



PROCESO DE TEST ÁGIL PARA EL DESARROLLO DE VIDEOJUEGOS  
SERIOS EDUCATIVOS

DAVID ALEJANDRO RAMÍREZ MARÍN

UNIVERSIDAD AUTÓNOMA DE MANIZALES

FACULTAD DE INGENIERÍA

MAESTRÍA EN GESTIÓN Y DESARROLLO DE PROYECTOS DE SOFTWARE

MANIZALES

2022

PROCESO DE TEST ÁGIL PARA EL DESARROLLO DE VIDEOJUEGOS  
SERIOS EDUCATIVOS

Autor

DAVID ALEJANDRO RAMÍREZ MARÍN

Proyecto de grado para optar al título de Magister en gestión y desarrollo de  
proyectos de software

Tutora

SANDRA VICTORIA HURTADO GIL

UNIVERSIDAD AUTÓNOMA DE MANIZALES

FACULTAD DE INGENIERÍA

MAESTRÍA EN GESTIÓN Y DESARROLLO DE PROYECTOS DE SOFTWARE

MANIZALES

2022

## **DEDICATORIA**

Dedicar un triunfo, dedicar una victoria o dedicar un simple paso más en la vida a una persona que no nos acompaña en este mundo material es algo que los comunistas materialistas no osamos hacer por nuestra concepción material y no espiritual de la vida, pero ¿quién por amor a alguien no ha realizado las cosas menos pensadas? Por eso hoy dedico este paso en mi vida a la mujer que ha sido fuente de total e inagotable apoyo, a esa mujer que hoy más que nunca quisiera que celebrara junto a mí en este mundo, madre, Angela María Marín Cuervo, te dedico este paso por lo que hiciste en tu vida material, por lo que hiciste en mí, esperando estar equivocado y que después de este mundo material estes en un lugar que te permita ver cada uno de estos pasos o logros que ayudaste a forjar.

## **AGRADECIMIENTOS**

Por forjar disciplina y entrega al pueblo en cada uno de los aspectos de nuestras vidas, le agradezco al Movimiento Por La Defensa Derechos Del Pueblo (MODEP), movimiento que desde la militancia política me apoyo en los procesos de formación para que otro mundo en mí sea posible. Que el conocimiento y el poder iluminen el sendero de liberación de los pueblos.

Esta investigación recibió apoyo del Fondo de Becas de Investigación Manizales + Innovadora, creado por la Alcaldía de Manizales y Manizales Campus Universitario. Las opiniones, tesis y argumentos expresados son de propiedad exclusiva de el/los autores y no representan el punto de vista de la Alcaldía de Manizales ni de ninguna de las instituciones que hacen parte del programa Manizales Campus Universitario.

## RESUMEN

**Objetivo:** Diseñar un proceso de test ágil específico para video juegos serios educativos, que verifique y valide la usabilidad y la calidad del videojuego.

**Metodología:** Por medio de mapeo sistemático y encuesta con el sector de la industria del videojuego en la región se determinaron los procedimientos y prácticas actuales para el diseño de un proceso de prueba ágil para videojuegos serios educativos, el proceso fue validado por evaluación de expertos y validación en la implementación del proceso en el desarrollo de un proyecto.

**Resultados:** Se identifican metodologías para el desarrollo de videojuegos serios, artefactos y tipos de pruebas para el aseguramiento de la calidad, se diseña el proceso prueba de videojuegos serios educativos; este proceso consta de diferentes tipos de prueba involucrando roles internos y externos al proceso de desarrollo, en donde se aplican heurísticas para la medición de usabilidad, jugabilidad y experiencia de aprendizaje; el proceso se evaluó de forma satisfactoria por expertos con retroalimentaciones para el involucramiento de los clientes e interesados; en la validación se observa una mayor aplicabilidad de pruebas a diferencia de lo aplicado por el grupo control.

**Conclusiones:** La madurez de aplicación del proceso determinara mejores mediciones para realizar mejoras al proceso en su implementación, se observa ausencia de la aplicabilidad de pruebas automatizadas, independiente de su definición como actividad en el proceso; sin embargo, de acuerdo con su validación, se determina la complementariedad del proceso de prueba para los procesos de desarrollo de videojuegos serios educativos.

**Palabras Claves:** Prueba ágil, Videojuego Serio, Videojuego Educativo, Proceso de prueba.

## ABSTRACT

**Objective:** Design a specific agile test process for serious educational video games, which verifies and validates the usability and quality of the video game.

**Methodology:** Through systematic mapping and survey with the video game industry sector in the region, current procedures and practices were determined for the design of an agile testing process for serious educational video games, the process was validated by expert evaluation and validation in the implementation of the process in the development of a project.

**Results:** Methodologies for the development of serious videogames, artifacts and types of tests for quality assurance are identified; the trial process for serious educational videogames is designed; This process consists of different types of tests involving internal and external roles to the development process, where heuristics are applied to measure usability, playability and learning experience; the process was satisfactorily evaluated by experts with feedback for the involvement of customers and stakeholders; in the validation a greater applicability of tests is observed in contrast to that applied by the control group.

**Conclusion:** The maturity of the application of the process will determine better measurements to make better the process in its implementation, there is an absence of the applicability of automated tests, independent of its definition as an activity in the process; however, according to its validation, it is observed that it determines the complementarity of the testing process for serious educational video game development processes.

**Keywords:** Agile Test, Serious Video Game, Educational Video Game, Test Process.

## CONTENIDO

1	PRESENTACIÓN.....	15
2	ANTECEDENTES.....	16
3	ÁREA PROBLEMÁTICA Y PREGUNTA DE INVESTIGACIÓN .....	20
3.1	PREGUNTA DE INVESTIGACIÓN.....	24
4	JUSTIFICACIÓN.....	25
5	MARCO TEÓRICO.....	27
6	OBJETIVOS.....	29
6.1	OBJETIVO GENERAL.....	29
6.2	ESPECÍFICOS .....	29
7	METODOLOGÍA .....	30
7.1	DISEÑO DE PROCESO DE TEST ÁGIL .....	30
7.1.1	Planear .....	30
7.1.2	Hacer.....	31
7.1.3	Verificar.....	32
7.1.4	Actuar .....	34
8	RESULTADOS.....	35
8.1	RESULTADOS DEL MAPEO SISTEMÁTICO .....	35
8.1.1	Necesidades Para Revisar .....	35
8.1.2	Preguntas De Investigación .....	35
8.1.3	Diseño De Búsqueda.....	36
8.1.4	Cadena De Búsqueda.....	37
8.1.5	Selección De Estudios .....	37

8.1.6	Extracción De Datos .....	43
8.1.7	Análisis Y Resultados Del Mapeo Sistemático .....	58
8.2	ENCUESTAS SECTOR INDUSTRIA DE VIDEOJUEGOS .....	58
8.3	DISEÑO DE PROCESO DE PRUEBA DE VIDEOJUEGOS SERIOS EDUCATIVOS – AVIGATE .....	64
8.4	EVALUACIÓN POR EXPERTOS .....	69
8.5	VALIDACIÓN DEL PROCESO DE PRUEBA EN EL DESARROLLO DE VIDEOJUEGO .....	74
9	DISCUSIÓN DE RESULTADOS .....	80
10	CONCLUSIONES .....	84
11	RECOMENDACIONES .....	86
12	BIBLIOGRAFÍA .....	87
13	ANEXOS .....	94



## LISTA DE TABLAS

Tabla 1. Prácticas y actividades según tasa de éxito en proyectos de software .....	22
Tabla 2. Metodologías tradicionales vs Metodologías Ágiles.....	23
Tabla 3. Relación métodos de desarrollo y estadísticas de defectos .....	24
Tabla 4. Necesidades para revisar .....	35
Tabla 5. Preguntas de investigación .....	36
Tabla 6. Cadenas de búsqueda.....	37
Tabla 7. Filtros de búsqueda.....	38
Tabla 8 Documentos seleccionados.....	40
Tabla 9 Tipos de pruebas identificadas .....	45
Tabla 10 Listado de Heurísticas aplicadas en desarrollo.....	52
Tabla 11 Comparación general de grupos de desarrollo y pruebas .....	77
Tabla 12 Comparación cuantitativa de resultados de los grupos.....	78

## LISTA DE FIGURAS

Figura 1. Fases del proceso Huddle.....	17
Figura 2. Flujo de trabajo de la Metodología SUM.....	17
Figura 3. Ciclo de vida de CASCRUM .....	18
Figura 4. Ingresos de Juegos de PC y Consolas .....	20
Figura 5. Ciclo de vida de PDCA (PHVA) .....	30
Figura 6. Pasos de Mapeo Sistemático .....	31
Figura 7. Marco conceptual de la implementación de SPEM 2.0 .....	32
Figura 8 Marco conceptual de la implementación de SPEM 2.0 .....	38
Figura 9 Proceso de evaluación UX-Cube .....	44
Figura 10 Proceso User-Centred Design (UCD) .....	46
Figura 11 Modelo diseño de juegos educativos recopilado.....	47
Figura 12 Propuesta de modelo de diseño de juegos educativos.....	48
Figura 13 Propuesta fases adaptadas de GeoPGD.....	49
Figura 14 Modelo de metodología .....	50
Figura 15 Modelo de fase de pruebas.....	51
Figura 16 CheckList Game Accessibility.....	54
Figura 17 Testing checklist.....	54
Figura 18 Ciclo de vida para juego educativo móvil.....	55
Figura 19 Ciclo básico de producción de videojuegos .....	56
Figura 20 Múltiples ciclos de producción .....	57
Figura 21 Ciclo de vida para el desarrollo de videojuegos.....	57

Figura 22 Respuestas ¿Alguno de los videojuegos desarrollados se podría identificar como un videojuego serio? (Videojuego Serio: Educativo, formativo, Capacitación, terapéutico) .....	59
Figura 23 Respuestas ¿Para la gestión de proyectos de desarrollo de videojuegos, la empresa y el equipo de desarrollo hace uso del marco de trabajo SCRUM? .....	60
Figura 24 Respuestas ¿Qué ciclo de vida de desarrollo de software se implementa en el desarrollo de videojuegos? .....	60
Figura 25 Respuestas ¿En la empresa se tiene definido o se sigue alguna metodología de desarrollo de videojuegos? .....	61
Figura 26 Respuestas ¿Existe en su empresa algún proceso de prueba definido específicamente para videojuegos? .....	61
Figura 27 Respuestas ¿En la empresa en la cual labora, se realiza automatización de pruebas para los videojuegos?.....	62
Figura 28 Respuestas Identificación de tipos de pruebas aplicadas .....	63
Figura 29 Respuestas Heurísticas jugabilidad y usabilidad.....	63
Figura 30 Respuestas Pruebas para validar pedagogía del VJS .....	64
Figura 31 Esquema general de ciclo de desarrollo .....	64
Figura 32 Modelamiento proceso en EPF Composer.....	65
Figura 33 Diagrama proceso de pruebas AVIGATE.....	67
Figura 34 Nivel de formación de expertos evaluadores .....	70
Figura 35 ¿Cuál es su tiempo de experiencia en el área de Ingeniería de Software? (En años) .....	70
Figura 36 ¿Cuál es su tiempo de experiencia en el área de Calidad del Software? (en años) .....	71
Figura 37 Resultado encuesta actividades y flujo de forma lógica .....	72

Figura 38 ¿Cree usted que el proceso de prueba diseñado abarca las actividades generales de un proceso de prueba? .....	72
Figura 39¿Cree usted que la descripción gráfica del proceso permite un entendimiento general del proceso? .....	73
Figura 40 ¿Cree usted que el proceso se enmarca en los principios ágiles? .....	73
Figura 41 ¿Cree usted que los roles definidos en el proceso son adecuados y suficientes?.	74
Figura 42 Desarrollo G1 Batalla del Puente de Boyacá .....	75
Figura 43 Desarrollo G2 Revuelta de los comuneros.....	75
Figura 44 Modelamiento terreno .....	76

## LISTA DE ANEXOS

Anexo 1 Elementos generales del proceso .....	94
Anexo 2 Rol de desarrollador .....	95
Anexo 3 Rol de experto de conocimiento .....	95
Anexo 4 Rol de probador .....	96
Anexo 5 Rol de usuario probador.....	96
Anexo 6 Artefacto casos de pruebas.....	97
Anexo 7 Requisitos y elementos de diseño .....	97
Anexo 8 Artefacto evaluación de aprendizaje.....	98
Anexo 9 Artefacto incremento de videojuego.....	98
Anexo 10 Artefacto pruebas automatizadas .....	99
Anexo 11 Artefacto reporte de incidencias .....	99
Anexo 12 Artefacto reporte de heurísticas .....	100
Anexo 13 Tarea de aplicación de heurísticas .....	100
Anexo 14 tarea de automatizar pruebas.....	101
Anexo 15 Tarea de diseño de casos de prueba .....	101
Anexo 16 Tarea de diseño evaluación de aprendizaje.....	102
Anexo 17 Tarea de ejecución de casos de prueba .....	102
Anexo 18 Tarea de entrega de incremento .....	103
Anexo 19 Tarea de evaluación de aprendizaje .....	103
Anexo 20 Artefacto formato de caso de prueba .....	104
Anexo 21 Artefacto formato de reporte de incidencia .....	104
Anexo 22 Artefacto guía de aplicación de heurísticas parte 1 .....	105

Anexo 23 Artefacto guía de aplicación de heurísticas parte 2 .....	106
Anexo 24 Artefacto Biblioteca de pruebas parte 1 .....	107
Anexo 25 Artefacto Biblioteca de pruebas parte 2.....	108

## **1 PRESENTACIÓN**

El actual documento contiene la estructura determinada para el desarrollo del informe final del proyecto, en los antecedentes se realiza un contexto del estado de los procesos de desarrollo de videojuegos y sus prácticas de aseguramiento de la calidad, entre ellas las pruebas; posteriormente en el área problemática se revisa el estado de la industria del desarrollo de software y las implicaciones por la ausencia de prácticas de prueba o de calidad del software.

En la justificación se evidencia la carencia de aspectos procedimentales para el aseguramiento de la calidad del software en la industria de la región y junto a ellos, se realiza la definición teórica de aspectos relevantes para la investigación y se presenta el esquema metodológico, en donde se indica el procedimiento para la ejecución del proyecto en cuestión; seguidamente se da desarrollo a los resultados, la discusión de estos y las conclusiones de la investigación.

## 2 ANTECEDENTES

Al concebir el videojuego (VJ) como un software, podremos hablar de procesos planteados para su desarrollo. Ahmad, Barakji, Abou Shahada & Anabtawi (2017), recopilan algunos aspectos importantes para el lanzamiento exitoso de los VJ; entre ellos, un aspecto que analizan es el ciclo de vida del desarrollo de software (SDLC, por sus siglas en inglés). Estos autores proponen en su investigación un marco de trabajo referenciado en cascada (WATERFALL) teniendo en cuenta algunas de sus características: “El SDLC es un proceso que tiene como objetivo desarrollar y modificar software basado en secuencias de fases y actividades planificadas. Entre los modelos más conocidos del SDLC, el modelo WATERFALL es el más popular y antiguo” (Ahmad *et al.*, 2017, p. 3).

Ahmad *et al.* (2017), además de su propuesta, destacan dos *frameworks* generales para el desarrollo de VJ, conocidos como “Mechanics, Dynamics and Aesthetics” (MDA) y “Design, Play and Experience” (DPE). Los autores resaltan a MDA como un *framework* útil para el desarrollo de VJ de entretenimiento, pero no para el desarrollo de videojuegos serios (VJS), ya que no aborda aspectos para educar, recurrentes en los juegos serios. Por otra parte, DPE referencia la importancia del juego y sus experiencias, apuntándole a características o aspectos como la jugabilidad, aprendizaje y narración de historias.

Para Morales Urrutia, Nava López, Fernández Martínez y Rey Corral (2010), el método WATERFALL es el más implementado por la industria del desarrollo de VJ, aunque para ese entonces no referenciaron procesos específicos conocidos; no obstante, consideraron la posibilidad de la existencia de modelos, plantillas o procesos de carácter privado. Por esta razón, los autores plantearon un prototipo de proceso para el desarrollo de VJ, proceso llamado “Huddle” basado en SCRUM: “la filosofía es que, mediante breves reuniones de planeación a corto plazo, se planea cada ‘jugada’ que se inicie; con esto se da un seguimiento más estrecho al avance del proyecto y es posible hacer correcciones tempranas a posibles desviaciones” (Morales Urrutia, *et al.*, 2010, p. 28).



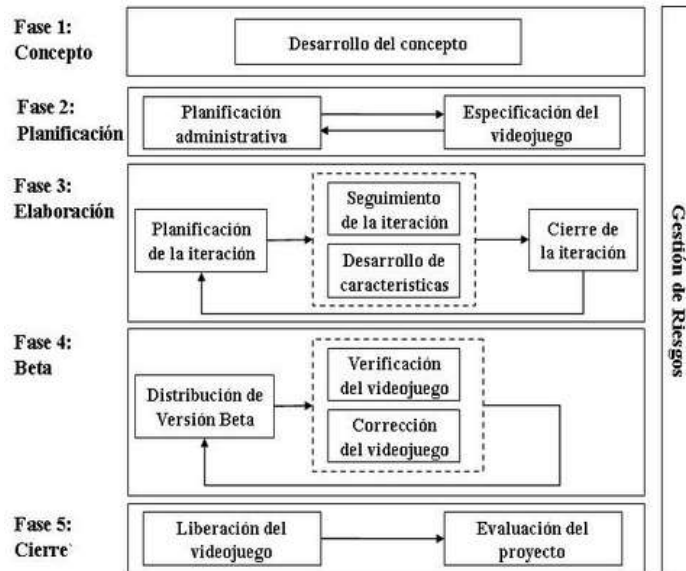
Figura 1. Fases del proceso Huddle



Fuente: (Morales Urrutia, *et al.*, 2010, p. 28)

Además del proceso descrito anteriormente, existe un proceso denominado SUM para el desarrollo de VJ, esta metodología adopta en gran medida la estructura y roles del marco de trabajo SCRUM. Tal como lo recopilan Acerenza, Coppes, Mesa y Viera en (2009) SUM, se divide en cinco fases iterativas e incrementales, y sus cinco fases a ejecutar son: concepto, planificación, elaboración, beta y cierre.

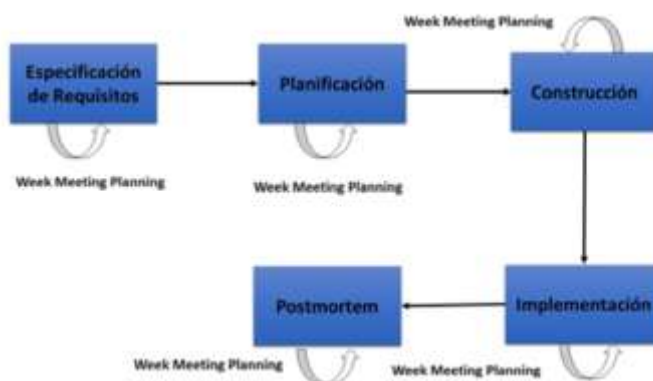
Figura 2. Flujo de trabajo de la Metodología SUM



Fuente: (SUM, 2009)

González Gill y Pérez (2016) plantean una propuesta metodológica para el desarrollo de VJ, híbrida entre CASCADA y SCRUM: “CASCRUM retoma una importante característica de las metodologías ágiles: incluir a los usuarios finales como parte del equipo de desarrollo. Esto ayuda en gran medida a lograr el éxito del proyecto, ya que son ellos quienes saben lo que quieren, por lo tanto, se debe tener el cuidado de propiciar un ambiente de confianza y colaboración con ellos” (González Gill y Pérez, 2016, p. 65).

Figura 3. Ciclo de vida de CASCRUM



Fuente: (González Gill y Pérez, 2016)

En referencia a las actividades de testing para videojuegos, se encuentran pocas reseñas. Por ejemplo, en los exámenes de certificación relacionados con testing, liderados por la International Software Testing Qualifications Board (ISTQB), se cuenta con una especialidad para pruebas de juegos, Foundation Level Gambling Industry Tester (2019), aunque esta está relacionada con la industria de juegos de azar o de apuestas; no se encuentran certificaciones para pruebas de VJ y menos aún de VJS.

Además, un referente importante en tests de VJ es el libro *Game Testing: All in one* de Schultz y Bryant (2011) el cual presenta una serie de fases para las pruebas de VJ, que son: pre-producción, pruebas alfa, pruebas beta, pruebas de oro y post-lanzamiento. Estas mismas fases son retomadas en el artículo “An Overview of Game Testing Techniques”, de Redavid y Farid (2011) adicionando nuevas fases como Kickoff y Release. Algunos

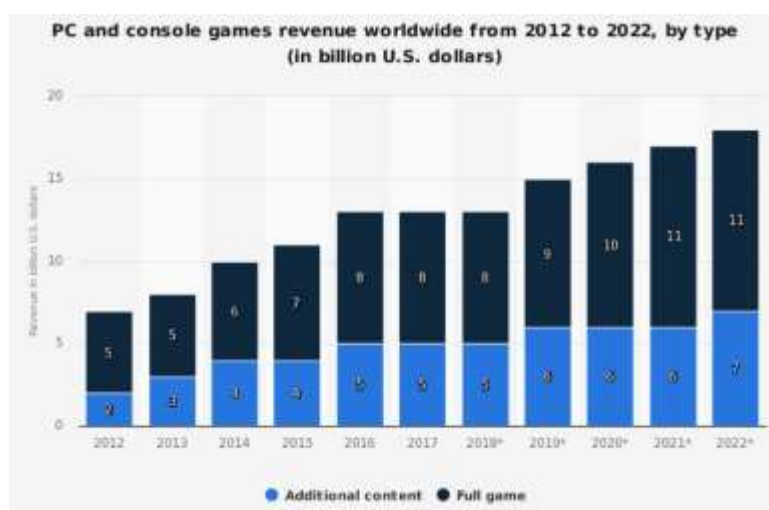
elementos de estas fases serán considerados para el diseño del proceso, pero deben ser revisadas y adaptadas para el área específica de los VJS, tema que no es considerado en las propuestas.

Se evidencian, a partir de lo anterior, algunos *frameworks*, procesos o metodologías para el ciclo completo del desarrollo de VJ o para pruebas de videojuegos, pero no se identificaron procesos específicos estandarizados de pruebas de los VJ serios. Actualmente, las pruebas de este tipo de VJ se realizan en un procedimiento en el cual “se designa un equipo de personas el cual su trabajo es encontrar errores, inconsistencias o funcionalidades incompletas a lo cual son reportadas al equipo de mantenimiento. Este equipo recibe el nombre de *Beta Testers*” (Perchy y Nishi, 2013, p. 9), no obstante, lo anterior no responde a una metodología o un proceso de ingeniería específico.

