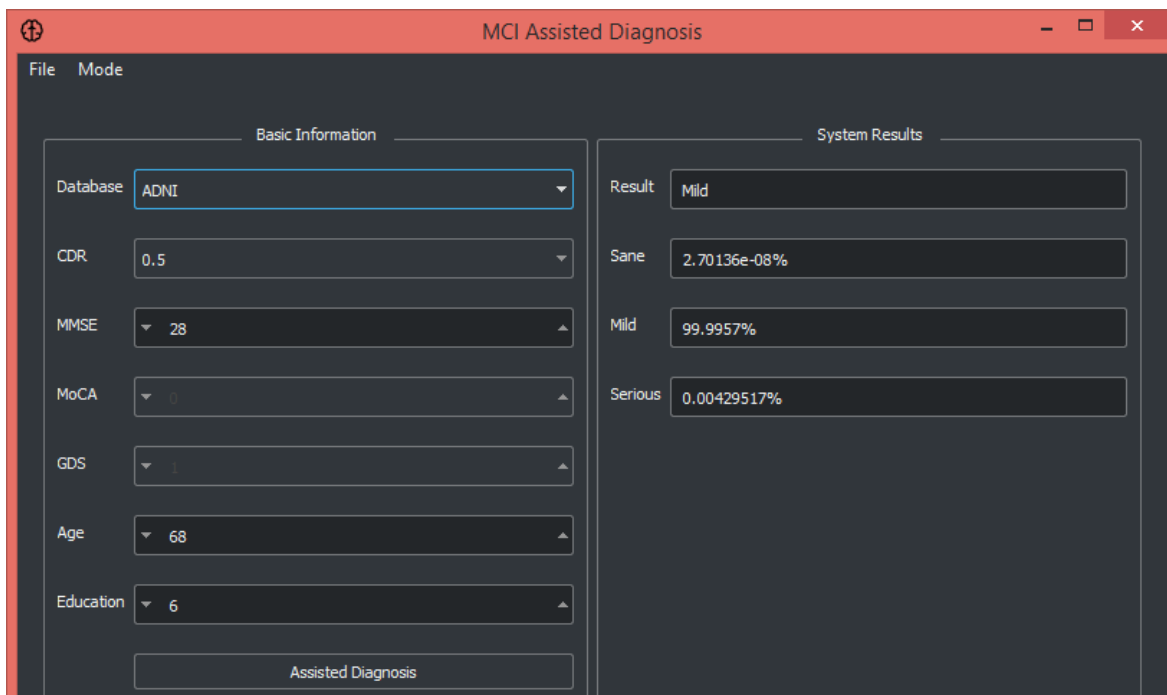


Ilustración 245 Modo de asistencia del prototipo base del subsistema computacional tipo B. Este permite conocer la probabilidad de que una persona padezca de Deterioro Cognitivo Leve o grave, o esté sana seleccionando el tipo de base de datos, las características de la persona y la opción de diagnóstico asistido. Igualmente, posee una interfaz gráfica de usuario equivalente a la del subsistema computacional tipo A.



Basic Information		System Results	
Database	ADNI	Result	Mild
CDR	0.5	Sane	2.70136e-08%
MMSE	28	Mild	99.9957%
MoCA	0	Serious	0.00429517%
GDS	1		
Age	68		
Education	6		
Assisted Diagnosis			

Ilustración 246 Modo de asistencia del prototipo base del subsistema computacional tipo B. Nótese el cambio en las características de la persona de acuerdo con el tipo de base de datos seleccionado. Igualmente, posee una interfaz gráfica de usuario equivalente a la del subsistema computacional tipo A.

4.3.5 Iteraciones sobre el prototipo base del subsistema computacional tipo B

4.3.5.1 Iteración N° 1

Con el objeto de expandir la funcionalidad del subsistema computacional tipo B se realiza una primera iteración que comprende dos requerimientos secundarios: REQ-B04 —persistencia de red neuronal artificial—, y REQ-B05 —cargue de red neuronal artificial—. Estos requerimientos no son cruciales en la asistencia al diagnóstico de Deterioro Cognitivo Leve, pero permiten agilizar el uso del sistema y evitan tener que volver a entrenar las técnicas de aprendizaje de máquina cada vez que se desee utilizar la herramienta. Las pruebas ejecutadas para los nuevos requerimientos se encuentran descritas en el ítem «11.5.2 Iteración N° 1» de la sección «11.5 Casos de prueba»

A continuación, se presentan algunas capturas de pantalla relativas a la implementación de los requerimientos adicionales en la primera iteración sobre el prototipo base del subsistema computacional tipo B.

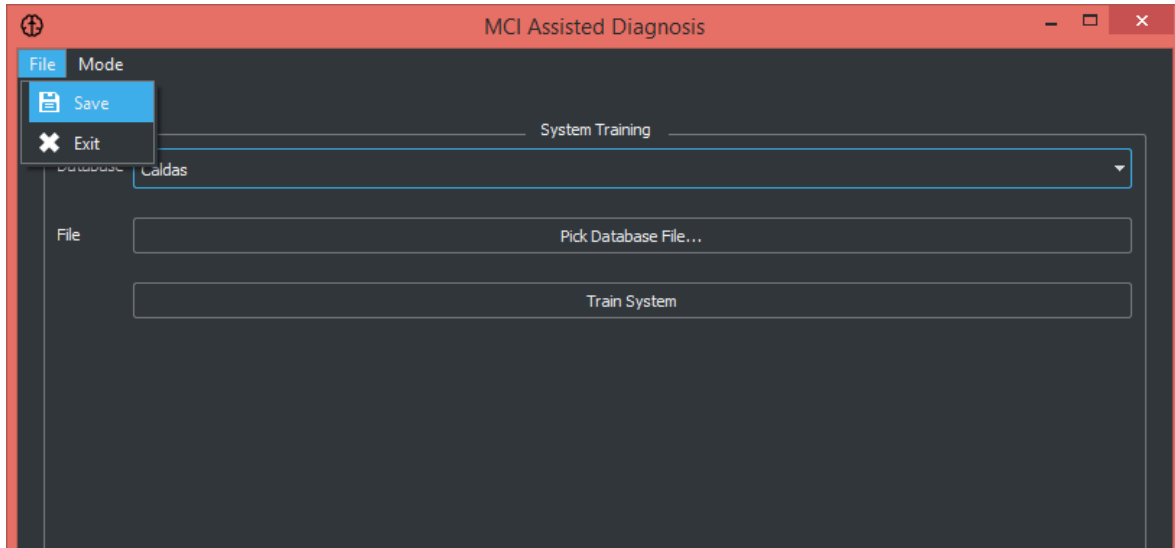


Ilustración 247 Menú principal del subsistema computacional tipo B. Este posee una opción adicional para guardar las redes neuronales entrenadas para el diagnóstico asistido de Deterioro Cognitivo Leve. Las redes neuronales persistidas se cargan automáticamente en los usos posteriores del subsistema computacional.

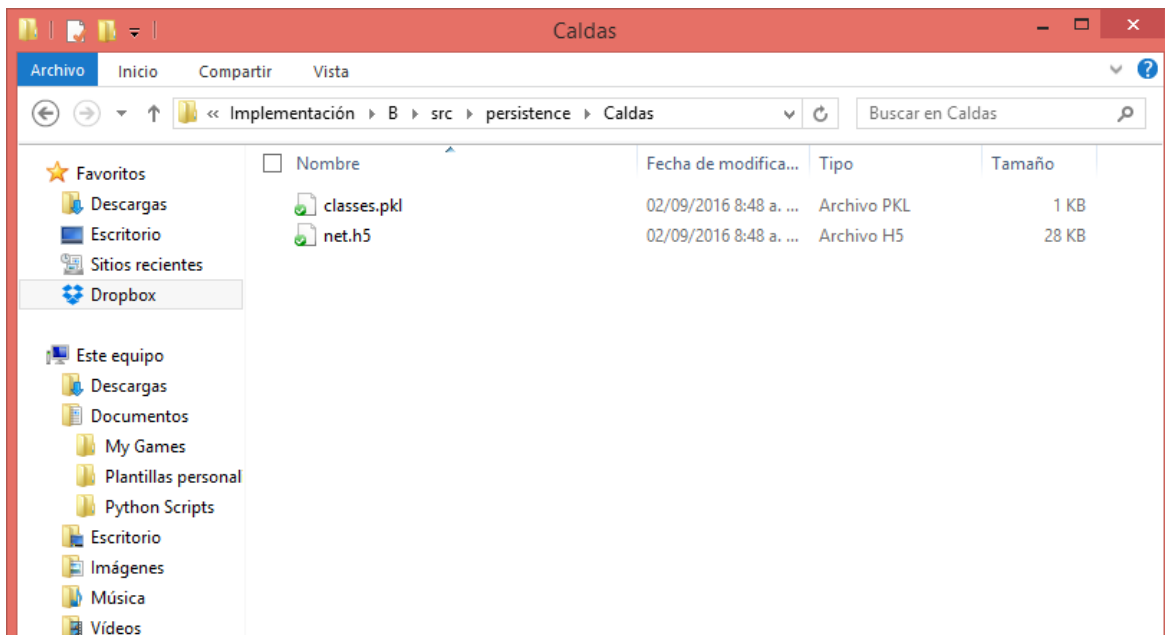


Ilustración 248 Archivos físicos resultantes de la persistencia de las redes neuronales en el subsistema computacional tipo B.

4.3.5.2 Iteración N° 2

Por último, se realiza una segunda iteración que comprende un único requerimiento: REQ-B06 —Desempeño de red neuronal artificial—. Este requerimiento no es crucial en la asistencia al diagnóstico de Deterioro Cognitivo Leve, pero permite observar cuán adecuada es la técnica de aprendizaje de máquina para dicha tarea. Las pruebas ejecutadas para el nuevo requerimiento se encuentran descritas en el ítem «11.5.3 Iteración N° 2» de la sección «11.5 Casos de prueba». Lo anterior faculta liberar el subsistema computacional tipo B para el ambiente de producción.

A continuación, se presenta una captura de pantalla relativa a la implementación del requerimiento adicionado en la segunda iteración sobre el prototipo base del subsistema computacional tipo B.

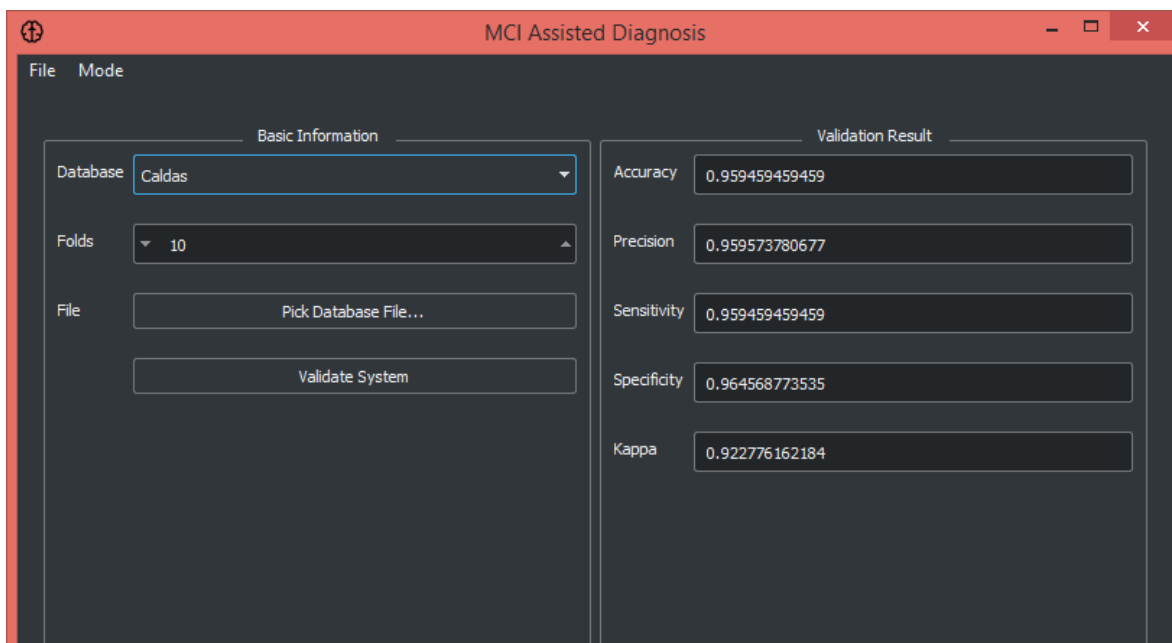


Ilustración 249 Modo de validación del prototipo base del subsistema computacional tipo B. Este permite conocer las métricas de desempeño de una red neuronal artificial entrenada para el diagnóstico asistido de Deterioro Cognitivo Leve seleccionando el tipo de base de datos, la cantidad de pliegues de validación, el archivo de base de datos y la opción de validación del subsistema computacional. Igualmente, posee una interfaz gráfica de usuario equivalente a la del subsistema computacional tipo A. La columna ubicada al lado derecho muestra las métricas de desempeño resultantes de la validación cruzada con 10 iteraciones de la red neuronal artificial entrenada con la base de datos de Caldas.

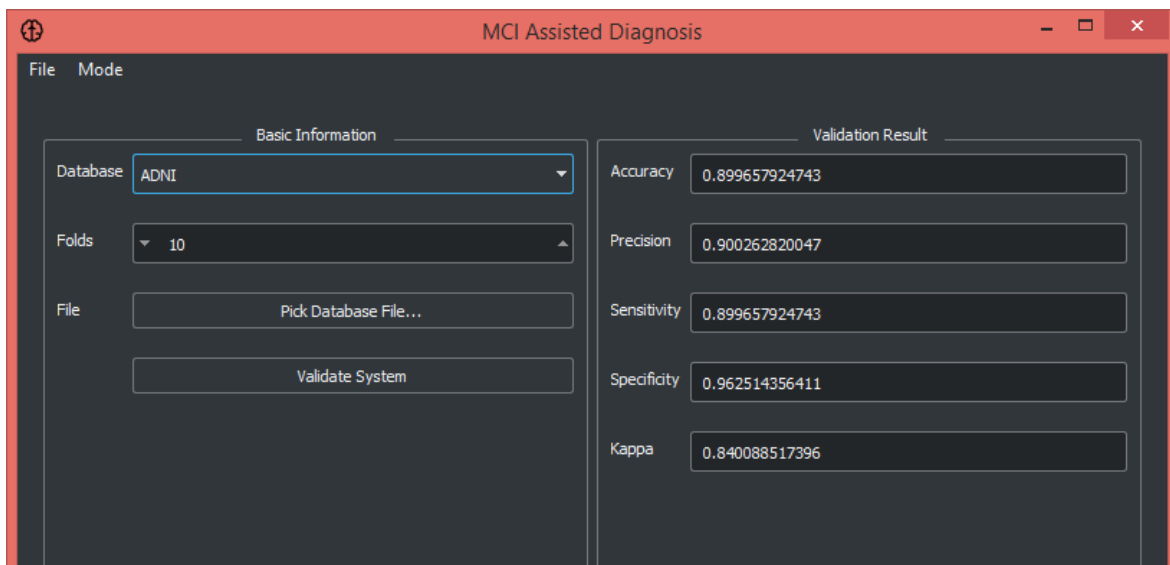


Ilustración 250 Modo de validación del prototipo base del subsistema computacional tipo B. La columna ubicada al lado derecho muestra las métricas de desempeño resultantes de la validación cruzada con 10 iteraciones de la red neuronal artificial entrenada con la base de datos de la ADNI.

En las dos ilustraciones anteriores se puede observar que las métricas resultantes de la validación cruzada con 10 pliegues de la red neuronal artificial, son inferiores a las obtenidas en el estudio comparativo de las técnicas de ML (ver tabla 28). Esto se debe a la reducción de variables realizada por medio de bosques aleatorios sobre las bases de datos de la *ADNI* y *Caldas* (ver anexos 8 y 9). A continuación, se muestra la cuantificación de dicha reducción.

	ADNI		Caldas	
Métricas	Original	Actual	Original	Actual
Exactitud	0,9042	0,8958 ± 0,0055	0,9649	0,9664 ± 0,0056
Precisión	0,9071	0,8957 ± 0,0058	0,9656	0,9556 ± 0,0367
Sensibilidad	0,9042	0,8958 ± 0,0055	0,9649	0,9664 ± 0,0056
Especificidad	0,9633	0,9618 ± 0,0016	0,9811	0,9692 ± 0,0035
Kappa	0,8471	0,8341 ± 0,0085	0,9335	0,9355 ± 0,0111

Tabla 48 Comparación de las métricas resultantes de la validación cruzada con 10 iteraciones del árbol de decisión, antes y después de haber reducido el conjunto de variables por medio de bosques aleatorios.

4.3.6 Código fuente del subsistema computacional tipo B

El código fuente del subsistema computacional tipo B puede ser consultado en línea en el repositorio <https://github.com/florez87/b-cadmci>

4.4 Fase 4: integración de los subsistemas computacionales de tipo A y B

4.4.1 Esquema de hibridación elegido

Para conjugar el funcionamiento de los subsistemas computacionales de tipo A y B se decide emplear el método denominado *Bagging*. Este es semejante al método detallado en la sección «4.1.11 *Decision Trees (Bagging)*», no obstante, cuando se emplean clasificadores heterogéneos como árboles de decisión y redes neuronales, es necesario adecuar el algoritmo de producción del ensamble (ver ilustración 251) para admitir dicha composición (Hsu, 2012).

Input: D es el conjunto de datos de entrenamiento, y N es el número de clasificadores base

Output: E es el ensamble resultante, esto es, un conjunto de clasificadores base

```

 $E \leftarrow \emptyset;$ 
for  $i \leftarrow 1; i \leq N$  do
     $S_i \leftarrow \text{Bootstrap}(D);$ 
    if  $i$  es par then
        especificar  $A_i$  como un árbol de decisión;
    else
        especificar  $A_i$  como una red neuronal artificial;
    end
     $C_i \leftarrow \text{ConstruirClasificador}(S_i, A_i);$ 
     $E \leftarrow E \cup C_i;$ 
end
return  $E;$ 

```

Ilustración 251 Algoritmo de producción del ensamble híbrido compuesto de árboles de decisión y redes neuronales artificiales. Los mejores resultados de clasificación se obtienen al emplear cinco clasificadores de cada tipo (Hsu, 2012).

Respecto al mecanismo de predicción del ensamble se propone un algoritmo (ver ilustración 252) fundamentado en el método de *Bagging*, que afronta las estimaciones tanto desde una perspectiva de clasificación como de regresión. Esto permite mitigar los inconvenientes presentes en dicho método cuando ocurren empates en el proceso de votación de la etiqueta de clase final.

Para lo anterior, se requiere que cada clasificador brinde su predicción individual como un par formado por la etiqueta de clase estimada y la probabilidad de pertenecer a cada una de las clases disponibles. Por ejemplo: siendo $\{Sane, Mild, Serious\}$ las posibles clases de predicción, la respuesta de un clasificador podría ser $(Sane, [0,7; 0,2; 0,1])$ o $(Mild, [0,1; 0,9; 0,0])$ o $(Serious, [0,1; 0,4; 0,5])$.

En caso tal que los promedios de las probabilidades de pertenecer a dos o más clases sean iguales, se efectúa el mecanismo de desempate sugerido en el método tradicional de *Bagging* que implica seleccionar la menor etiqueta de clase (Breiman, 1994).

Input: x son las variables demográficas y neuropsicológicas de una persona, Z es el conjunto de posibles etiquetas de clases, E es el conjunto de clasificadores base entrenados, y N es el número de clasificadores base

Output: E^* es la etiqueta de clase final que indica el estado mental del paciente

```

etiquetas  $\leftarrow [0 \dots N]$ ; // arreglo o lista
probabilidades  $\leftarrow [0 \dots N][\text{Tamaño}(Z)]$ ; // arreglo o lista
for  $i \leftarrow 1; i \leq N$  do
     $(etiquetas_i, probabilidades_i) \leftarrow \text{Predecir}(E_i, x)$ ;
end
 $E^* \leftarrow \arg \max_{z \in Z} \sum_{i: etiquetas_i = z} 1$ ; // etiqueta(s) más predicha(s)
if  $\text{Tamaño}(E^*) > 1$  then // existe empate en la votación
     $eje \leftarrow 0$ ;
     $promedios \leftarrow \text{Promedio}(probabilidades, eje)$ ;
     $E^* \leftarrow \arg \max_{z \in Z} promedios$ ; // etiqueta con mayor promedio
end
return  $E^*$ ;
```

Ilustración 252 Mecanismo de predicción para el ensamble híbrido. Este permite someter a votación las predicciones individuales de los clasificadores base como es sugerido en el Bagging. En caso de existir un empate, la decisión se realiza por el promedio de las probabilidades de pertenecer a cada una de las clases.

4.4.2 Integración de los subsistemas computacionales

Para la integración de los subsistemas computacionales a través del método *Bagging* que conlleva a la obtención del sistema computacional híbrido se definen cinco nuevos requerimientos funcionales:

Identificador	REQ-C01
Nombre	Entrenamiento de ensamble basado en <i>Bagging</i>
Descripción	El subsistema deberá entrenar un ensamble compuesto por cinco árboles de decisión y cinco redes neuronales artificiales para asistir al diagnóstico de Deterioro Cognitivo Leve empleando un conjunto de características y etiquetas
Entradas	- X: arreglo cargado con todas las características - Y: arreglo cargado con todas las etiquetas - Z: variable que designa las posibles clases de predicción
Salidas	- A: ensamble basado en <i>Bagging</i> entrenado para asistir al diagnóstico de Deterioro Cognitivo Leve

Tabla 49 Requerimiento funcional N° 1 para el sistema computacional híbrido.

Identificador	REQ-C02
Nombre	Asistencia al diagnóstico de Deterioro Cognitivo Leve mediante ensamble basado en <i>Bagging</i>
Descripción	El subsistema deberá indicar la probabilidad de que una persona padezca de Deterioro Cognitivo Leve por medio de un ensamble compuesto por cinco árboles de decisión y cinco redes neuronales artificiales
Entradas	- A: ensamble basado en <i>Bagging</i> entrenado para asistir al diagnóstico de Deterioro Cognitivo Leve - X: arreglo cargado con todas las características de la persona - Z: variable que designa las posibles clases de predicción
Salidas	- Pr: arreglo cargado con las probabilidades de que la persona pertenezca a cada una de las clases designadas por Z

Tabla 50 Requerimiento funcional N° 2 para el sistema computacional híbrido.

Identificador	REQ-C03
Nombre	Persistencia de ensamble basado en <i>Bagging</i>
Descripción	El sistema deberá persistir un ensamble compuesto por cinco árboles de decisión y cinco redes neuronales artificiales
Entradas	- <i>A</i>: ensamble basado en <i>Bagging</i> a persistir - <i>R</i>: ruta física donde se guarda el ensamble basado en <i>Bagging</i>
Salidas	- <i>W</i>: archivo que contiene el resultado de la persistencia de <i>A</i>

Tabla 51 Requerimiento funcional N° 3 para el sistema computacional híbrido.

Identificador	REQ-C04
Nombre	Cargue de ensamble basado en <i>Bagging</i>
Descripción	El sistema deberá cargar un ensamble compuesto por cinco árboles de decisión y cinco redes neuronales artificiales
Entradas	- <i>R</i>: ruta física donde se encuentra guardado el ensamble basado en <i>Bagging</i>
Salidas	- <i>A</i>: ensamble basado en <i>Bagging</i> cargado a partir de la información contenida en <i>R</i>

Tabla 52 Requerimiento funcional N° 4 para el sistema computacional híbrido.

