

**DESARROLLO DE UNA APLICACIÓN MÓVIL COMO APOYO PARA EL
PROCESO DE GESTIÓN DE LA PARAFISCALIDAD DE LA CAJA DE
COMPENSACIÓN FAMILIAR DEL TOLIMA “COMFATOLIMA”**



GERMÁN CASTELLANOS ROMERO

Director de tesis
Ing. Ms. Jorge Iván Meza Martínez

UNIVERSIDAD AUTÓNOMA DE MANIZALES
MAESTRÍA EN GESTIÓN Y DESARROLLO DE PROYECTOS DE SOFTWARE
LÍNEA DE INVESTIGACIÓN EN CALIDAD DE SOFTWARE
ÉNFASIS EN DESARROLLO DE SOFTWARE
MANIZALES
2017

**DESARROLLO DE UNA APLICACIÓN MÓVIL COMO APOYO PARA EL
PROCESO DE GESTIÓN DE LA PARAFISCALIDAD DE LA CAJA DE
COMPENSACIÓN FAMILIAR DEL TOLIMA “COMFATOLIMA”**



GERMÁN CASTELLANOS ROMERO

Director de tesis
Ing. Ms. Jorge Iván Meza Martínez

Informe final de la Tesis

UNIVERSIDAD AUTÓNOMA DE MANIZALES
MAESTRÍA EN GESTIÓN Y DESARROLLO DE PROYECTOS DE SOFTWARE
LÍNEA DE INVESTIGACIÓN EN CALIDAD DE SOFTWARE
ÉNFASIS EN DESARROLLO DE SOFTWARE
MANIZALES
2017

Resumen

La penetración de Internet y los dispositivos de comunicación móvil en todos los sectores de la sociedad, ha obligado a las empresas a replantear sus canales de comunicación, a redefinir sus procesos de manejo de información y a fortalecer su infraestructura tecnológica, esto para responder a las nuevas expectativas del cliente y a su creciente necesidad de poder realizar todas sus transacciones a través de su computadora, tableta o teléfono, desde cualquier lugar y a cualquier hora. Hoy más que nunca la computación ubicua es un reto inaplazable para las organizaciones que desean perdurar en el tiempo.

El propósito del presente proyecto es desarrollar una aplicación móvil para facilitarle al afiliado a Comfatolima, la posibilidad de comunicarse con la Caja desde cualquier dispositivo móvil conectado a Internet, poniendo a su disposición las transacciones que habitualmente realiza en forma presencial o a través de la página web corporativa, en lo referente al proceso de Gestión de la Parafiscalidad, concretamente al pago de la cuota monetaria del subsidio familiar.

Para llevar a cabo un proyecto de estas características, es preciso establecer aquellos trámites de carácter obligatorio para los beneficiarios del subsidio familiar, evaluar los procesos al interior de la Caja para adaptarlos al entorno móvil, diseñar una arquitectura adecuada que garantice el cumplimiento de los atributos de calidad tales como disponibilidad, modificabilidad, rendimiento, seguridad, usabilidad y realizar pruebas exhaustivas al producto. Se espera lograr un impacto positivo tanto para el afiliado, aligerando y dinamizando sus trámites y para Comfatolima, en la reducción de costos operativos y tiempos de respuesta.

Palabras clave: internet, aplicación móvil, computación ubicua, dispositivo móvil, Comfatolima, subsidio familiar.

Abstract

The penetration of Internet and mobile communication devices in all sectors of society, has forced companies to rethink their communication channels, to redefine their information management processes and strengthen its technological infrastructure, to respond to this new customer expectations and their increasing need to be able to perform all their transactions through your computer, table, or phone, anywhere, anytime. Today more than ever ubiquitous computing is an urgent challenge for organizations that want to survive over time.

The purpose of this project is to build a mobile application to provide to the member Comfatolima, the ability to communicate with the fund from any mobile device connected to the Internet, putting at your disposal transactions usually carried out in person or through the corporate website, in relation to the payment of the monetary quota of the family's allowance.

To carry out a project of this nature, is necessary to establish those procedures mandatory for beneficiaries of family allowance, evaluate processes within the Fund to adapt to the mobile environment, designing architecture to ensure compliance of the attributes quality such as availability, modifiability, performance, security, usability and conduct comprehensive product testing. It is expected to achieve a positive impact on both the affiliate, lightening and streamlining its procedures and Comfatolima, reducing operating costs and response times.

Keywords: internet, aplicación móvil, computación ubicua, dispositivo móvil, Comfatolima, subsidio familiar.

Introducción

El presente proyecto tiene como fin desarrollar una aplicación para dispositivos móviles con sistema operativo Android, que se constituya en un mecanismo de comunicación efectivo entre la Caja de Compensación Familiar del Tolima Comfatolima y sus trabajadores afiliados, en lo referente al proceso de Gestión de la Parafiscalidad, concretamente al pago de la cuota monetaria de Subsidio Familiar.

El proceso de pago de cuota monetaria, sobre el cual se enfoca el presente proyecto, junto con los procesos de recaudo de aportes parafiscales y afiliaciones de empresas y trabajadores, hacen parte del macro proceso de gestión de la parafiscalidad, el cual se constituye en la mayor fuente de ingresos de Comfatolima y en el principal servicio que reciben los afiliados de escasos recursos. Este proceso presenta inconvenientes tales como la dificultad de comunicarse con los afiliados, para dar solución a las diferentes novedades y el desconocimiento por parte de éstos, de las fechas límite y documentos requeridos para acceder al pago del Subsidio, el cual es un beneficio fundamental para un alto porcentaje de la población afiliada a Comfatolima.

Un problema frecuente del proceso es que al intentar comunicarse con el afiliado mediante teléfono celular, correo electrónico o comunicación escrita a su dirección de residencia, con datos tomados del sistema de información del área de subsidio, no es posible la comunicación debido a la desactualización de la base de datos.

Si bien Comfatolima cuenta con una página web que permite a los afiliados consultar su información y comunicarse con la Caja, de un total de 53.358 afiliados, solo 1.437 de ellos están registrados y la utilizan, es decir el 2.69%.

En Colombia existen un total de cuarenta y tres Cajas de Compensación Familiar, de las cuales solo nueve tienen implementadas aplicaciones móviles para el afiliado, enfocadas principalmente a la divulgación del portafolio de servicios, no hay un énfasis en el proceso de pago de cuota monetaria.

Se espera que la realización de este proyecto represente para Comfatolima una reducción de costos operativos y establezca un canal de comunicación ágil y dinámico, contribuyendo a la fidelización de los clientes y al posicionamiento de la imagen corporativa de la Caja.

Tabla de Contenidos

1. REFERENTE CONTEXTUAL	18
1.1 Descripción área problemática.....	18
1.2 Antecedentes	21
1.2.1 Interacción electrónica con el cliente en Comfatolima	21
1.2.2 Metodologías ágiles para el desarrollo de aplicaciones móviles.....	23
1.2.3 Otras metodologías propuestas.....	25
1.2.3.1 Metodología MDAM.....	26
1.2.3.2 Metodología de adaptación de la información al dispositivo.....	29
1.2.4 Revisión de un proyecto de desarrollo de aplicaciones móviles	31
1.3 Justificación	36
1.4 Formulación del problema	40
1.5 Objetivos	41
1.5.1 Objetivo general	41
1.5.2 Objetivos específicos.....	41
1.6 Alcances y limitaciones	42
1.7 Resultado esperado	43
2. ESTRATEGIA METODOLÓGICA.....	44
2.1 Metodología	44
2.1.1 Modelo de proceso incremental	44
2.1.2 PSP0 y PSP1.....	45
2.2 Recolección de información.....	50
2.2.1 Encuesta	50
2.2.1.1 Tamaño de la muestra	50
2.2.1.2 Herramienta de recolección de información	52
2.2.1.3 Resultados de la encuesta	53
2.3 Pruebas	56
2.4 Presupuesto	58
2.5 Cronograma.....	60
3. REFERENTE TEÓRICO.....	61
3.1 Cajas de compensación familiar	61
3.1.1 Cafam	61
3.1.2 Comfaboy.....	62

3.1.3 Cajacopi.....	62
3.1.4 Comfamiliar Huila.....	63
3.1.5 Comfenalco Antioquia	63
3.1.6 Comfenalco Valle.....	64
3.1.7 Colsubsidio.....	64
3.1.8 Comfamiliares	65
3.1.9 Comfamiliar Risaralda	65
3.2 Conceptos sobres aplicaciones móviles	67
3.2.1 Definición.....	67
3.2.2 Tipos de aplicaciones móviles.....	67
3.2.2.1 Aplicaciones nativas.....	67
3.2.2.2 Aplicaciones móviles basadas en la web.....	69
3.2.2.3 Aplicaciones híbridas	71
3.3 Modelos de negocio	73
3.3.1 Freemium	73
3.3.2 Suscripción.....	74
3.3.3 Por publicidad insertada	75
3.3.4 Canal de venta	75
3.3.5 Fidelización	76
3.4 Objetivos de negocio.....	77
3.4.1 Generación de imagen de marca	77
3.4.2 Fidelización de usuarios/clientes.....	77
3.4.3 Herramienta de gestión.....	78
3.4.4 Herramienta y canal de venta	78
3.4.5 Aplicaciones generadoras de ingresos.....	78
3.5 Arquitecturas móviles	78
3.5.1 Aplicaciones autónomas o independientes.....	78
3.5.2 Cliente liviano	79
3.5.3 Cliente enriquecido	80
3.5.4 Cliente pesado	81
3.6 Servicios web	83
3.6.1 RESTful.....	83
3.6.2 SOAP.....	84
3.7 Herramientas de desarrollo	85

3.7.1 Eclipse	85
3.7.2 Netbeans	86
3.7.3 Android Studio	86
3.8 La privacidad personal	86
3.8.1 Normatividad sobre aplicaciones y datos	88
3.8.1.1 Ley estatutaria 1266 de 2008	88
3.8.1.2 Constitución política de Colombia	88
3.8.1.3 Sentencia C-1011/08	89
3.8.1.4 Derecho al habeas data	89
3.8.1.5 ¿Qué es un dato personal?	89
3.8.1.6 Ley estatutaria 1581 de 2012	90
3.8.1.7 Control de la SIC	90
3.8.1.8 Política de privacidad	92
4. DESARROLLO	93
4.1 Fase de requerimientos	93
4.2 Diseño de la arquitectura	94
4.3 Iteraciones del proyecto	99
4.3.1 Ingreso al sistema y cambio de contraseña	99
4.3.1.1 Definición de requerimientos	99
4.3.1.2 Casos de uso textuales	99
4.3.1.3 Prototipos de interfaces y reportes	104
4.3.1.4 Diagrama de actividades	105
4.3.1.5 Pruebas	106
4.3.1.6 Despliegue	106
4.3.1.7 PSP - tamaños y tiempos estimados versus reales	107
4.3.1.8 PSP – defectos detectados y removidos	108
4.3.1.9 Análisis de la iteración	109
4.3.2 Consultar aportes	111
4.3.2.1 Definición de requerimientos	111
4.3.2.2 Casos de uso textuales	111
4.3.2.3 Prototipos de interfaces y reportes	113
4.3.2.4 Diagrama de actividades	114
4.3.2.5 Pruebas	114
4.3.2.6 Despliegue	115

4.3.2.7 PSP - tamaños y tiempos estimados versus reales	116
4.3.2.8 PSP – defectos detectados y removidos	117
4.3.2.9 Análisis de la iteración	118
4.3.3 Consultar subsidios	119
4.3.3.1 Definición de requerimientos	119
4.3.3.2 Casos de uso textuales.....	119
4.3.3.3 Prototipos de interfaces y reportes	121
4.3.3.4 Diagrama de actividades	122
4.3.3.5 Pruebas	123
4.3.3.6 Despliegue	123
4.3.3.7 PSP - tamaños y tiempos estimados versus reales	125
4.3.3.8 PSP – defectos detectados y removidos	126
4.3.3.9 Análisis de la iteración	127
4.3.4 Consultar tarjeta	128
4.3.4.1 Definición de requerimientos	128
4.3.4.2 Casos de uso textuales.....	128
4.3.4.3 Prototipos de interfaces y reportes	130
4.3.4.4 Diagrama de actividades	131
4.3.4.5 Pruebas	132
4.3.4.6 Despliegue	132
4.3.4.7 PSP - tamaños y tiempos estimados versus reales	133
4.3.4.8 PSP – defectos detectados y removidos	134
4.3.4.9 Análisis de la iteración	135
4.3.5 Consultar datos familia.....	136
4.3.5.1 Definición de requerimientos	136
4.3.5.2 Casos de uso textuales.....	136
4.3.5.3 Prototipos de interfaces y reportes	139
4.3.5.4 Diagrama de actividades	140
4.3.5.5 Pruebas	141
4.3.5.6 Despliegue	141
4.3.5.7 PSP - tamaños y tiempos estimados versus reales	142
4.3.5.8 PSP – defectos detectados y removidos	143
4.3.5.9 Análisis de la iteración	143
4.3.6 Generar certificados de afiliación	145

4.3.6.1 Definición de requerimientos	145
4.3.6.2 Casos de uso textuales.....	145
4.3.6.3 Prototipos de interfaces y reportes	148
4.3.6.4 Diagrama de actividades	149
4.3.6.5 Pruebas	151
4.3.6.6 Despliegue	151
4.3.6.7 PSP - tamaños y tiempos estimados versus reales	152
4.3.6.8 PSP – defectos detectados y removidos	153
4.3.6.9 Análisis de la iteración	153
4.3.7 Actualizar mensajes.....	155
4.3.7.1 Definición de requerimientos	155
4.3.7.2 Casos de uso textuales.....	155
4.3.7.3 Prototipos de interfaces y reportes	160
4.3.7.4 Diagrama de actividades	161
4.3.7.5 Pruebas	163
4.3.7.6 Despliegue	163
4.3.7.7 PSP - tamaños y tiempos estimados versus reales	165
4.3.7.8 PSP – defectos detectados y removidos	166
4.3.7.9 Análisis de la iteración	167
4.3.8 Actualizar mensajes administrador	168
4.3.8.1 Definición de requerimientos	168
4.3.8.2 Casos de uso textuales.....	168
4.3.8.3 Diagrama de caso de uso	169
4.3.8.4 Diagrama de actividades	169
4.3.8.5 Diagrama de clases	171
4.3.8.6 Diagramas de secuencia	172
4.3.8.7 Diagramas de despliegue.....	174
4.3.8.8 Pruebas	174
4.3.8.9 Despliegue	175
4.3.8.10 PSP - tamaños y tiempos estimados versus reales	179
4.3.8.11 PSP – defectos detectados y removidos	179
5. ANALISIS DE RESULTADOS	185
6. CONCLUSIONES	189
7. RECOMENDACIONES	193

8. LECCIONES APRENDIDAS	195
9. BIBLIOGRAFIA	206
10. ANEXOS	210

Lista de tablas

Tabla 1. Matriz DOFA Comfatolima	21
Tabla 2. Análisis 6M para modelo MDAM (Gasca et ál., 2014)	29
Tabla 3. Fases PSP a aplicar en el proyecto	46
Tabla 4. Metodología aplicar en el proyecto.....	47
Tabla 5. Resultados encuesta Comfatolima.....	54
Tabla 6.Resultados teléfono inteligente versus plan de datos	55
Tabla 7.Costos del personal involucrado en el proyecto.	58
Tabla 8. Equipos requeridos para desarrollo.....	58
Tabla 9. Equipos requeridos para pruebas	58
Tabla 10. Insumos y materiales requeridos en el proyecto	59
Tabla 11. Costos totales del proyecto	59
Tabla 12.Cronograma del proyecto	60
Tabla 13. Funcionalidades de las aplicaciones móviles de Cajas de Compensación	65
Tabla 14. Características aplicaciones nativas por sistema operativo (IBM, 2012, p. 3)	68
Tabla 15. Características de aplicaciones basadas en web (IBM, 2012, p. 5)	70
Tabla 16. Comparativo aplicaciones nativas, híbridas y web (IBM, 2012, p.8)	72
Tabla 17. Caso de uso textual CU01	99
Tabla 18. Caso de uso textual CU02	102
Tabla 19. Caso de uso textual CU05	111
Tabla 20. Caso de uso textual CU06	119
Tabla 21. Caso de uso textual CU07	128
Tabla 22. Caso de uso textual CU03	136
Tabla 23. Caso de uso textual CU04	145
Tabla 24. Caso de uso textual CU08_01	155
Tabla 25. Caso de uso textual CU08_02	157
Tabla 26. Caso de uso textual CU08_03	158
Tabla 27. Defectos detectados y removidos	179
Tabla 28.Cronograma estimado para el proyecto.....	186
Tabla 29.Cronograma final del proyecto	187
Tabla 30.Costos de personal presupuestados	187
Tabla 31.Costos de personal ejecutados	188

Lista de figuras

<i>Figura 1. Etapas de la metodología MDAM. (Gasca et ál., 2014)</i>	26
<i>Figura 2. Arquitectura proyecto SEDE (Gasca et ál., 2014)</i>	33
<i>Figura 3. Diagrama de secuencia en aplicación móvil mediante SMS. (Gama et ál., 2014)</i>	34
<i>Figura 4. Módulo de consulta y generación de documentos. (Gama et ál., 2014)</i>	34
<i>Figura 5. Solicitud de documento desde dispositivo móvil. (Gama et ál., 2014)</i>	35
<i>Figura 6. Modelo incremental. (Pressman, 2010, p.35)</i>	45
<i>Figura 7. Notificación número de afiliados a Comfatolima con teléfono celular registrado</i>	51
<i>Figura 8. Notificación resultado encuestas</i>	53
<i>Figura 9. Aplicación móvil Cafam</i>	61
<i>Figura 10. Aplicación móvil Comfaboy</i>	62
<i>Figura 11. Aplicación móvil Cajacopi</i>	62
<i>Figura 12. Aplicación móvil Comfamiliar Huila</i>	63
<i>Figura 13. Aplicación móvil Comfenalco Antioquia</i>	63
<i>Figura 14. Aplicación móvil Comfenalco Valle</i>	64
<i>Figura 15. Aplicación móvil Colsubsidio</i>	64
<i>Figura 16. Aplicación móvil Confamiliares</i>	65
<i>Figura 17. Aplicación móvil Comfamiliar Risaralda</i>	65
<i>Figura 18. Arquitectura de una aplicación nativa.</i>	69
<i>Figura 19. Arquitectura de una aplicación.</i>	71
<i>Figura 20. Arquitectura de una aplicación híbrida.</i>	72
<i>Figura 21. Ejemplo de arquitectura autónoma.</i>	79
<i>Figura 22. Ejemplo de arquitectura cliente liviano.</i>	80
<i>Figura 23. E.g. Arquitectura cliente enriquecido.</i>	81
<i>Figura 24. Ejemplo de Arquitectura cliente pesado (Washke, 2015, p.93).</i>	83
<i>Figura 25. Servicio REST .</i>	84
<i>Figura 26. Servicio Soap.</i>	85
<i>Figura 27. Diagrama de casos de uso aplicación móvil</i>	95
<i>Figura 28. Diagrama de casos de uso para el subsistema de administración.</i>	95
<i>Figura 29. Diagrama de componentes aplicación móvil.</i>	96
<i>Figura 30. prototipo funcionalidad ingresar al sistema</i>	104
<i>Figura 31. Diagrama de actividades caso de uso CU01</i>	105
<i>Figura 32. Pantalla funcionalidades de ingresar al sistema y cambiar o restablecer clave.</i>	106

<i>Figura 33. Tamaños y tiempos estimados versus reales.</i>	107
<i>Figura 34. Defectos detectados y removidos.</i>	108
<i>Figura 35. Prototipo funcionalidad consultar aportes.</i>	113
<i>Figura 36. Diagrama de actividades caso de uso CU05.</i>	114
<i>Figura 37. Pantalla que ofrece la funcionalidad de consultar aportes</i>	115
<i>Figura 38. Tamaños y tiempos estimados versus reales.</i>	116
<i>Figura 39. Defectos detectados y removidos.</i>	117
<i>Figura 40. Prototipo funcionalidad consultar subsidios</i>	121
<i>Figura 41. Diagrama de actividades caso de uso CU06.</i>	122
<i>Figura 42. La pantalla que ofrece la funcionalidad de consultar subsidios</i>	123
<i>Figura 43. Tamaños y tiempos estimados versus reales</i>	125
<i>Figura 44. Defectos detectados y removidos.</i>	126
<i>Figura 45. Prototipo funcionalidad consultar tarjeta.</i>	130
<i>Figura 46 diagrama de actividades caso de uso CU07.</i>	131
<i>Figura 47. Pantalla que ofrece la funcionalidad de consultar tarjeta</i>	132
<i>Figura 48. Tamaños y tiempos estimados versus reales.</i>	133
<i>Figura 49. Defectos detectados y removidos.</i>	134
<i>Figura 50. Prototipo funcionalidad consultar datos familia.</i>	139
<i>Figura 51. Diagrama de actividades caso de uso CU03.</i>	140
<i>Figura 52. Pantalla que ofrece la funcionalidad de consultar datos familia.</i>	141
<i>Figura 53. Tamaños y tiempos estimados versus reales.</i>	142
<i>Figura 54. Defectos detectados y removidos.</i>	143
<i>Figura 55. Prototipo funcionalidad generar certificados.</i>	148
<i>Figura 56. Diagrama de actividades caso de uso CU04.</i>	149
<i>Figura 57. Pantallas que ofrece la funcionalidad de generar certificados de afiliación</i>	151
<i>Figura 58. Tamaños y tiempos estimados versus reales</i>	152
<i>Figura 59. Defectos detectados y removidos</i>	153
<i>Figura 60. Prototipo funcionalidad actualizar mensajes</i>	160
<i>Figura 61. Diagrama de actividades caso de uso CU08_01</i>	161
<i>Figura 62. Diagrama de actividades caso de uso CU08_02</i>	162
<i>Figura 63. Las pantallas que ofrece la funcionalidad de generar certificados de afiliación.</i>	163
<i>Figura 64. Tamaños y tiempos estimados versus reales.</i>	165
<i>Figura 65. Defectos detectados y removidos.</i>	166
<i>Figura 66. Diagrama de casos de uso.</i>	169

<i>Figura 67. Diagrama de actividades leer mensaje.....</i>	<i>169</i>
<i>Figura 68. Diagrama de actividades redactar mensaje.</i>	<i>170</i>
<i>Figura 69. Diagrama de actividades borrar mensaje.</i>	<i>170</i>
<i>Figura 70. Diagrama de clases actualizar mensajes.....</i>	<i>171</i>
<i>Figura 71. Diagrama de secuencia leer y responder mensajes.....</i>	<i>172</i>
<i>Figura 72. Diagrama de secuencia redactar mensaje.....</i>	<i>173</i>
<i>Figura 73. Diagrama de secuencia borrar mensaje.....</i>	<i>173</i>
<i>Figura 74. Diagrama de despliegue sistema de administración.</i>	<i>174</i>
<i>Figura 75. Ingreso al sistema de administración.</i>	<i>175</i>
<i>Figura 76. Usuarios sistema de administración.</i>	<i>175</i>
<i>Figura 77. Agregar usuarios sistema de administración.</i>	<i>176</i>
<i>Figura 78. Modificar usuarios sistema de administración.....</i>	<i>176</i>
<i>Figura 79. Lista mensajes sistema de administración.</i>	<i>176</i>
<i>Figura 80. Leer mensaje sistema de administración.</i>	<i>177</i>
<i>Figura 81. Redactar mensaje sistema de administración.....</i>	<i>178</i>

1. Referente Contextual

1.1. Descripción área problemática

La Caja de Compensación Familiar del Tolima Comfatolima, es una entidad de derecho privado sin ánimo de lucro creada el 25 de junio de 1993, según resolución No. 0413 de la Superintendencia del Subsidio Familiar, producto de la fusión de las Cajas de Compensación Comfaro y Comfacopi del Tolima (Comfatolima, 2016a).

Las Cajas de Compensación funcionan enmarcadas dentro de la siguiente normatividad:

- Ley 21 de 1982, mediante la cual se define todo el régimen del Subsidio Familiar (“Por la cual se modifica el régimen del Subsidio Familiar y se dictan otras disposiciones”). (Comfatolima, 2016b).
- Ley 789 de 2002, mediante la cual se modifican aspectos de la ley 21 de 1982 (“Por la cual se dictan normas para apoyar el empleo y ampliar la protección social y se modifican algunos artículos del Código Sustantivo del Trabajo”). (Comfatolima, 2016b).

Según cifras suministradas por la Oficina de Subsidio y Aportes de Comfatolima, el número de afiliados que se benefician del subsidio familiar a través de la cuota monetaria asciende a 18.051, que equivale al 33.83% del total de afiliados. Este número de beneficiarios redime el valor pagado por la Caja con su tarjeta de afiliación, también deben actualizar periódicamente los documentos de sus familiares (constancias de estudio) para acceder a este beneficio y a su vez, dependen del pago de aportes de sus respectivos empleadores para que la Caja realice el pago mensual del subsidio.

El proceso de pago de cuota monetaria depende de terceros para funcionar adecuadamente, tales como los operadores financieros y tecnológicos del Sistema de Planilla Integrada de Liquidación de Aportes (PILA), quienes reportan a las cajas los pagos de aportes parafiscales realizados por las empresas afiliadas. También hacen parte del esquema los establecimientos comerciales donde los afiliados redimen sus cuotas monetarias y finalmente la red Redeban Multicolor, donde se manejan los movimientos y saldos de las tarjetas. Es importante señalar que ninguno de los terceros de los cuales depende el proceso suministra a la Caja información en línea, toda la información llega a Comfatolima en archivos planos mínimo 24 horas después de hacerse realizado la transacción.

Aparte de los inconvenientes que genera el no disponer de información inmediata por parte de los operadores, los principales factores por los cuales se ve afectado el pago de la cuota monetaria a los afiliados son los siguientes:

- Inconsistencias en los datos suministrados por los operadores de la Planilla Integrada de Liquidación de Aportes (pila)
- Datos desactualizados tanto del trabajador como del núcleo familiar
- La no entrega o entrega extemporánea por parte de los trabajadores afiliados, de los certificados de escolaridad de sus hijos, ya sea por desconocimiento de las fechas límite, por distancia geográfica o por demoras en los centro educativos
- Las variaciones transitorias de salario (vacaciones, primas, bonificaciones, entre otras), que hacen que el nivel de ingresos del mes exceda el valor de referencia y por ende no se le pague el subsidio familiar.

Los trabajadores afiliados a Comfatolima en su gran mayoría son personas con edades entre los 17 y 60 años distribuidas en todos los municipios del Departamento del Tolima,

con un contrato de trabajo en empresas legalmente constituidas, con formación académica que va desde básica primaria, secundaria, hasta universitaria. El 33.83% ganan menos de cuatro salarios mínimos y tienen un grupo familiar. Este porcentaje de afiliados recibe mensualmente la cuota monetaria del subsidio familiar que otorga el gobierno por estas condiciones socio económicas.

1.2. Antecedentes

1.2.1. Interacción electrónica con el cliente en Comfatolima

La Caja cuenta con la página web “www.comfatolima.com.co”, la cual pone a disposición de los afiliados a la Caja los siguientes trámites que pueden realizarse virtualmente:

- Consulta de datos de afiliación del trabajador y su grupo familiar. Por cada uno de los miembros del grupo se visualiza el tipo de identificación, el tipo de afiliación (cónyuge, hijo, padre, hermano), nombre completo, sexo, fecha de nacimiento. A través de esta opción el afiliado también puede solicitar la actualización de datos, por ejemplo el cambio de documento de identidad de un hijo, el cambio de número celular dirección de residencia entre otros (estos últimos datos muy pocas veces son actualizados), ver anexos 1 y 2.
- Consulta de los seis últimos pagos de cuota monetaria que se le han realizado, discriminados por cada uno de los beneficiarios que tienen derecho al pago, ver anexo 3.
- Generación y descarga de certificados de afiliación de todos los miembros del núcleo familiar, solo aplica para afiliados en estado activo, ver anexo 4.

No obstante la existencia de esta herramienta para consultar y actualizar datos, de un total de 51.669 afiliados solo están registrados en el portal 1.253, que corresponden al 2.43% de la población. Teniendo en cuenta que estas opciones están disponibles desde el año 2011, se concluye que esta herramienta no está resultando atractiva para los afiliados, por lo tanto es necesario proponer estrategias que permitan incrementar la interacción Afiliado - Caja a través de Internet.

Con el objetivo de realizar un diagnóstico de la situación actual de la Caja, en el tema de actualización de bases de datos y mecanismos de comunicación con el afiliado, se ha elaborado la siguiente matriz DOFA:

Tabla 1. Matriz DOFA Comfatolima

Debilidades		Oportunidades	
D1	Base de datos desactualizada	O1	Herramientas software libre y metodologías de desarrollo ágil
D2	Poco uso del portal web por parte del trabajador afiliado para temas de subsidio y actualización de datos	O2	Buena oferta de infraestructura tecnológica para implementar el proyecto
D3	Carencia de conexión entre página web y sistema de información de subsidio (este proceso no es en línea y genera carga operativa por migración de archivos e inserción en la base de datos del portal)	O3	Se tiene identificada una parte de la población afiliada, que está habituada a realizar transacciones en línea con el Colegio Comfatolima ¹
D4	Actualización de datos manual e ineficaz (las solicitudes de actualización realizadas por el portal, son revisadas por un funcionario e ingresadas al sistema de subsidio, no se actualizan todos los datos)		

¹ Colegio Comfatolima: Servicio social de educación prestado por Comfatolima en la ciudad de Ibagué, cuenta con oferta educativa en preescolar, educación básica y media a una población de 990 estudiantes (Colegio Comfatolima, 2016).

Fortalezas		Amenazas	
F1	Apoyo Gerencial para la solucionar la problemática (asignación de recursos financieros y humanos)	A1	Cambios en la normatividad del subsidio familiar, tendiente a unificar las cajas de compensación en el Departamento del Tolima
F2	Área de tecnología con experiencia en desarrollo de software	A2	Ataques informáticos, intrusiones a la red corporativa
F3	Coordinador del área de subsidio y aportes con formación tecnológica	A3	Normatividad referente a la protección de datos de usuario y sanciones asociadas
F4	Imagen corporativa de Comfatolima en la Región, identificada por los Ibaguereños como "La Caja del Tolima"		

Fuente: elaboración propia

1.2.2. Metodologías ágiles para el desarrollo de aplicaciones móviles. En el artículo denominado “Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles” de (Amaya, 2013), se plantea la heterogeneidad de plataformas, canales, capacidades y dispositivos móviles existentes en el mercado, haciendo énfasis en que muchos de los desarrollos de aplicaciones para este tipo de dispositivos se hacen de manera desordenada y por desarrolladores individuales, los cuales no aplican ningún tipo de metodología que garantice la calidad de los productos generados.

Para solucionar esta problemática y responder a las necesidades y expectativas del mercado, se plantea el uso de metodologías ágiles y el desarrollo iterativo, centrado especialmente en los requisitos cambiantes y la gestión del riesgo. Con esto se busca reducir tanto el tiempo de desarrollo como la incertidumbre mediante la división del proyecto en iteraciones independientemente funcionales, donde cada una de ellas produce un producto terminado o entregable.

Se hace énfasis en los beneficios de incorporar metodologías ágiles a proyectos pequeños de desarrollo de software, sin reemplazar las metodologías tradicionales sino trabajando en conjunto. Adicional a esto todas las características de las metodologías ágiles concuerdan con el desarrollo de aplicaciones móviles, enfrentando la complejidad de no poder caracterizar completamente quién será el cliente de la aplicación, especialmente las metodologías incrementales enfocadas a las rápidas entregas de software y a la reducción del riesgo.

Las metodologías concretas a que hace referencia el artículo, están orientadas a responder a las demandas de un mercado cambiante y con requisitos volátiles que deben resolverse en un tiempo corto, entre otras propone:

- *Extreme Programming (XP)*: planificación, pequeñas emisiones, diseño sencillo, pruebas, refactorización, programación en parejas. Es una metodología muy liviana para equipos medianos y pequeños que deben afrontar proyecto con requerimientos cambiantes.
- *Scrum*: planificación, *scrum master*, equipos, *sprints* (incrementos), pila de producto, *product backlog*, *burndown chart*. Se aplica principalmente en proyectos donde es muy difícil predecir y planificar el futuro, mediante la priorización de las iteraciones.
- *Test Driven Development (TDD)*: Se enfoca en la calidad del producto final, se escriben primero las pruebas a realizar, siendo estas las que determinan el código que se debe

escribir. Por sus buenos resultados, existe una tendencia a incorporar *TDD* a otras metodologías.

Se hace una revisión de los sistemas operativos más influyentes en el mercado de los dispositivos móviles y con gran acogida entre los desarrolladores:

1. *Android* (77%): Propiedad de *Google*, de plataforma abierta, basado en *Linux*
2. *IOS* (66%): Propiedad de *Apple*, inicialmente de plataforma cerrada
3. *Windows Phone* (37%): Propiedad de *Microsoft*, enfocado en el consume empresarial
4. *Blackberry OS* (34%): Propiedad de *Research in Motion (RIM)*, exclusivo para dispositivos *Blackberry*.

1.2.3. Otras metodologías propuestas.

A continuación se estudian dos metodologías adicionales para el desarrollo de aplicaciones móviles.

1.2.3.1 Metodología MDAM. La metodología propuesta por Gasca, Camargo y Medina (2014), incorpora elementos de metodologías ágiles para el desarrollo de software y está constituida por cinco etapas: análisis, diseño, desarrollo, prueba y entrega. La principal motivación de los autores para plantear esta metodología, es responder a los desafíos que deben enfrentar los proyectos de desarrollo de aplicaciones móviles, tales como las limitaciones en los canales de comunicación, la heterogeneidad de los dispositivos, el reducido tamaño de las pantallas y teclados, la movilidad de los usuarios, el tiempo esperado de salida al mercado versus el tiempo de desarrollo entre otros.

El manejo adecuado de las características anteriores no garantiza el éxito de un proyecto, a esto debemos sumarle el conocimiento electrónico y de comunicaciones por parte del equipo de

desarrollo, el entorno de ejecución de la aplicación y a nivel estratégico, el conocimiento del cliente que utilizará la aplicación. En el caso del presente proyecto, son usuarios residentes casi en su totalidad en el Departamento de Tolima, con empleos formales y con grupo familiar.

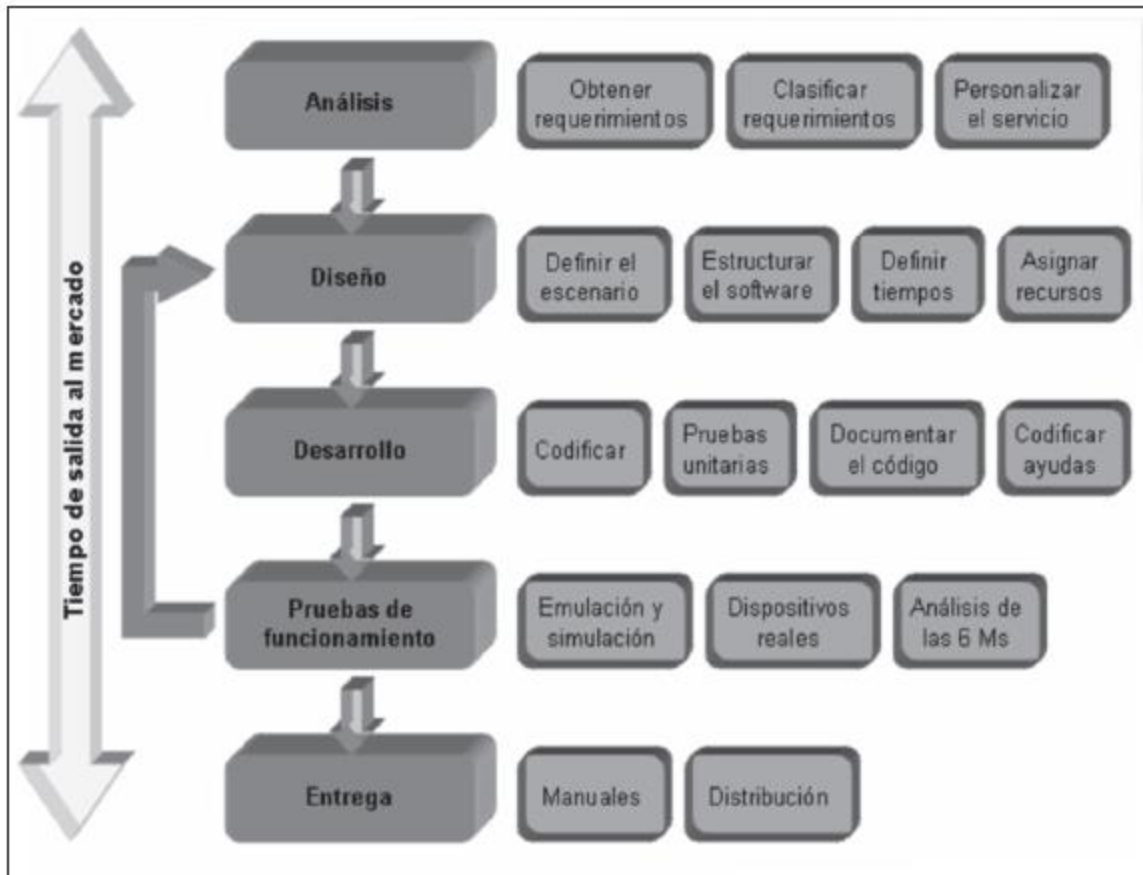
La metodología se denomina “MDAM” (Metodología para el Desarrollo de Aplicaciones Móviles), la cual se aplica en el desarrollo de una aplicación móvil orientada a servicios de salud. Los principios que hereda de las metodologías ágiles son los siguientes:

- Desarrollar software que funciona, más que conseguir buena documentación.
- La respuesta ante el cambio es más importante que seguir un plan.
- Colaboración con el cliente sobre negociación contractual.
- Individuos e interacciones sobre procesos y herramientas.

Se aplica a su vez el concepto de las “Seis M”, el cual hace énfasis en que las aplicaciones móviles deben satisfacer las expectativas del cliente y a la vez ganar dinero. El éxito de este cometido se logra bajo la medición de seis atributos a saber: *Movement* (movimiento), *Moment* (momento), *Me* (yo), *Multi-user* (Multiusuario), *Money* (dinero) y *Machines* (Máquinas).

Las etapas de la metodología son las siguientes:

Figura 1. Etapas de la metodología MDAM. (Gasca et ál., 2014).



Las fases de la metodología son muy similares a las de un desarrollo de software convencional, sin embargo existen algunos aspectos diferenciadores que focalizan el proceso hacia el entorno de las aplicaciones móviles, a saber:

Fase de análisis. En esta fase los requerimientos se clasifican en “Entorno”, “Mundo”, “Funcionales” y “No funcionales”. El concepto de mundo hace referencia a cómo interactúa el usuario con la aplicación, mediante la GUI, los datos de salida y toda la interacción hombre-máquina, considerando las limitaciones de los dispositivos móviles. En lo referente a requerimientos no funcionales, estos abarcan aspectos como estabilidad, portabilidad, rendimiento, tiempo de salida al mercado (aspecto fundamental en el mercado de aplicaciones móviles) y costos. También se realizan actividades de “personalización del servicio”, orientadas

al conocimiento del cliente que va a utilizar la aplicación, buscando la aceptación del servicio (Gasca et ál., 2014).

Fase de diseño. Se hace una “definición del escenario”, en la cual se tienen en cuenta los diferentes escenarios de ejecución de la aplicación, la sincronización con el servidor, los modos de ejecución “desconectado”, “semi-conectado” y “conectado” y cuál debe ser la respuesta de la aplicación a cada uno de dichos estados (Gasca et ál., 2014).

Fase de desarrollo. Se aplican pruebas unitarias para cada uno de los elementos que intervienen en la aplicación, luego se pone en funcionamiento el conjunto de elementos, se prueban y se documentan los resultados (Gasca et ál., 2014).

Fase de pruebas de funcionamiento. Se hacen pruebas de emulación y simulación y en dispositivos reales (Gasca et ál., 2014).

Resultados de la utilización de la Metodología: se desarrolló un proyecto denominado “DrMovil”, orientado al monitoreo de pacientes con problemas de Glucosa e Hipertensión Arterial, en un tiempo de cuatro meses y con una respuesta satisfactoria por parte de los usuarios.

- Se generaron versiones para los sistemas operativos *Android*, *Symbian*, *OS7 (BlackBerry)* y una versión *web* donde se incorpora la lógica de la aplicación.
- Se realizaron jornadas de utilización de la aplicación por parte de grupos de estudiantes de la Universidad del Magdalena, donde se tomaron muestras de glucosa y presión arterial y las cargaron en las aplicaciones.
- En los dispositivos con Sistema Operativo *Windows Phone*, no pudieron instalar ni utilizar el servicio.
- Se realizó el análisis de las Seis M, los resultados fueron los siguientes:

Tabla 2 Análisis 6M para modelo MDAM. (Gasca et ál., 2014).

Atributo	Definición	Calificación	Justificación
Momento	Un servicio que cuente con este atributo debe estar disponible en cualquier instante de tiempo en que el usuario desee usar dicho servicio.	4	El paciente puede ingresar la medida en cualquier momento, pero la respuesta del diagnóstico depende del horario de atención del médico.
Movilidad	Un servicio móvil debe ser “móvil” por naturaleza, la ubicación debe ser una parte integral del servicio.	4	El usuario se puede desplazar a cualquier lugar y realizar sus registros, siempre y cuando tenga cobertura del operador. La atención médica se ajusta al área de cobertura del operador.
Dinero	Como cualquier acción comercial, un servicio móvil tiene un fin lucrativo, ya sea para el operador, para el proveedor del servicio o para el usuario.	5	Aunque la aplicación es de descarga gratuita, ésta genera ingresos al operador al ocasionar tráfico en la red de datos. La entidad prestadora de servicios de salud (EPS) reduce costos por la atención de pacientes, lo que permite redirigir los ingresos a casos de mayor relevancia. El paciente reduce los costos de traslado a la EPS.
Yo	Se refiere al nivel de personalización de un servicio.	3	El servicio presenta un grado de personalización porque permite que el usuario seleccione: el médico, el modo de ver los registros almacenados y el tipo de conexión para enviar los datos.
Máquina	La tecnología (terminal o redes) siempre es el factor que posibilita o limita; el atributo máquina busca añadir potencia a los dispositivos de última generación que cada vez tienen mayores prestaciones a nivel de hardware y software.	4	El servicio solo puede ser soportado por celulares de gama media que posean Java o Android, y conexión a la red de datos. El usuario puede hacer un buen uso de la aplicación sin importar la resolución de la pantalla y el tipo del teclado.
Multiusuario	Busca extenderse dentro de la comunidad, que el servicio sea interactivo y que pueda utilizarse por múltiples usuarios de manera simultánea.	1	Aunque la aplicación permite una comunicación en dos direcciones, no permite interactuar con la persona en el otro extremo del sistema.

1.2.3.2. Metodología de Adaptación de la información al dispositivo. En su artículo, (Aragón et ál., 2009) presentan un estado del arte sobre la adaptación de la información con respecto al contenido y al despliegue en el dispositivo de acceso, sobre el cual se propone un modelo de adaptación denominado Modelo de Adaptación Integrado “IAM”, por su sigla en

inglés. Para esto se hizo un estudio de trabajos relacionados donde se analizan diversas características que se consideran para adaptar la información al dispositivo donde se va a desplegar. En el contexto de los dispositivos de cómputo, usualmente la información se entrega de manera general, sin tener en cuenta la heterogeneidad de los usuarios ni su interacción con el sistema, por lo tanto se corre el riesgo de entregar información que no corresponde a las expectativas, preferencias y contexto de uso y por ende no es útil. Una estrategia para contrarrestar esta problemática es la “personalización” del cliente, es decir conocer a fondo su entorno, características, cultura, historia, localización, creando así perfiles de usuario agrupados por contexto (Aragón et ál., 2009).

El objeto es ofrecer al usuario la información que éste busca teniendo en cuenta tres aspectos: i. El perfil del usuario, ii. El perfil del dispositivo, iii. La localización y el momento en que se realiza la consulta. Uno de los aspectos que debe evitarse es la sobrecarga de información, que hace que el usuario no sepa cómo manejar la información y por lo tanto no le interese. Es necesario entender la heterogeneidad de los usuarios y focalizar adecuadamente el objetivo de la aplicación.

Para lograr una adaptación adecuada de la información, es necesario tener en cuenta que existen diferentes dispositivos de acceso, tales como computadoras de escritorio, portátiles, teléfonos celulares, los cuales son diferentes tanto en hardware como en software, por lo tanto deben considerarse las restricciones de despliegue de los dispositivos y adaptar la información a éstas.

Aspectos del perfil de usuario heterogéneo:

- Datos personales, intereses, calidad de los datos esperados, seguridad.

- El momento en el cual el usuario interactúa con la aplicación, es decir el lugar o la actividad que está realizando en un momento específico.

Aspectos del perfil de dispositivo heterogéneo:

- Un usuario puede realizar una consulta de grandes volúmenes de datos, los cuales no siempre están en capacidad de desplegarse en su dispositivo de acceso.
- Memoria del dispositivo, velocidad del procesamiento, dispositivos de entrada y salida, Sistema operativo, características de red.
- Navegador, otras aplicaciones en ejecución.

Aspectos del contexto a tener en cuenta para el despliegue de la información:

- Localización, actividades realizadas, condiciones ambientales, hora, preferencias de usuario
- Información que no debe ser mostrada a algunos usuarios, plazo máximo de inicio y terminación de una actividad.

1.2.4. Revisión de un Proyecto de desarrollo de aplicaciones móviles. En su artículo, (Gama et ál., 2014), plantean el diseño de una plataforma de aplicaciones para la emisión de documentos académicos de manera electrónica a través de la computación móvil, para uso de alumnos de la institución educativa Instituto Tecnológico de Tlajomulco (México). El proyecto se denomina: Servicio para Emisión de Documentos Electrónicos “SEDE”.

En dicha plataforma se pueden generar y constancias de inscripción, créditos cursados, promedio general. La solicitud de puede hacer mediante SMS y a través de una aplicación móvil diseñada para ambientes *Android* e *IOS* conectada a un *Webservice*. Los documentos se

generarán en formato *PDF* y son enviados al correo electrónico del estudiante. El costo de servicio se cobrará en la línea telefónica y también se podrá cancelar a través de *PayPal*.

La necesidad de este proyecto surgió por la siguiente problemática:

- Para la generación y entrega de documentos académico impreso, se requiere la presencia física del interesado, lo cual genera grandes filas en las instalaciones de la universidad.
- El pago de dichos documentos por parte del estudiante, requiere desplazamiento hasta la tesorería de la Universidad o hasta una sucursal bancaria.
- La limitación de horario de los funcionarios operativos de la universidad.

Los beneficios que plantea la realización de este proyecto son:

- Para la universidad, reducir los costos operativos que generan las filas de estudiantes, mejorar los tiempos de respuesta.
- Para los alumnos, poder generar un certificado a cualquier hora y desde cualquier lugar, mediante el uso de un dispositivo móvil.
- Para el medio ambiente, reducir el uso de papel.

Algunos ejemplos de plataformas similares en México son:

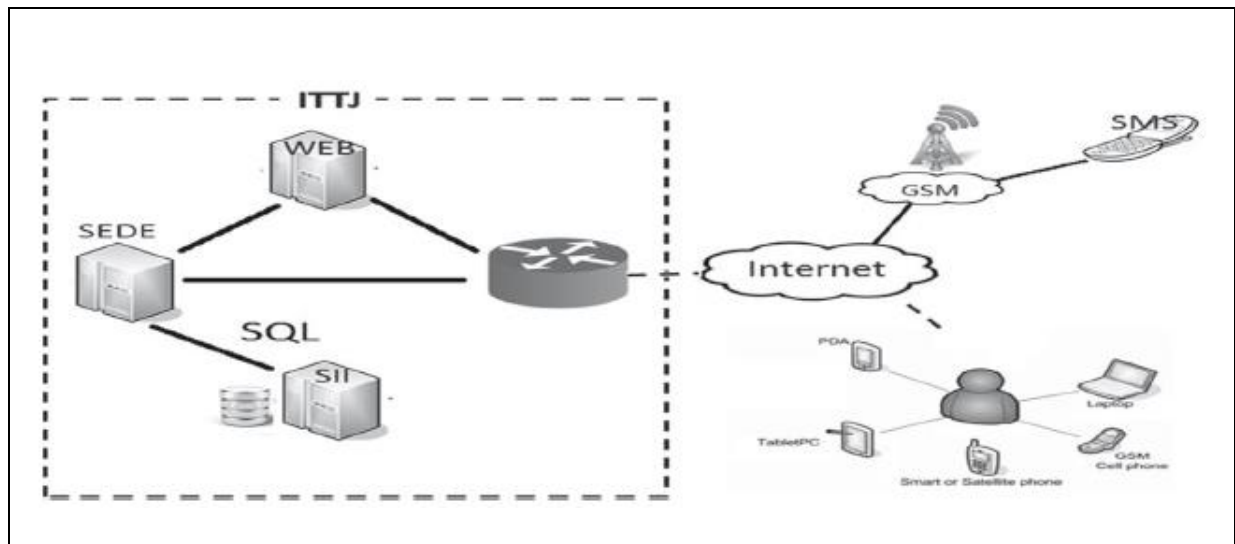
- Sistema Integral de Información y Administración Universitaria “SIIAU”: creado por la Universidad de Guadalajara, Jalisco, opera en los procesos de administración de recursos y gestión académica. El software genera el documento la ficha de depósito para que el estudiante la descargue, se desplace al banco a hacer el respectivo pago y luego regrese por

el documento físico, pero no cuenta con alternativas de pago vía web ni con envío del documento al correo electrónico.