### 3 ÁREA PROBLEMÁTICA Y PREGUNTA DE INVESTIGACIÓN

La industria de los VJ en el marco económico ha tenido una tendencia de crecimiento; el valor global del mercado para 2020 se pronosticaba cerca a los 90 billones de dólares según lo reporta la agencia Statista (2018b). Así mismo para el 2017 se registró un total de 13 billones de dólares en ingresos en los juegos de PC y consolas, y se pronosticaba para 2020, 16 billones de dólares en ingresos:

Figura 4. Ingresos de Juegos de PC y Consolas



Fuente: (Statista, 2018a)

Por lo anterior, es de vital importancia tener claras las implicaciones de la calidad del VJ, dado que también tendrá un impacto en cuanto a lo económico. Como lo mencionan Ahmad *et al.* (2017), hay ejemplos en donde los VJ son golpeados en el mercado, un ejemplo de lo anterior fue el lanzamiento del VJ Assanssin's Cread Unity, ya que fue fuertemente criticado por la gran cantidad de errores y fallas.

Como lo recopilan Lin, Bezemer y Hassan (2019) cerca del 80 % de los VJ *steam*, necesitan lanzar actualizaciones o paquetes de urgencia para solucionar problemas, generando más esfuerzo por malos funcionamientos o errores en los juegos; lo anterior a pesar de los grandes esfuerzos que se dedican en el proceso de desarrollo del VJ.

Existen diferentes factores que contribuyen al aseguramiento de la calidad de un producto; en la calidad del software, que es lo que nos compete, se observan diferentes factores:

La calidad total depende de la calidad con la cual se lleve a cabo todo el proceso - las cinco etapas del ciclo de vida de un desarrollo de software (análisis, diseño, implementación, pruebas e implantación)- y cada subproceso o fase o etapa del proyecto... (López Echeverry, Cabrera y Valencia Ayala, 2008, p. 328)

Uno de los factores son las pruebas, que son una parte importante del proceso de aseguramiento de la calidad del software; las pruebas según lo recopilado por Hernández Bécares, Costero Valero & Gómez Martín (2016), son un proceso de análisis de elementos de software para hallar e interpretar las diferencias entre las condiciones actuales y las esperadas de algún elemento. Además, se indica sobre las diferentes pruebas o aspectos que se pueden revisar o verificar dependiendo del enfoque o prueba que se realice.

Para Redavid y Farid (2011) las pruebas cumplen un rol importante en los juegos, por las diferencias que hay entre las pruebas de un juego y las pruebas de un software; en el proceso de prueba de juegos se inmiscuyen muchos pasos, sumado a esto los diferentes casos de prueba que por lo general son caja negra. Por lo anterior toma relevancia las particularidades y las especificidades de un proceso para probar VJ. También es importante mencionar sobre pruebas de VJS por sus particularidades. Por ejemplo: Moreno, Torrente, Hsieh & Lester (2012), mencionan que el fin de los VJS es involucrar a los usuarios, a los jugadores en actividades y procesos de aprendizaje, por lo tanto, es importante evaluar la efectividad del aprendizaje y lo idóneo de su contenido para el público del VJS.

Capers Jones de Namcook Analytics LLC en 2013 en la Conferencia de Calidad del Software Noroeste del Pacífico (PNSQC, por sus siglas en inglés); menciona algunos factores de la calidad en el marco del éxito de los proyectos de software y relaciona algunas prácticas o actividades de test y de calidad con respecto al éxito del proyecto, como se verá a continuación:

Tabla 1. Prácticas y actividades según tasa de éxito en proyectos de software

	<b>Baja calidad: éxito &lt; 25%</b>	<b>Regular calidad: éxito &lt; 50%</b>	<b>Buena calidad: éxito &gt; 90%</b>
<b>Prácticas, actividades</b>	<ul style="list-style-type: none"> <li>- <i>Testing</i> como único método para detectar y eliminar defectos.</li> <li>- <i>Testing</i> informal con personal no certificado.</li> <li>- <i>Testing</i> realizado por desarrolladores.</li> <li>- Aseguramiento de la calidad pasivo (&lt; 3\$ equipo de SQA).</li> <li>- Métricas basadas en SLOC para la calidad (omiten errores diferentes a código)</li> </ul>	<ul style="list-style-type: none"> <li>- Programación por pares.</li> <li>- Nivel 3 en CMMI.</li> <li>- Métodos de sala limpia para cambios rápidos en requerimientos.</li> <li>- Verificación y validación independientes.</li> <li>- Estándares de calidad ISO e IEEE.</li> </ul>	<ul style="list-style-type: none"> <li>- Inspecciones formales (código, diseño, requerimientos).</li> <li>- Análisis estáticos.</li> <li>- Mediciones en eficiencia en detección y eliminación de defectos.</li> <li>- Herramientas para seguimiento automático de defectos.</li> <li>- Aseguramiento de calidad activo (&gt;3 equipo de SQA).</li> <li>- Métodos efectivos (Agile, XP, TSP)</li> <li>- Probadores certificados.</li> <li>- Diseño de casos de prueba basado en el diseño experimental.</li> </ul>

Fuente: adaptada de Jones (2013)

Como se observa, la calidad del software no depende solo de las pruebas, sino también de otros aspectos como los métodos usados (Ágiles, XP, TSP), inspecciones y análisis estático, entre otros. Algunos de estos elementos pueden integrarse en un proceso de pruebas, como el que se propone en este proyecto.

Con respecto a los métodos, como lo plantearon en Navarro Cadavid, Fernández Martínez y Morales Vélez (2013), se contemplan para el desarrollo de software

metodologías ágiles que han surgido buscando minimizar la posibilidad de fracaso en la estimación de tiempos, costos o funcionalidades en los proyectos; estas metodologías ágiles surgen como reacción a la burocracia inmersa en las metodologías tradicionales. A continuación, algunos aspectos que son comparados según los autores:

Tabla 2. Metodologías tradicionales vs Metodologías Ágiles

Metodologías tradicionales	Metodologías ágiles
Predictivos	Adaptativos
Orientados a procesos	Orientados a personas
Proceso rígido	Proceso flexible
Se concibe como un proyecto	Un proyecto es subdividido en varios proyectos más pequeños
Poca comunicación con el cliente	Comunicación constante con el cliente
Entrega de software al finalizar el desarrollo	Entregas constantes de software
Documentación extensa	Poca documentación

Fuente: Navarro, et al, (2013)

Las metodologías o procesos de desarrollo de software ágil se han planteado “de una manera que proclama ser más rápida que los métodos tradicionales de desarrollo de software, el desarrollo ágil se presenta como una alternativa para generar software de calidad” (Zapata, Arboleda y Castrillón, 2010, p. 17) a su vez, estos autores indican que el involucrar al testing en el ciclo de vida de las metodologías ágiles ha permitido vislumbrar algunas limitantes, problemas de comunicación entre el equipo y falta de automatización de las pruebas.

Caspers Jones en la PNSQC en 2013 presenta la relación entre los métodos de desarrollo de software y el número de defectos entregados (por puntos de función), defectos potenciales y porcentaje en la eficiencia de eliminación de defectos; esto se verá reflejado en la tabla 3:

Tabla 3. Relación métodos de desarrollo y estadísticas de defectos

MÉTODOS	Defectos potenciales	Eficiencia en eliminación	Defectos entregados
Cascada	7.00	75%	1.75
Iterativo	6.25	82%	1.13
OO	5.75	85%	0.86
RUP	5.50	90%	0.55
<b>Agile</b>	<b>5.50</b>	<b>87%</b>	<b>0.72</b>
PSP y TSP	5.00	94%	0.30
MDD	4.00	96%	0.15
85% re-uso certificado	2.25	96%	0.09

Fuente: Jones (2013)

La ISTQB en Agile Tester Extension in a Nutshell (2014) plantea que las pruebas ágiles siguen los principios del manifiesto ágil, siendo este un enfoque nuevo en las pruebas de software. En los documentos de certificación de la ISTQB (2014), específicamente en *Foundation Level Extension Syllabus Agile Tester*, se plantea que los modelos ágiles difieren en el cómo se desarrollan las actividades de prueba y de desarrollo, también del cómo pueden diferir los criterios de los diferentes niveles de prueba y se indica la capacidad de adaptabilidad según el contexto del proyecto.

### 3.1 PREGUNTA DE INVESTIGACIÓN

Por todo lo expuesto anteriormente en la revisión realizada de los *frameworks*, metodologías y de los procesos para el desarrollo de VJS y del poco contexto evidenciado de procesos específicos de test ágil para este tipo de software, se realiza la siguiente pregunta:

¿Cómo verificar la usabilidad y calidad de un videojuego serio, con la implementación de un proceso de test ágil específico?



## 4 JUSTIFICACIÓN

Con lo recopilado se evidencia la existencia de diferentes procesos o marcos de trabajo para el desarrollo de software o en específico para el desarrollo de videojuegos, sin embargo, no se encuentran procesos de test específicos para VJS en un marco de desarrollo ágil, pudiendo generarse con lo anterior una brecha en la calidad. Además, se observa el potencial económico de la industria de los VJ, junto con criterios de éxito de los proyectos, pero que se pueden ver afectados por las actividades o procesos de test o de calidad inmersas en los proyectos de desarrollo de software.

A lo anterior se suma lo indagado al consultar con el sector de la industria de videojuegos de la región, en donde el 42.9% de los consultados indican que en su empresa no cuentan con proceso de prueba específico para videojuegos, pero por lo contrario un 71.4% sí cuentan con proceso o metodología para el desarrollo; además 57.1% no usa pruebas para validar la pedagogía en el videojuego serio educativo.

Aunque la calidad del software no solo se determina por los procesos de test, sino también por el conjunto del proceso de desarrollo; con lo recopilado se puede observar un potencial mayor de éxito y de calidad en los proyectos de software que dependen de los procesos de test inmersos en su desarrollo; sin embargo, lo anterior es una desventaja para el desarrollo de VJS, al no contar con procesos específicos de test (diferentes a las pruebas beta) ampliamente reconocidos para este tipo de software. Por lo dicho, es que toma relevancia esta propuesta investigativa, es decir, debido a la brecha que se podría saldar por la ausencia de procesos de test que impactarían la calidad de este tipo de software.

También se aprecia la relación entre los métodos de desarrollo de software con aspectos de calidad (cantidad de defectos, eficiencia en eliminación de defectos) en los proyectos en los cuales se implementaron dichos métodos. Se evidencia con esto y las recopilaciones realizadas, que para el desarrollo de VJ tiene más relevancia en la industria la implementación de metodologías tipo cascadas, pese a los aspectos de calidad ya mencionados, en los que se constata mejor calidad que en otras metodologías, en específico

en las ágiles. Por lo tanto, sabiendo que el test ágil cumple y que se rige por los principios del manifiesto ágil y que estos test ágiles se pueden adaptar a diferentes tipos de proyectos, es por lo cual que se propone en la investigación un proceso de test para VJS en base con los principios ágiles, es decir, un proceso de test ágil para VJS.

## 5 MARCO TEÓRICO

Los videojuegos, según de Kerckhove (1999) “(...) son un nuevo medio interactivo tan diferente como lo fue en su día la televisión para con la radio. Representan nuevas formas de comunicación y abarcan un gran potencial de entretenimiento y educativo (p. 52)”. La relación de los videojuegos y las Tecnologías de la Información y Comunicación (TIC) se puede ver planteada a continuación ya que según Pernet (2012) los VJ actualmente tienen componentes de comunicación y un claro ejemplo de la implementación de las TIC son los juegos de rol y aquellos videojuegos que hacen uso de internet, es decir, los VJ online.

A través de la relación de los VJ y las TIC se podría hablar de los videojuegos en cuanto al software, tal como lo recopilan Pérez García y Ortega Carrillo (2011) en una de sus investigaciones; estos autores mencionan que un videojuego se puede entender como un software para videoconsola o PC, así como para diferentes plataformas emergentes, teléfonos móviles y sitios web.

Marcano (2008) habla sobre los beneficios que proporcionan los VJ y su aprovechamiento para los procesos educativos, de entrenamiento y de información. Al conjunto de estos juegos se les ha denominado video juegos serios, ya que son usados para capacitar en el área de la salud, emergencias, educación, industria comercial e instituciones de desarrollo científico. Además, otra de las características de los juegos serios, según Kosa, Souza, Veado & Figueiredo (2016), es que el aprendizaje por medio de los juegos serios mejora la experiencia del alumno y, a su vez, el nivel de aprendizaje.

Para la ISTQB el Ciclo de Vida del Desarrollo de Software (SDLC, por sus siglas en inglés), es un derrotero de actividades realizadas en cada fase, junto con la relación lógica y cronológica entre estas etapas. Al hablar de SDLC es importante definir el concepto de metodologías del desarrollo de software: tal como lo describe Awad (2005), las metodologías del desarrollo de software son la imposición de un proceso ordenado y disciplinado con el fin de hacer predecible y eficiente el desarrollo del software; además

estas metodologías se encuentran segmentadas entre las tradicionales y las ágiles. En el caso de la investigación es importante recalcar la definición del proceso ágil de desarrollo, ISTQB (2011) lo define como un conjunto de metodologías de desarrollo que se basan en el desarrollo incremental iterativo en donde por medio de la autoorganización y las multifuncionalidades, los equipos colaborativamente evolucionan los requisitos y las soluciones de los mismos.

Según la ISTQB (2018) el proceso de test es el conjunto de actividades fundamentales relacionadas entre sí, para el análisis, planificación, diseño, control, monitoreo y ejecución de las pruebas. Así mismo, la ISTQB (2011) define el test ágil, como una práctica de prueba en marco de proyectos desarrollados con metodologías ágiles, en donde el desarrollo es el cliente de las pruebas, además se enfoca en el diseño de las pruebas primero. A lo anterior se suma lo planteado por la IEEE en el estándar 1059 (1993) en donde expresa que el proceso de test, es un proceso que analiza las condiciones existentes contra las condiciones requeridas, evaluando así las características del software.

Según Mascheroni et. al (2012) la usabilidad es un atributo intangible del software, su visualización y medición es compleja y la atención por este aspecto contribuye al incremento de la calidad del software percibido por el usuario. Nielsen & Molich (1990) enfatizan en la variabilidad de métodos existentes para evaluar la usabilidad, entre ellos la evaluación heurística, esta se realiza por medio de observación y a través de opiniones sobre aspectos de las interfaces; los usuarios realizan las evaluaciones según lo estipulado en ciertas reglas. En este caso la capacidad educativa se propone también sea medible por medio de heurísticas como las planteadas por Pinelle, Wong & Stach (2008) y por Hsieh & Lester (2012).

## **6 OBJETIVOS**

### **6.1 OBJETIVO GENERAL**

Diseñar un proceso de test ágil específico para video juegos serios (VJS), que verifique y valide la usabilidad y la calidad del videojuego.

### **6.2 ESPECÍFICOS**

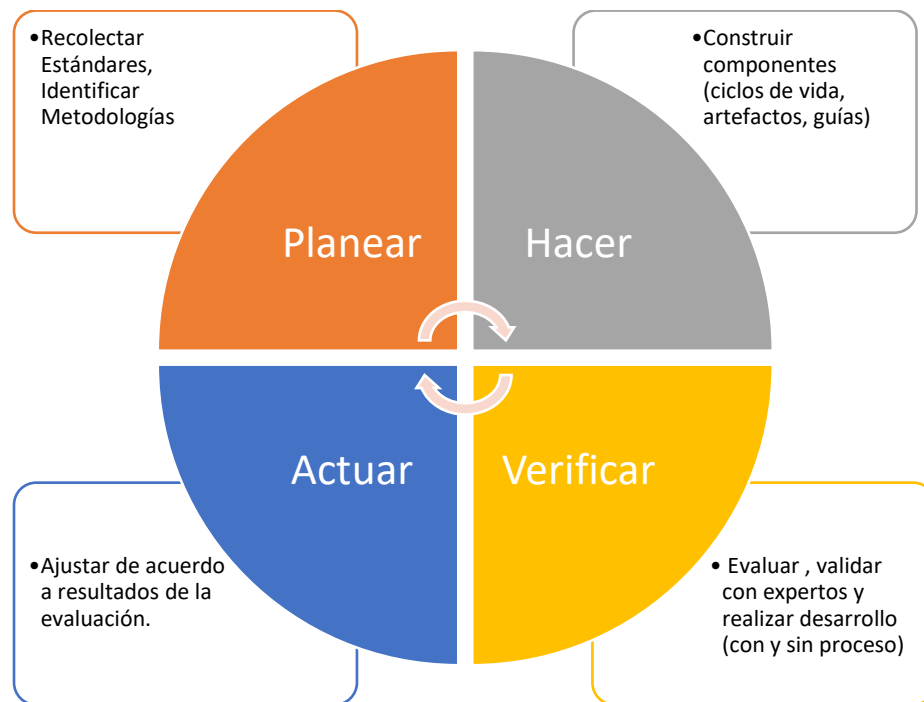
- 1- Definir los elementos del proceso de test ágil para videojuegos, a partir de referentes bibliográficos y de experiencias de la industria local.
- 2- Modelar los elementos del proceso, haciendo uso del lenguaje SPEM 2.0
- 3- Evaluar con expertos en el área de Ingeniería de software y Calidad de software, el proceso de test ágil diseñado.
- 4- Validar el proceso con la implementación piloto en el desarrollo de un nivel de un videojuego serio, y compararlo con el desarrollo de un nivel de un videojuego serio que no use el proceso.

## 7 METODOLOGÍA

### 7.1 DISEÑO DE PROCESO DE TEST ÁGIL

Para el diseño del proceso de test ágil de desarrollo de videojuegos se implementará la metodología IMPACT (2001), la cual se ocupa de realizar mejoras en los procesos de software, esta lleva consigo ciclos iterativos con presencia de cuatro fases básicas (planear, hacer, verificar y actuar).

Figura 5. Ciclo de vida de PDCA (PHVA)



Fuente: Adaptada de Moen & Norman (2006)

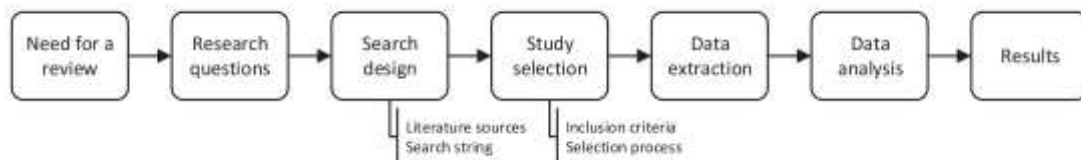
#### 7.1.1 Planear

Para el cumplimiento del primer objetivo específico, en la fase de planeación se realizará el proceso de recolección y sistematización de datos e información, mediante una revisión de literatura y encuestas a la industria local de VJ. Lo anterior, incluye los siguientes aspectos y temas de interés:

- Características específicas de los VJS y VJ.
- Metodologías, procesos e instrumentos de test existentes para VJ.

Para el mapeo sistemático se aplicarán los siguientes pasos:

Figura 6. Pasos de Mapeo Sistemático



Fuente: (Franco, Ameller, Costal & Franch, 2017)

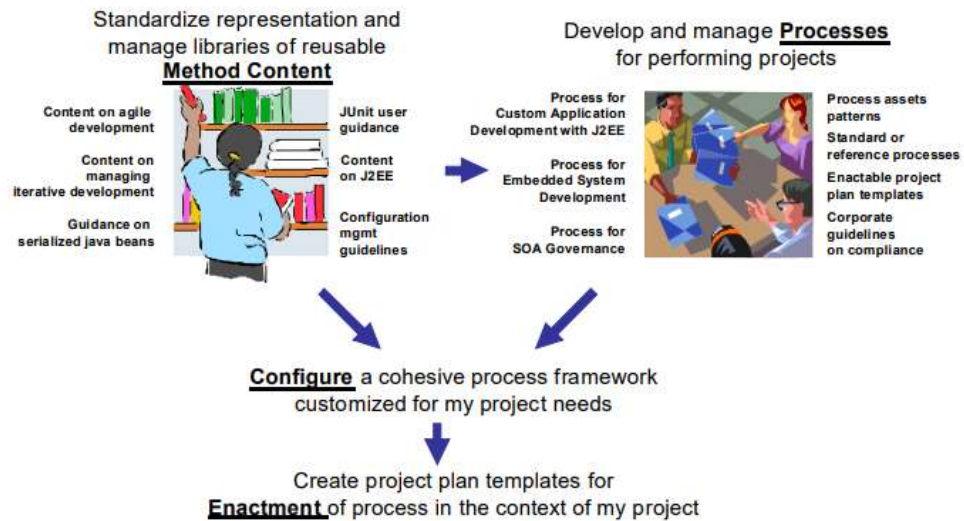
En cuanto a la encuesta, tal como lo mencionan Torres, Paz y Salazar (2019) “es un método descriptivo con el que se pueden detectar ideas, necesidades, preferencias, hábitos de uso, etc.” (p. 4). Según lo anterior, se realizará una encuesta para identificar metodologías y *frameworks* de test ágil de videojuegos usados en la industria del desarrollo de VJ en el ámbito local.

### 7.1.2 Hacer

Diseñar y modelar los componentes (ciclos de vida, artefactos, procedimientos, estructura) del proceso test ágil para VJS, a partir de la revisión sistemática y encuestas realizadas en la fase de *planear* del ciclo PHVA; de tal forma se cumplirá con el segundo objetivo específico. Para ello se hará uso de Software & System Process Engineering Metamodel versión 2.0 (SPEM 2.0). Según Object Management Group (OMG) (2008) SPEM 2.0 es utilizado para definir procesos de desarrollo de software, sistemas y sus componentes.

SPEM 2.0 es un metamodelo de ingeniería de procesos, a su vez es un marco conceptual que proporciona conceptos para modelar, documentar, presentar, gestionar, intercambiar y promulgar métodos y procesos de desarrollo (OMG, 2008).

Figura 7. Marco conceptual de la implementación de SPEM 2.0



Fuente: (OMG, 2008)

Éste es recomendado para implementaciones que se focalizan en administrar la documentación descriptiva de los métodos de desarrollo, técnica y mejores prácticas. Los ejecutores de esta compilación generalmente no requieren modelos formales, ya que proporcionan descripciones de sus métodos con conceptos básicos, como producto del trabajo, tareas, roles y definiciones; representando una visión ágil para comunicar el conocimiento del desarrollo (OMG, 2008).

### 7.1.3 Verificar

Se diseñará una encuesta para evaluar el proceso de test ágil para el desarrollo de VJS, la cual será aplicada a expertos. Esta encuesta permitirá valorar la claridad, completitud, coherencia y pertinencia del proceso. Cumpliendo con esto, el tercer objetivo específico planteado.

Para validar el proceso de test ágil para VJS y cumplir el cuarto objetivo específico, se desarrollarán dos niveles de un VJS, en el desarrollo de un nivel el equipo encargado implementará el proceso y el otro equipo no lo implementará. Y para el desarrollo de ambos niveles se usará el marco de gestión de trabajo SCRUM; buscando que ambos



equipos lleven a cabo las actividades de ingeniería, implementando una misma metodología de desarrollo.

Para validar la usabilidad y calidad del VJS, se realizarán las siguientes mediciones:

#### 7.1.3.1 Usabilidad.

Nielsen & Molich (1990) enfatizan en la variabilidad de métodos existentes para evaluar la usabilidad, entre ellos la evaluación heurística, esta se realiza por medio de observación y a través de opiniones sobre aspectos de las interfaces; los usuarios realizan las evaluaciones según lo estipulado en ciertas reglas. Para la validación de usabilidad del VJS se aplicarán algunas de las heurísticas seleccionadas de la investigación: “*Heuristic evaluation for games: usability principles for video game design*” de Pinelle, Wong & Stach (2008) y también en el artículo: “*Usability Testing for Serious Games: Making Informed Design Decisions with User*” de Moreno, Torrente, Hsieh & Lester (2012).