Identificador	REQ-C05
Nombre	Desempeño de ensamble basado en <i>Bagging</i>
Descripción	El sistema deberá medir el desempeño de un ensamble compuesto por cinco árboles de decisión y cinco redes neuronales artificiales en el diagnóstico asistido de Deterioro Cognitivo Leve mediante la técnica <i>10-fold cross-validation</i>
Entradas	- <i>A</i>: ensamble basado en <i>Bagging</i> entrenado para asistir al diagnóstico de Deterioro Cognitivo Leve - <i>X</i>: arreglo cargado con todas las características - <i>Y</i>: arreglo cargado con todas las etiquetas

	- Z: variable que designa las posibles clases de predicción
Salidas	- Acc: exactitud del ensamble basado en <i>Bagging</i> - Pr: precisión del ensamble basado en <i>Bagging</i> - Se: sensibilidad del ensamble basado en <i>Bagging</i> - Sp: especificidad del ensamble basado en <i>Bagging</i> - K: coeficiente <i>kappa</i> del ensamble basado en <i>Bagging</i>

Tabla 53 Requerimiento funcional N° 5 para el sistema computacional híbrido.

Asimismo, se realiza el diagrama de clases (ver ilustración 253) que refleja la estructura deseada para la hibridación de los subsistemas computacionales de tipo A y B.

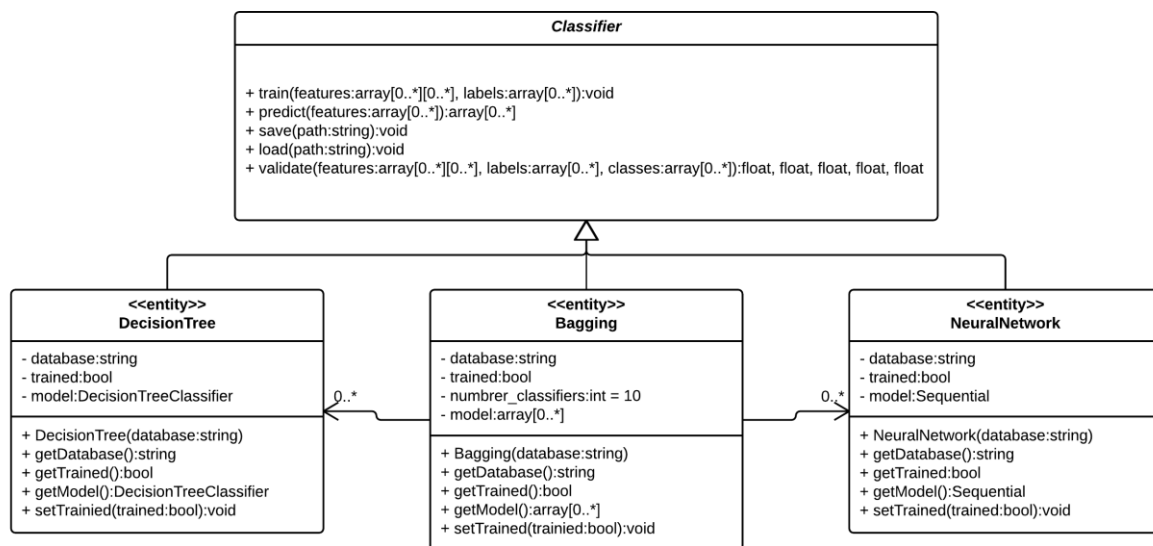


Ilustración 253 Diagrama de clases para la hibridación de los subsistemas computacionales de tipo A y B. El mayor cambio radica en la inclusión de un nuevo tipo de clasificador para el sistema denominado *Bagging*. Este define un modelo compuesto por clasificadores de tipo *DecisionTree* y *NeuralNetwork* los cuales provienen de los subsistemas computacionales de tipo A y B construidos con anterioridad. Las demás clases se mantienen iguales a las diseñadas para los subsistemas computacionales.

Finalmente, se implementa la hibridación de los subsistemas computacionales de tipo A y B, se ejecuta un nuevo plan de pruebas (ver anexo 12) para verificar el funcionamiento del sistema computacional híbrido y se libera para el ambiente de producción.

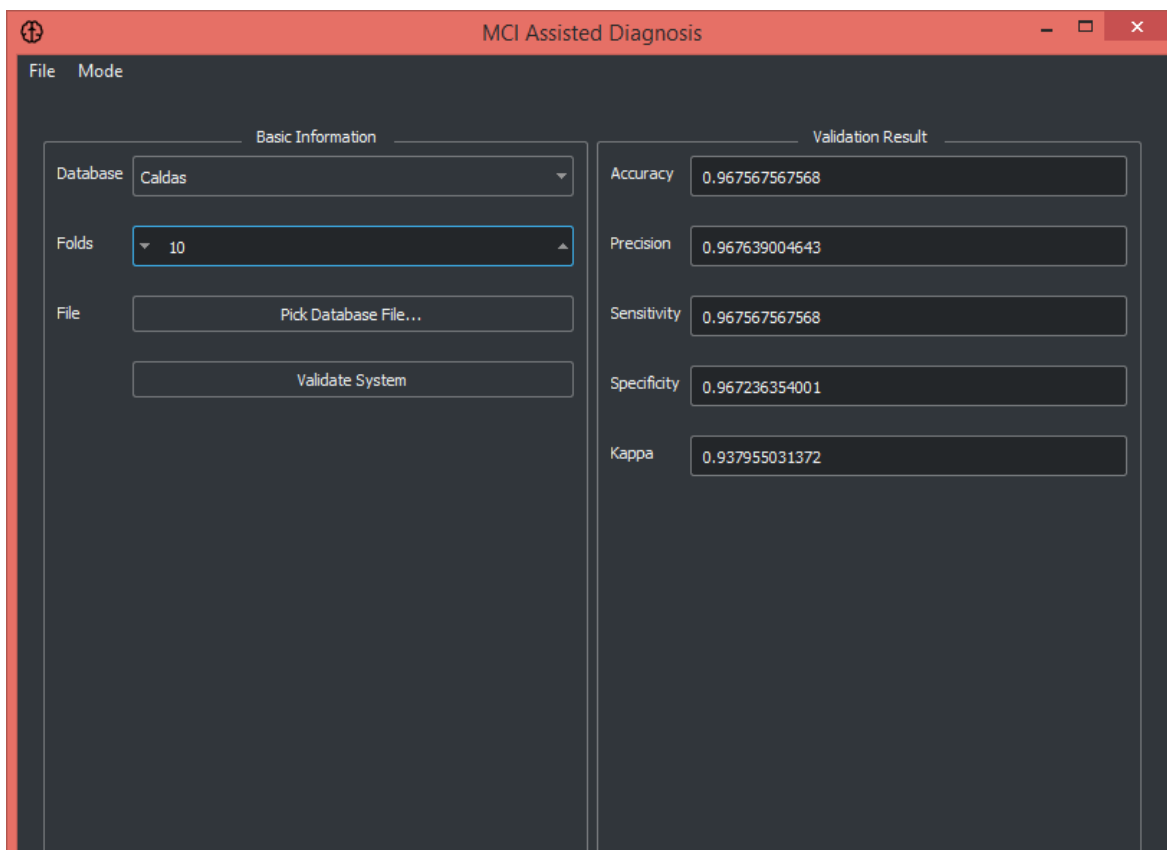
4.4.3 Código fuente del sistema computacional híbrido

El código fuente del sistema computacional híbrido puede ser consultado en línea en el repositorio <https://github.com/florez87/hybrid-cadmci>

4.5 Fase 5: validación del sistema computacional híbrido

4.5.1 Desempeño del sistema mediante validación cruzada con 10 iteraciones

A continuación, se presentan dos capturas de pantalla relativas a las métricas de desempeño obtenidas en la validación del sistema computacional híbrido mediante la técnica denominada *10-fold cross-validation*.



The screenshot displays the 'MCI Assisted Diagnosis' application window. The interface is divided into two main sections: 'Basic Information' on the left and 'Validation Result' on the right. In the 'Basic Information' section, the 'Database' is set to 'Caldas', 'Folds' is set to '10', and there are buttons for 'Pick Database File...' and 'Validate System'. The 'Validation Result' section displays the following metrics:

Metric	Value
Accuracy	0.967567567568
Precision	0.967639004643
Sensitivity	0.967567567568
Specificity	0.967236354001
Kappa	0.937955031372

Ilustración 254 Resultados de la validación cruzada con 10 iteraciones del sistema computacional híbrido para el diagnóstico asistido de Deterioro Cognitivo Leve respecto a la base de datos de Caldas.

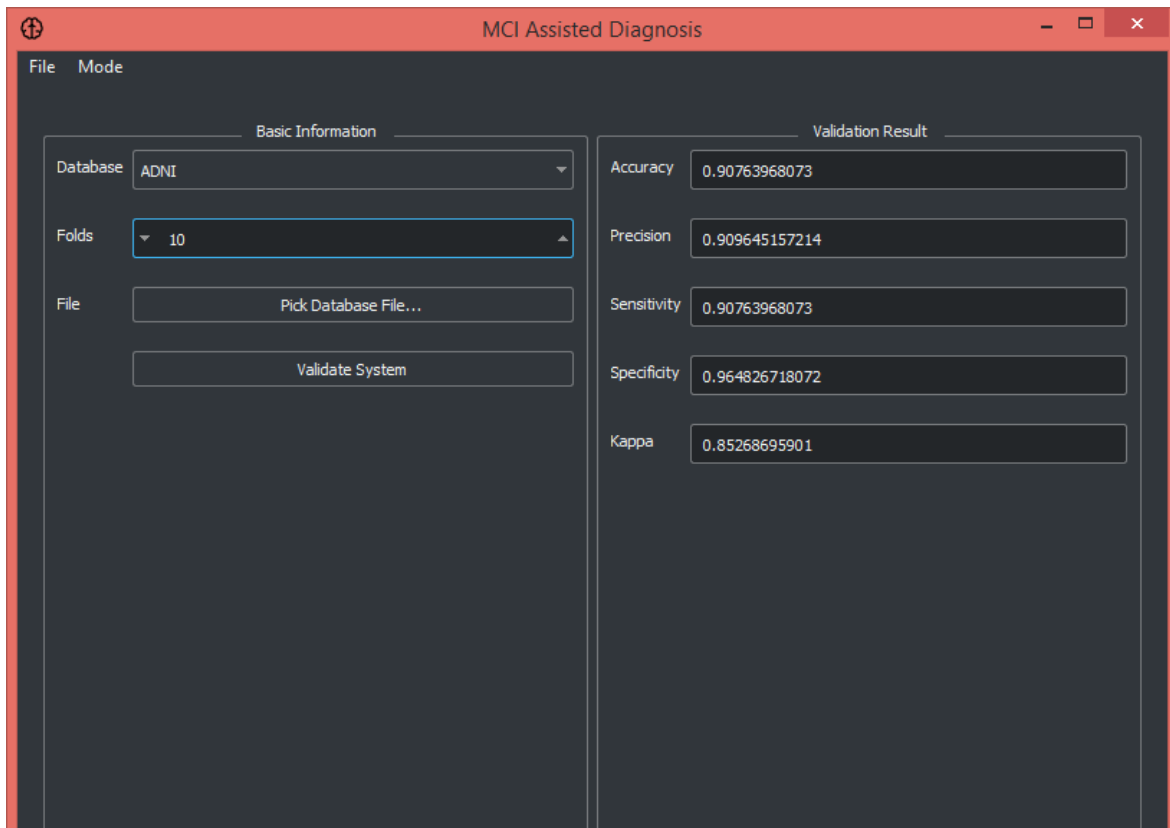


Ilustración 255 Resultados de la validación cruzada con 10 iteraciones del sistema computacional híbrido para el diagnóstico asistido de Deterioro Cognitivo Leve respecto a la base de datos de la ADNI.

En las dos ilustraciones anteriores se puede observar que las métricas resultantes de la validación cruzada con 10 pliegues del sistema híbrido basado en *Bagging*, son superiores a las obtenidas en la validación de los subsistemas computacionales de tipo A y B (ver ilustraciones 241, 242, 249 y 250). A continuación, se muestra la cuantificación de dicho incremento.

	ADNI		Caldas	
Métricas	A	Bagging	A	Bagging
Exactitud	0,8818 ± 0,0018	0,9149 ± 0,0102	0,9526 ± 0,0127	0,9770 ± 0,0046
Precisión	0,8818 ± 0,0020	0,9155 ± 0,0096	0,9526 ± 0,0128	0,9771 ± 0,0046
Sensibilidad	0,8818 ± 0,0019	0,9149 ± 0,0102	0,9526 ± 0,0127	0,9770 ± 0,0046
Especificidad	0,9479 ± 0,0010	0,9683 ± 0,0040	0,9509 ± 0,0137	0,9775 ± 0,0045
Kappa	0,8123 ± 0,0030	0,8642 ± 0,0003	0,9091 ± 0,0237	0,9554 ± 0,0087

Tabla 54 Comparación de las métricas resultantes de la validación cruzada con 10 iteraciones del sistema híbrido respecto a las obtenidas en la validación del subsistema computacional tipo A.

	ADNI		Caldas	
Métricas	B	Bagging	B	Bagging
Exactitud	0,8958 ± 0,0055	0,9149 ± 0,0102	0,9664 ± 0,0056	0,9770 ± 0,0046
Precisión	0,8957 ± 0,0058	0,9155 ± 0,0096	0,9556 ± 0,0367	0,9771 ± 0,0046
Sensibilidad	0,8958 ± 0,0055	0,9149 ± 0,0102	0,9664 ± 0,0056	0,9770 ± 0,0046
Especificidad	0,9618 ± 0,0016	0,9683 ± 0,0040	0,9692 ± 0,0035	0,9775 ± 0,0045
Kappa	0,8341 ± 0,0085	0,8642 ± 0,0003	0,9355 ± 0,0111	0,9554 ± 0,0087

Tabla 55 Comparación de las métricas resultantes de la validación cruzada con 10 iteraciones del sistema híbrido respecto a las obtenidas en la validación del subsistema computacional tipo B.

Igualmente, se puede observar que las métricas resultantes de la validación cruzada con 10 pliegues del sistema híbrido basado en *Bagging*, son superiores a las obtenidas en el estudio comparativo por las técnicas que componen el ensamble (ver tablas 12 y 28). A continuación, se muestra la cuantificación de dicho incremento.

	ADNI		Caldas	
Métricas	Árbol	Bagging	Árbol	Bagging
Exactitud	0,9019	0,9149 ± 0,0102	0,9676	0,9770 ± 0,0046
Precisión	0,9036	0,9155 ± 0,0096	0,9676	0,9771 ± 0,0046
Sensibilidad	0,9019	0,9149 ± 0,0102	0,9676	0,9770 ± 0,0046
Especificidad	0,9630	0,9683 ± 0,0040	0,9642	0,9775 ± 0,0045
Kappa	0,8438	0,8642 ± 0,0003	0,9378	0,9554 ± 0,0087

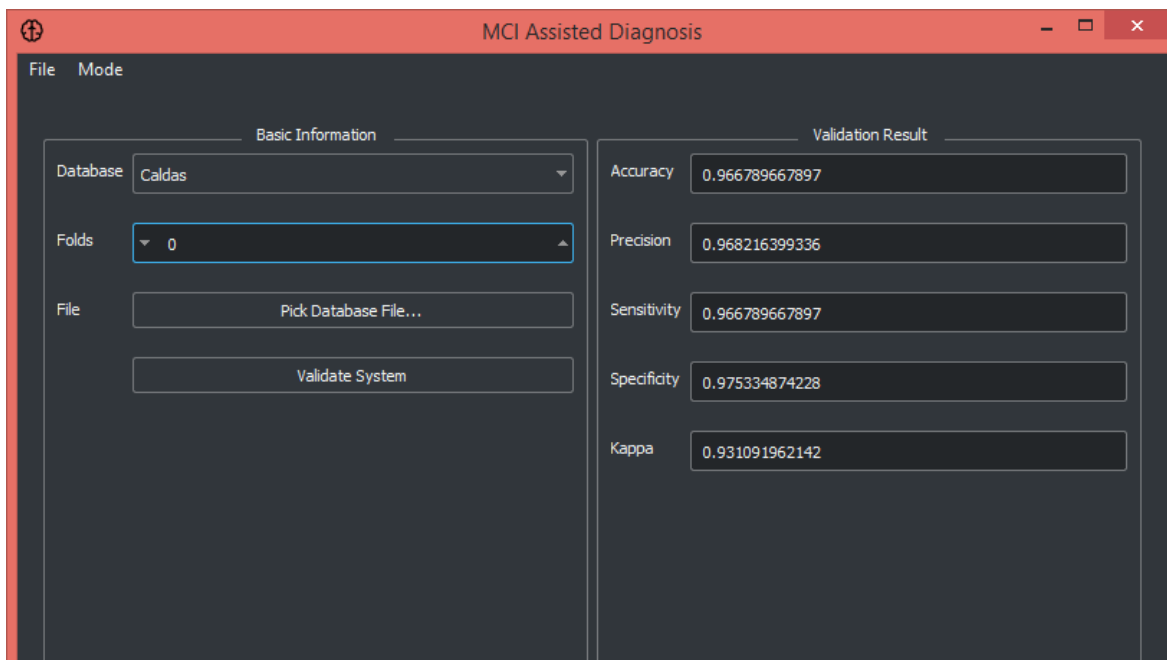
Tabla 56 Comparación de las métricas resultantes de la validación cruzada con 10 iteraciones del sistema híbrido respecto a las obtenidas por el árbol de decisión en el estudio comparativo de las técnicas de ML.

	ADNI		Caldas	
Métricas	ANN	Bagging	ANN	Bagging
Exactitud	0,9042	0,9149 ± 0,0102	0,9649	0,9770 ± 0,0046
Precisión	0,9071	0,9155 ± 0,0096	0,9656	0,9771 ± 0,0046
Sensibilidad	0,9042	0,9149 ± 0,0102	0,9649	0,9770 ± 0,0046
Especificidad	0,9633	0,9683 ± 0,0040	0,9811	0,9775 ± 0,0045
Kappa	0,8471	0,8642 ± 0,0003	0,9335	0,9554 ± 0,0087

Tabla 57 Comparación de las métricas resultantes de la validación cruzada con 10 iteraciones del sistema híbrido respecto a las obtenidas por la red neuronal artificial en el estudio comparativo de las técnicas de ML.

4.5.2 Desempeño del sistema con nuevos datos

A continuación, se presenta una captura de pantalla relativa a las métricas de desempeño obtenidas en la validación del sistema computacional híbrido con un conjunto de nuevos datos.



The screenshot shows a software window titled "MCI Assisted Diagnosis". It has a menu bar with "File" and "Mode". The interface is divided into two main sections: "Basic Information" and "Validation Result".

Basic Information:

- Database:** A dropdown menu showing "Caldas".
- Folds:** A dropdown menu showing "0".
- File:** A button labeled "Pick Database File..."
- Validate System:** A button labeled "Validate System"

Validation Result:

- Accuracy:** 0.966789667897
- Precision:** 0.968216399336
- Sensitivity:** 0.966789667897
- Specificity:** 0.975334874228
- Kappa:** 0.931091962142

Ilustración 256 Resultados de la validación del sistema computacional híbrido para el diagnóstico asistido de Deterioro Cognitivo Leve respecto a un conjunto de datos inéditos para el sistema.

Los nuevos datos provienen del proyecto denominado «**Implementación del programa para diagnóstico y control de enfermedades crónicas no transmisibles y cáncer de cérvix y mama, con el apoyo de TIC en el departamento de Caldas**» en donde diferentes especialistas continúan realizando el diagnóstico de la condición mental de las personas de Caldas. Por consiguiente, se cuenta con un total de 271 nuevos registros clínicos (180 sanos, 76 con DCL y 15 con deterioro cognitivo moderado o severo) aparte de los 370 que se encontraban inicialmente en la base de datos de *Caldas*.

Las métricas visibles en la ilustración 253 representan el desempeño del sistema computacional híbrido al estimar el diagnóstico de los 271 nuevos registros siendo entrenado solamente con los 370 registros originales de la base de datos de *Caldas*.

5. DISCUSIÓN DE LOS RESULTADOS

Con base en los resultados obtenidos, se encontró que:

- 1) Al reducir el conjunto original de variables por medio de bosques aleatorios, también se redujo el desempeño de las técnicas de aprendizaje de máquina (ver tablas 41 y 48). No obstante, dicha reducción fue poco significativa y se pudo llevar a cabo manteniendo un alto nivel de desempeño en la asistencia al diagnóstico de Deterioro Cognitivo Leve.
- 2) El actual sistema computacional híbrido para el diagnóstico asistido de Deterioro Cognitivo Leve basado en la técnica de ensamble denominada *Bagging* obtuvo un mejor desempeño respecto a las técnicas individuales empleadas en los subsistemas computacionales (ver tablas 54 y 55). El incremento en el desempeño por medio del *Bagging* contrarrestó la disminución del mismo causado por la reducción del conjunto original de variables y permitió mejorar la asistencia al diagnóstico de Deterioro Cognitivo Leve incluso respecto a las técnicas entrenadas con la totalidad de las variables (ver tablas 56 y 57).
- 3) El actual sistema computacional híbrido para el diagnóstico asistido de Deterioro Cognitivo Leve no se encuentra sobre-ajustado a las bases de datos destinadas para el entrenamiento de los clasificadores. Lo anterior se sustenta en el desempeño sobresaliente obtenido en la validación del sistema computacional híbrido con el conjunto de datos inéditos del departamento de Caldas (ver ilustración 256).
- 4) El actual sistema computacional híbrido para el diagnóstico asistido de Deterioro Cognitivo Leve obtuvo un mejor desempeño respecto a los métodos analizados en la revisión de antecedentes (ver tabla 1). Concretamente, es significativo comparar la exactitud del actual sistema híbrido con la exactitud de los clasificadores implementados por Umer bajo la técnica de validación cruzada con 10 iteraciones, ya que ambos emplean la base de datos de la *ADNI* para las etapas de entrenamiento, predicción y validación. Sin embargo, no es posible comparar las demás métricas de desempeño (precisión, sensibilidad, especificidad y *kappa*) puesto que estas fueron omitidas en los antecedentes revisados.

Los beneficios del presente estudio se enumeran de la siguiente forma:

- 1) El actual sistema híbrido permite asistir al diagnóstico de Deterioro Cognitivo Leve con un alto grado de exactitud. Además, su funcionamiento requiere únicamente algunos datos demográficos y pruebas cognitivas, en lugar de imágenes médicas o biomarcadores. Por consiguiente, es apto para ser popularizado en los centros de salud y hospitales de bajo nivel de complejidad los cuales poseen escasos equipos clínicos especializados.
- 2) El actual sistema híbrido brinda una guía a los profesionales de la salud respecto a la condición mental de los pacientes sin ser intensivo en costo y tiempo. Lo anterior permite a los médicos poder iniciar un tratamiento temprano de las enfermedades relacionadas con el deterioro cognitivo, sin tener que depender enteramente de los especialistas en el tema. Adicionalmente, el sistema accede a beneficiar personas de bajos recursos económicos dado que las pruebas cognitivas no representan un alto gasto monetario.
- 3) El actual sistema híbrido se basa únicamente en unos cuantos datos demográficos y puntajes (edad, educación, MMSE, MoCA, CDR y GDS). Cualquier otro tipo de enfermedad que actualmente no cuente con un «criterio de oro» para su diagnóstico y posea pruebas de características similares, podría ensayar la técnica de diagnóstico asistido propuesta —realizando ajustes mínimos al sistema—.
- 4) El actual sistema híbrido se liberó *online* bajo la licencia GPLv3. Esto permite que pueda ser utilizado no solo en el proyecto **«Implementación del programa para diagnóstico y control de enfermedades crónicas no transmisibles y cáncer de cérvix y mama, con el apoyo de TIC en el departamento de Caldas»** sino por cualquier organización o persona que desee hacerlo.