- Sistema de Control Escolar Universidad de Colima “SICEUC”: Creado por la Universidad de Colima, cuenta con módulos informativos, de pre-inscripción, verificación de pagos, pero no genera documentos académicos.
- Alexia: Creada por la Agencia de Certificación en Innovación Española, integra el área de gestión académica y administrativa y la comunicación entre la institución y la familia.

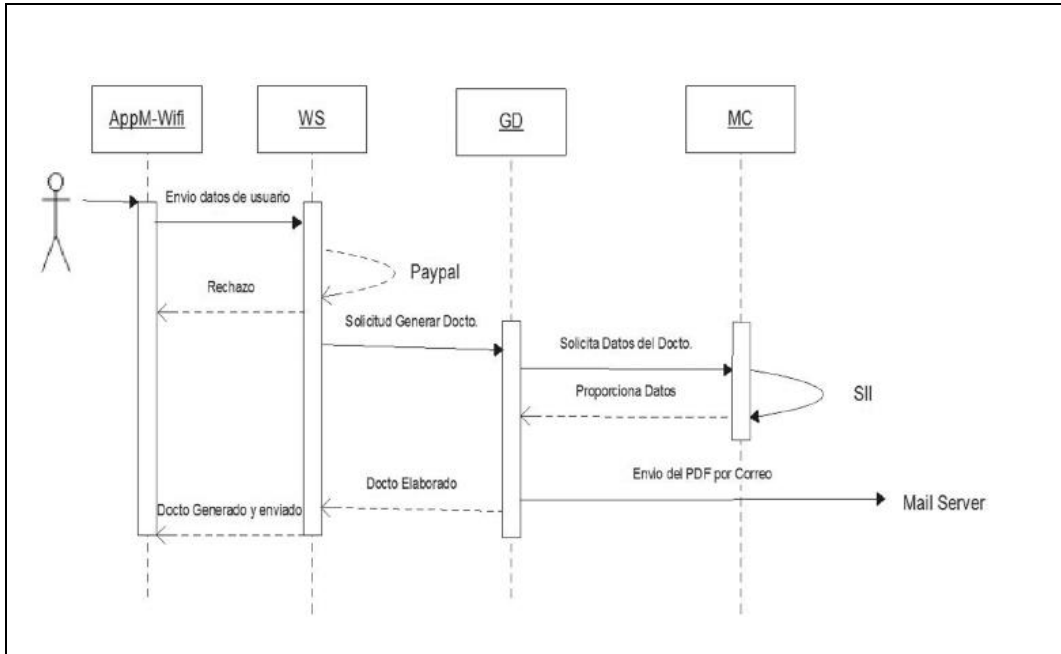
El proyecto SEDE busca mejorar el servicio de solicitud y emisión de documentos académicos de manera electrónica. A continuación se puede observar la arquitectura propuesta:

Figura 2. Arquitectura proyecto SEDE. (Gama et ál., 2014).



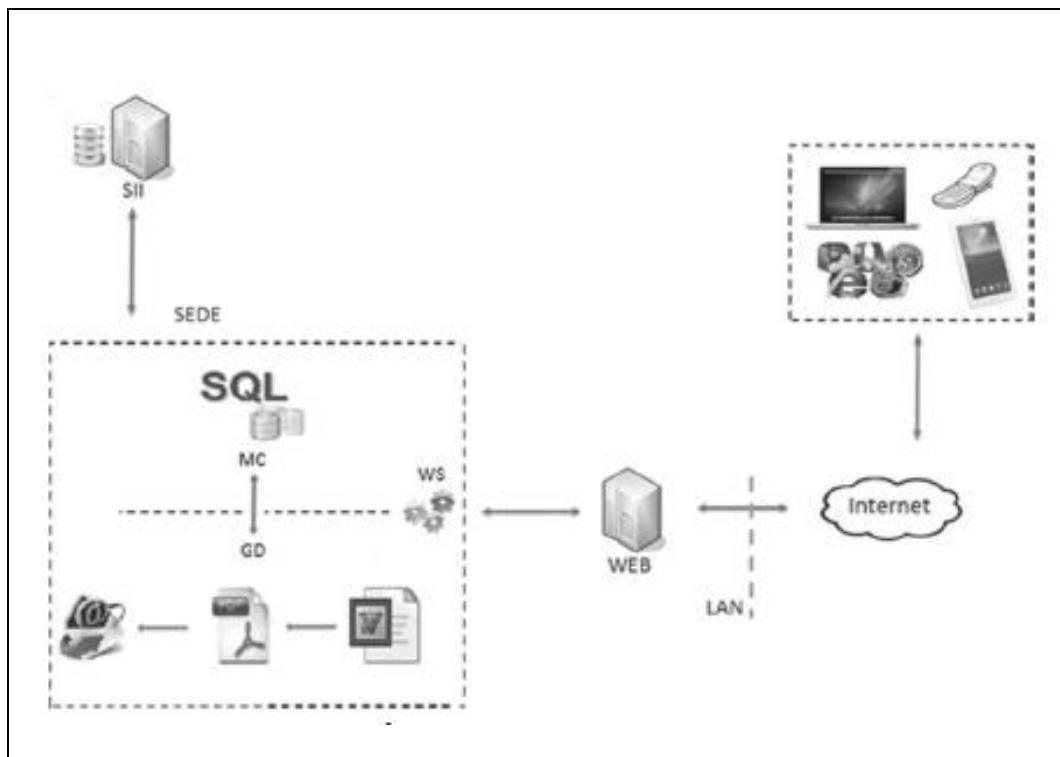
El diagrama de secuencia propuesto para la generación de documentos desde la aplicación móvil a través de SMS es el siguiente:

Figura 3. Diagrama de secuencia en aplicación móvil mediante SMS. (Gama et ál., 2014).



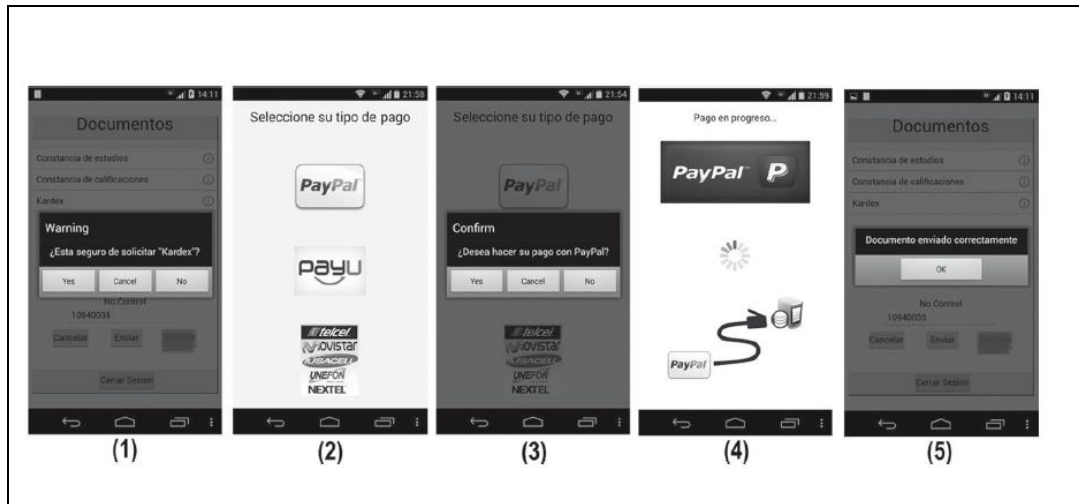
El módulo de consulta y generación de documentos propuestos es el siguiente:

Figura 4. Módulo de consulta y generación de documentos. (Gama et ál., 2014).



El paso a paso desde la solicitud del documento hasta el envío del mismo es el siguiente:

Figura 5. Solicitud de documento desde dispositivo móvil. (Gama et ál., 2014).



1.3. Justificación

Desde el año 2010, Colombia ha mostrado un notable nivel de penetración de la telefonía móvil, culminando el segundo trimestre del año 2015 con un total de 55.982.456 abonados telefónicos y un índice de penetración del 116.1%. (MinTic, 2016a). Si se compara esta cifra de abonados con los resultados arrojados por el último censo poblacional realizado por el Dane en el País, según el cual hay una población de 48.974.540 (Dane, 2016). Se puede afirmar que hay más teléfonos celulares que habitantes en Colombia. En el tema de acceso a Internet, el gran número de teléfonos celulares existentes en Colombia influye en el modo de acceso a la red por parte de los usuarios. De un total de 11.819.265 conexiones (penetración del 24.5%), el 54.72% corresponde a Internet Móvil con un total de 6.468.695, mientras que el 45.27% de ellas corresponden conexiones a Internet Fijo con un total de 5.350.700 (MinTic, 2016b). A nivel de Sistemas Operativos, a septiembre de 2015 *Android* representaba el 86.6% del mercado en el país, mientras que sistema operativo *IOS* solo representaba el 7.9%. El 28.6% restante se reparte entre *Windows Phone* y otros sistemas operativos (ComScore, 2016).

En Comfatolima, de 51.669 afiliados a corte 4 de octubre de 2016, un total de 38.979 tenía registrado un número de teléfono celular en la base de datos de la Caja, con esta información se procedió a realizar una encuesta para determinar el acceso de los afiliados a teléfonos inteligentes y a Internet móvil, arrojando los siguientes resultados:

Tamaño de la muestra	400
Contactos fallidos	20.25% (81)
Tienen teléfono inteligente	61% (244)
No tienen teléfono inteligente	18.75% (75)

De los 244 afiliados encuestados que tienen teléfono inteligente, 174 de ellos tienen plan de datos, lo que equivale al 71.31%.

Los 81 contactos fallidos en la muestra evidencian la necesidad de actualizar la información de contacto de los afiliados, lo cual hace parte de las utilidades que tendrá la aplicación móvil. Para mayores detalles sobre la encuesta, ver el numeral 3.3, referente a la recolección de información.

En este orden de ideas, se puede afirmar que las condiciones de conectividad y acceso a dispositivos están dadas para la realización del proyecto, toda vez que la telefonía móvil y el acceso a internet son variables no controladas pero están adquiriendo relevancia en el tiempo y el entorno.

La competencia directa de Comfatolima en el Departamento del Tolima, es la Caja de Compensación Familiar Comfenalco Tolima, quienes a la fecha de realización del presente documento no cuentan con una aplicación móvil al servicio de los afiliados, por esta razón la Caja estaría dando un paso adelante frente a la competencia al implementar una herramienta tecnológica de estas características, que resultará novedosa para los afiliados a las Cajas de Compensación de la Región.

El horario de atención al público de la Caja es de lunes a viernes de 7:30am a 12:30pm y de 2:00pm a 5:30pm, esto ocasiona que los afiliados deban solicitar permiso en horas laborales para desplazarse a realizar una consulta o trámite. A través de la aplicación móvil los afiliados podrán acceder a su información a cualquier hora y desde cualquier lugar, sin tener que desplazarse físicamente o acceder a una computadora para esto.

El beneficio de la aplicación móvil será más notable si se tiene en cuenta que Comfatolima tiene oficinas en seis municipios del Departamento del Tolima a saber: Ibagué,

Espinal, Líbano, Melgar, Lérída y Mariquita, pero tiene trabajadores afiliados en los 47 municipios del Departamento. Los afiliados en municipios donde no hay oficinas de la Caja, tienen que esperar la visita mensual o bimestral de los asesores comerciales de para formular sus inquietudes o realizar sus trámites. Se abre así la posibilidad de establecer un canal de comunicación más dinámico, aligerando trámites e informando acerca de la actualidad del negocio (MMA, s.f, p.8).

Los costos operativos y tecnológicos que acarreará el funcionamiento del sistema de comunicación móvil para Comfatolima, están compuestos por la adquisición de la licencia de la aplicación, el soporte técnico, la designación inicialmente de un funcionario para la Administración del sistema, el servicio de *backend* en la nube para la gestión de base de datos, seguridad y almacenamiento de los documentos enviados por los usuarios.

No obstante los costos anteriores, la implementación de una aplicación móvil en una organización representa una reducción de costos operativos misionales, debido a que los empleados se vuelven más móviles y por ende, más productivos, ya que cuentan con un canal de comunicación inmediato con sus clientes haciendo los procesos más eficientes (Unipymes, 2015). En Comfatolima esta reducción se materializará en el ahorro de tiempo, tanto en las llamadas de atención al cliente como en la elaboración y envío de comunicados físicos.

Según MMA (s.f) las aplicaciones móviles suelen convertirse en importantes herramientas de gestión de la calidad, control de procesos, control del equipo humano, gestión del conocimiento, permitiendo a las organizaciones realizar sus actividades de manera más rentable y eficaz, así como agregar valor al cliente.

En términos de fidelización, una aplicación móvil permite acceder a clientes diferenciados y establecer comunicación directa con ellos, estableciendo un vínculo a muy bajo

costo. El estar presente en su teléfono y agregarle valor al mismo, hace que “se fomente la intensificación de la relación entre el cliente y la marca, que suele tener como consecuencia el aumento de la frecuencia de compra y/o el aumento del valor medio de la misma” (MMA, s.f, p.9). Por otro lado la aplicación móvil será vista por personas que visitan las tiendas de aplicaciones, lo cual ocasionará que la marca Comfatolima aparezca entre las búsquedas realizadas por dispositivos móviles, contribuyendo al posicionamiento de la imagen corporativa.

Un caso de éxito en el tema de las aplicaciones móviles es la Nueva EPS, quienes implementaron una aplicación móvil para sus afiliados orientada a trámites. En los primeros seis meses de operación, se registraron 18.000 usuarios y un total de 70.745 visitas y trámites en ese mismo lapso de tiempo (Nueva Eps, 2014).

1.4. Formulación del problema

La pregunta de investigación que intenta resolver el proyecto es: “¿Cómo fortalecer el proceso de Gestión de la Parafiscalidad de Comfatolima, mediante el desarrollo de una aplicación móvil, que le permita al afiliado realizar en línea trámites que la infraestructura tecnológica actual no satisface?”

1.5. Objetivos

1.5.1. Objetivo general. Desarrollar una aplicación móvil que le permita al afiliado a Comfatolima realizar en línea consultas y trámites inherentes al proceso de Gestión de la Parafiscalidad.

1.5.2 Objetivos específicos.

1. Diseñar la arquitectura del sistema, que permita operar la aplicación desde dispositivos móviles con sistema operativo android.
2. Desarrollar un subsistema de consultas para el afiliado en lo referente a datos de afiliación, pago y redención de la cuota monetaria y pago de aportes del empleador.
3. Desarrollar un subsistema de envío y recepción de mensajes entre Comfatolima y los afiliados, con la opción para el usuario de enviar certificados de escolaridad y supervivencia de sus beneficiarios.
4. Desarrollar un subsistema de administración de la aplicación móvil en entorno web, para ser operado por la Oficina de Subsidio y Aportes.

1.6. Alcances y limitaciones

- La aplicación móvil se desarrollará para Sistema Operativo *Android*, obedeciendo a las cifras mostradas en el numeral 1.2 (antecedentes), según las cuales éste es el Sistema Operativo más utilizado por la población. En etapas posteriores se generarán versiones de la aplicación compatibles con otros sistemas operativos.
- La aplicación apoyará exclusivamente al proceso de pago de cuota monetaria, ejecutado por la Oficina de Subsidio y Aportes de Comfatolima, en etapas posteriores se considerará extenderla a otras áreas.
- El afiliado no realizará pagos mediante el aplicativo móvil, ya que no se enfoca en la compra de servicios de la Caja.
- El aplicativo móvil sólo prestará sus servicios a los trabajadores afiliados, no a las empresas donde ellos laboran, las cuales también tienen la calidad de afiliadas a Comfatolima.

1.7. Resultado esperado

- Diseño arquitectónico de la aplicación
- Aplicación móvil como apoyo al proceso de pago de cuota monetaria de Comfatolima, compuesta por los siguientes subsistemas:
 - *Backend* de la aplicación instalado en un *Webservice*, como puente de comunicación entre el sistema de información de subsidio y aportes de Comfatolima y la aplicación móvil
 - Subsistema de consulta y actualización de datos básicos del afiliado, con la opción de generar certificados de afiliación
 - Subsistema bidireccional de comunicación, con la opción de adjuntar archivos en los mensajes enviados por el usuario (para envío de documentos soportes)
 - Subsistema de consulta de histórico de pagos de subsidio, pagos de aportes, movimientos de tarjeta y datos de la tarjeta Comfatolima
 - Subsistema de administración de la aplicación móvil en entorno web, para la gestión de la aplicación en lo referente a recepción de solicitudes, notificación al afiliado de la gestión realizada y envío de mensajes masivos y personalizados
- Documentación referente a las pruebas realizadas al software
- Resultados e interpretación de las mediciones obtenidas mediante la aplicación del *proceso personal de software PSP0 y PSP1* al proyecto
- Manual técnico, compuesto por la documentación del diseño de la aplicación, código fuente y especificación de las pruebas realizadas durante el proyecto y sus resultados
- Manuales de usuario de la aplicación y del subsistema de administración

2. Estrategia Metodológica

2.1. Metodología

2.1.1. Modelo de proceso incremental. Se propone aplicar el modelo enunciado por Pressman (2010) denominado “Modelo de proceso incremental” (p.35). Esta decisión se basa en el estudio de antecedentes, en el cual se establece que los proyectos de desarrollo de aplicaciones móviles deben ser abordados por metodologías ágiles, las cuales permitan generar incrementos funcionales en corto tiempo, reduciendo así el riesgo del proyecto y adaptándose a la frecuente ocurrencia de cambios en los requerimientos.

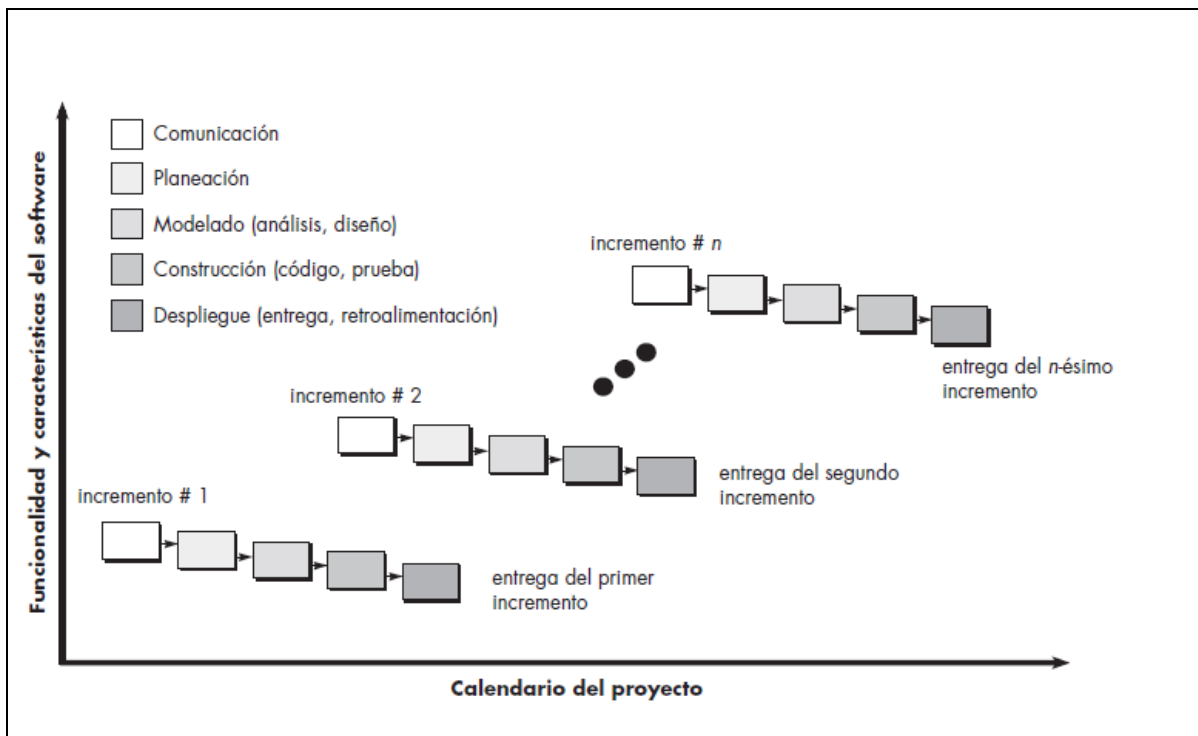
Las características más relevantes de este modelo son:

- Las fases se dividen en incrementos, cada uno de los cuales desarrolla una parte de la funcionalidad y se valida el producto resultante
- Una vez que los productos de una fase se consideran listos, no se modifican en las siguientes fases
- Cada incremento añade una nueva funcionalidad al sistema, por lo que el usuario puede ir utilizando y validando cada funcionalidad antes de terminar el sistema completo
- El sistema se desarrolla como una serie de pasos e iteraciones una vez establecida la arquitectura global
- El usuario tiene contacto con el producto resultante en cada iteración
- Los componentes validados y terminados no se cambian, excepto cuando se encuentra un error o es absolutamente necesario gestionar el cambio en un requerimiento

- Cada iteración se refina lo realizado en la iteración anterior, por lo tanto se van mejorando los productos hasta llegar a un nivel de refinamiento óptimo

A continuación se puede ver el esquema de funcionamiento del modelo incremental.

Figura 6. Modelo incremental. (Pressman, 2010, p.35).



2.1.2. PSP0 y PSP1. Según Pressman (2010), “Todo desarrollador utiliza algún proceso para elaborar software de cómputo. El proceso puede ser caprichoso o ad hoc; quizá cambie a diario; tal vez no sea eficiente, eficaz o incluso no sirva; pero sí existe un proceso” (p.48). PSP pone se enfoca en la medición personal del producto del trabajo y de la calidad del mismo, permitiendo desde el primer momento establecer líneas base para los proyectos.

Una de las razones para aplicar *PSP* en el proyecto, es que según Pressman (2010): “éste enfatiza la necesidad de detectar pronto los errores; para esto se debe ejercer una actividad de evaluación rigurosa ejecutada para todos los productos del trabajo que se

generen” (p.49). Presman manifiesta que “Cuando se introduce el PSP en forma apropiada en los ingenieros de *software*, es significativa la mejora resultante en la productividad de la ingeniería respectiva y en la calidad del *software*” Humphrey y Ferguson (citado en Presman, 2010).

Las fases de *PSP* que se aplicarán en el desarrollo del proyecto son las siguientes:

Tabla 3. Fases PSP a aplicar en el proyecto

No.	Iteración	Fase PSP
1	Diseño de la arquitectura	Estimación
2	Desarrollo sist. consulta y actualización de datos	Planeación
3	Desarrollo sist. comunicación bidireccional (incluido el envío de documentos digitales como adjuntos de un mensaje)	Estimación
4	Desarrollo sist. Administración	Desarrollo(diseño, codificación y pruebas)
		<i>Postmortem</i>

Fuente: Elaboración propia

Nota: En cada iteración se diseñará la porción del *Backend* que corresponda.

A continuación se muestra en forma detallada el paso de paso del proyecto, haciendo énfasis en lo siguiente:

- Para cada iteración se utilizarán solamente aquellos elementos metodológicos que se estimen convenientes para garantizar el éxito de la fase.

- Todos los diagramas y conceptos que se generen, serán validados por el Asesor del proyecto y funcionarios de Comfatolima.

Tabla 4. Metodología a aplicar en el proyecto

METODOLOGÍA DE REALIZACIÓN DEL PROYECTO PARA CADA ITERACIÓN			
Nota: De las estrategias, metodologías y herramientas propuestas, para cada iteración se aplicarán solo las que se consideren necesarias para la respectiva fase.			
No.	Etapas	Estrategias / Metodologías	Entregable
1	Comunicación	- Entrevista con el Coordinador de Subsidio y Aportes y con trabajadores afiliados a Comfatolima	Documento de definición de requerimientos con detalles precisos para desarrolladores y examinadores
2	Planeación (PSP)	- Producir declaración de requisitos - Estimar el tamaño en líneas de código - Estimar el tiempo de desarrollo requerido - Registrar fechas y tamaños estimados en el plan del proyecto	- Requisitos seleccionados - Estimación de defectos - Programa del proyecto - Estimación de tiempos

3.1	Análisis (Booch Et ál., 2000) lo definen como: “Etapas de un sistema que captura los requisitos y el dominio del problema. El análisis se centra en lo que hay que hacer, mientras que el diseño se centra en cómo hacerlo” (p.129).	Se realizarán reuniones con el Coordinador de Subsidio y Aportes de Comfatolima, en las cuales se llevarán a cabo las siguientes actividades (mediante análisis orientado a objetos): - Establecer las necesidades básicas y los alcances del aplicativo - Identificación de los elementos del proceso - Establecer las funcionalidades, usabilidad, reportes, interfaces gráficas, generando los respectivos prototipos que respondan a las necesidades de los afiliados a Comfatolima - Establecer modos de operación (<i>stand alone / on line</i>) - Elaborar diagramas de casos de uso - Planeación y priorización de las actividades a desarrollar - Realizar actas de las reuniones de trabajo	- Documentos de análisis - Requerimientos funcionales y no funcionales - Diagramas de casos de uso y escenarios - Prototipos de interfaces y reportes
3.2	Diseño	A partir de la vista de casos de uso, se	Diagramas de:

	Según (Larman, 2003): “El diseño pone de relieve una solución lógica: cómo el sistema cumple con los requerimientos (...) se procura definir los objetos lógicos del <i>software</i> que finalmente serán implementados en un lenguaje de programación orientado a objetos” (p.6).	realizará el modelado de la arquitectura del sistema, teniendo en cuenta elementos como: funcionamiento, capacidad de crecimiento, rendimiento, topología del sistema, distribución, entrega, instalación del aplicativo (mediante diseño orientado a objetos). Se modelarán los siguientes aspectos: - El vocabulario del espacio del problema y del espacio de la solución - Distribución de los procesos - Componentes y archivos que se utilizarán para ensamblar y hacer disponible el sistema físico - Nodos de la topología de hardware sobre la que se ejecutará el sistema	- Secuencia - Clases - Objetos - Actividades
4	Desarrollo (Pressman, 2010): “Los modelos se traducen a código fuente.” (p.75).	Se realizará el desarrollo de <i>software</i> correspondiente a la aplicación	- Decisiones iniciales de implementación - Implementación de clases de componentes - Implementación de

			comportamiento de las clases
4.2	Prueba	Ver el numeral 3.2 Pruebas	
5	Despliegue (entrega, retroalimentación) Según (Sommerville, 2011): "Se crea la liberación de un producto, se distribuye a los usuarios y se instala en su lugar de trabajo" (p.52).	El despliegue se hará inicialmente en el servidor web de Comfatolima, las pruebas iniciales se realizarán con los trabajadores de la Caja, con el fin de analizar su desempeño en condiciones de operación totalmente reales, para luego extenderlo a los demás trabajadores afiliados.	- <i>Backend</i> disponible mediante <i>webservice</i> - Aplicación disponible para su uso
6	Elaboración del informe final	Con los resultados del procesamiento de datos y las herramientas desarrolladas se elaborará un informe final el cual incluirá metodologías aplicadas y el funcionamiento del modelo.	- Informe final

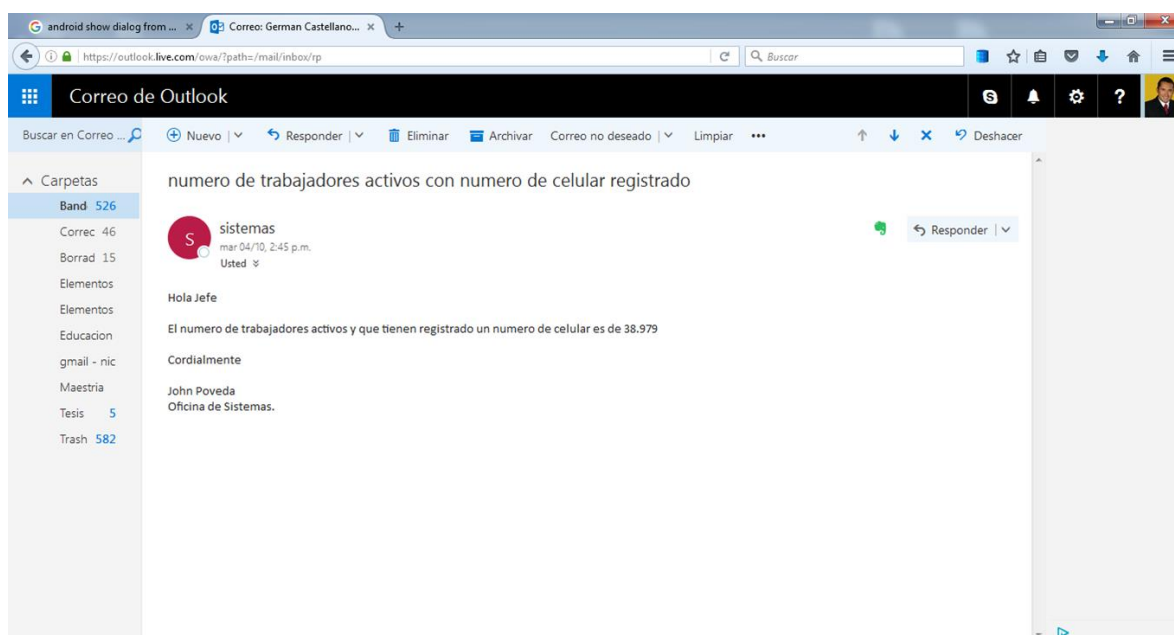
Fuente: Elaboración propia

2.2. Recolección de información

2.2.1. Encuesta. Con el objetivo de establecer el impacto de la implementación del presente proyecto, se procedió a diseñar una encuesta para ser aplicada entre los afiliados a la Caja.

2.2.1.1. Tamaño de la muestra. El día 4 de octubre de 2016, la oficina de sistemas de Comfatolima informa que el número de afiliados que tienen registrado un teléfono celular en la base de datos de subsidio, es de 38.979.

Figura 7. Notificación número de afiliados a Comfatolima con teléfono celular registrado.



Fuente: Elaboración propia

Para calcular el tamaño de la muestra, se tomaron como base los conceptos planteados en (Hernández et al., 2010). El cálculo se realizó mediante la herramienta “Decision Analyst STATS 2.0”, que viene en el libro en referencia.

Como criterio de selección de afiliados a encuestar, se tomó el número de trabajadores con teléfono celular, independientemente si reciben subsidio familiar o no, debido al nivel de desactualización de la base de datos de Subsidio Comfatolima, teniendo en cuenta que las llamadas se deberán realizar en horas hábiles, donde los trabajadores no se encuentran en sus domicilios y en las empresas es muy complicado llamar al teléfono que aparece registrado en la afiliación de las empresas y que comuniquen al trabajador. También es una oportunidad para que el área de Telemercadeo de Comfatolima revise cuales números celulares están actualizados y corresponden a los trabajadores afiliados. Datos del cálculo (Hernández et al., 2010, p. 179):

Tamaño del Universo	:	38.979
Error máximo aceptable	:	5%
Porcentaje estimado	:	50%
Nivel deseado de confianza	:	95%
Tamaño de la muestra	:	380 encuestas

2.2.1.2. Herramienta de recolección de información. Para establecer el nivel de uso de teléfonos inteligentes por parte de los afiliados a Comfatolima, cuántos de ellos tienen conexión a Internet desde sus dispositivos móviles y el nivel de experiencia como usuarios, se diseñó la siguiente encuesta para ser aplicada telefónicamente a cuatrocientos afiliados seleccionados al azar:

ENCUESTA AFILIADOS COMFATOLIMA

- a. Cuenta usted con teléfono Inteligente (Smartphone)? SI () NO ()

(Responder negativamente la pregunta anterior es excluyente para las siguientes preguntas)

- b.Cuál es la plataforma de su teléfono móvil

Android ()

Iphone ()

Blackberry ()

Windows Phone ()

Symbian ()

Lo desconozco ()

- c. Cuenta con acceso a Internet desde su teléfono a través de un plan de datos o mediante conexión a Internet WI-FI? SI () NO ()

(Responder negativamente la pregunta anterior es excluyente para las siguientes preguntas)

- d. Considera útil contar con una aplicación móvil que le permita consultar y actualizar sus datos de afiliación, sus pagos, sus movimientos de tarjeta, comunicarse con la Caja y recibir información de utilidad para los afiliados?

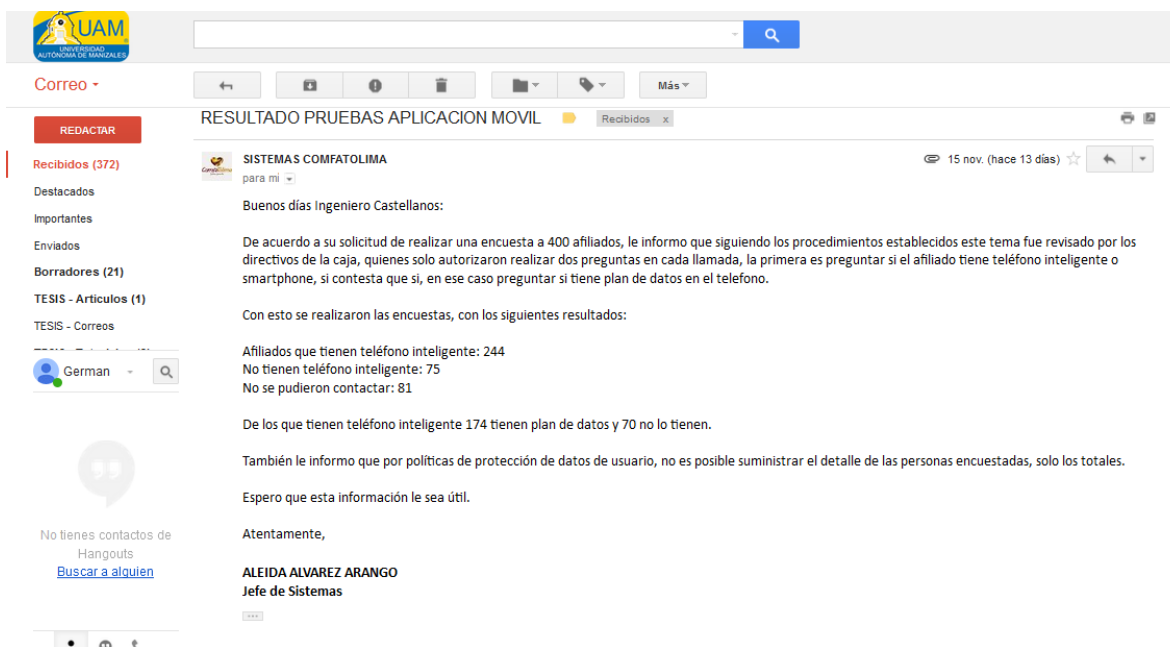
SI () NO ()

- e. Qué información, servicios o trámites ante Comfatolima le gustaría realizar o consultar mediante una aplicación móvil?
-

Con la información anteriormente citada, se solicitó a la Oficina de Sistemas de Comfatolima la realización de cuatrocientas encuestas.

2.2.1.3 Resultados de la encuesta. El día 15 de noviembre de 2016, la Oficina de Sistemas de Comfatolima informa que cumpliendo las políticas establecidas por la Caja, solicitaron autorización a las Directivas para realizar la encuesta, pero solo se autorizó realizar dos preguntas por cada llamada, para establecer si el afiliado tiene teléfono inteligente y si tiene plan de datos.

Figura 8. Notificación resultado encuestas



Fuente: Elaboración propia

Los resultados arrojados por la encuesta fueron los siguientes:

Tabla 5.Resultados encuesta Comfatolima

Total encuestas realizadas	400	100,00%
No se pudieron contactar	81	20,25%
No tienen teléfono inteligente	75	18,75%
Tienen teléfono inteligente	244	61,00%

Fuente: Elaboración propia

Referente a los afiliados que no pudieron ser contactados, es evidente que existe desactualización de la base de datos de Subsidio, por lo tanto se desconoce si poseen teléfono inteligente y plan de datos, lo cual podría aumentar el número de personas que utilizarán la aplicación móvil.

De los encuestados que tienen teléfono inteligente, el 71,31% tienen plan de datos, los resultados fueron:

Tabla 6.Resultados teléfono inteligente versus plan de datos

Tienen teléfono inteligente	244	100,00%
Tienen plan de datos	174	71,31%
No tienen plan de datos	70	28,69%

Fuente: Elaboración propia

2.3. Pruebas

Se aplicará el siguiente plan de pruebas propuesto por Ramírez (2013) para el desarrollo de aplicaciones móviles:

- Pruebas unitarias (probar de manera desacoplada)
- Pruebas de usabilidad (sencillez con la que se usará el sistema)
- Adaptable (respuesta óptima y ágil a las peticiones del usuario)
- Optimización de recursos
- Accesibilidad (diferentes formas de acceder a la aplicación)

Por cada subsistema se generarán informes de:

- Pruebas unitarias
- Pruebas de usabilidad
- Pruebas de adaptabilidad a diferentes tipos de dispositivos
- Pruebas de optimización
- Pruebas de accesibilidad

Según Pressman (2010): “La meta de probar es encontrar errores, y una buena prueba es aquella que tiene una alta probabilidad de encontrar uno” (p.412).

Para la realización de las diferentes pruebas se involucrarán los siguientes grupos humanos:

- Grupo de prueba compuesto por funcionarios de la Oficina de Sistemas de Comfatolima, para la realización de pruebas unitarias y el testeo general de cada incremento (ver cuadro de pruebas), pruebas de verificación
- Grupo de prueba compuesto por funcionarios de la Oficina de Subsidio y Aportes de Comfatolima, para las pruebas de validación

- Grupo de prueba compuesto por todos los funcionarios de Comfatolima, para pruebas de desempeño de la aplicación

2.4. Presupuesto

Tabla 7. Costos del personal involucrado en el proyecto.

COSTOS DE PERSONAL							
No	Participante	Formación	Función	Dedicación (hrs/sem)	Duración (meses)	Costo mensual	Costo total
1	Jorge Iván Meza M.	Ing, Ms	Director	8	6	400.000	2.400.000
2	Germán Castellanos	Ing, Especialista	Autor Proyecto	20	6	1.200.000	7.200.000
3	Angel Bernate	Ingeniero	Tester funcional	1	4	200.000	800.000
4	Carlos Beltrán	Ingeniero	Tester funcional	1	4	200.000	800.000
5	Luis F. Rojas	Tecnólogo	Tester funcional	1	4	200.000	800.000
6	Por definir	Por definir	Pares acad. (2)	1	2	600.000	1.200.000
					Total	2.800.000	13.200.000

Fuente: Elaboración propia

Tabla 8. Equipos requeridos para desarrollo.

EQUIPOS REQUERIDOS PARA EL DESARROLLO						
No	Justificación	Unidades	Compra	Arriendo	Costo unitario	Costo total
1	Computador	1	X		2.200.000	2.200.000
2	Impresora	1	X		300.000	300.000
				Total	2.500.000	2.500.000

Fuente: Elaboración propia

Tabla 9. Equipos requeridos para pruebas.

EQUIPOS REQUERIDOS PARA PRUEBAS						
No	Justificación	Unidades	Compra	Arriendo	Costo unitario	Costo total
1	Huawei G Play	1	X		600.000	600.000
2	Samsung Galaxy S4	1		X	120.000	120.000
3	Motorola Moto G Plus	1		X	120.000	120.000
4	LG D335	1		X	120.000	120.000
5	Galaxy Tab sm P555m	1		X	120.000	120.000
				Total	1.080.000	1.080.000

Fuente: Elaboración propia

Tabla 10. Insumos y materiales requeridos en el proyecto.

INSUMOS Y MATERIALES					
No	Nombre	Uso	Unds.	Costo unitario	Costo total
1	Insumo impresora	Informes y artículos	1	200.000	200.000
2	Papeleria	Informes, fotocopias	1	100.000	100.000
3	Marcadores, lapiceros	Diagramación	10	3.000	30.000
4	Licencias DBMS	Desarrollo de software	1	-	-
5	Licencias Lenguaje de prog.	Desarrollo de software	1	-	-
6	Licencias IDE's	Desarrollo de software	1	-	-
7	Licencias software pruebas	Desarrollo de software	1	-	-
			Total	303.000	330.000

Fuente: Elaboración propia

Tabla 11. Costos totales del proyecto.

COSTOS TOTALES DEL PROYECTO		
No	Nombre	Total
1	Costos de personal	13.200.000
2	Equipos para desarrollo	2.500.000
3	Equipos para pruebas	1.080.000
4	Insumos y materiales	330.000
Total		17.110.000

Fuente: Elaboración propia

2.5. Cronograma

Tabla 12.Cronograma del proyecto.

No	Actividad	Tiempo																							
		ago-16			sep-16			oct-16			nov-16			dic-16			ene-17			feb-17					
1	Diseño arquitectura																								
2	Desarrollo subsistema de consultas																								
3	Desarrollo subsistema de mensajes																								
4	Desarrollo subsistema de administración																								
5	Elaboración documentación e informe final																								

3. Referente Teórico

3.1. Cajas de Compensación Familiar

Las Cajas de Compensación Familiar hacen parte del denominado “Sistema de la Parafiscalidad” en Colombia, el cual también está constituido por el Servicio Nacional de Aprendizaje “SENA” y el Instituto Colombiano de Bienestar Familiar. Si bien estos dos últimos componentes son del orden Nacional, las Cajas de Compensación son regionales.

Según la Superintendencia del Subsidio Familiar, en Colombia existen cuarenta y tres (43) Cajas de Compensación Familiar, de las cuales solo nueve registran en sus páginas web la implementación de aplicaciones móviles como canal de comunicación para el afiliado (Superintendencia del Subsidio Familiar, s.f.).

A continuación se hace una revisión de las nueve aplicaciones móviles encontradas:

3.1.1. Cafam

Imagen de la app:

Figura 9. Aplicación móvil Cafam



Fuente: Elaboración propia

Apple Store - Iphone:

Puntuación: No registra Información
Descargas: No registra Información

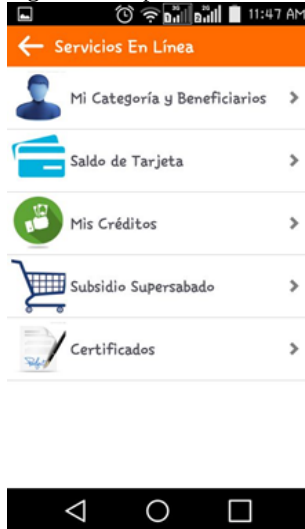
Play Store - Android:

Puntuación: 3.2 (124 votos)
Descargas: De 5,000 a 10,000

3.1.2. Comfaboy.

Imagen de la app:

Figura 10. Aplicación móvil Comfaboy



Fuente: Elaboración propia

Apple Store – Iphone:

Puntuación: No registra Información

Descargas: No registra Información

Play Store – Android:

Puntuación: 0 (0 votos)

Descargas: De 50 a 100

3.1.3. Cajacopi.

Imagen de la app:

Figura 11. Aplicación móvil Cajacopi



Fuente: Elaboración propia

Apple Store – Iphone:

El link en la imagen es inaccesible.

Play Store – Android:

Puntuación: 5 (6 votos)

Descargas: De 10 a 50

3.1.4. Comfamiliar Huila.

Imagen de la app:

Figura 12. Aplicación móvil Comfamiliar Huila



Fuente: Elaboración propia

Apple Store – Iphone:

No registra Información

Play Store – Android:

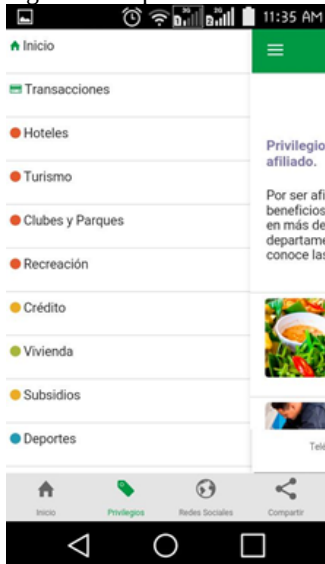
Puntuación: 4.2 (5 votos)

Descargas: De 100 a 500

3.1.5. Comfenalco Antioquia.

Imagen de la app:

Figura 13. Aplicación móvil Comfenalco Antioquia



Fuente: Elaboración propia

Apple Store – Iphone:

No registra Información

Play Store – Android:

Puntuación: 4.7 (71 votos)

Descargas: De 1.000 a 5.000

3.1.6. Comfenalco Valle.

Imagen de la app:

Figura 14. Aplicación móvil Comfenalco Valle



Fuente: Elaboración propia

Apple Store – Iphone:

No registra Información

Play Store – Android:

Puntuación: 3.8 (28 votos)

Descargas: De 1.000 a 5.000

3.1.7. Colsubsidio.

Imagen de la app:

Figura 15. Aplicación móvil Colsubsidio



Fuente: Elaboración propia

Apple Store – Iphone:

No registra Información

Play Store – Android:

Puntuación: 3.1 (492 votos)

Descargas: De 10.000 a 50.000

3.1.8. Confamiliares.

Imagen de la app:

Figura 16. Aplicación móvil Confamiliares



Fuente: Elaboración propia

Apple Store – Iphone:

No registra Información

Play Store – Android:

Puntuación: 5 (1 voto)

Descargas: De 50 a 100

3.1.9. Comfamiliar Risaralda.

Imagen de la app:

Figura 17. Aplicación móvil Comfamiliar Risaralda



Fuente: Elaboración propia

Apple Store – Iphone:

No registra Información

Play Store – Android:

Puntuación: 1 (1 voto)

Descargas: De 50 a 100

Conclusión de la revisión

Todas las aplicaciones de las Cajas de Compensación se centran en publicitar los servicios sociales, pero en la mayoría de éstas no se ha evidenciado que exista un enfoque directo en el proceso de pago de cuota monetaria, tema sobre el cual se centrará el presente proyecto.

Nota: se ha establecido contacto con algunas Cajas de Compensación relacionadas en este numeral, con el fin de obtener información acerca de las experiencias en el proceso de desarrollo de las aplicaciones. Frente a esto se informa que han sido realizadas por terceros, por lo tanto no se ha conseguido información concluyente.

A continuación se relacionan las características más relevantes de las aplicaciones vistas:

Tabla 13. Funcionalidades de las aplicaciones móviles de Cajas de Compensación

No.	Caja de compensación	Funcionalidades
1	Cafam	Portafolio de servicios sociales, convenios, eventos, radio, geo referenciación, directorio, descarga de certificados, gestión de tarjeta, código QR, pagos en línea para el servicio de educación, en el tema de salud gestión de citas y consulta de exámenes médicos
2	Comfaboy	Portafolio de servicios, pago seguridad social, directorio de prestadores de salud, chat, noticias, consulta de categoría y beneficiarios del trabajador, saldo de tarjeta, consulta de créditos, consulta de subsidios y descarga de certificados, mapa con la ubicación de las sedes de la Caja
3	Cajacopi	No suministra información
4	Comfamiliar Huila	Portafolio de los centros recreacionales, contactos
5	Comfenalco Antioquia	Portafolio de los servicios sociales, solicitud de crédito, transacciones en línea, directorio de sedes, consulta de cuota monetaria, directorio de contactos
6	Comfenalco Valle	Portafolio de servicios, calendario de eventos, noticias
7	Colsubsidio	Portafolio de servicios, avances en efectivo, publicidad de convenios
8	Confamiliares	Solicitud de citas médicas, simulador de crédito, portafolio de recreación
9	Comfamiliar Risaralda	Portafolio de programas, mapa de sedes, notificaciones

Fuente: Elaboración propia

Nota: El anexo 5 contiene imágenes de las funcionalidades de las aplicaciones revisadas.

3.2. Conceptos sobre aplicaciones móviles

3.2.1. Definición. Salz & Moranz (2013) “una aplicación móvil es una pieza de *software* específicamente diseñada para ejecutarse en un dispositivo móvil, ya sea un *smartphone* o una *tablet*. La aplicación es usualmente descargada e instalada y una vez allí, trabaja sobre el sistema operativo de éste” (p.14).