#### 7.1.3.2 Calidad.

Como se presentó en capítulos anteriores, para mejorar la calidad del producto (en este caso del video juego serio), se deben combinar diferentes estrategias. De este modo, en este caso se realizará la medición de las siguientes métricas:

- **Mediciones de Análisis de código estático:** para la ISTQB (2012), es el análisis que se realiza sin ejecutar el software sobre el código fuente que se desarrolló. Para el caso de la investigación se aplicará alguna de las herramientas libres conocidas para obtener los valores de estas mediciones.
- **Cobertura de código:** según la ISTQB (2011), la cobertura de código es un método de análisis que determina qué partes del software han sido ejecutadas por el conjunto de pruebas y qué partes no se han ejecutado. Se realizará un análisis en el cual se detallará el porcentaje de código probado y ejecutado
- **Nivel de Cobertura de pruebas:** para Lee, Kang & Jung (2020), el nivel de cobertura de pruebas es el grado de elementos de cobertura (requisitos) ejecutados

por un conjunto de pruebas. En este nivel se analizará la cantidad de pruebas ejecutadas en función de la cantidad de requisitos.

- **Resultados de ejecución de las pruebas:** Se analizarán los resultados de la cantidad de pruebas ejecutadas y la cantidad de defectos identificados en el desarrollo y en el proceso de test del VJS.

#### **7.1.4 Actuar**

Se definirán los cambios a realizarse de acuerdo con los resultados obtenidos en la fase de verificación del ciclo PHVA, estos cambios tendrán efecto para una versión posterior del proceso diseñado, lo cual no será del alcance de esta investigación.

## 8 RESULTADOS

Siguiendo la metodología planteada para el mapeo sistemático y para el proceso de selección de estudios, a continuación, se describen las actividades realizadas.

### 8.1 RESULTADOS DEL MAPEO SISTEMÁTICO

#### 8.1.1 Necesidades Para Revisar

De acuerdo con el contexto de la investigación y de la revisión literaria que le concierne, se determina a continuación, cuáles son las necesidades de conceptos importantes para el proceso de la revisión de artículos:

Tabla 4. Necesidades para revisar

Código	Necesidades
NRA01	Hallar la existencia de diferentes procesos, metodologías, marcos de trabajo para el desarrollo de VJS.
NRA02	Identificar los diferentes procesos de prueba o pruebas realizadas en las metodologías, <i>framework</i> o procesos de desarrollo de VJS.
NRA03	Identificar herramientas, guías y artefactos usados en el proceso de prueba de VJS.
NRA04	Identificar métricas de calidad, usabilidad y de pedagogía para medir en el VJS.

Fuente: Elaboración propia

#### 8.1.2 Preguntas De Investigación

Posterior a la definición de las necesidades de investigación, se definieron preguntas con el fin de satisfacer sus respuestas con la revisión sistemática.

Tabla 5. Preguntas de investigación

Necesidad	Código	Preguntas
NRA01	RQ1	¿Qué procesos, metodologías, ciclos de vida o <i>framework</i> de trabajo se usan para el desarrollo de VJS?
NRA02	RQ2	¿Qué pruebas o procesos de prueba se usan en el desarrollo de VJS?
NRA03	RQ3	¿Qué artefactos, herramientas o guías se involucran en el proceso de prueba de VJS?
NRA04	RQ4	¿Cómo se mide la calidad, usabilidad y de pedagogía en el VJS?

Fuente: Elaboración propia

### 8.1.3 Diseño De Búsqueda

#### 8.1.3.1 Fuentes de literatura

Para la revisión bibliográfica, se segmentó la búsqueda en las bases de datos proporcionadas por las librerías digitales Science Direc y, Dialnet; principalmente libros y artículos de las diversas publicaciones que se agrupan allí.

### 8.1.4 Cadena De Búsqueda

Con el fin de dar respuesta a cada una de las preguntas de investigación se plantearon las cadenas búsqueda, permitiendo abarcar una gran cantidad de población necesaria de fuentes, pero a la vez segmentando de acuerdo con las necesidades de búsqueda.

Tabla 6. Cadenas de búsqueda

Código	Cadena
SST1	"Serious Game" AND (Testing OR Test)
SST2	Videogames AND (Testing OR Test)
SST3	“Proceso de prueba de videojuegos”
SST4	“Proceso de pruebas de videojuegos serios”
SST5	“Metodología de desarrollo de videojuegos”
SST6	“Metodología de desarrollo de videojuegos serios”

Fuente: Elaboración propia

Las cadenas de búsqueda se aplicaron de acuerdo con los estándares y requerimientos de los diferentes buscadores científicos ya nombrados en la sección anterior.

### 8.1.5 Selección De Estudios

#### 8.1.5.1 Criterios de inclusión.

Para la selección de estudios, en su búsqueda se aplicó un primer filtro, este consta de segmentación por tipo de documento y por rango de fecha de publicación. Estos criterios de inclusión se aplicaron en el resultado de las búsquedas realizadas con las cadenas de la *Tabla 6*.

Tabla 7. Filtros de búsqueda

Tipo filtro	Filtro
Rango de fechas	2010-2022
Tipo de documento	<ul style="list-style-type: none"> <li>- Review articles</li> <li>- Research articles</li> <li>- Book chapters</li> <li>- Conference abstracts</li> </ul>

Fuente: Elaboración propia

### 8.1.5.2 Proceso de selección

En la figura se hace una relación de la cantidad de documentos investigativos encontrados de acuerdo con cada paso de la fase de selección de estudios.

Figura 8 Resultado selección de estudios



Fuente: Elaboración propia

#### **Paso 1 – Búsqueda automática**

De acuerdo con las cadenas de búsqueda *Tabla 6* y de los criterios de inclusión *Tabla 7*, en este paso se logró identificar un total de 411 artículos de base para su revisión.

#### **Paso 2 – Remoción de duplicados**

En la búsqueda automática de las cadenas se encontraron repetidos algunos documentos, posterior a su remoción, se obtienen un total de 356 documentos para su lectura.

### **Paso 3 – Títulos y resúmenes**

Después de la revisión de los títulos y de los resúmenes de los documentos depurados en el paso 2, se obtuvo un total de 96 documentos investigativos de interés para el estudio, estos artículos pasaron a un proceso de una revisión más profunda.

### **Paso 4 – Lectura rápida**

La lectura rápida de los 96 documentos del paso anterior, dejaron un total de 15 artículos de interés; estos fueron la base de la extracción de información para el análisis y posterior desarrollo del proyecto.

### **Paso 5 – Estudios secundarios**

De los artículos revisados en el paso anterior, se obtuvieron 3 estudios secundarios al objeto de necesidades de estudio, por lo tanto, fueron incluidos para el proceso de revisión, pasando de 15 a 18 documentos.

### **Paso 6 – Búsqueda manual**

Se complementó la búsqueda, realizando indagaciones manuales en el buscador académico de Google Scholar con el fin de ubicar documentos de interés que hubiesen sido ignorados por los buscadores de los repositorios consultados de forma automática. En este paso se hallaron 5 nuevos artículos de interés para la revisión; lo anterior arrojó un total de 23 artículos investigativos seleccionados para realizar la extracción de información (ver tabla).

Tabla 8 Documentos seleccionados

<b>Código</b>	<b>Título documento</b>	<b>Autores</b>
DOC01	An approach to automated videogame beta testing	Jennifer Hernández Bécares, Luis Costero Valero, Pedro Pablo Gómez (2017)
DOC02	Using a videogame to facilitate nursing and medical students' first visit to the operating theatre. A randomized controlled trial	Ángel del Blanco, Javier Torrente, Baltasar Fernández Manjón, Pedro Ruiz, Manuel Giner (2017)
DOC03	Serious Game for Teaching Undergraduate Medical Students in Cleft lip and Palate Treatment Protocol	Patison Palee, Noppon Wongta, Krit Khwanngern, Waritsara Jitmun, Noppon Choosri (2020)
DOC04	An approach to evaluating the user experience of serious games	Jonathan Moizer, Jonathan Lean, Elena Dell'Aquila, Paul Walsh, Alphonsus (Alfie) Keary, Deirdre O'Byrne, Andrea Di Ferdinando, Orazio Miglino, Ralf Friedrich, Roberta Asperges, Luigia Simona Sica (2019)
DOC05	Designing a BIM-based serious game for fire safety evacuation simulations	Uwe Rüppel, Kristian Schatz (2011)
DOC06	Herbopolis – A mobile serious game to educate players on herbal medicines	Ryan Wei Xuan Ee, Kai Zhen Yap, Kevin Yi-Lwern Yap (2018)



DOC07	Deploying learning materials to game content for serious education game development: A case study	Harit Ar Rosyid, Matt Palmerlee, Ke Chen (2018)
DOC08	User-centered design with visually impaired pupils: A case study of a game editor for orientation and mobility training	Elke Mattheiss, Georg Regal, David Sellistsch, Manfred Tscheligi (2017)
DOC9	Aplicación de analíticas en la sistematización del diseño y validación de juegos serios para usuarios con discapacidad intelectual	Ana Rus Cano (2019)
DOC10	Mejorando la evaluación de juegos serios mediante el uso de analíticas de aprendizaje	Angel Serrano Laguna (2017)
DOC11	Micromundo etnoeducativo como apoyo al docente en las prácticas de enseñanza de la lengua nam trik de Totoró “namoi po jaumai amkun	Edinson Ancizar Castillo, Erwin Meza Vega, Sara Garcés, Wilmer Camacho (2016)
DOC12	GeoPGD metodología para la implementación de juegos pervasivos georreferenciados apoyados en Linked Open Data	Jeferson Arango López (2019)

DOC13	Metodología de desarrollo para videojuegos educativos basada en notaciones gráficas	Rafael A. Prieto de Lope (2018)
DOC14	Metodologías de desarrollo en videojuegos	Manuel Ridao Pineda (2018)
DOC15	Ciclo de vida Juego educativo móvil "Educamovil"	Judith del Pilar Rodríguez, Oscar Alberto Gallardo, Ángela Roció Rivera, Oliver Smith Cadena
DOC16	Educational Games (EG) Design Framework: Combination of Game Design, Pedagogy and Content Modeling	Roslina Ibrahim, Azizah Jaafar (2009)
DOC17	The game production handbook. Jones & Bartlett Publishers	Heatber Maxwell Chandler (2009)
DOC18	Game development life cycle guidelines	Rido Ramadan, Yani Widayani (2013)
DOC19	Proceso de pruebas en el desarrollo de videojuegos	Yeili Ibarra Monteagudo (2018)
DOC20	How to launch a successful video game: A framework	Norita B. Ahmad, Salahudin Abdul Rahman Barakji, Tarak Mohamed Abou Shahada, Zeid Ayman Anabtawi (2017)
DOC21	Procesos de desarrollo para videojuegos	Gerardo Abraham Moralrez Urrutia, Claudia Esther Nava López, Luis Felipe Fernández

		Martínez, Mirsha Aarón Rey Corral (2010)
DOC22	Una metodología Ágil para Desarrollo de Videojuego	Nicolas Acerenza, Ariel Coppes, Gustavo Mesa, Alejandro Viera (2009)
DOC23	Una propuesta metodológica para la construcción de videojuegos	José Ángel González Gill, Efraín Pérez (2016)

Fuente: Elaboración propia

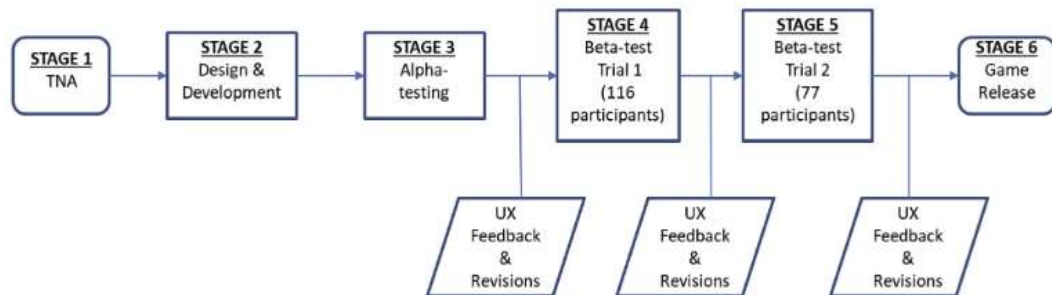
### 8.1.6 Extracción De Datos

Para dar respuesta a las preguntas investigativas planteadas en la *Tabla 5* a continuación, se plantea el desarrollo de la extracción de datos, allí se observan los diferentes ciclos de vida, metodologías procesos para el desarrollo de VJS, también datos sobre procesos y tipos de prueba aplicados, artefactos usados y criterios de medición de usabilidad y pedagogía en el VJ.

#### 8.1.6.1 Proceso de evaluación UX CUBE

Moizer et. al (2019) propone un proceso de evaluación de VJ, esta evaluación permite a los desarrolladores obtener retroalimentaciones de Experiencia de Usuario (UX) por sus siglas en inglés, en dos momentos por parte de los usuarios, estos momentos son llamados pruebas Alfa y pruebas Beta.

Figura 9 Proceso de evaluación UX-Cube



Fuente: Moizer et. al (2019)

Además del ciclo de vida de evaluación, los autores involucran en su evaluación artefactos de heurísticas, permitiendo la valoración de usabilidad, experiencia de usuario y experiencia de aprendizaje. Entre este artefacto, se reconocen dimensiones que cuentan con características y su vez estas características se miden con elementos de encuestas.

#### 8.1.6.2 *Proceso de desarrollo basado en Building Information Modeling (BIM)*

Rüppel y Schatz (2011) desarrollaron un videojuego serio para el rescate de humanos basado en BIM, en este caso los autores por medio de pasos y de forma iterativa, llevaron a cabo el proceso de desarrollo, en este caso: Planeación, requerimientos, diseño + análisis, implementación, pruebas, evaluación y SHRG; en este caso SHRG es el videojuego desarrollado por los autores, correspondería al producto final del proceso descritos por ellos.

#### 8.1.6.3 *Enfoque para la automatización de pruebas beta*

Bécares, Costero y Gómez (2017) en su investigación recopilan tipos de prueba de utilidad para el proceso de automatización de VJ, las cuales se presentan a continuación:

Tabla 9 Tipos de pruebas identificadas

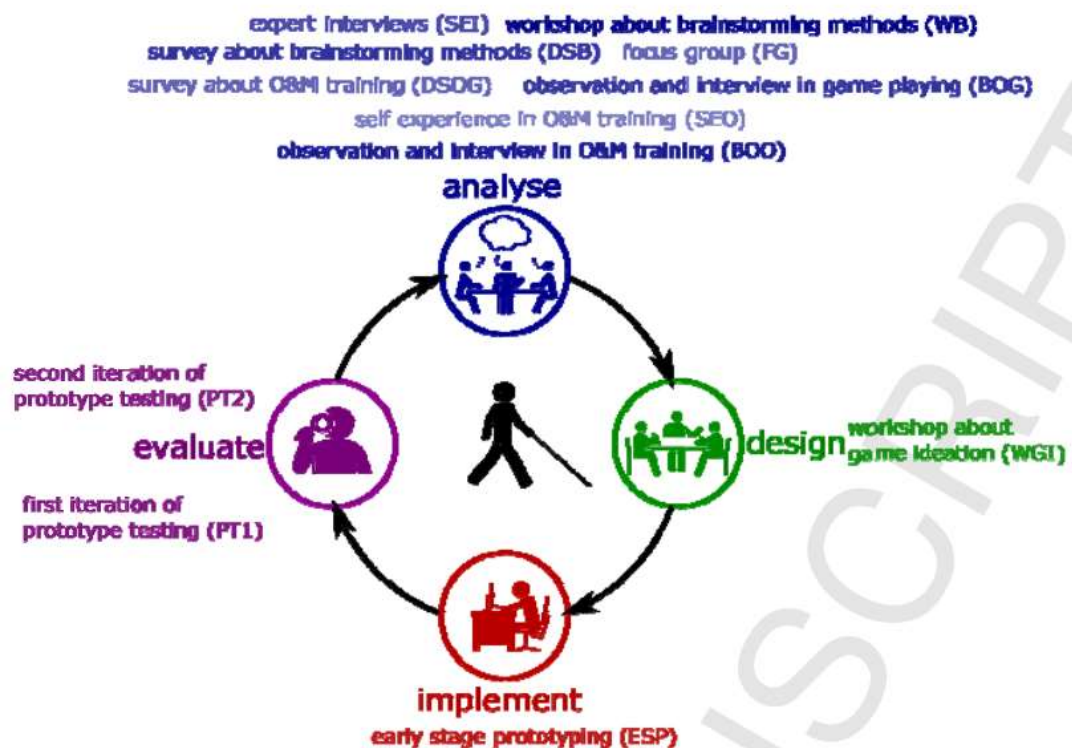
<b>Tipos de pruebas</b>
<p>Las "Monkey testing" (o prueba de mono) que son pruebas de caja negra. <i>"Se basa en la idea teórica de que un mono que usa al azar una máquina de escribir eventualmente teclea todos los escritos de Shakespeare. Cuando se aplica a las pruebas, consiste en un flujo aleatorio de eventos de entrada que se inyectan en la aplicación para hacer que se bloquee. Aunque esta técnica de prueba ejecuta ciegamente el juego sin ningún objetivo particular, es útil para detectar errores ocultos. Esta técnica se puede mejorar si los registros se analizan después de cada prueba de mono y se alimenta un algoritmo evolutivo con las conclusiones para hacer que la prueba sea cada vez más destructiva."</i></p>
<p>Para las pruebas funcionales e inmersa en ellas las pruebas de regresión, los autores plantean la automatización de pruebas: <i>"En la era de la nube, esto se ha convertido en un servicio proporcionado por empresas dedicadas a ofrecer entornos de desarrollo basados en la nube. Para aplicaciones con interfaces gráficas de usuario, esta prueba basada en casos de prueba requiere marcos específicos que verifiquen si la GUI cumple con sus especificaciones. Ejemplos de este tipo de software son Selenium (para aplicaciones web) y Appium (para Android e iOS). Deben hacer frente a los cambios de la GUI, manteniendo los casos de prueba originales aún válidos, uno de los problemas que abordamos en este trabajo para el campo de los videojuegos."</i></p>
<p><i>"Beta tester: Aspectos como la jugabilidad y la experiencia del usuario deben ser revisadas por beta testers, que son usuarios humanos que juegan los niveles del juego una y otra vez, haciendo algo ligeramente diferente cada vez. Su propósito es encontrar errores, fallas en las imágenes o comportamientos incorrectos e inesperados. Forman parte de la fase de prueba de garantía de calidad del software, que es una parte importante de todo el proceso de desarrollo del software. El uso de probadores beta requiere mucho tiempo y esfuerzo, y aumenta los costos de desarrollo"</i></p>

Fuente: Adaptado de Bécares, Costero y Gómez (2017)

#### 8.1.6.4 Enfoque de diseño basado en el usuario

Mattheiss et. al (2017) para el diseño de un juego serio, hicieron uso de User-Centred Design (UCD) este se encuentra definido por ISO 9241-210 (2009), en donde se propicia por una fuerte integración de los usuarios en las diferentes fases del proyecto, a continuación, en la figura, se detallan los métodos y proceso seguido:

Figura 10 Proceso User-Centred Design (UCD)



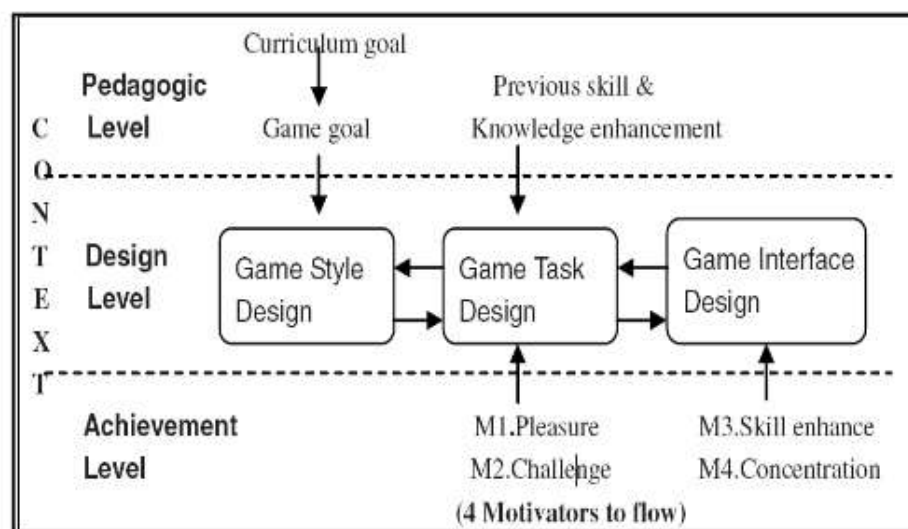
Fuente: Mattheiss et. al (2017)

Asimismo, en otros estudios como Del Blanco et. al (2017) y Palee et. al (2020) en donde se desarrollaron VJS, los autores en su proceso de desarrollo dieron importancia al involucramiento de usuarios o expertos en el área de enseñanza-aprendizaje; para el caso del proyecto de Palee et.al (2020) MechanicsDynamics Aesthetics MDA fue la base del proceso para el desarrollo del VJS.

### 8.1.6.5 Propuesta de modelo de diseño de juegos educativos

Ibrahim y Jaafar (2009) proponen el diseño de un marco de trabajo que incluye diseño de juegos, pedagogía y modelado de contenidos, como se ilustra en la siguiente figura:

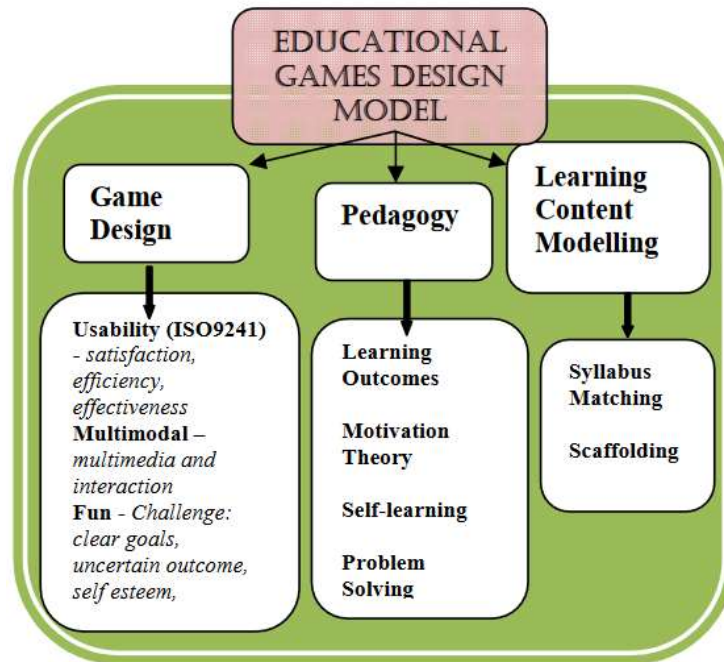
Figura 11 Modelo diseño de juegos educativos recopilado



Fuente: Ibrahim y Jaafar (2009)

Este modelo hace énfasis en la importancia de la disminución de complejidad de las tareas de diseño, generando aumentos en el flujo de cuatro factores: habilidades, desafío, concentración y placer. Partiendo de lo mencionado y recopilado por otros autores Ibrahim y Jaafar (2009) propusieron un modelo de diseño de juegos educativos, en donde el diseño de juego, la pedagogía y el modelamiento de contenido de aprendizaje, cumple un rol preponderante en su propuesta:

Figura 12 Propuesta de modelo de diseño de juegos educativos



Fuente: Ibrahim y Jaafar (2009)

Si bien el diseño de la propuesta no tiene enfoque al desarrollo de software o de VJ, permite, de acuerdo con el autor, determinar aspectos de carácter educativo a tener en cuenta para el desarrollo de VJS, o en nuestro caso para evaluar, tales como usabilidad, eficiencia, eficacia, resultados de aprendizaje, solución de problemas.