También es importante resaltar que el presente estudio posee algunas limitaciones:

- 1) Generalmente, las técnicas de aprendizaje de máquina poseen propiedades de alto nivel que deben ser elegidas manualmente durante la configuración de los modelos. Dichas propiedades son denominadas hiperparámetros. Por ejemplo, en las redes

neuronales artificiales se deben hacer múltiples elecciones respecto a su estructura y funcionamiento como la cantidad de capas, la cantidad de neuronas en cada capa, las funciones de activación de las neuronas artificiales, las funciones de pérdida, el algoritmo de entrenamiento, entre otras. Estas decisiones repercuten en el desempeño de las tareas de clasificación o regresión (Bergstra & Bengio, 2012).

En el presente estudio se emplearon múltiples técnicas de aprendizaje de máquina. Sin embargo, los valores de los hiperparámetros dispuestos para cada modelo se limitaron a las recomendaciones brindadas por los autores de las técnicas y/o los *frameworks* en los que se encontraban implementadas. No se realizó una búsqueda exhaustiva hacia el hallazgo de los hiperparámetros ideales para cada caso particular.

- 2) En el presente estudio se emplearon solamente dos bases de datos para explorar el beneficio de las técnicas de aprendizaje de máquina en la asistencia al diagnóstico de Deterioro Cognitivo Leve. Dado que la calificación de las pruebas cognitivas y el diagnóstico brindado por los especialistas posee cierto grado de subjetividad, es posible que los datos empleados provengan con algún tipo de sesgo.
- 3) En el presente estudio no se pudieron homogenizar las variables provenientes de las dos bases de datos en un solo conjunto —debido a sus diferencias en rango, dominio y significado—. En consecuencia, se debieron implementar técnicas de aprendizaje de máquina particulares a cada base de datos, lo cual repercute en la complejidad del sistema híbrido y su costo computacional.
- 4) Actualmente, no existen nuevos registros de pacientes en la *ADNI* por lo que no se pudo realizar la actividad de validación del sistema híbrido con datos inéditos —respecto a dicha base de datos—.

6. CONCLUSIONES

El propósito del presente estudio se enfocaba en explorar la aplicación conjunta de diferentes técnicas de aprendizaje de máquina para asistir al diagnóstico de Deterioro Cognitivo Leve en personas mayores de 60 años. Lo anterior convergió en la construcción de un sistema computacional híbrido capaz de analizar los resultados de múltiples pruebas de valoración cognitiva para apoyar el dictamen de MCI en la población objetivo.

El actual sistema híbrido basa su funcionamiento en el método de *Bagging*, el cual somete a votación y promedio —en caso de existir un empate— las predicciones individuales de múltiples clasificadores simbólicos y subsimbólicos para brindar a los profesionales de la salud una guía respecto a la condición mental de los pacientes. Dicho sistema fue validado mediante la técnica denominada *10-fold cross-validation* con dos bases de datos diferentes y se ratificó su alto rendimiento al ser sometido a una valoración de métricas de desempeño con datos inéditos.

De acuerdo con los resultados obtenidos se concluye que:

- Fue posible construir un sistema computacional basado en técnicas simbólicas y subsimbólicas de aprendizaje de máquina que permitiera el diagnóstico asistido de Deterioro Cognitivo Leve en personas mayores de 60 años. Aunque dicho sistema posee pocos parámetros de entrada, presenta un alto desempeño en la asistencia al diagnóstico de Deterioro Cognitivo Leve (para la base de datos de la *ADNI*: exactitud = $0,9149 \pm 0,0102$; sensibilidad = $0,9149 \pm 0,0102$; especificidad = $0,9683 \pm 0,0040$; para la base de datos de *Caldas*: exactitud = $0,9770 \pm 0,0046$; sensibilidad = $0,9770 \pm 0,0046$; especificidad = $0,9775 \pm 0,0045$).
- El actual sistema computacional híbrido puede ser desplegado fácilmente en hospitales y centros de salud de bajo nivel de complejidad dado que no requiere hardware clínico especializado y es liberado en línea bajo una licencia de código abierto. Los parámetros de entrada se relacionan directamente con los resultados de cuatro pruebas cognitivas cuya aplicación demanda bajos costos monetarios y es poco

intensiva en tiempo. Lo anterior posibilita el beneficio de un gran número de personas, aun cuando estas posean bajos recursos económicos.

- El subsistema computacional construido con base en la técnica simbólica de árboles de decisión es capaz de asistir al diagnóstico de deterioro cognitivo leve por si solo mediante el uso de un único clasificador. No obstante, su desempeño es menor al obtenido por el sistema computacional híbrido.
- El subsistema computacional construido con base en la técnica subsimbólica de redes neuronales es capaz de asistir al diagnóstico de deterioro cognitivo leve por si solo mediante el uso de un único clasificador. No obstante, su desempeño es menor al obtenido por el sistema computacional híbrido.
- Aunque los subsistemas computacionales pueden operar de forma independiente el verdadero potencial se alcanza por medio de su integración, la cual permite mejorar el desempeño de ambas partes al trabajar conjuntamente en un esquema híbrido.
- Tanto los subsistemas computacionales como el sistema computacional híbrido fueron validados en diferentes instancias mediante un esquema de validación cruzada. Igualmente, se observó un alto desempeño con un conjunto de datos inéditos para el sistema. Por otro lado, se verificaron los procesos de construcción e integración de los subsistemas computacionales mediante la realización de diferentes planes de prueba para cada fase.

7. TRABAJO FUTURO

- Realizar una búsqueda exhaustiva con variados hiperparámetros sobre las técnicas de aprendizaje de máquina, de modo que se puedan observar los beneficios o perjuicios causados por estos en la asistencia al diagnóstico de Deterioro Cognitivo Leve.
- Incorporar un mayor número de bases de datos relativas a los resultados de las pruebas cognitivas para compensar el sesgo que pudieran presentar los datos empleados en el presente estudio.
- Explorar la posibilidad de homogenizar las entradas dispares causadas por las diferencias en las variables de las bases de datos para reducir la complejidad del actual sistema computacional híbrido.
- Explorar diferentes esquemas de hibridación de las técnicas de aprendizaje de máquina a fin de mejorar la asistencia al diagnóstico de Deterioro Cognitivo Leve. La técnica denominada *Bagging* es solo una de incontables formas existentes para conjugar el funcionamiento de diversos clasificadores en uno solo.
- Explorar mediante las técnicas de aprendizaje de máquina no solamente los aspectos relativos al diagnóstico asistido de Deterioro Cognitivo Leve sino también a su evolución y progresión hacia enfermedades más graves como la demencia.
- Probar el actual sistema computacional híbrido con otras enfermedades que aún no posean un «criterio de oro» para su diagnóstico. Esto requeriría realizar ajustes en las entradas y salidas del sistema. No obstante, se podría mantener indemne el esquema de hibridación planteado con su respectivo funcionamiento.

8. BIBLIOGRAFÍA

- Abraham, A. (2005). *Artificial Neural Networks*. Recuperado el 06 de 08 de 2016, de Softcomputing: <http://goo.gl/ckYMnO>
- Alzheimer's Disease Research Center. (2001). *Clinical Dementia Rating (CDR) Scale*. Recuperado el 08 de 09 de 2015, de Alzheimer's Disease Research Center: <http://goo.gl/AxNW6X>
- Alzheimer's Association. (2009). *Deterioro Cognitivo Leve*. Recuperado el 03 de 08 de 2015, de Alzheimer's Association: <http://goo.gl/vQgUKb>
- Alzheimer's Association. (2013). *Cognitive Assessment Toolkit*. Recuperado el 07 de 08 de 2015, de Alzheimer's Association: <http://goo.gl/5jUCKP>
- Alzheimer's Society. (2015). *Helping you to assess cognition: A practical toolkit for clinicians*. Recuperado el 06 de 08 de 2015, de Alzheimer's Society: <http://goo.gl/vgOKDz>
- American Psychiatric Association. (2013). *Diagnostic and Statistical Manual of Mental Disorders*. Londres: American Psychiatric Publishing.
- Anderson, T. M., Sachdev, P. S., Brodaty, H., Trollor, J. N., & Andrews, G. (2007). Effects of Sociodemographic and Health Variables on Mini-Mental State Exam Scores in Older Australians. *The American Journal of Geriatric Psychiatry*, 467-476.
- Armitage, P., & Colton, T. (2005). *Encyclopedia of Biostatistics: 8-Volume Set, 2nd Edition*. Hoboken: Wiley.
- Asociación Colombiana de Gerontología y Geriatria. (2009). *La guía Colombiana para el diagnóstico, tratamiento y seguimiento de las demencias*. Recuperado el 02 de 09 de 2015, de Revista de la Asociación Colombiana de Gerontología y Geriatria: <http://goo.gl/ujwS39>
- Atabay, D. (2016). *PythonForArtificialIntelligence*. Recuperado el 23 de 08 de 2016, de Python: <http://goo.gl/MyVUAu>

- Belden, C. M., Burciu, C., & Sabbagh, M. (2014). Global Deterioration Scale (Individuals). En A. C. Michalos, *Encyclopedia of Quality of Life and Well-Being Research* (págs. 2562-2564). Holanda: Springer.
- Beleites, C., Salzer, R., & Sergo, V. (2013). Validation of Soft Classification Models using Partial Class Memberships: An Extended Concept of Sensitivity & Co. applied to the Grading of Astrocytoma Tissues. *Chemometrics and Intelligent Laboratory Systems*, 12-22.
- Bergstra, J., & Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 281-305.
- Berzal, F. (31 de 05 de 2016). *Clasificación y Predicción*. Obtenido de Minería de Datos, Inteligencia Artificial y Sistemas Expertos: <http://goo.gl/TuD6WR>
- Blockeel, H., & Struyf, J. (2002). Efficient Algorithms for Decision Tree Cross-validation. *Journal of Machine Learning Research*, 621-650.
- Borson, S., Scanlan, J., Hummel, J., Gibbs, K., Lessing, M., & Zuhr, E. (2007). Implementing routine cognitive screening of older adults in primary care: process and impact on physician behavior. *Journal of General Internal Medicine*, 811-817.
- Bramer, M. (2009). *Artificial Intelligence: An International Perspective*. New York: Springer.
- Breiman, L. (1994). *Bagging Predictors*. Recuperado el 30 de 07 de 2016, de Statistics at UC Berkeley: <http://goo.gl/A4aI4C>
- Breiman, L. (2001). Random Forests. *Machine Learning*, 5–32.
- Budde, R., Kuhlenkamp, K., Mathiassen, L., & Züllighoven, H. (1984). *Approaches to Prototyping*. Berlin: Springer-Verlag.
- CB Insights. (2016). *From Virtual Nurses To Drug Discovery: 32 Artificial Intelligence Startups In Healthcare*. Recuperado el 22 de 05 de 2016, de CB Insights: <https://goo.gl/s9ZLIX>

- Centers for Disease Control and Prevention. (2011). *Cognitive Impairment: A Call for Action, Now!* Recuperado el 04 de 08 de 2015, de Centers for Disease Control and Prevention: <http://goo.gl/aihiQK>
- Chainer. (2015). *Activation Functions*. Recuperado el 07 de 08 de 2016, de Chainer: <http://goo.gl/V78ejj>
- Chang, Q., Chen, Q., & Wang, X. (2005). Scaling Gaussian RBF kernel width to improve SVM classification. *2005 International Conference on Neural Networks and Brain*, 19 - 22.
- Chatelat, G., Desgranges, B., de la Sayette, V., Viader, F., Eustache, F., & Baron, J. C. (2003). Mild cognitive impairment: can FDG-PET predict who is to rapidly convert to Alzheimer's disease? . *Neurology*, 1374-1377.
- Cilimkovic, M. (2015). *Neural Networks and Back Propagation Algorithm*. Recuperado el 06 de 08 de 2016, de Data Mining Masters: <http://goo.gl/mcpX2q>
- Cohen, J. R., Asarnow, R. F., Sabb, F. W., Bilder, R. M., Bookheimer, S. Y., Knowlton, B. J., & Poldrack, R. A. (2010). Decoding developmental differences and individual variability in response inhibition through predictive analyses across individuals. *Frontiers in Human Neuroscience*.
- Cortes, F., Nourhashémi, F., Guérin, O., Cantet, C., Gillette-Guyonnet, S., Andrieu, S., . . . Vellas, B. (2008). Prognosis of Alzheimer's disease today: A two-year prospective study in 686 patients from the REAL-FR study. *Alzheimer's and Dementia*, 22–29.
- Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., & Singer, Y. (2006). Online Passive-Aggressive Algorithms. *Journal of Machine Learning Research*, 551-585.
- Darken, C., Chang, J., & Moody, J. (1992). Learning rate schedules for faster stochastic gradient search. *Neural Networks for Signal Processing [1992] II., Proceedings of the 1992 IEEE-SP Workshop*, 3-12.

- Dauphin, Y., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., & Bengio, Y. (2014). *Identifying and attacking the saddle point problem in high-dimensional non-convex optimization*. Recuperado el 06 de 08 de 2016, de arXiv: <http://goo.gl/ALTAF6>
- Davis, J., & Goadrich, M. (2006). The Relationship Between Precision-Recall and ROC Curves. *Proceedings of the 23rd International Conference on Machine Learning*, 233-240.
- Doi, K. (2007). Computer-aided diagnosis in medical imaging: Historical review, current status and future potential. *Computerized Medical Imaging and Graphics*, 198-211.
- Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 2121-2159.
- Duthey, B. (2013). *Priority Medicines for Europe and the World "A Public Health Approach to Innovation"*. Bruselas: World Health Organization.
- EPIA. (2015). *Artificial Intelligence in Medicine*. Recuperado el 17 de 09 de 2015, de EPIA 2015 - 17th Portuguese Conference on Artificial Intelligence: <http://goo.gl/sYe2OH>
- Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters* 27, 861–874.
- Flicker, C., Ferris, S. H., & Reisberg, B. (1991). Mild cognitive impairment in the elderly: Predictors of dementia. *Neurology*, 1006-1009.
- Florida Health Care Association. (2006). *The Global Deterioration Scale for Assessment of Primary Degenerative Dementia*. Recuperado el 08 de 09 de 2015, de Florida Health Care Association: <http://goo.gl/5lpMhN>
- Folstein, M. F., Folstein, S. E., & McHugh, P. R. (1975). "Mini-mental state": a practical method for grading the cognitive state of patients for the clinician. *Journal of Psychiatric Research*, 189-198.
- Forrest, C. (2015). *5 AI startups to watch in 2015*. Recuperado el 22 de 05 de 2016, de TechRepublic: <http://goo.gl/sYxNQ0>

- Gauthier, S., Reisberg, B., Zaudig, M., Petersen, R. C., Ritchie, K., . . . Winblad, B. (2006). Mild Cognitive Impairment. *The Lancet*, 1262-1270.
- Gavidia, G., Soudah, E., & Oñate, E. (2012). *Clasificadores basados en Máquinas de Soporte Vectorial para el Diagnóstico y Predicción de la Enfermedad de Alzheimer*. Barcelona: Centro Internacional de Métodos Numéricos en Ingeniería.
- Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, 3–42.
- Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep Sparse Rectifier Neural Networks. *Journal of Machine Learning Research*, 315-323.
- Glover, L. (2014). Why Does an MRI Cost So Darn Much? *Time*.
- Gold, D. A. (2012). An examination of instrumental activities of daily living assessment in older adults and mild cognitive impairment. *Journal of Clinical and Experimental Neuropsychology*, 11-34.
- Gómez, J. F., Curcio, C. L., & Gómez, D. E. (1995). *Evaluación de la salud de los ancianos*. Manizales: Talleres litográficos de Información y Publicaciones.
- Haller, S. (2014). MRI Technique Detects Evidence of Cognitive Decline Before Symptoms Appear. *RSNA Press Release*, 1-2.
- Hand, D. J., & Till, R. J. (2001). A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems. *Machine Learning*, 171-186.
- Hsu, K.-W. (2012). Hybrid Ensembles of Decision Trees and Artificial Neural Networks. *CyberneticsCom*, 25-29.
- Hughes, C. P., Berg, L., & Danziger, W. L. (1982). A new clinical scale for the staging of dementia. *The British Journal of Psychiatry*, 566-572.
- Hughes, C. P., Berg, L., Danziger, W. L., Coben, L. A., & Martin, R. L. (1982). A new clinical scale for the staging of dementia. *British Journal of Psychiatry*, 17-24.

- IBM. (2014). *Test plan template reference*. Recuperado el 20 de 10 de 2014, de IBM Knowledge Center: <http://goo.gl/zVViewY>
- Jack, C. R., Bernstein, M. A., & Fox, N. C. (2008). The alzheimer's disease neuroimaging initiative (adni). *Magn Reson Imaging*, 685-691.
- Jack, C. R., Petersen, R. C., Xu, Y. C., O'Brien, P. C., Smith, G. E., Ivnik, R. J., . . . Kokmen, E. (1999). Prediction of AD with MRI-based hippocampal volume in mild cognitive impairment. *Neurology*, 1397-1403.
- Jeffrey, S. (2013). *Neurologist Shortage Projected to Worsen by 2025*. Recuperado el 2015 de 09 de 2015, de Medscape: <http://goo.gl/8Tek4K>
- Jin, E. T., Strauss, E., & Sherman, E. (2011). Clinical Dementia Rating. En J. Kreutzer, J. DeLuca, & B. Caplan, *Encyclopedia of Clinical Neuropsychology* (págs. 587-590). New York: Springer.
- Jin, X., Xu, C., Feng, J., Wei, Y., Xiong, J., & Yan, S. (2016). *Deep Learning with S-shaped Rectified Linear Activation Units*. Recuperado el 07 de 08 de 2016, de arXiv: <http://goo.gl/F2WvjX>
- Keras. (2016). *FAQ*. Recuperado el 23 de 08 de 2016, de Keras: <http://goo.gl/w7vF7T>
- Keras. (2016). *Keras: Deep Learning library for Theano and TensorFlow*. Recuperado el 21 de 07 de 2016, de Keras: <http://keras.io/>
- Kingma, D. P., & Ba, J. L. (2015). ADAM: A Method for Stochastic Optimization. *International Conference on Learning Representations*, 1-13.
- Kingsford, C., & Salzberg, S. L. (2008). What are decision trees? *Nat Biotechnol*, 1011–1013.
- Korf, E. S., Wahlund, L. O., Visser, P. J., & Scheltens, P. (2004). Medial temporal lobe atrophy on MRI predicts dementia in patients with mild cognitive impairment. *Neurology*, 94-100.

- Krenker, A., Bešter, J., & Kos, A. (2011). Introduction to the Artificial Neural Networks. En K. Suzuki, *Artificial Neural Networks - Methodological Advances and Biomedical Applications* (págs. 1-18). InTech.
- Krinsky-McHale, S. J., & Silverman, W. (2013). Dementia and mild cognitive impairment in adults with intellectual disability: Issues of diagnosis. *Developmental Disabilities Research Reviews*, 31-42.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems* 25, 1106-1114.
- Langa, K. M., & Levine, D. A. (2014). The Diagnosis and Management of Mild Cognitive Impairment: A Clinical Review. *JAMA*, 2551-2561.
- Lasagne. (2015). *Non-linear activation functions for artificial neurons*. Recuperado el 07 de 08 de 2016, de Lasagne: <http://goo.gl/7I3JCY>
- Lawton, M. P., & Brody, E. M. (1969). Assessment of older people: Self-maintaining and instrumental activities of daily. *The Gerontologist*, 179-186.
- Lebedev, A. V. (2014). *BORA - UiB*. Recuperado el 15 de 06 de 2015, de Cognitive impairment in neurodegenerative diseases: <http://goo.gl/AxWn1w>
- Ledley, R. S., & Lusted, L. B. (1959). Reasoning foundations of medical diagnosis. *Science*, 9-21.
- Lee, E. K., Wu, T.-L., Goldstein, F., & Levey, A. (2012). Predictive Model for Early Detection of Mild Cognitive Impairment and Alzheimer's Disease. En P. M. Pardalos, T. F. Coleman, & P. Xanthopoulos, *Optimization and Data Analysis in Biomedical Informatics* (págs. 83-97). New York: Springer.
- Libon, D. J., Xie, S. X., Eppig, J., Wicas, G., Lamar, M., Lippa, C., . . . Wambach, D. M. (2010). The heterogeneity of mild cognitive impairment: A neuropsychological analysis. *Journal of the International Neuropsychological Society*, 84-93.

- Liu, S., Liu, S., Cai, W., Pujol, S., Kikinis, R., & Feng, D. (2014). Early diagnosis of Alzheimer's disease with deep learning. *Biomedical Imaging (ISBI), 2014 IEEE 11th International Symposium on*, 1015-1018.
- Liu, S., Song, Y., Cai, W., Pujol, S., Kikinis, R., Wang, X., & Feng, D. (2013). Multifold Bayesian Kernelization in Alzheimer's Diagnosis. En K. Mori, I. Sakuma, Y. Sato, C. Barillot, & N. Navab, *Medical Image Computing and Computer-Assisted Intervention –MICCAI2013* (págs. 303-310). Londres: Springer.
- Mapi Research Institute. (2006). *Valoración Clínica de la Demencia (CRD)*. Recuperado el 15 de 09 de 2015, de Alzheimer's Disease Research Center: <http://goo.gl/I9SBmk>
- Mariani, E., Monastero, R., & Mecocci, P. (2007). Mild Cognitive Impairment: A Systematic Review. *Journal of Alzheimer's Disease*, 23-35.
- MathWorks. (2016). *Choose Classifier Options*. Recuperado el 27 de 07 de 2016, de MathWorks: <http://goo.gl/DEdjK8>
- MathWorks. (2016). *Ensemble Methods*. Recuperado el 04 de 08 de 2016, de MathWorks: <http://goo.gl/7XOPbF>
- MathWorks. (2016). *MathWorks - Makers of MATLAB and Simulink*. Recuperado el 21 de 07 de 2016, de Statistics and Machine Learning Toolbox: <http://goo.gl/lkg5u5>
- MathWorks. (2016). *Support Vector Machines for Binary Classification*. Recuperado el 24 de 07 de 2016, de MathWorks: <http://goo.gl/trS5DQ>
- Mccormick, C. (2013). *K-Fold Cross-Validation, with Matlab Code*. Recuperado el 15 de 10 de 2015, de Computer Vision and Machine Learning Projects and Tutorials: <https://goo.gl/wdv01Q>
- McDade, E. M., & Petersen, R. C. (2015). *Mild cognitive impairment: Epidemiology, pathology, and clinical assessment*. Recuperado el 16 de 05 de 2016, de UpToDate: <http://goo.gl/QD5L2E>

Millington, I., & Funge, J. (2009). *Artificial Intelligence for Games*. Massachusetts: Morgan Kaufmann Publishers.

Ministerio de Salud de Colombia. (1999). *Resolución 1995 de 1999*.

Mueller, S., Weiner, M., Thal, L., Petercen, R. C., Clifford, J., Jagust, W., . . . Beckett, L. (2005). The alzheimer's disease neuroimaging initiative. *Neuroimaging Clinics of North America*, 869-877.