Un dispositivo móvil tiene características similares a una computadora portátil, tales como conexión a *internet* y suministro autónomo de energía.

3.2.2. Tipos de aplicaciones móviles. Según IBM (2012, p. 7) las organizaciones que planean implementar una estrategia móvil deben escoger el enfoque de desarrollo para las aplicaciones, teniendo en cuenta aspectos como presupuestos, plazos de entrega, destinatarios del servicio, funcionalidades que debe ofrecer, entre otros. Las aplicaciones móviles se clasifican en tres tipos: Nativas, *Web* e Híbridas.

Cada uno de los enfoques presenta ventajas y desventajas, por lo tanto escoger el enfoque que más se adapte a las necesidades de la organización no es una tarea fácil. A continuación se hace una breve descripción de cada uno de estos enfoques:

3.2.2.1. Aplicaciones nativas. Según IBM (2012, p. 2) están compuestas por archivos ejecutables binarios que se descargan al dispositivo y se almacenan de modo local. Se pueden obtener en las tiendas de aplicaciones. Una vez que se ha instalado la aplicación en el dispositivo, se puede ejecutar como cualquier otro servicio o programa y ésta se conecta

directamente con el sistema operativo sin que intervenga ningún tipo de intermediario. Una ventaja fundamental es que puede acceder libremente a todas las *APIs* que el proveedor del sistema operativo suministre, aprovechando características y funciones únicas de ese sistema en particular.

Aunque el proceso de desarrollo de una aplicación nativa es similar para diferentes sistemas operativos, cada plataforma tiene su *SDK* específico, esto constituye una de las principales desventajas de este tipo de aplicaciones: el código escrito en una plataforma no puede utilizarse en otra, por lo tanto el mantenimiento de estas aplicaciones para los diferentes sistemas operativos puede ser engorroso.

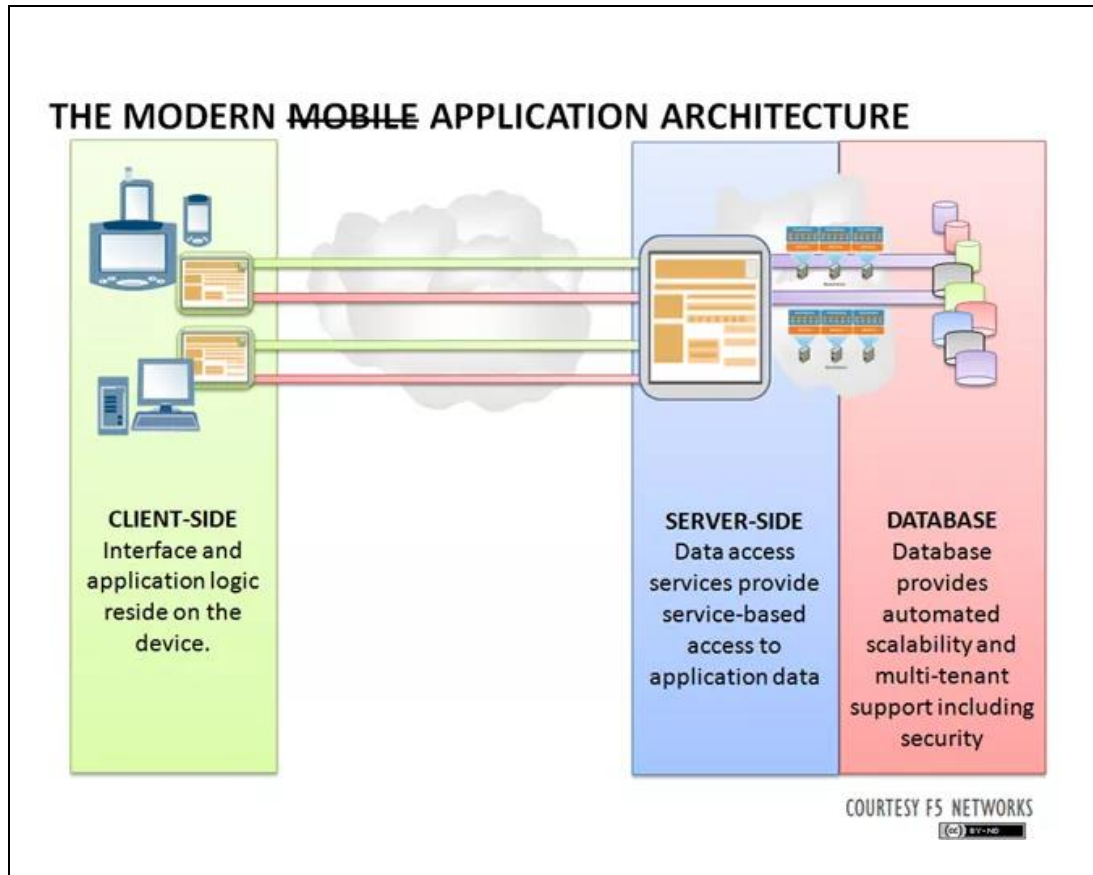
La siguiente es una lista de características de desarrollo de aplicaciones nativas discriminada por sistema operativo:

Tabla 14. Características aplicaciones nativas por sistema operativo. (IBM, 2012, p. 3).

	Apple iOS	Android	Blackberry OS	Windows Phone
Lenguajes	Objective-C, C, C++	Java (algunos C, C++)	Java	C#, VB.NET, etc.
Herramientas	Xcode	Android SDK	BB Java Eclipse Plug-in	Visual Studio, Windows Phone
Formato	.app	.apk	.cod	.xap
Tiendas	Apple App Store	Google Play	Blackberry App World	Windows Phone Marketplace

A continuación se puede apreciar un ejemplo de arquitectura en una aplicación nativa:

Figura 18. Arquitectura de una aplicación nativa.



Fuente: <https://socialinsider.wordpress.com/2011/02/13/comparing-html5-and-native-apps/>

3.2.2.2. Aplicaciones móviles basadas en la Web. Según IBM (2012, p. 4) es un estándar para el desarrollo de aplicaciones complejas basadas en el navegador del dispositivo móvil, aprovechando las funcionalidades de *HTML5*, *CSS3* y *JavaScript*.. Cabe anotar que hay dos enfoques de aplicaciones web:

- Los sitios web optimizados para dispositivos móviles, los cuales al ser accedidos mediante estos equipos, preparan páginas *HTML* adaptadas a las dimensiones y características específicas del dispositivo.

- Los sitios *web* móviles similares a aplicaciones nativas y se ejecutan mediante accesos directos desde el dispositivo.

Una ventaja de este tipo de aplicaciones es su compatibilidad con diferentes plataformas y su reducido costo de desarrollo.

Contrastando lo anterior, las desventajas que presentan este tipo de aplicaciones son las siguientes:

- Las aplicaciones *Web* no se ejecutan directamente en el sistema operativo como sucede con las aplicaciones nativas, su ejecución se realiza a través del navegador *web* de la terminal móvil.
- Aunque el navegador del dispositivo es una aplicación nativa que se ejecuta directamente en el sistema operativo, muchas *APIs* no están expuestas a las aplicaciones que se ejecutan a través del navegador, por lo tanto no se tiene acceso a todas las características del dispositivo.

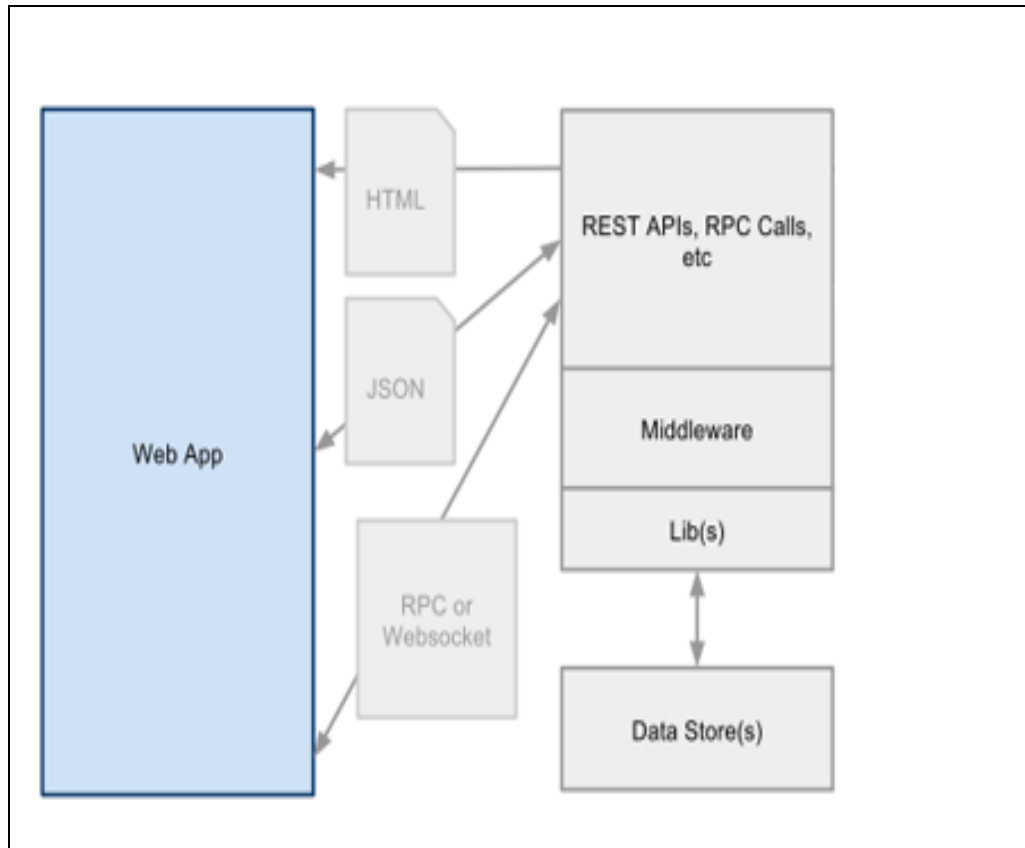
Las siguientes son las características más relevantes de las aplicaciones móviles basadas en la *web*:

Tabla 15. Características de aplicaciones basadas en *web* (IBM, 2012, p. 5).

Característica	Apl. Web solo móviles	Sitios Web solo móviles
Herramientas y conocimientos	Escritas totalmente en HTML, CSS y JavaScript	Escritas totalmente en HTML, CSS y JavaScript
Ejecución	Acceso directo "Instalado", lanzado mediante apl. nativa	Navegando por un sitio mediante URL (Uniform Resource Locator)
Experiencia del usuario	Touch-friendly, interactive UI	IU mediante navegación entre páginas que muestran datos estáticos
Desempeño	IU reside localmente: aplicación con capacidad de respuesta y acceso offline	Todo el código se ejecuta desde un servidor: el rendimiento depende de la red

A continuación se puede apreciar un ejemplo de arquitectura en una aplicación basada en la web:

Figura 19. Arquitectura de una aplicación.



Fuente: web. <http://blog.nodejitsu.com/single-page-apps-with-nodejs/>

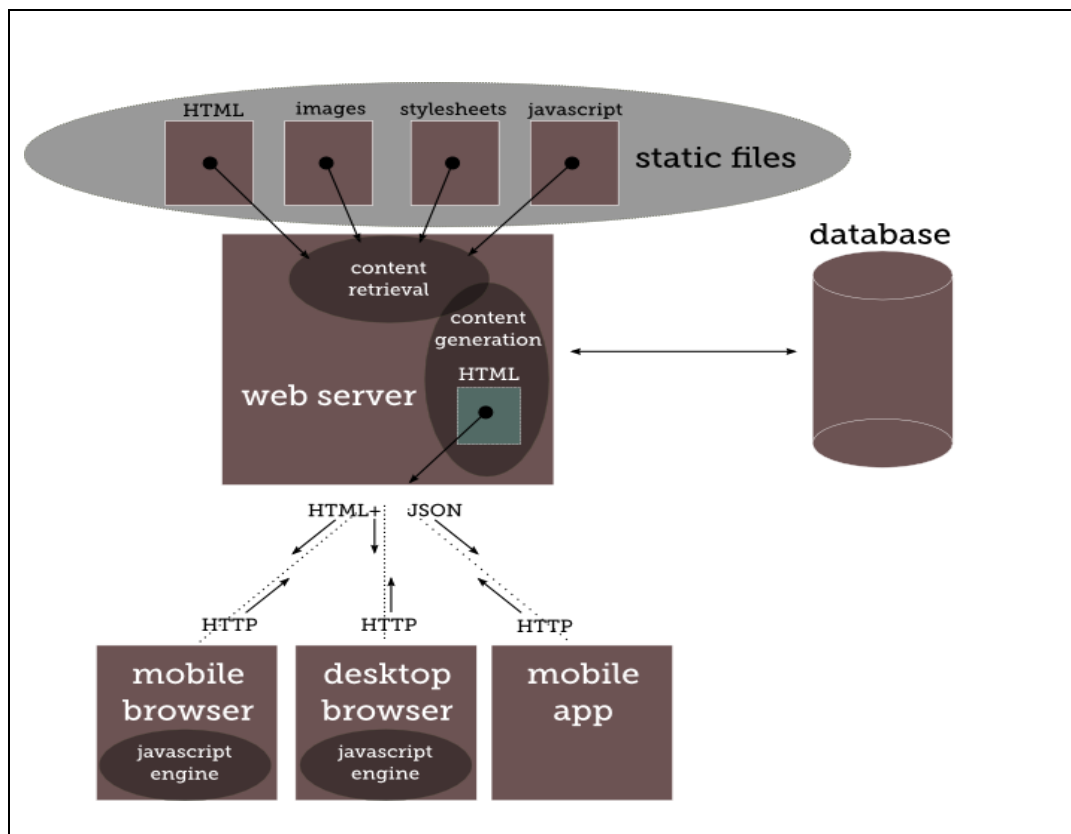
3.2.2.3. Aplicaciones híbridas. Según IBM (2012, p. 6) son la combinación entre el enfoque nativo y la tecnología Web. Con este esquema, los programadores escriben una parte de la aplicación en tecnología web multiplataforma, manteniendo el acceso directo a las APIs cuando se requiera.

Para lograr la comunicación entre el extremo web y las APIs propias del sistema operativo, la parte nativa de la aplicación usa a su vez APIs específicas para crear un motor de búsqueda HTML, aprovechando así todas las características de los dispositivos.

Una de las desventajas de este enfoque es que no cuenta con disponibilidad *offline*, ya que no hay acceso al contenido cuando el dispositivo está desconectado de la red. También sucede que la incorporación del código *HTML* a la aplicación puede mejorar el desempeño, pero no son posibles las actualizaciones remotas.

A continuación se puede apreciar un ejemplo de arquitectura en una aplicación híbrida:

Figura 20. Arquitectura de una aplicación híbrida.



Fuente: <http://mnb.ociweb.com/mnb/MiddlewareNewsBrief-201008.html>

A continuación se muestra un comparativo entre los tres enfoques:

Tabla 16 Comparativo aplicaciones nativas, híbridas y web (IBM, 2012, p. 8).

Característica	Aplicación nativa	Aplicación híbrida	Aplicación Web
Lenguaje de desarrollo	Solo nativo	Nativo y Web o solo nativo	Solo Web
Portabilidad y optimización de código	Bajo	Alto	Alto
Características de acceso específicas del dispositivo	Alto	Mediano	Bajo
Uso de conocimiento existente	Bajo	Alto	Alto
Gráficos avanzados	Alto	Mediano	Mediano
Flexibilidad de actualizaciones	Bajo (Siempre Tiendas)	Mediano (Con frecuencia Tiendas)	Alto
Experiencia de instalación	Alta (A partir de la tienda)	Alta (A partir de la tienda)	Mediana (Mediante navegador móvil)

3.3. Modelos de negocio

3.3.1. Freemium. Según Alcázar (2013) el modelo clásico cuando se trata de cobrar por algo que no nos compran a priori es el *freemium*. Ofreces algo a tus clientes de forma gratuita pero por modelos superiores debes pagar. “En *software* de gestión empresarial, como *Basecamp*, es muy habitual. Tienes unas funcionalidades básicas gratis, pero en el momento que necesitan compartir con clientes, ficheros que te envían casi inmediatamente, tienes que pagar. Aquí lo crítico es conseguir volumen. Tamaño. Por eso se suelen buscar proyectos que tengan alcance global.

Según Maltz & Barney (2012, p. 1) el producto *freemium* perfecto debe “mercadearse” a sí mismo, adquirir nuevos usuarios (tanto gratuitos como de pago), retener a los usuarios, y automatizar el servicio al cliente, todo esto con poca o ninguna intervención humana y un gasto mínimo. Esto sólo es posible con un diseño metodológico que obedezca a lo que los usuarios

realmente quieren. Las Empresas *freemium* típicas convierten entre el 1% y el 10% de los usuarios en posibles clientes de pago.

Como resultado de ello, es importante que un producto *freemium* tenga costos marginales bajos y un mínimo de gastos de comercialización. Debido a que la gran mayoría de los usuarios no pagan, una empresa *freemium* tendrá que luchar si su producto es caro de comercializar. El producto debe venderse y atraer a los usuarios basándose puramente en sus propios méritos y sin un gran presupuesto de *marketing*. Para el usuario, el producto tiene que ser simple. Esto no quiere decir que no debe ser técnicamente complejo o desafiante.

Por otra parte, la sencillez y la calidad del producto debe ser coherente en tanto en su versión gratuita como en la de pago. Muchas empresas *freemium* fallan debido a que proporcionan un producto gratuito que proporciona poco valor, con la esperanza de que los usuarios se convertirán a la versión de pago. Las mejores empresas *freemium* proveen valor real para los usuarios gratuitos y de pago, y pasan la mayor parte de su tiempo mejorando su producto.

En pocas palabras: no se puede atraer a los usuarios o clientes libres pagados a menos que se construya un gran producto. Una vez se ha diseñado y construido un producto de clase mundial, es el momento de darlo a conocer al cliente, para entender si el modelo *freemium* es adecuado para éste.

3.3.2. Suscripción. Una de las ventajas del modelo referenciado por Rojas et ál. (2014, p. 22) es la posibilidad de trabajar con una base de clientes fija en un tiempo concreto y con un flujo de ingresos fijo, debido a que los clientes pagan por adelantado. Este modelo ha tenido un

auge particular en el negocio de la música, un ejemplo de esto es *Spotify* que cuenta con 40 millones de usuarios, de los cuales 10 millones son suscriptores de pago.

En el negocio de las publicaciones y editorial, se ha desarrollado también el modelo de suscripción en diarios como *The New York Times* y *Financial Times*, quienes fueron los primeros en probar políticas de pago *online* frente a la gratuidad, que venía siendo el modelo general de los contenidos *online*. Este auge también se ha extendido al mundo del libro, donde se pueden visualizar casos de éxito como *Safari Books Online*, la cual suministra por suscripción libros electrónicos de Diseño, Tecnología, Informática y al mismo tiempo videos, conferencias, audiolibros, entre otros.

3.3.3 Por publicidad insertada. Según Rojas et ál. (2014, p. 34) consiste en ofrecer contenido gratuito pero con publicidad insertada, diferente al modelo “*Premium*” que va sin publicidad, además de obtener otras ventajas por el pago. Cabe anotar que no está demostrado que la publicidad insertada o contextual dé para tanto como para mantener lo gratuito, diferente a la suscripción *premium*.

Estas empresas desarrollan aplicaciones basadas en ideas innovadoras, de utilidad en contextos empresariales o que consiguen un espacio dentro del mercado del ocio. Aparte de la publicidad, usualmente van acompañadas de un esquema de obtención de ingresos por ventas dentro de la propia aplicación y merchandising.

3.3.4. Canal de venta. Según Alcázar (2013) consiste en vender a tus clientes productos o servicios dentro de un producto que encuentran gratis. Este concepto de compras dentro de una aplicación ha sido popularizado particularmente por las empresas productoras de videojuegos.

Cualquier aplicación que ofrezca una utilidad a la que pasado un tiempo sea difícil de renunciar, como un *software* de gestión de presupuestos, gratuito, que obligue a realizar un pago por formularios especiales.

Para implementar este modelo de negocio, es importante establecer previamente si un porcentaje significativo de los clientes se encuentra dentro del segmento de usuarios de *smartphone* y/o consumen los productos o servicios vía móvil. Antes de invertir en el desarrollo de una aplicación de este tipo, conviene evaluar si un sitio *web* optimizado para dispositivos móviles sería suficiente.

Las aplicaciones de canal de venta aprovechan la rapidez y sencillez de las transacciones móviles para transformar la impulsividad de sus compradores en ventas (Bilib, 2013).

3.3.5. Fidelización. Las empresas implementan aplicaciones orientadas a agregar valor a la operación del negocio, con el fin de posicionarse como marca y consolidar su imagen corporativa (Bilib, 2013).

Extender un negocio ya existente en el mundo móvil: Con este modelo de negocio se persiguen dos objetivos principales (Woodbridge, s.f):

- Mejora de una línea existente de negocios: mediante la identificación de las áreas clave del negocio y extenderlas al mundo móvil.
- Ampliar el alcance a nuevos clientes: los dispositivos móviles tienen el potencial de ampliar el alcance del producto o servicio ofrecido llegando a nuevos clientes, los cuales no están consumiendo lo que la empresa está vendiendo mediante sus canales actuales.

Para el caso Comfatolima, el modelo negocio que más se adapta es “extender un negocio ya existente en el mundo móvil”, orientado a mejorar una línea existente de negocio. Por lo cual el uso de la aplicación será completamente gratuito para afiliados a Comfatolima y permitirá la realización de trámites que actualmente funcionan a través de la página web y presencial en las instalaciones de Comfatolima.

3.4. Objetivos de negocio

Según la *Marketing Mobile Association* España en su Libro Blanco de Apps: “Con las aplicaciones móviles podemos conseguir varios objetivos de comunicación y/o de negocio de una forma eficaz y eficiente” (MMA, s.f). A continuación se hace un resumen de los principales objetivos que se buscan con la implementación de aplicaciones móviles, según la publicación citada:

3.4.1. Generación de imagen de marca. Busca impactar eficazmente en el público objetivo generando notoriedad y una imagen positiva de marca. Si se consigue que una aplicación aporte valor al usuario, esta se puede convertir en un objeto de recomendación iniciando un proceso viral y repercusión mediática.

3.4.2. Fidelización de usuarios/clientes. Puede establecer un canal de comunicación permanente y personalizado entre marca y usuario, orientado a conseguir la fidelización del cliente. De esta forma se fomenta el incremento de la relación entre el cliente y la marca, aumentando a su vez la frecuencia de compra.

3.4.3. Herramienta de gestión. Puede constituirse en una herramienta de seguimiento de los procesos de trabajo, localización de los miembros del equipo, control de calidad, gestión del conocimiento, obteniendo valor añadido para sus clientes.

3.4.4. Herramienta y canal de venta (mobile commerce). Busca crear nuevos hábitos de compra, puede ser complementaria a otros mecanismos de venta *on u off line*, logrando llegar a clientes que hasta el momento no han sido abordados.

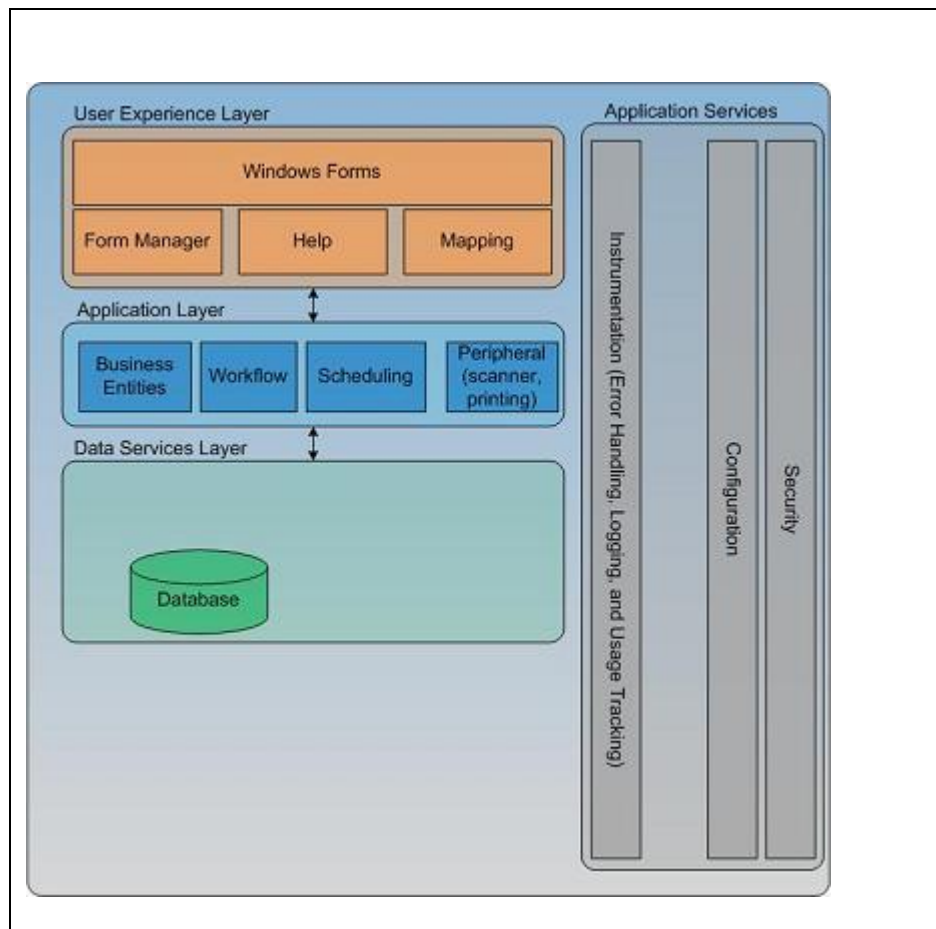
3.4.5. Aplicaciones generadoras de ingresos. Las aplicaciones de éxito generan ingresos a sus desarrolladores o editores, dividiéndose en dos sub modelos: por un lado comercializar espacios publicitarios en la aplicación distribuida gratuitamente, por otro lado la venta de la aplicación en sí.

3.5. Arquitecturas móviles

3.5.1. Aplicaciones autónomas o independientes. Son aquellas que se pueden ejecutar en el dispositivo sin ningún otro requisito o recursos, e.g. sin conexión a *Internet*. Se componen de una capa de presentación, una capa de aplicación y, finalmente, una capa de datos. Además algunos servicios cruzan estas tres capas principales.

A continuación se muestra un ejemplo de arquitectura Autónoma:

Figura 21. Ejemplo de arquitectura autónoma.

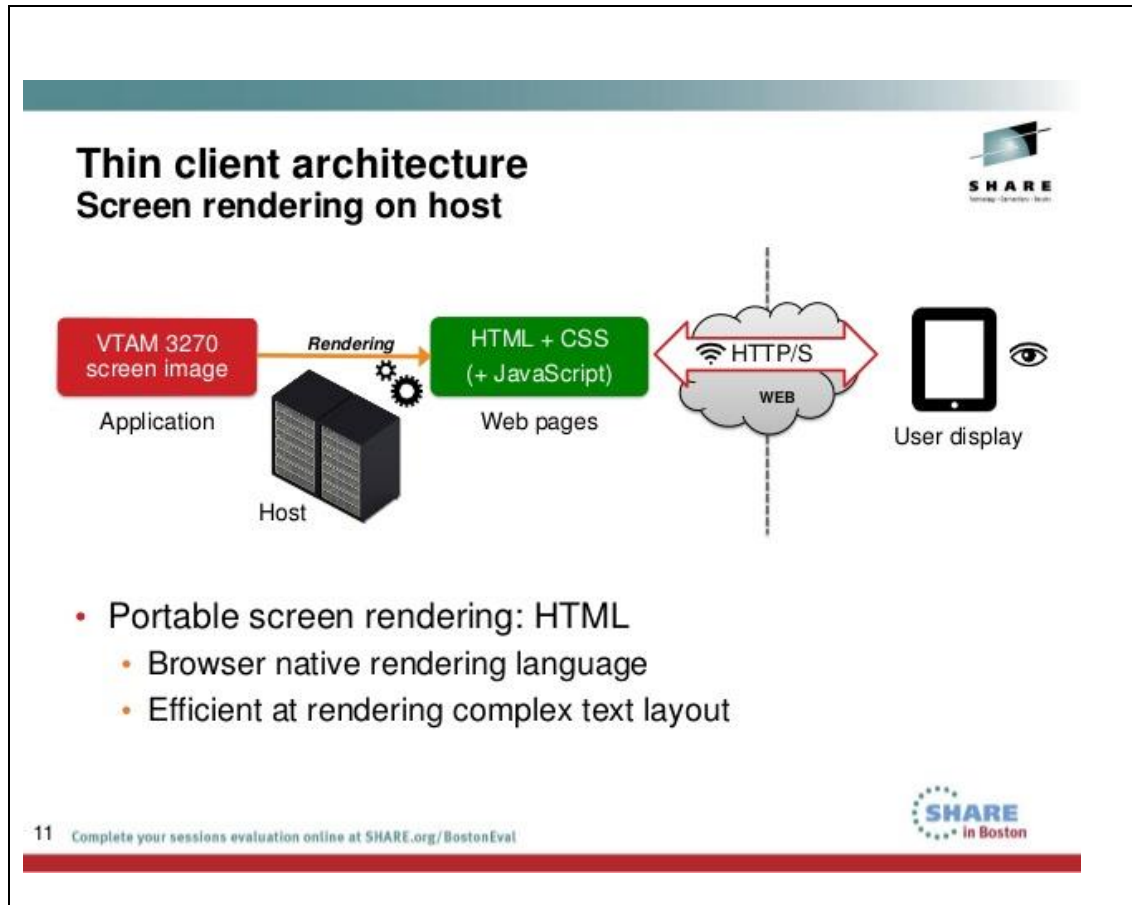


Fuente: <http://www.satter.org/2008/01/microsoft-mobil.html>

3.5.2. Cliente liviano. Hace referencia a una aplicación a la que se puede acceder por una interfaz Web en *HTML* y que se puede visualizar mediante un navegador Web. En este tipo de aplicaciones, toda la lógica del negocio se desarrolla en el lado del servidor. Algunas veces al navegador se le denomina cliente universal (CCM, 2016a).

A continuación se muestra un ejemplo de arquitectura “Cliente Liviano”:

Figura 22. Ejemplo de arquitectura cliente liviano.

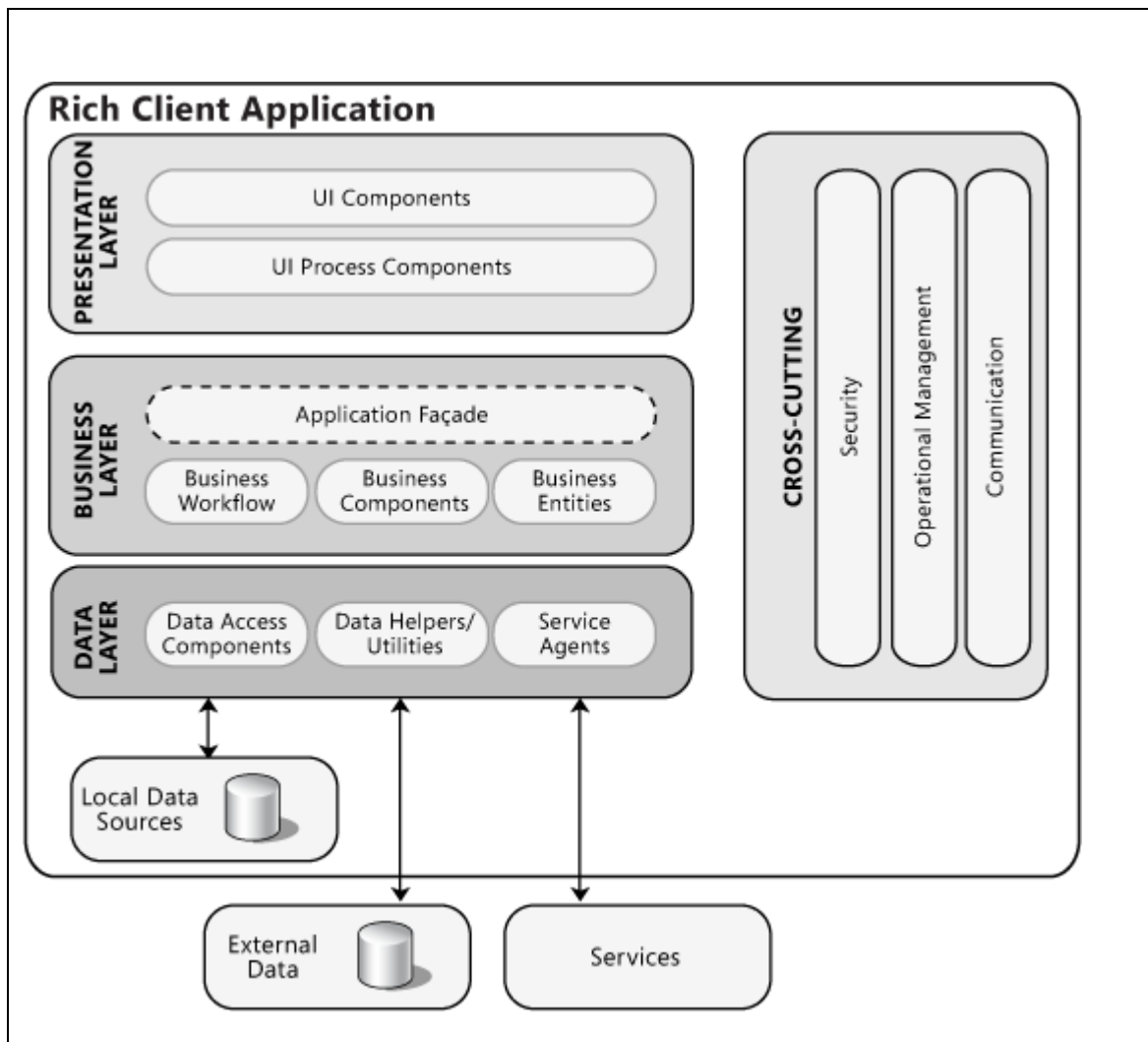


Fuente: <http://www.slideshare.net/Virtel/mobile-interfaces-to-mainframe-applications-architecture-considerations>

3.5.3. Cliente enriquecido. Es un término medio entre el cliente liviano y el cliente pesado. El objetivo de este tipo de cliente consiste en proporcionar una interfaz gráfica, escrita con una sintaxis basada en *XML*, que proporciona funcionalidades similares a las del cliente pesado, tales como arrastrar y soltar, pestañas, ventanas múltiples, menús desplegables (CCM, 2016b).

A continuación se muestra un ejemplo de arquitectura “Cliente Enriquecido”:

Figura 23. E.g. Arquitectura cliente enriquecido.



Fuente: <https://msdn.microsoft.com/en-us/library/ee658087.aspx>

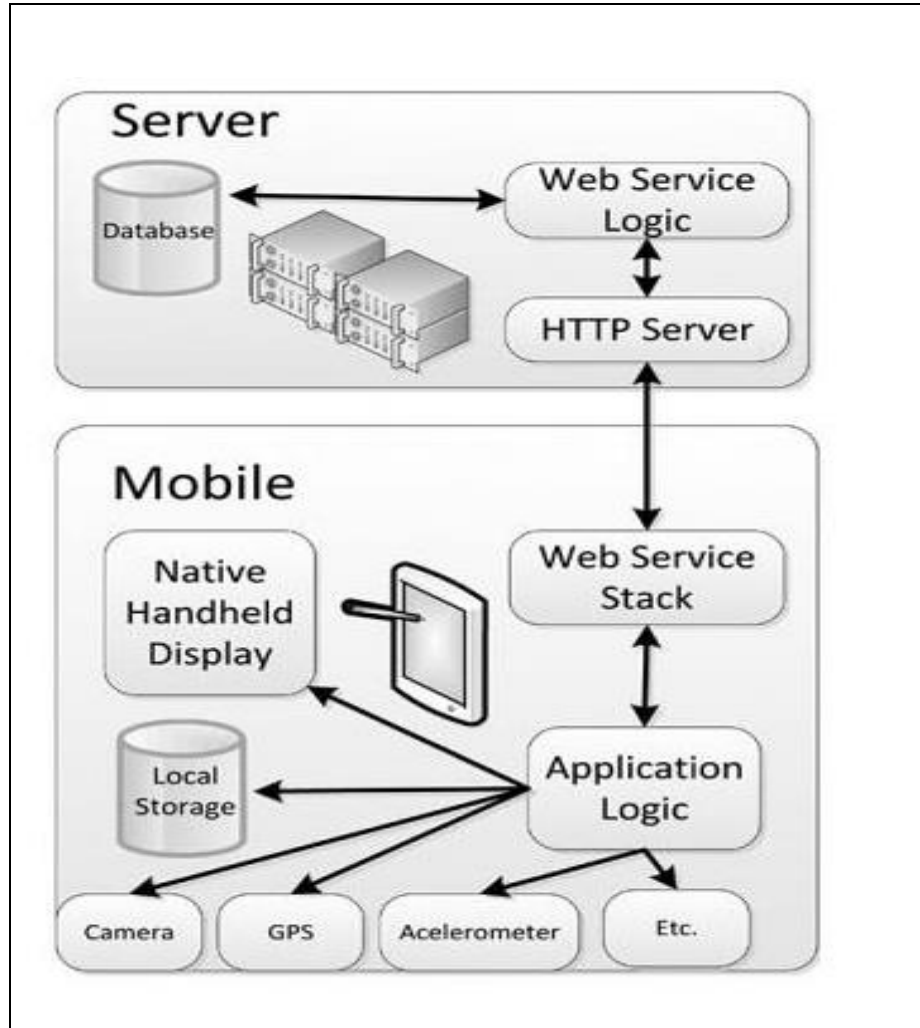
3.5.4. Cliente pesado. Se utiliza para una aplicación gráfica de cliente que se ejecuta en el sistema operativo del usuario. Suele tener una mayor capacidad de procesamiento y es posible

que tenga una interfaz gráfica sofisticada. Sin embargo, esto conlleva un desarrollo adicional y suele ser una mezcla de la lógica de presentación con la lógica de la aplicación.

Esta aplicación se instala en el sistema operativo del usuario y cada que se realiza una actualización, debe instalarse la nueva versión en el equipo cliente. Para contrarrestar esto, los programadores de este tipo de aplicaciones han incluido funcionalidades que se ejecutan al iniciar la aplicación y que verifican en un servidor remoto si existe una versión más reciente que la instalada. De ser así, le indica al usuario que debe descargar e instalar la actualización (CCM, 2016c).

A continuación se muestra un ejemplo de arquitectura “Cliente Pesado” propuesto:

Figura 24. Ejemplo de Arquitectura cliente pesado (Washke, 2015, p.93).



3.6. Servicios web

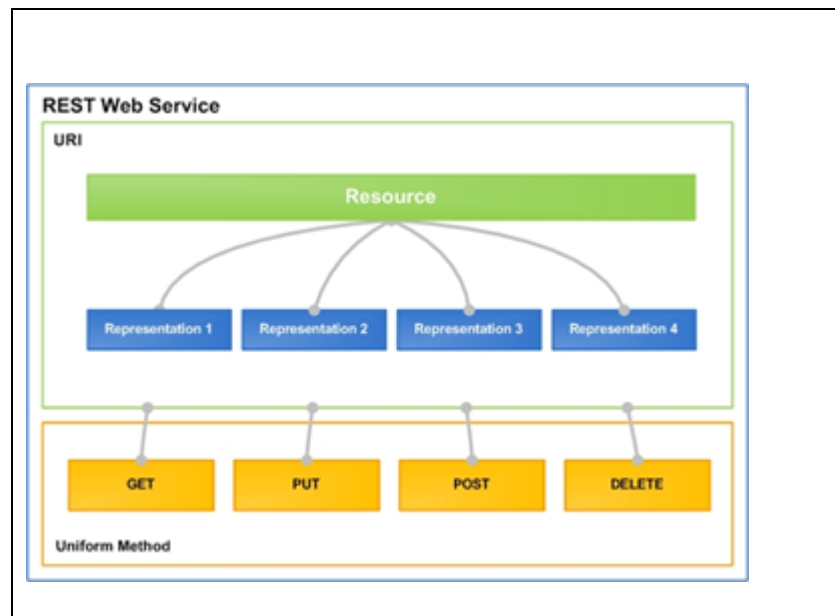
Los tipos de servicios web más populares son *SOAP* y *REST*, a continuación se hace una breve descripción de ellos:

3.6.1. RESTful. Son *APIs* que se adhieren a las restricciones arquitectónicas de *Representational State Transfer (REST)*, el cliente no requiere conocer ningún aspecto de la estructura del *API*. En este servicio el navegador no sabe por anticipado qué datos debe enviar ni

a dónde debe hacerlo. Datos como el *URI* y los datos requeridos son provistos por el servidor para que el cliente invoque al *API* (IBM, 2014).

A continuación se muestra un ejemplo de servicio “*REST*”:

Figura 25. Servicio REST.



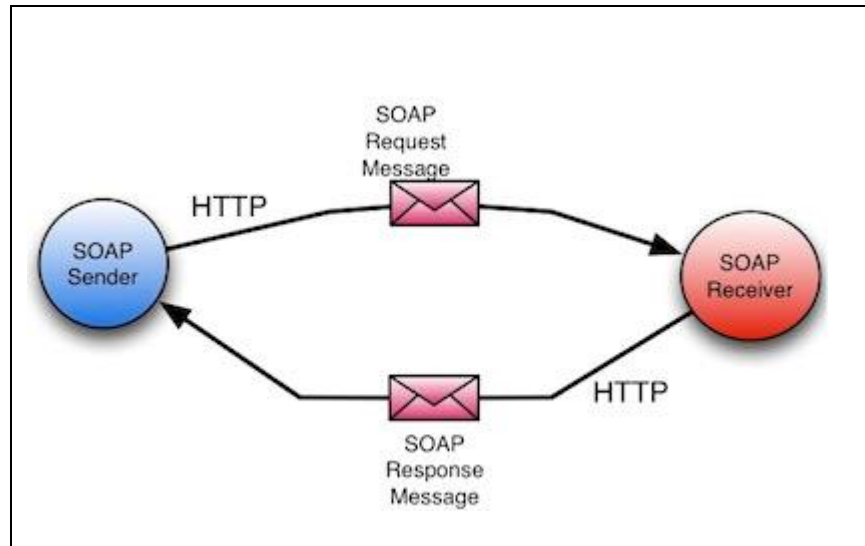
Fuente: <http://wink.apache.org/1.0/html/attachments/2328642/2523181.jpg>

3.6.2. SOAP. Es un formato de mensaje *XML* utilizando en interacciones de servicios *web*, los mensajes *SOAP* usualmente se envían sobre *HTTP* o *JMS*. El uso de *SOAP* en un servicio *web* específico se describe mediante la definición del formato estándar *Web Services Description Language* (IBM, 2015).

La especificación *SOAP* básicamente detalla el contenido de los mensajes y la manera en que la aplicación debe tratar los diferentes aspectos de estos, e.g. el encabezado. Con este esquema de funcionamiento se hace posible crear aplicaciones en las cuales un mensaje pase por diferentes intermediarios hasta llegar a su destino final (IBM, 2011).

A continuación se muestra un ejemplo de servicio “*SOAP*”:

Figura 26. Servicio Soap.



Fuente: <http://dasunhegoda.com/wp-content/uploads/2014/11/SOAP-Request.jpg>

3.7. Herramientas de desarrollo

Algunos de los entornos integrados de desarrollo de aplicaciones más populares son:

3.7.1. Eclipse. Es una plataforma diseñada para construir aplicaciones *web* y herramientas integradas. Permite el rápido desarrollo de aplicaciones mediante sus características basadas en el modelo “*plug-in*”. Proporciona un modelo común de interfaz de usuario, compatible con múltiples sistemas operativos.

Eclipse ofrece una arquitectura abierta para que los equipos de desarrollo de *plug-ins* se concentren en un área específica, permitiendo así agregar a la plataforma nuevas características y niveles de integración (Eclipse, s.f).

3.7.2. Netbeans. Permite desarrollar aplicaciones Java para escritorio y *web*, es gratuito, de código abierto y cuenta con una gran comunidad de usuarios y desarrolladores alrededor del mundo. *Netbeans* es el *IDE* oficial para Java 8, permite actualizar fácilmente las aplicaciones para emplear las instrucciones contenidas en esta nueva versión de Java. También permite gestionar proyectos de *software* de una forma fácil y eficiente mediante el uso de diferentes vistas de los datos, herramientas de manejo de versiones e integración con *Git* (Netbeans, s.f).

3.7.3. Android Studio. Ha sido elegido por *Google* como el *IDE* oficial para desarrollo de aplicaciones *Android*, cuenta con características como plantillas de código para construir características de aplicaciones comunes, editor de diseño enriquecido para arrastrar y soltar, herramientas para analizar el rendimiento, compatibilidad de versiones, entre otras (Android Studio, s.f).

3.8. La privacidad personal

Según Salz & Moranz (2013) “La cantidad de información que se puede obtener de los clientes cuando éstos usan una aplicación móvil es asombrosa. Incluye datos como los pagos realizados, claves de acceso, datos financieros, correos electrónicos, mensajes de texto, historial de compras, contactos” (p.108). Evidentemente esta información puede ser filtrada.

La información del cliente es reunida mediante las tiendas virtuales de descarga de aplicaciones cada vez que un cliente descarga una aplicación o interactúa con ésta. Aunque las tiendas han implementado sistemas de seguridad, los hackers han encontrado la manera de explotar las vulnerabilidades de los dispositivos, instalando código malicioso en ellos.

Según *juniper networks* (www.juniper.net), a 2013 se registró un incremento en *Malware* para *Android* de un cuatrocientos por ciento, que puede ser creado con fines maliciosos como alterar la operación del dispositivo, obtener información personal, inyectar virus, gusanos, caballos de troya, entre otros programas maliciosos.

Las personas confían en sus dispositivos móviles, especialmente en las aplicaciones que hacen todo. Por esta razón, antes de descargar e instalar aplicaciones en sus dispositivos, deben asegurarse de que sus datos personales, tales como cuentas bancarias, datos privados, contenido digital, se encuentran protegidos. Esta información y su vulnerabilidad al estar en un dispositivo móvil, se ha convertido en una de las principales preocupaciones de los usuarios, lo cual he llevado a muchos de ellos incluso a des-instalar las aplicaciones. Por esta razón es necesario diseñar las políticas de seguridad basadas en un profundo conocimiento de los requerimientos del usuario final.

Tanto los desarrolladores como las tiendas de aplicaciones, deben proporcionar:

- Una sencilla política de privacidad, única para cada aplicación móvil, que explique al consumidor cómo, porqué y con qué fin sus datos serán recopilados.
- Pantallas de ejemplo, que muestran cómo se le pedirá información al usuario
- Un ícono junto con la información de comercialización, que invite al consumidor a ver cómo se utilizará la información

Conforme la tecnología avance, los dispositivos móviles adquirirán más importancia en la vida de las personas. Las oportunidades de observar mediante éstos los comportamientos y actividades del consumidor serán mayores, así como también las probabilidades de abusar de los datos personales y la privacidad.

3.8.1. Normatividad sobre aplicaciones y datos. Teniendo en cuenta que las aplicaciones móviles hacen uso de la información del usuario, la cual existe en el dispositivo donde son instaladas, es necesario conocer la normatividad vigente en Colombia que regula el uso de dicha información dentro de los límites jurídicos y para fines estrictamente corporativos.

3.8.1.1. Ley estatutaria 1266 de 2008. Tiene por objeto desarrollar el derecho constitucional que tienen todas las personas a conocer, actualizar y rectificar las informaciones que se hayan recogido sobre ellas en bancos de datos, y los demás derechos, libertades y garantías constitucionales relacionadas con la recolección, tratamiento y circulación de datos personales a que se refiere el artículo 15 y el artículo 20 de la Constitución Política de Colombia, particularmente en relación con la información financiera y crediticia, comercial, de servicios y la proveniente de terceros países (Superintendencia de Industria y Comercio, 2008a).

3.8.1.2. Constitución política de Colombia. Artículo 15 (Delegatura de protección de datos), hace referencia a los siguientes derechos de las personas (Superintendencia de Industria y Comercio, s.f.):

- Derecho a la intimidad personal, familiar y al buen nombre
- Derecho a conocer, actualizar y rectificar informaciones que sobre ellas reposen en bancos de datos y archivos de entidades públicas y privadas
- La correspondencia y las formas de comunicación privada son inviolables, excepto mediante orden judicial

3.8.1.3. Sentencia C-1011/08. “por la cual se dictan las disposiciones generales del hábeas data y se regula el manejo de la información contenida en bases de datos personales, en especial la financiera, crediticia, comercial, de servicios y la proveniente de terceros países y se dictan otras disposiciones.” (Superintendencia de Industria y Comercio, 2008b).

