

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

ESTUDIO DE METODOLOGIAS DE DESARROLLO DE SOFTWARE Y JUEGOS SERIOS PARA LA DEFINICION DE CRITERIOS DE DISEÑO DE JUEGOS SERIOS EDUCATIVOS

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

DANIEL MOISES CAJAS MONTUFAR

daniel.cajas@epn.edu.ec

ALEX JOSE FREIRE BUSTOS

alex.freire01@epn.edu.ec

DIRECTOR: MARCO SANTORUM, PHD.

marco.santorum@epn.edu.ec

CO-DIRECTOR: MAYRA CARRIÓN TORO, MSC.

mayra.carrion@epn.edu.ec

Quito, septiembre de 2019

APROBACION DEL DIRECTOR

Como director y codirectora del trabajo de titulación “Estudio de Metodologías de Desarrollo de Software y Juegos Serios para la Definición de Criterios de Diseño de Juegos Serios Educativos” desarrollado por Daniel Moisés Cajas Montúfar y Alex José Freire Bustos, estudiantes de la Facultad de Ingeniería de Sistemas Informáticos y de Computación, habiendo supervisado la realización de este trabajo y realizado las correcciones correspondientes, damos por aprobada la redacción final del documento escrito para que prosiga con los trámites correspondientes a la sustentación de la Defensa oral.

Mayra Carrión Toro, MSc.

CODIRECTOR DEL TRABAJO DE TITULACIÓN

Marco Santórum Gaibor, PhD.

DIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Nosotros, Daniel Moisés Cajas Montúfar y Alex José Freire Bustos, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Daniel Moisés Cajas Montúfar

Alex José Freire Bustos

AGRADECIMIENTO

A mis padres por el apoyo no solo en esta etapa, sino durante todo el transcurso de mi vida, además, por haberse convertido en el motor de lucha y superación diaria a pesar de todas las adversidades que hemos atravesado.

A mi compañero Alex, por el tiempo compartido durante la realización de este trabajo, por todo el tiempo y la dedicación empeñada en el mismo.

A mis directores del proyecto de investigación Marquito y Mayrita. Realmente el agradecimiento a ellos en esta etapa es invaluable, no solamente por el acompañamiento en la elaboración de este trabajo, sino por todo el apoyo, la confianza y la paciencia que depositaron en mí.

A mi amiga Imira, quien siempre me motivó para continuar y nunca desistir, para seguir luchando y lograr culminar el proyecto de investigación, por apoyarme en muchos ámbitos de mi vida.

Daniel Cajas

AGRADECIMIENTO

A mis padres, por haber participado activamente en mi formación personal y profesional a partir de mis primeros años de vida y nunca haber desistido.

A mi compañero de proyecto, Daniel Cajas, por su amistad, a pesar de todas las dificultades por las que tuvimos que pasar.

A los directores del presente proyecto de investigación y grandes amigos, Mayra Carrión y Marco Santórum, por su infinita paciencia y su apoyo incondicional, sin los cuales, no hubiera sido posible culminar este trabajo.

A todas y todos con los que he interactuado de distinta manera a través de mi vida; ya que, cada uno de ellos ha puesto su granito de arena para alcanzar este logro.

Alex Freire

ÍNDICE DE CONTENIDO

APROBACION DEL DIRECTOR.....	I
DECLARACIÓN DE AUTORÍA.....	II
AGRADECIMIENTO.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
LISTA DE FIGURAS	VII
LISTA DE TABLAS.....	VIII
LISTA DE ANEXOS DEL CD	XI
RESUMEN	XII
ABSTRACT	XIII
CAPITULO 1	1
1. INTRODUCCIÓN	1
1.1 Problema.....	2
1.2 Objetivos e Hipótesis.....	2
Objetivo general.....	2
Objetivos específicos.....	2
Hipótesis.....	2
1.3 Justificación y Alcance	2
1.4 Marco Teórico	3
1.4.1 Juegos como recurso pedagógico	3
1.4.2 Metodologías de desarrollo de software	5
Scrum.....	6
Programación Extrema.....	6
Crystal.....	8
Huddle	9
Sum.....	10
Game-Scrum.....	13
Extreme Game Development.....	13
Game Development Process	15
1.4.3 Metodologías de juegos serios.....	17
Proceso de Ágil para el Desarrollo de Juegos Serios (ASGDP)	17
Emergo	19