Nasreddine, Z. (2004). *Montreal Cognitive Assessment (MoCA)*. Recuperado el 15 de 09 de 2015, de Mocatest: <http://goo.gl/zUmalC>

Nasreddine, Z. S., Phillips, N. A., Bédirian, V., Charbonneau, S., Whitehead, V., Collin, I., . . . Chertkow, H. (2005). The Montreal Cognitive Assessment, MoCA: A Brief Screening Tool For Mild Cognitive Impairment. *Journal of the American Geriatrics Society*, 695-699.

Natekin, A., & Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics*, 7-21.

National Institutes of Health. (s.f.). *Cognitive and Emotional Health Project: The Healthy Brain*. Recuperado el 07 de 08 de 2015, de National Institutes of Health: <http://goo.gl/67NILh>

NCSS Statistical Software. (2016). *Ridge Regression*. Recuperado el 24 de 07 de 2016, de NCSS Statistical Software: <http://goo.gl/Bwzg6m>

Newell, A., & Simon, H. (1976). Computer Science as Empirical Inquiry: Symbols and Search. *Communication of ACM*, 113-126.

Ollier, D. (2015). *12 Ways Artificial Intelligence Will Transform Health Care*. Recuperado el 22 de 05 de 2016, de Hospitals & Health Networks: <http://goo.gl/WUttwwQ>

Onieva, E., Santos, I., Osaba, E., Quintián, E., & Corchado, E. (2015). *Hybrid Artificial Intelligent Systems: 10th International Conference*. Bilbao: Springer.

- OpenEye Scientific. (2015). *Drawing ROC Curve*. Recuperado el 31 de 05 de 2016, de OpenEye Python Cookbook: <https://goo.gl/1kXhyM>
- Owen, S. (2015). *Python vs R for machine learning*. Recuperado el 23 de 08 de 2016, de DataScience Stack Exchange: <http://goo.gl/c6luKp>
- Parbst, F. (1984). Experience with Prototyping in an IBM-Based Installation. En R. Budde, K. Kuhlenkamp, L. Mathiassen, & H. Züllighoven, *Approaches to Prototyping* (págs. 152-164). Berlin: Springer-Verlag.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 2825-2830.
- Petersen, R. C. (2004). Mild cognitive impairment as a diagnostic entity. *Journal of Internal Medicine*, 183-194.
- Petersen, R. C., & Morris, J. C. (2005). Mild Cognitive Impairment as a Clinical Entity and Treatment Target. *Arch Neurol*, 1160-1163.
- Petersen, R. C., Doody, R., Kurz, A., Mohs, R. C., Morris, J. C., Rabins, P. V., . . . Winblad, B. (2001a). Current concepts in mild cognitive impairment. *Arch Neurol*, 1985-1992.
- Petersen, R. C., Smith, G. E., Ivnik, R. J., Tangalos, E. G., Schaid, D. J., Thibodeau, S. N., . . . Kurland, L. T. (1995). Apolipoprotein E status as a predictor of the development of Alzheimer's disease in memory-impaired individuals. *JAMA*, 1274-1278.
- Petersen, R. C., Smith, G. E., Waring, S. C., Ivnik, R. J., Tangalos, E. G., & Kokmen, E. (1999). Mild Cognitive Impairment Clinical Characterization and Outcome. *Arch Neurol*, 303-308.
- Petersen, R. C., Stevens, J. C., Ganguli, M., Tangalos, E. G., Cummings, J. L., & DeKosky, S. T. (2001b). Practice parameter: Early detection of dementia: Mild cognitive impairment (an evidence-based review). *Neurology*, 1133-1142.

- Powers, D. M. (2007). *Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation*. Recuperado el 28 de 07 de 2016, de Flinders University: <https://goo.gl/IsePr0>
- Powers, D. M. (2015). *What the F-measure doesn't measure....* Recuperado el 15 de 08 de 2016, de arXiv: <https://goo.gl/EJ1XWv>
- Qt Company. (2016). *Our company at a glance*. Recuperado el 23 de 08 de 2016, de Qt: <http://goo.gl/SVrwbi>
- Ramírez, M. E. (1995). *El interrogatorio en la Semiología*. Santafé de Bogotá: ECOE Ediciones.
- Rebizant, W., Szafran, J., & Wiszniewski, A. (2011). *Digital Signal Processing in Power System Protection and Control*. Londres: Springer.
- Refaeilzadeh, P., Tang, L., & Liu, H. (2009). *Encyclopedia of Database Systems*. Atlanta: Springer.
- Reisberg, B., Ferris, S. H., de Leon, M. J., & Crook, T. (1982). The Global Deterioration Scale for assessment of primary degenerative dementia. *The American Journal of Psychiatry*, 1136–1139.
- Richards, M. (2015). *Software architecture patterns*. Recuperado el 22 de 08 de 2016, de O'Reilly: <https://goo.gl/cgAO8l>
- Rodríguez Silva, H. (2010). La relación médico-paciente. *MediSur*, 41-44.
- Roe, B. P., Yang, H.-J., & Zhu, J. (2005). Boosted Decision Trees, a Powerful Event Classifier. En L. Lyons, & M. K. Ünel, *Statistical Problems in Particle Physics, Astrophysics and Cosmology* (págs. 139-142). Londres: Imperial College Press.
- Rouse, M. (2016). *machine learning*. Recuperado el 09 de 09 de 2016, de WhatIs: <http://goo.gl/3Pg8X7>

- Roy, E. (2013). Cognitive Impairment. En M. D. Gellman, & J. R. Turner, *Encyclopedia of Behavioral Medicine* (págs. 449-450). New York: Springer.
- Ruder, S. (2016). *An overview of gradient descent optimization algorithms*. Recuperado el 06 de 08 de 2016, de Sebastian Ruder: <http://goo.gl/JYp0Nn>
- Ruiz de Sánchez, C., Pérez, G. E., & Medina, J. M. (2013). *Diagnóstico y tratamiento integral de las demencias*. Bogotá: Asociación Colombiana de Neurología.
- Ruiz, C., Pérez, G. E., & Medina, J. M. (2013). *Diagnóstico Integral y tratamiento de las demencias*. Bogotá: Asociación Colombiana de Neurología.
- Rumelheart, D. E., & McClelland, J. (1986). *Parallel distributed processing: explorations in the microstructure of cognition*. Massachusetts: MIT Press Cambridge.
- Russel, S. (2013). *Artificial Intelligence*. California: Content Technologies.
- Saabas, A. (2014). *Selecting good features – Part III: random forests*. Recuperado el 13 de 08 de 2016, de Diving into data: <http://goo.gl/c1kTe6>
- Saito, T., & Rehmsmeier, M. (2015). The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLoS ONE*, 1-21.
- Schapire, R. E. (2003). The Boosting Approach to Machine Learning An Overview. *Nonlinear Estimation and Classification*, 149-171.
- Schneider, J. (1977). *Cross Validation*. Recuperado el 14 de 10 de 2015, de Carneige Mellon University: <https://goo.gl/JyfWv7>
- Scikit-learn. (2014). *Model persistence*. Recuperado el 23 de 08 de 2016, de Scikit-learn: <http://goo.gl/eTQrQX>
- Scikit-learn. (2014). *Who is using scikit-learn?* Recuperado el 23 de 08 de 2016, de Scikit-learn: <http://goo.gl/licuVp>

- Scikit-learn. (2015). *User Guide*. Recuperado el 21 de 07 de 2016, de Scikit-learn: <http://goo.gl/RuApOI>
- Seiffert, C., Khoshgoftaar, T. M., Hulse, J. V., & Napolitano, A. (2009). RUSBoost: A Hybrid Approach to Alleviating Class Imbalance. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 185-197.
- Sherrell, L. (2013). *Evolutionary Prototyping*. Holanda: Springer.
- Shimeall, T. J. (1990). *TWIRP: Testing Within Iterative Rapid Prototyping*. Monterey: Naval Postgraduate School.
- Simmers, L., Simmers-Nartker, K., & Simmers-Kobelak, S. (2014). *DHO: Health Science, 8th Edition*. Stamford: Cengage Learning.
- Smolensky, P. (1988). On the proper treatment of connectionism. *Behavioral and Brain Sciences*, 1-23.
- Stern, Y., Gurland, B., Tatemichi, T. K., Tang, M. X., Wilder, D., & Mayeux, R. (1994). Influence of education and occupation on the incidence of Alzheimer's disease. *JAMA*, 1004-1010.
- Stoub, T. R., Bulgakova, M., Leurgans, S., Bennett, D. A., Fleischman, D., Turner, D. A., & de Toledo-Morrell, L. (2005). MRI predictors of risk of incident Alzheimer disease: a longitudinal study. *Neurology*, 1520-1524.
- Suk, H.-I., & Shen, D. (2013). Deep Learning-Based Feature Representation for AD/MCI Classification. En K. Mori, I. Sakuma, Y. Sato, C. Barillot, & N. Navab, *Medical Image Computing and Computer-Assisted Intervention –MICCAI2013* (págs. 583-590). Londres: Springer.
- Sun, R. (2000). *Artificial intelligence Connectionist and Symbolic Approaches*. Recuperado el 17 de 09 de 2015, de Cognitive Science Department - Rensselaer Polytechnic Institute (RPI): <http://goo.gl/icQUIF>

- Sun, Y., Tang, Y., Ding, S., Lv, S., & Cui, Y. (2011). Diagnose the mild cognitive impairment by constructing Bayesian network with missing data. *Expert Systems with Applications*, 442-449.
- Tan, P.-N., Steinbach, M., & Kumar, V. (2006). *Classification: Basic Concepts, Decision Trees, and Model Evaluation*. Recuperado el 07 de 27 de 2016, de Computer Science & Engineering: <https://goo.gl/ZSDK6Q>
- The Hartford Institute for Geriatric Nursing. (1999). *try this: Best Practices in Nursing Care to Older Adults*. Recuperado el 26 de 08 de 2015, de MountSinai: <http://goo.gl/MiTMtA>
- The Hartford Institute for Geriatric Nursing. (2012). *try this: Best Practices in Nursing Care to Older Adults*. Recuperado el 26 de 08 de 2015, de ConsultGeri: <http://goo.gl/R9aHro>
- The Hartford Institute for Geriatric Nursing. (2013). *try this: Best Practices in Nursing Care to Older Adults*. Recuperado el 25 de 08 de 2015, de ConsultGeri: <http://goo.gl/TDexGd>
- Umer, R. (2011). *University of Georgia*. Recuperado el 09 de 07 de 2015, de Electronic Theses and Dissertations: <https://goo.gl/4SgjY5>
- University of Illinois. (2014). *When Healthcare and Computer Science Collide*. Recuperado el 17 de 05 de 2016, de Health Informatics: <http://goo.gl/YxG13P>
- Voisin, T., Touchon, J., & Vellas, B. (2003). Mild cognitive impairment: a nosological entity? *Curr Opin Neurol*, 43-45.
- Weiner, M. W., Veitch, D. P., Aisen, P. S., Beckett, L. A., Cairns, N. J., Cedarbaum, J., . . . Trojanowski, J. Q. (2015). 2014 Update of the Alzheimer's Disease Neuroimaging Initiative: A review of papers published since its inception. *Alzheimers Dement*, e1-e120.

- Weiner, M. W., Veitch, D. P., Aisen, P. S., Beckett, L. A., Cairns, N. J., Green, R. C., . . . Trojanowski, J. Q. (2012). The Alzheimer's Disease Neuroimaging Initiative: A review of papers published since its inception. *Alzheimer's & Dementia*, S1-S68.
- Werner, P., & Korczyn, A. D. (2008). Mild cognitive impairment: Conceptual, assessment, ethical, and social issues. *Clinical Interventions in Aging*, 413-420.
- Williams, J. A., Weakley, A., Cook, D. J., & Schmitter-Edgecombe, M. (2013). Machine Learning Techniques for Diagnostic Differentiation of Mild Cognitive Impairment and Dementia. *Workshops at the Twenty-Seventh AAAI Conference on Artificial Intelligence* (págs. 71-76). Bellevue: AAAI Publications.
- Wilson, B. (2012). *The Machine Learning Dictionary*. Obtenido de Computer Science and Engineering: <http://goo.gl/jwvTf9>
- Winder-Rhodes, S. E., Barker, R. A., & Williams-Gray, C. H. (2015). The pathophysiological and prognostic heterogeneity of mild cognitive impairment in Parkinson's disease. En M. Emre, *Cognitive Impairment and Dementia in Parkinson's Disease* (págs. 223-226). Oxford: Oxford University Press.
- Wood, D. P., & Kang, K. C. (1992). *A Classification and Bibliography of Software Prototyping*. Pittsburgh: Carnegie Mellon University.
- World Health Organization. (2012). *Mental Health*. Recuperado el 10 de 07 de 2015, de World Health Organization: <http://goo.gl/UR16mZ>
- Wu, M.-S., Lan, T.-H., Chen, C.-M., Chiu, H.-C., & Lan, T.-Y. (2011). Socio-demographic and health-related factors associated with cognitive impairment in the elderly in Taiwan. *BMC Public Health*, 11-22.
- Ye, B. S., Chin, J., Kim, S. Y., Lee, J.-S., Kim, E.-J., Lee, Y., . . . Seo, S. W. (2015). The Heterogeneity and Natural History of Mild Cognitive Impairment of Visual Memory Predominant Type. *Journal of Alzheimer's Disease*, 143-152.

- Yesavage, J. A., Brink, T. L., Rose, T. L., Lum, O., Huang, V., Adey, M., & Leirer, V. O. (1982). Development and validation of a geriatric depression screening scale: a preliminary report. *Journal of Psychiatric Research*, 37-49.
- Yin, Z., Zhao, Y., Lu, X., & Duan, H. (2015). A Hybrid Intelligent Diagnosis Approach for Quick Screening of Alzheimer's Disease Based on Multiple Neuropsychological Rating Scales. *Computational and Mathematical Methods in Medicine*, 27-40.
- Zadikoff, C., Fox, S. H., Tang-Wai, D. F., Thomsen, T., de Bie, R. M., Wadia, P., . . . Marras, C. (2008). A comparison of the mini mental state exam to the montreal cognitive assessment in identifying cognitive deficits in Parkinson's disease. *Movement Disorders*, 297-299.
- Zeiler, M. D. (2012). *ADADELTA: An Adaptive Learning Rate Method*. Recuperado el 10 de 08 de 2016, de arXiv: <http://goo.gl/pCLYEm>
- Zhu, J., Rosset, S., Zou, H., & Hastie, T. (2006). *Multi-class AdaBoost*. Recuperado el 31 de 07 de 2016, de Stanford University: <https://goo.gl/3eQPiv>
- Zisserman, A. (2015). *SVM dual, kernels and regression*. Recuperado el 24 de 07 de 2016, de Information Engineering: <http://goo.gl/mCuLXS>

9. ANEXOS

1 Presupuesto del proyecto

Rubro	Nombre	Descripción	Valor
Hardware	Computador Portátil	Equipo de cómputo para realizar la documentación, el desarrollo y las pruebas del proyecto	\$1.750.000
		TOTAL DEL RUBRO	\$1.750.000
Personal	Ing. Juan Camilo Flórez	Responsable de realizar el proyecto	\$7.245.000
	Msc. Santiago Murillo	Asesor del proyecto	\$1.291.650
	Evaluadores	Responsables de evaluar el proyecto	\$1.248.600
		TOTAL DEL RUBRO	\$9.785.250
Materiales	Fotocopias	Copias de libros, artículos e información relacionada con el proyecto	\$20.000
	CD/DVD	Discos para almacenar y entregar la información derivada del proyecto	\$10.000
	USB	Dispositivo para trasladar y resguardar la información del proyecto	\$15.000
	Internet	Servicio de internet requerido para el desarrollo del proyecto	\$600.000
		TOTAL DEL RUBRO	\$645.000
Publicaciones	Gastos de publicación	Gastos relacionados con la elaboración del artículo y su respectiva publicación en una revista indexada	\$1.000.000
		TOTAL DEL RUBRO	\$1.000.000
Transporte	Pasajes terrestres	Movilización del personal en los 14 municipios del departamento de Caldas	\$3.000.000
		TOTAL DEL RUBRO	\$3.000.000
		TOTAL DEL PROYECTO	\$16.180.250

Tabla 58 Presupuesto asociado al proyecto.

Recurso	Valor
Ing. Juan Camilo Flórez	\$11.500/hora * 18 horas/semana * 35 semanas
Msc. Santiago Murillo	\$1.291.650
Evaluadores	\$1.248.600
Internet	\$75.000/mes * 8 meses

Tabla 59 Costo unitario por recurso.

2 Formato de plan de pruebas

IBM (2014) refiere que el formato básico de plan de pruebas contiene las siguientes secciones:

- a) Resumen: descripción detallada del plan de pruebas.
- b) Requerimientos: conjunto de requerimientos incluidos en el plan de pruebas.

Identificador	Nombre	Iteración N°

Tabla 60 Plantilla para registrar los requerimientos en el plan de pruebas.

- c) Calendario de pruebas: cronograma de las actividades de pruebas e iteraciones.

Fecha	Actividad	Iteración N°

Tabla 61 Plantilla para registrar las fechas y actividades en el plan de pruebas.

- d) Bancos de pruebas: colección de casos de prueba, agrupados con fines de ejecución.

ID Banco de pruebas	Descripción	Iteración N°

Tabla 62 Plantilla para registrar los bancos de pruebas en el plan de pruebas.

- e) Casos de prueba: especificación de casos de prueba asociados con el plan.

ID Banco de pruebas	
ID Caso de prueba	
Descripción	
Valores de entrada	
Pasos	
Resultado esperado	

Resultado actual

Tabla 63 Plantilla para especificar los casos de prueba en el plan de pruebas.

- f) Adjuntos: archivos y documentos que se anexan al plan de pruebas, como capturas de pantalla, material de soporte, entre otros.

3 Prueba cognitiva: IADL de Lawton

Escala de actividades instrumentales de la vida diaria de Lawton modificada por el grupo de Geriatría de la Universidad de Caldas (Gómez, Curcio, & Gómez, 1995).

¿Realiza la actividad?	Sí						No	
	Sin dificultad		Con dificultad, pero sin ayuda		Solamente con ayuda			
	previo	actual	previo	actual	previo	actual	previo	actual
Salir de la casa								
Caminar por las habitaciones								
Usar el teléfono								
Ir de compras								
Tomar su propio medicamento								
Abrir y cerrar ventanas								
Manejar su propio dinero								
Encender y apagar la radio y la TV								
Manipular interruptores								
Hacer su propia comida								
Manipular llaves								
Cortarse las uñas								
Hacer trabajo liviano en la casa, lavar platos, etc.								
Hacer trabajo pesado en la casa, lavar ventanas, pisos, en general limpiar la casa								
TOTAL								
PREVIO								

Tabla 64 Escala de IADL de Lawton modificada (Gómez, Curcio, & Gómez, 1995).

4 Prueba cognitiva: MMSE

Mini Examen del Estado Mental adaptado al idioma español para Colombia (Ruiz, Pérez, & Medina, 2013).

Orientación

/5 Diga en que (año) (mes) (día) (día/semana) (hora) nos encontramos?

/5 Diga en que (país) (ciudad) (departamento) (hospital) (piso) nos encontramos?

Memoria

/3 Diga tres palabras: casa / mesa / árbol. Un segundo para cada una. Luego pida al paciente que las repita. Un punto por cada una. Repítalas hasta que el paciente las registre. Anote el número de ensayos: _____

Atención y Cálculo

/5 Pida al paciente restar de 7 en 7 desde 100 en forma sucesiva. Pare a la 5 respuesta, registre un punto por cada respuesta correcta.

93 – 86 – 79 – 72 – 63

(Si es analfabeta → meses del año al revés)

Diciembre, noviembre, octubre, septiembre, agosto

Evocación

/3 Recuerdo de las tres palabras antes presentadas. Registre el número de palabras – y las palabras que recuerda: _____

Lenguaje

/2 Denominar dos objetos (reloj, lápiz) registre un punto por cada acierto

/1 Repetir: En un trigal habían cinco perros. Registre un punto si acierta

/3 Comprensión: Obedecer una orden en 3 etapas

«Tome la hoja con la mano derecha, dóblela por la mitad y póngala en el piso»

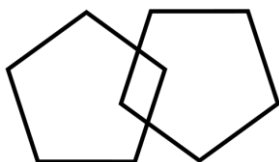
Registre un punto por cada orden ejecutada correctamente

Lea y obedezca las siguientes órdenes: (un punto por cada acierto)

/1 «Cierre los ojos»

/1 «Escriba una frase»

/1 «Copie el diseño»



5 Prueba cognitiva: MoCA

Evaluación cognitiva de Montreal adaptada al idioma español (Nasreddine Z. , 2004).