3.8.1.4. Derecho al hábeas data. “El derecho al hábeas data es definido por la jurisprudencia constitucional como aquel que otorga la facultad al titular de datos personales de exigir de las administradoras de esos datos el acceso, inclusión, exclusión, corrección, adición, actualización y certificación de los datos, así como la limitación en las posibilidades de divulgación, publicación o cesión de los mismos, de conformidad con los principios que regulan el proceso de administración de datos personales. Este derecho tiene naturaleza autónoma y notas características que lo diferencian de otras garantías con las que, empero, está en permanente relación, como los derechos a la intimidad y a la información.”

3.8.1.5. ¿Qué es un dato personal? “Es cualquier información vinculada o que pueda asociarse a una o varias personas naturales que, dependiendo de su grado de utilización y acercamiento con la intimidad de las personas podrá ser pública, semiprivada o privada” (Banco Caja Social, 2012).

- Dato personal público, son aquellos datos personales que las normas y la Constitución han determinado expresamente como públicos y, para cuya recolección y tratamiento, no es necesaria la autorización del titular de la información. (e.g. Dirección, teléfono, datos contenidos en sentencias judiciales ejecutoriadas, datos sobre el estado civil de las personas, entre otros.)

- Dato personal semiprivado, son datos que no tienen una naturaleza íntima, reservada, ni pública y cuyo conocimiento o divulgación puede interesar no sólo a su titular, sino a un grupo de personas o a la sociedad en general. Para su tratamiento se requiere la autorización expresa del titular de la información. (e.g. Dato financiero y crediticio).
- Dato personal privado, es un dato personal que por su naturaleza íntima o reservada sólo interesa a su titular y para su tratamiento requiere de su autorización expresa. (e.g. Nivel de escolaridad)
- Dato personal sensible, es aquel dato personal de especial protección, por cuanto afecta la intimidad del titular y su tratamiento puede generar discriminación. NO puede ser objeto de tratamiento a menos que sea requerido para salvaguardar un interés vital del titular o éste se encuentre incapacitado y su obtención haya sido autorizada expresamente. (e.g. Origen racial o étnico, orientación política, convicciones religiosas, datos biométricos, relativos a la salud, entre otros.)”.

3.8.1.6. Ley estatutaria 1581 de 2012. “Es una ley que complementa la regulación vigente para la protección del derecho fundamental que tienen todas las personas naturales a autorizar la información personal que es almacenada en bases de datos o archivos, así como su posterior actualización y rectificación. Esta ley se aplica a las bases de datos o archivos que contengan datos personales de personas naturales.” (Banco Caja Social, 2012).

3.8.1.7. Control de la SIC. En Colombia, la Superintendencia de Industria y Comercio se unió a la Red Global de Vigilancia de la Privacidad, GPEN por su sigla en inglés, institución conformada por veintiséis autoridades de protección de datos. Estas autoridades realizaron un

barrido del 12 al 18 de mayo de 2014. En dicha revisión se tuvieron en cuenta aspectos como: los permisos que las aplicaciones buscaban obtener, el exceso de las aplicaciones en la obtención de permisos y la manera como las aplicaciones le explicaban al usuario los permisos que deseaban obtener y para qué los iban a utilizar. Los resultados fueron:

- El 75% de todas las aplicaciones examinadas solicitaron uno o más permisos, para acceder a cuentas de usuario, cámara del dispositivo, contactos. Esto revela la necesidad de que las aplicaciones sean más transparentes frente a este tema.
- Con cerca del 59% de las aplicaciones, los participantes del barrido tuvieron enormes dificultades en encontrar, antes de la instalación, algún comunicado de la empresa en cuestión que explicara su política en materia de protección de datos personales. No hay claridad sobre lo que harán con la información o muestran políticas de privacidad que no corresponden con la aplicación que el usuario está instalando.
- En el caso de casi un tercio (31%) de las aplicaciones, los participantes en el barrido expresaron inquietud sobre la índole de los permisos que eran solicitados. Los usuarios tienen la sensación de que las aplicaciones solicitan acceso a más funciones e información de las que necesitan.
- Cerca de 43% de las aplicaciones no habían adaptado, para lectura en pantalla pequeña, sus comunicados relativos a la protección de datos personales. Predominando los textos demasiado extensos y la letra pequeña.
- Sólo un reducido número (15%) de las aplicaciones examinadas suministró una explicación clara sobre la manera en que los datos personales iban a ser recopilados, utilizados y divulgados.

- Estos aspectos deberán ser tenidos en cuenta para el desarrollo de la aplicación móvil de Comfatolima.

3.8.1.8. Política de privacidad. Los requisitos legales de las aplicaciones móviles son iguales a las del *software*, por lo tanto al momento de desarrollarlas se deben tener en cuenta los siguientes aspectos legales (Abogalista, 2013):

- Privacidad y datos personales: no sólo los que identifiquen directamente a la persona, sino también aquellos que la hagan identificable.
- Nombre de la aplicación: denominación, marca, nombre comercial, el registro del dominio.
- La autoría de la aplicación: la aplicación se encuentra protegida por la ley y los derechos de propiedad intelectual para reconocer y proteger a sus creadores y beneficiarnos de su explotación comercial frente a intentos de plagio.
- La política de privacidad debe especificar los fines que se le darán a la información recopilada del usuario

4. Desarrollo

4.1. Fase de requerimientos

De acuerdo con el cronograma de actividades establecido para la realización del proyecto, la fecha de inicio de actividades era el día primero de agosto de 2016. Esta fecha no pudo ser cumplida debido a que en la Oficina de Sistemas de Comfatolima se redujo el personal en dos unidades (Jefe de la Oficina y Analista Programador), por lo tanto fue necesario abrir convocatoria para reemplazar a los funcionarios que no continuaron con la entidad y reorganizar las actividades de la Oficina para garantizar la continuidad de la operación y responder a los requerimientos de información de los entes de control internos y externos. Esta novedad ocasionó que el proyecto fuera postergado por un espacio de dos meses, por lo tanto la fecha de inicio fue el día 2 de octubre de 2016.

El levantamiento de información acerca de los requerimientos de la aplicación móvil se realizó mediante mesas de trabajo, a las cuales asistieron como clientes por parte de Comfatolima el Coordinador de la Oficina de Subsidio y Aportes, la nueva Jefe de la Oficina de Sistemas y dos funcionarios Operadores de Sistemas de la Caja, reunidos junto con el desarrollador del proyecto. En dichas mesas de trabajo se establecieron todas y cada una de las funcionalidades con que debería contar la nueva aplicación, se analizaron documentos de referencia y se clarificaron aspectos como la protección de datos de usuario, la confidencialidad de la información, la cantidad de datos históricos que visualizarían los usuarios, los mecanismos de registro y autenticación y las alternativas de extracción de datos del sistema de información de subsidio y aportes, su procesamiento y carga de datos en el servidor en la nube y las actividades

y controles que debería realizar el administrador de la aplicación móvil mediante la herramienta de software requerida para este fin.

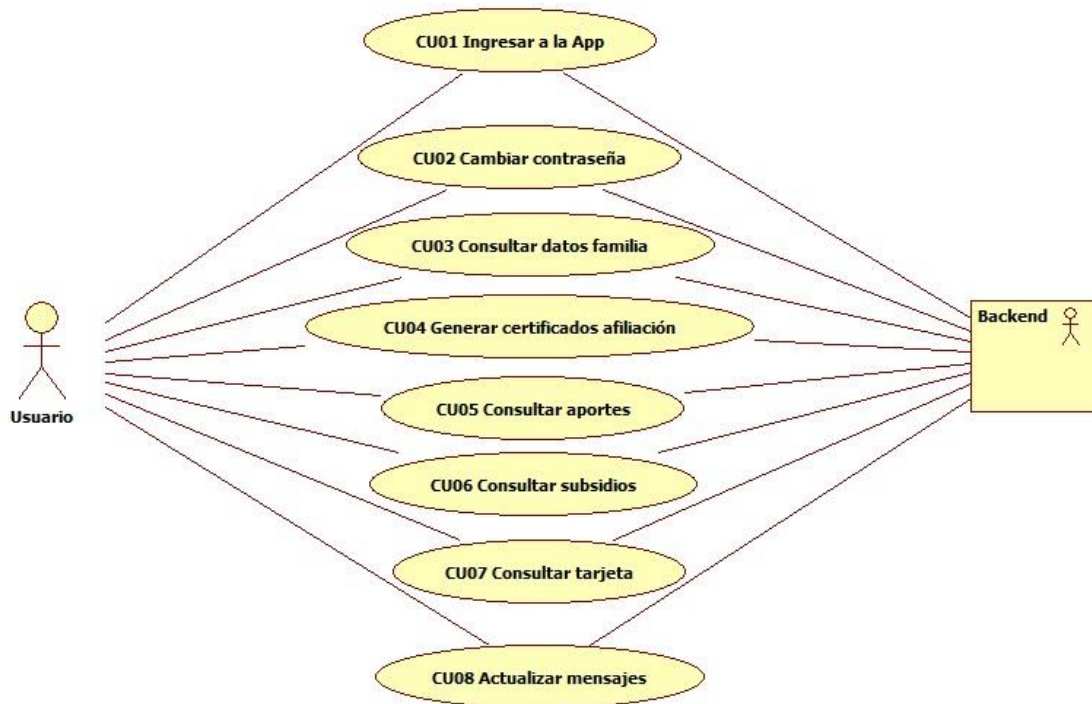
El proceso de levantamiento de información inició el día 2 de octubre de 2016 y finalizó el día 18 de octubre de 2016, tiempo en el cual se generó un documento de requerimientos que fue revisado y optimizado gradualmente por los clientes y el desarrollador y finalmente se consolidó el documento final. Ver anexo 6. (Especificación de requisitos). Posteriormente se dio inicio a cada una de las iteraciones que componen el proyecto.

4.2. Diseño de la arquitectura

Para el diseño de la arquitectura de la aplicación se utilizó el modelo de 4+1 vistas de *Kruchten*. A continuación se muestran en resumen los diagramas generales del modelo arquitectónico y en cada iteración se pueden observar los diagramas específicos, para más detalles ver anexo 7 (documento de arquitectura).

VISTA DE ESCENARIOS

Figura 27. Diagrama de casos de uso aplicación móvil.



Fuente: elaboración propia

Figura 28. Diagrama de casos de uso para el subsistema de administración.

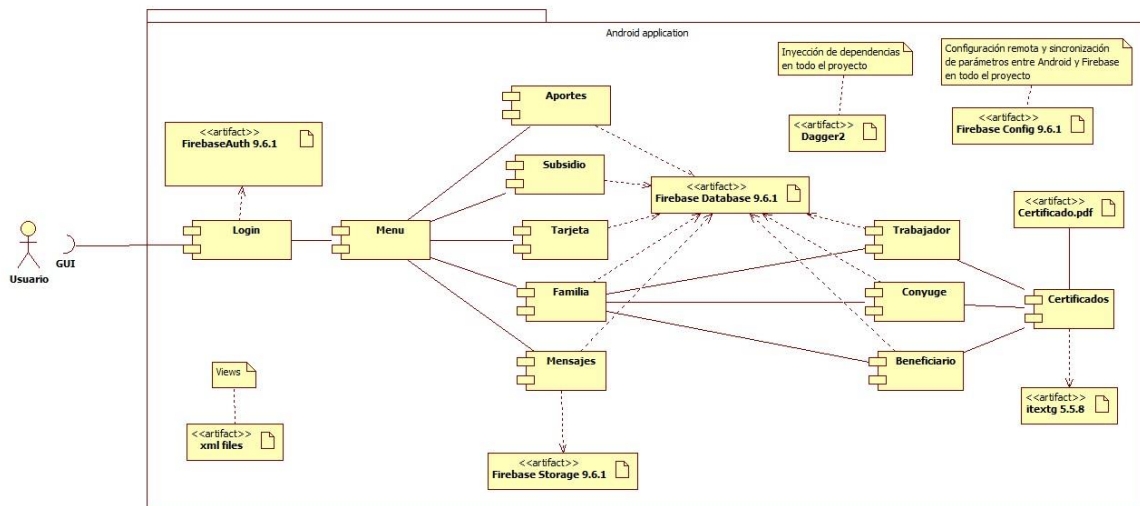


Fuente: elaboración propia

VISTA DE DESPLIEGUE

Diagrama de componentes

Figura 29. Diagrama de componentes aplicación móvil

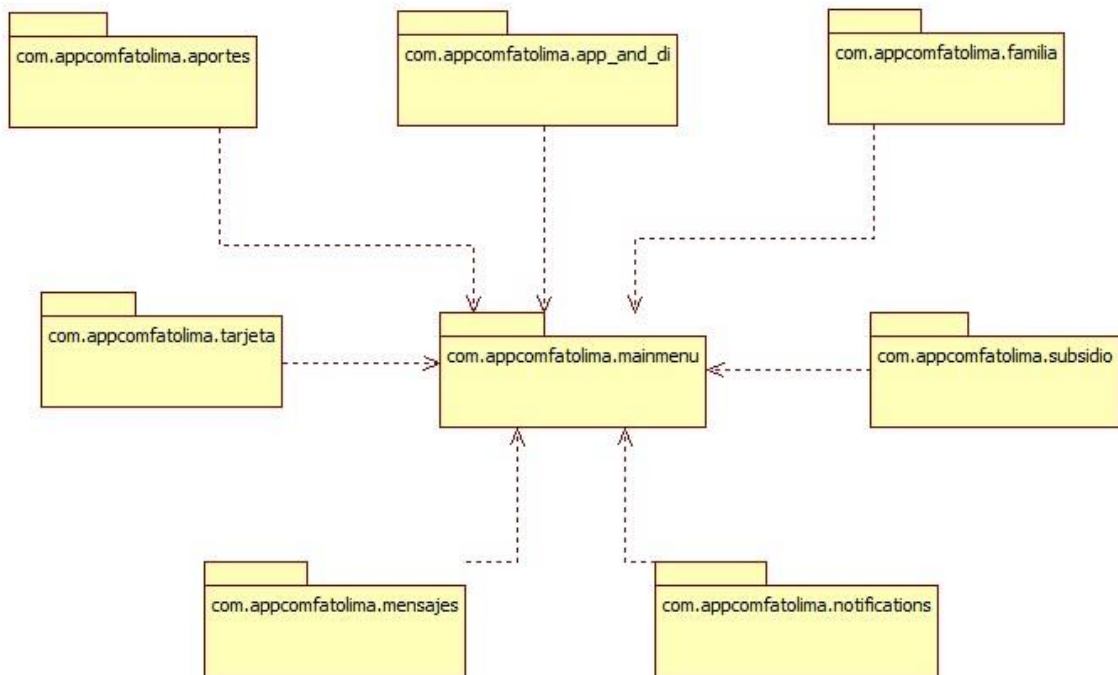


Fuente: elaboración propia

Diagramas de paquetes

A continuación se muestra el diagrama de paquetes general del proyecto, para más detalles ver el anexo XXX (documento de arquitectura).

Figura 30. Diagrama de paquetes general aplicación móvil

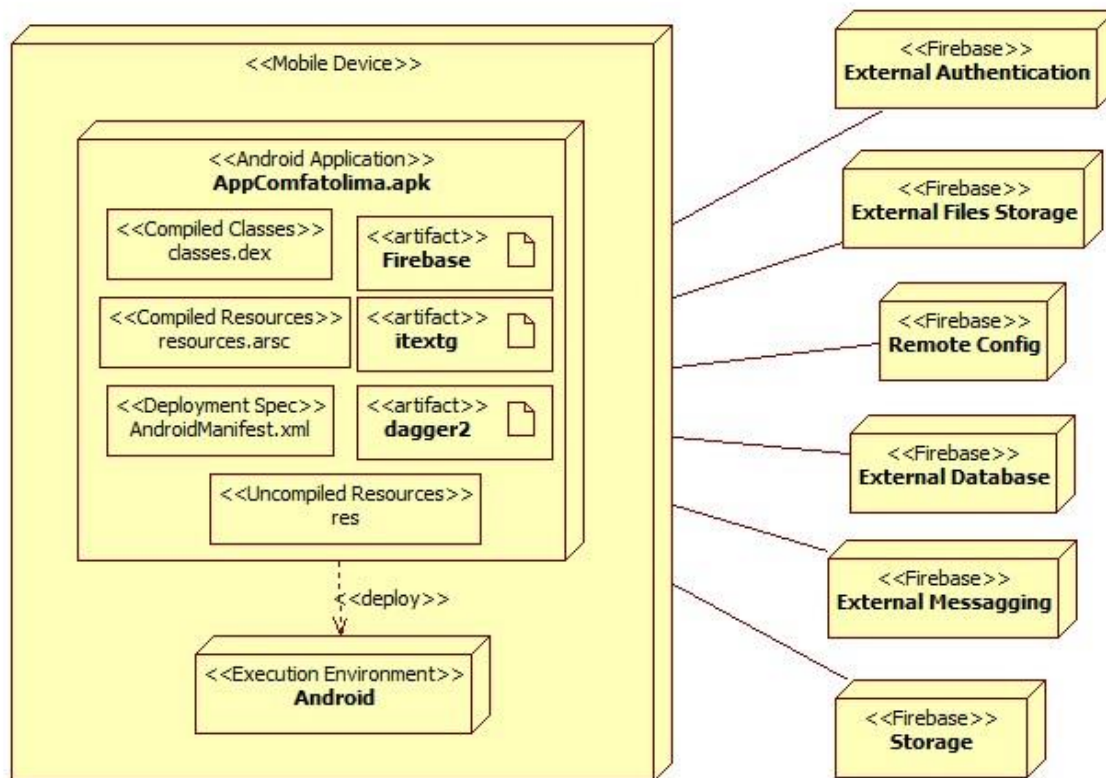


Fuente: elaboración propia

VISTA FISICA

Diagrama de despliegue

Figura 31. Diagrama de despliegue aplicación móvil.



Fuente: elaboración propia

4.3. Iteraciones del proyecto

4.3.1. Ingreso al sistema y cambio de contraseña.

4.3.1.1. Definición de requerimientos. En esta iteración se desarrollaron los casos de uso CU01 - ingresar a la aplicación y CU02 – cambiar contraseña, los cuales se derivan de los requerimientos RE_APP_01 - ingresar por primera vez a la aplicación y RE_APP_02 - ingresar a la aplicación. Ver anexo 6 -especificación requerimientos, páginas 14 y 16 respectivamente. Los requerimientos no funcionales se pueden consultar en la página 25 del mismo anexo.

4.3.1.2. Casos de uso textuales.

Tabla 17. Caso de uso textual CU01

CASO DE USO TEXTUAL	
Codigo	CU01
Nombre	Ingresar al sistema
Descripcion	Establece los procesos que se ejecutan cuando el usuario intenta ingresar normalmente a la aplicación
Actores	Afiliado a Comfatolima registrado como usuario de la aplicación móvil
Disparador	El usuario ejecuta la aplicación desde su dispositivo móvil
Precondiciones	1. Usuario afiliado a Comfatolima en estado Activo 2. Usuario definido en el sistema de autenticación del backend y en estado activo en el mismo 3. Ejecución exitosa de la obtención de clave de acceso a la aplicación (aplicación instalada en el dispositivo, clave de acceso asignada)

Postcondiciones	1. Usuario autenticado en la aplicación 2. Menú principal de la aplicación desplegado
Flujo normal	1. El usuario ejecuta la aplicación en su dispositivo móvil 2. El sistema verifica que haya conexión a internet en el dispositivo (E1) 3. El sistema despliega la opción para ingresar 4. El usuario ingresa su correo electrónico (E2) 5. El usuario ingresa su clave de acceso (E3) 6. El sistema verifica que el correo y la clave concuerden con las registradas en el sistema (E4, E5) 7. El sistema verifica que el usuario se encuentre activo en el sistema (E6) 8. El sistema despliega el menú principal
Flujos alternativos	1. El usuario ejecuta la opción de cambiar contraseña, se ejecuta el caso de uso (CU02 - Cambiar contraseña)
Excepciones	E1. No hay conexión a internet en el dispositivo, mostrar mensaje "El dispositivo no cuenta con conexión a internet, podrá visualizar los últimos datos almacenados", continuar con el paso 4 del flujo normal E2. El usuario deja en blanco su correo electrónico o ingresa un dato con formato incorrecto, mostrar mensaje "Debe ingresar un correo electrónico válido", se repite paso 4 del flujo normal E3. El usuario deja en blanco su clave de acceso, mostrar mensaje "Debe ingresar su clave de acceso", se repite paso 5 del flujo normal E4. El correo electrónico no existe en el sistema, mostrar mensaje: "Su correo electrónico no aparece registrado en Comfatolima", se repite el paso 4 del

	flujo normal E5. La clave de acceso ingresada por el usuario es incorrecta, mostrar mensaje "Clave de acceso incorrecta", se repite el paso 5 del flujo normal E6. El usuario se encuentra inactivo en el sistema, mostrar mensaje "Su usuario se encuentra desactivado en el sistema", termina la ejecución del caso de uso
Prioridad	Alta
Frecuencia de uso	Puede llegar a ser diaria, no significa conexión constante
Reglas del negocio	1. Si el usuario se encuentra en estado inactivo, no podrá ingresar a la aplicación 2. Si el usuario se encuentra en estado inactivo, no podrá cambiar su clave de acceso 3. Si el afiliado se inactiva en Comfatolima, su usuario de la aplicación móvil también será desactivado 4. El ingreso a la aplicación solo se podrá realizar cuando exista conexión a Internet en el dispositivo
Requerimientos especiales	N/A
Suposiciones	N/A
Notas y preguntas	Al momento de desactivar un usuario y que este se encuentre logueado en la aplicación, seguirá logueado hasta cuando Firebase intente actualizar el token de acceso, lo cual sucede aproximadamente cada hora. En ese momento la

	actualización de token fallará y al cambiar de opción, inmediatamente será redireccionado al formulario login
--	---

Tabla 18, caso de uso textual CU02, fuente: elaboración propia

CASO DE USO TEXTUAL	
Codigo	CU02
Nombre	Cambiar contraseña
Descripcion	Establece o cambia la contraseña de acceso a la aplicación, a solicitud del usuario (caso de uso implementado con base en la utilidad de cambio de clave de Firebase)
Actores	Afiliado a Comfatolima registrado como usuario de la aplicación móvil
Disparador	El usuario ejecuta la aplicación desde su dispositivo móvil
Precondiciones	1. Usuario afiliado a Comfatolima en estado Activo 2. Usuario definido en el sistema de autenticación del backend y en estado activo en el mismo
Postcondiciones	1. Clave de usuario actualizada 2. Opción para ingresar al sistema desplegada
Flujo normal	1. El usuario ejecuta la aplicación en su dispositivo móvil 2. El sistema verifica que haya conexión a internet en el dispositivo (E1) 3. El sistema despliega la opción para ingresar 4. El usuario escoge la opción para cambiar o restablecer clave 5. El sistema debe verificar si el correo electrónico ingresado es correcto a

	nivel de formato (E2) 6. El sistema generará un correo electrónico con el enlace para que el usuario establezca su clave de acceso 7. Luego de esto el usuario podrá ingresar al sistema con la clave que ha establecido
Flujos alternativos	1. El usuario ingresa correo electrónico y contraseña en los campos correspondientes, se ejecuta el caso de uso (CU01 - Ingresar al sistema)
Excepciones	E1. No hay conexión a internet en el dispositivo, mostrar mensaje "El dispositivo no cuenta con conexión a internet, podrá visualizar los últimos datos almacenados", continuar con el paso 4 del flujo normal E2. El usuario deja en blanco su correo electrónico o ingresa un dato con formato incorrecto, mostrar mensaje "Debe ingresar un correo electrónico válido", se repite paso 4 del flujo normal
Prioridad	Alta
Frecuencia de uso	Baja, la primera vez que se ejecutará la aplicación o cuando el usuario requiera restablecer su contraseña
Reglas del negocio	1. Si el usuario se encuentra en estado inactivo, no podrá ingresar a la aplicación 2. Si el usuario se encuentra en estado inactivo, no podrá cambiar su clave de acceso 3. Si el afiliado se inactiva en Comfatolima, su usuario de la aplicación móvil también será desactivado
Requerimientos	N/A

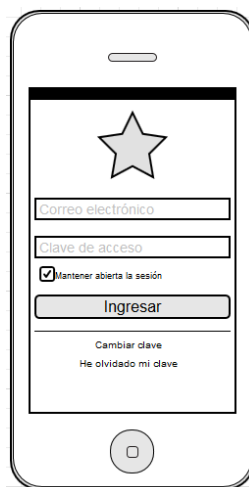
especiales	
Suposiciones	N/A
Notas y preguntas	

Fuente: elaboración propia

Estos casos de uso hacen parte del anexo 8 – casos de uso textuales, páginas 2 y 4 respectivamente.

4.3.1.3. Prototipos de interfaces y reportes. Se crea el prototipo de la interfaz de usuario para la opción de ingreso al sistema.

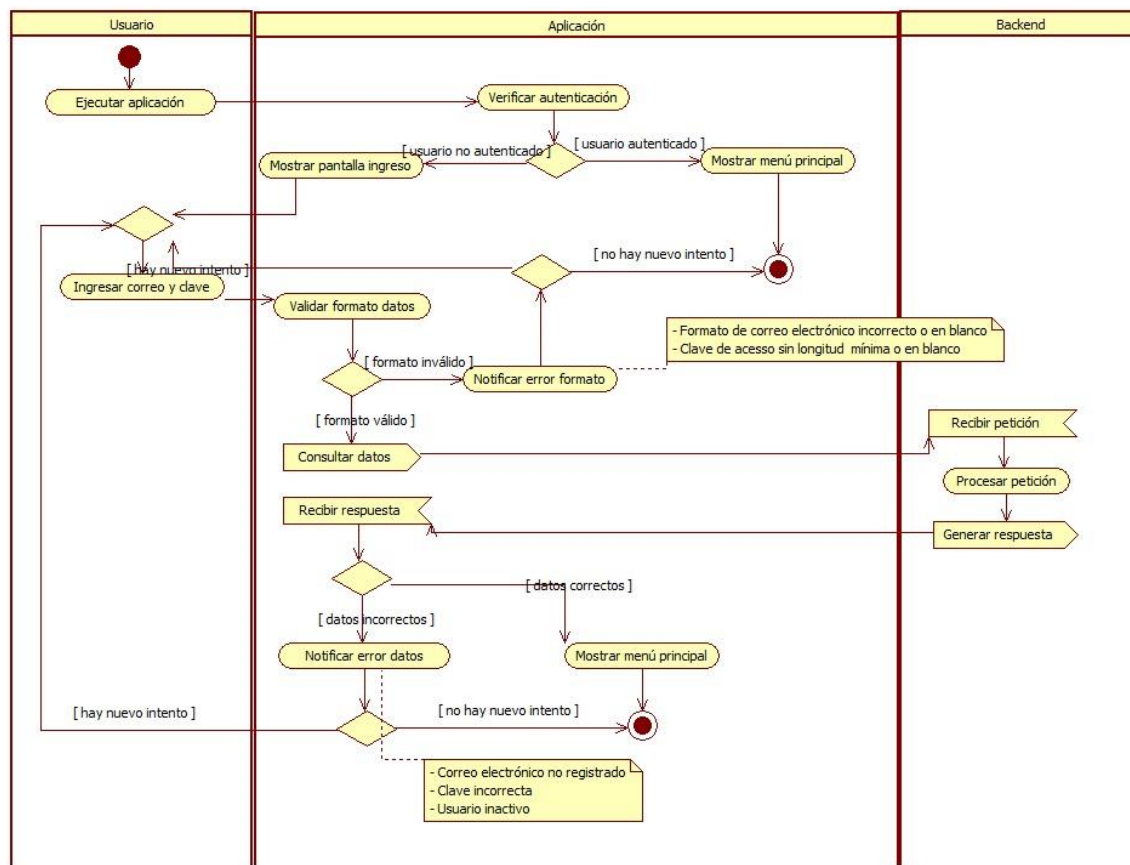
Figura 30. Prototipo funcionalidad ingresar al sistema.



Fuente: elaboración propia

4.3.1.4. Diagrama de actividades.

Figura 31. Diagrama de actividades caso de uso CU01.



Fuente: elaboración propia

El los diagramas arquitectónicos de esta iteración pueden consultarse en el anexo 7 – documento de arquitectura, así:

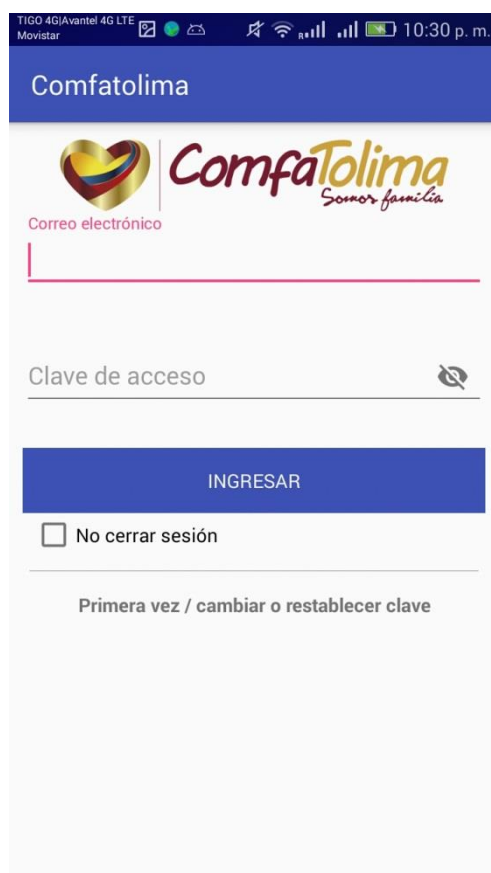
- Diagrama de clases, página 5
- Diagrama de comunicación, página 15
- Diagrama de secuencia, página 19
- Diagrama de actividades, página 24
- Diagrama de paquetes, página 33

4.3.1.5. Pruebas. Se realizaron pruebas unitarias a un total de 30 clases codificadas en la iteración. El detalle se puede consultar en el anexo 9 – pruebas unitarias y de integración, página 2.

Se realizaron pruebas de usabilidad y adaptación a diferentes tipos de dispositivos, utilizando los criterios establecidos en el anexo 10 – pruebas de adaptación y usabilidad.

4.3.1.6. Despliegue. A continuación se presenta la pantalla que ofrece las funcionalidades de ingresar al sistema y cambiar o restablecer clave:

Figura 32. Pantalla funcionalidades de ingresar al sistema y cambiar o restablecer clave.



The screenshot displays the Comfatolima mobile application interface. At the top, the status bar shows 'TIGO 4G/Avantel 4G LTE' and 'Movistar'. The app header is blue with the text 'Comfatolima'. Below the header is the Comfatolima logo, which includes a stylized heart and the text 'ComfaTolima Somos familia'. The main form area contains a 'Correo electrónico' field, a 'Clave de acceso' field with a toggle icon, a blue 'INGRESAR' button, a checkbox labeled 'No cerrar sesión', and a link that reads 'Primera vez / cambiar o restablecer clave'.

Fuente: Elaboración propia

4.3.1.7. PSP – tamaños y tiempos estimados versus reales.

Figura 33. Tamaños y tiempos estimados versus reales.

Overall Metrics

Summary	Plan	Actual	A la Fecha
<i>Size/Hour</i>	7,32	#VALUE!	53
Program Size			
Base (B)	0	0	0
Deleted (D)	0	0	0
Modified (M)	0	0	0
Added (A)	1090	#VALUE!	5044
Reused (R)	0	0	3965
Added and Modified (A+M)	1090	#VALUE!	5044
Total Size (T)	1090	#VALUE!	9009
Total New Reusable	440	334	1078
Estimated Proxy Size (E)	1945	1421	2552

Time In Phase

	Plan	Actual	A la Fecha	% A la Fecha
Planeación	5:27	5:02	3:29	3,66%
Diseño	3:13	13:06	2:03	2,15%
Codificación	82:37	33:45	52:47	55,40%
Compilación	11:32	4:31	7:22	7,74%
Pruebas	43:32	10:06	27:49	29,20%
Postmortem	2:40	0:43	1:42	1,79%
Total	149:00	67:13	95:12	

Fuente: Elaboración propia

4.3.1.8. PSP – defectos detectados y removidos

Figura 34. Defectos detectados y removidos.

Defects Injected

	Actual	A la Fecha	% A la Fecha
Antes de Desarrollo	0	0	0%
Planeación	0	0	0%
Diseño	0	25	28,70%
Codificación	25	62	71,30%
Compilación	0	0	0%
Pruebas	0	0	0%
Total	25	87	

Defects Removed

	Actual	A la Fecha	% A la Fecha
Planeación	0	0	0%
Diseño	0	5	5,75%
Codificación	5	35	40,20%
Compilación	2	15	17,20%
Pruebas	18	32	36,80%
Total	25	87	
After Development	0	0	

Reporte generado a las 11:36 AM el 30/06/2017

Fuente: Elaboración propia

4.3.1.9. Análisis de la iteración. La estimación inicial de tamaño fue de 1945 LOC y el tamaño final de la iteración fue de 1421 LOC, es decir que se codificó sólo un 73% de lo estimado. Las causas de esta favorable diferencia fueron:

Más de la mitad de la aplicación fue re-escrita nuevamente, debido a que para el desarrollo anterior se estaba utilizando el *PAAS Openshift*, el cual ofrece algunos servicios como base de datos y almacenamiento de archivos, pero no cuenta con utilidades de autenticación y mensajes *push*, características fundamentales de la aplicación. Adicional a esto, el diseño arquitectónico que se había concebido inicialmente presentaba una gran complejidad y fomentaba la repetición de código, lo cual hizo replantear la estructura del proyecto.

En ese momento el autor encontró información sobre el servicio *Firebase* prestado por *Google*. *Firebase* es un *Backend as a Service*, por lo tanto ofrece en forma integrada todas las funcionalidades que requiere una aplicación móvil. Este fue el motivo de tomar la decisión de volver a empezar la aplicación, utilizando ahora *Firebase* y adaptando el patrón *Model View Presenter*, el cual sumado a la poca cantidad de código que se escribe cuando se usa *Firebase*, permitió el ahorro en tamaño.

Frente a los plazos acordados con el cliente para la entrega del proyecto, existe la ventaja de que la implementación de la aplicación está proyectada para el año 2017, sin una fecha de lanzamiento específica, debido a la complejidad que representa coordinar armónicamente los datos que se extraen del sistema de información de subsidio y los datos que se pondrán a disposición de los usuarios de la aplicación. Esta situación también motivó al cambio de servicio en la nube y a la reescritura del programa.

El tiempo estimado de desarrollo fue de 149 horas y el tiempo real fue de 67 horas, es decir el 45% del tiempo estimado. Esta situación obedece al uso de *Firebase*. No obstante, como se podrá apreciar en los registros de tiempo, el avance no es diario, debido a que el desarrollador tiene que enfrentar una fuerte curva de aprendizaje de *Java Android*, *Firebase*, *NodeJs*, *Spring Framework*, *Hibernate*. Estas tecnologías se utilizan en la realización del proyecto mediante el poco aconsejable esquema de aprender-hacer.

Con lecciones aprendidas de proyectos anteriores, se detectaron y corrigieron un total de 25 defectos significativos en la fase, muchos de ellos en clases que serán reutilizadas, por lo tanto se espera que los errores tiendan a disminuir.

Todos los reportes PSP de esta iteración se pueden consultar en el anexo 11 – PSP iteración 01.

4.3.2. Consultar aportes

4.3.2.1 Definición de requerimientos. En esta iteración se desarrolló el caso de uso CU05 – consultar aportes, el cual se deriva del requerimiento RE_APP_05 – consultar pago aportes. Ver anexo 6 -especificación requerimientos, página 20. Los requerimientos no funcionales se pueden consultar en la página 25 del mismo anexo.

4.3.2.2 Casos de uso textuales.

Tabla 19. Caso de uso textual CU05.

CASO DE USO TEXTUAL	
Codigo	CU05
Nombre	Consultar Aportes
Descripcion	Establece los procesos que se ejecutan cuando el usuario ejecuta la opción para consultar Aportes
Actores	Afiliado a Comfatolima registrado como usuario de la aplicación móvil
Disparador	El usuario ejecuta la opción de consulta de aportes
Precondiciones	1. Dispositivo móvil con conexión a Internet 2. Usuario afiliado a Comfatolima en estado Activo 3. Ejecución exitosa del caso de uso CU01 (ingreso al sistema)
Postcondiciones	1. Listado de Aportes del usuario visualizados en la aplicación
Flujo normal	1. El usuario ejecuta la opción de consultar Aportes 2. El sistema verifica si el usuario registra aportes a su nombre

	3. Si el usuario registra aportes a su nombre, el sistema procede a desplegarlos (ver especificación de requerimientos para la lista de datos a desplegar) 4. Los aportes se visualizarán por mes, si tiene más de un aporte realizado a su nombre para un mes, los mostrará por separado 5. Si el usuario selecciona uno aportes, el sistema deberá visualizar el(los) detalle(s) del aporte realizado 6. Si el usuario no tiene aportes a su nombre, el sistema le notificará esto mediante un mensaje
Flujos alternativos	1. El usuario luego de ver sus datos selecciona la opción de enviar un mensaje a la Caja, se ejecuta el caso de uso 08 (Actualizar Mensajes)
Excepciones	E1. No hay conexión a internet en el dispositivo, mostrar mensaje "El dispositivo no cuenta con conexión a internet", se visualizarán los últimos datos consultados
Prioridad	Alta
Frecuencia de uso	Puede llegar a ser diaria, no significa conexión constante
Reglas del negocio	1. Si el usuario podrá consultar esta información solo si se encuentra en estado activo 2. El usuario no podrá modificar ni eliminar esta información, es solo de consulta 3. Si el usuario se había logueado, pero su dispositivo no tiene conexión en el momento, le mostrará los últimos datos consultados

Requerimientos especiales	N/A
Suposiciones	N/A
Notas y preguntas	N/A

Fuente: Elaboración propia

Este caso de uso hace parte del anexo 8 – casos de uso textuales, página 9.

4.3.2.3. Prototipos de interfaces y reportes. Se crea el prototipo de la interfaz de usuario para la opción de consultar aportes.

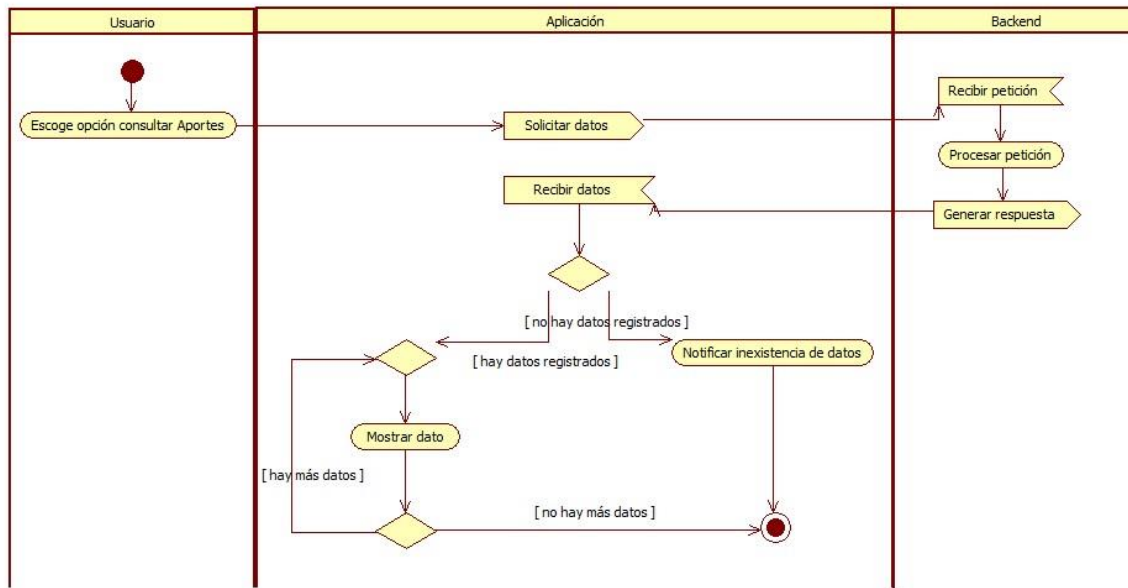
Figura 35. Prototipo funcionalidad consultar aportes.



Fuente: Elaboración propia

4.3.2.4. Diagrama de actividades

Figura 36. Diagrama de actividades caso de uso CU05.



Fuente: Elaboración propia

El los diagramas arquitectónicos de esta iteración pueden consultarse en el anexo 7 – documento de arquitectura, así:

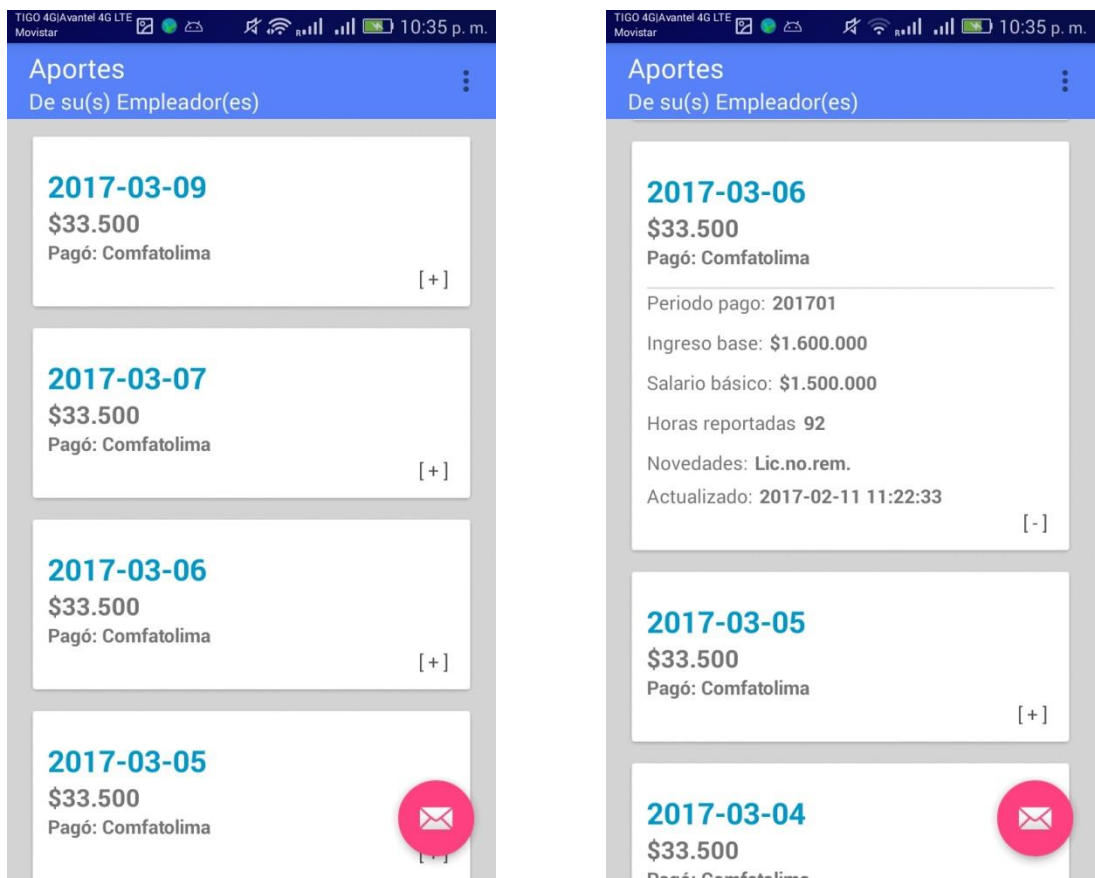
- Diagrama de clases, página 11
- Diagrama de comunicación, páginas 16 y 17
- Diagrama de secuencia, página 21 y 22
- Diagrama de actividades, página 28
- Diagrama de paquetes, página 33

4.3.2.5 Pruebas. Se realizaron pruebas unitarias a un total de 10 clases codificadas en la iteración. El detalle se puede consultar en el anexo 9 – pruebas unitarias y de integración, página 11.

Se realizaron pruebas de usabilidad y adaptación a diferentes tipos de dispositivos, utilizando los criterios establecidos en el anexo 10 – pruebas de adaptación y usabilidad.

4.3.2.6 Despliegue. A continuación se presenta la pantalla que ofrece la funcionalidad de consultar aportes:

Figura 37. Pantalla que ofrece la funcionalidad de consultar aportes



Fuente: Elaboración propia

4.3.2.7. PSP – tamaños y tiempos estimados versus reales.

Figura 38. Tamaños y tiempos estimados versus reales.

Overall Metrics

Summary	Plan	Actual	A la Fecha
Size/Hour	11,2	#VALUE!	23,2
Program Size			
Base (B)	0	0	0
Deleted (D)	0	0	0
Modified (M)	0	0	0
Added (A)	954	#VALUE!	5044
Reused (R)	0	0	3965
Added and Modified (A+M)	954	#VALUE!	5044
Total Size (T)	954	#VALUE!	9009
Total New Reusable	0	257	1669
Estimated Proxy Size (E)	954	980	4953

Time In Phase

	Plan	Actual	A la Fecha	% A la Fecha
Planeación	3:07	3:33	12:04	5,54%
Diseño	1:50	14:26	29:35	13,60%
Codificación	47:08	24:05	110:37	50,80%
Compilación	6:35	8:28	20:21	9,34%
Pruebas	24:50	4:18	42:13	19,40%
Postmortem	1:31	0:34	2:59	1,37%
Total	85:00	55:24	217:49	

Fuente: Elaboración propia

4.3.2.8 PSP – defectos detectados y removidos.

Figura 39. Defectos detectados y removidos.

Defects Injected

	Actual	A la Fecha	% A la Fecha
Antes de Desarrollo	0	0	0%
Planeación	0	0	0%
Diseño	1	26	26,30%
Codificación	11	73	73,70%
Compilación	0	0	0%
Pruebas	0	0	0%
Total	12	99	

Defects Removed

	Actual	A la Fecha	% A la Fecha
Planeación	0	0	0%
Diseño	0	5	5,05%
Codificación	3	38	38,40%
Compilación	0	15	15,20%
Pruebas	9	41	41,40%
Total	12	99	
After Development	0	0	

Fuente: Elaboración propia

4.3.2.9. Análisis de la iteración

La estimación inicial de tamaño fue de 954 LOC y el tamaño final de la iteración fue de 980 LOC, es decir que se codificó el 100.02% de lo estimado.

El tiempo estimado de desarrollo fue de 85 horas y el tiempo real fue de 55 horas, es decir el 64.7% del tiempo estimado. Esta situación obedece que en la fase anterior se codificaron clases que fueron reutilizadas en la presente fase.

Se detectaron y corrigieron 12 defectos significativos frente a 25 defectos de la fase anterior, es decir el 48%, esto se debe a que varios de los defectos de la fase anterior correspondían a clases reutilizables, por lo tanto se espera que los errores tiendan a disminuir.

Existe una gran dificultad en habituarse a registrar tiempos, interrupciones y defectos en la herramienta *ProcessDashboard*, por lo tanto el nivel de registro de la información no es total.

Todos los reportes PSP de esta iteración se pueden consultar en el anexo 12 – PSP iteración 02.

4.3.3. Consultar subsidios.

4.3.3.1. Definición de requerimientos. En esta iteración se desarrolló el caso de uso CU06 – consultar subsidios, el cual se deriva del requerimiento RE_APP_06 – consultar pago subsidios. Ver anexo 6 -especificación requerimientos, página 21. Los requerimientos no funcionales se pueden consultar en la página 25 del mismo anexo.

4.3.3.2 Casos de uso textuales.

Tabla 20. Caso de uso textual CU06.

CASO DE USO TEXTUAL	
Codigo	CU06
Nombre	Consultar Subsidios
Descripcion	Establece los procesos que se ejecutan cuando el usuario ejecuta la opción para consultar Subsidios
Actores	Afiliado a Comfatolima registrado como usuario de la aplicación móvil
Disparador	El usuario ejecuta la opción de consulta de subsidios
Precondiciones	1. Dispositivo móvil con conexión a Internet 2. Usuario afiliado a Comfatolima en estado Activo 3. Ejecución exitosa del caso de uso CU01 (ingreso al sistema)
Postcondiciones	1. Listado de subsidios pagados al afiliado visualizados en la aplicación
Flujo normal	1. El usuario ejecuta la opción de consultar Subsidios

	2. El sistema verifica si el usuario registra subsidios a su nombre 3. Si el usuario registra subsidios a su nombre (ya sea como trabajador o como cónyuge que recibe la cuota monetaria), el sistema procede a desplegarlos (ver especificación de requerimientos para la lista de datos a desplegar) 4. Los subsidios se visualizarán por mes, agrupados por el numero de cuotas y el total pagado 5. Si el usuario selecciona uno de los meses, el sistema deberá visualizar el(los) detalle(s) de los subsidios 6. Si el usuario no tiene subsidios a su nombre, el sistema le notificará esto mediante un mensaje
Flujos alternativos	1. El usuario luego de ver sus datos selecciona la opción de enviar un mensaje a la Caja, se ejecuta el caso de uso 08 (Actualizar Mensajes)
Excepciones	E1. No hay conexión a internet en el dispositivo, mostrar mensaje "El dispositivo no cuenta con conexión a internet", se visualizarán los ultimos datos consultados
Prioridad	Alta
Frecuencia de uso	Puede llegar a ser diaria, no significa conexión constante
Reglas del negocio	1. Si el usuario podrá consultar esta información solo si se encuentra en estado activo 2. El usuario no podrá modificar ni eliminar esta información, es solo de consulta

	3. Si el usuario se había logueado, pero su dispositivo no tiene conexión en el momento, le mostrará los últimos datos consultados
Requerimientos especiales	N/A
Suposiciones	N/A
Notas y preguntas	N/A

Fuente: Elaboración propia

Este caso de uso hace parte del anexo 8 – casos de uso textuales, página 10.