Meconesis	21
Barajas Saavedra	23
1.5 Conclusiones del Capítulo 1	24
1.6 Resumen del Capítulo 1	29
CAPITULO 2	30
2. METODOLOGÍA	30
2.1 Proceso General	31
2.1.1 Proceso Antecedente 1 (Opinión del investigador)	35
2.1.2 Proceso Antecedente 2 (Análisis Ponderado)	38
2.1.3 Proceso Consecuente (Obtención de Criterios clave)	43
2.2 Conclusiones del Capítulo 2	51
2.2 Resumen del Capítulo 2	52
CAPITULO 3	53
3. EJECUCIÓN, RESULTADOS Y DISCUSIÓN	53
3.1 Ejecución de la metodología	53
3.1.1 Enfoque 1	55
Proceso Antecedente 1 (Opinión del Investigador)	55
Proceso Antecedente 2 (Análisis ponderado)	57
Proceso Consecuente (Opinión del investigador)	58
Proceso Consecuente (Análisis Ponderado)	60
3.1.2 Enfoque 2	62
Proceso Antecedente 1 (Opinión del Investigador)	62
Proceso Antecedente 2 (Análisis Ponderado)	63
Proceso Consecuente (Opinión del Investigador)	64
Proceso Consecuente (Análisis Ponderado)	66
3.2. Resultados y Discusión	68
Enfoque 1	68
Enfoque 2	69
3.3 Conclusiones del Capítulo 3	70
3.4 Resumen Capítulo 3	71
CAPITULO 4	72
4. CONCLUSIONES	72
5. REFERENCIAS BIBLIOGRÁFICAS	73

LISTA DE FIGURAS

Figura 1 - Proceso de Desarrollo de XP	7
Figura 2 - Criticidad de Cristal	8
Figura 3 - Proceso de Desarrollo Huddle.....	9
Figura 4 - Proceso de Desarrollo de Sum.....	11
Figura 5 - Fases de un Proyecto Extreme Game Development	15
Figura 6 - Proceso de Desarrollo GDP	16
Figura 7 - Proceso de Desarrollo ASGDP	18
Figura 8 - Proceso de Desarrollo Emergo	20
Figura 9 - Proceso de Desarrollo Meconesis.....	22
Figura 10 - Proceso de Desarrollo Barajas Saavedra	23
Figura 11 - Diagrama de Flujo del Proceso General (PG).....	35
Figura 12 - Diagrama de Flujo del Proceso Antecedente 1	36
Figura 13 - Diagrama de Flujo del Proceso Antecedente 2	39
Figura 14 - Diagrama de Flujo del Proceso Consecuente	44
Figura 15 - Representación Gráfica de Criterios Necesarios y No Suficientes.	48
Figura 16 - Representación Gráfica de Criterios Necesarios y Suficientes	49
Figura 17 - Representación Gráfica de Criterios Suficientes y No Necesarios	50
Figura 18 - Representación Gráfica de Criterios No Suficientes y No Necesarios.....	50

LISTA DE TABLAS

Tabla 1 – Resumen de metodologías de desarrollo de juegos serios y de software	25
Tabla 2 - Codificación de las Tablas Usadas en la Metodología	32
Tabla 3 - Estructura de la Codificación	33
Tabla 4 - Conjunto de Documentos de las Metodologías (PG-E-CDM)	33
Tabla 5 - Conjunto de Criterios Base (PG-E-CCB)	34
Tabla 6 - Matriz de Justificación (PA1-A-MJ)	37
Tabla 7 - Matriz Booleana del Proceso Antecedente 1 (PA1-A-MB).....	37
Tabla 8 - Matriz de Análisis Reducida (PA2-A-MAR).....	39
Tabla 9 - Matriz de Frecuencia Absoluta (PA2-A-MFA)	40
Tabla 10 - Matriz de Frecuencia Normalizada (PA2-A-MFN).....	41
Tabla 11 - Matriz de Frecuencia Inversa (PA2-A-MFI).....	42
Tabla 12 - Matriz NTF-IDF (PA2-A-MNTFIDF).....	42
Tabla 13 - Matriz Booleana (PA2-A-MB).....	42
Tabla 14 - Matriz de Instanciación (PC-A-MI)	45
Tabla 15 - Matriz de Comportamiento (PC-A-MC)	45
Tabla 16 - Tabla de Patrones Lógicos	47
Tabla 17 - Matriz de Emparejamiento (PC-A-ME).....	51
Tabla 18 – Ejemplo - Conjunto de Documentos de las Metodologías (PG-E-CDM).....	53
Tabla 19 – Ejemplo – Conjunto de Criterios Base (PG-E-CCB)	55
Tabla 20 - Ejemplo Enfoque 1– Matriz de Justificación (PA1-A-MJ).....	56
Tabla 21 – Ejemplo Enfoque 1 - Matriz Booleana del Proceso Antecedente 1 (PA1-A-MB).	57
Tabla 22 - Ejemplo Enfoque 1- Matriz de Análisis Reducida (PA2-A-MAR)	57
Tabla 23 - Ejemplo Enfoque 1- Matriz de Frecuencia Absoluta (PA2-A-MFA)	57
Tabla 24 - Ejemplo Enfoque 1– Matriz NTF-IDF (PA2-A-MNTFIDF)	57
Tabla 25 - Ejemplo Enfoque 1– Matriz Booleana del Proceso Antecedente 2 (PA2-A-MB)	57
Tabla 26 - Ejemplo Enfoque 1– Matriz de Instanciación de la Opinión del Investigador (PC-A-MI).....	58
Tabla 27 - Ejemplo Enfoque 1– Matriz de Comportamiento de la Opinión del Investigador (PC-A-MC).....	58
Tabla 28 - Ejemplo Enfoque 1– Matriz de Emparejamiento de la Opinión del Investigador (PC-A-ME)	59