MONTREAL COGNITIVE ASSESSMENT (MOCA)
(EVALUACIÓN COGNITIVA MONTREAL)

NOMBRE:
Nivel de
estudios:
Sexo:

Fecha de nacimiento:

FECHA:

VISUOESPACIAL / EJECUTIVA		Copiar el cubo		Dibujar un reloj (Once y diez) (3 puntos)		Puntos																											
				[] [] Contorno Números Agujas		___/5																											
IDENTIFICACIÓN																																	
						___/3																											
MEMORIA		<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%; padding: 2px;">Lea la lista de palabras, el paciente debe repetirlas. Haga dos intentos. Recuérdese las 5 minutos más tarde.</td> <td style="width: 10%;"></td> <td style="width: 10%;">ROSTRO</td> <td style="width: 10%;">SEDA</td> <td style="width: 10%;">IGLESIA</td> <td style="width: 10%;">CLAVEL</td> <td style="width: 10%;">ROJO</td> </tr> <tr> <td>1er intento</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>2º intento</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </table>					Lea la lista de palabras, el paciente debe repetirlas. Haga dos intentos. Recuérdese las 5 minutos más tarde.		ROSTRO	SEDA	IGLESIA	CLAVEL	ROJO	1er intento							2º intento							Sin puntos					
Lea la lista de palabras, el paciente debe repetirlas. Haga dos intentos. Recuérdese las 5 minutos más tarde.		ROSTRO	SEDA	IGLESIA	CLAVEL	ROJO																											
1er intento																																	
2º intento																																	
ATENCIÓN		<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; padding: 2px;">Lea la serie de números (1 número/seg.)</td> <td style="width: 50%; padding: 2px;">El paciente debe repetirla. [] 2 1 8 5 4</td> </tr> <tr> <td></td> <td style="padding: 2px;">El paciente debe repetirla a la inversa. [] 7 4 2</td> </tr> </table>					Lea la serie de números (1 número/seg.)	El paciente debe repetirla. [] 2 1 8 5 4		El paciente debe repetirla a la inversa. [] 7 4 2	___/2																						
Lea la serie de números (1 número/seg.)	El paciente debe repetirla. [] 2 1 8 5 4																																
	El paciente debe repetirla a la inversa. [] 7 4 2																																
Lea la serie de letras. El paciente debe dar un golpecito con la mano cada vez que se diga la letra A. No se asignan puntos si ≥ 2 errores.		[] FBACMNAAJKLBAFAKDEAAAJAMOFAB					___/1																										
Restar de 7 en 7 empezando desde 100.		[] 93	[] 86	[] 79	[] 72	[] 65	___/3																										
		4 o 5 sustracciones correctas: 3 puntos, 2 o 3 correctas: 2 puntos, 1 correcta: 1 punto, 0 correctas: 0 puntos.																															
LENGUAJE		Repetir: El gato se esconde bajo el sofá cuando los perros entran en la sala. [] Espero que él le entregue el mensaje una vez que ella se lo pida. []					___/2																										
Fluidez del lenguaje. Decir el mayor número posible de palabras que comiencen por la letra "P" en 1 min.		[] _____ (N ≥ 11 palabras)					___/1																										
ABSTRACCIÓN		Similitud entre p. ej. manzana-naranja = fruta [] tren-bicicleta [] reloj-regla					___/2																										
RECUERDO DIFERIDO		<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%; padding: 2px;">Debe acordarse de las palabras SIN PISTAS</td> <td style="width: 10%;">ROSTRO</td> <td style="width: 10%;">SEDA</td> <td style="width: 10%;">IGLESIA</td> <td style="width: 10%;">CLAVEL</td> <td style="width: 10%;">ROJO</td> <td rowspan="3" style="width: 20%; padding: 2px; text-align: center; font-size: small;">Puntos por recuerdos SIN PISTAS únicamente</td> </tr> <tr> <td></td> <td style="text-align: center;">[]</td> <td style="text-align: center;">[]</td> <td style="text-align: center;">[]</td> <td style="text-align: center;">[]</td> <td style="text-align: center;">[]</td> </tr> <tr> <td style="text-align: center; padding: 2px;">Optativo</td> <td style="padding: 2px;">Pista de categoría</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td style="padding: 2px;">Pista elección múltiple</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </table>					Debe acordarse de las palabras SIN PISTAS	ROSTRO	SEDA	IGLESIA	CLAVEL	ROJO	Puntos por recuerdos SIN PISTAS únicamente		[]	[]	[]	[]	[]	Optativo	Pista de categoría						Pista elección múltiple						___/5
Debe acordarse de las palabras SIN PISTAS	ROSTRO	SEDA	IGLESIA	CLAVEL	ROJO	Puntos por recuerdos SIN PISTAS únicamente																											
	[]	[]	[]	[]	[]																												
Optativo	Pista de categoría																																
	Pista elección múltiple																																
ORIENTACIÓN		<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%; padding: 2px;">[]</td> <td style="width: 20%; padding: 2px;">Día del mes (fecha)</td> <td style="width: 10%; padding: 2px;">[]</td> <td style="width: 10%; padding: 2px;">Mes</td> <td style="width: 10%; padding: 2px;">[]</td> <td style="width: 10%; padding: 2px;">Año</td> <td style="width: 10%; padding: 2px;">[]</td> <td style="width: 10%; padding: 2px;">Día de la semana</td> <td style="width: 10%; padding: 2px;">[]</td> <td style="width: 10%; padding: 2px;">Lugar</td> <td style="width: 10%; padding: 2px;">[]</td> <td style="width: 10%; padding: 2px;">Localidad</td> </tr> </table>					[]	Día del mes (fecha)	[]	Mes	[]	Año	[]	Día de la semana	[]	Lugar	[]	Localidad	___/6														
[]	Día del mes (fecha)	[]	Mes	[]	Año	[]	Día de la semana	[]	Lugar	[]	Localidad																						

© Z. Nasreddine MD Versión 07 noviembre 2004

www.mocatest.org

Normal ≥ 26 / 30

TOTAL

___/30

Añadir 1 punto si tiene ≤ 12 años de estudios

6 Prueba cognitiva: CDR

Escala Clínica de Demencia adaptada al idioma español (Mapi Research Institute, 2006).

Valoración Clínica de Demencia	0	0,5	1	2	3
--------------------------------	---	-----	---	---	---

	Deterioro				
	Ninguno 0	Dudoso 0,5	Leve 1	Moderado 2	Severo 3
Memoria	Sin pérdida de memoria o leves olvidos inconstantes	Olvidos leves constantes; recolección parcial de eventos; olvidos «benignos»	Pérdida moderada de memoria; más marcada para eventos recientes; el defecto interfiere con las actividades diarias	Pérdida grave de memoria; sólo retiene materias con mucho aprendizaje; materias nuevas se pierden con rapidez	Pérdida grave de memoria; sólo retiene fragmentos
Orientación	Completamente orientado	Completamente orientado pero con leve dificultad para las relaciones temporales	Dificultad moderada con las relaciones temporales; orientado en el lugar del examen; puede tener algo de desorientación geográfica en otro lugar	Dificultad grave con las relaciones temporales; habitualmente desorientado en el tiempo; a menudo en el lugar	Orientado sólo en persona
Razonamiento y solución de problemas	Resuelve los problemas diarios y se encarga bien de los negocios y finanzas; razonamiento bueno con relación al comportamiento previo	Leve dificultad para resolver problemas, similitudes y diferencias	Dificultad moderada para hacer frente a problemas, similitudes y diferencias; razonamiento social habitual mantenido	Grandes dificultades para hacer frente a problemas, similitudes y diferencias; juicio social habitual limitado	Incapaz de razonar o resolver problemas
Actividades fuera de casa	Función independiente a su nivel habitual en el trabajo, compras, voluntariado y agrupaciones sociales	Leve dificultad en estas actividades	Incapaz de ser independiente en estas actividades, aunque aún puede participar en alguna; parece normal a primera vista	Incapaz de ser independiente fuera de casa Parece estar lo suficientemente bien como para realizar funciones fuera de casa	
Actividades domésticas y aficiones	Vida en casa, aficiones e intereses intelectuales bien conservados	Vida en casa, aficiones e intereses intelectuales algo limitados	Dificultad leve pero clara de su actividad doméstica; abandono de las tareas más difíciles; abandono de las aficiones e intereses más complicados	Sólo realiza tareas simples; intereses muy restringidos y mal mantenidos	Sin función significativa en casa
Cuidado personal	Completamente capaz de cuidarse por sí mismo/a		Necesita recordatorios	Requiere ayuda para vestirse, asearse y encargarse de sus efectos personales	Requiere mucha ayuda para su cuidado personal; incontinencia frecuente

Tabla 65 Escala Clínica de Demencia (CDR) adaptada al idioma español (Mapi Research Institute, 2006).

7 Prueba cognitiva: GDS

Escala de Deterioro Global adaptada al idioma español (Ruiz, Pérez, & Medina, 2013).

1. Ningún déficit cognitivo: NORMAL
No presenta queja subjetiva de memoria. No se encuentra déficit evidente en la evaluación.
2. Muy ligero déficit cognitivo: NORMAL
Queja subjetiva de déficit amnésico. (Ejemplo: olvida donde deja las cosas, nombres y reconocimiento de rostros). No presenta quejas en el campo laboral o social. En la evaluación no hay evidencia de déficit a nivel de memoria.
3. Leve déficit cognitivo: DEMENCIA LEVE
Clara señal de déficit temprano. Se manifiesta en una de las siguientes áreas: a) El paciente puede perderse a pesar de encontrarse en un lugar conocido. b) Los compañeros de trabajo refieren «pobreza» en las ejecuciones. c) Es evidente la dificultad en las palabras y en la nominación. d) El paciente lee párrafos pero solo logra retener una pequeña cantidad. e) Puede presentar dificultad para recordar nombres y rostros nuevos. f) Se puede perder o puede extraviar objetos de valor. g) Déficit de concentración evidente en la evaluación clínica. Hay evidencia objetiva de déficit de memoria (bien delineados en la evaluación clínica y entrevista detallada), disminución en las actividades laborales y sociales o negociación (que se manifiesta por ansiedad del paciente).
4. Leve o moderado déficit cognitivo: DEMENCIA LEVE
Clara señal de déficit en la intervención clínica. Déficit manifestado en las siguientes áreas: a) Disminución del conocimiento para eventos recientes. b) Déficit en la memoria relacionada con eventos de su historia personal. c) Déficit en concentración (detectada en la sustracción serial). d) Disminución de la habilidad para viajar y manejar finanzas. Con frecuencia se observa déficit en las siguientes áreas: a) Orientación en tiempo y persona. b) Reconocimiento de personas y caras familiares. c) Habilidad para conducirse en sitios conocidos. d) Habilidad para realizar tareas complejas. El mecanismo principal de defensa es la negación y el afecto puede estar aplanado.
5. Moderado déficit cognitivo: DEMENCIA MODERADA
El paciente no puede sobrevivir sin asistencia y es incapaz de recordar aspectos relevantes de su vida actual. Ejemplo: dirección, teléfono. Edad, nombre de miembros de su familia, etc. Es frecuente encontrarlo desorientado en tiempo o lugar. El paciente con buen nivel educativo es incapaz de hacer secuencias en regresión. La persona en este estado, tiene mejor conocimiento de los hechos relacionados en consigo mismo. No presenta mayor dificultad para comer, pero si para vestirse.

6. Moderado a severo déficit cognitivo: DEMENCIA MODERADA

Ocasionalmente olvida el nombre de su esposa, de quien es enteramente dependiente para sobrevivir. Es inconsciente de los eventos recientes y experiencias en su vida. Retiene algún conocimiento de su vida pasada, pero es solo un bosquejo. Requiere de asistencia para las actividades de la vida diaria. En ocasiones presenta incontinencia y requiere de ayuda para conducirse, aunque puede hacerlo en lugares conocidos. Su ritmo de vida diaria se ve afectado.

Se presentan cambios emocionales y de personalidad, que pueden ser rápidos y variables presentando:

- a) Comportamiento delirante. Ejemplo: puede acusar de impostora a su esposa, o hablar con personas inexistentes, o con su imagen en el espejo.
- b) Síntomas obsesivos.
- c) Síntomas de ansiedad, agitación y en ocasiones, comportamientos violentos
- d) Abulia cognitiva (pérdida de la voluntad para llevar a cabo el pensamiento y tener un propósito de acción).

7. Profundo o muy severo déficit cognitivo: DEMENCIA SEVERA

Hay pérdida de todas las habilidades verbales. Frecuentemente esto se convierte en gruñidos o simplemente no habla. Presenta incontinencia urinaria y requiere de asistencia permanente. Hay pérdida de las habilidades motoras básicas. El cerebro parece incapaz de decir que ocurre con el cuerpo.

Tabla 66 Escala de Deterioro Global (GDS) adaptada al idioma español (Ruiz, Pérez, & Medina, 2013).

8 Plantilla N° 1 para la importación de variables del sistema

Plantilla que describe la estructura del archivo CSV en relación con las variables de la base de datos de la ADNI. Se incluyen únicamente las variables que sobrepasan el umbral de importancia referido en la sección «4.1.14.5 Importancia de las variables en la base de datos de la ADNI» y el diagnóstico de la persona.

CDR	MMSE	Age	Education	Diagnosis
{0, 0.5, 1, 2, 3}	{0, 1, 2, ..., 30}	$\{x \in \mathbb{R} \mid x \geq 50\}$	$\{x \in \mathbb{N}_0\}$	{Sane, Mild, Serious}

Tabla 67 Plantilla N°1 en la cual se detallan las columnas y los tipos de datos que debe contener el archivo CSV para poder ser leído correctamente por el sistema. Los nombres de las columnas deberán ser iguales a los encabezados de la plantilla. Las filas de registros deberán cumplir con los tipos de datos y los rangos establecidos para cada variable.

9 Plantilla N° 2 para la importación de variables del sistema

Plantilla que describe la estructura del archivo CSV en relación con las variables de la base de datos de la *ADNI*. Se incluyen únicamente las variables que sobrepasan el umbral de importancia referido en la sección «4.1.14.6 Importancia de las variables en la base de datos de Caldas» y el diagnóstico de la persona.

CDR	MMSE	MoCA	GDS	Diagnosis
{0, 0.5, 1, 2, 3}	{0, 1, 2, ..., 30}	{0, 1, 2, ..., 30}	{1, 2, 3, ..., 7}	{Sane, Mild, Serious}

Tabla 68 Plantilla N°2 en la cual se detallan las columnas y los tipos de datos que debe contener el archivo CSV para poder ser leído correctamente por el sistema. Los nombres de las columnas deberán ser iguales a los encabezados de la plantilla. Las filas de registros deberán cumplir con los tipos de datos y los rangos establecidos para cada variable.

10 Plan de pruebas para el subsistema computacional tipo A

10.1 Resumen

El presente plan contempla los casos de prueba para los requerimientos del subsistema computacional tipo A. Estos se distribuyen en tres etapas: prototipo base (iteración N° 0), iteración N° 1 e iteración N° 2. Los casos de prueba pertenecientes a las primeras iteraciones se vuelven a ejecutar en cada iteración posterior; esto permite verificar que no se hayan introducido nuevos errores en las funcionalidades probadas con anterioridad.

10.2 Requerimientos

Identificador	Nombre	Iteración N°
REQ-A01	Importación de variables	0, 1, 2
REQ-A02	Entrenamiento de árbol de decisión	0, 1, 2
REQ-A03	Asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión	0, 1, 2

REQ-A04	Persistencia de árbol de decisión	1, 2
REQ-A05	Cargue de árbol de decisión	1, 2
REQ-A06	Desempeño de árbol de decisión	2

Tabla 69 Requerimientos incluidos en el plan de pruebas para el subsistema computacional tipo A.

10.3 Calendario de pruebas

Fecha	Actividad	Iteración N°
28/08/2016	Ejecución del banco de pruebas BDP-REQ-A01	0
28/08/2016	Ejecución del banco de pruebas BDP-REQ-A02	0
28/08/2016	Ejecución del banco de pruebas BDP-REQ-A03	0
30/08/2016	Ejecución del banco de pruebas BDP-REQ-A01	1
30/08/2016	Ejecución del banco de pruebas BDP-REQ-A02	1
30/08/2016	Ejecución del banco de pruebas BDP-REQ-A03	1
30/08/2016	Ejecución del banco de pruebas BDP-REQ-A04	1
30/08/2016	Ejecución del banco de pruebas BDP-REQ-A05	1
31/08/2016	Ejecución del banco de pruebas BDP-REQ-A01	2
31/08/2016	Ejecución del banco de pruebas BDP-REQ-A02	2
31/08/2016	Ejecución del banco de pruebas BDP-REQ-A03	2
31/08/2016	Ejecución del banco de pruebas BDP-REQ-A04	2
31/08/2016	Ejecución del banco de pruebas BDP-REQ-A05	2
31/08/2016	Ejecución del banco de pruebas BDP-REQ-A06	2

Tabla 70 Calendario de pruebas para el subsistema computacional tipo A.

10.4 Bancos de pruebas

ID Banco	Descripción	Iteración N°
BDP-REQ-A01	Agrupar todos los casos de prueba para verificar el requerimiento REQ-A01	0, 1, 2



BDP-REQ-A02	Agrupar todos los casos de prueba para verificar el requerimiento REQ-A02	0, 1, 2
BDP-REQ-A03	Agrupar todos los casos de prueba para verificar el requerimiento REQ-A03	0, 1, 2
BDP-REQ-A04	Agrupar todos los casos de prueba para verificar el requerimiento REQ-A04	1, 2
BDP-REQ-A05	Agrupar todos los casos de prueba para verificar el requerimiento REQ-A05	1, 2
BDP-REQ-A06	Agrupar todos los casos de prueba para verificar el requerimiento REQ-A06	2

Tabla 71 Bancos de pruebas para verificar los requerimientos del subsistema computacional tipo A.

10.5 Casos de prueba

10.5.1 Iteración N° 0 (prototipo base)

ID Banco de pruebas	BDP-REQ-A01
ID Caso de prueba	CP-A01-01
Descripción	Importar un archivo que cumple con la plantilla descrita en el anexo 9
Valores de entrada	- Base de datos: <i>Caldas</i> - Archivo: <i>Caldas_Small.csv</i> el cual cumple con la plantilla descrita en el anexo 9
Pasos	- Seleccionar la base de datos <i>Caldas</i> - Seleccionar el archivo <i>Caldas_Small.csv</i> - Llamar al método de chequeo de archivos con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: <i>True</i>

Resultado actual	- Retorno: <i>True</i>
-------------------------	------------------------

Tabla 72 Ejecución del caso de prueba N° 1 para verificar el requerimiento REQ-A01 —importación de variables—.

ID Banco de pruebas	BDP-REQ-A01
ID Caso de prueba	CP-A01-02
Descripción	Importar un archivo que cumple con la plantilla descrita en el anexo 8
Valores de entrada	- Base de datos: <i>ADNI</i> - Archivo: <i>ADNI_Small.csv</i> el cual cumple con la plantilla descrita en el anexo 8
Pasos	- Seleccionar la base de datos <i>ADNI</i> - Seleccionar el archivo <i>ADNI_Small.csv</i> - Llamar al método de chequeo de archivos con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: <i>True</i>
Resultado actual	- Retorno: <i>True</i>

Tabla 73 Ejecución del caso de prueba N° 2 para verificar el requerimiento REQ-A01 —importación de variables—.

ID Banco de pruebas	BDP-REQ-A01
ID Caso de prueba	CP-A01-03
Descripción	Importar un archivo que no cumple con la plantilla descrita en el anexo 9
Valores de entrada	- Base de datos: <i>Caldas</i> - Archivo: <i>Caldas_Wrong.csv</i> el cual no cumple con la plantilla descrita en el anexo 9
Pasos	- Seleccionar la base de datos <i>Caldas</i> - Seleccionar el archivo <i>Caldas_Wrong.csv</i>

	- Llamar al método de chequeo de archivos con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: <i>False</i>
Resultado actual	- Retorno: <i>False</i>

Tabla 74 Ejecución del caso de prueba N° 3 para verificar el requerimiento REQ-A01 —importación de variables—.

ID Banco de pruebas	BDP-REQ-A01
ID Caso de prueba	CP-A01-04
Descripción	Importar un archivo que no cumple con la plantilla descrita en el anexo 8
Valores de entrada	- Base de datos: <i>ADNI</i> - Archivo: <i>ADNI_Wrong.csv</i> el cual cumple con la plantilla descrita en el anexo 8
Pasos	- Seleccionar la base de datos <i>ADNI</i> - Seleccionar el archivo <i>ADNI_Wrong.csv</i> - Llamar al método de chequeo de archivos con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: <i>False</i>
Resultado actual	- Retorno: <i>False</i>

Tabla 75 Ejecución del caso de prueba N° 4 para verificar el requerimiento REQ-A01 —importación de variables—.

ID Banco de pruebas	BDP-REQ-A02
ID Caso de prueba	CP-A02-01
Descripción	Entrenar un árbol de decisión con características y etiquetas relativas a la base de datos de <i>Caldas</i>
Valores de entrada	- Base de datos: <i>Caldas</i>

	- Características: $\begin{bmatrix} [0, 30, 30, 1] \\ [0.5, 27, 25, 2] \\ [1, 26, 24, 3] \\ [1, 19, 22, 3] \\ [2, 18, 21, 4] \\ [3, 0, 0, 7] \end{bmatrix}$ - Etiquetas: $\begin{bmatrix} Sane \\ Sane \\ Mild \\ Mild \\ Serious \\ Serious \end{bmatrix}$
Pasos	- Ingresar al modo de entrenamiento del sistema - Seleccionar la base de datos <i>Caldas</i> - Seleccionar las características y etiquetas para el entrenamiento - Llamar al método de entrenamiento de árbol de decisión con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: <i>True</i>
Resultado actual	- Retorno: <i>True</i>

Tabla 76 Ejecución del caso de prueba N° 5 para verificar el requerimiento REQ-A02 —entrenamiento de árbol de decisión—.

ID Banco de pruebas	BDP-REQ-A02
ID Caso de prueba	CP-A02-02
Descripción	Entrenar un árbol de decisión con características y etiquetas relativas a la base de datos de la <i>ADNI</i>
Valores de entrada	- Base de datos: <i>ADNI</i> - Características: $\begin{bmatrix} [0, 30, 60, 20] \\ [0, 24, 90, 15] \\ [0.5, 28, 60, 14] \\ [0.5, 27, 90, 10] \\ [1, 23, 60, 10] \\ [3, 0, 90, 0] \end{bmatrix}$

	- Etiquetas: $\begin{bmatrix} Sane \\ Sane \\ Mild \\ Mild \\ Serious \\ Serious \end{bmatrix}$
Pasos	- Ingresar al modo de entrenamiento del sistema - Seleccionar la base de datos <i>ADNI</i> - Seleccionar las características y etiquetas para el entrenamiento - Llamar al método de entrenamiento de árbol de decisión con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: <i>True</i>
Resultado actual	- Retorno: <i>True</i>

Tabla 77 Ejecución del caso de prueba N° 6 para verificar el requerimiento REQ-A02 —entrenamiento de árbol de decisión—.

ID Banco de pruebas	BDP-REQ-A03
ID Caso de prueba	CP-A03-01
Descripción	Asistir al diagnóstico de Deterioro Cognitivo Leve mediante un árbol de decisión con características relativas a una persona sana en la base de datos de <i>Caldas</i>
Valores de entrada	- Base de datos: <i>Caldas</i> - Características: $[0, 30, 30, 1]$
Pasos	- Ingresar al modo de asistencia al diagnóstico - Seleccionar la base de datos <i>Caldas</i> - Seleccionar las características para la asistencia - Llamar al método de asistencia con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: $[Sane] [0.1.0.]$
Resultado actual	- Retorno: $[Sane] [0.1.0.]$

Tabla 78 Ejecución del caso de prueba N° 7 para verificar el requerimiento REQ-A03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión—.

ID Banco de pruebas	BDP-REQ-A03
ID Caso de prueba	CP-A03-02
Descripción	Asistir al diagnóstico de Deterioro Cognitivo Leve mediante un árbol de decisión con características relativas a una persona con Deterioro Cognitivo Leve en la base de datos de <i>Caldas</i>
Valores de entrada	- Base de datos: <i>Caldas</i> - Características: $[1, 19, 22, 3]$
Pasos	- Ingresar al modo de asistencia al diagnóstico - Seleccionar la base de datos <i>Caldas</i> - Seleccionar las características para la asistencia - Llamar al método de asistencia con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: <i>[Mild]</i> [1. 0. 0.]
Resultado actual	- Retorno: <i>[Mild]</i> [1. 0. 0.]

Tabla 79 Ejecución del caso de prueba N° 8 para verificar el requerimiento REQ-A03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión—.

ID Banco de pruebas	BDP-REQ-A03
ID Caso de prueba	CP-A03-03
Descripción	Asistir al diagnóstico de Deterioro Cognitivo Leve mediante un árbol de decisión con características relativas a una persona con deterioro cognitivo grave en la base de datos de <i>Caldas</i>
Valores de entrada	- Base de datos: <i>Caldas</i> - Características: $[3, 0, 0, 7]$
Pasos	- Ingresar al modo de asistencia al diagnóstico - Seleccionar la base de datos <i>Caldas</i> - Seleccionar las características para la asistencia

	- Llamar al método de asistencia con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: [<i>Serious</i>] [0. 0. 1.]
Resultado actual	- Retorno: [<i>Serious</i>] [0. 0. 1.]

Tabla 80 Ejecución del caso de prueba N° 9 para verificar el requerimiento REQ-A03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión—.

ID Banco de pruebas	BDP-REQ-A03
ID Caso de prueba	CP-A03-04
Descripción	Asistir al diagnóstico de Deterioro Cognitivo Leve mediante un árbol de decisión con características relativas a una persona sana en la base de datos de la <i>ADNI</i>
Valores de entrada	- Base de datos: <i>ADNI</i> - Características: $[0, 30, 60, 20]$
Pasos	- Ingresar al modo de asistencia al diagnóstico - Seleccionar la base de datos <i>ADNI</i> - Seleccionar las características para la asistencia - Llamar al método de asistencia con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: [<i>Sane</i>] [0. 1. 0.]
Resultado actual	- Retorno: [<i>Sane</i>] [0. 1. 0.]