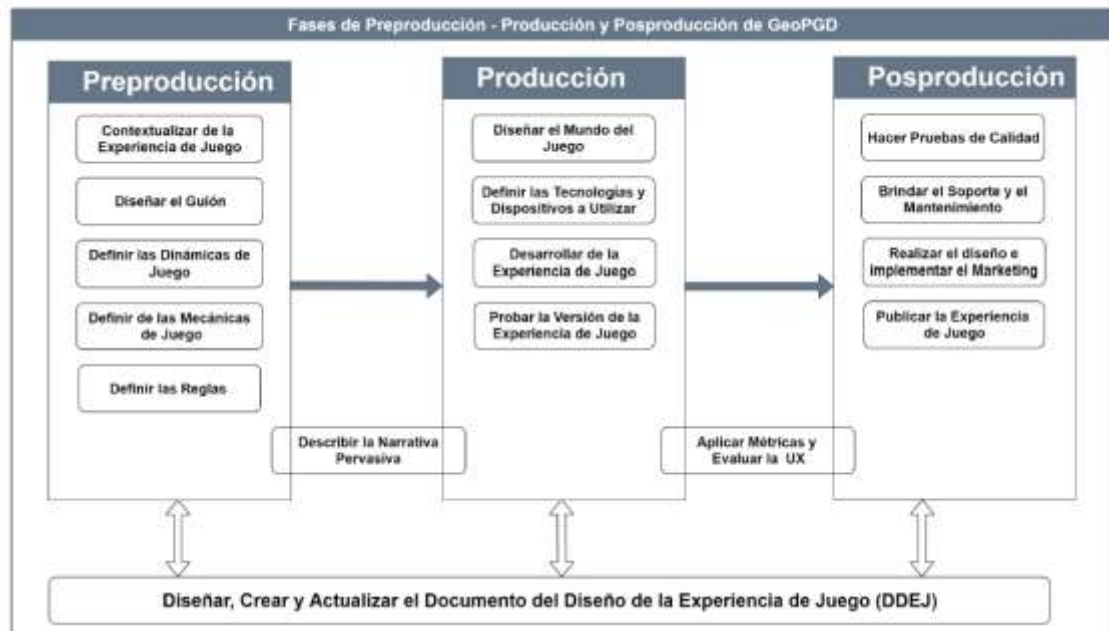
#### 8.1.6.6 Metodología para implementar videojuegos pervasivos

Arango (2019) plantea como objetivo de su investigación la propuesta de una metodología de software para el diseño e implementación de juegos pervasivos georreferenciados, en este caso el autor propone GeoPGD, esta metodología o proceso cuenta con 4 fases cíclicas: Evolución de la narrativa pervasiva, Mundo del juego, Reglas del juego y Dinámicas pervasivas; también el autor en cuestión propone una adaptación de GeoPGD en fases de preproducción, producción y posproducción, estas cercanas en los



conceptos con otros propuestas, a continuación se referencia el modelo adaptado de GeoPGD:

Figura 13 Propuesta fases adaptadas de GeoPGD



Fuente: Arango (2019)

De esta propuesta se resalta a nivel de las fases las siguientes actividades referentes a calidad:

### Producción

- Pruebas de Versión: se necesita realizar pruebas constantes de funcionalidades para mejorar en cada iteración y generar nuevas versiones.
- Métricas y Evaluación de UX: en esta fase es importante dar inicio a las mediciones y evaluación de los elementos de la experiencia de usuario que se deseen para la experiencia de juego. No es necesario considerar todas las posibilidades o mediciones disponibles, por el contrario, se debe tener en cuenta la opinión de un grupo de expertos para elegir las adecuadas.

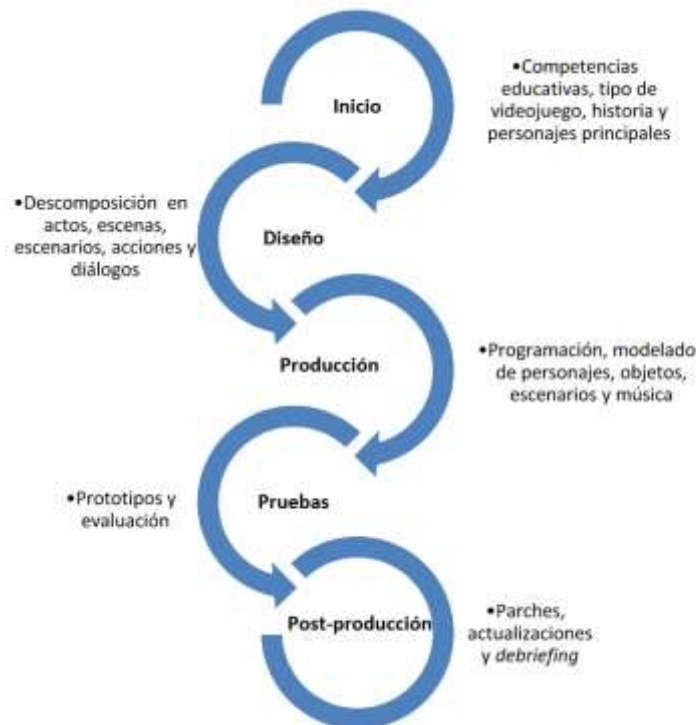
## Posproducción

- Pruebas de Calidad: en el equipo de desarrollo e infraestructura se deben definir las mediciones a aplicar en el entorno de la experiencia para dar una medida de calidad. Métricas y Evaluación de UX: al igual que en el atributo de calidad, la experiencia de usuario debe ser medida en función de los intereses del equipo de trabajo. Estas mediciones sirven como soporte para la toma de decisiones e incorporar o remover **funcionalidades, personajes, elementos, lugares, etc.**

### 8.1.6.7 Metodología para desarrollar videojuegos educativos

Para esta propuesta metodológica Prieto (2018) de forma cíclica el autor define algunas fases generales para el desarrollo de los videojuegos educativos:

Figura 14 Modelo de metodología



Fuente: Prieto (2018)



Tabla 10 Listado de Heurísticas aplicadas en desarrollo

<b>Tipo</b>	<b>Heurística</b>
Heurísticas de usabilidad: mediciones de la interfaz de usuario y los controles del juego	La terminología que se usó en el juego se podía entender fácilmente
	Los controles del juego eran intuitivos y fáciles de usar.
	Iconos y gráficos denotaron con precisión sus funciones.
	Preferiría que el juego se cambiara a paisaje en lugar de la orientación de retrato actual
Heurística de jugabilidad: mediciones de la participación del usuario y el juego continuo por parte del usuario	El tutorial del juego fue útil para aprender a jugar el juego.
	Los objetivos y desafíos del juego eran repetitivos.
	El progreso a través del juego fue demasiado lento en algunos puntos
	Las clasificaciones y los elementos sociales promovieron la competencia entre jugadores
	La competencia me motivó a jugar más el juego
Heurísticas de educabilidad: medidas de las características del juego que facilitan el aprendizaje	Se requería mucha memorización para jugar el juego.
	El juego era más educativo que entretenido.
	El juego proporcionó suficientes incentivos para saber más sobre las hierbas
	Aprendí más sobre los usos de varias hierbas al jugar el juego

	Usé los recursos con hipervínculos en el juego para leer más sobre las hierbas.
	El juego se volvió más fácil a medida que aprendí más sobre las hierbas
Heurística de movilidad: mediciones de las características del juego que facilitan el juego en movimiento y en diferentes entornos.	Pude jugar el juego en diferentes entornos, como en casa, en la escuela, en el transporte o mientras caminaba.
	Cada sesión de juego se puede detener o pausar fácilmente si es necesario.

Fuente: Adaptada de Prieto (2018)

#### 8.1.6.9 *Artefactos en los procesos de prueba*

De acuerdo a la revisión de Cano (2020) se identifica Game Accessibility Guidelines de Ablegamers Foundation (2012) que de acuerdo a su sitio web, es una guía creada colaborativamente por expertos, desarrolladores y académicos, para contribuir a la no exclusión de los jugadores, esta guía contiene 3 listas: básica, intermedia y avanzada, en cada una de ellas reposa recomendaciones específicas para aplicar en el proceso de desarrollo, pero así mismo sirve de base de pruebas al validar el cumplimiento de estas recomendaciones bajo el uso de listas de chequeo.

Figura 16 CheckList Game Accessibility

Guideline	Relevant to mechanic?	Implemented?
	[*no* automatically greys out row]	
<b>Motor</b>		
(Control / mobility)		
<b>Basic</b>		
<a href="#">Allow controls to be remapped / reconfigured</a>		
<a href="#">Ensure controls are as simple as possible, or provide a simpler alternative</a>		
<a href="#">Ensure that all areas of the user interface can be accessed using the same input method as the gameplay</a>		
<a href="#">Include an option to adjust the sensitivity of controls</a>		
<a href="#">Ensure interactive elements / virtual controls are large and well spaced, particularly on small or touch screens</a>		
<b>Intermediate</b>		
<a href="#">Support more than one input device</a>		
<a href="#">Make interactive elements that require accuracy (eg, cursor/touch controlled menu options) stationary</a>		

Fuente: Ablegamers Foundation (2012)

A lo anterior se suma Chandler (2009) puesto que, en su libro para la producción de videojuegos, plantea una lista de chequeo para la fase de pruebas. En esta lista se logran identificar algunos artefactos tal como el plan de pruebas:

Figura 17 Testing checklist

TESTING CHECKLIST	Y / N	NOTES
<b>VALIDATE PLAN</b>		
Is test plan written?		
Is game plan updated for QA?		
Has test plan been updated with any changes to the game plan?		
Are testing milestones accounted for in the schedule?		
Is bug-tracking software available for the testers and development team?		
Are all areas of the game tested?		
Are all bugs regressed and closed?		
<b>CODE RELEASE</b>		
Has development team submitted a final code release candidate?		
Is there ample time in the schedule for QA to complete the test plan on the code release candidate?		
Has QA approved the product for code release?		
CONSOLE ONLY: Has code released game been submitted to console manufacturer for approval?		
CONSOLE ONLY: Has console manufacturer approved game for final replication?		

Fuente: Chandler (2009)

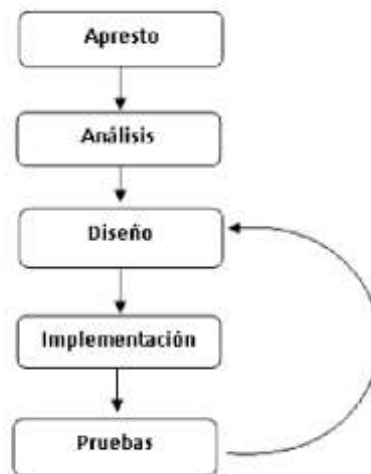
Para Ibarra (2018) es importante la liberación del producto, pero a la vez afirma que también tiene importancia lo que se debe comunicar sobre cada error, lo anterior permitirá que el equipo encargado de correcciones, cuenten con información precisa para dar atención a los errores, por lo anterior el autor propone un artefacto de reporte, entre los cuales resalta algunos aspectos a tener en cuenta: gravedad, frecuencia, prioridad y entre otros atributos.

Morales Urrutia, Nava López, Fernández Martínez y Rey Corral (2010) en su propuesta Huddle, mencionan otros artefactos tales como Sprint backlog, Burn-down charts, pero así mismo resaltan en su proceso los Buglist, elementos de reporte de errores para el proceso de depuración; también se destaca otro artefacto llamado Documento de diseño, este es el contenedor de especificaciones del videojuego.

#### 8.1.6.10 Ciclo de vida Juego educativo móvil “Educamovil”

Rodriguez et.al (2013) en su desarrollo aplican un ciclo de vida por fases, en el cual se incluye la fase de pruebas:

Figura 18 Ciclo de vida para juego educativo móvil



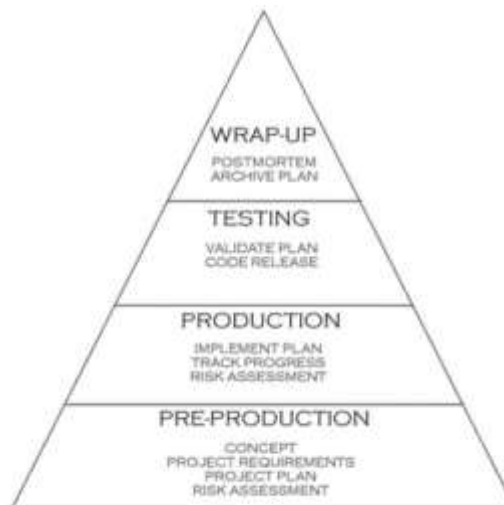
Fuente: Rodriguez et.al (2013)

En la fase de prueba, no se especifica más allá de mencionar, que se podrán aplicar pruebas por personal externo o interno, diferente a las personas involucradas en el desarrollo, también se referencia la posibilidad de aplicación de pruebas unitarias o simulaciones en el sistema, aunque lo anterior no entrega información más detallada de las pruebas a usar en este ciclo de vida.

#### 8.1.6.11 Otras propuestas para el desarrollo de videojuegos

Chandler (2009) propone un ciclo para la producción de desarrollo de videojuegos fases de preproducción, producción, testing y empaquetado, a continuación, se observa la figura con el ciclo básico y además la figura con múltiple iteración del ciclo de producción de desarrollo.

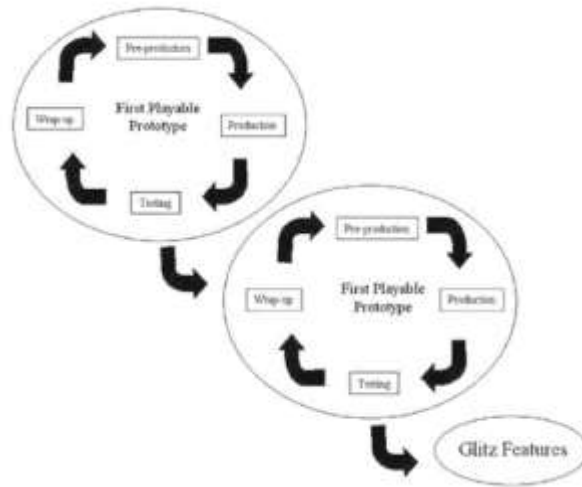
Figura 19 Ciclo básico de producción de videojuegos



Fuente: Chandler (2009)



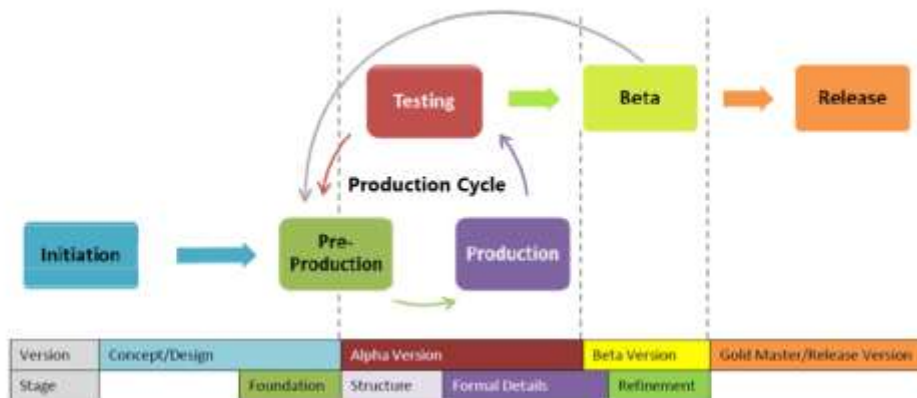
Figura 20 Múltiples ciclos de producción



Fuente: Chandler (2009)

Ramadan y Widyani (2013) realizan la propuesta de pautas para el ciclo de desarrollo de VJ, consta de etapas que contemplan algunas fases, tales como iniciación, preproducción, producción, beta y *release*. Algunas de estas fases en conjunto se repiten de forma iterativa (ciclo de producción) mientras que las demás obedecen a una lógica secuencial.

Figura 21 Ciclo de vida para el desarrollo de videojuegos



Fuente: Ramadan y Widyani (2013)

### **8.1.7 Análisis Y Resultados Del Mapeo Sistemático**

De acuerdo con la información general que se logró extraer, se puede apreciar que a nivel académico y productivo existen esquemas de ciclos de vida y procesos para el desarrollo de videojuegos, entre ellos se observan estructuras basadas en fases pero que incluyen subprocesos iterativos. En un gran número de las propuestas se aprecian etapas de preproducción y producción y de pruebas.

Si bien las propuestas en su mayoría contemplan la fase o etapa de pruebas, no se profundiza conceptualmente en esta fase vital para el aseguramiento de la calidad del software, en este caso de los videojuegos, sin embargo, se hace mención de diferentes tipos de pruebas sugeridas para su aplicabilidad: pruebas exploratorias, pruebas unitarias, funcionales y pruebas automatizadas; para los procesos de prueba se destaca dos momentos: pruebas alfa y pruebas beta, estos momentos recopilan diferentes tipos de prueba a ejecutarse de acuerdo al valor que tenga el producto en la línea de producción del juego, es decir son pruebas ejecutadas de acuerdo al tipo de versión de desarrollo.

En los procesos de desarrollo y en las fases de prueba se observa la participación de expertos en conocimientos, educadores para instruir y acompañar al equipo de desarrollo para el diseño y prueba del contenido a enseñar, aquí es relevante también destacar la existencia de herramientas como heurísticas y de evaluaciones para medir la calidad del videojuego, usabilidad, experiencia de usuario y la apropiación del conocimiento. Los aspectos mencionados, es lo reflejado en el diseño del proceso de prueba propuesto en la investigación que desarrolla en la sección 9.3.

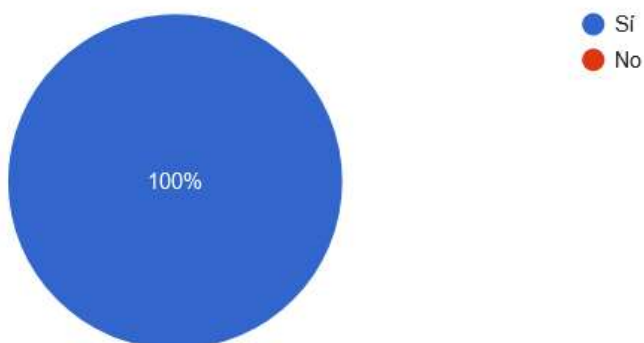
## **8.2 ENCUESTAS SECTOR INDUSTRIA DE VIDEOJUEGOS**

Con el fin de analizar el estado actual de los procesos de desarrollo y de prueba de videojuegos en la industria, se realizó una encuesta en donde participaron miembros de instituciones de carácter público del país, las cuales fueron: Tecnoparque SENA y Universidad de Caldas; entidades que tienen una experiencia adquirida por los procesos

académicos, investigativos y de producción de videojuegos. A lo anterior se suma las respuestas obtenidas a la encuesta por parte de trabajadores de empresas de carácter privadas dedicadas al desarrollo de videojuego, las empresas son: son Kiwanolab SAS, Wulfgames, Avalon Studios S.A.S.

Las personas de las anteriores instituciones y empresas en la encuesta manifestaron tener experiencia como gerentes de proyectos, líderes de diseño, director creativo, profesores e investigadores. El total de los consultados manifestó que la institución o empresa para la que trabaja ha desarrollado algún videojuego serio (educativo, formativo, de capacitación o terapéutico):

Figura 22 Respuestas ¿Alguno de los videojuegos desarrollados se podría identificar como un videojuego serio? (Videojuego Serio: Educativo, formativo, Capacitación, terapéutico)

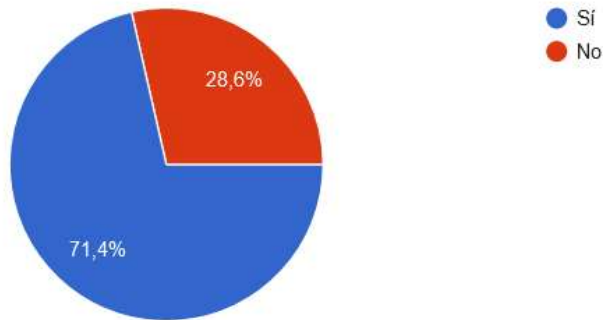


Fuente: Elaboración propia

Los encuestados manifiestan que han participado en proyectos de desarrollo de videojuegos serios para el tratamiento de fobias, enseñanza de contenido paisajista, entrenamiento de trabajadores para el mantenimiento de redes eléctricas y para el aprendizaje de autoexámenes para el diagnóstico de cáncer de mama. Lo anterior permitiendo definir que existe una diversidad entre los diferentes tipos de videojuegos serios desarrollados, pero a la vez identificar que hay experiencia en ellos en el desarrollo de videojuegos dedicados a la enseñanza.

A la población se le indagó por la gestión de sus proyectos, el 71.4% indicó que SCRUM es el marco de trabajo usado para la gestión y desarrollo de sus proyectos, el restante, es decir el 28.6% no hace uso de SCRUM para la gestión de sus proyectos:

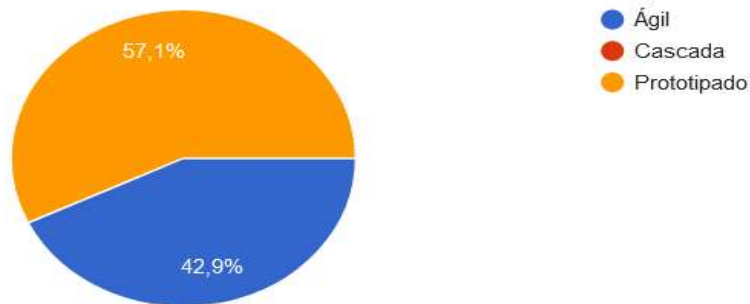
Figura 23 Respuestas ¿Para la gestión de proyectos de desarrollo de videojuegos, la empresa y el equipo de desarrollo hace uso del marco de trabajo SCRUM?