4.3.3.3 Prototipos de interfaces y reportes. Se crea el prototipo de la interfaz de usuario para la opción de consultar subsidios.

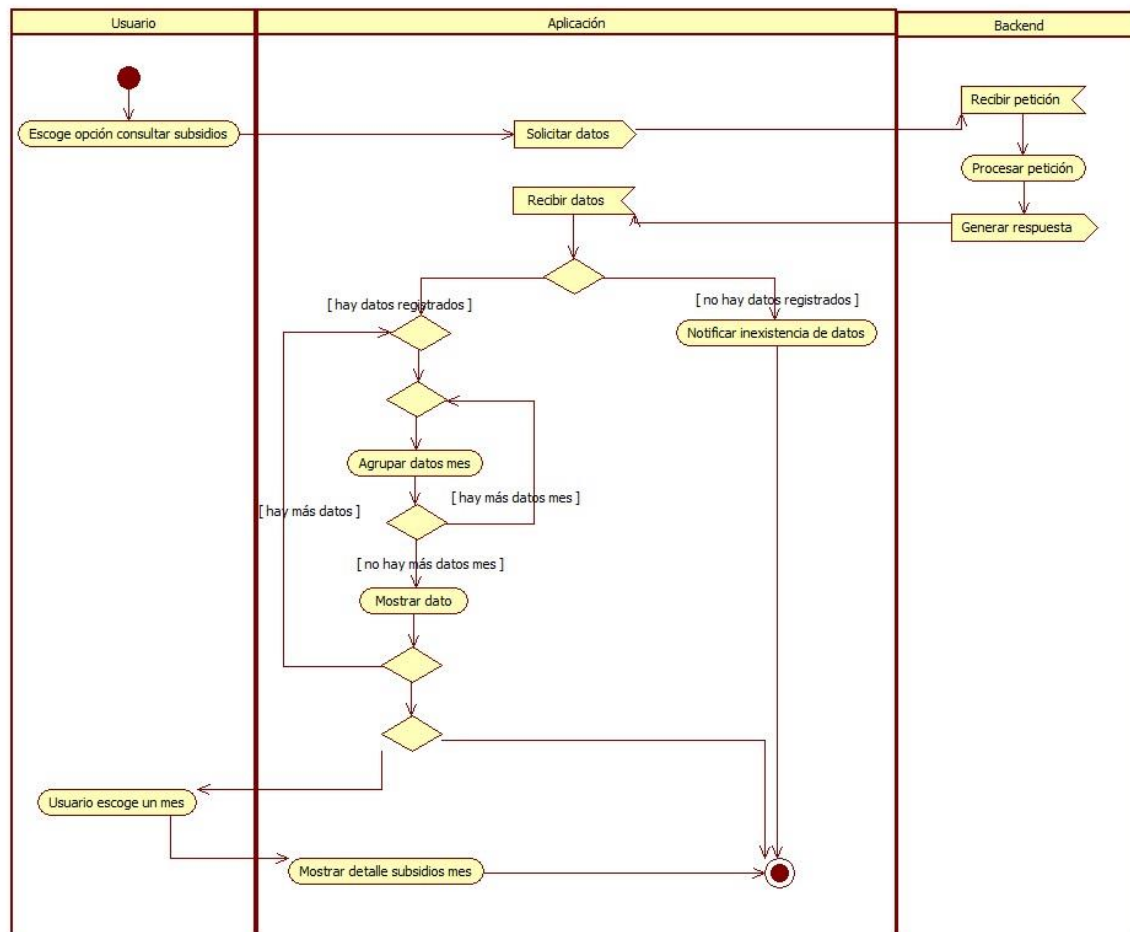
Figura 40. Prototipo funcionalidad consultar subsidios.



Fuente: Elaboración propia

4.3.3.4 Diagrama de actividades.

Figura 41. Diagrama de actividades caso de uso CU06.



Fuente: Elaboración propia

El los diagramas arquitectónicos de esta iteración pueden consultarse en el anexo 7 – documento de arquitectura, así:

- Diagrama de clases, página 12
- Diagrama de comunicación, páginas 16 y 17
- Diagrama de secuencia, página 21 y 22

- Diagrama de actividades, página 29
- Diagrama de paquetes, página 33

4.3.3.5 Pruebas. Se realizaron pruebas unitarias a un total de 8 clases codificadas en la iteración. El detalle se puede consultar en el anexo 9 – pruebas unitarias y de integración, página 17.

Se realizaron pruebas de usabilidad y adaptación a diferentes tipos de dispositivos, utilizando los criterios establecidos en el anexo 10 – pruebas de adaptación y usabilidad.

4.3.3.6 Despliegue. A continuación se presenta la pantalla que ofrece la funcionalidad de consultar subsidios:

Figura 42. La pantalla que ofrece la funcionalidad de consultar subsidios



Fuente: Elaboración propia

4.3.3.7 PSP – tamaños y tiempos estimados versus reales

Figura 43. Tamaños y tiempos estimados versus reales

Overall Metrics

Summary	Plan	Actual	A la Fecha
Size/Hour	15,8	#VALUE!	18,8
Program Size			
Base (B)	0	0	0
Deleted (D)	0	0	0
Modified (M)	0	0	0
Added (A)	1029	#VALUE!	5044
Reused (R)	0	0	3965
Added and Modified (A+M)	1029	#VALUE!	5044
Total Size (T)	1029	#VALUE!	9009
Total New Reusable	7,6	4	1673
Estimated Proxy Size (E)	1024	926	5879

Time In Phase

	Plan	Actual	A la Fecha	% A la Fecha
Planeación	3:36	4:16	16:20	6,08%
Diseño	8:50	16:04	45:39	17,00%
Codificación	33:01	21:18	131:55	49,10%
Compilación	6:04	6:45	27:06	10,10%
Pruebas	12:36	1:56	44:09	16,40%
Postmortem	0:53	0:27	3:26	1,28%
Total	65:00	50:46	268:35	

Fuente: Elaboración propia

4.3.3.8 PSP – defectos detectados y removidos.

Figura 44. Defectos detectados y removidos.

Defects Injected

	Actual	A la Fecha	% A la Fecha
Antes de Desarrollo	0	0	0%
Planeación	0	0	0%
Diseño	0	26	25,70%
Codificación	2	75	74,30%
Compilación	0	0	0%
Pruebas	0	0	0%
Total	2	101	

Defects Removed

	Actual	A la Fecha	% A la Fecha
Planeación	0	0	0%
Diseño	0	5	4,95%
Codificación	0	38	37,60%
Compilación	0	15	14,90%
Pruebas	2	43	42,60%
Total	2	101	
After Development	0	0	

Fuente: Elaboración propia

4.3.3.9. Análisis de la iteración. La estimación inicial de tamaño fue de 1024 LOC y el tamaño final de la iteración fue de 926 LOC, es decir que se codificó el 90.42% de lo estimado.

El tiempo estimado de desarrollo fue de 65 horas y el tiempo real fue de 50 horas, es decir el 76.92% del tiempo estimado. Esta situación obedece que la estructura de clases de esta iteración es muy similar a la iteración anterior (aportes), por lo tanto gran parte del código se copió y se adaptó a.

Se detectaron y corrigieron 2 defectos significativos frente a 12 defectos de la fase anterior, es decir el 16.66%, esto se debe a que esta iteración es muy similar a la anterior a nivel de código, por lo tanto al adaptar el código de la iteración anterior, todos los métodos hacen cosas similares y ya venían corregidos los defectos detectados en las pruebas. Los errores de esta fase obedecen al tema concreto de subsidio.

Todos los reportes PSP de esta iteración se pueden consultar en el anexo 13 – PSP iteración 03.

4.3.4. Consultar tarjeta.

4.3.4.1. Definición de requerimientos. En esta iteración se desarrolló el caso de uso CU07 – consultar tarjeta, el cual se deriva del requerimiento RE_APP_07 – consultar movimientos tarjeta. Ver anexo 6 -especificación requerimientos, página 22. Los requerimientos no funcionales se pueden consultar en la página 25 del mismo anexo.

4.3.4.2 Casos de uso textuales.

Tabla 21. Caso de uso textual CU07.

CASO DE USO TEXTUAL	
Codigo	CU07
Nombre	Consultar Tarjeta
Descripcion	Establece los procesos que se ejecutan cuando el usuario ejecuta la opción para consultar Tarjeta
Actores	Afiliado a Comfatolima registrado como usuario de la aplicación móvil
Disparador	El usuario ejecuta la opción de consulta de Tarjeta
Precondiciones	1. Dispositivo móvil con conexión a Internet 2. Usuario afiliado a Comfatolima en estado Activo 3. Ejecución exitosa del caso de uso CU01 (ingreso al sistema)
Postcondiciones	1. Listado de tarjetas (activas e inactivas) del usuario desplegadas en el sistema 2. Movimientos de una tarjeta específica previamente seleccionada por el usuario
Flujo normal	1. El usuario ejecuta la opción de consultar Tarjeta

	2. El sistema verifica si el usuario registra tarjetas a su nombre 3. Si el usuario registra tarjetas a su nombre (ya sea como trabajador o como cónyuge que recibe la cuota monetaria), el sistema procede a mostrar los datos de cada una de ellas (ver especificación de requerimientos para la lista de datos a desplegar) 4. Las tarjetas se visualizarán ordenadas por estado, mostrando en primer lugar las tarjetas activas 5. Si el usuario selecciona una de las tarjetas, el sistema deberá desplegar los últimos seis movimientos de esta tarjeta (ver especificación de requerimientos para la lista de datos a desplegar) 6. Si el usuario no tiene tarjetas a su nombre, el sistema le notificará esto mediante un mensaje
Flujos alternativos	1. El usuario luego de ver sus datos selecciona la opción de enviar un mensaje a la Caja, se ejecuta el caso de uso 08 (Actualizar Mensajes)
Excepciones	E1. No hay conexión a internet en el dispositivo, mostrar mensaje "El dispositivo no cuenta con conexión a internet", se visualizarán los últimos datos consultados
Prioridad	Media
Frecuencia de uso	Puede llegar a ser mensual, no significa conexión constante
Reglas del negocio	1. El usuario podrá consultar esta información solo si se encuentra en estado activo

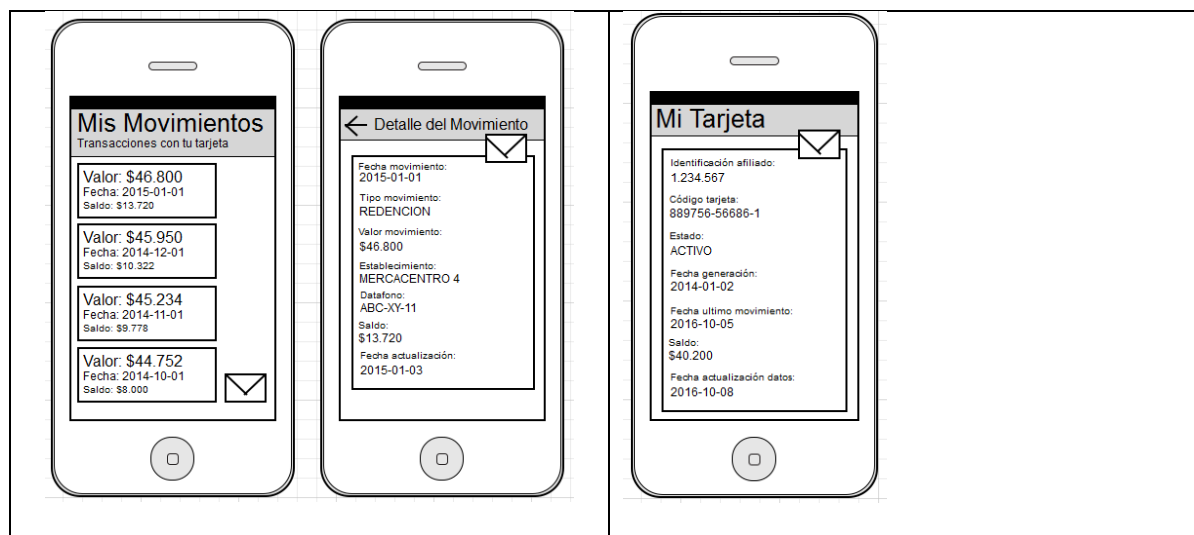
	2. El usuario no podrá modificar ni eliminar esta información, es solo de consulta 3. Si el usuario se había logueado, pero su dispositivo no tiene conexión en el momento, le mostrará los últimos datos consultados
Requerimientos especiales	N/A
Suposiciones	N/A
Notas y preguntas	N/A

Fuente: Elaboración propia

Este caso de uso hace parte del anexo 8 – casos de uso textuales, página 11.

4.3.4.3 Prototipos de interfaces y reportes. Se crea el prototipo de la interfaz de usuario para la opción de consultar tarjeta.

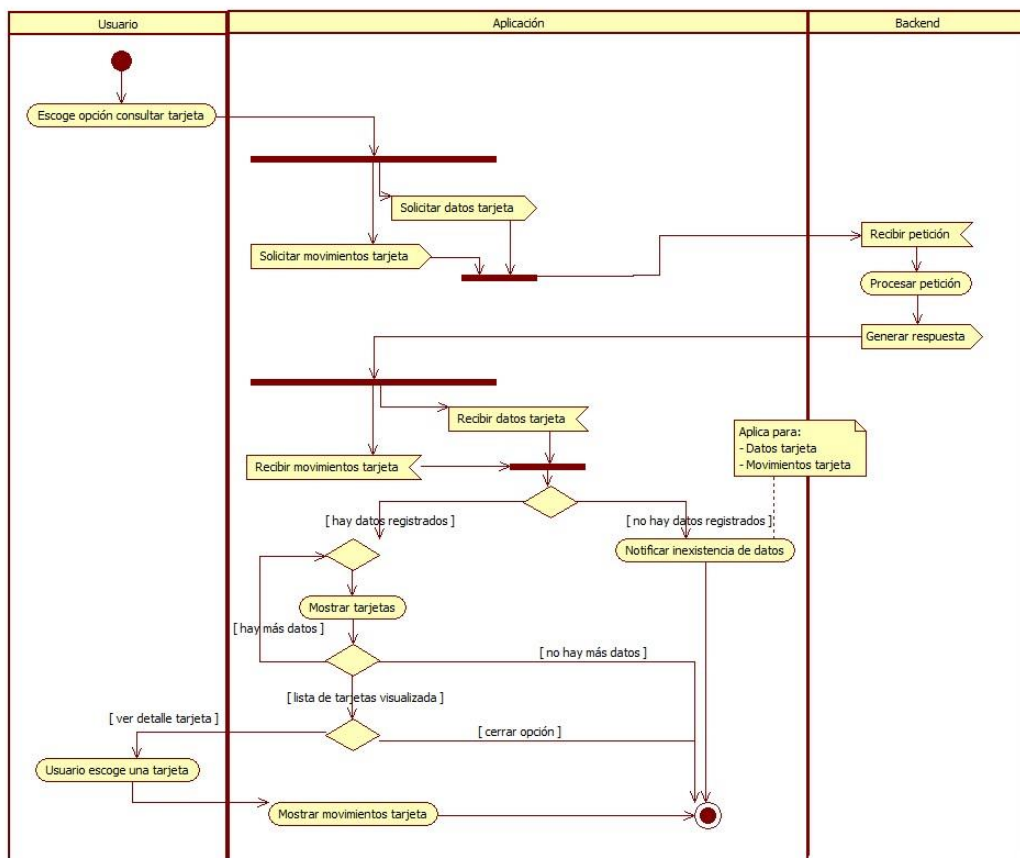
Figura 45. Prototipo funcionalidad consultar tarjeta.



Fuente: Elaboración propia

4.3.4.4 Diagrama de actividades.

Figura 46 diagrama de actividades caso de uso CU07.



Fuente: Elaboración propia

El los diagramas arquitectónicos de esta iteración pueden consultarse en el anexo 7 – documento de arquitectura, así:

- Diagrama de clases, página 13
- Diagrama de comunicación, páginas 16 y 17
- Diagrama de secuencia, página 21 y 22
- Diagrama de actividades, página 30

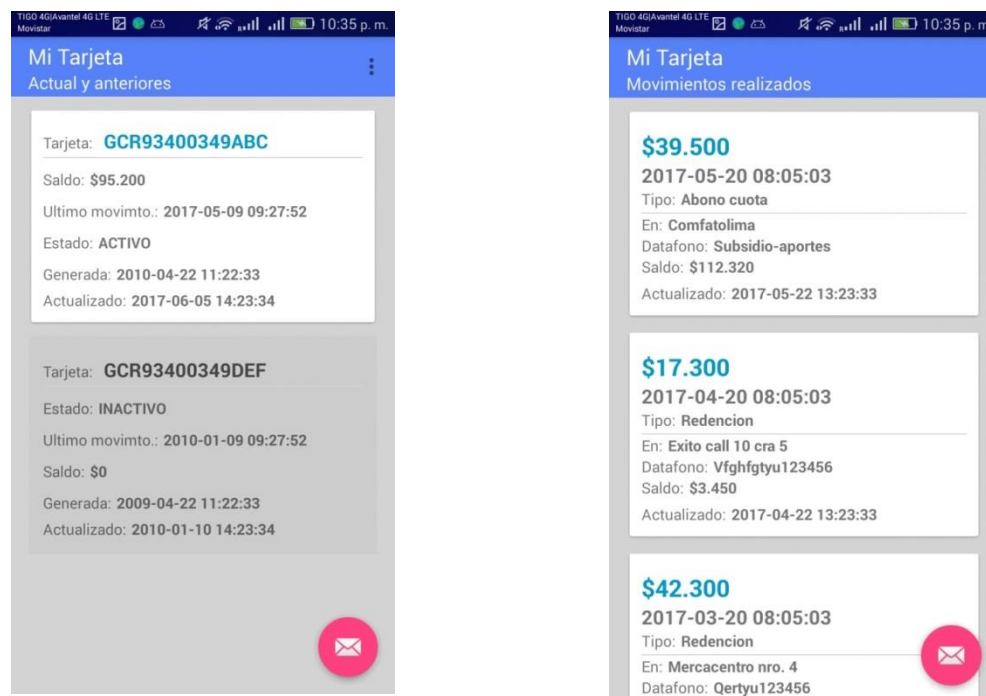
- Diagrama de paquetes, página 33

4.3.4.5 Pruebas. Se realizaron pruebas unitarias a un total de 8 clases codificadas en la iteración. El detalle se puede consultar en el anexo 9 – pruebas unitarias y de integración, página 19.

Se realizaron pruebas de usabilidad y adaptación a diferentes tipos de dispositivos, utilizando los criterios establecidos en el anexo 9 – pruebas de adaptación y usabilidad.

4.3.4.6 Despliegue. A continuación se presenta la pantalla que ofrece la funcionalidad de consultar tarjeta:

Figura 47. Pantalla que ofrece la funcionalidad de consultar tarjeta



Fuente: Elaboración propia

4.3.4.7 PSP – tamaños y tiempos estimados versus reales.

Figura 48. Tamaños y tiempos estimados versus reales.

Overall Metrics

Summary	Plan	Actual	A la Fecha
<i>Size/Hour</i>	20	#VALUE!	15,8
Program Size			
Base (B)	0	0	0
Deleted (D)	0	0	0
Modified (M)	0	0	0
Added (A)	1302	#VALUE!	5044
Reused (R)	0	0	3965
Added and Modified (A+M)	1302	#VALUE!	5044
Total Size (T)	1302	#VALUE!	9009
Total New Reusable	0	0	1673
<i>Estimated Proxy Size (E)</i>	1292	1000	6879

Time In Phase

	Plan	Actual	A la Fecha	% A la Fecha
Planeación	3:57	3:50	20:10	6,32%
Diseño	11:03	17:27	63:06	19,80%
Codificación	31:56	20:01	151:56	47,60%
Compilación	6:34	6:57	34:03	10,70%
Pruebas	10:41	1:51	46:00	14,40%
Postmortem	0:50	0:31	3:57	1,24%
Total	65:00	50:37	319:12	

Fuente: Elaboración propia

4.3.4.8 PSP – defectos detectados y removidos.

Figura 49. Defectos detectados y removidos.

Defects Injected

	Actual	A la Fecha	% A la Fecha
Antes de Desarrollo	0	0	0%
Planeación	0	0	0%
Diseño	0	26	25,20%
Codificación	2	77	74,80%
Compilación	0	0	0%
Pruebas	0	0	0%
Total	2	103	

Defects Removed

	Actual	A la Fecha	% A la Fecha
Planeación	0	0	0%
Diseño	0	5	4,85%
Codificación	2	40	38,80%
Compilación	0	15	14,60%
Pruebas	0	43	41,70%
Total	2	103	
After Development	0	0	

Fuente: Elaboración propia

4.3.4.9. Análisis de la iteración. La estimación inicial de tamaño fue de 1292 LOC y el tamaño final de la iteración fue de 1000 LOC, es decir que se codificó el 77.4% de lo estimado.

El tiempo estimado de desarrollo fue de 65 horas y el tiempo real fue de 50 horas, es decir el 76.92% del tiempo estimado. Esta situación obedece que la estructura de clases de esta iteración es muy similar a la iteración anterior (subsídios), por lo tanto gran parte del código se copió y se adaptó a.

Se detectaron y corrigieron 2 defectos significativos frente a 2 defectos de la fase anterior, es decir que no hubo variación, esto se debe a que esta iteración es muy similar a la anterior a nivel de código, por lo tanto al adaptar el código de la iteración anterior, todos los métodos hacen cosas similares y ya venían corregidos los defectos detectados en las pruebas. Los errores de esta fase obedecen al tema concreto de tarjeta.

Todos los reportes PSP de esta iteración se pueden consultar en el anexo 14 – PSP iteración 04.

4.3.5. Consultar datos familia.

4.3.5.1. Definición de requerimientos. En esta iteración se desarrolló el caso de uso CU03 – consultar datos familia, el cual se deriva del requerimiento RE_APP_03 – consultar datos familia. Ver anexo 6 -especificación requerimientos, página 16. Los requerimientos no funcionales se pueden consultar en la página 25 del mismo anexo.

4.3.5.2. Casos de uso textuales

Tabla 22. Caso de uso textual CU03.

CASO DE USO TEXTUAL	
Codigo	CU03
Nombre	Consultar datos Familia
Descripcion	Establece los procesos que se ejecutan cuando el usuario ejecuta la opción de consultar datos del núcleo familiar
Actores	Afiliado a Comfatolima registrado como usuario de la aplicación móvil
Disparador	El usuario ejecuta la opción de consulta de datos de familia
Precondiciones	1. Dispositivo móvil con conexión a Internet 2. Usuario afiliado a Comfatolima en estado Activo 3. Ejecución exitosa del caso de uso CU01 (ingreso al sistema)
Postcondiciones	1. Visualización de datos del trabajador (si los hay), conyuge (si los hay) y beneficiarios (si los hay)
Flujo normal	1. El usuario ejecuta la opción de consultar Familia

2. El sistema verifica si el usuario registra datos de afiliación, datos de conyuge(s) y datos de beneficiarios(s)
3. Si el usuario es trabajador, el sistema muestra sus datos (ver especificación de requerimientos para la lista de datos a desplegar)
4. Si el usuario es trabajador y conyuge a la vez, el sistema muestra sus datos (ver especificación de requerimientos para la lista de datos a desplegar)
5. Si el usuario es trabajador y tiene conyuges(s), el sistema muestra sus datos (ver especificación de requerimientos para la lista de datos a desplegar)
6. Si el usuario es trabajador y tiene beneficiarios(s), el sistema muestra sus datos (ver especificación de requerimientos para la lista de datos a desplegar)
7. Si el usuario es conyuge, el sistema muestra sus datos pero no los de los trabajadores asociados (ver especificación de requerimientos para la lista de datos a desplegar), esto dando cumplimiento a la ley de protección de datos de usuario
8. Si el usuario es conyuge y tiene beneficiarios, el sistema muestra sus datos (ver especificación de requerimientos para la lista de datos a desplegar)
9. Para cada una de las listas de datos desplegadas, si no registra datos, el sistema deberá notificar mediante mensaje

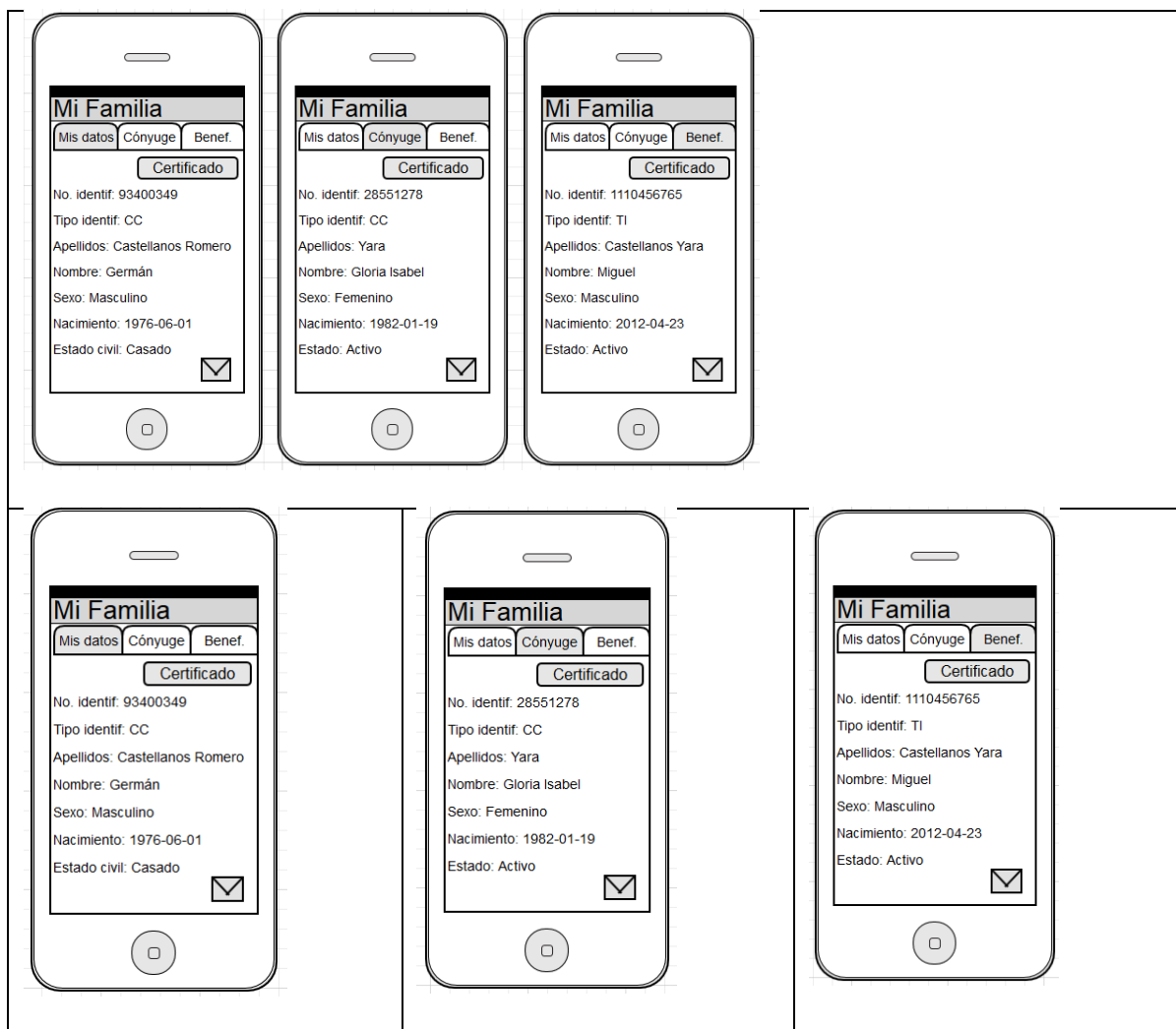
Flujos alternativos	1. El usuario luego de ver sus datos selecciona la opción de enviar un mensaje a la Caja, se ejecuta el caso de uso 08 (Actualizar Mensajes)
Excepciones	E1. No hay conexión a internet en el dispositivo, mostrar mensaje "El dispositivo no cuenta con conexión a internet", se visualizarán los últimos datos consultados
Prioridad	Alta
Frecuencia de uso	Puede llegar a ser diaria, no significa conexión constante
Reglas del negocio	1. Si el usuario podrá consultar esta información solo si se encuentra en estado activo 2. El usuario no podrá modificar ni eliminar esta información, es solo de consulta 3. Si el usuario se había logueado, pero su dispositivo no tiene conexión en el momento, le mostrará los últimos datos consultados
Requerimientos especiales	Para el caso de las conyuges, estas no podrán ver los datos de afiliación del trabajador o los trabajadores que le hayan afiliado, solo verá los beneficiarios que tiene registrados a su nombre y solo verá los subsidios a su nombre (proteccion y confidencialidad de datos de usuario)
Suposiciones	N/A
Notas y preguntas	N/A

Fuente: elaboración propia

Este caso de uso hace parte del anexo 8 – casos de uso textuales, página 5.

4.3.5.3. Prototipos de interfaces y reportes. Se crea el prototipo de la interfaz de usuario para la opción de consultar datos familia.

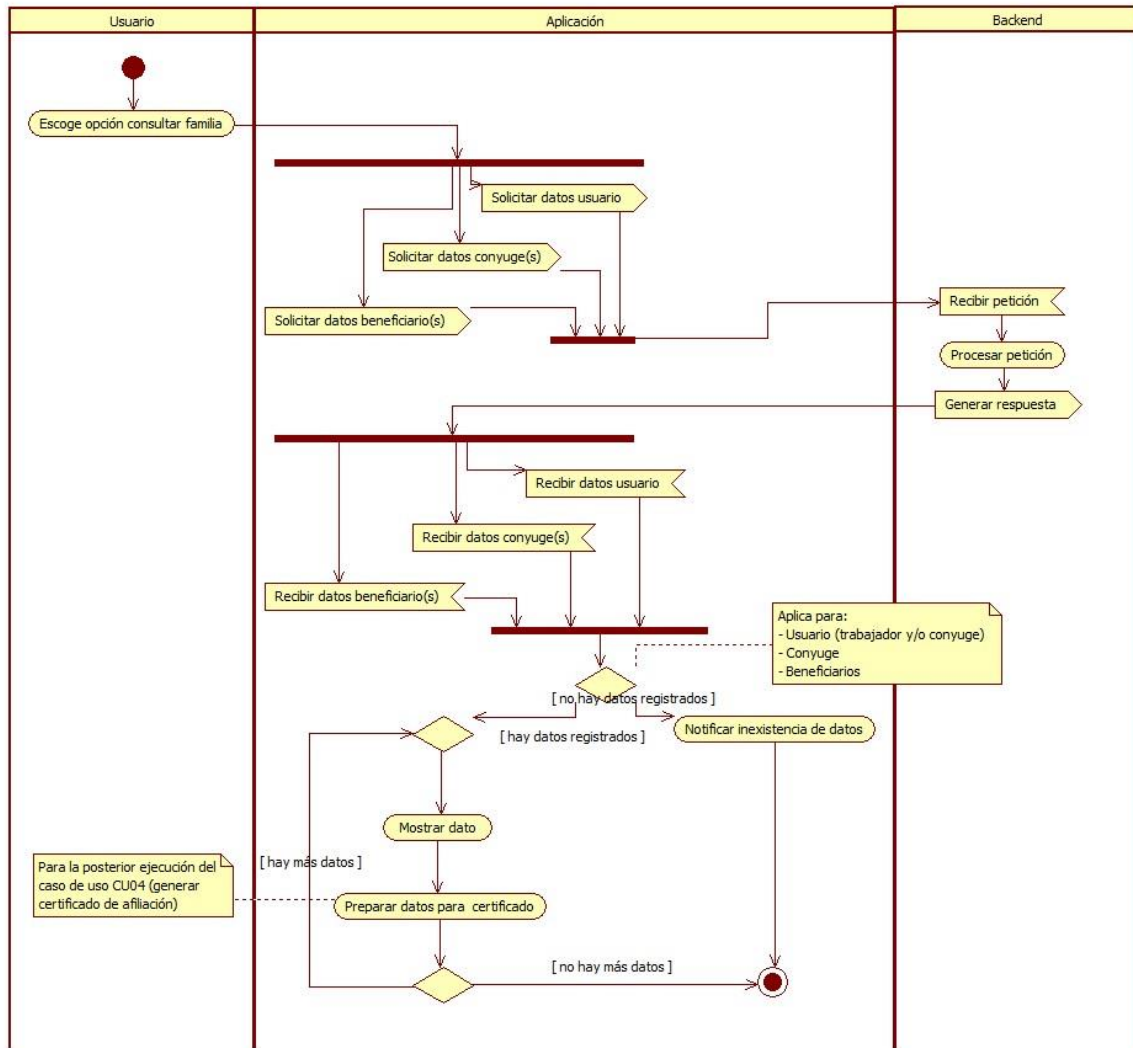
Figura 50. Prototipo funcionalidad consultar datos familia.



Fuente: elaboración propia

4.3.5.4. Diagrama de actividades.

Figura 51. Diagrama de actividades caso de uso CU03.



Fuente: elaboración propia

El los diagramas arquitectónicos de esta iteración pueden consultarse en el anexo 7 – documento de arquitectura, así:

- Diagrama de clases, páginas 6, 7 y 8
- Diagrama de comunicación, páginas 16 y 17

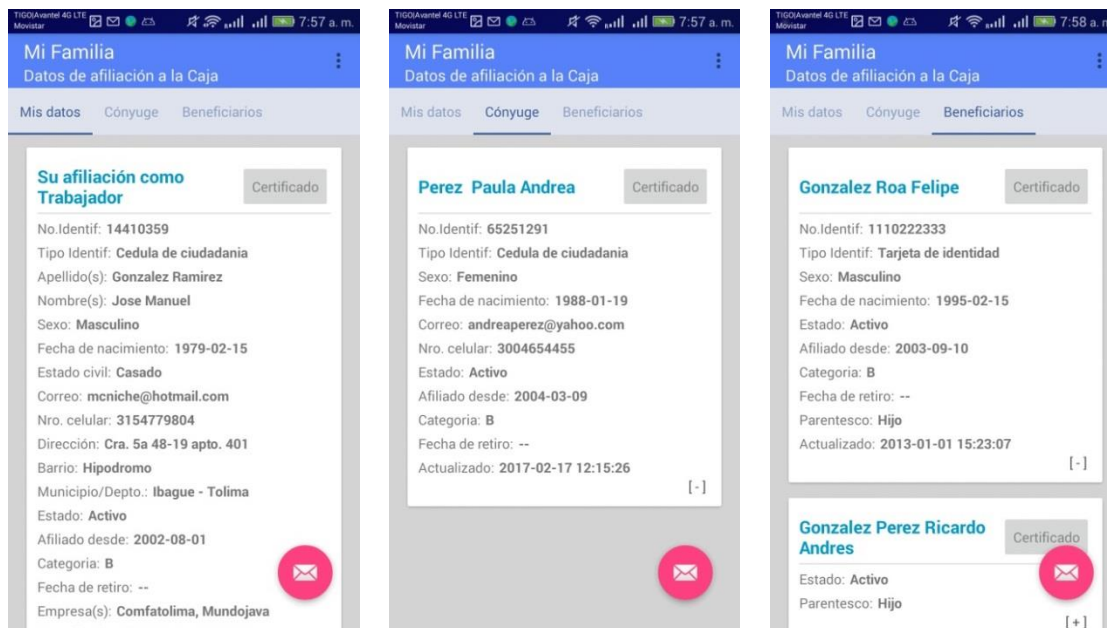
- Diagrama de secuencia, página 21 y 22
- Diagrama de actividades, página 26
- Diagrama de paquetes, página 33

4.3.5.5 .Pruebas. Se realizaron pruebas unitarias a un total de 15 clases codificadas en la iteración. El detalle se puede consultar en el anexo 9 – pruebas unitarias y de integración, página 22.

Se realizaron pruebas de usabilidad y adaptación a diferentes tipos de dispositivos, utilizando los criterios establecidos en el anexo 10 – pruebas de adaptación y usabilidad.

4.3.5.6. Despliegue. A continuación se presenta la pantalla que ofrece la funcionalidad de consultar datos familia:

Figura 52. Pantalla que ofrece la funcionalidad de consultar datos familia.



Fuente: elaboración propia

4.3.5.7 PSP – tamaños y tiempos estimados versus reales.

Figura 53. Tamaños y tiempos estimados versus reales.

Overall Metrics

Summary	Plan	Actual	A la Fecha
<i>Size/Hour</i>	25,2	#VALUE!	15,8
Program Size			
Base (B)	0	0	0
Deleted (D)	0	0	0
Modified (M)	0	0	0
Added (A)	1637	#VALUE!	5044
Reused (R)	0	0	3965
Added and Modified (A+M)	1637	#VALUE!	5044
Total Size (T)	1637	#VALUE!	9009
Total New Reusable	125	126	1673
<i>Estimated Proxy Size (E)</i>	1637	2113	6879

Time In Phase

	Plan	Actual	A la Fecha	% A la Fecha
Planeación	4:06	3:06	20:10	6,32%
Diseño	12:51	14:27	63:06	19,80%
Codificación	30:56	34:15	151:56	47,60%
Compilación	6:56	7:46	34:03	10,70%
Pruebas	9:22	6:28	46:00	14,40%
Postmortem	0:48	1:09	3:57	1,24%
Total	65:00	67:11	319:12	

Fuente: elaboración propia

4.3.5.8. PSP – defectos detectados y removidos.

Figura 54. Defectos detectados y removidos.

Defects Injected

	Actual	A la Fecha	% A la Fecha
Antes de Desarrollo	0	0	0%
Planeación	0	0	0%
Diseño	0	26	25,20%
Codificación	3	77	74,80%
Compilación	0	0	0%
Pruebas	0	0	0%
Total	3	103	

Defects Removed

	Actual	A la Fecha	% A la Fecha
Planeación	0	0	0%
Diseño	0	5	4,85%
Codificación	3	40	38,80%
Compilación	0	15	14,60%
Pruebas	0	43	41,70%
Total	3	103	
After Development	0	0	

Fuente: elaboración propia

4.3.5.9. Análisis de la iteración. La estimación inicial de tamaño fue de 1637 LOC y el tamaño final de la iteración fue de 2113 LOC, es decir que se codificó el 129% de lo estimado. Esto debido al tamaño de la opción, que implicaba ejecutar tres vista en forma coordinada y detalle de cada una de ellas.

El tiempo estimado de desarrollo fue de 65 horas y el tiempo real fue de 68 horas, es decir el 100.04% del tiempo estimado. Situación que contrasta con el tamaño en LOC de la iteración.

Se detectaron y corrigieron 3 defectos significativos frente a 2 defectos de la fase anterior, esto se debe a que esta iteración, si bien es más grande que la anterior, es muy similar a nivel de código, por lo tanto al adaptar el código de la iteración anterior, todos los métodos hacen cosas similares y ya venían corregidos los defectos detectados en las pruebas. Adicional a esto, persiste resistencia a registrar con precisión los errores y los tiempos en el *software* destinado a ese fin, por lo tanto pueden existir errores que se hayan solucionado durante la codificación pero no se hayan registrado.

Todos los reportes PSP de esta iteración se pueden consultar en el anexo 15 – PSP iteración 05.

4.3.6. Generar certificados de afiliación.

4.3.6.1. Definición de requerimientos. En esta iteración se desarrolló el caso de uso CU04 – generar certificados de afiliación, el cual se deriva del requerimiento RE_APP_04 – generar certificados de afiliación. Ver anexo 6 -especificación requerimientos, página 18. Los requerimientos no funcionales se pueden consultar en la página 25 del mismo anexo.

4.3.6.2 Casos de uso textuales.

Tabla 23. Caso de uso textual CU04.

CASO DE USO TEXTUAL	
Codigo	CU04
Nombre	Generar certificados afiliación
Descripcion	Establece los procesos que se ejecutan cuando el usuario ejecuta la opción de generar certificado de afiliación
Actores	Afiliado a Comfatolima registrado como usuario de la aplicación móvil
Disparador	El usuario ejecuta la opción de generar certificado de afiliación para uno de los datos visualizados en la opcion del caso de uso 03 (consultar datos familia)
Precondiciones	1. Dispositivo móvil con conexión a Internet 2. Usuario afiliado a Comfatolima en estado Activo 3. Ejecución exitosa del caso de uso CU03 (consultar datos de familia) con existencia de datos y su correspondiente visualización
Postcondiciones	1. Certificado de afiliación a la caja visualizado en pantalla o enviado por correo electrónico

Flujo normal	1. El usuario ejecuta la opción de generar certificado de afiliación 2. El sistema verifica si el dispositivo cuenta con conexión de a Internet (E1) 3. Si el afiliado sobre el cual se intenta generar el certificado está inactivo, el sistema no generará certificado y notificará al usuario (E2) 4. El sistema generará un consecutivo de seguridad para el certificado, el cual podrá ser confirmado con el administrador de la aplicación 5. Si el proceso se realiza desde un usuario que corresponde a un trabajador afiliado a la caja, cada uno de los certificados que se generen desde este (certificados para trabajador, conyuges y beneficiarios), visualizarán además los datos del trabajador que los tiene afiliados a la caja 6. Si el proceso se realiza desde un usuario que corresponde a un conyuge afiliado a la caja, se podrá generar su propio certificado y los de los beneficiarios que registre, pero en estos certificados no se visualizarán los datos del trabajador que los tiene afiliados a la caja, esto dando cumplimiento a la ley de protección de datos de usuario 7. Los certificados que se generen a nombre del trabajador o del conyuge, no contendrán datos del núcleo familiar, por cuanto un trabajador puede tener hijos con varias cónyuges y esta información solo le interesa al afiliado, mostrar todos los beneficiarios que tiene estaría presentando un riesgo de confidencialidad de la información. Por lo anterior cada certificado del trabajador, conyuge, beneficiario serán individuales, dando cumplimiento a la ley de protección de datos de usuario

	8. Una vez generado el certificado, el usuario podrá elegir si lo visualiza y lo guarda en el dispositivo, o si lo envía por correo electrónico
Flujos alternativos	N/A
Excepciones	E1. No hay conexión a internet en el dispositivo, mostrar mensaje "El dispositivo no cuenta con conexión a internet", se visualizarán los últimos datos consultados E2. Mensaje "solo se generan certificados a afiliados activos"
Prioridad	Baja
Frecuencia de uso	Esporádica
Reglas del negocio	1. El usuario podrá consultar esta información solo si se encuentra en estado activo 2. El usuario no podrá modificar ni eliminar esta información, es solo de consulta 3. Si el usuario se había logueado, pero su dispositivo no tiene conexión en el momento, le mostrará los últimos datos consultados pero no podrá generar certificados
Requerimientos especiales	Dar prelación a la ley de protección de datos de usuario, para esto solo se generarán certificados individuales y solo tendrán datos del trabajador aquellos certificados que se generen desde un usuario trabajador, de lo contrario solo se mostrarán los datos de afiliación del beneficiario al cual se le está generando el

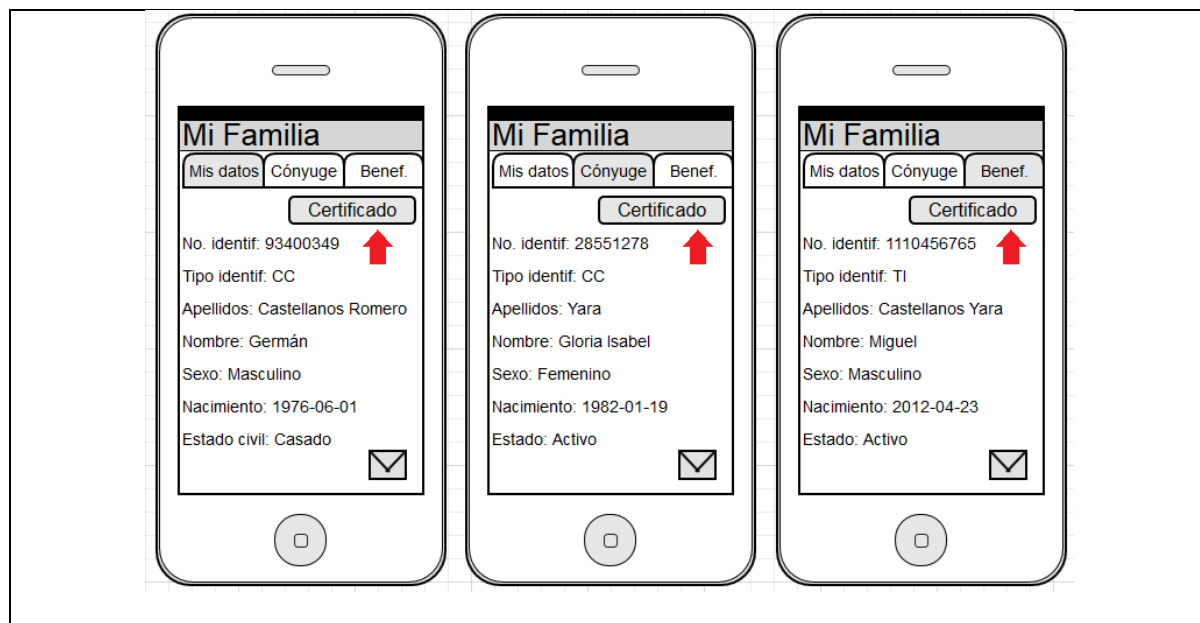
	certificado
Suposiciones	N/A
Notas y preguntas	N/A

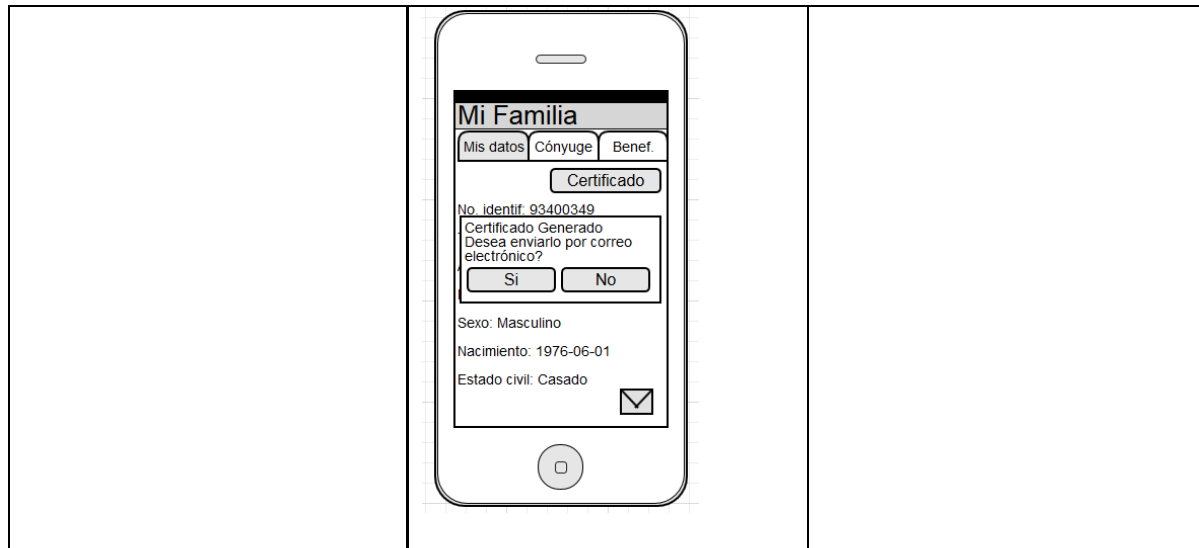
Fuente: Elaboración propia

Este caso de uso hace parte del anexo 8 – casos de uso textuales, página 7.

4.3.6.3 Prototipos de interfaces y reportes. Se crea el prototipo de la interfaz de usuario para la opción de generar certificados de afiliación.

Figura 55. Prototipo funcionalidad generar certificados.