Tabla 81 Ejecución del caso de prueba N° 10 para verificar el requerimiento REQ-A03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión—.

ID Banco de pruebas	BDP-REQ-A03
ID Caso de prueba	CP-A03-05
Descripción	Asistir al diagnóstico de Deterioro Cognitivo Leve mediante un árbol de decisión con características relativas a una persona con Deterioro Cognitivo Leve en la base de datos de la <i>ADNI</i>
Valores de entrada	- Base de datos: <i>ADNI</i>

	- Características: $[0.5, 27, 90, 10]$
Pasos	- Ingresar al modo de asistencia al diagnóstico - Seleccionar la base de datos <i>ADNI</i> - Seleccionar las características para la asistencia - Llamar al método de asistencia con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: <i>[Mild]</i> [1. 0. 0.]
Resultado actual	- Retorno: <i>[Mild]</i> [1. 0. 0.]

Tabla 82 Ejecución del caso de prueba N° 11 para verificar el requerimiento REQ-A03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión—.

ID Banco de pruebas	BDP-REQ-A03
ID Caso de prueba	CP-A03-06
Descripción	Asistir al diagnóstico de Deterioro Cognitivo Leve mediante un árbol de decisión con características relativas a una persona con deterioro cognitivo grave en la base de datos de la <i>ADNI</i>
Valores de entrada	- Base de datos: <i>ADNI</i> - Características: $[3, 0, 90, 0]$
Pasos	- Ingresar al modo de asistencia al diagnóstico - Seleccionar la base de datos <i>ADNI</i> - Seleccionar las características para la asistencia - Llamar al método de asistencia con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: <i>[Serious]</i> [0. 0. 1.]
Resultado actual	- Retorno: <i>[Serious]</i> [0. 0. 1.]

Tabla 83 Ejecución del caso de prueba N° 12 para verificar el requerimiento REQ-A03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión—.

10.5.2 Iteración N° 1

ID Banco de pruebas	BDP-REQ-A01
ID Caso de prueba	CP-A01-01
Descripción	Re-ejecución del caso de prueba CP-A01-01 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: <i>True</i>
Resultado actual	- Retorno: <i>True</i>

Tabla 84 Re-ejecución del caso de prueba N° 1 para verificar el requerimiento REQ-A01 —importación de variables—.

ID Banco de pruebas	BDP-REQ-A01
ID Caso de prueba	CP-A01-02
Descripción	Re-ejecución del caso de prueba CP-A01-02 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: <i>True</i>
Resultado actual	- Retorno: <i>True</i>

Tabla 85 Re-ejecución del caso de prueba N° 2 para verificar el requerimiento REQ-A01 —importación de variables—.

ID Banco de pruebas	BDP-REQ-A01
ID Caso de prueba	CP-A01-03
Descripción	Re-ejecución del caso de prueba CP-A01-03 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: <i>False</i>
Resultado actual	- Retorno: <i>False</i>

Tabla 86 Re-ejecución del caso de prueba N° 3 para verificar el requerimiento REQ-A01 —importación de variables—.

ID Banco de pruebas	BDP-REQ-A01
ID Caso de prueba	CP-A01-04
Descripción	Re-ejecución del caso de prueba CP-A01-04 para verificar la no

	inclusión de nuevos errores
Resultado esperado	- Retorno: <i>False</i>
Resultado actual	- Retorno: <i>False</i>

Tabla 87 Re-ejecución del caso de prueba N° 4 para verificar el requerimiento REQ-A01 —importación de variables—.

ID Banco de pruebas	BDP-REQ-A02
ID Caso de prueba	CP-A02-01
Descripción	Re-ejecución del caso de prueba CP-A02-01 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: <i>True</i>
Resultado actual	- Retorno: <i>True</i>

Tabla 88 Re-ejecución del caso de prueba N° 5 para verificar el requerimiento REQ-A02 —entrenamiento de árbol de decisión—.

ID Banco de pruebas	BDP-REQ-A02
ID Caso de prueba	CP-A02-02
Descripción	Re-ejecución del caso de prueba CP-A02-02 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: <i>True</i>
Resultado actual	- Retorno: <i>True</i>

Tabla 89 Re-ejecución del caso de prueba N° 6 para verificar el requerimiento REQ-A02 —entrenamiento de árbol de decisión—.

ID Banco de pruebas	BDP-REQ-A03
ID Caso de prueba	CP-A03-01
Descripción	Re-ejecución del caso de prueba CP-A03-01 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: [<i>Sane</i>] [0.1.0.]
Resultado actual	- Retorno: [<i>Sane</i>] [0.1.0.]

Tabla 90 Re-ejecución del caso de prueba N° 7 para verificar el requerimiento REQ-A03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión—.

ID Banco de pruebas	BDP-REQ-A03
ID Caso de prueba	CP-A03-02
Descripción	Re-ejecución del caso de prueba CP-A03-02 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: [<i>Mild</i>] [1. 0. 0.]
Resultado actual	- Retorno: [<i>Mild</i>] [1. 0. 0.]

Tabla 91 Re-ejecución del caso de prueba N° 8 para verificar el requerimiento REQ-A03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión—.

ID Banco de pruebas	BDP-REQ-A03
ID Caso de prueba	CP-A03-03
Descripción	Re-ejecución del caso de prueba CP-A03-03 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: [<i>Serious</i>] [0. 0. 1.]
Resultado actual	- Retorno: [<i>Serious</i>] [0. 0. 1.]

Tabla 92 Re-ejecución del caso de prueba N° 9 para verificar el requerimiento REQ-A03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión—.

ID Banco de pruebas	BDP-REQ-A03
ID Caso de prueba	CP-A03-04
Descripción	Re-ejecución del caso de prueba CP-A03-04 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: [<i>Sane</i>] [0. 1. 0.]
Resultado actual	- Retorno: [<i>Sane</i>] [0. 1. 0.]

Tabla 93 Re-ejecución del caso de prueba N° 10 para verificar el requerimiento REQ-A03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión—.

ID Banco de pruebas	BDP-REQ-A03
ID Caso de prueba	CP-A03-05
Descripción	Re-ejecución del caso de prueba CP-A03-05 para verificar la no inclusión de nuevos errores

Resultado esperado	- Retorno: [<i>Mild</i>] [1. 0. 0.]
Resultado actual	- Retorno: [<i>Mild</i>] [1. 0. 0.]

Tabla 94 Re-ejecución del caso de prueba N° 11 para verificar el requerimiento REQ-A03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión—.

ID Banco de pruebas	BDP-REQ-A03
ID Caso de prueba	CP-A03-06
Descripción	Re-ejecución del caso de prueba CP-A03-06 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: [<i>Serious</i>] [0. 0. 1.]
Resultado actual	- Retorno: [<i>Serious</i>] [0. 0. 1.]

Tabla 95 Re-ejecución del caso de prueba N° 12 para verificar el requerimiento REQ-A03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión—.

ID Banco de pruebas	BDP-REQ-A04
ID Caso de prueba	CP-A04-01
Descripción	Persistir un árbol de decisión entrenado con la base de datos de <i>Caldas</i>
Valores de entrada	- Base de datos: <i>Caldas</i> - Ruta: <i>persistence/Caldas</i> - Árbol de decisión: modelo entrenado con la base de datos de <i>Caldas</i>
Pasos	- Seleccionar la base de datos <i>Caldas</i> - Seleccionar el árbol de decisión entrenado - Llamar al método de guardado de árbol de decisión con las entradas mencionadas
Resultado esperado	- Existencia del archivo <i>tree.pkl</i> en la ruta <i>persistence/Caldas</i> con los datos del árbol de decisión entrenado
Resultado actual	- Existencia del archivo <i>tree.pkl</i> en la ruta <i>persistence/Caldas</i> con los datos del árbol de decisión entrenado

Tabla 96 Ejecución del caso de prueba N° 13 para verificar el requerimiento REQ-A04 —persistencia de árbol de decisión—.

ID Banco de pruebas	BDP-REQ-A04
ID Caso de prueba	CP-A04-02
Descripción	Persistir un árbol de decisión entrenado con la base de datos de la <i>ADNI</i>
Valores de entrada	- Base de datos: <i>ADNI</i> - Ruta: <i>persistence/ADNI</i> - Árbol de decisión: modelo entrenado con la base de datos de <i>ADNI</i>
Pasos	- Seleccionar la base de datos <i>ADNI</i> - Seleccionar el árbol de decisión entrenado - Llamar al método de guardado de árbol de decisión con las entradas mencionadas
Resultado esperado	- Existencia del archivo <i>tree.pkl</i> en la ruta <i>persistence/ADNI</i> con los datos del árbol de decisión entrenado
Resultado actual	- Existencia del archivo <i>tree.pkl</i> en la ruta <i>persistence/ADNI</i> con los datos del árbol de decisión entrenado

Tabla 97 Ejecución del caso de prueba N° 14 para verificar el requerimiento REQ-A04 —persistencia de árbol de decisión—.

ID Banco de pruebas	BDP-REQ-A05
ID Caso de prueba	CP-A05-01
Descripción	Cargar un árbol de decisión entrenado con la base de datos de <i>Caldas</i>
Valores de entrada	- Base de datos: <i>Caldas</i> - Ruta: <i>persistence/Caldas</i>
Pasos	- Seleccionar la base de datos <i>Caldas</i> - Llamar al método de cargue de árbol de decisión con las entradas mencionadas

Resultado esperado	- Árbol de decisión cargado con los datos del archivo <i>persistence/Caldas/tree.pkl</i>
Resultado actual	- Árbol de decisión cargado con los datos del archivo <i>persistence/Caldas/tree.pkl</i>

Tabla 98 Ejecución del caso de prueba N° 15 para verificar el requerimiento REQ-A05 —cargue de árbol de decisión—.

ID Banco de pruebas	BDP-REQ-A05
ID Caso de prueba	CP-A05-02
Descripción	Cargar un árbol de decisión entrenado con la base de datos de la <i>ADNI</i>
Valores de entrada	- Base de datos: <i>ADNI</i> - Ruta: <i>persistence/ADNI</i>
Pasos	- Seleccionar la base de datos <i>ADNI</i> - Llamar al método de cargue de árbol de decisión con las entradas mencionadas
Resultado esperado	- Árbol de decisión cargado con los datos del archivo <i>persistence/ADNI/tree.pkl</i>
Resultado actual	- Árbol de decisión cargado con los datos del archivo <i>persistence/ADNI/tree.pkl</i>

Tabla 99 Ejecución del caso de prueba N° 16 para verificar el requerimiento REQ-A05 —cargue de árbol de decisión—.

10.5.3 Iteración N° 2

ID Banco de pruebas	BDP-REQ-A01
ID Caso de prueba	CP-A01-01
Descripción	Re-ejecución del caso de prueba CP-A01-01 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: <i>True</i>

Resultado actual	- Retorno: <i>True</i>
-------------------------	------------------------

Tabla 100 Re-ejecución del caso de prueba N° 1 para verificar el requerimiento REQ-A01 —importación de variables—.

ID Banco de pruebas	BDP-REQ-A01
ID Caso de prueba	CP-A01-02
Descripción	Re-ejecución del caso de prueba CP-A01-02 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: <i>True</i>
Resultado actual	- Retorno: <i>True</i>

Tabla 101 Re-ejecución del caso de prueba N° 2 para verificar el requerimiento REQ-A01 —importación de variables—.

ID Banco de pruebas	BDP-REQ-A01
ID Caso de prueba	CP-A01-03
Descripción	Re-ejecución del caso de prueba CP-A01-03 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: <i>False</i>
Resultado actual	- Retorno: <i>False</i>

Tabla 102 Re-ejecución del caso de prueba N° 3 para verificar el requerimiento REQ-A01 —importación de variables—.

ID Banco de pruebas	BDP-REQ-A01
ID Caso de prueba	CP-A01-04
Descripción	Re-ejecución del caso de prueba CP-A01-04 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: <i>False</i>
Resultado actual	- Retorno: <i>False</i>

Tabla 103 Re-ejecución del caso de prueba N° 4 para verificar el requerimiento REQ-A01 —importación de variables—.

ID Banco de pruebas	BDP-REQ-A02
----------------------------	-------------

ID Caso de prueba	CP-A02-01
Descripción	Re-ejecución del caso de prueba CP-A02-01 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: <i>True</i>
Resultado actual	- Retorno: <i>True</i>

Tabla 104 Re-ejecución del caso de prueba N° 5 para verificar el requerimiento REQ-A02 —entrenamiento de árbol de decisión—.

ID Banco de pruebas	BDP-REQ-A02
ID Caso de prueba	CP-A02-02
Descripción	Re-ejecución del caso de prueba CP-A02-02 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: <i>True</i>
Resultado actual	- Retorno: <i>True</i>

Tabla 105 Re-ejecución del caso de prueba N° 6 para verificar el requerimiento REQ-A02 —entrenamiento de árbol de decisión—.

ID Banco de pruebas	BDP-REQ-A03
ID Caso de prueba	CP-A03-01
Descripción	Re-ejecución del caso de prueba CP-A03-01 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: [<i>Sane</i>] [0.1.0.]
Resultado actual	- Retorno: [<i>Sane</i>] [0.1.0.]

Tabla 106 Re-ejecución del caso de prueba N° 7 para verificar el requerimiento REQ-A03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión—.

ID Banco de pruebas	BDP-REQ-A03
ID Caso de prueba	CP-A03-02
Descripción	Re-ejecución del caso de prueba CP-A03-02 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: [<i>Mild</i>] [1.0.0.]

Resultado actual	- Retorno: [<i>Mild</i>] [1. 0. 0.]
-------------------------	---------------------------------------

Tabla 107 Re-ejecución del caso de prueba N° 8 para verificar el requerimiento REQ-A03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión—.

ID Banco de pruebas	BDP-REQ-A03
ID Caso de prueba	CP-A03-03
Descripción	Re-ejecución del caso de prueba CP-A03-03 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: [<i>Serious</i>] [0. 0. 1.]
Resultado actual	- Retorno: [<i>Serious</i>] [0. 0. 1.]

Tabla 108 Re-ejecución del caso de prueba N° 9 para verificar el requerimiento REQ-A03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión—.

ID Banco de pruebas	BDP-REQ-A03
ID Caso de prueba	CP-A03-04
Descripción	Re-ejecución del caso de prueba CP-A03-04 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: [<i>Sane</i>] [0. 1. 0.]
Resultado actual	- Retorno: [<i>Sane</i>] [0. 1. 0.]

Tabla 109 Re-ejecución del caso de prueba N° 10 para verificar el requerimiento REQ-A03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión—.

ID Banco de pruebas	BDP-REQ-A03
ID Caso de prueba	CP-A03-05
Descripción	Re-ejecución del caso de prueba CP-A03-05 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: [<i>Mild</i>] [1. 0. 0.]
Resultado actual	- Retorno: [<i>Mild</i>] [1. 0. 0.]

Tabla 110 Re-ejecución del caso de prueba N° 11 para verificar el requerimiento REQ-A03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión—.

ID Banco de pruebas	BDP-REQ-A03
----------------------------	-------------

ID Caso de prueba	CP-A03-06
Descripción	Re-ejecución del caso de prueba CP-A03-06 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: [<i>Serious</i>] [0. 0. 1.]
Resultado actual	- Retorno: [<i>Serious</i>] [0. 0. 1.]

Tabla 111 Re-ejecución del caso de prueba N° 12 para verificar el requerimiento REQ-A03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante árbol de decisión—.

ID Banco de pruebas	BDP-REQ-A04
ID Caso de prueba	CP-A04-01
Descripción	Re-ejecución del caso de prueba CP-A04-01 para verificar la no inclusión de nuevos errores
Resultado esperado	- Existencia del archivo <i>tree.pkl</i> en la ruta <i>persistence/Caldas</i> con los datos del árbol de decisión entrenado
Resultado actual	- Existencia del archivo <i>tree.pkl</i> en la ruta <i>persistence/Caldas</i> con los datos del árbol de decisión entrenado

Tabla 112 Re-ejecución del caso de prueba N° 13 para verificar el requerimiento REQ-A04 —persistencia de árbol de decisión—.

ID Banco de pruebas	BDP-REQ-A04
ID Caso de prueba	CP-A04-02
Descripción	Re-ejecución del caso de prueba CP-A04-02 para verificar la no inclusión de nuevos errores
Resultado esperado	- Existencia del archivo <i>tree.pkl</i> en la ruta <i>persistence/ADNI</i> con los datos del árbol de decisión entrenado
Resultado actual	- Existencia del archivo <i>tree.pkl</i> en la ruta <i>persistence/ADNI</i> con los datos del árbol de decisión entrenado

Tabla 113 Re-ejecución del caso de prueba N° 14 para verificar el requerimiento REQ-A04 —persistencia de árbol de decisión—.

ID Banco de pruebas	BDP-REQ-A05
----------------------------	-------------

ID Caso de prueba	CP-A05-01
Descripción	Re-ejecución del caso de prueba CP-A05-01 para verificar la no inclusión de nuevos errores
Resultado esperado	- Árbol de decisión cargado con los datos del archivo <i>persistence/Caldas/tree.pkl</i>
Resultado actual	- Árbol de decisión cargado con los datos del archivo <i>persistence/Caldas/tree.pkl</i>

Tabla 114 Re-ejecución del caso de prueba N° 15 para verificar el requerimiento REQ-A05 —cargue de árbol de decisión—.

ID Banco de pruebas	BDP-REQ-A05
ID Caso de prueba	CP-A05-02
Descripción	Re-ejecución del caso de prueba CP-A05-02 para verificar la no inclusión de nuevos errores
Resultado esperado	- Árbol de decisión cargado con los datos del archivo <i>persistence/ADNI/tree.pkl</i>
Resultado actual	- Árbol de decisión cargado con los datos del archivo <i>persistence/ADNI/tree.pkl</i>

Tabla 115 Re-ejecución del caso de prueba N° 16 para verificar el requerimiento REQ-A05 —cargue de árbol de decisión—.

ID Banco de pruebas	BDP-REQ-A06
ID Caso de prueba	CP-A06-01
Descripción	Validar el desempeño de un árbol de decisión entrenado con la base de datos de <i>Caldas</i>
Valores de entrada	- Base de datos: <i>Caldas</i> - Árbol de decisión: modelo entrenado con la base de datos de <i>Caldas</i> - Archivo: <i>Caldas_Small.csv</i> el cual cumple con la plantilla descrita en el anexo 9

	- Cantidad de pliegues: 10
Pasos	- Ingresar al modo de validación del sistema - Seleccionar la base de datos <i>Caldas</i> - Seleccionar el árbol de decisión entrenado - Seleccionar el archivo <i>Caldas_Small.csv</i> - Llamar al método de validación de árbol de decisión con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: <i>True True 0.9486 0.9486 0.9486 0.9466 0.9016</i>
Resultado actual	- Retorno: <i>True True 0.9486 0.9486 0.9486 0.9466 0.9016</i>

Tabla 116 Ejecución del caso de prueba N° 17 para verificar el requerimiento REQ-A06 —desempeño de árbol de decisión—.

ID Banco de pruebas	BDP-REQ-A06
ID Caso de prueba	CP-A06-02
Descripción	Validar el desempeño de un árbol de decisión entrenado con la base de datos de la <i>ADNI</i>
Valores de entrada	- Base de datos: <i>ADNI</i> - Árbol de decisión: modelo entrenado con la base de datos de <i>ADNI</i> - Archivo: <i>ADNI_Small.csv</i> el cual cumple con la plantilla descrita en el anexo 8 - Cantidad de pliegues: 10
Pasos	- Ingresar al modo de validación del sistema - Seleccionar la base de datos <i>ADNI</i> - Seleccionar el árbol de decisión entrenado - Seleccionar el archivo <i>ADNI_Small.csv</i> - Llamar al método de validación de árbol de decisión con las entradas mencionadas - Obtener el retorno del método

Resultado esperado	- Retorno: <i>True True 0.8779 0.8774 0.8779 0.9563 0.8061</i>
Resultado actual	- Retorno: <i>True True 0.8779 0.8774 0.8779 0.9563 0.8061</i>

Tabla 117 Ejecución del caso de prueba N° 18 para verificar el requerimiento REQ-A06 —desempeño de árbol de decisión—.

11 Plan de pruebas para el subsistema computacional tipo B

11.1 Resumen

El presente plan contempla los casos de prueba para los requerimientos del subsistema computacional tipo B. Estos se distribuyen en tres etapas: prototipo base (iteración N° 0), iteración N° 1 e iteración N° 2. Los casos de prueba pertenecientes a las primeras iteraciones se vuelven a ejecutar en cada iteración posterior; esto permite verificar que no se hayan introducido nuevos errores en las funcionalidades probadas con anterioridad.

11.2 Requerimientos

Identificador	Nombre	Iteración N°
REQ-B01	Importación de variables	0, 1, 2
REQ-B02	Entrenamiento de red neuronal artificial	0, 1, 2
REQ-B03	Asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial	0, 1, 2
REQ-B04	Persistencia de red neuronal artificial	1, 2
REQ-B05	Cargue de red neuronal artificial	1, 2
REQ-B06	Desempeño de red neuronal artificial	2

Tabla 118 Requerimientos incluidos en el plan de pruebas para el subsistema computacional tipo B.

11.3 Calendario de pruebas

Fecha	Actividad	Iteración N°
01/09/2016	Ejecución del banco de pruebas BDP-REQ-B01	0
01/09/2016	Ejecución del banco de pruebas BDP-REQ-B02	0
01/09/2016	Ejecución del banco de pruebas BDP-REQ-B03	0
02/09/2016	Ejecución del banco de pruebas BDP-REQ-B01	1
02/09/2016	Ejecución del banco de pruebas BDP-REQ-B02	1
02/09/2016	Ejecución del banco de pruebas BDP-REQ-B03	1
02/09/2016	Ejecución del banco de pruebas BDP-REQ-B04	1
02/09/2016	Ejecución del banco de pruebas BDP-REQ-B05	1
03/09/2016	Ejecución del banco de pruebas BDP-REQ-B01	2
03/09/2016	Ejecución del banco de pruebas BDP-REQ-B02	2
03/09/2016	Ejecución del banco de pruebas BDP-REQ-B03	2
03/09/2016	Ejecución del banco de pruebas BDP-REQ-B04	2
03/09/2016	Ejecución del banco de pruebas BDP-REQ-B05	2
03/09/2016	Ejecución del banco de pruebas BDP-REQ-B06	2

Tabla 119 Calendario de pruebas para el subsistema computacional tipo B.

11.4 Bancos de pruebas

ID Banco	Descripción	Iteración N°
BDP-REQ-B01	Agrupar todos los casos de prueba para verificar el requerimiento REQ-B01	0, 1, 2
BDP-REQ-B02	Agrupar todos los casos de prueba para verificar el requerimiento REQ-B02	0, 1, 2
BDP-REQ-B03	Agrupar todos los casos de prueba para verificar el requerimiento REQ-B03	0, 1, 2
BDP-REQ-B04	Agrupar todos los casos de prueba para verificar el requerimiento REQ-B04	1, 2

BDP-REQ-B05	Agrupar todos los casos de prueba para verificar el requerimiento REQ-B05	1, 2
BDP-REQ-B06	Agrupar todos los casos de prueba para verificar el requerimiento REQ-B06	2

Tabla 120 Bancos de pruebas para verificar los requerimientos del subsistema computacional tipo B.