Fuente: Elaboración propia

Se le consultó a los encuestados por los ciclos de vida implementados en el desarrollo de videojuegos, siendo prototipados y ágiles los más usados respectivamente, a continuación, se relacionan los resultados, a diferente de lo recopilado en el mapeo sistemático, en este caso, las empresas no implementan ciclos de vida cascada:

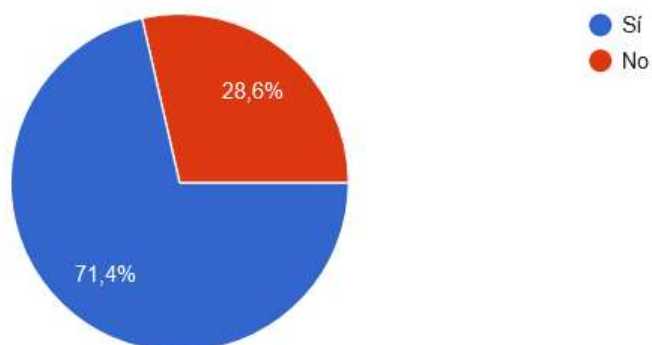
Figura 24 Respuestas ¿Qué ciclo de vida de desarrollo de software se implementa en el desarrollo de videojuegos?



Fuente: Elaboración propia

Ante la pregunta de si la empresa tiene definida alguna metodología para el desarrollo de videojuegos, el 28.6% respondió que no y el 71.4% indica que sí:

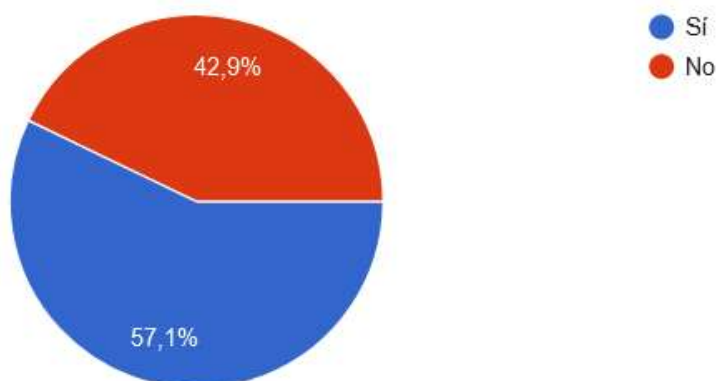
Figura 25 Respuestas ¿En la empresa se tiene definido o se sigue alguna metodología de desarrollo de videojuegos?



Fuente: Elaboración propia

Se consultó sobre la existencia de procesos definidos específicamente para probar videojuegos; en las respuestas se evidencia que, si bien cuentan con procesos de desarrollo definidos, este valor disminuye en cuanto a los procesos de prueba:

Figura 26 Respuestas ¿Existe en su empresa algún proceso de prueba definido específicamente para videojuegos?



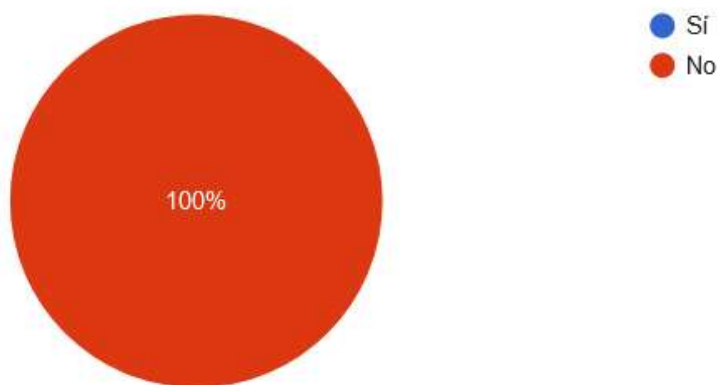
Fuente: Elaboración propia

Al solicitarle a los encuestados una breve descripción de los procesos de prueba usados para videojuegos, indican la aplicabilidad de lo siguiente:

- Testing interno > debug > Testing personas externas > debug > Test cliente > debug
- Uso de tester de videojuegos.
- Se realizan pruebas internas (alfa y beta) con diferentes grupos, algunos sin manuales de usuario, para probar no solo la mecánica del juego, sino que también qué tan intuitivo y fácil de usar/complementar es.

Se logra observar que si bien hay esbozos de posibles procesos, algunos de ellos describieron tipos de pruebas más que procesos en sí. Al consultar sobre la automatización de pruebas todos los consultados responden que no realizan automatización de pruebas para videojuegos:

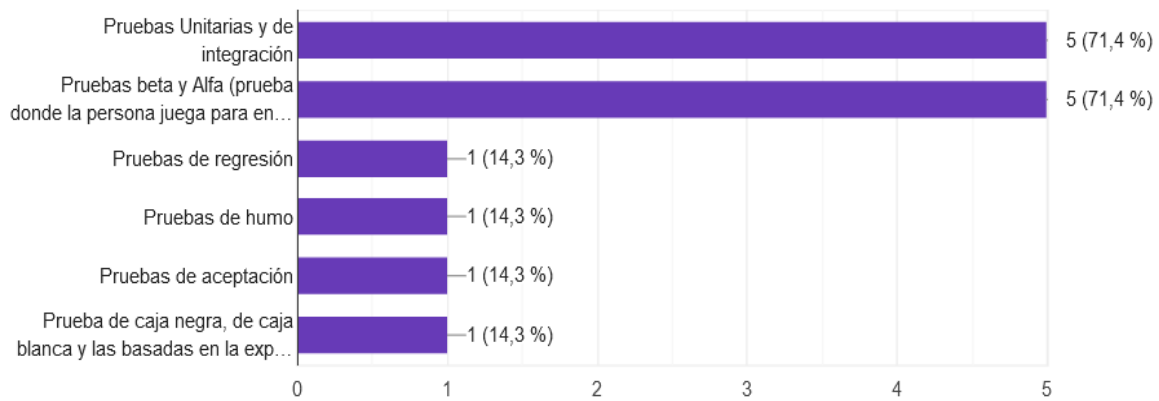
Figura 27 Respuestas ¿En la empresa en la cual labora, se realiza automatización de pruebas para los videojuegos?



Fuente: Elaboración propia

Ante la pregunta de los diferentes tipos de pruebas usadas y aplicadas, para los encuestados, las pruebas de integración, unitarias y pruebas alfa y beta con jugadores, son las más aplicadas:

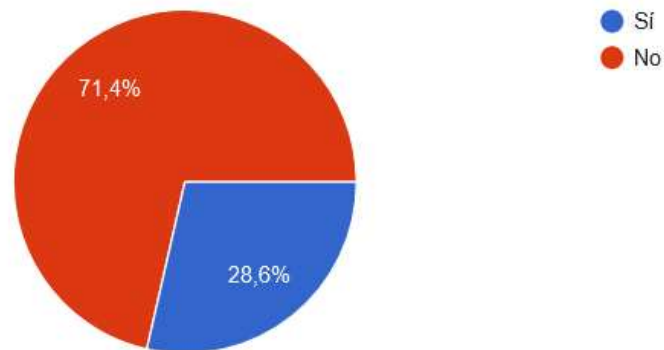
Figura 28 Respuestas Identificación de tipos de pruebas aplicadas



Fuente: Elaboración propia

Sobre la pregunta de implementación de heurísticas para medir usabilidad y jugabilidad el 71.4% indica que no implementan este tipo de validaciones, solo el 28.6% sí las implementa:

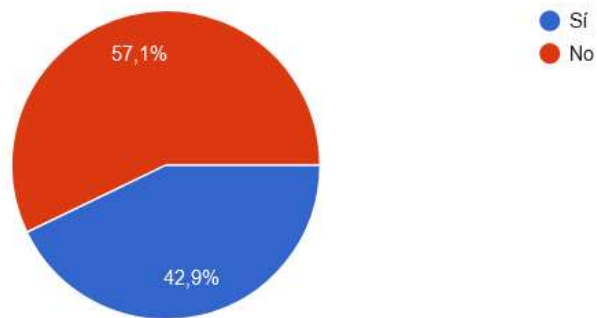
Figura 29 Respuestas Heurísticas jugabilidad y usabilidad



Fuente: Elaboración propia

Sobre la pedagogía o pruebas para validar la pedagogía de los VJS el 42.9% implementa algún tipo de prueba para medir lo mencionado, el 57.1% no implementa este tipo de pruebas:

Figura 30 Respuestas Pruebas para validar pedagogía del VJS



Fuente: Elaboración propia

Para finalizar el total de los encuestados, es decir el 100% indica que en sus desarrollos involucran expertos en el área o temática específica del contenido del VJS.

### 8.3 DISEÑO DE PROCESO DE PRUEBA DE VIDEOJUEGOS SERIOS EDUCATIVOS – AVIGATE

De acuerdo con los resultados obtenidos en el mapeo sistemático y en el proceso de encuesta del sector industrial de VJ, se realiza la propuesta de un proceso de prueba para videojuegos educativos, denominado AVIGATE; este proceso se propone en marco de un ciclo iterativo incremental ágil de desarrollo de software.

Figura 31 Esquema general de ciclo de desarrollo

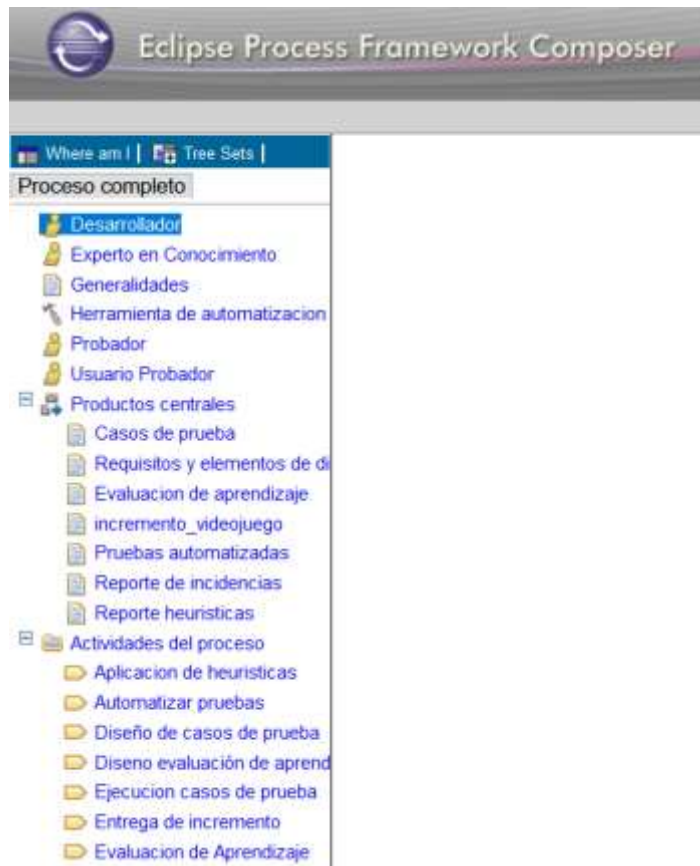


Fuente: Elaboración propia



De acuerdo con la propuesta metodológica se realiza un diseño de proceso haciendo uso de Software & System Process Engineering Metamodel versión 2.0 (SPEM 2.0). Para el proceso modelado en SPEM se identifican, roles, artefactos, actividades y flujos haciendo uso de la herramienta Eclipse Process Framework Composer (EPF).

Figura 32 Modelamiento proceso en EPF Composer



Fuente: Elaboración propia

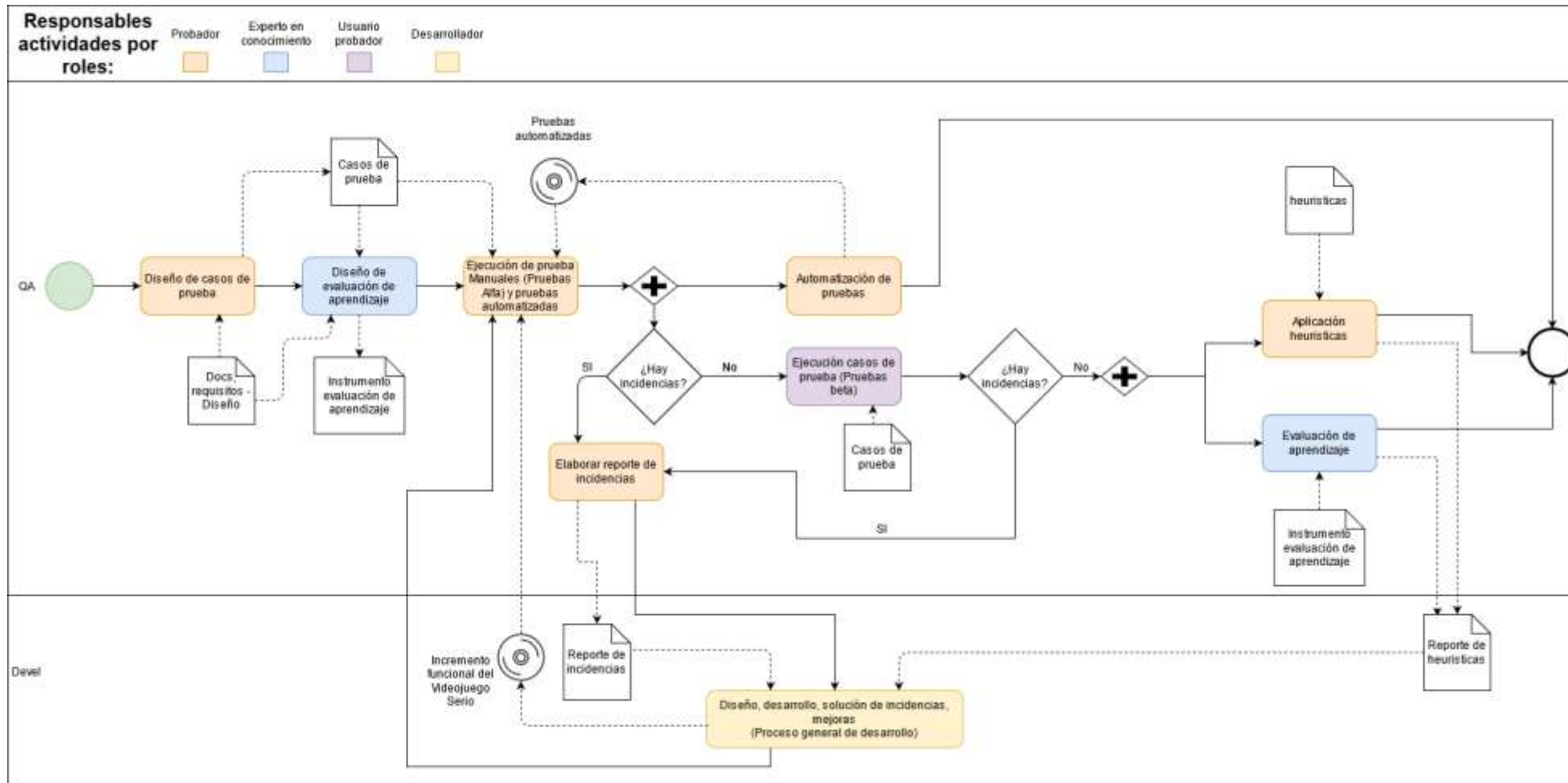
Para la propuesta de proceso de prueba y de acuerdo con el mapeo sistemático, se logran identificar algunos roles inmersos en el proceso de prueba:

- **Desarrollador:** Si bien dentro del proceso de prueba el desarrollador no cumple un papel preponderante más allá del desarrollo, se hace referencia a este rol, para dejar claro las actividades a realizar en el proceso, además es

importante indicar que el rol de desarrollador no tiene como fin realizar pruebas, pero en sus responsabilidades de desarrollo sí realiza actividades de aseguramiento de calidad, tales como aplicar estándares, pruebas unitarias, entre otras. En el caso del proceso de pruebas propuesto, es el encargado de hacer entrega del valor a probar y de atender las incidencias reportadas por los probadores.

- **Probador:** Es el encargado de validar el cumplimiento de los requerimientos del incremento de valor entregado por los desarrolladores, es decir, es el rol que se encarga de las actividades de diseñar y ejecutar casos de prueba (Pruebas alfa), automatizar pruebas, realizar reporte de incidencias y hacer aplicación de heurísticas.
- **Probador usuario:** Este rol es el encargado de hacer ejecución de las pruebas Beta y es en quien se aplican las heurísticas y la evaluación de aprendizaje, es un probador externo al equipo de desarrollo del proyecto, se sugiere que el probador usuario haga parte de la población objetivo del desarrollo del videojuego serio educativo.
- **Experto en conocimiento:** Al tratarse de videojuegos serios educativos, la difusión y comunicación de contenido específico para su apropiación, requiere de personas expertas en el área para validar que el contenido de interés de enseñanza que se está usando en el videojuego esté de acuerdo a cómo se estipule en los requerimientos de desarrollo; el experto es quien realiza la propuesta de diseño de evaluación de aprendizaje y es quien aplica la evaluación diseñada en los usuarios probadores para medir el nivel de aprendizaje del videojuego al usarse.

Figura 33 Diagrama proceso de pruebas AVIGATE



Fuente: Elaboración propia

Con EPF se realizó una descripción detallada del proceso (ver anexos del 1 al 25), pero también haciendo uso de Business Process Model and Notation (BPMN), se realizó el modelamiento del proceso de prueba descrito en la *figura 33*, este diagrama permite tener una óptica general del ciclo de actividades, relacionamiento de productos y de roles en el proceso.

En el proceso se cuenta con tareas determinadas en momentos específicos del ciclo de prueba, estas contemplan el uso de artefactos o documentos como entradas para la generación de productos: reportes, casos de prueba, automatizaciones. A continuación, se listan las actividades con sus entradas, productos y roles encargados:

- Diseño de casos de prueba:
  - Rol: Probador
  - Entradas: Documentos de requisitos y de diseño
  - Productos: Casos de prueba
- Diseño de evaluación de aprendizaje
  - Rol: Experto en conocimiento
  - Entradas: Documentos de requisitos y de diseño. Casos de prueba.
  - Productos: Instrumento de evaluación de aprendizaje
- Ejecución de pruebas Manuales y pruebas automatizadas (Pruebas alfa):
  - Rol: Probador
  - Entradas: Incremento funcional del videojuego serio, Casos de prueba y Pruebas automatizadas (Opcional).
  - Productos: No aplica
- Automatización de pruebas (Opcional):
  - Rol: Probador
  - Entradas: No aplica
  - Productos: Pruebas automatizadas
- Elaborar reporte de incidencias:
  - Rol: Probador

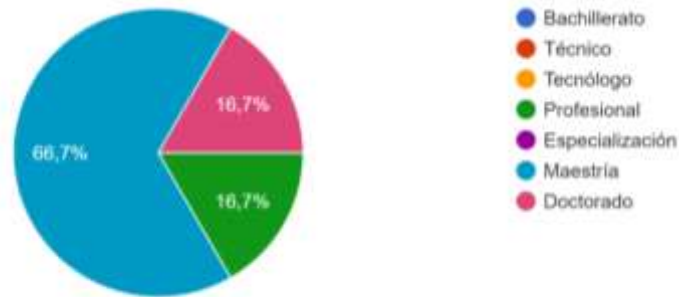
- Entradas: No aplica
- Productos: Reporte de incidencia
- Ejecución de casos de prueba (Pruebas beta):
  - Rol: Usuario probador
  - Entradas: Casos de prueba
  - Productos: No aplica
- Aplicación de heurísticas:
  - Rol: Probador
  - Entradas: Heurísticas (Usabilidad, Experiencia de aprendizaje, Experiencia de juego y fidelidad)
  - Productos: Reporte de heurísticas
- Evaluación de aprendizaje:
  - Rol: Experto en conocimiento
  - Entradas: Instrumento de evaluación de aprendizaje
  - Productos: Reporte de heurísticas
- Diseño, desarrollo, solución de incidentes, mejoras:
  - Rol: Desarrollador
  - Entradas: Reporte de incidencias y reporte de heurísticas
  - Productos: Incremento funcional del videojuego serio
  - NOTA: Si muy bien estas son actividades de desarrollo y no de prueba, se referencian en el proceso para indicar en cuál momento se sugiere atender los reportes, producto de las actividades de prueba para las acciones correspondiente. No es del alcance de la investigación indicar el cómo y cuándo realizar el desarrollo.

## **8.4 EVALUACIÓN POR EXPERTOS**

Para verificar la pertinencia del proceso de prueba de videojuegos serios educativos, se socializó el proceso diseñado con expertos en las áreas de interés, posterior a ello por medio de una encuesta se logró obtener la retroalimentación de validación del proceso. Al

consultar sobre el nivel académico de los evaluadores evidenciamos que un 67% de la población encuestada tiene maestrías, un 16,7% cuenta con un doctorado y el 16.7% restante tiene un título profesional, lo cual indica que el nivel académico con el que cuentan las personas encuestadas demuestra capacidad de experticia o conocimientos por su formación:

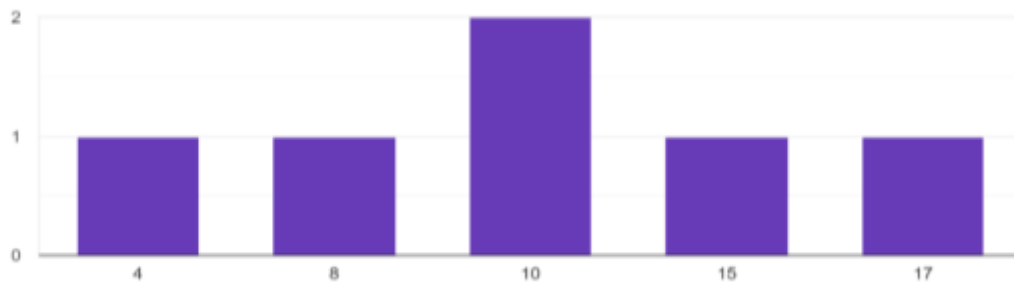
Figura 34 Nivel de formación de expertos evaluadores



Fuente: Elaboración propia

A lo anterior, se suma la pregunta en el cual se indaga por la formación profesional de los encuestados, las respuestas son del núcleo del área de la ingeniería de sistemas y computación y afines, además del área de diseño y desarrollo 3D. En cuanto a la experiencia en el área de Ingeniería de Software, todos los encuestados manifestaron tener experiencia:

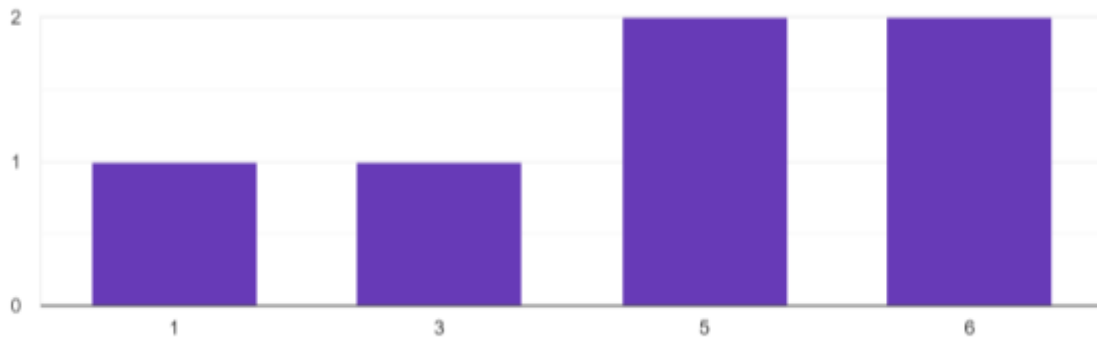
Figura 35 ¿Cuál es su tiempo de experiencia en el área de Ingeniería de Software? (En años)



Fuente: Elaboración propia

Con lo anterior se detalla que los encuestados manifestaron tener entre 4 y 17 años en experiencia en el área de ingeniería de software, dos de ellos con 10 años de experiencia. Para el área de Calidad del Software, si bien el valor es mucho menor en cuanto a lo consultado en la *figura 36*, todos los encuestados dieron a conocer su experiencia en esta área:

Figura 36 ¿Cuál es su tiempo de experiencia en el área de Calidad del Software? (en años)



Fuente: Elaboración propia

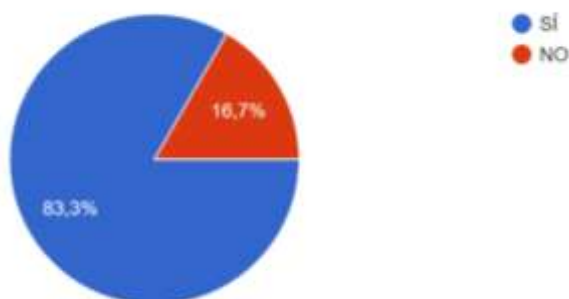
Al consultar a los expertos por su experiencia en el área de desarrollo de videojuegos, solo uno de los consultados no tenía experiencia en este tipo de desarrollos, los demás tenían entre 2 a 6 años de experiencia.