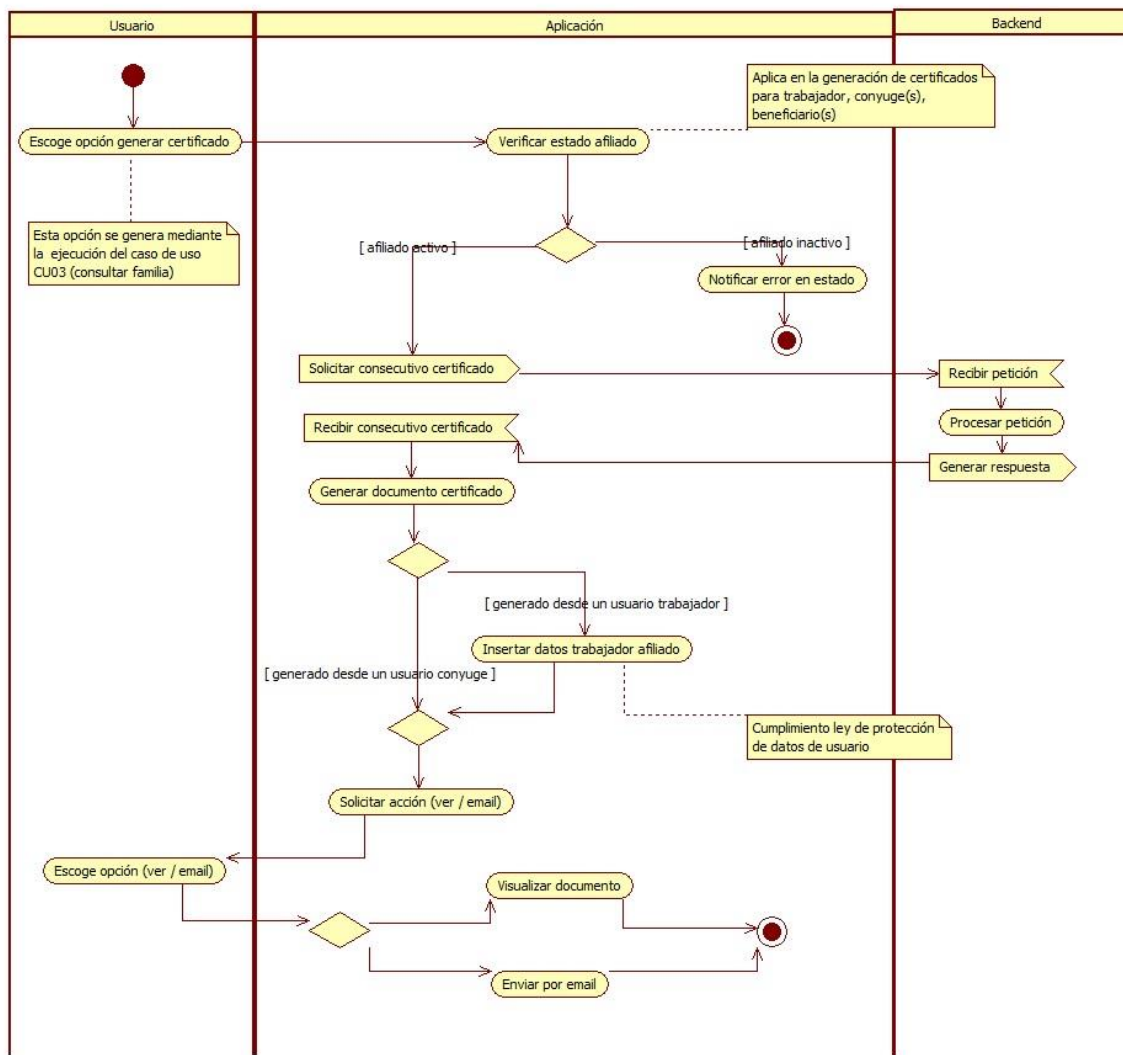




Fuente: Elaboración propia

4.3.6.4. Diagrama de actividades.

Figura 56. Diagrama de actividades caso de uso CU04.



Fuente: Elaboración propia

El los diagramas arquitectónicos de esta iteración pueden consultarse en el anexo 7 – documento de arquitectura, así:

- Diagrama de clases, página 10
- Diagrama de comunicación, página 17
- Diagrama de secuencia, página 23

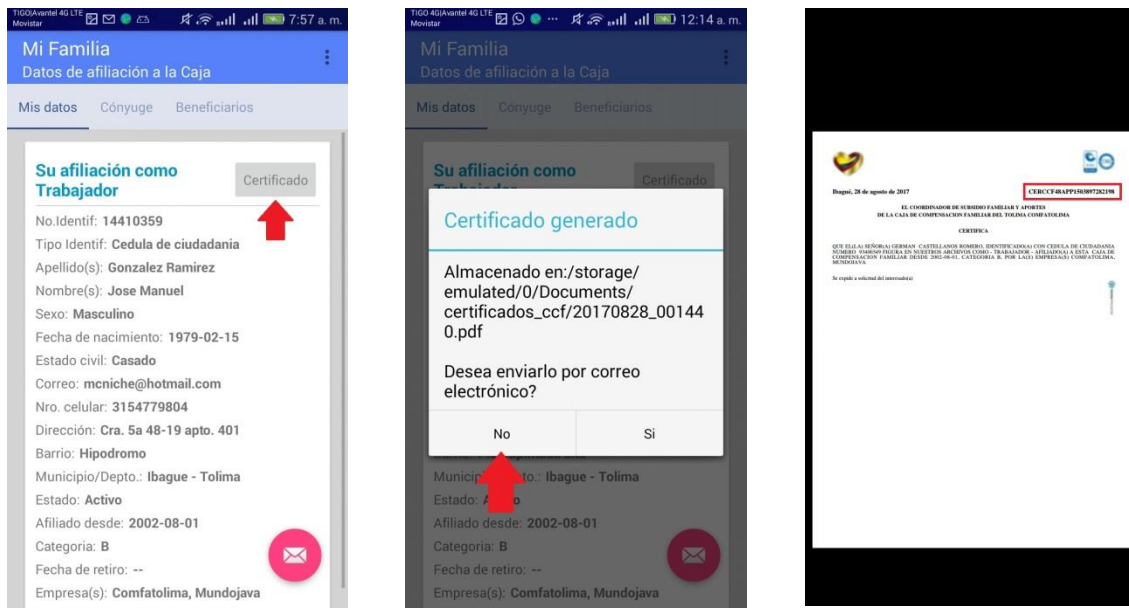
- Diagrama de actividades, página 27
- Diagrama de paquetes, página 35

4.3.6.5. Pruebas. Se realizaron pruebas unitarias a un total de 12 clases codificadas en la iteración. El detalle se puede consultar en el anexo 9 – pruebas unitarias y de integración, página 27.

Se realizaron pruebas de usabilidad y adaptación a diferentes tipos de dispositivos, utilizando los criterios establecidos en el anexo 10 – pruebas de adaptación y usabilidad.

4.3.6.6. Despliegue. A continuación se presentan las pantallas que ofrece la funcionalidad de generar certificados de afiliación:

Figura 57. Pantallas que ofrece la funcionalidad de generar certificados de afiliación.



Fuente: Elaboración propia

4.3.6.7 PSP – tamaños y tiempos estimados versus reales.

Figura 58. Tamaños y tiempos estimados versus reales.

Overall Metrics

Summary	Plan	Actual	A la Fecha
<i>Size/Hour</i>	24,2	#VALUE!	11,4
Program Size			
Base (B)	695	0	0
Deleted (D)	0	0	0
Modified (M)	0	0	0
Added (A)	970	#VALUE!	5044
Reused (R)	0	0	3965
Added and Modified (A+M)	970	#VALUE!	5044
Total Size (T)	1665	#VALUE!	9009
Total New Reusable	0	105	1904
<i>Estimated Proxy Size (E)</i>	970	1005	9997

Time In Phase

	Plan	Actual	A la Fecha	% A la Fecha
Planeación	2:25	2:14	25:30	5,76%
Diseño	8:02	7:43	85:16	19,30%
Codificación	19:16	35:22	221:33	50,10%
Compilación	4:20	4:59	46:48	10,60%
Pruebas	5:26	5:33	58:01	13,10%
Postmortem	0:32	0:18	5:24	1,22%
Total	40:00	56:09	442:32	

Fuente: Elaboración propia

4.3.6.8 PSP – defectos detectados y removidos.

Figura 59. Defectos detectados y removidos.

Defects Injected

	Actual	A la Fecha	% A la Fecha
Antes de Desarrollo	0	0	0%
Planeación	0	0	0%
Diseño	0	26	22,80%
Codificación	8	88	77,20%
Compilación	0	0	0%
Pruebas	0	0	0%
Total	8	114	

Defects Removed

	Actual	A la Fecha	% A la Fecha
Planeación	0	0	0%
Diseño	0	5	4,39%
Codificación	0	43	37,70%
Compilación	0	15	13,20%
Pruebas	8	51	44,70%
Total	8	114	
After Development	0	0	

Fuente: Elaboración propia

4.3.6.9 Análisis de la iteración. La estimación inicial de tamaño fue de 970 LOC y el tamaño final de la iteración fue de 1005 LOC, es decir que se codificó el 103.6% de lo estimado. Esto debido al tamaño de la opción, que implicaba codificar clases que interactuaran con el

sistema operativo, en cuanto a la generación de archivos, sistema de almacenamiento y envío de correos electrónicos desde la cuenta configurada en el dispositivo.

El tiempo estimado de desarrollo fue de 40 horas y el tiempo real fue de 56 horas, es decir el 140% del tiempo estimado. Esto debido a que es una iteración diferente a las que se han venido desarrollando, que consisten en consultas de datos. Por esta razón no ha sido posible reutilizar y adaptar mucho código y esto se ve reflejado en el tiempo.

Se detectaron y corrigieron 8 defectos significativos frente a 3 defectos de la fase anterior. En las iteraciones se están registrando errores que impliquen la interrupción del programa u otros efectos críticos. Errores en distribución de espacios en vistas, valores nulos y demás no se están registrando.

Todos los reportes PSP de esta iteración se pueden consultar en el anexo 16 – PSP iteración 06.

4.3.7. Actualizar mensajes.

4.3.7.1. Definición de requerimientos. En esta iteración se desarrolló el caso de uso CU08 – actualizar mensajes, el cual se deriva del requerimiento RE_APP_08 – enviar mensaje personalizado. Ver anexo 6 -especificación requerimientos, página 24. Los requerimientos no funcionales se pueden consultar en la página 25 del mismo anexo.

4.3.7.2. Casos de uso textuales.

Tabla 24. Caso de uso textual CU08_01.

CASO DE USO TEXTUAL	
Codigo	CU08_01
Nombre	Redactar Mensaje
Descripcion	Establece los procesos que se ejecutan cuando el usuario envía un mensaje al administrador desde la aplicación móvil
Actores	Afiliado a Comfatolima registrado como usuario de la aplicación móvil
Disparador	El usuario ejecuta la opción de enviar mensaje
Precondiciones	1. Dispositivo móvil con conexión a Internet 2. Usuario afiliado a Comfatolima en estado Activo 3. Ejecución exitosa de uno de los casos de uso CU03, CU04, CU05, CU06, CU07, desde los cuales el usuario podrá enviar un mensaje si tiene alguna inquietud acerca de la información o del servicio de la Caja
Postcondiciones	1. Mensaje enviado al administrador de la aplicación
Flujo normal	1. El usuario ejecuta la opción de enviar mensaje

	2. El sistema despliega la opción para enviar un mensaje 3. El usuario ingresa el asunto del mensaje (E1) 4. El usuario ingresa el texto del mensaje (E2) 5. Opcionalmente el usuario adjunta hasta un archivo al mensaje (E3) (E4) 6. El usuario envía el mensaje
Flujos alternativos	1. El usuario cancela el envío del mensaje, se interrumpe el proceo y se regresa a la opción desde donde se invocó la opción
Excepciones	E1. El usuario no ingresa asunto, el sistema le notificará que debe ingresar el dato E2. El usuario no ingresa texto del mensaje, el sistema le notificará que debe ingresar el texto del mensaje E3. El archivo adjunto no corresponde al formato permitido (pdf, jpg, jpeg, png), notificar al usuario los tipos de archivos permitidos E4. El archivo adjunto es de un tamaño mayor al permitido (2MB), notificar al usuario el tamaño máximo permitido
Prioridad	Alta
Frecuencia de uso	Puede llegar a ser semanal, no significa conexión constante
Reglas del negocio	
Requerimientos	N/A

especiales	
Suposiciones	N/A
Notas y preguntas	N/A

Fuente: Elaboración propia

Tabla 25. Caso de uso textual CU08_02.

CASO DE USO TEXTUAL	
Codigo	CU08_02
Nombre	Leer Mensaje
Descripcion	Establece los procesos que se ejecutan cuando el usuario lee un mensaje
Actores	Afiliado a Comfatolima registrado como usuario de la aplicación móvil
Disparador	El usuario ejecuta la opción de leer un mensaje
Precondiciones	1. Dispositivo móvil con conexión a Internet 2. Usuario afiliado a Comfatolima en estado Activo
Postcondiciones	1. Mensaje entrante desplegado en pantalla
Flujo normal	1. El usuario ejecuta la opción de mensajes 2. El sistema despliega la lista de mensajes entrantes 3. El usuario selecciona un mensaje entrante 4. El mensaje entrante se despliega en pantalla 5. Si el usuario no tiene mensajes entrantes, el sistema de informará esto 6. El usuario podrá eliminar el mensaje entrante 7. El usuario podrá filtrar filtrar los mensajes entrantes por: todos y no

	leídos (por defecto)
Flujos alternativos	N/A
Excepciones	
Prioridad	Alta
Frecuencia de uso	Puede llegar a ser semanal, no significa conexión constante
Reglas del negocio	
Requerimientos especiales	N/A
Suposiciones	N/A
Notas y preguntas	N/A

Fuente: Elaboración propia

Tabla 26. Caso de uso textual CU08_03.

CASO DE USO TEXTUAL	
Codigo	CU08_03
Nombre	Responder Mensaje
Descripcion	Establece los procesos que se ejecutan cuando el usuario responde un mensaje entrante
Actores	Afiliado a Comfatolima registrado como usuario de la aplicación móvil
Disparador	El usuario ejecuta la opción de responder un mensaje entrante
Precondiciones	1. Dispositivo móvil con conexión a Internet 2. Usuario afiliado a Comfatolima en estado Activo 3. Caso uso CU08_02 (leer mensajes) ejecutando correctamente y con datos

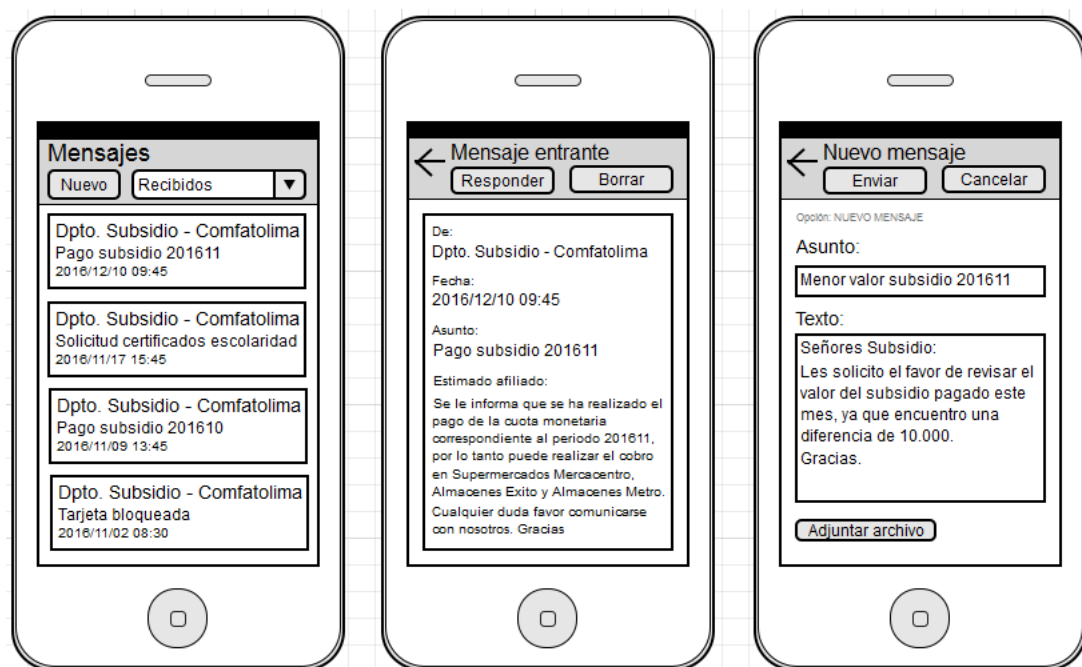
	existentes 4. Mensaje entrante visualizado en pantalla
Postcondiciones	1. Mensaje enviado al administrador
Flujo normal	1. El usuario ejecuta la opción de responder mensaje 2. Se ejecutan al pie de la letra los mismos pasos, excepciones y restricciones del caso de uso CU08_01 (enviar mensaje)
Flujos alternativos	N/A
Excepciones	
Prioridad	Alta
Frecuencia de uso	Puede llegar a ser semanal, no significa conexión constante
Reglas del negocio	
Requerimientos especiales	N/A
Suposiciones	N/A
Notas y preguntas	N/A

Fuente: Elaboración propia

Estos casos de uso hacen parte del anexo 8 – casos de uso textuales, página 12.

4.3.7.3 Prototipos de interfaces y reportes. Se crea el prototipo de la interfaz de usuario para la opción de generar certificados de afiliación.

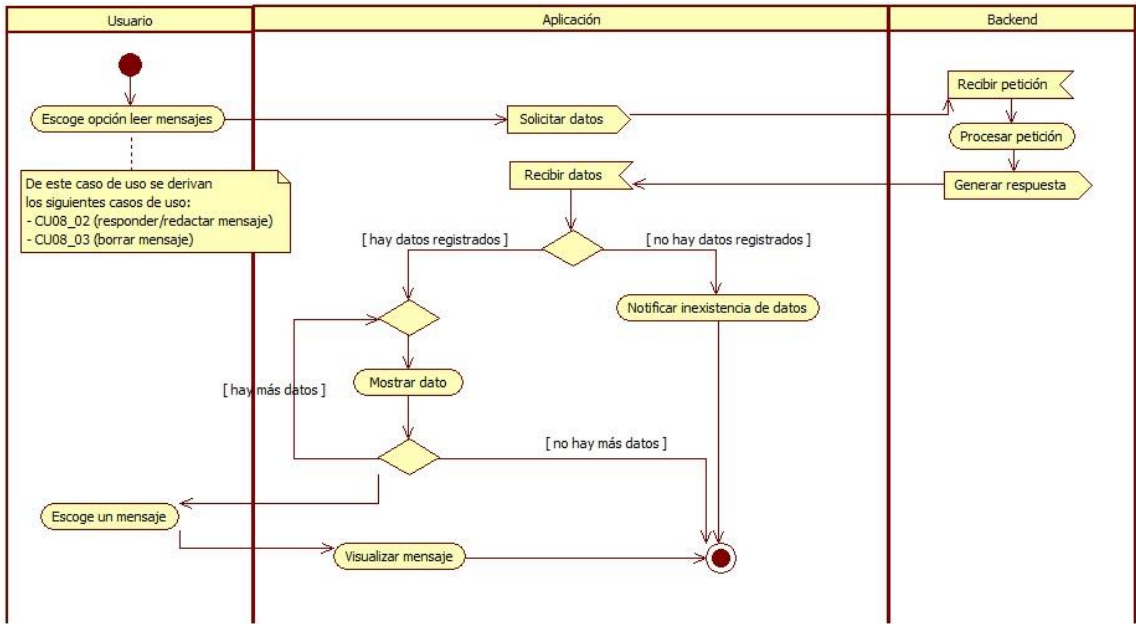
Figura 60. Prototipo funcionalidad actualizar mensajes.



Fuente: Elaboración propia

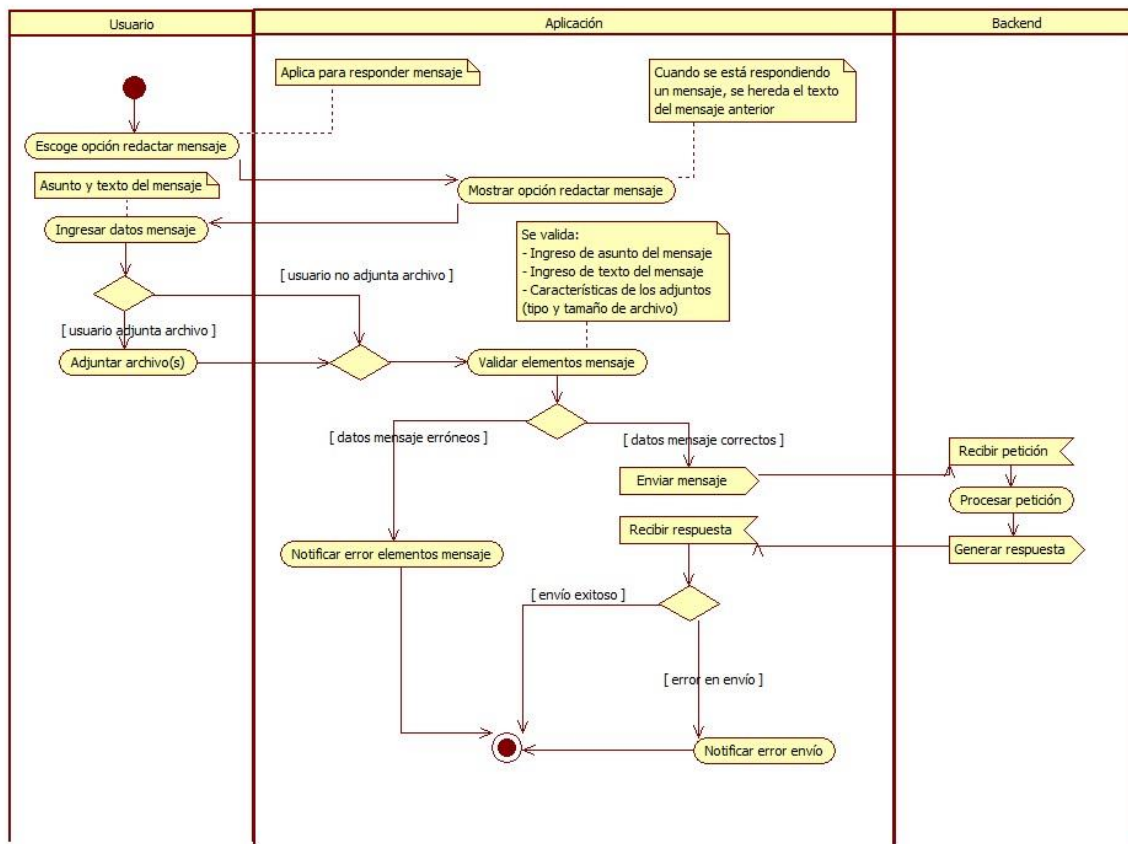
4.3.7.4. Diagrama de actividades.

Figura 61. Diagrama de actividades caso de uso CU08_01.



Fuente: Elaboración propia

Figura 62. Diagrama de actividades caso de uso CU08_02.



Fuente: Elaboración propia

El los diagramas arquitectónicos de esta iteración pueden consultarse en el anexo 7 – documento de arquitectura, así:

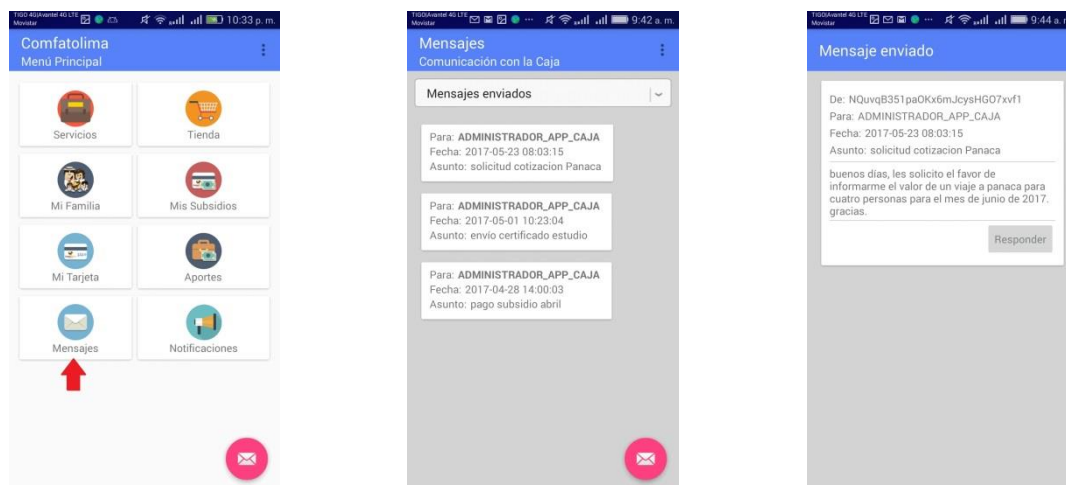
- Diagrama de clases, página 14
- Diagrama de comunicación, página 17
- Diagrama de secuencia, página 21
- Diagramas de actividades, página 31 y 32
- Diagrama de paquetes, página 33

4.3.7.5 Pruebas. Se realizaron pruebas unitarias a un total de 10 clases codificadas en la iteración. El detalle se puede consultar en el anexo 9 – pruebas unitarias y de integración, página 33.

Se realizaron pruebas de usabilidad y adaptación a diferentes tipos de dispositivos, utilizando los criterios establecidos en el anexo 10 – pruebas de adaptación y usabilidad.

4.3.7.6 Despliegue. A continuación se presentan las pantallas que ofrece la funcionalidad de generar certificados de afiliación:

Figura 63. Las pantallas que ofrece la funcionalidad de generar certificados de afiliación.



1100 (Movistar 4G LTE)
Movistar

Redactar Mensaje

Enviar Cancelar

Ingrese asunto del mensaje

Ingrese contenido del mensaje

ADJUNTAR ARCHIVOS Máximo 3

Archivos adjuntos

Fuente: Elaboración propia

4.3.7.7 PSP – tamaños y tiempos estimados versus reales.

Figura 64. Tamaños y tiempos estimados versus reales.

Overall Metrics

Summary	Plan	Actual	A la Fecha
<i>Size/Hour</i>	21	#VALUE!	10,2
Program Size			
Base (B)	0	0	0
Deleted (D)	0	0	0
Modified (M)	0	0	0
Added (A)	943	#VALUE!	5044
Reused (R)	0	0	3965
Added and Modified (A+M)	943	#VALUE!	5044
Total Size (T)	943	#VALUE!	9009
Total New Reusable	0	0	1904
<i>Estimated Proxy Size (E)</i>	937	1127	11124

Time In Phase

	Plan	Actual	A la Fecha	% A la Fecha
Planeación	2:36	3:36	29:06	5,91%
Diseño	8:40	11:33	96:49	19,70%
Codificación	22:32	16:42	238:15	48,40%
Compilación	4:46	5:36	52:24	10,60%
Pruebas	5:54	12:14	70:15	14,30%
Postmortem	0:33	0:11	5:35	1,13%
Total	45:00	49:52	492:24	

Fuente: Elaboración propia

4.3.7.8. PSP – defectos detectados y removidos.

Figura 65. Defectos detectados y removidos.

Defects Injected

	Actual	A la Fecha	% A la Fecha
Antes de Desarrollo	0	0	0%
Planeación	0	0	0%
Diseño	0	26	22,00%
Codificación	4	92	78,00%
Compilación	0	0	0%
Pruebas	0	0	0%
Total	4	118	

Defects Removed

	Actual	A la Fecha	% A la Fecha
Planeación	0	0	0%
Diseño	0	5	4,24%
Codificación	1	44	37,30%
Compilación	0	15	12,70%
Pruebas	3	54	45,80%
Total	4	118	
After Development	0	0	

Fuente: Elaboración propia

4.3.7.9 Análisis de la iteración. La estimación inicial de tamaño fue de 937 LOC y el tamaño final de la iteración fue de 1127 LOC, es decir que se codificó el 120.28% de lo estimado. Esto debido al tamaño de la opción, que implicaba codificar varios procesos (leer mensajes, responder mensaje, redactar mensaje, filtrar por tipo de mensaje).

El tiempo estimado de desarrollo fue de 45 horas y el tiempo real fue de 50 horas, es decir el 111% del tiempo estimado.

Se detectaron y corrigieron 4 defectos significativos frente a 3 defectos de la fase anterior. En las iteraciones se están registrando errores que impliquen la interrupción del programa u otros efectos críticos. Errores en distribución de espacios en vistas, valores nulos y demás no se están registrando.

Todos los reportes PSP de esta iteración se pueden consultar en el anexo 17 – PSP iteración 07.

4.3.8. Actualizar Mensajes Administrador.

4.3.8.1. Definición de Requerimientos. En esta iteración se desarrolló el caso de uso “actualizar mensajes” del sistema de administración, el cual conserva características idénticas al requerimiento RE_APP_08 – enviar mensaje personalizado de la aplicación móvil, con la diferencia que en este caso no se adjuntan archivos en los mensajes.

Cabe aclarar que esta opción manejará toda la comunicación personalizada con los usuarios de la aplicación móvil. El alcance total del sistema de administración comprende otras utilidades que no hacen parte del presente proyecto pero que son requerimientos concertados con el cliente dentro del acuerdo contractual.

A la fecha se han realizado otros desarrollos dentro del sistema de administración de la aplicación, tales como el sistema de autenticación y gestión de usuarios. Por lo tanto para esta opción se han reutilizado clases codificadas previamente y es un sistema que está en proceso de desarrollo actualmente. No existe línea base debido a que es un desarrollo *JEE*.

Para esta iteración se utilizó *PSP0* que se enfoca en el registro de tiempos y defectos.

4.3.8.2. Casos de uso textuales. Aplican los casos de uso CU08_01, CU08_02 y CU08_03.

4.3.8.3. Diagrama de casos de uso.

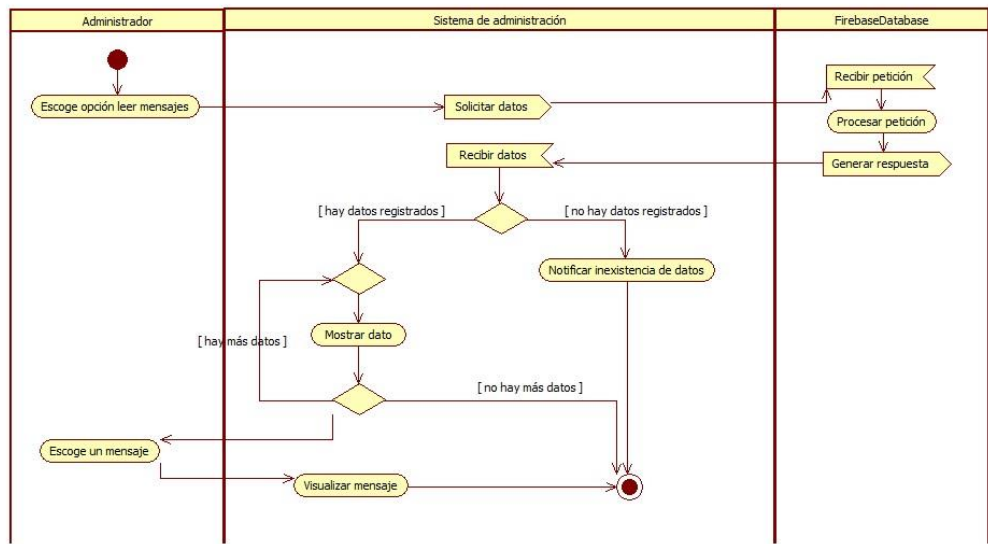
Figura 66. Diagrama de casos de uso.



Fuente: Elaboración propia

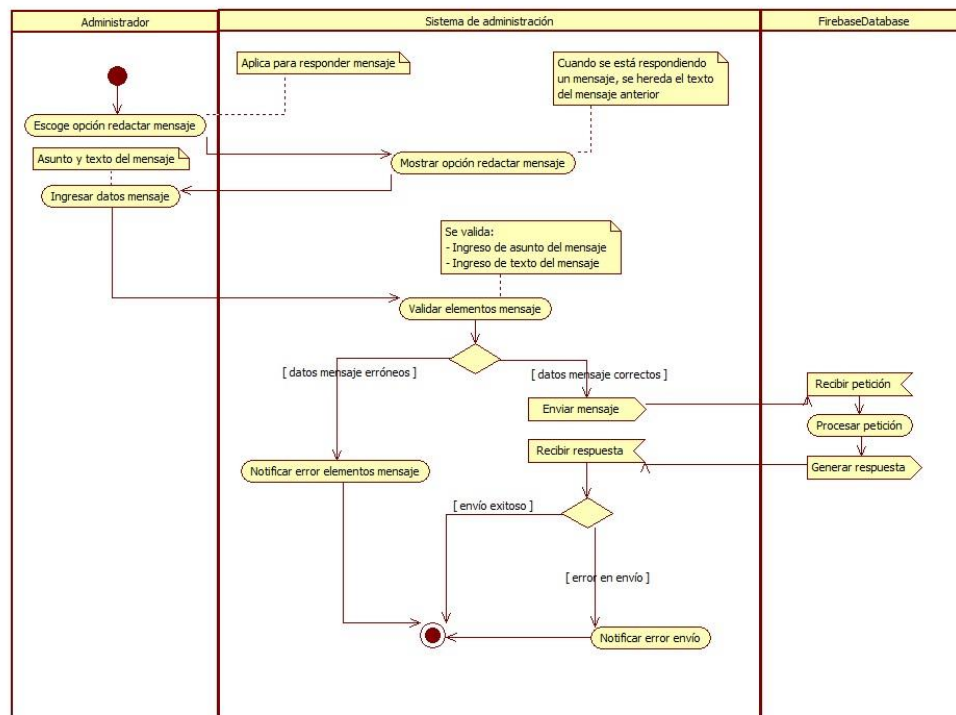
4.3.8.4 Diagrama de actividades.

Figura 67. Diagrama de actividades leer mensaje.



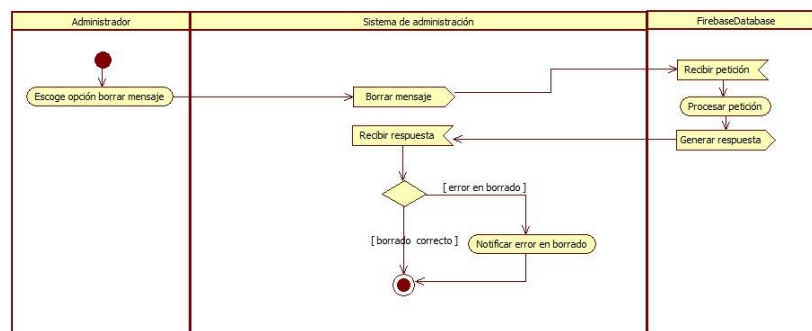
Fuente: Elaboración propia

Figura 68. Diagrama de actividades redactar mensaje.



Fuente: Elaboración propia

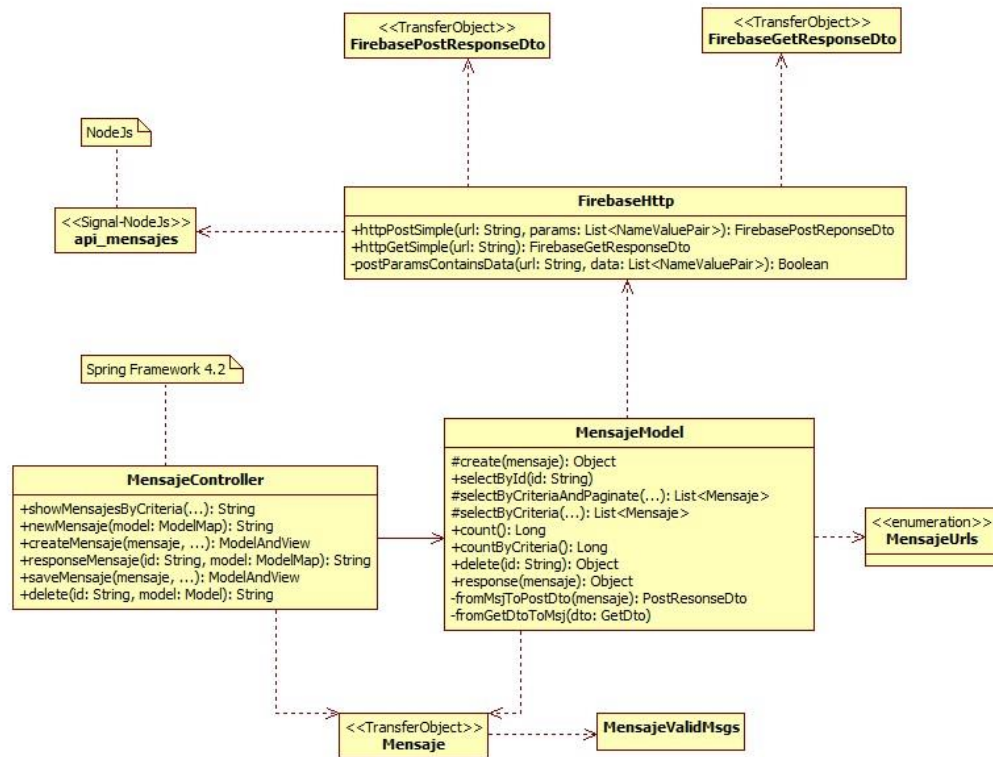
Figura 69. Diagrama de actividades borrar mensaje.



Fuente: Elaboración propia

4.3.8.5 Diagrama de clases.

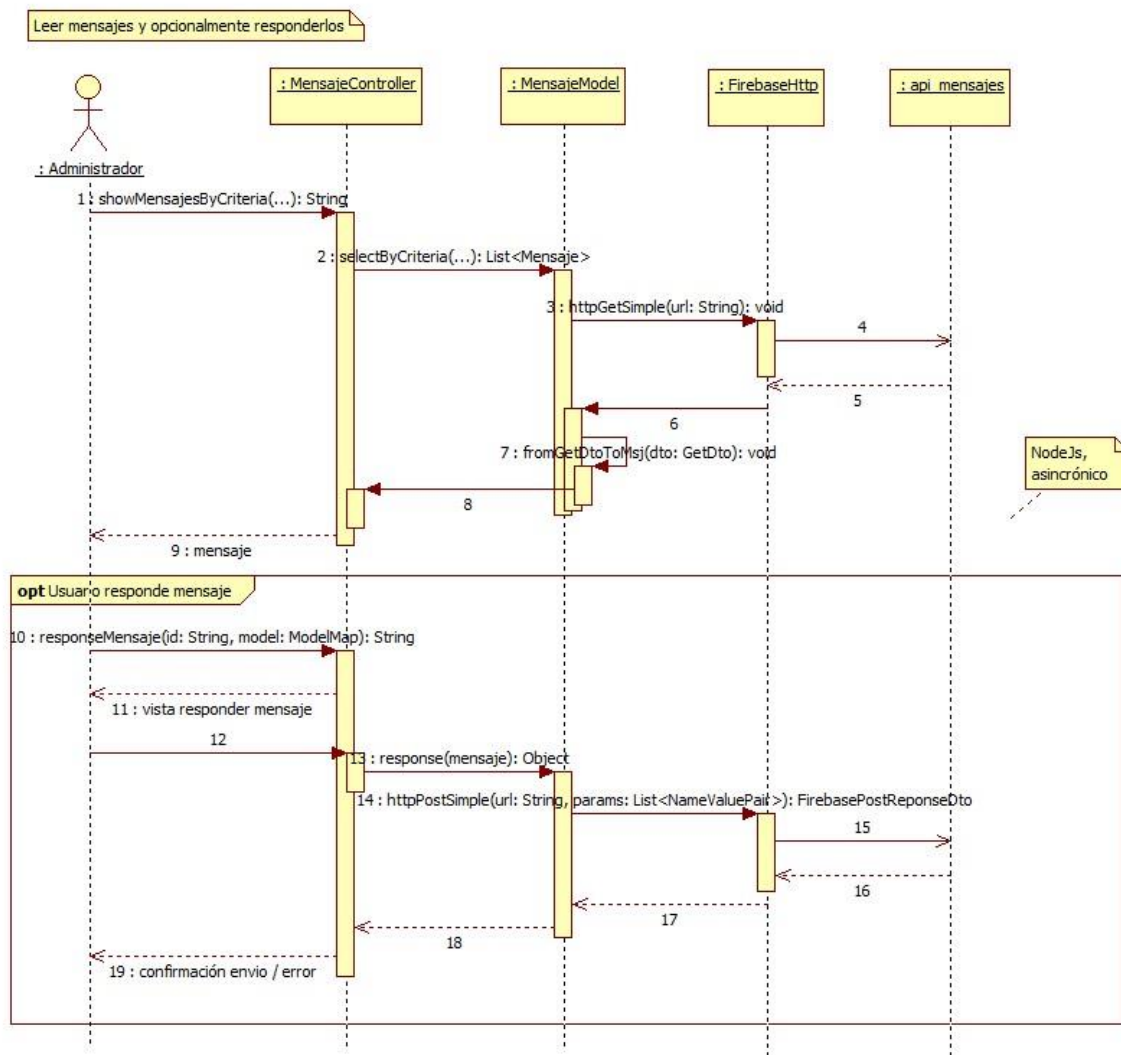
Figura 70. Diagrama de clases actualizar mensajes.



Fuente: Elaboración propia

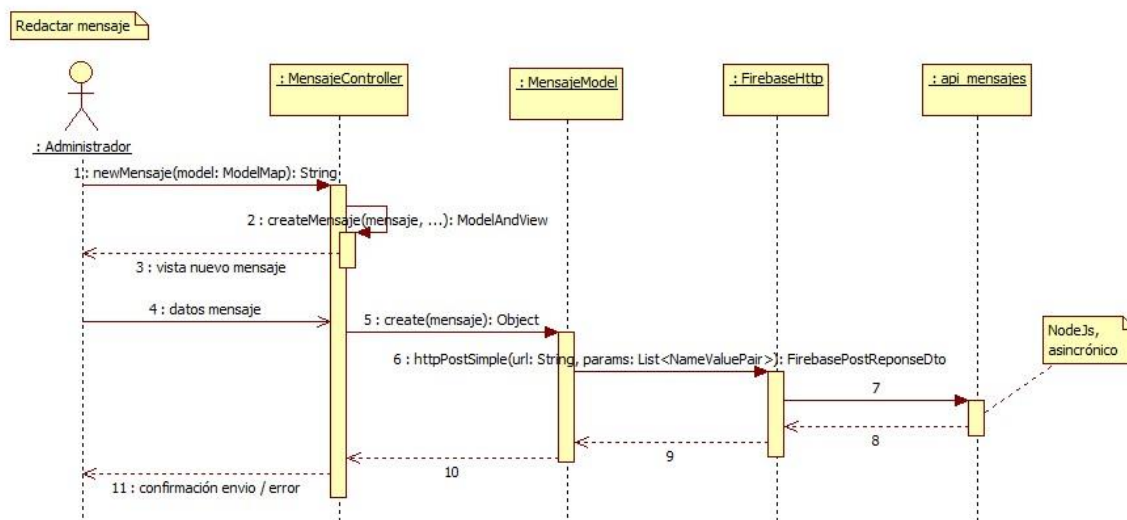
4.3.8.6 Diagramas de secuencia.

Figura 71. Diagrama de secuencia leer y responder mensajes.



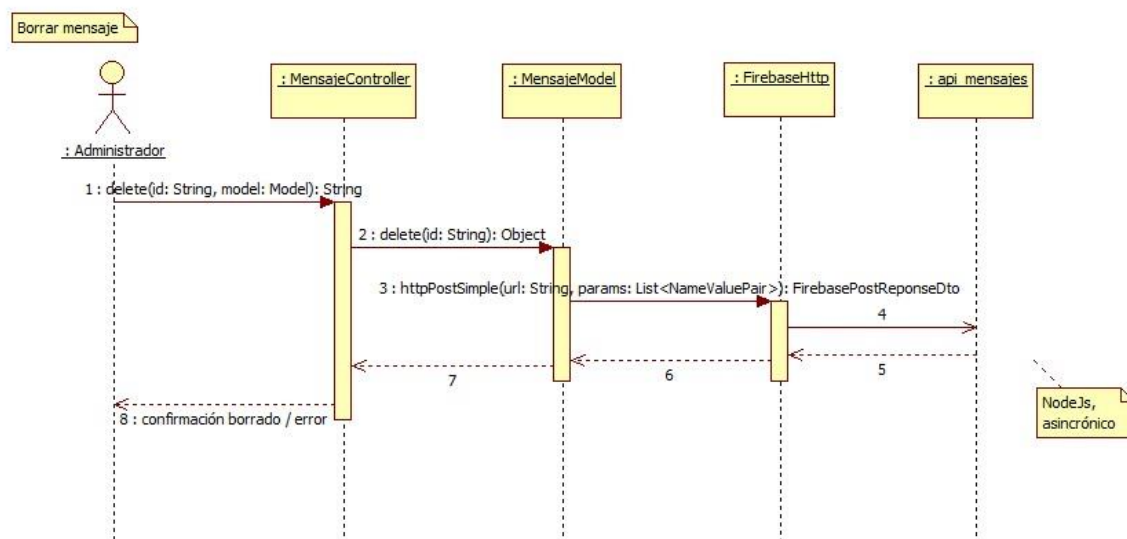
Fuente: Elaboración propia

Figura 72. Diagrama de secuencia redactar mensaje.



Fuente: Elaboración propia

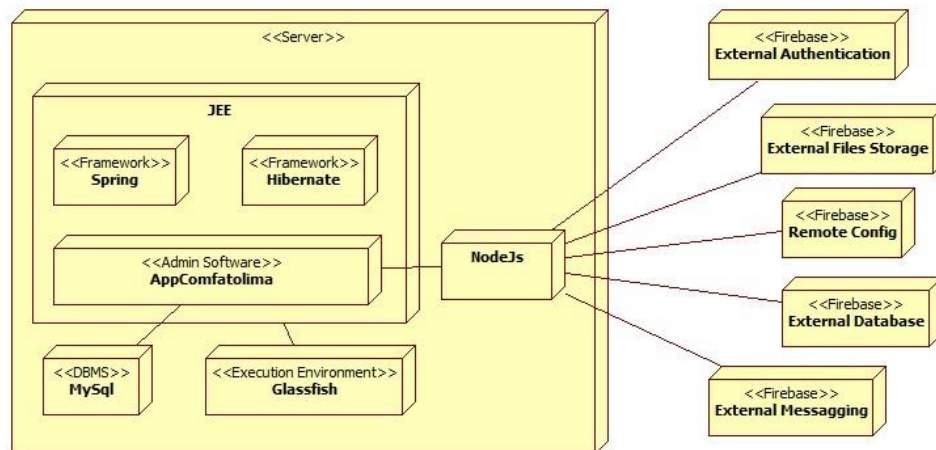
Figura 73. Diagrama de secuencia borrar mensaje.



Fuente: Elaboración propia

4.3.8.7. Diagrama de despliegue.

Figura 74. Diagrama de despliegue sistema de administración.



Fuente: Elaboración propia

4.3.8.8. Pruebas. Se realizaron pruebas unitarias a las clases MensajeModel, MensajeController. Los errores detectados y corregidos se muestran en los numerales siguientes de PSP.

4.3.8.9. Despliegue.

Figura 75. Ingreso al sistema de administración.



Fuente: Elaboración propia

Figura 76. Usuarios sistema de administración.



Fuente: Elaboración propia

Figura 77. Agregar usuarios sistema de administración.

The screenshot shows a web browser window with the URL `localhost:8080/BackendAppCaja/users/new`. The page title is "Nuevo usuario". The form contains the following fields and controls:

- Usuario:** Text input with placeholder "Ingrese usuario".
- Clave:** Text input with placeholder "Ingrese clave".
- Apellido 1:** Text input with placeholder "Ingrese primer apellido".
- Apellido 2:** Text input.
- Nombre 1:** Text input with placeholder "Ingrese primer nombre".
- Nombre 2:** Text input.
- Role de usuario:** Radio buttons for "Operador" (selected) and "Administrador".

At the bottom left, it says "© 2017 mundo.java.co". At the bottom right, there is a blue button with a white icon.

Fuente: Elaboración propia

Figura 78. Modificar usuarios sistema de administración.