11.5 Casos de prueba

11.5.1 Iteración N° 0 (prototipo base)

ID Banco de pruebas	BDP-REQ-B01
ID Caso de prueba	CP-B01-01
Descripción	Importar un archivo que cumple con la plantilla descrita en el anexo 9
Valores de entrada	- Base de datos: <i>Caldas</i> - Archivo: <i>Caldas_Small.csv</i> el cual cumple con la plantilla descrita en el anexo 9
Pasos	- Seleccionar la base de datos <i>Caldas</i> - Seleccionar el archivo <i>Caldas_Small.csv</i> - Llamar al método de chequeo de archivos con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: <i>True</i>
Resultado actual	- Retorno: <i>True</i>

Tabla 121 Ejecución del caso de prueba N° 1 para verificar el requerimiento REQ-B01 —importación de variables—.

ID Banco de pruebas	BDP-REQ-B01
ID Caso de prueba	CP-B01-02
Descripción	Importar un archivo que cumple con la plantilla descrita en el anexo 8

Valores de entrada	- Base de datos: <i>ADNI</i> - Archivo: <i>ADNI_Small.csv</i> el cual cumple con la plantilla descrita en el anexo 8
Pasos	- Seleccionar la base de datos <i>ADNI</i> - Seleccionar el archivo <i>ADNI_Small.csv</i> - Llamar al método de chequeo de archivos con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: <i>True</i>
Resultado actual	- Retorno: <i>True</i>

Tabla 122 Ejecución del caso de prueba N° 2 para verificar el requerimiento REQ-B01 —importación de variables—.

ID Banco de pruebas	BDP-REQ-B01
ID Caso de prueba	CP-B01-03
Descripción	Importar un archivo que no cumple con la plantilla descrita en el anexo 9
Valores de entrada	- Base de datos: <i>Caldas</i> - Archivo: <i>Caldas_Wrong.csv</i> el cual no cumple con la plantilla descrita en el anexo 9
Pasos	- Seleccionar la base de datos <i>Caldas</i> - Seleccionar el archivo <i>Caldas_Wrong.csv</i> - Llamar al método de chequeo de archivos con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: <i>False</i>
Resultado actual	- Retorno: <i>False</i>

Tabla 123 Ejecución del caso de prueba N° 3 para verificar el requerimiento REQ-B01 —importación de variables—.

ID Banco de pruebas	BDP-REQ-B01
----------------------------	-------------

ID Caso de prueba	CP-B01-04
Descripción	Importar un archivo que no cumple con la plantilla descrita en el anexo 8
Valores de entrada	- Base de datos: <i>ADNI</i> - Archivo: <i>ADNI_Wrong.csv</i> el cual cumple con la plantilla descrita en el anexo 8
Pasos	- Seleccionar la base de datos <i>ADNI</i> - Seleccionar el archivo <i>ADNI_Wrong.csv</i> - Llamar al método de chequeo de archivos con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: <i>False</i>
Resultado actual	- Retorno: <i>False</i>

Tabla 124 Ejecución del caso de prueba N° 4 para verificar el requerimiento REQ-B01 —importación de variables—.

ID Banco de pruebas	BDP-REQ-B02
ID Caso de prueba	CP-B02-01
Descripción	Entrenar una red neuronal artificial con características y etiquetas relativas a la base de datos de <i>Caldas</i>
Valores de entrada	- Base de datos: <i>Caldas</i> - Características: $\begin{bmatrix} [0, 30, 30, 1] \\ [0.5, 27, 25, 2] \\ [1, 26, 24, 3] \\ [1, 19, 22, 3] \\ [2, 18, 21, 4] \\ [3, 0, 0, 7] \end{bmatrix}$ - Etiquetas: $\begin{bmatrix} Sane \\ Sane \\ Mild \\ Mild \\ Serious \\ Serious \end{bmatrix}$

Pasos	- Ingresar al modo de entrenamiento del sistema - Seleccionar la base de datos <i>Caldas</i> - Seleccionar las características y etiquetas para el entrenamiento - Llamar al método de entrenamiento de red neuronal con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: <i>True</i>
Resultado actual	- Retorno: <i>True</i>

Tabla 125 Ejecución del caso de prueba N° 5 para verificar el requerimiento REQ-B02 —entrenamiento de red neuronal artificial—.

ID Banco de pruebas	BDP-REQ-B02
ID Caso de prueba	CP-B02-02
Descripción	Entrenar una red neuronal artificial con características y etiquetas relativas a la base de datos de la <i>ADNI</i>
Valores de entrada	- Base de datos: <i>ADNI</i> - Características: $\begin{bmatrix} [0, 30, 60, 20] \\ [0, 24, 90, 15] \\ [0.5, 28, 60, 14] \\ [0.5, 27, 90, 10] \\ [1, 23, 60, 10] \\ [3, 0, 90, 0] \end{bmatrix}$ - Etiquetas: $\begin{bmatrix} Sane \\ Sane \\ Mild \\ Mild \\ Serious \\ Serious \end{bmatrix}$
Pasos	- Ingresar al modo de entrenamiento del sistema - Seleccionar la base de datos <i>ADNI</i> - Seleccionar las características y etiquetas para el entrenamiento

	- Llamar al método de entrenamiento de red neuronal con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: <i>True</i>
Resultado actual	- Retorno: <i>True</i>

Tabla 126 Ejecución del caso de prueba N° 6 para verificar el requerimiento REQ-B02 —entrenamiento de red neuronal artificial—.

ID Banco de pruebas	BDP-REQ-B03
ID Caso de prueba	CP-B03-01
Descripción	Asistir al diagnóstico de Deterioro Cognitivo Leve mediante una red neuronal artificial con características relativas a una persona sana en la base de datos de <i>Caldas</i>
Valores de entrada	- Base de datos: <i>Caldas</i> - Características: $[0, 30, 30, 1]$
Pasos	- Ingresar al modo de asistencia al diagnóstico - Seleccionar la base de datos <i>Caldas</i> - Seleccionar las características para la asistencia - Llamar al método de asistencia con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: [<i>Sane</i>] [0.0002 0.9997 0.0000]
Resultado actual	- Retorno: [<i>Sane</i>] [0.0002 0.9997 0.0000]

Tabla 127 Ejecución del caso de prueba N° 7 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—.

ID Banco de pruebas	BDP-REQ-B03
ID Caso de prueba	CP-B03-02
Descripción	Asistir al diagnóstico de Deterioro Cognitivo Leve mediante una red neuronal artificial con características relativas a una persona con Deterioro Cognitivo Leve en la base de datos de <i>Caldas</i>

Valores de entrada	- Base de datos: <i>Caldas</i> - Características: $[[1, 19, 22, 3]]$
Pasos	- Ingresar al modo de asistencia al diagnóstico - Seleccionar la base de datos <i>Caldas</i> - Seleccionar las características para la asistencia - Llamar al método de asistencia con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: <i>[Mild]</i> [0.7844 0.0005 0.2149]
Resultado actual	- Retorno: <i>[Mild]</i> [0.7844 0.0005 0.2149]

Tabla 128 Ejecución del caso de prueba N° 8 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—.

ID Banco de pruebas	BDP-REQ-B03
ID Caso de prueba	CP-B03-03
Descripción	Asistir al diagnóstico de Deterioro Cognitivo Leve mediante una red neuronal artificial con características relativas a una persona con deterioro cognitivo grave en la base de datos de <i>Caldas</i>
Valores de entrada	- Base de datos: <i>Caldas</i> - Características: $[[3, 0, 0, 7]]$
Pasos	- Ingresar al modo de asistencia al diagnóstico - Seleccionar la base de datos <i>Caldas</i> - Seleccionar las características para la asistencia - Llamar al método de asistencia con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: <i>[Serious]</i> [0.0017 0.0000 0.9982]
Resultado actual	- Retorno: <i>[Serious]</i> [0.0017 0.0000 0.9982]

Tabla 129 Ejecución del caso de prueba N° 9 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—.

ID Banco de pruebas	BDP-REQ-B03
----------------------------	-------------

ID Caso de prueba	CP-B03-04
Descripción	Asistir al diagnóstico de Deterioro Cognitivo Leve mediante una red neuronal artificial con características relativas a una persona sana en la base de datos de la <i>ADNI</i>
Valores de entrada	- Base de datos: <i>ADNI</i> - Características: $[[0, 30, 60, 20]]$
Pasos	- Ingresar al modo de asistencia al diagnóstico - Seleccionar la base de datos <i>ADNI</i> - Seleccionar las características para la asistencia - Llamar al método de asistencia con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: [<i>Sane</i>] [0.0000 0.9999 0.0000]
Resultado actual	- Retorno: [<i>Sane</i>] [0.0000 0.9999 0.0000]

Tabla 130 Ejecución del caso de prueba N° 10 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—.

ID Banco de pruebas	BDP-REQ-B03
ID Caso de prueba	CP-B0B-05
Descripción	Asistir al diagnóstico de Deterioro Cognitivo Leve mediante una red neuronal artificial con características relativas a una persona con Deterioro Cognitivo Leve en la base de datos de la <i>ADNI</i>
Valores de entrada	- Base de datos: <i>ADNI</i> - Características: $[[0.5, 27, 90, 10]]$
Pasos	- Ingresar al modo de asistencia al diagnóstico - Seleccionar la base de datos <i>ADNI</i> - Seleccionar las características para la asistencia - Llamar al método de asistencia con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: [<i>Mild</i>] [0.8551 0.0000 0.1447]

Resultado actual	- Retorno: <i>[Mild]</i> [0.8551 0.0000 0.1447]
-------------------------	---

Tabla 131 Ejecución del caso de prueba N° 11 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—.

ID Banco de pruebas	BDP-REQ-B03
ID Caso de prueba	CP-B03-06
Descripción	Asistir al diagnóstico de Deterioro Cognitivo Leve mediante una red neuronal artificial con características relativas a una persona con deterioro cognitivo grave en la base de datos de la <i>ADNI</i>
Valores de entrada	- Base de datos: <i>ADNI</i> - Características: $[3, 0, 90, 0]$
Pasos	- Ingresar al modo de asistencia al diagnóstico - Seleccionar la base de datos <i>ADNI</i> - Seleccionar las características para la asistencia - Llamar al método de asistencia con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: <i>[Serious]</i> [0.0360 0.0000 0.9639]
Resultado actual	- Retorno: <i>[Serious]</i> [0.0360 0.0000 0.9639]

Tabla 132 Ejecución del caso de prueba N° 12 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—.

11.5.2 Iteración N° 1

ID Banco de pruebas	BDP-REQ-B01
ID Caso de prueba	CP-B01-01
Descripción	Re-ejecución del caso de prueba CP-B01-01 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: <i>True</i>
Resultado actual	- Retorno: <i>True</i>

Tabla 133 Re-ejecución del caso de prueba N° 1 para verificar el requerimiento REQ-B01 —importación de variables—.

ID Banco de pruebas	BDP-REQ-B01
ID Caso de prueba	CP-B01-02
Descripción	Re-ejecución del caso de prueba CP-B01-02 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: <i>True</i>
Resultado actual	- Retorno: <i>True</i>

Tabla 134 Re-ejecución del caso de prueba N° 2 para verificar el requerimiento REQ-B01 —importación de variables—.

ID Banco de pruebas	BDP-REQ-B01
ID Caso de prueba	CP-B01-03
Descripción	Re-ejecución del caso de prueba CP-B01-03 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: <i>False</i>
Resultado actual	- Retorno: <i>False</i>

Tabla 135 Re-ejecución del caso de prueba N° 3 para verificar el requerimiento REQ-B01 —importación de variables—.

ID Banco de pruebas	BDP-REQ-B01
ID Caso de prueba	CP-B01-04
Descripción	Re-ejecución del caso de prueba CP-B01-04 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: <i>False</i>
Resultado actual	- Retorno: <i>False</i>

Tabla 136 Re-ejecución del caso de prueba N° 4 para verificar el requerimiento REQ-B01 —importación de variables—.

ID Banco de pruebas	BDP-REQ-B02
ID Caso de prueba	CP-B02-01
Descripción	Re-ejecución del caso de prueba CP-B02-01 para verificar la no inclusión de nuevos errores

Resultado esperado	- Retorno: <i>True</i>
Resultado actual	- Retorno: <i>True</i>

Tabla 137 Re-ejecución del caso de prueba N° 5 para verificar el requerimiento REQ-B02 —entrenamiento de red neuronal artificial—.

ID Banco de pruebas	BDP-REQ-B02
ID Caso de prueba	CP-B02-02
Descripción	Re-ejecución del caso de prueba CP-B02-02 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: <i>True</i>
Resultado actual	- Retorno: <i>True</i>

Tabla 138 Re-ejecución del caso de prueba N° 6 para verificar el requerimiento REQ-B02 —entrenamiento de red neuronal artificial—.

ID Banco de pruebas	BDP-REQ-B03
ID Caso de prueba	CP-B03-01
Descripción	Re-ejecución del caso de prueba CP-B03-01 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: [<i>Sane</i>] [0.0002 0.9997 0.0000]
Resultado actual	- Retorno: [<i>Sane</i>] [0.0002 0.9997 0.0000]

Tabla 139 Re-ejecución del caso de prueba N° 7 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—.

ID Banco de pruebas	BDP-REQ-B03
ID Caso de prueba	CP-B03-02
Descripción	Re-ejecución del caso de prueba CP-B03-02 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: [<i>Mild</i>] [0.7844 0.0005 0.2149]
Resultado actual	- Retorno: [<i>Mild</i>] [0.7844 0.0005 0.2149]

Tabla 140 Re-ejecución del caso de prueba N° 8 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—.

ID Banco de pruebas	BDP-REQ-B03
ID Caso de prueba	CP-B03-03
Descripción	Re-ejecución del caso de prueba CP-B03-03 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: [<i>Serious</i>] [0.0017 0.0000 0.9982]
Resultado actual	- Retorno: [<i>Serious</i>] [0.0017 0.0000 0.9982]

Tabla 141 Re-ejecución del caso de prueba N° 9 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—.

ID Banco de pruebas	BDP-REQ-B03
ID Caso de prueba	CP-B03-04
Descripción	Re-ejecución del caso de prueba CP-B03-04 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: [<i>Sane</i>] [0.0000 0.9999 0.0000]
Resultado actual	- Retorno: [<i>Sane</i>] [0.0000 0.9999 0.0000]

Tabla 142 Re-ejecución del caso de prueba N° 10 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—.

ID Banco de pruebas	BDP-REQ-B03
ID Caso de prueba	CP-B03-05
Descripción	Re-ejecución del caso de prueba CP-B03-05 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: [<i>Mild</i>] [0.8551 0.0000 0.1447]
Resultado actual	- Retorno: [<i>Mild</i>] [0.8551 0.0000 0.1447]

Tabla 143 Re-ejecución del caso de prueba N° 11 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—.

ID Banco de pruebas	BDP-REQ-B03
ID Caso de prueba	CP-B03-06
Descripción	Re-ejecución del caso de prueba CP-B03-06 para verificar la no inclusión de nuevos errores

Resultado esperado	- Retorno: [<i>Serious</i>] [0.0360 0.0000 0.9639]
Resultado actual	- Retorno: [<i>Serious</i>] [0.0360 0.0000 0.9639]

Tabla 144 Re-ejecución del caso de prueba N° 12 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—.

ID Banco de pruebas	BDP-REQ-B04
ID Caso de prueba	CP-B04-01
Descripción	Persistir una red neuronal artificial entrenada con la base de datos de <i>Caldas</i>
Valores de entrada	- Base de datos: <i>Caldas</i> - Ruta: <i>persistence/Caldas</i> - Red neuronal artificial: modelo entrenado con la base de datos de <i>Caldas</i>
Pasos	- Seleccionar la base de datos <i>Caldas</i> - Seleccionar la red neuronal artificial entrenada - Llamar al método de guardado de red neuronal artificial con las entradas mencionadas
Resultado esperado	- Existencia del archivo <i>net.h5</i> en la ruta <i>persistence/Caldas</i> con los datos de la red neuronal artificial entrenada
Resultado actual	- Existencia del archivo <i>net.h5</i> en la ruta <i>persistence/Caldas</i> con los datos de la red neuronal artificial entrenada

Tabla 145 Ejecución del caso de prueba N° 13 para verificar el requerimiento REQ-B04 —persistencia de red neuronal artificial—.

ID Banco de pruebas	BDP-REQ-B04
ID Caso de prueba	CP-B04-02
Descripción	Persistir una red neuronal artificial entrenada con la base de datos de la <i>ADNI</i>
Valores de entrada	- Base de datos: <i>ADNI</i> - Ruta: <i>persistence/ADNI</i>

	- Red neuronal artificial: modelo entrenado con la base de datos de <i>ADNI</i>
Pasos	- Seleccionar la base de datos <i>ADNI</i> - Seleccionar la red neuronal artificial entrenada - Llamar al método de guardado de red neuronal artificial con las entradas mencionadas
Resultado esperado	- Existencia del archivo <i>net.h5</i> en la ruta <i>persistence/ADNI</i> con los datos de la red neuronal artificial entrenada
Resultado actual	- Existencia del archivo <i>net.h5</i> en la ruta <i>persistence/ADNI</i> con los datos de la red neuronal artificial entrenada

Tabla 146 Ejecución del caso de prueba N° 14 para verificar el requerimiento REQ-B04 —persistencia de red neuronal artificial—.

ID Banco de pruebas	BDP-REQ-B05
ID Caso de prueba	CP-B05-01
Descripción	Cargar una red neuronal artificial entrenada con la base de datos de <i>Caldas</i>
Valores de entrada	- Base de datos: <i>Caldas</i> - Ruta: <i>persistence/Caldas</i>
Pasos	- Seleccionar la base de datos <i>Caldas</i> - Llamar al método de cargue de red neuronal artificial con las entradas mencionadas
Resultado esperado	- Red neuronal artificial cargada con los datos del archivo <i>persistence/Caldas/net.h5</i>
Resultado actual	- Red neuronal artificial cargada con los datos del archivo <i>persistence/Caldas/net.h5</i>

Tabla 147 Ejecución del caso de prueba N° 15 para verificar el requerimiento REQ-B05 —cargue de red neuronal artificial—.

ID Banco de pruebas	BDP-REQ-B05
----------------------------	-------------

ID Caso de prueba	CP-B05-02
Descripción	Cargar una red neuronal artificial entrenada con la base de datos de la <i>ADNI</i>
Valores de entrada	- Base de datos: <i>ADNI</i> - Ruta: <i>persistence/ADNI</i>
Pasos	- Seleccionar la base de datos <i>ADNI</i> - Llamar al método de cargue de red neuronal artificial con las entradas mencionadas
Resultado esperado	- Red neuronal artificial cargada con los datos del archivo <i>persistence/ADNI/net.h5</i>
Resultado actual	- Red neuronal artificial cargada con los datos del archivo <i>persistence/ADNI/net.h5</i>

Tabla 148 Ejecución del caso de prueba N° 16 para verificar el requerimiento REQ-B05 —cargue de red neuronal artificial—.

11.5.3 Iteración N° 2

ID Banco de pruebas	BDP-REQ-B01
ID Caso de prueba	CP-B01-01
Descripción	Re-ejecución del caso de prueba CP-B01-01 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: <i>True</i>
Resultado actual	- Retorno: <i>True</i>

Tabla 149 Re-ejecución del caso de prueba N° 1 para verificar el requerimiento REQ-B01 —importación de variables—.

ID Banco de pruebas	BDP-REQ-B01
ID Caso de prueba	CP-B01-02
Descripción	Re-ejecución del caso de prueba CP-B01-02 para verificar la no inclusión de nuevos errores

Resultado esperado	- Retorno: <i>True</i>
Resultado actual	- Retorno: <i>True</i>

Tabla 150 Re-ejecución del caso de prueba N° 2 para verificar el requerimiento REQ-B01 —importación de variables—.

ID Banco de pruebas	BDP-REQ-B01
ID Caso de prueba	CP-B01-03
Descripción	Re-ejecución del caso de prueba CP-B01-03 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: <i>False</i>
Resultado actual	- Retorno: <i>False</i>

Tabla 151 Re-ejecución del caso de prueba N° 3 para verificar el requerimiento REQ-B01 —importación de variables—.

ID Banco de pruebas	BDP-REQ-B01
ID Caso de prueba	CP-B01-04
Descripción	Re-ejecución del caso de prueba CP-B01-04 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: <i>False</i>
Resultado actual	- Retorno: <i>False</i>

Tabla 152 Re-ejecución del caso de prueba N° 4 para verificar el requerimiento REQ-B01 —importación de variables—.

ID Banco de pruebas	BDP-REQ-B02
ID Caso de prueba	CP-B02-01
Descripción	Re-ejecución del caso de prueba CP-B02-01 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: <i>True</i>
Resultado actual	- Retorno: <i>True</i>

Tabla 153 Re-ejecución del caso de prueba N° 5 para verificar el requerimiento REQ-B02 —entrenamiento de red neuronal artificial—.

ID Banco de pruebas	BDP-REQ-B02
ID Caso de prueba	CP-B02-02
Descripción	Re-ejecución del caso de prueba CP-B02-02 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: <i>True</i>
Resultado actual	- Retorno: <i>True</i>

Tabla 154 Re-ejecución del caso de prueba N° 6 para verificar el requerimiento REQ-B02 —entrenamiento de red neuronal artificial—.

ID Banco de pruebas	BDP-REQ-B03
ID Caso de prueba	CP-B03-01
Descripción	Re-ejecución del caso de prueba CP-B03-01 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: [<i>Sane</i>] [0.0002 0.9997 0.0000]
Resultado actual	- Retorno: [<i>Sane</i>] [0.0002 0.9997 0.0000]

Tabla 155 Re-ejecución del caso de prueba N° 7 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—.

ID Banco de pruebas	BDP-REQ-B03
ID Caso de prueba	CP-B03-02
Descripción	Re-ejecución del caso de prueba CP-B03-02 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: [<i>Mild</i>] [0.7844 0.0005 0.2149]
Resultado actual	- Retorno: [<i>Mild</i>] [0.7844 0.0005 0.2149]

Tabla 156 Re-ejecución del caso de prueba N° 8 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—.

ID Banco de pruebas	BDP-REQ-B03
ID Caso de prueba	CP-B03-03
Descripción	Re-ejecución del caso de prueba CP-B03-03 para verificar la no inclusión de nuevos errores

Resultado esperado	- Retorno: [Serious] [0.0017 0.0000 0.9982]
Resultado actual	- Retorno: [Serious] [0.0017 0.0000 0.9982]

Tabla 157 Re-ejecución del caso de prueba N° 9 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—.

ID Banco de pruebas	BDP-REQ-B03
ID Caso de prueba	CP-B03-04
Descripción	Re-ejecución del caso de prueba CP-B03-04 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: [Sane] [0.0000 0.9999 0.0000]
Resultado actual	- Retorno: [Sane] [0.0000 0.9999 0.0000]

Tabla 158 Re-ejecución del caso de prueba N° 10 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—.

ID Banco de pruebas	BDP-REQ-B03
ID Caso de prueba	CP-B03-05
Descripción	Re-ejecución del caso de prueba CP-B03-05 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: [Mild] [0.8551 0.0000 0.0.1447]
Resultado actual	- Retorno: [Mild] [0.8551 0.0000 0.0.1447]

Tabla 159 Re-ejecución del caso de prueba N° 11 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—.

ID Banco de pruebas	BDP-REQ-B03
ID Caso de prueba	CP-B03-06
Descripción	Re-ejecución del caso de prueba CP-B03-06 para verificar la no inclusión de nuevos errores
Resultado esperado	- Retorno: [Serious] [0.0360 0.0000 0.9639]
Resultado actual	- Retorno: [Serious] [0.0360 0.0000 0.9639]

Tabla 160 Re-ejecución del caso de prueba N° 12 para verificar el requerimiento REQ-B03 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante red neuronal artificial—.