De acuerdo con estas primeras preguntas base realizadas a los expertos que evaluaron el proceso, se logra determinar y dar constancia de la experticia y conocimiento de los evaluadores en las áreas de interés de la investigación realizada, siendo idóneos para evaluar las características del proceso.

Se indagó primeramente a los encuestados sobre si el proceso de prueba ágil diseñado establecía sus actividades y flujo de forma lógica, para lo que las respuestas indican un 83.3% por sí y un 16.7% que no, al solicitar justificación de la respuesta, los encuestados que optaron por indicar que el proceso no cumplía con actividades y flujos lógicos, estos indicaron la importancia de considerar la evaluación de la calidad del

software y la calidad de aprendizaje de manera independiente y una después de la otra de forma respectiva.

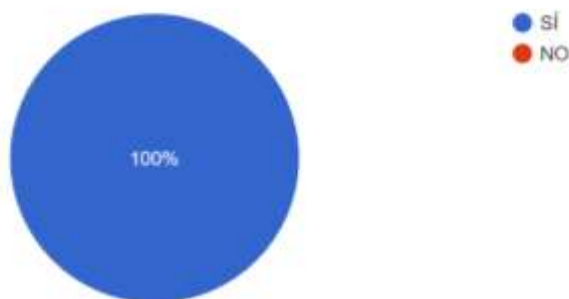
Figura 37 Resultado encuesta actividades y flujo de forma lógica



Fuente: Elaboración propia

Además de lo mencionado, uno de los evaluadores realiza un comentario que se considera importante, indicando que no todos los videojuegos serios permiten la enseñanza de contenido, muchos videojuegos serios contribuyen al desarrollo de otro tipo de habilidades. En otra pregunta se les indagó a los encuestados sobre si el proceso diseñado abarcaba las actividades generales de un proceso de prueba, el 100% indicó que sí, con ello obteniendo resultados satisfactorios para el proceso:

Figura 38 ¿Cree usted que el proceso de prueba diseñado abarca las actividades generales de un proceso de prueba?



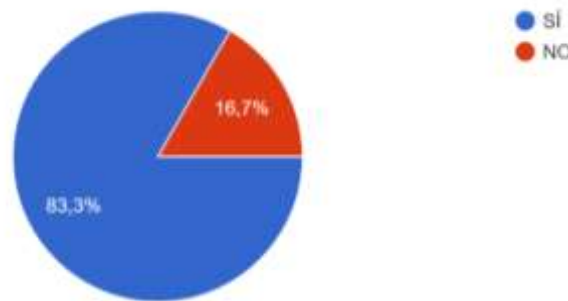
Fuente: Elaboración propia



Igual que a la anterior pregunta, los encuestados respondieron el 100% que sí, a las preguntas: ¿Cree que el proceso de prueba diseñado puede aportar al contexto académico y a la industria del desarrollo de videojuegos? ¿Cree usted que el proceso diseñado aporta a la calidad del videojuego y a la calidad de los contenidos a enseñar? Las respuestas anteriores fueron acompañadas por algunos comentarios de los encuestados en donde proponen la posible vinculación del proceso con la cultura DevOps, además sobre el cómo ejecutar pruebas manuales y cómo realizar la automatización de estas pruebas y como verificar UX.

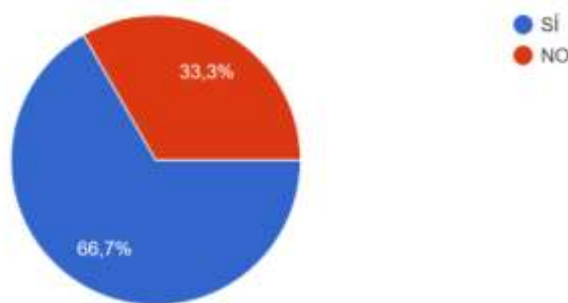
Para finalizar la encuesta se indagó sobre la descripción gráfica general del proceso, ver *figura 33*, se preguntó sobre el cumplimiento de los principios ágiles y sobre los roles definidos en el proceso; para esto los encuestados respondieron de la siguiente forma:

Figura 39 ¿Cree usted que la descripción gráfica del proceso permite un entendimiento general del proceso?



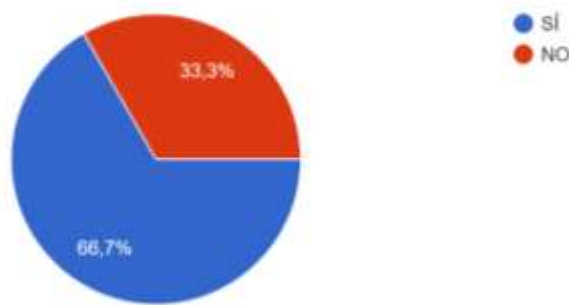
Fuente: Elaboración propia

Figura 40 ¿Cree usted que el proceso se enmarca en los principios ágiles?



Fuente: Elaboración propia

Figura 41 ¿Cree usted que los roles definidos en el proceso son adecuados y suficientes?



Fuente: Elaboración propia

En cuanto a las anteriores preguntas, se detallan comentarios de justificación de los encuestados en donde indican la relevancia de acuerdo con las metodologías ágiles, de la participación de los interesados, clientes o dueños del producto, no solo en el proceso de desarrollo, sino que también, en este caso, en el proceso de pruebas, también atribuyen falta de claridad en cuanto a lo que se entiende en el proceso como probador usuario.

## 8.5 VALIDACIÓN DEL PROCESO DE PRUEBA EN EL DESARROLLO DE VIDEOJUEGO

Para la validación del proceso de prueba de videojuegos serios educativos, se contó con el apoyo y participación de estudiantes de la Universidad de Caldas, del pregrado de Ingeniería en Sistemas y Computación que cursaban la asignatura de Proyecto Integrador de noveno semestre del programa. Uno de los grupos realizó su desarrollo y aplicó el proceso diseñado, dicho grupo recibió formación en el proceso de prueba por parte del investigador del proyecto y también contó con acompañamiento y apoyo con el fin de aportar madurez a la aplicabilidad del proceso.

Ambos equipos realizaron el desarrollo de un nivel de un VJS educativo para la enseñanza de la historia de Colombia, el grupo 1 (G1), con 3 miembros, desarrolló un nivel aplicando el proceso diseñado de prueba para videojuegos y la temática correspondiente fue al momento histórico de la Batalla de Boyacá y el grupo 2 (G2), con 2 miembros, realizó el

desarrollo sin aplicar el proceso, solamente basándose en las recomendaciones aplicadas desde la metodología o proceso de desarrollo; el desarrollo del G2 correspondió al momento histórico de la revuelta de los comuneros. Cada grupo, por el tiempo asignado al semestre (cuatro meses), realizó la ejecución de tres iteraciones de Scrum, es decir tres Sprint, por lo tanto, el tiempo dedicado no permitió la culminación de los desarrollos de cada grupo, pero sí permitió la obtención de información del proceso de prueba requerida para la validación.

Figura 42 Desarrollo G1 Batalla del Puente de Boyacá



Fuente: Elaboración propia

Figura 43 Desarrollo G2 Revuelta de los comuneros



Fuente: Elaboración propia

El desarrollo del VJS se realizó por medio del motor de videojuegos Unity, usando el lenguaje C#, el videojuego tiene formato de 3D en tercera persona, lo anterior generando un nivel mayor de complejidad por su interacción con los diferentes elementos relacionados en el mapa del videojuego, se involucraron mecánicas de animación de movimiento, de cámara y personajes y colisión con objetos como factor de disparadores de eventos.

Figura 44 Modelamiento terreno



Fuente: Elaboración propia

El modelamiento del terreno y objetos fue complejo por los conocimientos y conformación del equipo de desarrollo, puesto que su desarrollo carece de acompañamiento de personal idóneo y capacitado para la ilustración, renderización de texturas y modelamiento de objetos y personajes 3D, por consiguiente, se hizo uso de Assets disponibles gratuitamente o se diseñaron de forma empírica por los miembros del equipo de desarrollo; restándole tiempo a la dedicación de desarrollo de funcionalidades debido la búsqueda o diseño de objetos. Por lo anterior no se logró involucrar en el desarrollo tecnologías disruptivas como Realidad Virtual, aspecto que hubiese sido valioso medir o para resaltar aún más la dificultad del desarrollo.

Para el desarrollo del videojuego se contó con la participación del investigador como rol de *Product Owner* del producto, es decir del videojuego, así mismo, se contó con el acompañamiento de un experto en el área de Historia para el proceso de levantamiento de requerimientos del videojuego y para su inmersión en el proceso de prueba como rol de experto de conocimiento, para el diseño de evaluación de aprendizaje y aplicación.

Para ambos grupos se hizo uso del marco de trabajo SCRUM para hacer la gestión del proyecto, así mismo ambos equipos de desarrollo implementaron partes del proceso de desarrollo Ágil-UC, un proceso de desarrollo para pequeñas organizaciones planteado por Hurtado y Franco (2021); este proceso enmarca una serie de actividades, guías, roles y actividades que contribuyen a la orientación del desarrollo del software para pequeños equipos.

Al revisar el detalle de pruebas realizado por cada uno de los grupos se detallan algunos aspectos relevantes:

Tabla 11 Comparación general de grupos de desarrollo y pruebas

<b>G1</b>	<b>G2</b>
Diseñaron casos de prueba en todas las iteraciones.	Diseñaron casos de prueba en todas las iteraciones.
Hubo consistencia en los casos de prueba, aplicaron un mismo formato, el entregado por el proceso de prueba.	No hubo consistencia en los casos de prueba, no aplicaron un mismo formato.
Aplicaron pruebas Alfa y Beta, pruebas ejecutadas por diferentes personas (Personas internas y externas).	Aplicaron un solo ciclo de pruebas, pruebas por personal interno del equipo.
Se aplicaron heurísticas de UX, jugabilidad y aprendizaje.	No se aplicaron heurísticas.
No se automatizaron pruebas.	No se automatizaron pruebas.

Se realizaron reportes de incidencias y de heurísticas.	Se realizaron reportes de incidencias, no de heurísticas.
Hubo consistencia en los reportes de incidencias aplicaron un mismo formato, el entregado por el proceso de prueba.	Hubo consistencia en los casos de prueba, aplicaron un mismo formato.

Fuente: Elaboración propia

Otros aspectos importantes para destacar es la relación de reporte de incidencias, evaluación de aprendizaje y valoraciones de heurísticas:

Tabla 12 Comparación cuantitativa de resultados de los grupos

<b>G1</b>	<b>G2</b>
2 reportes de incidencias	3 reportes de incidencias.
12 incidentes reportados	36 incidentes reportados
1 evaluación de heurísticas aplicadas.  3 valoraciones de heurísticas:  - Dificultad del juego por orientaciones de cámara vs controles.  - Ubicación de menú de diálogos en pantalla.  - Dificultad de legibilidad de texto de aprendizaje	0 evaluaciones de heurísticas aplicadas.  0 valoraciones de heurísticas.
1 evaluación de aprendizaje diseñada y aplicada:	0 evaluación de aprendizaje

<ul style="list-style-type: none"><li>- 100% de evaluados (dos usuarios probadores) identifican correctamente el momento histórico del videojuego</li><li>- 0% de evaluados (dos usuarios probadores) identifican correctamente la batalla del pantano de Vargas como la batalla anterior a la batalla de Boyacá.</li></ul>	
---	--

Fuente: Elaboración propia

## 9 DISCUSIÓN DE RESULTADOS

De acuerdo con lo relacionado en la *tabla 8* y posteriormente en la *sección 8.1.5* se evidencian estudios relacionados con la investigación, un primer aspecto a resaltar de los resultados del mapeo sistemático realizado es la existencia de variedad de procesos o metodologías de desarrollo de VJ, esto a la vez refleja una variedad en los ciclos de vida aplicados en cada proceso. Si muy bien existen propuestas metodológicas o procedimentales para el desarrollo de VJ, existen un número inferior de propuestas para el desarrollo de VJS.

Asimismo, se puede evidenciar en el mapeo sistemático y en la extracción de información que, si bien estos procesos identificados indican actividades de aseguramiento de calidad o pruebas, no se profundiza en los planteamientos de subprocesos o actividades más concretas de procesos de prueba, lo anterior en concordancia con lo recopilado en la encuesta al sector industrial en la *sección 9.2* en donde se evidencia que si bien existen empresas o instituciones con procesos definidos para el desarrollo, en menor proporción cuentan con procesos definidos para realizar pruebas.

Se contó con referentes en donde los procesos de prueba aplicaron heurísticas o evaluaciones de experiencia de usuario, usabilidad, experiencia de juego y de aprendizaje, a pesar de la existencia de estos referentes, en la encuesta realizada al sector de la industria, se determina que mayoritariamente no todas las empresas o instituciones indagadas aplican heurísticas para realizar las mediciones ya mencionadas, teniendo en cuenta que estas instituciones indicaron realizar desarrollo de videojuegos serios.

Una de las problemáticas en el desarrollo y en los procesos de prueba es la repetición constante de pruebas y para ello la automatización cumple un papel importante, en nuestro caso, según lo consultado al sector industrial, no se realiza automatización de pruebas en sus ciclos de desarrollo y prueba, a esto suma lo ocurrido en la validación del proceso con el desarrollo de un VJS, el equipo de prueba omitió la realización de este tipo de actividad tal como se indica en la *Tabla 11*; lo anterior nos permite cuestionarnos sobre



las limitantes tecnológicas existentes para la realización de este tipo de pruebas en el desarrollo de VJS.

Para la validación del proceso, como es ya de conocimiento, se aplicó una evaluación con expertos y un experimento en donde se implementó el proceso diseñado en el desarrollo de un videojuego. De acuerdo con lo consultado primeramente con los expertos, se determina que por su campo de accionar laboral y por su experiencia, son personas con conocimiento en el área, dándonos la certeza de ser personas capacitadas para realizar una pertinente evaluación del proceso diseñado. Un primer aspecto cuestionado por uno de los expertos es el concepto de proceso de prueba para videojuegos serios, ya que no todos los videojuegos serios tienen como objetivo la enseñanza, en algunos casos tienen como fin el desarrollo de habilidades físicas, con esto, las actividades de evaluación de aprendizaje en nuestro proceso, no aplicarían para este tipo de videojuegos; por tal razón, y teniendo en cuenta la apreciación del evaluador, acotamos el objeto del proceso, limitando su aplicación al desarrollo de los videojuegos serios educativos y no a los videojuegos serios en general.

Otra apreciación general realizada por la evaluación de los expertos, es sobre el ciclo de vida en el cual se pretende implementar el proceso de prueba, es decir, en los ciclos de vida de desarrollo ágiles, puesto que ellos creen que para mantener los principios ágiles en el proceso, es pertinente -al igual que en el desarrollo- involucrar a los interesados o clientes en las pruebas; esta observación es pertinente revisarla para que pueda ser un aspecto de mejora en una nueva versión del proceso o también para dar una mayor claridad sobre si el interesado o cliente puede ocupar el rol de Usuario Probador, que actualmente se encuentra definido en el proceso de prueba.

En la validación del proceso por su implementación, se obtuvieron algunos aspectos importantes a destacar, primeramente, la evaluación de heurísticas y métricas de aprendizaje, puesto que es un aspecto que el grupo control G2 no tuvo en cuenta a la hora de plantear y ejecutar sus pruebas, así mismo la ejecución de pruebas beta permitieron al G1 que implementó el proceso diseñado, obtener mayor certeza de menos posibilidad de

presencia de incidencias en lo cubierto por las pruebas, ya que personal externo, es decir el rol Usuario Probador del proceso, ejecutó las pruebas en un segundo momento y así mismo este rol, con la aplicación de la evaluación de heurísticas, permitió retroalimentar funcionalidades que si bien cumplían con requerimientos funcionales, no satisfacían criterios de jugabilidad o aprendizaje para el usuario; esto último toma relevancia al valorar la prontitud de obtener dichas retroalimentaciones, sin esperar a realizar lanzamientos costosos y demorados para obtener evaluaciones de los usuarios finales de los videojuegos.

El G1 con la evaluación de conocimiento en los usuarios probadores, permitió identificar aspectos funcionales o pedagógicos que no contribuyen a la enseñanza de los contenidos, aspecto que no fue medido por el G2, lo anterior dando la oportunidad de retroalimentación oportuna que tuvo el G1 al identificar en la prueba de evaluación de conocimiento que el 0% de los jugadores, es decir que los usuarios probadores no identificaron correctamente en la evaluación un aspecto de antesala a la Batalla del Puente de Boyacá.

Otro aspecto importante a destacar, es que el G2 que no hizo implementación del proceso diseñado, realizó 3 informes de reportes de incidencia con 36 incidentes halladas, mientras que el G1 que sí implementó el proceso, logró realizar 2 informes de reportes de incidencias con 12 incidencias identificadas; si muy bien los valores cuantitativos destacan una mayor producción de informes y de incidentes para el G2 que no implementó el proceso, es importante mencionar que mayor cantidad de pruebas en el G2 no significó mayor cobertura en las pruebas, puesto que desde nuestro análisis, de las 36 incidencias halladas, 19 hacían referencia a un mismo núcleo de prueba, es decir, colisiones con objetos, si muy bien esto es de relevancia probar, no refleja mayor variabilidad de pruebas, a diferencia del G1 que en sus 12 incidencias, se observa mayor variabilidad en las funciones bajo prueba.

Si muy bien no se pretende cuestionar el diseño de las pruebas de ambos grupos, si se pretende destacar la variabilidad de pruebas tanto funcionales, de heurísticas y de evaluación de conocimiento ejecutadas por el G1 de acuerdo con lo estipulado en el

proceso. Por último, la ausencia de la automatización de pruebas, en la validación se asemeja a lo obtenido al consultar al sector productivo de la industria del videojuego, puesto que se determina la no realización de pruebas automáticas.

## 10 CONCLUSIONES

Se realizó una definición e implementación de un proceso para pruebas de VJS, el cual fue verificado por expertos y validado en el desarrollo de un videojuego en un proyecto académico. El nivel de madurez en su aplicación podría permitir realizar mayores mediciones o valoraciones a medida que un equipo a largo o mediano plazo gana mayor experiencia con el proceso; las métricas de heurísticas de usabilidad, jugabilidad y experiencia de aprendizaje, contenidas en los artefactos del proceso y aplicadas en la actividad de evaluación, permiten retroalimentar oportunamente a un equipo de desarrollo de videojuegos, sobre el uso correcto de buenas prácticas para este tipo de software que tiene una finalidad específica diferente a la de diversión.

La carencia de pruebas automáticas en este tipo de desarrollos, a pesar de su gran uso en el desarrollo y pruebas de otro tipo de software, abre la puerta para continuar con estudios en esta temática en específico, bien sea por un problema cultural en este segmento de la industria, pertinencia de este tipo de pruebas en el desarrollo de videojuegos o por la ausencia de herramientas tecnológicas para cumplir con la buena práctica ágil de automatización de pruebas de videojuegos.

El proceso cumple con una evaluación positiva, de acuerdo con lo recopilado en los resultados, aunque se deben revisar algunos aspectos concernientes a roles, en específico con el involucramiento de los interesados o clientes, este aspecto podría ser generador de una nueva versión del proceso para dar un mayor cumplimiento a los principios ágiles.

Los procesos de desarrollo de videojuegos y videojuegos serios se pueden ver beneficiados del proceso diseñado, por la complementariedad de un subproceso de prueba inmerso en el ciclo de desarrollo, presentando buenas prácticas y actividades para el aseguramiento de la calidad del videojuego, y no solo en los requerimientos funcionales, sino también en aspectos de la calidad del aprendizaje, de la jugabilidad y experiencia de usuario. Para ello en los artefactos planteados dentro del proceso se dejan a disposición

heurísticas para medir los aspectos ya mencionados, pero también es posible que estas heurísticas o mediciones puedan ser complementadas por otras expuestas por otros autores.

## 11 RECOMENDACIONES

Se recomienda para el proceso en una segunda versión, dar claridad sobre el papel de los interesados o clientes en el proceso de prueba, de acuerdo con lo comentado por la evaluación de expertos para dar un mayor cumplimiento o acercamiento a los principios ágiles. Para lograr un entendimiento general de proceso diseñado, no solo se recomienda revisar el diagrama general del proceso *Figura 33*, sino que también las particularidades de las actividades, roles y artefactos descritos en EPF Composer *Figura 32*.

También se recomienda revisar aspectos que impidan la realización de pruebas automáticas en dichos desarrollados por lo ya analizado con respecto a la falta de automatizaciones según lo recopilado en la industria y lo validado en el proceso de desarrollo; y para finalizar, se hace la recomendación de aplicar el proceso de prueba diseñado en un entorno productivo no académico, es decir en una empresa, puesto que las dinámicas académicas, limitan el accionar de los grupos y sus integrantes y por ende limita la ejecución e implementación del proceso.

## 12 BIBLIOGRAFÍA

- Acerenza, N., Coppes, A., Mesa, G. & Viera, A. (2009). *Una Metodología Ágil para Desarrollo de Videojuegos*. Recuperado de <https://www.fing.edu.uy/inco/grupos/gris/wiki/uploads/ProyectosGrado/Acerenza-etAl-ProyGrado.pdf>
- Agile Modeling. (2019). *Effective Practices for Modeling and Documentation*. Recuperado de <http://agilemodeling.com/>
- Ahmad, N.B., Barakji, S.A.R., Abou Shahada, T.M. & Anabtawi, Z.A. (2017). How to launch a successful video game: A framework. *Entertainment computing*, 23, 1-11. DOI: <https://doi.org/10.1016/j.entcom.2017.08.001>
- Arango, J. (2019) GeoPGD metodología para la implementación de juegos pervasivos georreferenciados apoyados en Linked Open Data. Recuperado de <https://dialnet.unirioja.es/servlet/tesis?codigo=244765>
- Awad, M.A. (2005). A comparison between agile and traditional software development methodologies. *University of Western Australia*, 30. Recuperado de [https://www.academia.edu/38895564/A\\_Comparison\\_between\\_Agile\\_and\\_Traditional\\_Software\\_Development\\_Methodologies](https://www.academia.edu/38895564/A_Comparison_between_Agile_and_Traditional_Software_Development_Methodologies)
- Blanco, A., Torrente, J., Fernández, B., Ruiz, P. & Giner, M. (2017) Using a videogame to facilitate nursing and medical students first visit to the operating theatre. A randomized controlled trial. Recuperado de <https://doi.org/10.1016/j.nedt.2017.04.026>
- Castillo, E., Meza, E., Garcés, S y Camacho, W. (2016) Micromundo etnoeducativo como apoyo al docente en las prácticas de enseñanza de la lengua nam trik de Totoró “namoi po jaumai amkun”. Recuperado de <https://dialnet.unirioja.es/servlet/articulo?codigo=5772740>
- Chandler, H. M. (2009). *The game production handbook*. Jones & Bartlett Publishers
- De Kerckhove, D. (1999). *Inteligencias en conexión: hacia una sociedad de la web*. Barcelona: Gedisa.