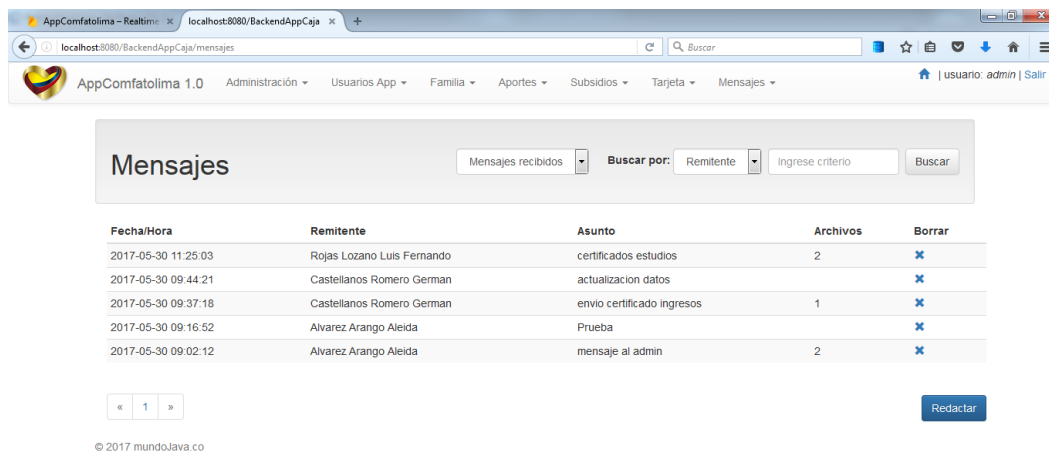
The screenshot shows a web browser window with the URL `localhost:8080/BackendAppCaja/users/update/3`. The page title is "Modificar usuario". The form contains the following fields and controls:

- Usuario:** Text input with value "Irojas".
- Apellido 1:** Text input with value "ROJAS".
- Apellido 2:** Text input with value "LOZANO".
- Nombre 1:** Text input with value "LUIS".
- Nombre 2:** Text input with value "FERNANDO".
- Activo:** Radio buttons for "Si" (selected) and "No".
- Role de usuario:** Radio buttons for "Operador" (selected) and "Administrador".

At the bottom left, it says "© 2017 mundo.java.co". At the bottom right, there is a blue button with a white icon.

Fuente: Elaboración propia

Figura 79. Lista mensajes sistema de administración.



Fuente: Elaboración propia

Figura 80. Leer mensaje sistema de administración.



Fuente: Elaboración propia

Figura 81. Redactar mensaje sistema de administración.

The screenshot shows a web browser window with the URL `localhost:8080/BackendAppCaja/mensajes/new`. The page title is "AppComfatolima 1.0". The navigation menu includes "Administración", "Usuarios App", "Familia", "Aportes", "Subsidios", "Tarjeta", and "Mensajes". The user is logged in as "usuario: admin" and can click "Salir".

The main form is titled "Redactar mensaje" and contains the following fields:

- Para:** A dropdown menu with the placeholder text "Escoja un usuario".
- Asunto:** A text input field with the placeholder text "ingrese un asunto".
- Mensaje:** A large text area for composing the message.

At the bottom of the form are two buttons: "Enviar" (Send) and "Cancelar" (Cancel). A copyright notice "© 2017 mundoJava.co" is visible in the bottom left corner.

Fuente: Elaboración propia

4.3.8.10 PSP – tiempo empleado en la iteración.

Figura 81. Tiempo empleado en la iteración.

Bitácora de TiempoPara /AppComfaDefinitiva/08 Actualizar mensajes admin

Proyecto/Tarea	Fase	Tiempo Inicio	Delta	Interrupción	Comentario
/AppComfaDefinitiva/08 Actualizar mensajes admin	Planning	may 01, 2017 08:40:26 AM	95	0	revisión casos de uso detallados
/AppComfaDefinitiva/08 Actualizar mensajes admin	Design	may 01, 2017 02:11:16 PM	5	0	Diagrama casos de uso
/AppComfaDefinitiva/08 Actualizar mensajes admin	Design	may 01, 2017 02:17:34 PM	47	0	Diagrama actividades leer mensajes
/AppComfaDefinitiva/08 Actualizar mensajes admin	Design	may 01, 2017 03:08:42 PM	39	0	Diagrama actividades redactar mensaje
/AppComfaDefinitiva/08 Actualizar mensajes admin	Design	may 01, 2017 04:46:13 PM	34	0	Diagrama actividades borrar mensaje
/AppComfaDefinitiva/08 Actualizar mensajes admin	Design	may 01, 2017 07:50:36 PM	108	0	Diagrama clases
/AppComfaDefinitiva/08 Actualizar mensajes admin	Design	may 01, 2017 10:25:54 PM	25	0	Diagrama secuencia leer mensajes
/AppComfaDefinitiva/08 Actualizar mensajes admin	Design	may 01, 2017 10:58:10 PM	19	0	Diagrama secuencia redactar mensaje
/AppComfaDefinitiva/08 Actualizar mensajes admin	Design	may 01, 2017 11:25:19 PM	21	0	Diagrama secuencia borrar mensaje
/AppComfaDefinitiva/08 Actualizar mensajes admin	Code	may 03, 2017 07:50:09 AM	194	0	codificación clase MensajeModel
/AppComfaDefinitiva/08 Actualizar mensajes admin	Code	may 03, 2017 03:52:05 PM	103	0	codificación clase MensajeController
/AppComfaDefinitiva/08 Actualizar mensajes admin	Code	may 03, 2017 07:22:16 PM	57	0	codificación vista mensajes
/AppComfaDefinitiva/08 Actualizar mensajes admin	Code	may 03, 2017 08:55:25 PM	34	0	codificación vista mensajeNew
/AppComfaDefinitiva/08 Actualizar mensajes admin	Code	may 03, 2017 10:04:16 PM	28	0	codificación vista mensajeShow
/AppComfaDefinitiva/08 Actualizar mensajes admin	Compile	may 04, 2017 09:56:45 AM	75	0	revisión código limpio
/AppComfaDefinitiva/08 Actualizar mensajes admin	Test	may 04, 2017 02:09:45 PM	157	0	pruebas a MensajeModel
/AppComfaDefinitiva/08 Actualizar mensajes admin	Test	may 04, 2017 06:00:06 PM	234	0	pruebas a MensajeController y vistas
/AppComfaDefinitiva/08 Actualizar mensajes admin	Postmortem	may 05, 2017 01:01:20 PM	103	0	corrección diagramas arquitectura
/AppComfaDefinitiva/08 Actualizar mensajes admin	Postmortem	may 05, 2017 03:24:39 PM	19	0	elaboración diagrama despliegue

Fuente: *ProcessDashboard*

Tiempo total empleado: 23:17

4.3.8.11 PSP – defectos detectados y removidos.

Tabla 27. Defectos detectados y removidos.

Bitácora de DefectosPara /AppComfaDefinitiva/08 Actualizar mensajes admin

Proyecto/Tarea	Fecha	ID	Tipo	Inyectado	Eliminado	Tiempo de Arre	Descripción
----------------	-------	----	------	-----------	-----------	----------------	-------------

						glo	
/AppComfaDefin itiva/08 Actualizar mensajes admin	05/04/2 017	1	Interf ace	Code	Test	39	en los enlaces que van dentro de los formularios jsp, al poner el url en algunas opciones no encontraba el url para ir a otra opcion, se desconocen las causas del fallo. se puso en cada url <code>\${pageContext.request.contextPath}/url</code> y funcionÃ³ bien
/AppComfaDefin itiva/08 Actualizar mensajes admin	05/04/2 017	2	Funct ion	Code	Test	12	en la clase MensajeModel, mÃ©todo <code>selectByCriteriaAndPaginated(String criteria, Integer offset, Integer maxResults)</code> , cuando es la primera vez que se carga la lista de datos, si los valores offset y maxResults son nulos, la paginaciÃ³n de la vista se desactiva. Se corrige la el

							metodo para que si offset es nulo, lo convierta en cero y si maxResults es nulo, asuma el valor de la constante MAX_RESULTS
/AppComfaDefinitiva/08 Actualizar mensajes admin	05/04/2017	3	System	Code	Test	5	para prevenir que nodejs atienda peticiones desde otros equipos, en el archivo index.js se modifica el parámetro IP_ADDRESS = "127.0.0.1", con esto queda atendiendo solo a las peticiones que vienen desde el mismo equipo
/AppComfaDefinitiva/08 Actualizar mensajes admin	05/04/2017	4	Function	Code	Test	57	Para realizar las validaciones de los formularios desde Spring, el pojo Mensaje se hizo a partir del pojo User pero no se cambió la clase que suministra los mensajes de validación al usuario. Al inducir un error, el mensaje

						no corresponde con el contexto que se está usando (arroja mensaje de error de datos de usuario cuando se estaba creando un mensaje. Se creó la clase MensajeValidMsgs y se modificó tanto el pojo como el archivo de validación.js
/AppComfaDefinitiva/08 Actualizar mensajes admin	05/04/2017	5	Function	Code	Test	49 En la validación de las vistas mediante Spring, al dejar espacios en blanco en campos que deberían ser obligatorios, arroja una excepción en nodejs al intentar grabar datos en Firebase. Se realiza validación mediante js en el pojo y en el controlador MensajeController se agrega el método initBinder(WebDataBinder

						binder) que convierte las cadenas nulas en vacías y esto permite guardar los datos sin problema en los campos opcionales de los formularios.
/AppComfaDefinitiva/08 Actualizar mensajes admin	05/04/2017	6	Funcion	Code	Test	17 en el método createMensaje, cuando se presentaba un error de ingreso de datos del usuario (asunto y/o texto del mensaje), el proceso de validación estaba redireccionando a la lista de mensajes directamente y no se sabía que pasaba con el proceso. Solución: en todos los casos, sea que la validación sea exitosa o que hayan errores, siempre se vuelve a redireccionar al formulario de redactar mensaje. En caso de que

						hayan errores se muestra el error y los datos viajan de nuevo a trav��s del model. Si la validaci��n es correcta los campos de la vista se limpia y se muestra un mensaje de proceso ok y un enlace para volver a la lista de mensajes.
--	--	--	--	--	--	--

Fuente: *ProcessDashboard*

5. Análisis de resultados

A continuación se hace una revisión de los resultados del proyecto luego de la fase de desarrollo frente a los resultados esperados iniciales:

Se hace entrega del documento de diseño arquitectónico de la aplicación utilizando el modelo 4+1 vistas de *Kruchten* para la aplicación móvil. En cuanto al subsistema de administración, los diagramas arquitectónicos se incluyen en la iteración número 8 del desarrollo del proyecto.

Se hace entrega al cliente del *Backend* de la aplicación mediante el servicio *Firebase Google*, con la estructura de la base de datos, la configuración de seguridad correspondiente y las opciones de autenticación, consultas de datos y gestión de mensajes.

Se hace entrega del Subsistema de consulta y actualización de datos básicos del afiliado, con la opción de generar certificados de afiliación. Por solicitud del cliente, la actualización de datos de afiliación no la hace directamente el usuario a través de la aplicación móvil, para no exponer el sistema de información de subsidio y aportes de la Caja. A cambio de esto el usuario solicita la actualización de datos mediante un mensaje y el administrador de la aplicación gestiona el cambio.

Se hace entrega del subsistema de consulta de histórico de pagos de subsidio, pagos de aportes, movimientos de tarjeta y datos de tarjeta en la aplicación móvil.

Se hace entrega del subsistema de administración de la aplicación móvil en entorno *web*, para gestionar lo referente a solicitudes, notificaciones y comunicación el usuario. Por solicitud

del cliente, la comunicación personalizada se hace a través de mensajes y la comunicación masiva se hace a través de notificaciones *push*.

En cada uno de los numerales que componen la fase de desarrollo, se especifican diferentes pruebas realizadas y los resultados obtenidos. Así como también al final de cada iteración se hace una interpretación de las mediciones obtenidas mediante *PSP*.

Se hace entrega del manual del usuario y manual técnico. El código fuente no se adjunta al presente informe, debido a que será entregado al cliente dentro del acuerdo contractual.

El proyecto total está compuesto por los entregables del presente proyecto y requerimientos adicionales acordados con el cliente, los cuales están en proceso de desarrollo actualmente.

A nivel de cronograma, el proyecto presentó las siguientes variaciones:

Tabla 28. Cronograma estimado para el proyecto.

No	Actividad	Tiempo																	
		ago-16			sep-16			oct-16		nov-16		dic-16		ene-17		feb-17			
1	Diseño arquitectura	■	■	■	■	■													
2	Desarrollo subsistema de consultas						■	■	■	■	■	■							
3	Desarrollo subsistema de mensajes											■	■	■					
4	Desarrollo subsistema de administración												■	■	■	■			
5	Elaboración documentación e informe final															■	■	■	■

Fuente: elaboración propia.

Tabla 29. Cronograma final del proyecto.

Nº	Actividad	Tiempo											
		oct-16	nov-16	dic-16	ene-17	feb-17	mar-17	abr-17	may-17	jun-17	jul-17	ago-17	
1	Diseño arquitectura y requerimientos	■	■										
2	Desarrollo subsistema de consultas		■	■	■	■	■	■	■				
3	Desarrollo subsistema de mensajes							■	■				
4	Desarrollo subsistema de administración								■	■			
5	Elaboración documentación e informe final									■	■	■	■

Fuente: elaboración propia.

Se estimaron 167 días para desarrollar el proyecto y la duración fue de 228 días, para un incremento del 36.53%. Para la elaboración del informe final se estimaron 37 días y la duración fue de 86 días, para un incremento del 132.43%.

El incremento en el tiempo de duración del proyecto también implicó más costos a nivel de personal:

Tabla 30. Costos de personal presupuestados.

COSTOS DE PERSONAL PRESUPUESTADOS							
No	Participante	Formación	Función	Dedicación	Duración	Costo	Costo
				(hrs/sem)	(meses)	mensual	total
1	Jorge Iván Meza M.	Ing, Ms	Director	8	6	400.000	2.400.000
2	Germán Castellanos	Ing, Especialista	Autor Proyecto	20	6	1.200.000	7.200.000
3	Angel Bernate	Ingeniero	Tester funcional	1	4	200.000	800.000
4	Carlos Beltrán	Ingeniero	Tester funcional	1	4	200.000	800.000
5	Luis F. Rojas	Tecnólogo	Tester funcional	1	4	200.000	800.000
6	Por definir	Por definir	Pares acad. (2)	1	2	600.000	1.200.000
					Total	2.800.000	13.200.000

Fuente: elaboración propia.

Tabla 31. Costos de personal ejecutados.

COSTOS DE PERSONAL FINALES							
N o	Participante	Formación	Función	Dedicación	Duración	Costo	Costo
				(hrs/sem)	(meses)	mensual	total
1	Jorge Iván Meza M.	Ing, Ms	Director	8	8	400.000	3.200.000
2	Germán Castellanos	Ing, Especialista	Autor Proyecto	20	8	1.200.000	9.600.000
3	Angel Bernate	Ingeniero	Tester funcional	1	4	200.000	800.000
4	Carlos Beltrán	Ingeniero	Tester funcional	1	4	200.000	800.000
5	Luis F. Rojas	Tecnólogo	Tester funcional	1	4	200.000	800.000
6	Por definir	Por definir	Pares acad. (2)	1	2	600.000	1.200.000
					Total	2.800.000	16.400.000

Fuente: elaboración propia.

Los costos de personal presupuestados fueron de \$13.200.000 y con el incremento del tiempo del proyecto ascendieron a \$16.400.000, para un incremento del 24.24%.

Las causas que generaron estas variaciones en el cronograma fueron las siguientes:

- El desarrollo del proyecto simultáneamente con el estudio, prueba y adaptación de las tecnologías a aplicar en el mismo, esto representó una curva de aprendizaje ardua que se vio reflejada en la falta de continuidad en los días de desarrollo, mientras se creaban ejemplos funcionales que después se adaptaban al proyecto.
- La decisión de replantear la arquitectura del proyecto y el *backend*, adoptando a *Firebase Google* como *Backend as service*. Esto implicó nuevamente una curva de aprendizaje y codificación nuevamente del proyecto con base en el patrón *Model View Presenter*.

En el numeral 8 de lecciones aprendidas, se describen más a fondo estas situaciones.

6. Conclusiones

El proceso de implementación de la aplicación móvil contó con la decidida participación tanto de los funcionarios de la Oficina de Sistemas como de la empresa en general, obedeciendo a las estrategias de posicionamiento de imagen corporativa y fortalecimiento tecnológico de la Caja de Compensación.

En la aplicación móvil y el sistema de administración destacan por su usabilidad, su adaptación a diferentes tipos de dispositivos *Android* y simplicidad de operación, logrando aceptación entre todas las personas que la han utilizado.

Si bien se presentaron desfases en el cronograma esto no afectó al cliente, ya que a nivel de planeación estratégica se determinó la implementación de una aplicación móvil para la Caja durante el año 2017, habiendo tomado en consideración todo el proceso que significaba la adaptación de la aplicación a las características del sistema de información de subsidio existente en la entidad, se estableció todo el año como tiempo margen para el proyecto y se cumplió la meta de implementarlo antes de finalizar el año.

Por ser la primera vez que se realizaba un proyecto de desarrollo de software de estas características mediante la aplicación de metodologías, el uso de *PSP1* dejó establecida una importante línea base acerca de la estimación de tiempos, tamaños, esfuerzos y resolución de problemas. Esta información permitirá en proyectos futuros realizar estimaciones más cercanas a la realidad y las lecciones aprendidas durante el desarrollo se traducirán en menos ensayo-error. No obstante lo anterior, se requiere practicar bastante en la incorporación de la herramienta *Process Dashboard* a las actividades del proyecto, debido a que hubo omisiones en el registro de

tiempos y errores, tanto por la inexperiencia en el uso de la herramienta como por el afán de cumplir con las fechas establecidas en el cronograma.

Como consecuencia de no contar con información histórica de proyectos similares y no tener total claridad acerca del reto que implicaba desarrollar la aplicación móvil, el *backend* y el sistema de administración, las estimaciones fueron bastante alejadas de la realidad. Sin embargo, en la medida que se avanzó en las iteraciones del proyecto, los datos de cada iteración culminada anterior permitieron acercarse cada vez más a la realidad. Para futuros proyectos se tomarán como referencia los resultados arrojados en el presente proyecto y se ajustarán a la complejidad del software a construir.

El uso del enfoque iterativo e incremental permitió una mejor comunicación con el cliente y redujo el riesgo de que aparecieran nuevos requerimientos luego de culminar las iteraciones. Mediante la retroalimentación constante se fueron ajustando cada uno de los entregables hasta satisfacer las expectativas del cliente.

Con la aplicación móvil y su sistema de comunicación entre los usuarios y el administrador de la aplicación, se ha logrado unificar y agilizar el diálogo con los afiliados, ya que el usuario canaliza todos sus requerimientos y solicitudes a una sola dependencia (administración de la aplicación) y no requiere enviar correos o llamar a varias dependencias para resolver sus inquietudes.

La plataforma *Firebase* de Google por ser un *backend as a service* facilitó en gran medida el desarrollo de la aplicación, ya que permitió implementar en forma sencilla el servicio de autenticación y gestión de usuarios, la base de datos, el almacenamiento de archivos, la configuración remota, las notificaciones *push* y el reporte de errores.

No obstante las bondades de *Firebase*, hay que tener en cuenta que por ser un *backend as a service (BAAS)* el proyecto quedó totalmente supeditado a *Google*. Esto podría representar un riesgo para la continuidad del proyecto, en el evento en que el proveedor decida modificar los términos del servicio de manera significativa, en este caso se estaría dependiendo totalmente del tiempo y las opciones de migración que el proveedor establezca. Sin embargo se considera que esta situación sería poco probable en el corto y mediano plazo, debido a la importancia que tienen para *Google* los dispositivos móviles y el sistema operativo *Android*.

La aplicación móvil de Comfatolima está orientada principalmente a la consulta de datos, siendo el registro inicial y la actualización de mensajes los únicos procesos que insertan datos en la base de datos *Firebase*. Al momento de escoger este sistema de base de datos para utilizarlo en una aplicación, es importante tener en cuenta que es un sistema *NoSql*, por lo tanto no maneja los conceptos de tablas, relaciones ni integridad referencial y la duplicación de datos no es una infracción. Con las bases de datos *NoSql* se sacrifica unicidad e integridad referencial pero se gana desempeño, siempre y cuando la indexación de los datos sea correcta. Si el proyecto a realizar requiere integridad referencial, se pueden considerar servicios como *OpenShift*.

Al utilizar un sistema de base de datos *NoSql*, hay que tener claro que los conceptos de *Data Definition Language*, *Data Control Language* y *Data Manipulation Language* propios de *SQL* desaparecen, por lo tanto la filosofía de trabajo cambia. Se sacrifica la versatilidad en la realización de consultas que ofrece *SQL* para ganar desempeño y tiempos de respuesta óptimos, necesarios en aplicaciones con alta demanda de usuarios y gestión de grandes volúmenes de datos.

En la realización de pruebas de software (unitarias, de integración, de adaptación a diferentes dispositivos), uno de los procesos que más consume tiempo es la redacción de los casos de prueba y el diligenciamiento de los formularios antes de iniciar con la codificación. Para esto resultó muy útil el uso de la herramienta *TestLink*, la cual permite crear y gestionar los planes de prueba, las *suites* de prueba, los casos de prueba por cada *suite*, los pasos de cada caso de prueba, ingresar los resultados de las pruebas y generar informes sobre el proceso rápidamente.

El uso de los *frameworks Hibernate y Spring* facilitó en gran medida el desarrollo del subsistema de administración en entorno *web*, simplificando el acceso a datos, la autenticación de usuarios, la aplicación del patrón MVC y la validación de formularios. Esto contrastado con la curva de aprendizaje que consumió tiempo del proyecto pero es una ganancia para proyectos futuros.

El uso de la librería *Bootstrap* significó un ahorro de tiempo en la aplicación de un concepto gráfico a las interfaces de usuario del subsistema de administración¹.

¹ Ver <https://v4-alpha.getbootstrap.com/>

7. Recomendaciones

Continuar con el desarrollo de las opciones para carga de datos provenientes del sistema de información de subsidio y aportes de la caja, mediante los *frameworks Hibernate y Spring* y haciendo uso del patrón *DAO (Data Access Object)*, el cual ante un eventual cambio en el sistema de gestión de base de datos de Comfatolima, permitirá cambiar la capa de datos del subsistema de administración sin que se vea afectada la lógica del negocio.

Comfatolima debe disponer de un funcionario encargado de la administración de la aplicación móvil, el cual deberá tener dedicación total a esta actividad, debido a que la demora en responder las solicitudes podría ocasionar un impacto negativo en la percepción de los usuarios, llegando a ocasionar que la Caja incurra en silencio administrativo. Por la variedad de procesos frecuentes que se deberán realizar para mantener actualizada la aplicación, no se recomienda que este trabajo se asigne a los *Web Master* que tiene actualmente la Caja, se sugiere que sea un funcionario de la Oficina de Subsidio y Aportes.

Para fomentar el uso de la aplicación por parte de los afiliados, se recomienda a Comfatolima crear estímulos tales como ofrecer un descuento del 10% en las boletas de la tradicional fiesta de amor amistad que organiza anualmente, para quienes las soliciten a través de la aplicación. Esto por supuesto deberá ir acompañado de una respuesta rápida por parte de la Caja.

Se recomienda un monitoreo constante del nivel de interacción entre los usuarios y la Caja a través de la aplicación móvil, para que en el momento en que la cantidad de solicitudes

y trámites realizados crezca, se estudie la posibilidad de asignar más funcionarios a la administración de la aplicación.

Comfatolima debe ser medida en el envío de información al usuario a través del sistema de notificaciones *push*, ya que según *We Make Apps*, 2017:

Un excesivo envío de notificaciones puede resultar irritante y llevar al usuario a eliminar la app. Las notificaciones automáticas son [“una molesta distracción” para el 52 % de los usuarios de apps](#). Si envías más de cinco a la semana, un tercio de tus usuarios dejará de utilizar la aplicación.

Por esta razón, para transmitir información misional al usuario, se recomienda enviar máximo 2 notificaciones *push* al mes y no enviar información publicitaria de terceros, solo enfocarse en el portafolio de servicios de Comfatolima.

8. Lecciones aprendidas

A continuación se detallan las principales lecciones aprendidas durante la realización del presente proyecto, se exponen las razones de cada afirmación y se recomiendan libros que ofrecieron orientación valiosa tanto para superar las dificultades como para en un futuro optimizar cada vez más la gestión de proyectos de software:

Antes de iniciar un proyecto de desarrollo de software

El proceso de aprendizaje de herramientas, lenguajes de programación y metodologías de desarrollo debe estar totalmente separado y debe hacerse antes de la realización del proyecto. Es tentador escoger una herramienta en la que no se tiene experiencia para realizar un proyecto buscando aprender mientras se va avanzando. Esto impacta directamente y en forma negativa en el cronograma, generando retraso en las actividades. Es frustrante.

Para introducirse en una nueva herramienta o lenguaje de desarrollo no es correcto tener la presión de estar dentro de unos tiempos fijados para la entrega de un producto de software, se debe practicar tranquilamente y programar ejemplos de opciones que posiblemente más adelante se vayan a utilizar en proyectos reales, tales como acceso al sistema, formularios *crud*, elaboración de reportes y otros, procurando llegar a un nivel de calidad y detalle que permitan reutilizar esas opciones en el mundo real más adelante. El proceso de desarrollo de software adquiere más dinamismo cuando se reutiliza código de calidad. Heredar código defectuoso o no probado meticulosamente es inyectarle nuevos problemas al producto.

El uso de *frameworks* es imprescindible para desarrollar en menos tiempo, evitar desgaste en aspectos que ya están hechos y probados y permite enfocarse en lo que debe hacer el software para cumplir las expectativas del cliente. Para desarrollar aplicaciones *web* en Java con acceso a bases de datos, son muy útiles los *frameworks* *Hibernate* para el acceso a datos y *Spring* para manejar la seguridad, presentación y lógica del negocio. También el uso de la librería *Bootstrap* permite embellecer las interfaces gráficas de usuario sin mucho esfuerzo. Programar línea por línea sin el uso de *frameworks* es recorrer a pie un largo trayecto cuando existen vehículos gratuitos y veloces a nuestra disposición. Por supuesto la curva de aprendizaje es empinada mientras se adquiere la destreza en su uso, por esta razón se hace énfasis en que se deben aprender los *frameworks* que se van a usar antes de la realización del proyecto, nunca durante el proyecto.

Es necesario antes de iniciar el proyecto, explorar al máximo las herramientas y facilidades que ofrece el *IDE* en el que se va a programar, así se evita realizar tareas tediosas por desconocimiento de la herramienta, por ejemplo buscar la utilización de una clase en todo el proyecto abriendo y revisando todas y cada una de las clases, por desconocer la existencia de la opción “*find usages*”, que rápidamente encuentra todos los sitios desde donde se referenció la clase, ahorrando mucho tiempo. El conocer a fondo el *IDE* evita cometer estos errores tan comunes.

En las primeras fases de un proyecto de software

Al iniciar el proceso de análisis y diseño se debe entender que el lenguaje unificado de modelado *UML* es una notación para comunicar lo que se pretende hacer en el proyecto a los diferentes interesados: programadores, usuario final, gerentes del proyecto, etc. Por lo tanto los

diagramas deben ser livianos, sencillos, entendibles, de lo contrario resultarán áridos hasta para quien los realizó.

Antes de realizar el análisis y diseño del *software* a construir, es imprescindible tener muy claro lo que la aplicación debe hacer, no solo es necesario para cumplir con las necesidades del cliente, sino que también permite visualizar cuales opciones del producto son muy similares y pueden ser programadas una sola vez y usarse donde se requiera, es decir, lograr el mayor nivel de abstracción posible. Una vez se detectan los procesos similares, se hace uno de un patrón arquitectónico que se adapte al problema que intentamos resolver. Por ejemplo, el estado de cuenta de un cliente se puede visualizar en pantalla en formato HTML y/o en un archivo *PDF*. En este caso el proceso de generar el estado de cuenta es el mismo y solo cambia la forma de presentación, es ideal el uso del patrón *Builder*, que abstrae la construcción de objetos complejos de su implementación.

Cuando se desarrolla un proyecto de software por primera vez, casi siempre resulta errado hacer estimaciones sin ningún tipo de información o elemento de juicio que respalde los “tres meses que me demoro”. Las estimaciones al tanteo traen oculto uno de los factores que más influyen en el fracaso de un proyecto: son excesivamente optimistas, alejadas de la realidad. En estos casos se debe conocer así sea en forma básica lo que debe hacer el software y se puede realizar la estimación mediante puntos función. Luego durante el transcurso del proyecto se puede ir contrastando lo estimado versus lo real y al final se obtendrá información real sobre cuánto nos demoramos en el proyecto y usar esto para estimaciones futuras.

El Proceso Software Personal (PSP versión 1) es una buena alternativa para realizar estimaciones, registrar los tiempos que se emplearon en el proyecto y registrar los

errores que se cometieron para contar con una base de conocimiento útil para proyectos futuros. En un principio es difícil habituarse a registrar tiempos y demoras o interrupciones de las actividades, pero finalmente la información que se obtiene es muy útil y mientras más datos históricos existan, más exactas serán las estimaciones y se correrá menos riesgo de agotar tempranamente los recursos financieros y de tiempo del proyecto.

Antes de programar

Se deben tener claros los aspectos mínimos de calidad que debe tener el software que se va a construir, tales como la baja complejidad ciclomática, legibilidad del código fuente, alta cohesión, bajo acoplamiento, métodos cortos, clases y métodos diseñados para un solo fin, correcta gestión de errores y excepciones, cumplimiento del acrónimo *SOLID* y demás aspectos que permitan probar fácilmente el software, que éste responda adecuadamente a las diferentes situaciones y asignarlo a otro programador sin que esto represente un desgaste muy significativo mientras se apropia del código fuente.

Al probar el software

Una de las partes más tediosas de los proyectos de desarrollo de software es la realización de pruebas unitarias y de integración. Todo el proceso que va desde la determinación de los casos de prueba, diligenciamiento de los formatos, escritura, ejecución de las pruebas y documentación de resultados, incluso puede llegar a durar más que la codificación. Pero al igual que son tediosas, son imprescindibles para garantizar que el producto responde adecuadamente a diversas situaciones. Para que esto sea más llevadero es necesario entrenarse previamente en el uso de herramientas de pruebas automatizadas como JUnit, la cual suele estar integrada con el

IDE y permite lograr un nivel de automatización que en cierta medida agilice este intrincado proceso.

Sobre las pruebas unitarias hay que tener en cuenta los siguientes aspectos que garantizan el éxito en esta fase:

Lo ideal es diseñar los casos de prueba antes de programar la clase, para esto es necesario tener clara la arquitectura de la aplicación que se va a desarrollar. Luego se programa la clase de manera que falle, empezando a pulir el código fuente hasta llegar a una ejecución satisfactoria.

El que una prueba unitaria sea exitosa desde la primera vez es síntoma de que no se están considerando todas las situaciones de ejecución que podrían darse cuando el software esté en producción.

Dejar las pruebas unitarias y de integración para el final es un grave error, ya que se pierde la perspectiva de lo que debe hacer cada clase y se pueden quedar sin probar aspectos extremos de la aplicación, tales como valores negativos donde deberían ser positivos, recepción de parámetros nulos u otras situaciones en que el software puede fallar. Es mejor diseñar la prueba, luego codificar la clase y luego probar.

El que una clase pase las pruebas de unidad no significa que sea perfecta o que nunca va a fallar, simplemente está ejecutando correctamente los casos de prueba que se diseñaron, por esta razón dichos casos de prueba deben considerar la mayor cantidad posible de escenarios de ejecución.

Hay que tener en cuenta que al realizar un cambio en una clase para solucionar un error, se pueden desencadenar errores en clases que ya habíamos probado y por ende el software

queda con errores. Las herramientas automatizadas de prueba tienen la ventaja de que, cada vez que se prueba una nueva clase, el sistema prueba todas las clases que ya se han probado anteriormente (la denominada prueba de regresión), por lo tanto se puede detectar si el cambio que se hizo afectó otras clases y tomar los correctivos a tiempo.

El hecho de que todas las clases pasen las pruebas unitarias, no significa que el software va a funcionar correctamente. Existe la posibilidad de que las clases funcionen correctamente en forma individual, pero al enlazar todas las clases existan fallas, esto se detecta mediante la realización de las pruebas de integración, que no es otra cosa que probar un conjunto de clases que trabajan entre sí para lograr un resultado. Para esto se debe tener clara la arquitectura de la aplicación.

Las pruebas automatizadas le dieron agilidad al proceso, también fueron importantes para el tema de regresión, debido a que un cambio en una clase podría afectar a clases que ya se probaron, por lo tanto en cada nueva prueba es ideal que se hagan nuevamente las pruebas que ya se hicieron para comprobar que ninguna clase ya probada se vea afectada. Las pruebas a las clases que no dependían de otras se hicieron automatizadas, pero las clases que sí dependían tales como los modelos, los presentadores y las vistas se hicieron manuales. Esto incrementa el riesgo de afectar clases ya probadas y que no se pueda controlar el error.

Acerca del uso de *Firestore*

Para que la utilización de *Firestore* en los proyectos de software no se convierta algo extremadamente complicado, primero hay que interiorizar lo siguiente:

Firestore es un *backend as a service*, es decir que ofrece todas las utilidades y características que requiere el extremo posterior de una aplicación móvil, como son la

autenticación, base de datos, almacenamiento, entre otros. Por lo tanto al adoptar este servicio, el proyecto se estará supeditando a google y a las utilidades que ofrece.

En la consola de administración de *Firebase* se tiene acceso a todas las opciones del sistema, por lo tanto dar este acceso al usuario final o al cliente es muy arriesgado ya que por ejemplo, podría borrar la base de datos o alterar la configuración. Se recomienda aparte de la aplicación móvil, programar una aplicación de administración que haga uso del *API* de *Firebase* para que el cliente ejecute acciones en forma restringida y segura.

Firebase proporciona un sistema de administración y autenticación de usuarios para las aplicaciones móviles, sin embargo esta información está protegida y no se maneja como una tabla más de la base de datos. Por ejemplo hasta el momento no es posible generar un listado de usuarios.

Hay que olvidarse de las bases de datos relacionales, de *SQL* y familiarizarse con las bases de datos *NoSql*. *Firebase* no maneja tablas ni relaciones, a cambio de eso maneja los datos en estructuras *JSON* denominadas nodos, las cuales tienen una clave primaria ya sea insertada o generada automáticamente y unos niveles de anidación, donde en aras de ganar velocidad de desempeño, se sacrifica la unicidad de los datos. Es decir que en una base de datos *NoSql*, los datos duplicados no son un pecado.

Al no existir *SQL* en *Firebase*, las consultas realizadas sobre los datos no permiten extraer solo algunos campos de un registro, se retornan los registros que cumplan el criterio de consulta con todos sus atributos en listas de objetos denominadas *snapshots*. Esto requiere más programación.

Firebase Database está diseñada para manejar grandes volúmenes de datos en forma eficiente, sin embargo esto solo se logra cuando se indexan adecuadamente los datos por campos no-clave y no hay muchos niveles de anidación.

La relación maestro – detalle se maneja usualmente por anidación de objetos *JSON* en la base de datos, por esta razón es aconsejable manejar máximo cinco o seis niveles de anidación por cada nodo de datos. Cuando la anidación tiene muchos niveles, al invocar un nodo o un registro específico de la lista, este arrastrará consigo todos sus datos anidados en todos sus subniveles, esto hará que la aplicación se torne lenta.

Para realizar correctamente los programas que interactúan con la base de datos *Firebase*, hay que familiarizarse con el concepto de *callback* o manejo de respuesta para peticiones asíncronas y con el concepto de *listener*.

Todas las consultas a la base de datos *Firebase* desde una aplicación *Android* se realizan mediante *listeners* que se adicionan a un objeto que apunta a la base de datos, este es un proceso asíncronico que cuanto obtiene respuesta, debe remitir los datos desde la capa de datos a las demás capas mediante una interfaz *callback*.

Cada vez que se va a realizar una petición a la base de datos, se debe agregar el *listener*, esperar y procesar el resultado de la petición y retirar el *listener*. Estos pasos de adición y retiro de *listeners* se recomienda definirlos en los métodos de inicio y fin de una vista y desde allí transferirse hasta la capa de datos. En el caso *Android* el *listener* se agrega en el método *onResume()* y se retira en el método *onPause()*.

Acerca del uso de *Process Dashboard*

El control y registro de las diferentes fases *PSP1* para cada iteración del proyecto se realizó mediante la herramienta *Process Dashboard*¹, la cual tiene incorporadas plantillas con cada una de las fases de *PSP1* y permite hacer estimaciones, registrar tiempos, errores y hacer análisis *post-mortem* para cada iteración, entre otras utilidades. No obstante, la falta de hábito en la utilización de la herramienta mezclada con las diferentes actividades del proyecto, ocasionó situaciones como las descritas a continuación.

El manejo del cronómetro para registrar los tiempos empleados en cada fase no fue totalmente óptimo por parte del desarrollador del proyecto, debido a que en muchos casos al tener que atender otros asuntos y levantarse del puesto de trabajo, se olvidaba detener el cronómetro, por lo tanto seguía corriendo el tiempo pero no se estaba trabajando en el proyecto. Incluso en varias ocasiones al finalizar la jornada de trabajo no se detuvo el cronómetro y este siguió contando durante toda la noche. Al día siguiente, la herramienta notifica que el cronómetro no se ha detenido durante una gran cantidad de tiempo y es necesario detenerlo y ajustar los tiempos manualmente, esto genera desfases en la información del tiempo empleado.

Al igual que la situación descrita en el párrafo anterior, sucedió el efecto contrario: existió una interrupción en el trabajo y el desarrollador tuvo que ausentarse, detuvo el cronómetro y al reiniciar labores, olvidó activarlo de nuevo, percatándose de esta situación al finalizar del día. De nuevo era necesario ajustar los tiempos manualmente para incluir las actividades no contabilizadas, ocasionando desfases.

¹ Ver <http://www.processdash.com/>

Debido a que para realizar la gran mayoría de las utilidades de la aplicación móvil, el desarrollador del proyecto consultaba ejemplos e iba adaptando o corrigiendo el código fuente, fue imposible cumplir a cabalidad con lo estipulado por *PSP* para las fases de codificación y compilación, consistente en que primero debe realizarse totalmente la codificación sin compilar y posteriormente realizar la compilación total de lo que se programó. Esto no pudo ser posible debido a que eran actividades y conceptos totalmente nuevos para el desarrollador, tales como generar un documento *PDF* directamente desde el dispositivo móvil y enviarlo por correo electrónico. Por esto fue necesario ir programando e ir probando, lo cual incorporaba implícitamente el proceso de compilación.

Frente al registro de errores, debido a que la compilación se realizó paralelamente a la codificación, los errores allí detectados se registraron en *PSP* una vez solucionados. Una situación frecuente es detectar el error durante la fase de codificación, solucionarlo y posteriormente registrar esto en *PSP* sin haber tenido en cuenta el tiempo que se tardó solucionando. Esto ocasiona que aunque se registre el tiempo total de codificación, no quede diferenciado el tiempo que se empleó solucionando los errores.

Si bien en *PSP* se propone codificar totalmente y en fases posteriores compilar y probar el código, esto resulta realmente complicado cuando se está desarrollando en un lenguaje de programación en el cual no se tiene amplia experiencia. Para el caso del presente proyecto, el desarrollo de la aplicación móvil se realizó en *Java Android* y muchos de los conceptos utilizados tuvieron que ser probados y mejorados durante la fase de codificación, de lo contrario se corría el riesgo de perder el control de la aplicación y tardar mucho más tiempo en entender lo que debería realizar la clase, ubicar el error e intentar solucionarlo.

Libros recomendados

Arquitectura Java Sólida, Cecilio Álvarez Caules, disponible en www.arquitecturajava.com/el-libro/.

Código Limpio, Manual de estilo para el desarrollo ágil de software, Robert C. Martin, Anaya Multimedia, 2012.

Desarrollo de aplicaciones mediante el *Framework* de *Spring*, Eugenia Pérez Martínez, Ediciones de la U, 2016.

Hibernate Persistencia de Objetos en JEE, Eugenia Pérez Martínez, Ediciones de la U, 2016.

Introducción al proceso software personal, Watts S. Humphrey, Editorial Addison Wesley, 2001.

Medición y Estimación del Software, Mario G. Piattini, Félix Oscar García, Javier Garzás y Marcela Genero, Editorial Alfaomega, 2008.

Patrones de diseño en Java, Laurent Debray, Ediciones ENI, 2014

UML Para Programadores Java, Robert C. Martin, Editorial Pearson Prentice Hall, 2004.

Herramientas recomendadas

Herramienta gratuita recomendada: Process Dashboard, disponible en www.processdash.com/download.

9. Bibliografía

- Abogalista. (2013). Aspectos legales a considerar en las *apps* (aplicaciones para móviles). Recuperado de <http://www.abogalista.com/recursos/temas/aspectos-legales-a-considerar-en-las-apps-aplicaciones-para-moviles.html>
- Alcázar, P. (2013). Emprendedores: las claves de la economía y el éxito profesional. Revista Emprendedores. 192. 44-49. Recuperado de <https://dialnet.unirioja.es/servlet/autor?codigo=2005899>
- Amaya, Y. (2013). Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles. Estado actual. Revista de Tecnología, 12(2), 111-124. Recuperado de <http://revistas.unbosque.edu.co/index.php/RevTec/article/view/1291>
- Android Studio. (s.f.). Meet Android Studio. Recuperado de <https://developer.android.com/studio/intro/index.html>
- Aragón, F., Cárdenas, J., Cristancho, J., Higuera, M., Marín, D., Nova, Juan., ..., Romero, A. (2009). Aspectos a considerar para adaptar el contenido y despliegue de la información. Revista avances en sistemas e informática. 6(2). 99-112. Recuperado de <http://revistas.unal.edu.co/index.php/avances/article/view/20348>
- Banco Caja Social. (2012). Ley 1581 de 2012. protección de datos personales. Recuperado de <https://www.bancocajasocial.com/abc-ley-1581-de-2012-proteccion-de-datos-personales>
- Bilib. (2013). *Apps* móviles, una fuente de negocio muy rentable. Recuperado de http://www.bilib.es/uploads/media/2828d4_articulo-bilib-apps-moviles.pdf
- Booch, G., Rumbaugh, J. y Jacobson, I. (2000). *El lenguaje unificado de modelado. Manual de referencia*, Madrid: Ed. Addison-Wesley
- CCM. (2016a). Redes-Cliente liviano. Recuperado de <http://es.ccm.net/contents/144-redes-cliente-liviano>
- CCM. (2016b). Redes-Cliente enriquecido. Recuperado de <http://es.ccm.net/contents/146-redes-cliente-enriquecido>
- CCM. (2016c). Redes-Cliente pesado. Recuperado de <http://es.ccm.net/contents/145-redes-cliente-pesado>
- Colegio Comfatolima (2016). Caja de Compensación Familiar del Tolima Comfatolima. (2016). Colegio Comfatolima. Ibagué, CO. Oficina de Sistemas de Comfatolima. Recuperado de <http://www.micolegiocomfatolima.edu.co/nosotros/historia.html>

Comfatolima (2016a). Caja de Compensación Familiar del Tolima Comfatolima. (2016). Historia de Comfatolima. Ibagué, CO. Oficina de Sistemas de Comfatolima. Recuperado de <https://www.comfatolima.com.co/historia-de-comfatolima>.

Comfatolima (2016b). Caja de Compensación Familiar del Tolima Comfatolima. (2016). Normatividad. Ibagué, CO. Oficina de Sistemas de Comfatolima. Recuperado de <https://www.comfatolima.com.co/n>.

ComScore (2016). Futuro Digital Colombia 2016, p. 25. Recuperado de: <https://www.comscore.com/lat/Prensa-y-Eventos/Presentaciones-y-libros-blancos/2015/Futuro-Digital-Colombia-2015>

Dane (2016). Población proyectada de Colombia. Recuperado de: <http://www.dane.gov.co/reloj/>

Eclipse. (s.f.). What is eclipse?. Recuperado de http://help.eclipse.org/mars/index.jsp?topic=%2Forg.eclipse.platform.doc.isv%2Fguide%2Fint_eclipse.htm

Estrategia Gobierno en Línea. (2014). Aplicaciones móviles al servicio de los ciudadanos. Recuperado de <http://estrategia.gobiernoenlinea.gov.co/623/w3-article-8310.html>

Ferguson, P., Humphrey, W., Khanejoori, S. y Macke, S. (1997). *Results of Applying the Personal Software Process*. En IEEE *Software*. 30(5). 24-31. Pittsburg: IEEE.

Gama, L., Juárez, M., Martínez, C., Ramírez, A., Torres & J., Torres, L. (2014). Plataforma de aplicaciones para documentos electrónicos a través de cómputo móvil. *Revista gerencia tecnológica informática*. 14(38). 75-84. Recuperado de <http://revistas.uis.edu.co/index.php/revistagti/article/view/4870>

Gasca Mantilla, M., Camargo Ariza, L., & Medina Delgado, B. (2014). Metodología para el desarrollo de aplicaciones móviles. *Revista Tecnura*, 18(40), 20-35. doi:<https://doi.org/10.14483/udistrital.jour.tecnura.2014.2.a02>

Hernández Sampieri, R., Fernández Collado, C., Baptista Lucio, P., (2010). *Metodología de la Investigación*. (5ª ed.). México D.F.: Mc Graw-Hill

Humphrey, W. (1996). Using a Defined and Measured Personal Software Process. En IEEE *Software*. 13(3). 77-88. Pittsburgh: IEEE

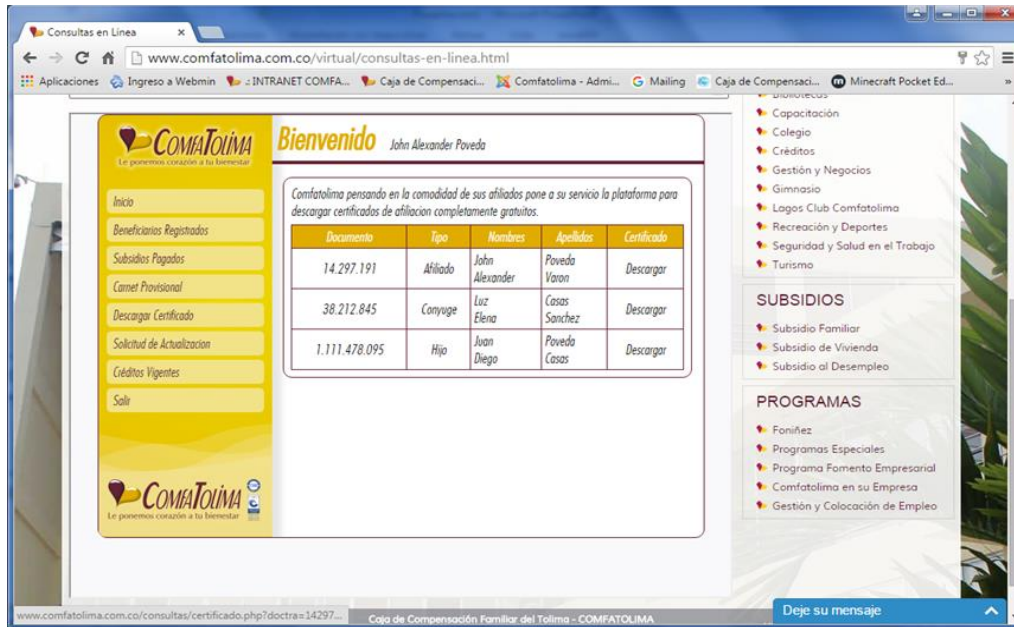
IBM. (2011). Comprender las especificaciones de los servicios web, Parte 1: SOAP. Recuperado de <https://www.ibm.com/developerworks/ssa/webservices/tutorials/ws-understand-web-services1/>

- IBM. (2012). El desarrollo de aplicaciones móviles nativas, web o híbridas. Recuperado de ftp://ftp.software.ibm.com/la/documents/gb/commons/27754_IBM_WP_Native_Web_or_hybrid_2846853.pdf
- IBM. (2014). Construya las aplicaciones móviles estilo MVC usando Bluemix. Recuperado de <http://www.ibm.com/developerworks/ssa/cloud/library/cl-mvc-mobile-bluemix-app/index.html>
- IBM. (2015). ¿Qué es SOAP?. Recuperado de http://www.ibm.com/support/knowledgecenter/es/SSKM8N_8.0.0/com.ibm.etools.mft.doc/ac55770_.htm
- Larman, C. (2003). *Uml y Patrones, una introducción al análisis y diseño orientado a objetos y al proceso unificado*. (2a Ed). México: Prentice Hall
- Maltz, J. & Barney, D. (2012). Freemium Software: A Guide for Startups. Institutional Venture Partners. Recuperado de http://www.ivp.com/assets/pdf/ivp_freemium_paper.pdf
- MinTic (2016a). Abonados e índice de penetración de telefonía móvil. Recuperado de <http://colombiatic.mintic.gov.co/estadisticas/stats.php?&pres=content&jer=1&cod=&id=89TTC>
- MinTic (2016b). Internet por tipo de acceso. Recuperado de: <http://colombiatic.mintic.gov.co/estadisticas/stats.php?&pres=content&jer=1&cod=&id=26TTC>
- MMA. (s.f.). Libro Blanco de Apps. Recuperado de: <http://mmaspain.com/libros-blancos/>
- Netbeans. (s.f.). Netbeans IDE – The smarter and faster way to code. Recuperado de <https://netbeans.org/features/index.html>
- Nueva Eps (2014), Más de 18.000 usuarios registrados en EPS Móvil, Bogotá, CO. Recuperado de <http://www.nuevaeps.com.co/Noticias/Breves/VisualizarBreves/tabid/206/ArticleId/659/Mas-de-18-mil-usuarios-registrados-en-NUEVA-EPS-MOVIL.aspx>
- Pressman, R. (2010). *Ingeniería del Software, un enfoque práctico*. (7ª ed.). México D.F.: Mc Graw-Hill
- Ramírez, R. (2013), Métodos para el desarrollo de aplicaciones móviles, Recuperado de [https://www.exabyteinformatica.com/uoc/Informatica/Tecnologia_y_desarrollo_en_dispositivos_moviles/Tecnologia_y_desarrollo_en_dispositivos_moviles_\(Modulo_4\).pdf](https://www.exabyteinformatica.com/uoc/Informatica/Tecnologia_y_desarrollo_en_dispositivos_moviles/Tecnologia_y_desarrollo_en_dispositivos_moviles_(Modulo_4).pdf)

- Rojas, M., Yuste, Elisa., Vásquez, J., Celaya, J., (2014). *Nuevos modelos de negocio en la era digital*, Madrid: Liber
- Salz, P., y Moranz, J. (2013). *The everything guide to mobile apps*. Avon: Adams Media
- Sommerville, I. (2011). *Ingeniería de Software*. (9ª ed.). Atlacomulco: Pearson Educación
- Superintendencia de Industria y Comercio. (s.f.). Constitución Nacional. Recuperado de <http://www.sic.gov.co/drupal/constitucion-nacional>
- Superintendencia de Industria y Comercio. (2008a). Ley estatutaria 1266 del 31 de diciembre de 2008. Recuperado de [http://www.sic.gov.co/drupal/sites/default/files/files/ley1266_31_12_2008\(1\).pdf](http://www.sic.gov.co/drupal/sites/default/files/files/ley1266_31_12_2008(1).pdf)
- Superintendencia de Industria y Comercio. (2008b). Sentencia C 1011 de 2008. Recuperado de http://www.sic.gov.co/drupal/recursos_user/documentos/sentencia_C_1011_de_2008.pdf
- Superintendencia del Subsidio Familiar. (s.f.). Directorio de Cajas de Compensación. Recuperado de <http://www.ssf.gov.co/wps/portal/ES/Cajas-Comp-familiar/cajas-afiliado/infocajas>
- Superintendencia del Subsidio Familiar. (2014). Circular 010 de 2014. Recuperado de www.ssf.gov.co/wps/docs/.../PDF_circulares/Circulares_2014/CE_0010_2014.pdf
- Techcetera. (2014). *Colombia está inundada de Internet*. Recuperado de <http://techcetera.co/colombia-esta-inundada-de-internet/>
- Unipymes. (2015). Las Apps el presente y el futuro de los procesos productivos de una empresa. Recuperado de <https://www.unipymes.com/las-apps-el-presente-y-el-futuro-de-los-procesos-productivos-de-una-empresa/>
- Washke, M., (2015). *How Clouds Hold IT Together*, New York: CA Technologies
- We Make Apps., We make apps faster, smarter, safer. (2017). 9 motivos por lo que un cliente abandona una app. F5 Networks Inc., US. Recuperado de <https://wemakeappsgo.f5.com/es/personas-y-tecnologia/9-motivos-por-los-que-un-cliente-abandona-una-app/>.
- Woodbridge, R., (s.f.). 9 Mobile Business Models that you can use right now to generate revenue. Recuperado de <http://untether.tv/2010/8-mobile-business-models-that-you-can-use-right-now-to-generate-revenue/>

10. Anexos

Anexo 1. Consulta datos básicos afiliado. www.comfatolima.com.co.