ID Banco de pruebas	BDP-REQ-B04
ID Caso de prueba	CP-B04-01
Descripción	Re-ejecución del caso de prueba CP-B04-01 para verificar la no inclusión de nuevos errores
Resultado esperado	- Existencia del archivo <i>net.h5</i> en la ruta <i>persistence/Caldas</i> con los datos de la red neuronal artificial entrenada
Resultado actual	- Existencia del archivo <i>net.h5</i> en la ruta <i>persistence/Caldas</i> con los datos de la red neuronal artificial entrenada

Tabla 161 Re-ejecución del caso de prueba N° 13 para verificar el requerimiento REQ-B04 —persistencia de red neuronal artificial—.

ID Banco de pruebas	BDP-REQ-B04
ID Caso de prueba	CP-B04-02
Descripción	Re-ejecución del caso de prueba CP-B04-02 para verificar la no inclusión de nuevos errores
Resultado esperado	- Existencia del archivo <i>net.h5</i> en la ruta <i>persistence/ADNI</i> con los datos de la red neuronal artificial entrenada
Resultado actual	- Existencia del archivo <i>net.h5</i> en la ruta <i>persistence/ADNI</i> con los datos de la red neuronal artificial entrenada

Tabla 162 Re-ejecución del caso de prueba N° 14 para verificar el requerimiento REQ-B04 —persistencia de red neuronal artificial—.

ID Banco de pruebas	BDP-REQ-B05
ID Caso de prueba	CP-B05-01
Descripción	Re-ejecución del caso de prueba CP-B05-01 para verificar la no inclusión de nuevos errores
Resultado esperado	- Red neuronal artificial cargada con los datos del archivo <i>persistence/Caldas/net.h5</i>
Resultado actual	- Red neuronal artificial cargada con los datos del archivo <i>persistence/Caldas/net.h5</i>

Tabla 163 Re-ejecución del caso de prueba N° 15 para verificar el requerimiento REQ-B05 —cargue de red neuronal artificial—.

ID Banco de pruebas	BDP-REQ-B05
ID Caso de prueba	CP-B05-02
Descripción	Re-ejecución del caso de prueba CP-B05-02 para verificar la no inclusión de nuevos errores
Resultado esperado	- Red neuronal artificial cargada con los datos del archivo <i>persistence/ADNI/net.h5</i>
Resultado actual	- Red neuronal artificial cargada con los datos del archivo <i>persistence/ADNI/net.h5</i>

Tabla 164 Re-ejecución del caso de prueba N° 16 para verificar el requerimiento REQ-B05 —cargue de red neuronal artificial—.

ID Banco de pruebas	BDP-REQ-B06
ID Caso de prueba	CP-B06-01
Descripción	Validar el desempeño de una red neuronal artificial entrenada con la base de datos de <i>Caldas</i>
Valores de entrada	- Base de datos: <i>Caldas</i> - Red neuronal artificial: modelo entrenado con la base de datos de <i>Caldas</i> - Archivo: <i>Caldas_Small.csv</i> el cual cumple con la plantilla descrita en el anexo 9 - Cantidad de pliegues: 10
Pasos	- Ingresar al modo de validación del sistema - Seleccionar la base de datos <i>Caldas</i> - Seleccionar la red neuronal artificial entrenada - Seleccionar el archivo <i>Caldas_Small.csv</i> - Llamar al método de validación de red neuronal artificial con las entradas mencionadas - Obtener el retorno del método

Resultado esperado	- Retorno: <i>True True 0.9675 0.9691 0.9675 0.9793 0.9385</i>
Resultado actual	- Retorno: <i>True True 0.9675 0.9691 0.9675 0.9793 0.9385</i>

Tabla 165 Ejecución del caso de prueba N° 17 para verificar el requerimiento REQ-B06 —desempeño de red neuronal artificial—.

ID Banco de pruebas	BDP-REQ-B06
ID Caso de prueba	CP-B06-02
Descripción	Validar el desempeño de una red neuronal artificial con la base de datos de la <i>ADNI</i>
Valores de entrada	- Base de datos: <i>ADNI</i> - Red neuronal artificial: modelo entrenado con la base de datos de <i>ADNI</i> - Archivo: <i>ADNI_Small.csv</i> el cual cumple con la plantilla descrita en el anexo 8 - Cantidad de pliegues: 10
Pasos	- Ingresar al modo de validación del sistema - Seleccionar la base de datos <i>ADNI</i> - Seleccionar la red neuronal artificial entrenada - Seleccionar el archivo <i>ADNI_Small.csv</i> - Llamar al método de validación de red neuronal artificial con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: <i>True True 0.8962 0.8961 0.8962 0.9620 0.8349</i>
Resultado actual	- Retorno: <i>True True 0.8962 0.8961 0.8962 0.9620 0.8349</i>

Tabla 166 Ejecución del caso de prueba N° 18 para verificar el requerimiento REQ-B06 —desempeño de red neuronal artificial—.

12 Plan de pruebas para el sistema computacional híbrido

12.1 Resumen

El presente plan contempla los casos de prueba para la integración de los subsistemas computacionales de tipo A y B en el sistema computacional híbrido. Estos se ejecutan en una sola iteración teniendo en cuenta que ya se han probado con anterioridad los subsistemas computacionales que hacen parte del esquema de hibridación.

12.2 Requerimientos

Identificador	Nombre	Iteración N°
REQ-C01	Entrenamiento de ensamble basado en Bagging	1
REQ-C02	Asistencia al diagnóstico de Deterioro Cognitivo Leve mediante ensamble basado en Bagging	1
REQ-C03	Persistencia de ensamble basado en Bagging	1
REQ-C04	Cargue de ensamble basado en Bagging	1
REQ-C05	Desempeño de ensamble basado en Bagging	1

Tabla 167 Requerimientos incluidos en el plan de pruebas para la integración de los subsistemas computacionales de tipo A y B en el sistema computacional híbrido.

12.3 Calendario de pruebas

Fecha	Actividad	Iteración N°
02/09/2016	Ejecución del banco de pruebas BDP-REQ-C01	1
02/09/2016	Ejecución del banco de pruebas BDP-REQ-C02	1
02/09/2016	Ejecución del banco de pruebas BDP-REQ-C03	1
02/09/2016	Ejecución del banco de pruebas BDP-REQ-C04	1
02/09/2016	Ejecución del banco de pruebas BDP-REQ-C05	1

Tabla 168 Calendario de pruebas para la integración de los subsistemas computacionales de tipo A y B en el sistema computacional híbrido.



12.4 Bancos de pruebas

ID Banco	Descripción	Iteración N°
BDP-REQ-C01	Agrupar todos los casos de prueba para verificar el requerimiento REQ-C01	1
BDP-REQ-C02	Agrupar todos los casos de prueba para verificar el requerimiento REQ-C02	1
BDP-REQ-C03	Agrupar todos los casos de prueba para verificar el requerimiento REQ-C03	1
BDP-REQ-C04	Agrupar todos los casos de prueba para verificar el requerimiento REQ-C04	1
BDP-REQ-C05	Agrupar todos los casos de prueba para verificar el requerimiento REQ-C05	1

Tabla 169 Bancos de pruebas para verificar los requerimientos de la integración de los subsistemas computacionales de tipo A y B en el sistema computacional híbrido.

12.5 Casos de prueba

12.5.1 Iteración N° 1

ID Banco de pruebas	BDP-REQ-C01
ID Caso de prueba	CP-C01-01
Descripción	Entrenar un ensamble compuesto por cinco árboles de decisión y cinco redes neuronales artificiales, con características y etiquetas relativas a la base de datos de <i>Caldas</i>
Valores de entrada	- Base de datos: <i>Caldas</i> - Características: $\begin{bmatrix} [0, 30, 30, 1] \\ [0.5, 27, 25, 2] \\ [1, 26, 24, 3] \\ [1, 19, 22, 3] \\ [2, 18, 21, 4] \\ [3, 0, 0, 7] \end{bmatrix}$

	- Etiquetas: $\begin{bmatrix} Sane \\ Sane \\ Mild \\ Mild \\ Serious \\ Serious \end{bmatrix}$
Pasos	- Ingresar al modo de entrenamiento del sistema - Seleccionar la base de datos <i>Caldas</i> - Seleccionar las características y etiquetas para el entrenamiento - Llamar al método de entrenamiento de ensamble con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: <i>True</i>
Resultado actual	- Retorno: <i>True</i>

Tabla 170 Ejecución del caso de prueba N° 1 para verificar el requerimiento REQ-C01 —entrenamiento de ensamble basado en Bagging—.

ID Banco de pruebas	BDP-REQ-C01
ID Caso de prueba	CP-C01-02
Descripción	Entrenar un ensamble compuesto por cinco árboles de decisión y cinco redes neuronales artificiales, con características y etiquetas relativas a la base de datos de la <i>ADNI</i>
Valores de entrada	- Base de datos: <i>ADNI</i> - Características: $\begin{bmatrix} [0, 30, 60, 20] \\ [0, 24, 90, 15] \\ [0.5, 28, 60, 14] \\ [0.5, 27, 90, 10] \\ [1, 23, 60, 10] \\ [3, 0, 90, 0] \end{bmatrix}$

	- Etiquetas: $\begin{bmatrix} Sane \\ Sane \\ Mild \\ Mild \\ Serious \\ Serious \end{bmatrix}$
Pasos	- Ingresar al modo de entrenamiento del sistema - Seleccionar la base de datos <i>ADNI</i> - Seleccionar las características y etiquetas para el entrenamiento - Llamar al método de entrenamiento de ensamble con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: <i>True</i>
Resultado actual	- Retorno: <i>True</i>

Tabla 171 Ejecución del caso de prueba N° 2 para verificar el requerimiento REQ-C01 —entrenamiento de ensamble basado en Bagging—.

ID Banco de pruebas	BDP-REQ-C02
ID Caso de prueba	CP-C02-01
Descripción	Asistir al diagnóstico de Deterioro Cognitivo Leve mediante un ensamble compuesto por cinco árboles de decisión y cinco redes neuronales artificiales, con características relativas a una persona sana en la base de datos de <i>Caldas</i>
Valores de entrada	- Base de datos: <i>Caldas</i> - Características: $[0, 30, 30, 1]$
Pasos	- Ingresar al modo de asistencia al diagnóstico - Seleccionar la base de datos <i>Caldas</i> - Seleccionar las características para la asistencia - Llamar al método de asistencia con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: $[Sane] [0.0007 \ 0.9992 \ 0.0569]$

Resultado actual	- Retorno: [<i>Sane</i>] [0.0007 0.9992 0.0569]
-------------------------	---

Tabla 172 Ejecución del caso de prueba N° 3 para verificar el requerimiento REQ-C02 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante ensamble basado en Bagging —.

ID Banco de pruebas	BDP-REQ-C02
ID Caso de prueba	CP-C02-02
Descripción	Asistir al diagnóstico de Deterioro Cognitivo Leve mediante un ensamble compuesto por cinco árboles de decisión y cinco redes neuronales artificiales, con características relativas a una persona con Deterioro Cognitivo Leve en la base de datos de <i>Caldas</i>
Valores de entrada	- Base de datos: <i>Caldas</i> - Características: $[1, 19, 22, 3]$
Pasos	- Ingresar al modo de asistencia al diagnóstico - Seleccionar la base de datos <i>Caldas</i> - Seleccionar las características para la asistencia - Llamar al método de asistencia con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: [<i>Mild</i>] [0.6903 0.0000 0.3096]
Resultado actual	- Retorno: [<i>Mild</i>] [0.6903 0.0000 0.3096]

Tabla 173 Ejecución del caso de prueba N° 4 para verificar el requerimiento REQ-C02 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante ensamble basado en Bagging —.

ID Banco de pruebas	BDP-REQ-C02
ID Caso de prueba	CP-C02-03
Descripción	Asistir al diagnóstico de Deterioro Cognitivo Leve mediante un ensamble compuesto por cinco árboles de decisión y cinco redes neuronales artificiales, con características relativas a una persona con deterioro cognitivo grave en la base de datos de <i>Caldas</i>
Valores de entrada	- Base de datos: <i>Caldas</i> - Características: $[3, 0, 0, 7]$

Pasos	- Ingresar al modo de asistencia al diagnóstico - Seleccionar la base de datos <i>Caldas</i> - Seleccionar las características para la asistencia - Llamar al método de asistencia con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: [<i>Serious</i>] [0.0034 0.0000 0.9965]
Resultado actual	- Retorno: [<i>Serious</i>] [0.0034 0.0000 0.9965]

Tabla 174 Ejecución del caso de prueba N° 5 para verificar el requerimiento REQ-C02 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante ensamble basado en Bagging—.

ID Banco de pruebas	BDP-REQ-C02
ID Caso de prueba	CP-C02-04
Descripción	Asistir al diagnóstico de Deterioro Cognitivo Leve mediante un ensamble compuesto por cinco árboles de decisión y cinco redes neuronales artificiales, con características relativas a una persona sana en la base de datos de la <i>ADNI</i>
Valores de entrada	- Base de datos: <i>ADNI</i> - Características: $[0, 30, 60, 20]$
Pasos	- Ingresar al modo de asistencia al diagnóstico - Seleccionar la base de datos <i>ADNI</i> - Seleccionar las características para la asistencia - Llamar al método de asistencia con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: [<i>Sane</i>] [0.0000 0.9999 0.0000]
Resultado actual	- Retorno: [<i>Sane</i>] [0.0000 0.9999 0.0000]

Tabla 175 Ejecución del caso de prueba N° 6 para verificar el requerimiento REQ-C02 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante ensamble basado en Bagging—.

ID Banco de pruebas	BDP-REQ-C02
ID Caso de prueba	CP-C02-05

Descripción	Asistir al diagnóstico de Deterioro Cognitivo Leve mediante un ensamble compuesto por cinco árboles de decisión y cinco redes neuronales artificiales, con características relativas a una persona con Deterioro Cognitivo Leve en la base de datos de la <i>ADNI</i>
Valores de entrada	- Base de datos: <i>ADNI</i> - Características: $[[0.5, 27, 90, 10]]$
Pasos	- Ingresar al modo de asistencia al diagnóstico - Seleccionar la base de datos <i>ADNI</i> - Seleccionar las características para la asistencia - Llamar al método de asistencia con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: <i>[Mild]</i> $[0.9368 \ 0.0000 \ 0.0631]$
Resultado actual	- Retorno: <i>[Mild]</i> $[0.9368 \ 0.0000 \ 0.0631]$

Tabla 176 Ejecución del caso de prueba N° 7 para verificar el requerimiento REQ-C02 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante ensamble basado en Bagging—.

ID Banco de pruebas	BDP-REQ-C02
ID Caso de prueba	CP-C02-06
Descripción	Asistir al diagnóstico de Deterioro Cognitivo Leve mediante un ensamble compuesto por cinco árboles de decisión y cinco redes neuronales artificiales, con características relativas a una persona con deterioro cognitivo grave en la base de datos de la <i>ADNI</i>
Valores de entrada	- Base de datos: <i>ADNI</i> - Características: $[[3, 0, 90, 0]]$
Pasos	- Ingresar al modo de asistencia al diagnóstico - Seleccionar la base de datos <i>ADNI</i> - Seleccionar las características para la asistencia - Llamar al método de asistencia con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: <i>[Serious]</i> $[0.0024 \ 0.0000 \ 0.9975]$

Resultado actual	- Retorno: [<i>Serious</i>] [0.0024 0.0000 0.9975]
-------------------------	--

Tabla 177 Ejecución del caso de prueba N° 8 para verificar el requerimiento REQ-C02 —asistencia al diagnóstico de Deterioro Cognitivo Leve mediante ensamble basado en Bagging—.

ID Banco de pruebas	BDP-REQ-C03
ID Caso de prueba	CP-C03-01
Descripción	Persistir un ensamble compuesto por cinco árboles de decisión y cinco redes neuronales artificiales, entrenado con la base de datos de <i>Caldas</i>
Valores de entrada	- Base de datos: <i>Caldas</i> - Ruta: <i>persistence/Caldas</i> - Ensamble basado en <i>Bagging</i>: modelo entrenado con la base de datos de <i>Caldas</i>
Pasos	- Seleccionar la base de datos <i>Caldas</i> - Seleccionar el ensamble basado en <i>Bagging</i> entrenado - Llamar al método de guardado de ensamble con las entradas mencionadas
Resultado esperado	- Existencia de los archivos <i>0tree.pkl</i> , <i>2tree.pkl</i> , <i>4tree.pkl</i> , <i>6tree.pkl</i> , <i>8tree.pkl</i> , <i>1net.h5</i> , <i>3net.h5</i> , <i>5net.h5</i> , <i>7net.h5</i> y <i>9net.h5</i> en la ruta <i>persistence/Caldas</i> con los datos del ensamble entrenado
Resultado actual	- Existencia de los archivos <i>0tree.pkl</i> , <i>2tree.pkl</i> , <i>4tree.pkl</i> , <i>6tree.pkl</i> , <i>8tree.pkl</i> , <i>1net.h5</i> , <i>3net.h5</i> , <i>5net.h5</i> , <i>7net.h5</i> y <i>9net.h5</i> en la ruta <i>persistence/Caldas</i> con los datos del ensamble entrenado

Tabla 178 Ejecución del caso de prueba N° 9 para verificar el requerimiento REQ-C03 —persistencia de ensamble basado en Bagging—.

ID Banco de pruebas	BDP-REQ-C03
ID Caso de prueba	CP-C03-02
Descripción	Persistir un ensamble compuesto por cinco árboles de decisión y

	cinco redes neuronales artificiales, entrenado con la base de datos de la <i>ADNI</i>
Valores de entrada	- Base de datos: <i>ADNI</i> - Ruta: <i>persistence/ADNI</i> - Ensamble basado en <i>Bagging</i>: modelo entrenado con la base de datos de la <i>ADNI</i>
Pasos	- Seleccionar la base de datos <i>ADNI</i> - Seleccionar el ensamble basado en <i>Bagging</i> entrenado - Llamar al método de guardado de ensamble con las entradas mencionadas
Resultado esperado	- Existencia de los archivos <i>0tree.pkl</i> , <i>2tree.pkl</i> , <i>4tree.pkl</i> , <i>6tree.pkl</i> , <i>8tree.pkl</i> , <i>1net.h5</i> , <i>3net.h5</i> , <i>5net.h5</i> , <i>7net.h5</i> y <i>9net.h5</i> en la ruta <i>persistence/ADNI</i> con los datos del ensamble entrenado
Resultado actual	- Existencia de los archivos <i>0tree.pkl</i> , <i>2tree.pkl</i> , <i>4tree.pkl</i> , <i>6tree.pkl</i> , <i>8tree.pkl</i> , <i>1net.h5</i> , <i>3net.h5</i> , <i>5net.h5</i> , <i>7net.h5</i> y <i>9net.h5</i> en la ruta <i>persistence/ADNI</i> con los datos del ensamble entrenado

Tabla 179 Ejecución del caso de prueba N° 10 para verificar el requerimiento REQ-C03 —persistencia de ensamble basado en *Bagging*—.

ID Banco de pruebas	BDP-REQ-C04
ID Caso de prueba	CP-C04-01
Descripción	Cargar un ensamble compuesto por cinco árboles de decisión y cinco redes neuronales artificiales, entrenado con la base de datos de <i>Caldas</i>
Valores de entrada	- Base de datos: <i>Caldas</i> - Ruta: <i>persistence/Caldas</i>
Pasos	- Seleccionar la base de datos <i>Caldas</i> - Llamar al método de cargue de ensamble con las entradas

	mencionadas
Resultado esperado	- Ensamble basado en <i>Bagging</i> cargado con los datos de los archivos <i>persistence/Caldas/*.*</i>
Resultado actual	- Ensamble basado en <i>Bagging</i> cargado con los datos de los archivos <i>persistence/Caldas/*.*</i>

Tabla 180 Ejecución del caso de prueba N° 11 para verificar el requerimiento REQ-C04 —cargue de ensamble basado en *Bagging*—.

ID Banco de pruebas	BDP-REQ-C04
ID Caso de prueba	CP-C04-02
Descripción	Cargar un ensamble compuesto por cinco árboles de decisión y cinco redes neuronales artificiales, entrenado con la base de datos de la <i>ADNI</i>
Valores de entrada	- Base de datos: <i>ADNI</i> - Ruta: <i>persistence/ADNI</i>
Pasos	- Seleccionar la base de datos <i>ADNI</i> - Llamar al método de cargue de ensamble con las entradas mencionadas
Resultado esperado	- Ensamble basado en <i>Bagging</i> cargado con los datos de los archivos <i>persistence/ADNI/*.*</i>
Resultado actual	- Ensamble basado en <i>Bagging</i> cargado con los datos de los archivos <i>persistence/ADNI/*.*</i>

Tabla 181 Ejecución del caso de prueba N° 12 para verificar el requerimiento REQ-C04 —cargue de ensamble basado en *Bagging*—.

ID Banco de pruebas	BDP-REQ-C05
ID Caso de prueba	CP-C05-01
Descripción	Validar el desempeño de un ensamble compuesto por cinco árboles de decisión y cinco redes neuronales artificiales, entrenado con la base de datos de <i>Caldas</i>

Valores de entrada	- Base de datos: <i>Caldas</i> - Ensamble basado en <i>Bagging</i>: modelo entrenado con la base de datos de <i>Caldas</i> - Archivo: <i>Caldas_Small.csv</i> el cual cumple con la plantilla descrita en el anexo 9 - Cantidad de pliegues: 10
Pasos	- Ingresar al modo de validación del sistema - Seleccionar la base de datos <i>Caldas</i> - Seleccionar el ensamble basado en <i>Bagging</i> entrenado - Seleccionar el archivo <i>Caldas_Small.csv</i> - Llamar al método de validación de ensamble con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: <i>True True 0.9675 0.9676 0.9675 0.9672 0.9379</i>
Resultado actual	- Retorno: <i>True True 0.9675 0.9676 0.9675 0.9672 0.9379</i>

Tabla 182 Ejecución del caso de prueba N° 13 para verificar el requerimiento REQ-C05 —desempeño de ensamble basado en *Bagging*—.

ID Banco de pruebas	BDP-REQ-C05
ID Caso de prueba	CP-C05-02
Descripción	Validar el desempeño de un ensamble compuesto por cinco árboles de decisión y cinco redes neuronales artificiales, entrenado con la base de datos de la <i>ADNI</i>
Valores de entrada	- Base de datos: <i>ADNI</i> - Ensamble basado en <i>Bagging</i>: modelo entrenado con la base de datos de <i>ADNI</i> - Archivo: <i>ADNI_Small.csv</i> el cual cumple con la plantilla descrita en el anexo 8 - Cantidad de pliegues: 10
Pasos	- Ingresar al modo de validación del sistema

	- Seleccionar la base de datos <i>ADNI</i> - Seleccionar el ensamble basado en <i>Bagging</i> entrenado - Seleccionar el archivo <i>ADNI_Small.csv</i> - Llamar al método de validación de ensamble con las entradas mencionadas - Obtener el retorno del método
Resultado esperado	- Retorno: <i>True True</i> 0.9076 0.9096 0.9076 0.9648 0.8526
Resultado actual	- Retorno: <i>True True</i> 0.9076 0.9096 0.9076 0.9648 0.8526

Tabla 183 Ejecución del caso de prueba N° 14 para verificar el requerimiento REQ-C05 —desempeño de ensamble basado en *Bagging*—.