- Franco, O., Ameller, D., Costal, D. & Franch, X. (2017). Open source software ecosystems: A Systematic mapping. *Information and software technology*, 91, 160-185. Recuperado de <http://www.sciencedirect.com/science/article/pii/S0950584917304512>
- González Gill, J.A. y Pérez, E. (2017). Una propuesta metodológica para la construcción de videojuegos. *Memorias De Congresos UTP*, 1(1), 64-70. 26 de mayo de 2019. Recuperado de <https://revistas.utp.ac.pa/index.php/memoutp/article/view/1299>
- Hernández Bécares, J., Costero Valero, L. & Gómez Martín, P. (2017). An approach to automated videogame beta testing. *Entertainment Computing*, 18, 79-92. Recuperado de <https://doi.org/10.1016/j.entcom.2016.08.002>
- Ibarra, Y. (2018). Proceso de pruebas en el desarrollo de videojuegos. Recuperado de <https://repositorio.uci.cu/handle/123456789/7929>
- Ibrahim, R. y Jaafar, A. (2009). Educational games (EG) design framework: Combination of game design, pedagogy and content modeling. pp. 293-298, doi: 10.1109/ICEEI.2009.5254771.
- IEEE Computer Society (1993). IEEE Guide for Software Verification and Validation Plans. Recuperado de <https://standards.ieee.org/standard/1059-1993.html>
- IEEE Computer Society (2014). *SWEBOK versión 3*. Recuperado de <https://www.computer.org/education/bodies-of-knowledge/software-engineering/v3>
- Internacional Software Testing Qualifications Board (ISTQB). (2011). *Code coverage*. Recuperado de <https://glossary.istqb.org/en/term/code-coverage>
- Internacional Software Testing Qualifications Board (ISTQB). (2011). *Software development lifecycle*. Recuperado de <https://glossary.istqb.org/en/search/SDLC>
- Internacional Software Testing Qualifications Board (ISTQB). (2011). *Agile software development*. Recuperado de <https://glossary.istqb.org/en/term/agile-software-development>
- Internacional Software Testing Qualifications Board (ISTQB). (2012). *Static code analysis*. Recuperado de <https://glossary.istqb.org/en/term/static-code-analysis>



- Internacional Software Testing Qualifications Board (ISTQB). (2014). *Foundation Level Extension Syllabus Agile Tester*. Recuperado de <https://www.istqb.org/downloads/category/5-agile-tester-extension-documents.html>
- Internacional Software Testing Qualifications Board (ISTQB). (2018). *Test process*. Recuperado de <https://glossary.istqb.org/en/term/test-process-3>
- Internacional Software Testing Qualifications Board (ISTQB). (2014). *Agile Tester Extension in a Nutshell*. Recuperado de [https://www.anztb.org/userfiles/files/Agile\\_Tester\\_Nutshell\\_2014\\_May.pdf](https://www.anztb.org/userfiles/files/Agile_Tester_Nutshell_2014_May.pdf)
- Internacional Software Testing Qualifications Board (ISTQB). (2011). *Agile testing*. Recuperado de <https://glossary.istqb.org/en/term/agile-testing-2>
- Internacional Software Testing Qualifications Board (ISTQB). (2019). *Foundation Level Gambling Industry Tester* Recuperado de <https://www.istqb.org/certification-path-root/gambling-industry-tester.html>
- Jones, C, (2013) [PNSQC]. *Pacific NW Software Quality Conference, octubre 14-16, Portland, Obregon - PNSQC 2013 - Capers Jones* [Archivo de video]. Recuperado de <https://www.pnsc.org/software-quality-in-2013-survey-of-the-state-of-the-art/>
- Kosa, M., Souza, L., Veado, R.T. & Figueiredo, H. (2016). Software engineering education and games: a systematic literature review. *Journal of Universal Computer Science*, 22(12), 1558-1574. Recuperado de [https://www.researchgate.net/publication/312530807\\_Software\\_Engineering\\_Education\\_and\\_Games\\_A\\_Systematic\\_Literature\\_Review](https://www.researchgate.net/publication/312530807_Software_Engineering_Education_and_Games_A_Systematic_Literature_Review)
- Lee, J., Kang, S. & Jung, P. (2020). Test coverage criteria for software product line testing: Systematic literature review. *Information and Software Technology*, 122, 106272. Recuperado de <https://www.sciencedirect.com/science/article/abs/pii/S0950584920300227>
- Lin, D., Bezemer, C. y Hassan, A. (2019). Identifying gameplay videos that exhibit bugs in computer games. *Empirical Software Engineering*, 24(6), 4006-4033. Recuperado de

[https://www.researchgate.net/publication/333259544\\_Identifying\\_gameplay\\_videos\\_that\\_exhibit\\_bugs\\_in\\_computer\\_games](https://www.researchgate.net/publication/333259544_Identifying_gameplay_videos_that_exhibit_bugs_in_computer_games)

López Echeverry, A., Cabrera, C. y Valencia Ayala, L. (2008). Introducción a la calidad de software. *Scientia et Technica*, 2(39), 326-331. DOI:

<http://dx.doi.org/10.22517/23447214.3241>

Mascheroni, M., Greiner, C., Petris, R., Dapozo, G. & Estayno, M. (2012). Calidad de software e ingeniería de usabilidad. Recuperado de

<http://sedici.unlp.edu.ar/handle/10915/19202>

Marcano, B. (2008). Juegos serios y entrenamiento en la sociedad digital. [Versión electrónica]. *Teoría de la Educación: educación y cultura en la sociedad de la información*, 9 (3), 93-107. Recuperado de

<https://gredos.usal.es/handle/10366/56633>

Mattheiss, E., Regal, G., Sellitsch, D. y Tscheligi, M. (2017). User-centred design with visually impaired pupils: A case study of a game editor for orientation and mobility training. Recuperado de <https://doi.org/10.1016/j.ijcci.2016.11.001>

Moen, R. & Norman, C. (2006). *Evolution of the PDCA cycle*. Recuperado de <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.470.5465&rep=rep1&type=pdf>

Moizer, J., Lean, J., Dell'Aquila, E., Walsh, P., Keary, A., O'Byrne, D., Di Fernando, A., Miglino, O., Friedrich, R., Asperges, R. & Sica, L. (2019). An approach to evaluating the user experience of serious game. *Computers & Education*.

Recuperado de <https://doi.org/10.1016/j.compedu.2019.04.006>

Morales Urrutia, G., Nava López, C., Fernández Martínez, L. y Rey Corral, M. (2010). Procesos de desarrollo para videojuegos. *Cultura Científica y Tecnológica*, 7(36/37), 25-39. Recuperado

de <http://148.210.132.19/ojs/index.php/culcyt/article/view/299>

Moreno, P., Torrente, J., Hsieh, Y. & Lester, W. (2012). Usability Testing for Serious Games: Making Informed Design Decisions with User Data. *Hindawi*. DOI: 10.1155/2012/369637

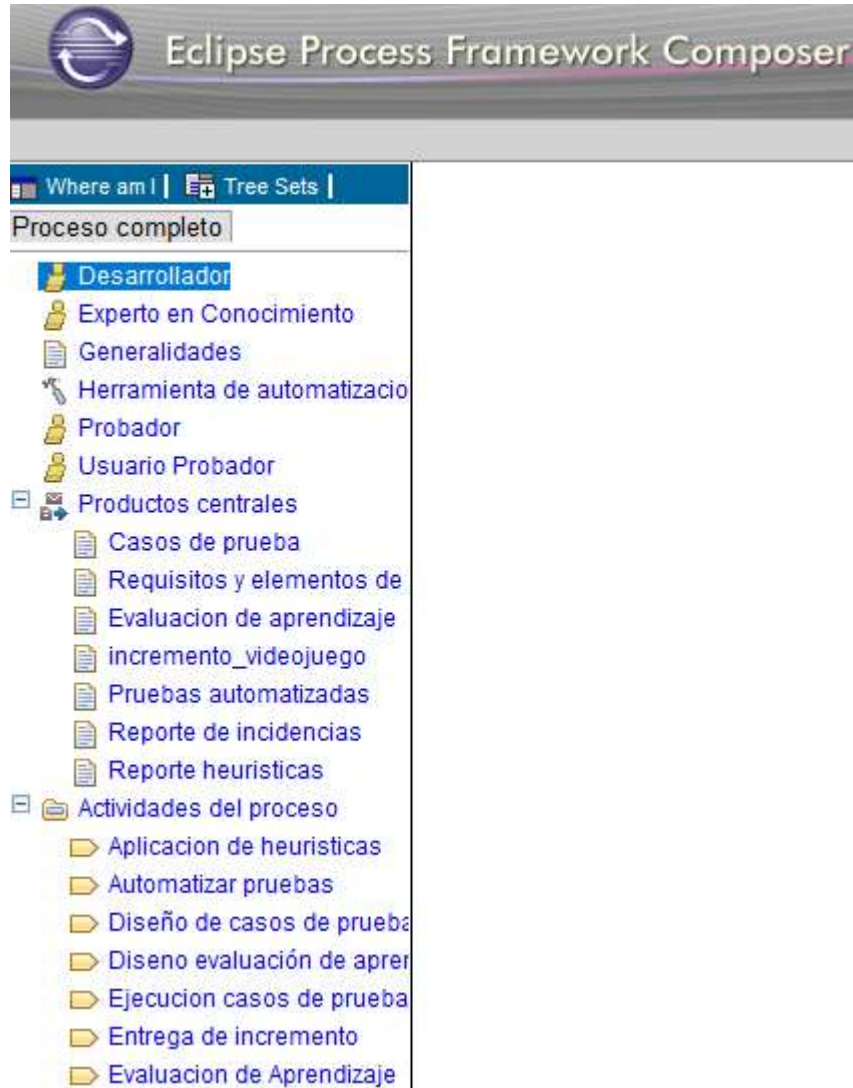
- Navarro Cadavid, A., Fernández Martínez, J. y Morales Vélez, J. (2013). Revisión de metodologías ágiles para el desarrollo de software. *Prospectiva*, 11(2), 30-39. Recuperado de <https://dialnet.unirioja.es/servlet/articulo?codigo=4752083>
- Nielsen, J. & Molich, R. (1990). Heuristic evaluation of user interfaces. En *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 249-256). Recuperado de <https://dl.acm.org/doi/abs/10.1145/97243.97281>
- Object Management Group (OMG). (abril de 2008). SPEM Software & Systems Process Engineering Metamodel. *OMG*. Recuperado de <https://www.omg.org/spec/SPEM/2.0/>
- Palee, P., Wongta, N., Khwanngern, K. y Choosri, N (2020). Serious Game for Teaching Undergraduate Medical Students in Cleft lip and Palate Treatment Protocol. Recuperado de <https://doi.org/10.1016/j.ijmedinf.2020.104166>
- Pernett Estrada, A. (2012) Relaciones entre el juego, el videojuego y la comunicación en el ámbito educativo. *Arte & Diseño*, 10(1), 6-8. Recuperado de <http://repositorio.uac.edu.co/handle/11619/1139>
- Perchy, S. y Nishi, S. (2013). *Tecnologías y Metodologías para la Producción, Diseño y Desarrollo de Videojuegos*. Recuperado de [http://cic.javerianacali.edu.co/~ysperchy/home/articles/game\\_production.pdf](http://cic.javerianacali.edu.co/~ysperchy/home/articles/game_production.pdf)
- Pérez García, A. y Ortega Carrillo, J. (2011). El potencial didáctico de los videojuegos: “The Movies”, un videojuego que fomenta la creatividad audiovisual. *Etic@net: Revista científica electrónica de Educación y Comunicación en la Sociedad del Conocimiento*, (10). Recuperado de <https://dialnet.unirioja.es/servlet/articulo?codigo=3702672>
- Pinelle, D., Wong, N. & Stach, T. (abril de 2008). Heuristic evaluation for games: usability principles for video game design. En *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1453-1462). Recuperado de <https://dl.acm.org/doi/abs/10.1145/1357054.1357282>
- Prieto, R., (2018) Metodología de desarrollo para videojuegos educativos basada en notaciones gráficas. Recuperado de <https://digibug.ugr.es/handle/10481/52309>

- Ramadan, R. y Widyani, Y. (2013). Game development life cycle guidelines. pp. 95-100, doi: 10.1109/ICACSSIS.2013.6761558
- Redavid, C. & Farid, A. (2011). *An Overview of game Testing Techniques*. Recuperado de <https://api.semanticscholar.org/CorpusID:5653083>
- Ridao, M. (2018). Metodologías de desarrollo en videojuegos. Recuperado de <https://dialnet.unirioja.es/servlet/articulo?codigo=6595158>
- Rodriguez, J., Gallardo, O., Rivera, A. y Smith, O. (2013). Ciclo de vidaJuego educativo movil “Educamovil”. Recuperado de <https://dialnet.unirioja.es/servlet/articulo?codigo=4966236>
- Rosyid, H., Palmerlee, M. y Chen, K. (2018) Deploying learning materials to game content for serious education game development: A case study. Recuperado de <https://doi.org/10.1016/j.entcom.2018.01.001>
- Ruppel, U. y Schatz, K. (2011). Designing a BIM-based serious game for fire safety evacuation simulations. Recuperado de <https://doi.org/10.1016/j.aei.2011.08.001>
- Rus, A. (2019). Aplicación de analíticas en la sistematización del diseño y validación de juegos serios para usuarios con discapacidad intelectual. Recuperado de <https://dialnet.unirioja.es/servlet/tesis?codigo=251540>
- Schultz, C. & Bryant, R. (2011). *Game Testing: All in onet*. Recuperado de <https://books.google.com.co/books?id=OD6TDgAAQBAJ&lpg=PT18&dq=video%20game%20testing%20method&lr&pg=PT19#v=onepage&q&f=true>
- Scott, L., Jeffery, R., Carvalho, L., D’Ambra, J. & Rutherford, P. (2001). *Practical software process improvement – The IMPACT project*. Recuperado de <https://ieeexplore.ieee.org/abstract/document/948512>
- Serrano, A. (2017). Mejorando la evaluación de juegos serios mediante el uso de analíticas de aprendizaje. Recuperado de <https://dialnet.unirioja.es/servlet/tesis?codigo=151994>
- Statista. (marzo de 2018a). PC and console games revenue worldwide from 2012 to 2022, by type. *Statista*. Recuperado de <https://www.statista.com/statistics/237187/global-pc-console-games-revenue-by-type/>

- Statista. (septiembre de 2018b). Value of the global video games market from 2012 to 2021. *Statista*. Recuperado de <https://www.statista.com/statistics/246888/value-of-the-global-video-game-market/>
- SUM. (sin fecha). SUM para desarrollo de videojuegos. *SUM*. Recuperado de <http://www.gemserk.com/sum/>
- Torres, M., Paz, K. y Salazar, F. (2019). Métodos de recolección de datos para una investigación. *Boletín Electrónico*, 03. Recuperado de <http://biblioteca.udgvirtual.udg.mx/jspui/handle/123456789/2817>
- Zapata, C.M., Arboleda, A. S. y Castrillón, C.E. (2010). Una representación gráfica del testing ágil. *Avances en Sistemas e Informática*, 7(2), 17-26. Recuperado de <https://revistas.unal.edu.co/index.php/avances/article/view/26492/26914>
- Xuan, R., Zhen, K. & Yi-Lwern, K. (2018) Herbopolis – A mobile serious game to educate players on herbal medicines. *Complementary therapies in medicine*. Recuperado de <https://doi.org/10.1016/j.ctim.2018.05.004>

## 13 ANEXOS

### Anexo 1 Elementos generales del proceso



Fuente: Elaboración propia

## Anexo 2 Rol de desarrollador

**Rol: Desarrollador**

Rol asignado a miembros del equipo encargados de la producción de valor, incremento funcional del Videjuego Serio

Expand All Sections Collapse All Sections

**Relationships**

Categories	<ul style="list-style-type: none"> <li>Proceso completo</li> </ul>
Primary Performs	<ul style="list-style-type: none"> <li>Entrega de incremento</li> </ul>
Modifies	<ul style="list-style-type: none"> <li>incremento_videojuego</li> </ul>

Back to top

**Main Description**

Es el encargado de diseñar y desarrollar las necesidades de requerimientos del videojuego y corregir incidencias reportadas por tester.

Mark as done

Fuente: Elaboración propia

## Anexo 3 Rol de experto de conocimiento

**Rol: Experto en Conocimiento**

Rol asignado a miembro del equipo experto en el área objeto de enseñanza del videojuego serio.

Expand All Sections Collapse All Sections

**Relationships**

Categories	<ul style="list-style-type: none"> <li>Proceso completo</li> </ul>
Primary Performs	<ul style="list-style-type: none"> <li>Diseño evaluación de aprendizaje</li> <li>Evaluación de Aprendizaje</li> </ul>
Modifies	<ul style="list-style-type: none"> <li>Evaluación de aprendizaje</li> <li>Reporte heurísticas</li> </ul>

Back to top

**Main Description**

Es el encargado de asesorar al equipo en el área de conocimiento que se espera enseñar en el videojuego, es el encargado de diseñar instrumento de evaluación de conocimiento generado por el videojuego, se encarga de evaluar y generar recomendaciones.

Back to top

Fuente: Elaboración propia

## Anexo 4 Rol de probador

**Role: Probador**

Rol asignado a miembros del equipo: encargados de validar el cumplimiento de requerimientos del incremento desarrollado por el Rol Desarrollador

Expand All Sections Collapse All Sections

**Relationships**

<b>Categories</b>	<ul style="list-style-type: none"> <li>Proceso completo</li> </ul>
<b>Primary Performs</b>	<ul style="list-style-type: none"> <li>Automatizar pruebas</li> <li>Diseño de casos de prueba</li> <li>Ejecucion casos de prueba</li> </ul>
<b>Additionally Performs</b>	<ul style="list-style-type: none"> <li>Aplicación de heurísticas</li> <li>Diseño evaluación de aprendizaje</li> <li>Evaluación de Aprendizaje</li> </ul>
<b>Modifies</b>	<ul style="list-style-type: none"> <li>Casos de prueba</li> <li>Pruebas automatizadas</li> <li>Reporte de incidencias</li> </ul>

Back to top

**Main Description**

Es el rol encargado de aportar al diseño del videojuego, diseñar los casos de prueba, desarrollar las pruebas automatizadas, ejecutar pruebas y pasar la aplicación de las heurísticas.

Fuente: Elaboración propia

## Anexo 5 Rol de usuario probador

**Role: Usuario Probador**

Rol asignado a usuarios del videojuego: encargados de aplicar el instrumento de heurísticas

Expand All Sections Collapse All Sections

**Relationships**

<b>Categories</b>	<ul style="list-style-type: none"> <li>Proceso completo</li> </ul>
<b>Primary Performs</b>	<ul style="list-style-type: none"> <li>Aplicación de heurísticas</li> <li>Evaluación de Aprendizaje</li> </ul>
<b>Modifies</b>	<ul style="list-style-type: none"> <li>Reporte de incidencias</li> <li>Reporte heurísticas</li> </ul>

Back to top

**Main Description**

Encargado de usar (probar) el videojuego para aplicar el instrumento de heurísticas y resaltar las necesidades funcionales/no funcionales del videojuego.

Back to top

Fuente: Elaboración propia



## Anexo 6 Artefacto casos de pruebas

**Artifac: Casos de prueba**

Documento descriptivo de las pruebas diseñadas para validar el cumplimiento de los requerimientos desarrollados en los videojuegos.  
Work Product Kinds: Productos centrales

[Expand All Sections](#) [Collapse All Sections](#)

**Purpose**  
Debe describir los requisitos que se esperan en el videojuego de forma detallada.

**Relationships**

Roles	Responsible	Modified By
	<ul style="list-style-type: none"><li>• Probador</li></ul>	<ul style="list-style-type: none"><li>• Probador</li></ul>
Tasks	<p>Input To:</p> <ul style="list-style-type: none"><li>• Aplicación de heurísticas</li><li>• Automatizar pruebas</li><li>• Diseño evaluación de aprendizaje</li><li>• Ejecución casos de prueba</li></ul>	<p>Output From:</p> <ul style="list-style-type: none"><li>• Diseño de casos de prueba</li></ul>

[Back to top](#)

**Description**

**Main Description**  
Es la especificación detalla de de los pasos a realizar por parte del tester para validar el funcionamiento correcto y alineado a los requerimientos del software (videojuego)

[Back to top](#)

**Illustrations**

**Templates**

- Biblioteca de pruebas
- Formato Casos de Prueba
- Formato Reporte de Incidencias

[Back to top](#)

**Key Considerations**  
Debe indicar claramente las acciones a realizarse y el resultado esperado, para definir el cumplimiento o no de los criterios.