Bienvenido John Alexander Poveda

Comfatolima pensando en la comodidad de sus afiliados pone a su servicio la plataforma para descargar certificados de afiliación completamente gratuitos.

Documento	Tipo	Nombres	Apellidos	Certificado
14.297.191	Afiliado	John Alexander	Poveda Varon	Descargar
38.212.845	Conyuge	Luz Elena	Casas Sanchez	Descargar
1.111.478.095	Hijo	Juan Diego	Poveda Casas	Descargar

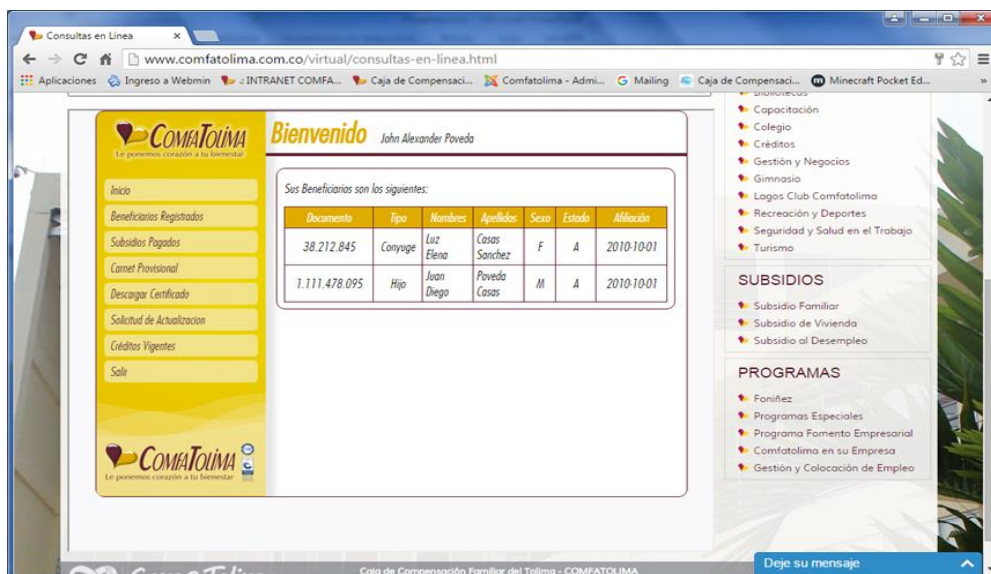
SUBSIDIOS

- Subsidio Familiar
- Subsidio de Vivienda
- Subsidio al Desempleo

PROGRAMAS

- Fonifex
- Programas Especiales
- Programa Fomento Empresarial
- Comfatolima en su Empresa
- Gestión y Colocación de Empleo

Anexo 2. Consulta datos básicos núcleo familiar. www.comfatolima.com.co.



Bienvenido John Alexander Poveda

Sus Beneficiarios son los siguientes:

Documento	Tipo	Nombres	Apellidos	Sexo	Estado	Afiliación
38.212.845	Conyuge	Luz Elena	Casas Sanchez	F	A	2010-10-01
1.111.478.095	Hijo	Juan Diego	Poveda Casas	M	A	2010-10-01

SUBSIDIOS

- Subsidio Familiar
- Subsidio de Vivienda
- Subsidio al Desempleo

PROGRAMAS

- Fonifex
- Programas Especiales
- Programa Fomento Empresarial
- Comfatolima en su Empresa
- Gestión y Colocación de Empleo

Anexo 3. Consulta subsidios pagados. www.comfatolima.com.co.



COMFATOLIMA
Le ponemos corazón a tu bienestar

Bienvenido John Alexander Poveda

Relación de los Subsidios Pagados

Periodo	Beneficiario	Giro	Valor
201505	Juan Diego Poveda Casas	201505	\$ 24.782
201506	Juan Diego Poveda Casas	201506	\$ 24.782
201507	Juan Diego Poveda Casas	201507	\$ 24.782
201508	Juan Diego Poveda Casas	201508	\$ 24.782
201509	Juan Diego Poveda Casas	201509	\$ 24.782

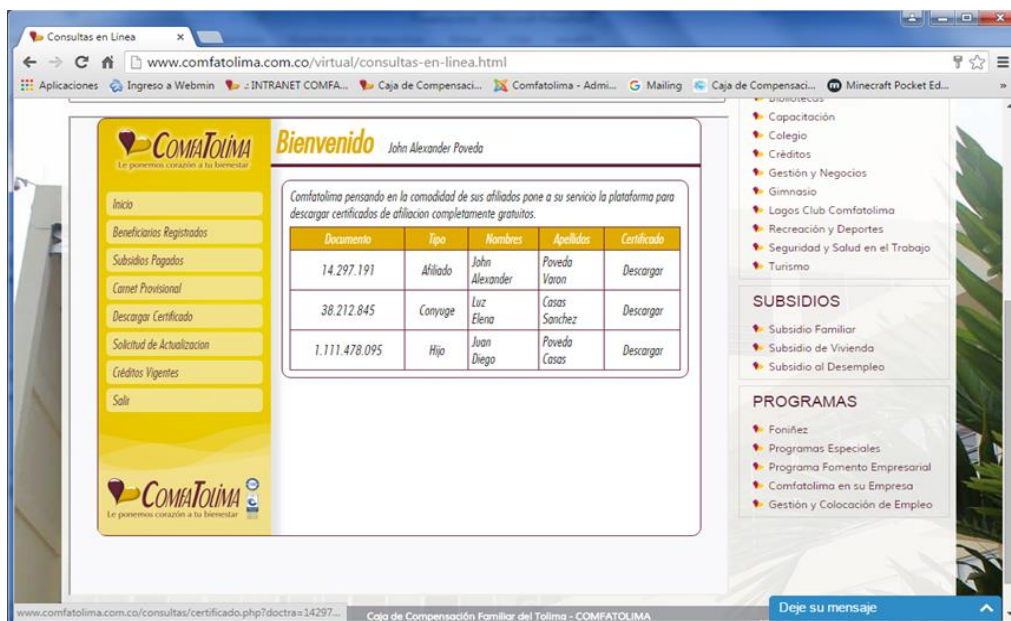
SUBSIDIOS

- Subsidio Familiar
- Subsidio de Vivienda
- Subsidio al Desempleo

PROGRAMAS

- Foníñez
- Programas Especiales
- Programa Fomento Empresarial
- Comfatolima en su Empresa
- Gestión y Colocación de Empleo

Anexo 4. Descarga de certificados de afiliación. www.comfatolima.com.co.



COMFATOLIMA
Le ponemos corazón a tu bienestar

Bienvenido John Alexander Poveda

Comfatolima pensando en la comodidad de sus afiliados pone a su servicio la plataforma para descargar certificados de afiliación completamente gratuitos.

Documento	Tipo	Nombres	Apellidos	Certificado
14.297.191	Afiliado	John Alexander	Poveda Varon	Descargar
38.212.845	Conyuge	Luz Elena	Casas Sanchez	Descargar
1.111.478.095	Hijo	Juan Diego	Poveda Casas	Descargar

SUBSIDIOS

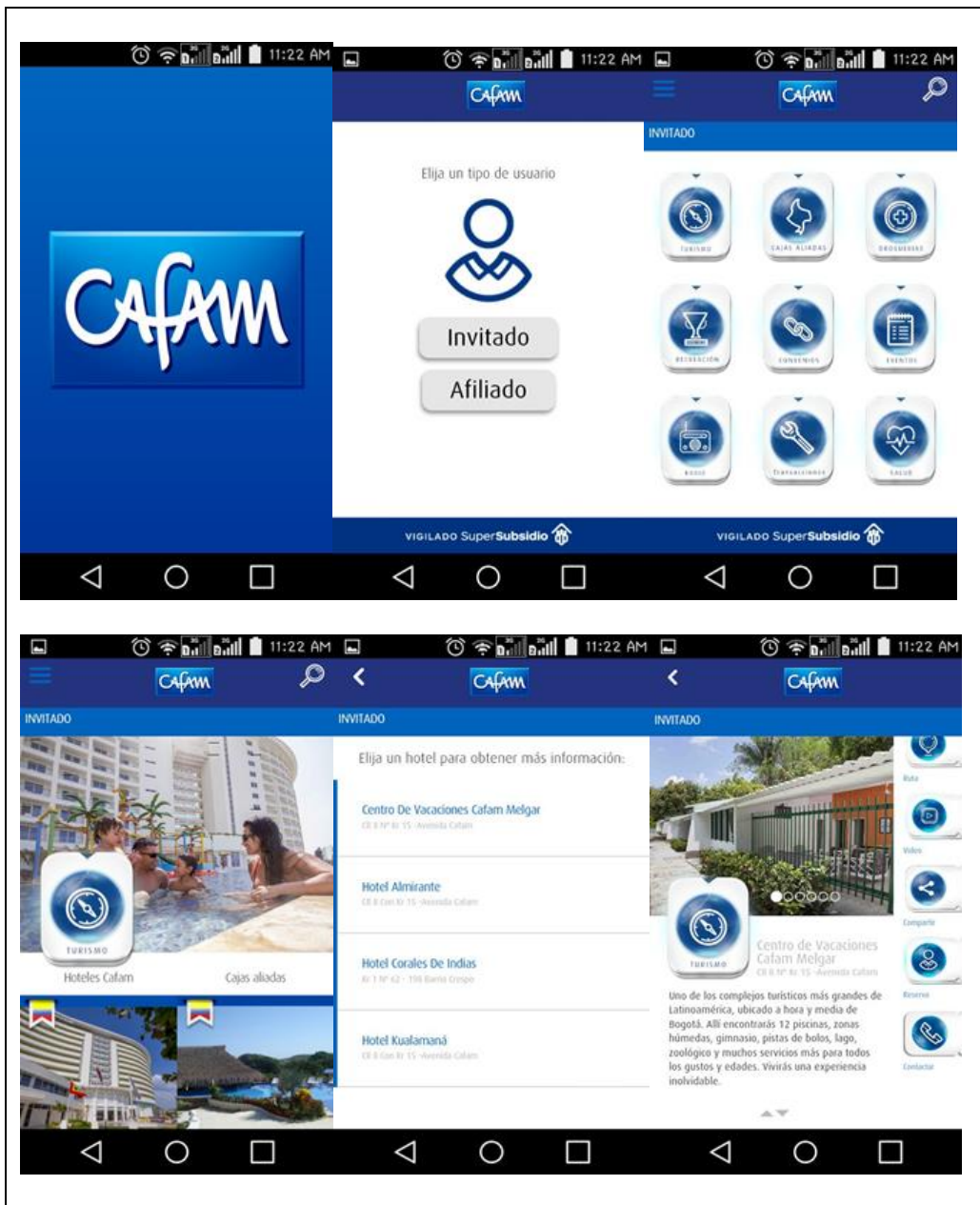
- Subsidio Familiar
- Subsidio de Vivienda
- Subsidio al Desempleo

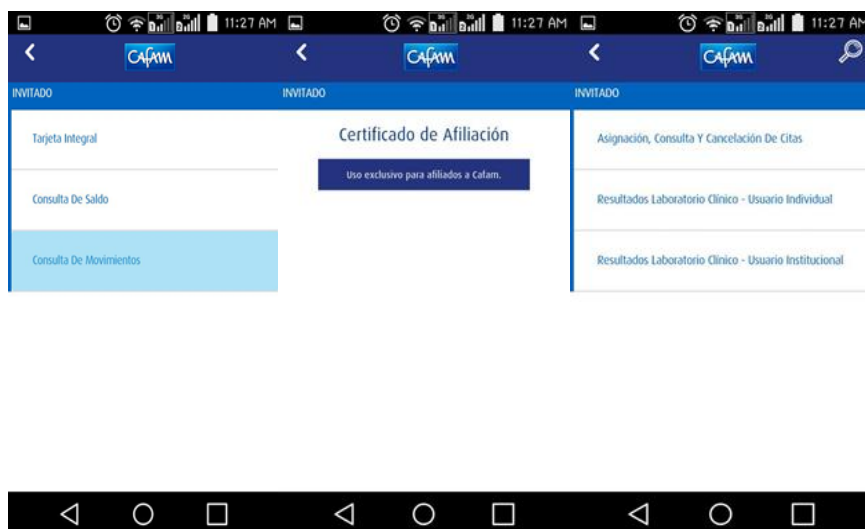
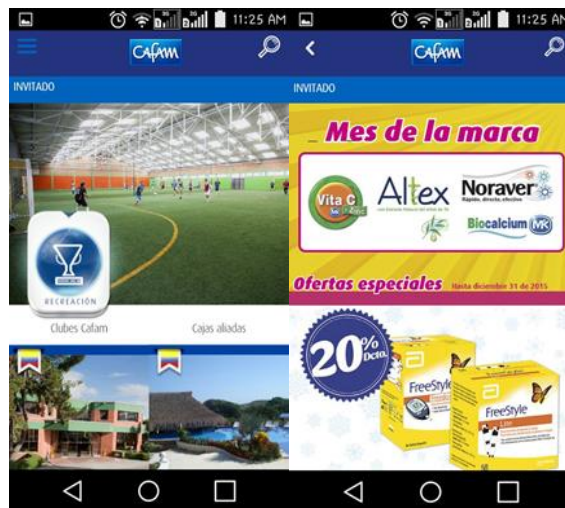
PROGRAMAS

- Foníñez
- Programas Especiales
- Programa Fomento Empresarial
- Comfatolima en su Empresa
- Gestión y Colocación de Empleo

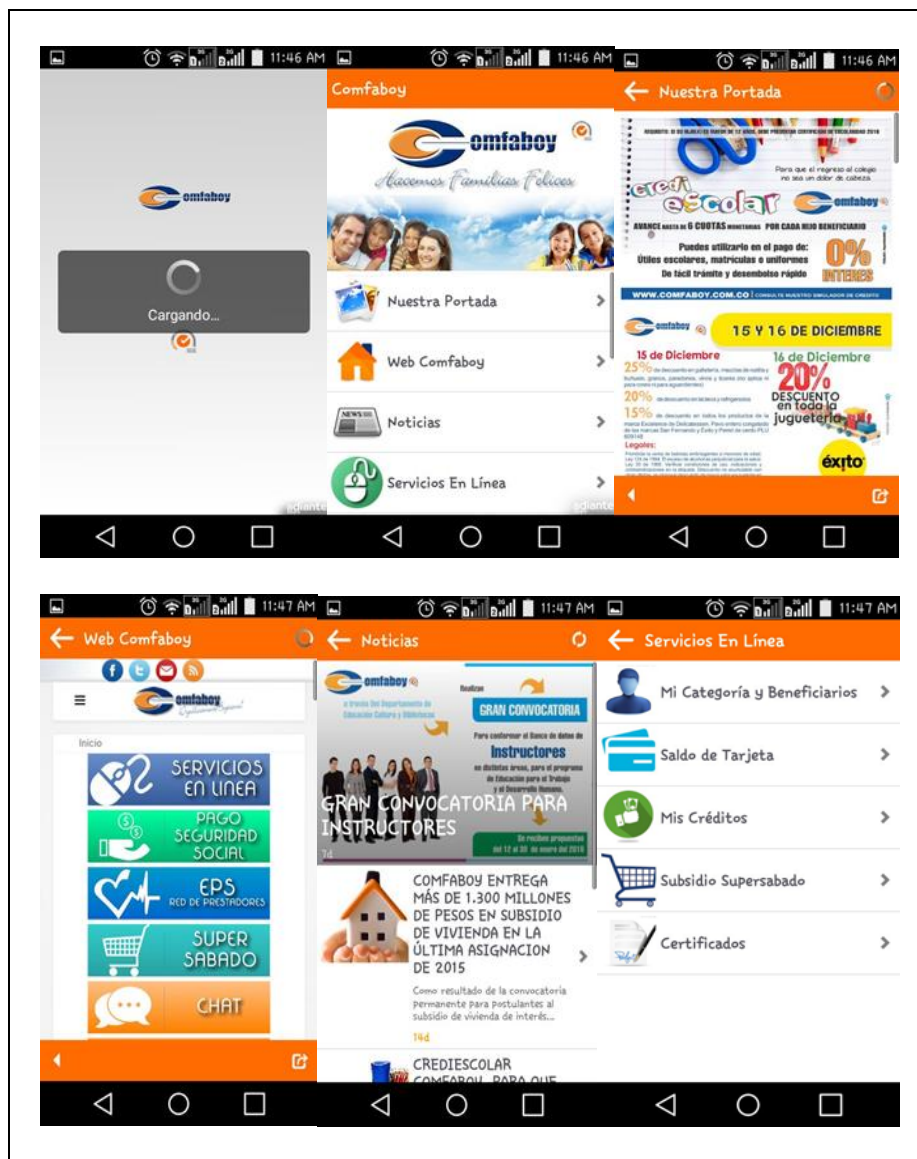
Anexo 5. Imágenes de aplicaciones móviles de Cajas de Compensación

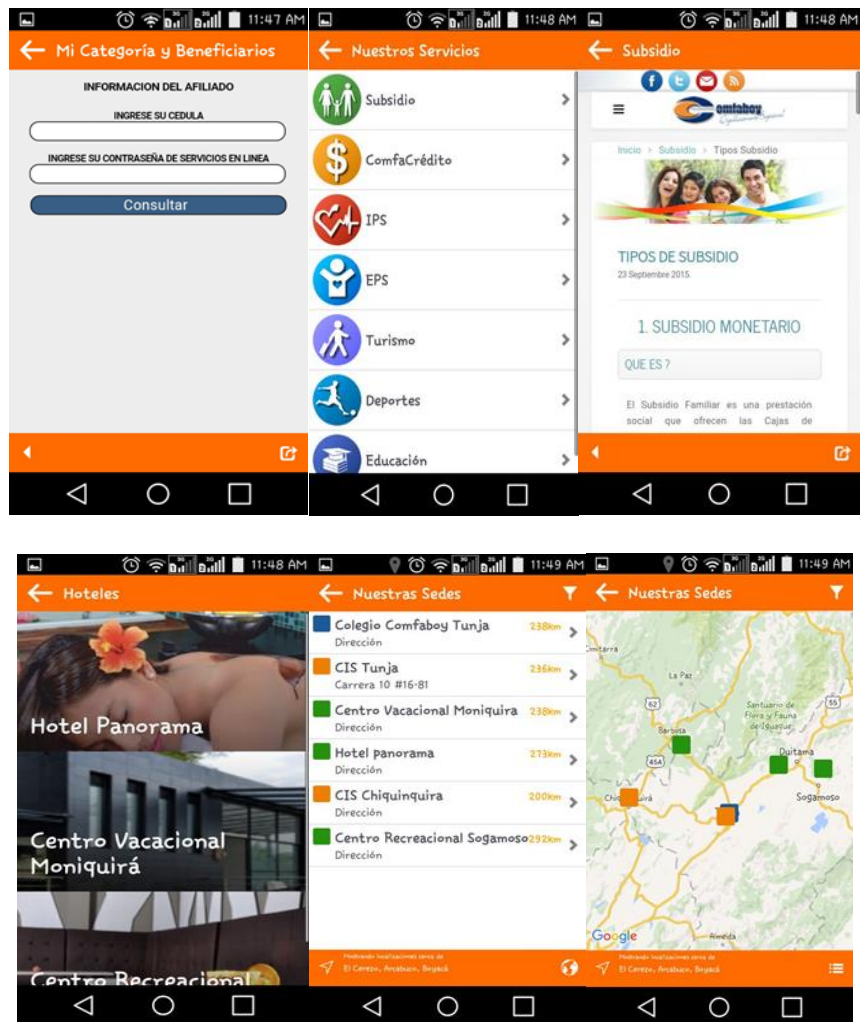
Cafam



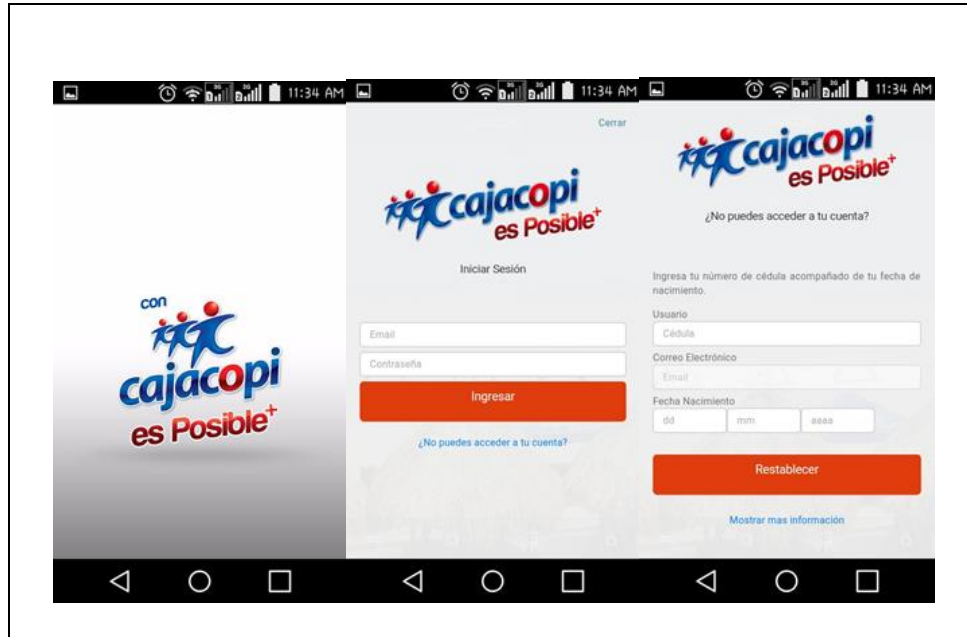


Comfaboy

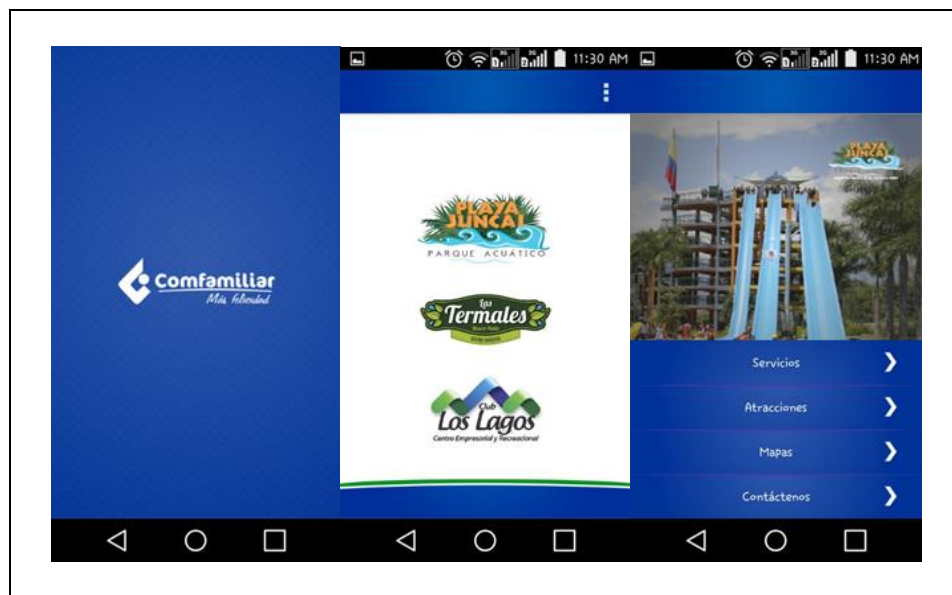


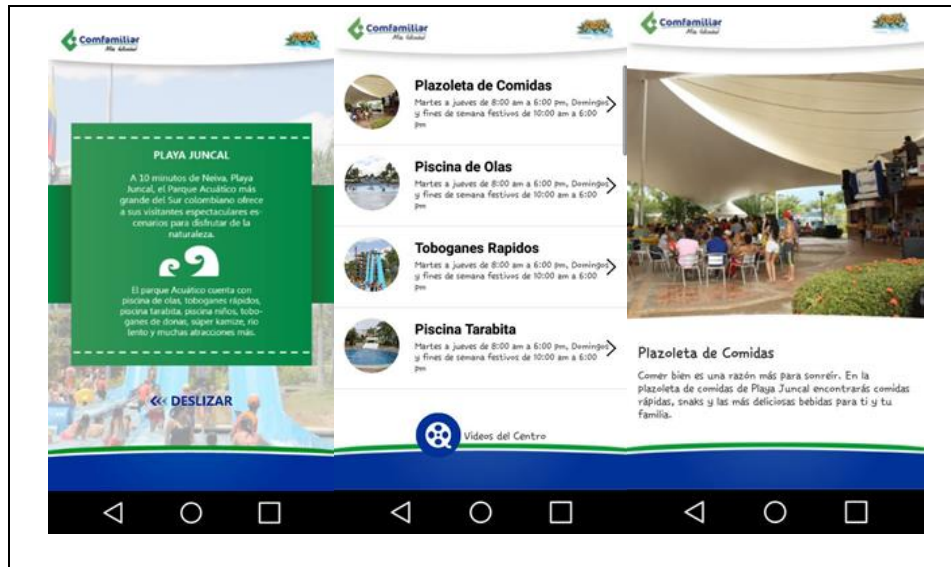


Cajacopi

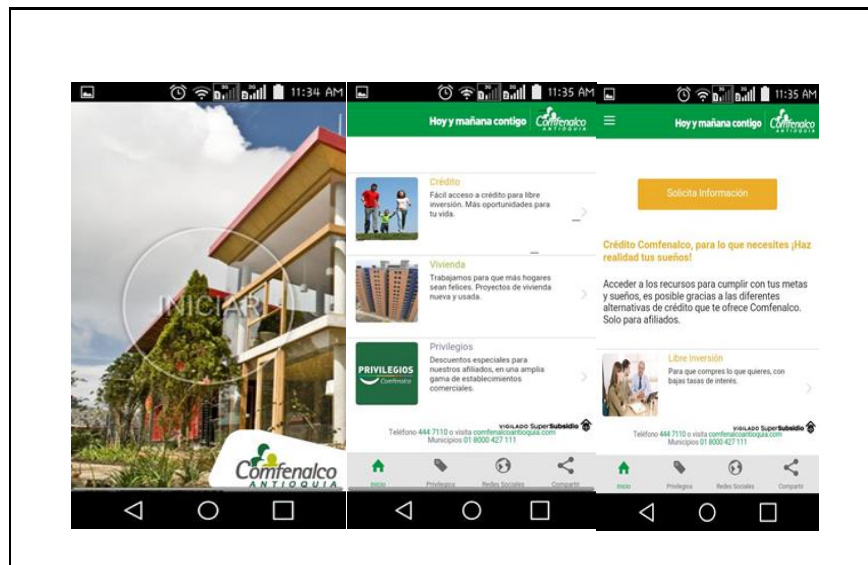


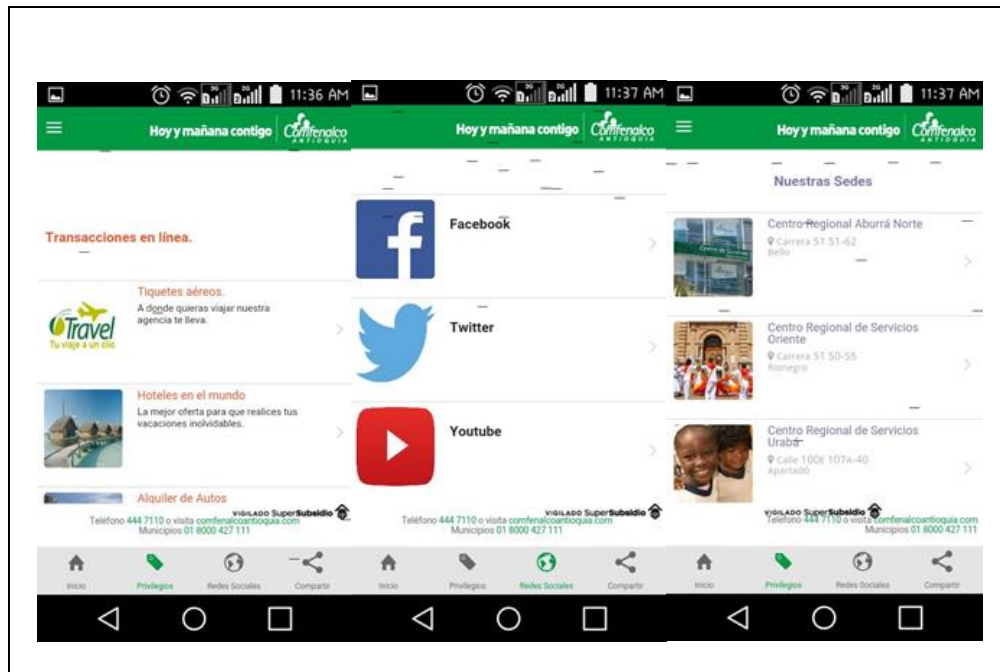
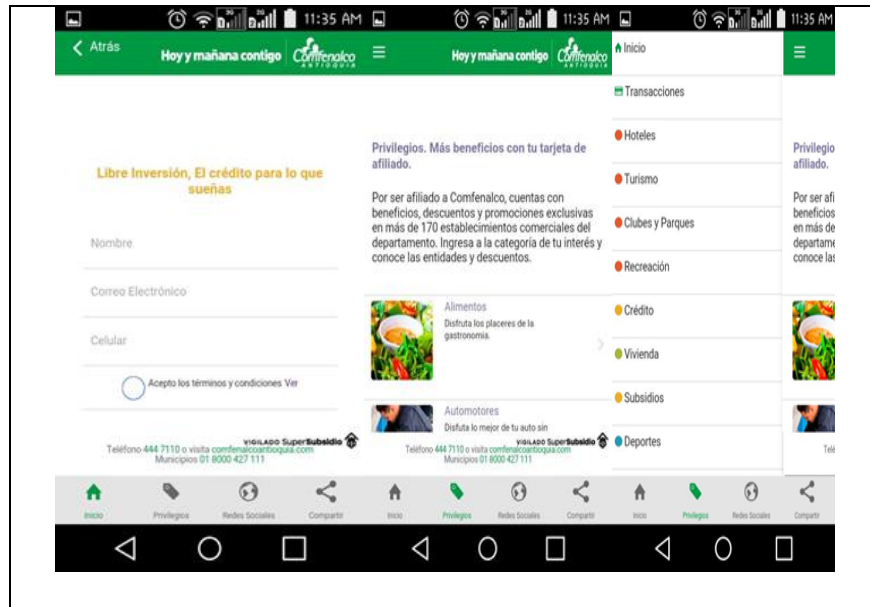
Comfamiliar Huila

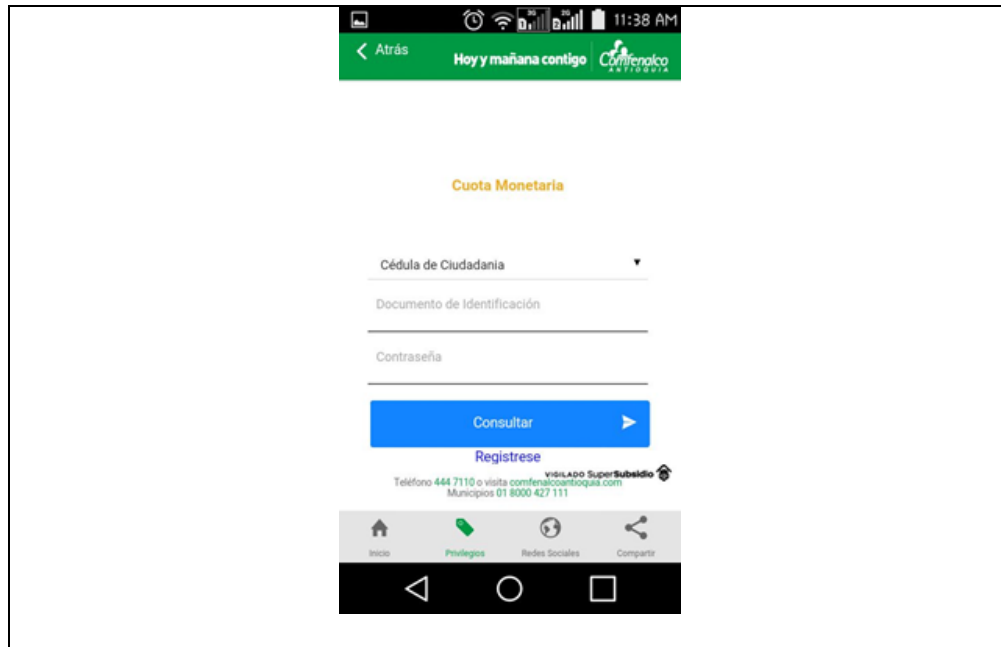




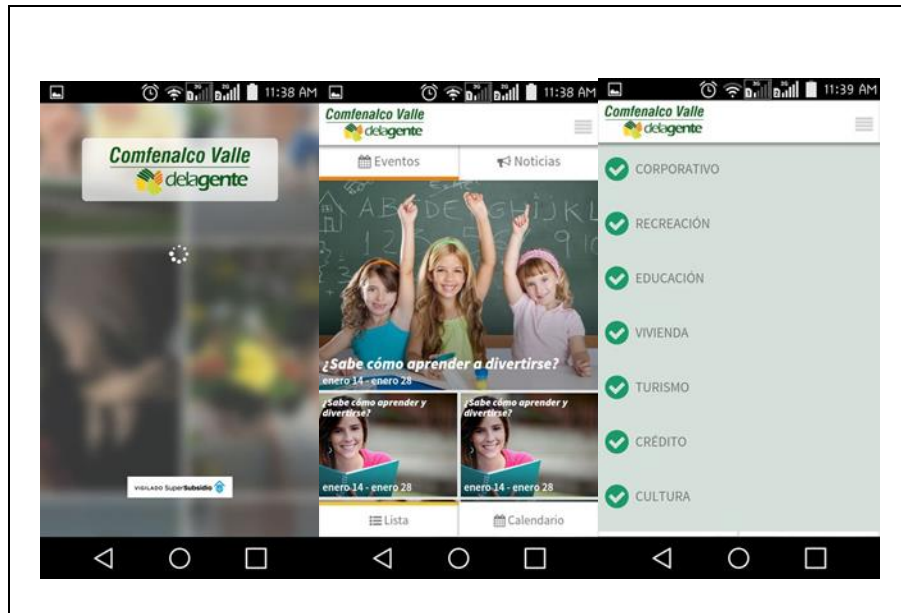
Comfenalco Antioquia

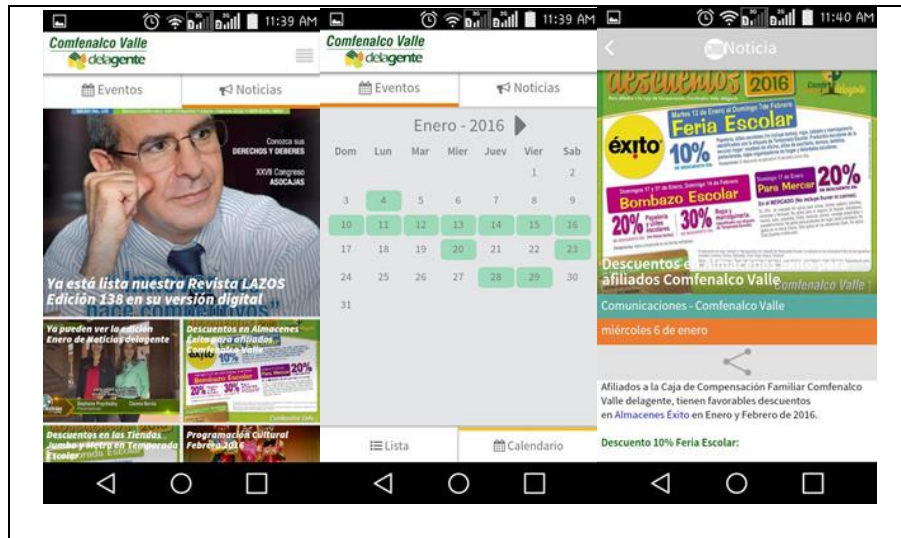




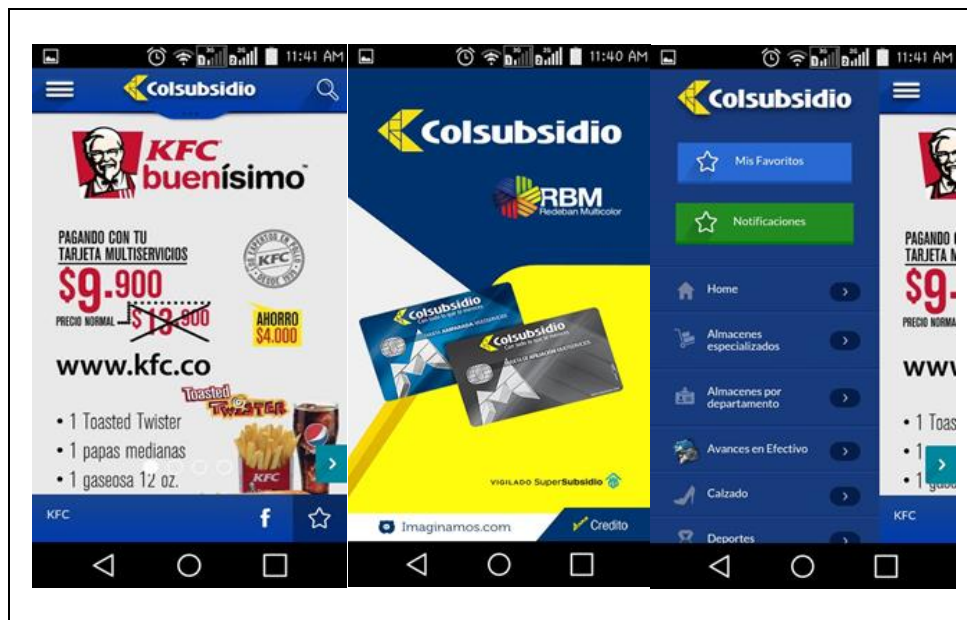


Comfenalco Valle

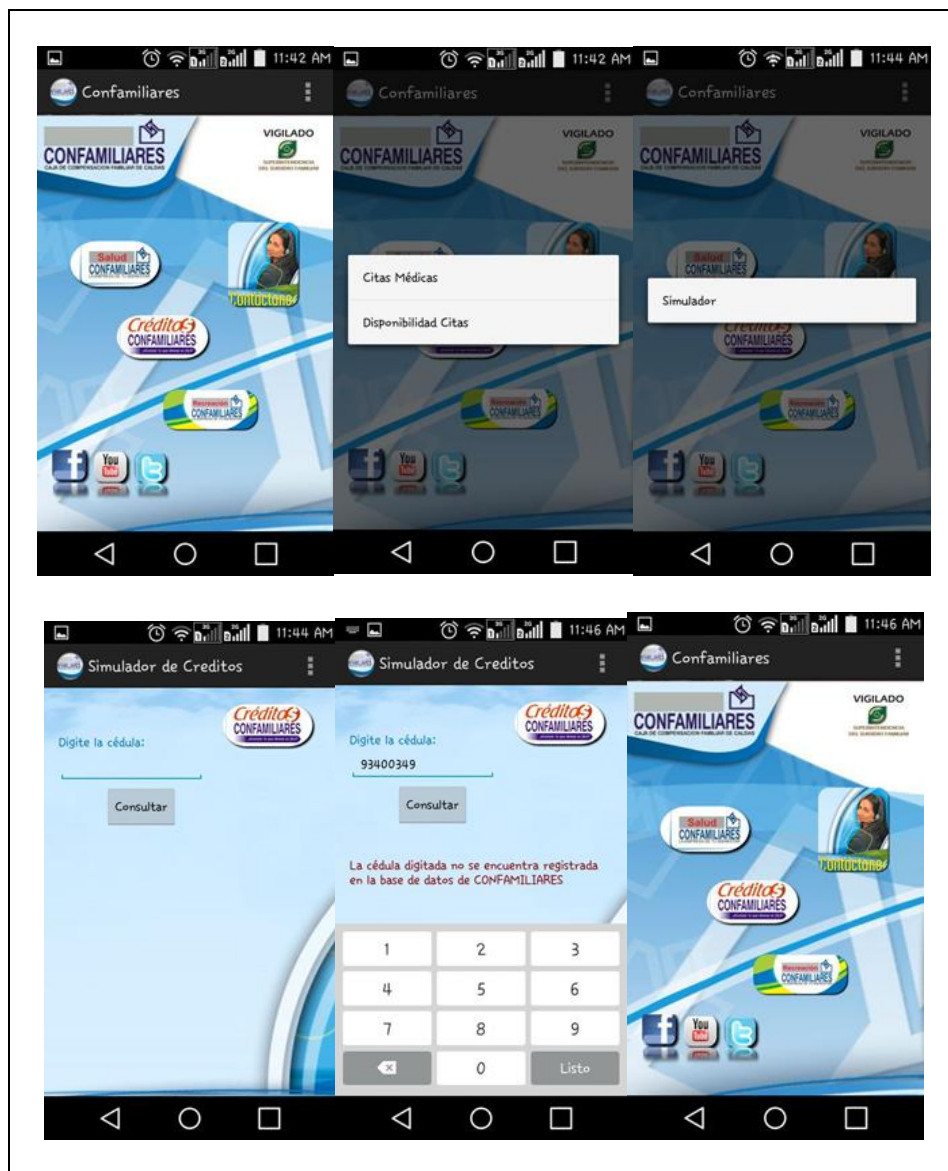




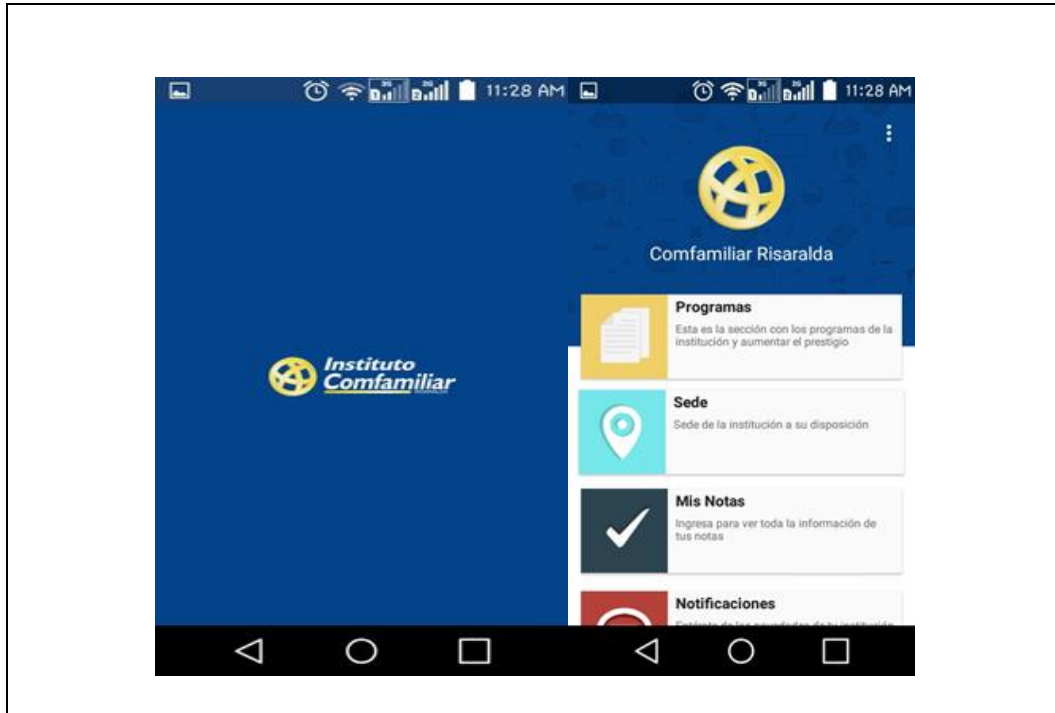
Colsubsidio



Confamiliares



Confamiliar Risaralda



Anexo 6. Especificación de requisitos (documento adjunto)

Anexo 7. Documento de arquitectura (documento adjunto)

Anexo 8. Casos de uso textuales (documento adjunto)

Anexo 9. Pruebas unitarias y de integración (documento adjunto)

Anexo 10. Pruebas de adaptación y usabilidad (documento adjunto)

Anexo 11. PSP Iteración 1 (documento adjunto)

Anexo 12. PSP Iteración 2 (documento adjunto)

Anexo 13. PSP Iteración 3 (documento adjunto)

Anexo 14. PSP Iteración 4 (documento adjunto)

Anexo 15. PSP Iteración 5 (documento adjunto)

Anexo 16. PSP Iteración 6 (documento adjunto)

Anexo 17. PSP Iteracion 7 (documento adjunto)

Anexo 18. PSP Iteracion 8 (documento adjunto)

Anexo 19. Manual de usuario (documento adjunto)

Anexo 20. Manual técnico (documento adjunto)