[Back to top](#)

Fuente: Elaboración propia

## Anexo 7 Requisitos y elementos de diseño

**Artifac: Requisitos y elementos de diseño**

Es la documentación que detalla los requisitos del videojuego, diseño detallado y diseño de interfaz.  
Work Product Kinds: Productos centrales

[Expand All Sections](#) [Collapse All Sections](#)

**Purpose**  
Debe describir los requisitos que se esperan en el videojuego de forma detallada.

**Relationships**

Roles	Responsible	Modified By
	<ul style="list-style-type: none"><li>• Desarrollador</li><li>• Probador</li></ul>	
Tasks	<p>Input To:</p> <ul style="list-style-type: none"><li>• Diseño evaluación de aprendizaje</li><li>• Diseño de casos de prueba</li><li>• Entrega de incremento</li></ul>	<p>Output From:</p>

[Back to top](#)

**Description**

**Main Description**  
Si muy bien el alcance del proceso de test no es determinar los elementos del proceso de desarrollo, pero si los de prueba, se recomienda que este producto, sea la entrada relacionada a la documentación que describe los requisitos y diseño de los módulos o niveles del videojuego (historias de usuario, flujos de conversación, diseños de mundos, características de enseñanza)

[Back to top](#)

**Illustrations**

**Templates**

- Formato Casos de Prueba

[Back to top](#)

Fuente: Elaboración propia

## Anexo 8 Artefacto evaluación de aprendizaje

**Artifact: Evaluacion de aprendizaje**

 Es un instrumento que se aplicara para medir lo aprendido por los usuarios  
Work Product Kinds: Productos centrales

[Expand All Sections](#) [Collapse All Sections](#)

**Purpose**  
El proposito es medir lo que fue aprendido por el usuario que uso el videojuego serio.  
[Back to top](#)

**Relationships**

<b>Roles</b>	<b>Responsible:</b> <ul style="list-style-type: none"><li>• Experto en Conocimiento</li></ul>	<b>Modified By:</b> <ul style="list-style-type: none"><li>• Experto en Conocimiento</li></ul>
<b>Tasks</b>	<b>Input To:</b> <ul style="list-style-type: none"><li>• Evaluacion de Aprendizaje</li></ul>	<b>Output From:</b> <ul style="list-style-type: none"><li>• Diseño evaluación de aprendizaje</li></ul>

[Back to top](#)

**Description**

<b>Main Description</b>	Es un instrumento de cuestionario, diseñado por expertos o conocedores del area especifica a enseñar en el videojuego, este cuestionario mide que fue aprendido por los usuarios del videojuego. El cuestionario no tiene formato o guía, se crea a consideración de los expertos de cada area.
-------------------------	---

[Back to top](#)

Fuente: Elaboración propia

## Anexo 9 Artefacto incremento de videojuego

**Artifact: incremento\_videojuego**

 Desarrollo de incremento funcional del videojuego producido por el rol Devel  
Work Product Kinds: Productos centrales

[Expand All Sections](#) [Collapse All Sections](#)

**Purpose**  
Es la funcionalidad desarrollada por el rol Devel.  
[Back to top](#)

**Relationships**

<b>Roles</b>	<b>Responsible:</b> <ul style="list-style-type: none"><li>• Desarrollador</li></ul>	<b>Modified By:</b> <ul style="list-style-type: none"><li>• Desarrollador</li></ul>
<b>Tasks</b>	<b>Input To:</b> <ul style="list-style-type: none"><li>• Ejecucion casos de prueba</li></ul>	<b>Output From:</b> <ul style="list-style-type: none"><li>• Entrega de incremento</li></ul>

[Back to top](#)

**Description**

<b>Main Description</b>	Representa el incremento funcional del videojuego dispuesto para ser validado por el rol Tester.
-------------------------	--

[Back to top](#)

**Key Considerations**  
Debe cumplir con los requerimientos y diseños definidos, debe ser funcional para ser probado.  
[Back to top](#)

Fuente: Elaboración propia

## Anexo 10 Artefacto pruebas automatizadas

**Artifact: Pruebas automatizadas**

Casos de prueba automatizados con herramienta de automatización  
Work Product Kind: Productos centrales

Expand All Sections Collapse All Sections

**Purpose**  
Indica el listado de incidencias halladas en los procesos de prueba ejecutados.

**Relationships**

<b>Roles</b>	<b>Responsible</b> <ul style="list-style-type: none"><li>• Probador</li></ul>	<b>Modified By</b> <ul style="list-style-type: none"><li>• Probador</li></ul>
<b>Tasks</b>	<b>Input To</b> <ul style="list-style-type: none"><li>• Ejecucion casos de prueba</li></ul>	<b>Output From</b> <ul style="list-style-type: none"><li>• Automatizar pruebas</li></ul>

**Description**

**Main Description**  
Son pruebas automatizadas por herramientas específicas, complementa o se construyen de casos de prueba, para agilizar de forma incremental el proceso de prueba de los ideos y posibilitando el aumento de cobertura de pruebas.

**Illustrations**

**Templates**

- Biblioteca de pruebas
- Formato Casos de Prueba

**Key Considerations**

**More Information**

- Herramienta de automatización

Fuente: Elaboración propia

## Anexo 11 Artefacto reporte de incidencias

**Artifact: Reporte de incidencias**

Indica el contenido de incidencias halladas en los procesos de prueba  
Work Product Kind: Productos centrales

Expand All Sections Collapse All Sections

**Purpose**  
Indica el listado de incidencias halladas en los procesos de prueba ejecutados.

**Relationships**

<b>Roles</b>	<b>Responsible</b> <ul style="list-style-type: none"><li>• Probador</li></ul>	<b>Modified By</b> <ul style="list-style-type: none"><li>• Probador</li><li>• Usuario Probador</li></ul>
<b>Tasks</b>	<b>Input To</b>	<b>Output From</b> <ul style="list-style-type: none"><li>• Aplicación de heurísticas</li><li>• Ejecucion casos de prueba</li></ul>

**Description**

**Main Description**  
Es el listado que detallada los incidentes hallados en pruebas, cada incidente contiene información relevante para que el equipo de desarrollo pueda entender y solucionar la incidencia.

**Key Considerations**  
Debe expresar y explicar detalladamente el incidente y relacionar el caso de prueba que reproduzca la incidencia.

Fuente: Elaboración propia

## Anexo 12 Artefacto reporte de heurísticas

**Artifact: Reporte heurísticas**

Detalles de la aplicación de la evaluación de heurísticas  
Work Product Kind: Productos controlados

Expand All Sections Collapse All Sections

**Purpose**  
Reportar resultados de aplicación de evaluación de heurísticas

**Relationships**

<b>Roles</b>	<b>Responsible</b> <ul style="list-style-type: none"><li>Experto en Conocimiento</li><li>Probador</li><li>Usuario Probador</li></ul>	<b>Modified By</b> <ul style="list-style-type: none"><li>Experto en Conocimiento</li><li>Usuario Probador</li></ul>
<b>Tasks</b>	<b>Used To</b> <ul style="list-style-type: none"><li>Evaluación de Aprendizaje</li></ul>	<b>Output From</b> <ul style="list-style-type: none"><li>Aplicación de heurísticas</li><li>Evaluación de Aprendizaje</li></ul>

**Description**

**Main Description**  
Se detalla el resultado de la aplicación de heurísticas de usabilidad, experiencia de aprendizaje, experiencia de juego.

**Illustrations**

**Templates**

- Formato de medición de heurísticas

**Key Considerations**  
Se debe indicar que heurísticas se aplicaron y el contenido de los resultados.

Fuente: Elaboración propia

## Anexo 13 Tarea de aplicación de heurísticas

**Task: Aplicación de heurísticas**

Permite aplicar las heurísticas para medir experiencias de usuario y de aprendizaje

Expand All Sections Collapse All Sections

**Purpose**  
Evaluar criterios de calidad del videojuego Serio, tales como las experiencias de usuario y las experiencias de aprendizaje.

**Relationships**

<b>Categorías</b>	• Actividades del proceso	
<b>Roles</b>	<b>Primary Performer</b> <ul style="list-style-type: none"><li>Usuario Probador</li></ul>	<b>Additional Performer</b> <ul style="list-style-type: none"><li>Probador</li></ul>
<b>Inputs</b>	<b>Material</b> <ul style="list-style-type: none"><li>Casos de prueba</li></ul>	<b>Optional</b> <ul style="list-style-type: none"><li>None</li></ul>
<b>Outputs</b>	<ul style="list-style-type: none"><li>Reporte de incidencias</li><li>Reporte heurísticas</li></ul>	

**Steps**

**Selección de heurísticas**

- Definir características a medir (Experiencia de juego, usabilidad, experiencia de aprendizaje)
- Definir heurísticas a usar del Formato de medición de heurísticas.

**Immersion en el videojuego**

- Seleccionar usuario probador e iniciar videojuego serio
- Ejecución del videojuego serio (jugar videojuego)

**Evaluar probador con heurísticas**

- Aplicar heurísticas seleccionadas
- Realizar reporte de medición

Fuente: Elaboración propia

## Anexo 14 tarea de automatizar pruebas

**Task: Automatizar pruebas**

Permite definir la automatización de los casos de prueba que se consideren posibles y necesarios para la automatización.

Expand All Sections Collapse All Sections

**Purpose**

Permite generar cobertura de casos de prueba automatizados para ejecutarse de forma automática cuando lo considere el proceso o el probador.

Back to top

**Relationships**

<b>Categorías</b>	• Actividades del proceso	
<b>Roles</b>	<b>Primary Performer:</b> • Probador	<b>Additional Performer:</b>
<b>Inputs</b>	<b>Mandatory:</b> • Casos de prueba	<b>Optional:</b> • None
<b>Outputs</b>	• Pruebas automatizadas	

Back to top

**Steps**

Definir funcionalidad para generar cobertura

- Identificar viabilidad de automatización
- Seleccionar casos de prueba a automatizar

Automatizar Casos de prueba

- Definir herramienta de automatización
- Desarrollar automatización de casos de prueba
- Validar funcionamiento de automatización
- Liberar automatización

Fuente: Elaboración propia

## Anexo 15 Tarea de diseño de casos de prueba

**Task: Diseño de casos de prueba**

Permite definir los escenarios de prueba que se realizarán con valores de entrada y resultados esperados, para validar el funcionamiento del incremento del videojuego.

Expand All Sections Collapse All Sections

**Purpose**

Permite definir los escenarios de prueba que se realizarán con valores de entrada y resultados esperados, para validar el funcionamiento del incremento del videojuego.

Expand All Sections Collapse All Sections

**Relationships**

<b>Categorías</b>	• Actividades del proceso	
<b>Roles</b>	<b>Primary Performer:</b> • Probador	<b>Additional Performer:</b>
<b>Inputs</b>	<b>Mandatory:</b> • Requisitos y elementos de diseño	<b>Optional:</b> • None
<b>Outputs</b>	• Casos de prueba	

Back to top

**Steps**

Revisión de requisitos

- Identificar descripción de entradas, salidas, resultados esperados, procedimientos, acciones o cálculos.
- Retroalimentar o complementar información de ser necesario.

Escribir casos de prueba

- Identificar funcionalidad para escritura de casos de prueba
- Especificar y escribir caso de prueba usando Formato Casos de Prueba definido o (Escritura de caso de prueba construido por automatización)

Revisión de casos de prueba

- Observar cumplimiento de criterios de aceptación definidos en los casos de prueba
- Revisar cumplimiento del formato establecido

Fuente: Elaboración propia

## Anexo 16 Tarea de diseño evaluación de aprendizaje

**Task: Diseño evaluación de aprendizaje**

Permite diseñar instrumento cuestionario para evaluar el conocimiento adquirido por los usuarios.

Expand All Sections Collapse All Sections

**Purpose**

Evaluar si el usuario obtuvo el conocimiento esperado al usar el videojuego.

Back to top

**Relationships**

Categories	<ul style="list-style-type: none"> <li>Actividades del proceso</li> </ul>	
Roles	<b>Primary Performer</b> <ul style="list-style-type: none"> <li>Experto en Conocimiento</li> </ul>	<b>Additional Performer</b> <ul style="list-style-type: none"> <li>Probador</li> </ul>
Inputs	<b>Mandatory</b> <ul style="list-style-type: none"> <li>Casos de prueba</li> <li>Requisitos y elementos de diseño</li> </ul>	<b>Optional</b> <ul style="list-style-type: none"> <li>None</li> </ul>
Outputs	<ul style="list-style-type: none"> <li>Evaluación de aprendizaje</li> </ul>	

Back to top

**Steps**

Identificar el conocimiento a enseñar en el nivel, módulo del videojuego.

- Definir conocimiento de interés aplicado para enseñanza en el módulo de videojuego

---

Construir preguntas de medición de conocimiento

Definir instrumento evaluativo para verificar enseñanza del conocimiento esperado identificado

Fuente: Elaboración propia

## Anexo 17 Tarea de ejecución de casos de prueba

**Task: Ejecución casos de prueba**

Permite aplicar los casos de prueba diseñados o agregar otras pruebas definidas previamente en la biblioteca de pruebas.

Expand All Sections Collapse All Sections

**Purpose**

**Relationships**

Categories	<ul style="list-style-type: none"> <li>Actividades del proceso</li> </ul>	
Roles	<b>Primary Performer</b> <ul style="list-style-type: none"> <li>Probador</li> </ul>	<b>Additional Performer</b>
Inputs	<b>Mandatory</b> <ul style="list-style-type: none"> <li>Casos de prueba</li> <li>Incremento videojuego</li> <li>Pruebas automatizadas</li> </ul>	<b>Optional</b> <ul style="list-style-type: none"> <li>None</li> </ul>
Outputs	<ul style="list-style-type: none"> <li>Reporte de incidencias</li> </ul>	

Back to top

**Steps**

Definir casos a ejecutar

- Identificar incremento funcional del videojuego
- Seleccionar casos de prueba a ejecutar

---

Ejecutar casos de prueba

- Ejecutar pasos y acciones definidos en casos de prueba
- Verificar cumplimiento de resultados esperados

---

Reporte de incidencias

- Observar e identificar incumplimiento de criterios o resultados esperados
- Otorgar formato Reporte de incidencias con hallazgos
- Socializar reporte de incidencias con equipo

Fuente: Elaboración propia

## Anexo 18 Tarea de entrega de incremento

**Task: Entrega de incremento**

Permita realizar entrega del desarrollo del incremento funcional del videojuego para el proceso de prueba.

Expand All Sections Collapse All Sections

**Purpose**

Formalizar entrega de incremento funcional del videojuego para ser probado por el rol Tester.

Back to top

**Relationships**

Categories	• Actividades del proceso	
Roles	Primary Performer: • Desarrollador	Additional Performer:
Inputs	Mandatory: • Requisitos y elementos de diseño	Optional: • None
Outputs	• incremento_videojuego	

Back to top

Fuente: Elaboración propia

## Anexo 19 Tarea de evaluación de aprendizaje

**Task: Evaluación de Aprendizaje**

Aplicar cuestionario que mide conocimiento adquirido por el videojuego serio.

Expand All Sections Collapse All Sections

**Purpose**

Aplicar cuestionario o instrumento que mida el conocimiento adquirido por los usuarios del videojuego serio.

Back to top

**Relationships**

Categories	• Actividades del proceso	
Roles	Primary Performer: • Experto en Conocimiento • Usuario Probador	Additional Performer: • Probador
Inputs	Mandatory: • Evaluación de aprendizaje • Reporte heurísticas	Optional: • None
Outputs	• Reporte heurísticas	

Back to top

**Steps**

Aplicar evaluación de aprendizaje

- Aplicar cuestionario o instrumento a usuario del videojuego
- Revisar cuestionario aplicado en el usuario

Generar reporte de evaluación

- Identificar balanza o carencia de conocimiento en el usuario del videojuego
- Realizar propuesta de posible mejora
- Generar reporte con lo identificado en los puntos anteriores

Fuente: Elaboración propia





Anexo 22 Artefacto guía de aplicación de heurísticas parte 1

**Guía de aplicación de heurísticas**

A continuación, observaran el listado que contiene las heurísticas para evaluar usabilidad, jugabilidad, experiencia de aprendizaje y otras heurísticas. De acuerdo con su necesidad a evaluar y/o medir, podrá seleccionar las heurísticas a usar en la iteración de pruebas a ejecutar. Se recomienda aplicar el set completo de heurísticas en liberaciones de pruebas beta.

dimensión	Atributo	Elemento de encuesta	Valoración entre (0-10)
Experiencia de juego	Desafío	La experiencia fue desafiante, el juego me pareció estimulante. Escala: cero (0) poco estimulante - diez (10) muy estimulante	
		Los objetivos y desafíos del juego eran repetitivos. Escala: cero (0) muy repetitivos - diez (10) poco repetitivos	
	Competencia	Pude lograr los objetivos marcados en el juego. Escala: cero (0) No se lograron objetivos - diez (10) Se lograron los objetivos	
	Flujo	Me mantuve concentrado en el juego durante todo el proceso. Escala: cero (0) poca concentración - diez (10) mucha concentración	
	Inmersión	La experiencia fue inmersiva. Escala: cero (0) poca inmersión - diez (10) bastante inmersión.	
Experiencia de aprendizaje	Afecto	La experiencia en general fue positiva. Escala: cero (0) negativa - diez (10) positiva.	
	Tensión	Las interacciones en el juego fueron tensas. Escala: cero (0) muy tensas - diez (10) poco tensas.	
	Objetivo de aprendizaje	Los objetivos de aprendizaje del juego fueron claros. Escala: cero (0) poco claros - diez (10) muy claros.	
	Contenido adecuado	El escenario del juego tiene relevancia para el tema del desarrollo de las habilidades comunicativas. Escala: cero (0) relación baja - diez (10) relación alta.	
	Integración	El juego fue más educativo que entretenido. Escala: cero (0) más entretenido que educativo - diez (10) más educativo que entretenido.	
Usabilidad		La terminología que se usó en el juego se podía entender fácilmente. Escala: cero (0) difícil de entender - diez (10) fácil de entender.	
	Retroalimentación	El juego brinda oportunidades para recibir retroalimentación. Escala: cero (0) no hay retroalimentación - diez (10) Suficiente retroalimentación.	
	Extensibilidad	Reconozco el valor del juego como herramienta de aprendizaje. Escala: cero (0) no se reconoce como herramienta de aprendizaje - diez (10) se reconoce como herramienta de aprendizaje.	
	Ritmo del juego	El ritmo del juego fue demasiado rápido Escala: cero (0) muy rápido - diez (10) ritmo adecuado.	
	Interfaz	La interfaz de usuario fue fácil de usar Escala: cero (0) difícil de usar - diez (10) fácil de usar.	
	Interacción	Fue fácil iniciar el software Escala: cero (0) difícil de iniciar - diez (10) fácil de iniciar.	

Fuente: Elaboración propia

Anexo 23 Artefacto guía de aplicación de heurísticas parte 2

		Los controles del juego eran intuitivos y fáciles de usar. Escala: cero (0) poco intuitivos, difíciles de usar - diez (10) intuitivos, fáciles de usar.	
		Aprendió a usar el software rápidamente Escala: cero (0) Aprendizaje lento - diez (10) aprendizaje rápido.	
		Era fácil moverse Escala: cero (0) difícil moverse - diez (10) ritmo adecuado.	
		Era fácil entablar dialogo Escala: cero (0) difícil dialogar - diez (10) fácil dialogar.	

		Fue fácil responder a las preguntas dentro del juego Escala: cero (0) difícil responder - diez (10) fácil de responder.	
Fidelidad	Atractivo visual	El entorno de juego era visualmente atractivo. Escala: cero (0) poco atractivo - diez (10) muy atractivo.	
		Los personajes animados eran visualmente atractivos. Escala: cero (0) poco atractivos - diez (10) muy atractivos.	
		El tutorial del juego fue útil para aprender a jugar. Escala: cero (0) poco útil - diez (10) muy útil.	
TOTAL			<u>Sumatoria</u> resultado / Total heurísticas aplicadas

Tabla 1. Heurísticas usabilidad y aprendizaje adaptadas de Moizer et al. (2019) y de Xuan et al (2018)

De acuerdo con el resultado total de las heurísticas, se ubicará el valor en el nivel indicado a continuación

No deseable	Aceptable	Satisfactorio
3 <=	> 4	>= 7

Tabla 2. Niveles de evaluación de heurísticas

Se recomienda para mejorar la evaluación de heurísticas, atender aquellas que se encuentran en nivel **No deseable**, posteriormente las **Aceptable** hasta lograr tener un nivel de satisfactorio.

Fuente: Elaboración propia

### **Biblioteca de pruebas para Videojuegos Serios**

A continuación, se relaciona la descripción de diferentes tipos de prueba aplicadas a Videojuegos, estas podrán ser de utilidad y por tal razón se indican en esta biblioteca de pruebas, las siguientes no excluyen las pruebas conocidas para cualquier tipo de software (Unitarias, Integración, Sistema, Compatibilidad):

#### **Pruebas Alfa**

Las Pruebas Alfa son pruebas con jugadores-probadores, y responden también a pruebas que se realizan en alguna parte de las fases de pruebas, las pruebas Alfa según lo recopilado por algunos autores como Moiser et al. (2019) son pruebas internas que se realizan para identificar posibilidades de perfeccionamiento o como lo indica Prieto (2018) para corregir errores de código o en la funcionalidad (bugs).

#### **Pruebas Beta**

Si muy las Pruebas Beta hacen parte de las fases de prueba, tal como lo indican Hernández et al. (2017) estas pruebas toman relevancia ya que permite a jugadores-probadores revisar aspectos importantes como la jugabilidad y experiencia del usuario, estas pruebas en otras son pruebas que hacen usuarios humanos que juegan constantemente variando en algunos aspectos sus acciones, para lograr hallar errores, funcionamientos incorrectos o inesperados, estas pruebas requieren de tiempo y esfuerzo, acarreando altos costos. Estas pruebas suelen hacerse posterior a las pruebas alfa con un mayor nivel de integración de producto o de valor.

#### **Pruebas de Mono**

De acuerdo con lo recopilado por Hernández et al. (2017) estas pruebas son de caja negra, ya que en su concepto son pruebas que se ejecutan sin un orden establecido. Estas pruebas consisten en imaginar un Mono escribiendo entradas en el ordenador, buscando que se genere un bloqueo en el software y así lograr encontrar errores ocultos, estas pruebas pueden alimentar algoritmos evolutivos para hacer que las pruebas sean más eficientes o destructivas.

#### **Pruebas automatizadas de GUI**

Hernández et al. (2017) también recopilan y mencionan otro tipo de pruebas que podrían ser de utilidad para el desarrollo de videojuegos, en este caso si muy bien las pruebas automatizadas representan para el software clásico una gran forma para realizar pruebas, para los videojuego suele ser un problema por la variabilidad del entorno y la variabilidad de casos de prueba; aun así para los videojuegos basados en la nube o web, existen herramientas como Selenium y Appium, para realizar automatización en navegadores web i aplicaciones Android e IOs respectivamente.

### **Grabación de sesiones de juego para compatibilidad y regresión**

Hernández et al. (2017) describen estas pruebas como la posibilidad de almacenar, guardar las entradas de los jugadores beta, para posteriormente replicar o reproducir estas acciones en computadoras, es decir que en forma generar las sesiones de juego permiten escribir casos de prueba, que posteriormente serán ejecutados por computadora.

### **Pruebas de aprendizaje**

Tal como lo recopilan y mencionan Blanco et al. (2017) en otras revisiones de juegos serios, ninguno de los juegos completaba el proceso de validación, es decir que no realizaban una validación para el propósito de uso. Si muy bien los videojuegos tienen un propósito de entretenimiento, los Videojuegos Serios además de lo anterior tienen otro propósito, por lo anterior para problemas educativos específicos se deben integrar diseñadores y educadores para la validación del juego serio.

Además de involucrar educadores de un área en específico para validar el aprendizaje del juego, en algunas otras investigaciones Moiser et al. (2019), se definen pruebas en las que se aplican instrumentos (heurísticas) para medir la capacidad de enseñanza de los videojuegos (experiencia de aprendizaje). En este caso en los artefactos del proceso, se recopilan y proponen algunas heurísticas de utilidad para este caso en concreto.

### **Pruebas de experiencia de juego**

Las pruebas de experiencia de juego son pruebas que evalúan aspectos como la usabilidad, adaptabilidad y entre otras características de UX, en estas también se mide la experiencia de aprendizaje tal como lo propone Moiser et al. (2019), para estos casos se definen heurísticas que se aplican a usuarios finales con el fin de evaluar y validar los aspectos ya descritos, para el proceso en cuestión se recopiló y propone algunas heurísticas en un artefacto.

Fuente: Elaboración propia