Tabla 29 – Ejemplo Enfoque 1- Conjunto de Criterios Clasificados por la Opinión del Investigador (PG-S-CCC).....	59
Tabla 30 - Ejemplo Enfoque 1 – Matriz de Instanciación del Método Ponderado (PC-A-MI).....	60
Tabla 31 - Ejemplo Enfoque 1– Matriz de Comportamiento del Método Ponderado (PC-A-MC).....	60
Tabla 32 - Ejemplo Enfoque 1– Matriz de Emparejamiento del Método Ponderado (PC-A-ME).....	61
Tabla 33 - Ejemplo Enfoque 1- Conjunto de Criterios Clasificados por el Método Ponderado (PG-S-CCC).....	61
Tabla 34 – Ejemplo Enfoque 2 – Matriz de Justificación (PA1-A-MJ).....	62
Tabla 35 – Ejemplo Enfoque 2 - Matriz Booleana del Proceso Antecedente 1 (PA1-A-MB).....	63
Tabla 36 - Ejemplo Enfoque 2 - Matriz de Análisis Reducida (PA2-A-MAR).....	63
Tabla 37 – Ejemplo Enfoque 2 - Matriz de Frecuencia Absoluta (PA2-A-MFA).....	63
Tabla 38 - Ejemplo Enfoque 2– Matriz de NTF-IDF (PA2-A-MNTFID).....	63
Tabla 39 – Ejemplo Enfoque 2 – Matriz Booleana del Proceso Antecedente 2 (PA-A-MB).....	64
Tabla 40 – Ejemplo Enfoque 2 – Matriz de Instanciación de la Opinión del Investigador (PC-A-MI).....	64
Tabla 41 - Ejemplo Enfoque 2 – Matriz de Comportamiento de la Opinión del Investigador (PC-A-MC).....	64
Tabla 42 - Ejemplo Enfoque 2– Matriz de Emparejamiento de la Opinión del Investigador (PC-A-ME).....	65
Tabla 43 – Ejemplo Enfoque 2- Conjunto de Criterios Clasificados por la Opinión del Investigador (PG-S-CCC).....	65
Tabla 44 - Ejemplo Enfoque 2 – Matriz de Instanciación del Método Ponderado (PC-A-MI).....	66
Tabla 45 - Ejemplo Enfoque 2– Matriz de Comportamiento del Método Ponderado (PC-A-MC).....	66
Tabla 46 - Ejemplo Enfoque 2 – Matriz de Emparejamiento del Método Ponderado (PC-A-ME).....	67
Tabla 47 - Ejemplo Enfoque 2- Conjunto de Criterios Clasificados por el Método Ponderado (PG-S-CCC).....	67
Tabla 48 - Validación de los Resultados del Enfoque 1 con la Varianza para los Criterios Necesarios o Suficientes.....	69

Tabla 49 - Validación de Resultados del Enfoque 2 con la Varianza para los Criterios Necesarios o Suficientes.	70
Tabla 50 – Relación entre los criterios y las metodologías de desarrollo de juegos serios	72

LISTA DE ANEXOS DEL CD

Anexo 1 – Documentos de Entrada de las Metodologías a Evaluar

Anexo 2 – Criterios Base de Entrada

Anexo 3 – Matrices de Justificación

Anexo 4 – Matriz de Análisis Reducida

Anexo 5 – Enfoque 1. Opinión del Investigador

Anexo 6 – Enfoque 1. Análisis Ponderado

Anexo 7 – Enfoque 2. Opinión del Investigador

Anexo 8 – Enfoque 2. Análisis Ponderado

Anexo 9 – Resultados del Enfoque 1 y 2

RESUMEN

En el presente trabajo se realiza un estudio de metodologías de desarrollo de software y juegos serios para determinar un conjunto de criterios relevantes en el diseño de juegos serios que faciliten la creación de nuevas metodologías de desarrollo en este campo. En primera instancia, se recurrió a una búsqueda sistemática de literatura de publicaciones científicas relacionadas a la temática de estudio para lo cual se determinó un conjunto de criterios de inclusión y exclusión. Posteriormente, el conjunto de metodologías de entrada fue sometida a varios análisis recurriendo al método cualitativo, llamado también como “opinión del investigador”, y el método cuantitativo, denominado “método ponderado”; los cuales, en el proceso de síntesis, convergen en un algoritmo que hace uso de lógica matemática de primer orden para su clasificación. Finalmente, en base al procedimiento anterior, se determinan qué criterios son necesarios y suficientes para el desarrollo de juegos serios.

PALABRAS CLAVE: videojuegos, juegos serios, juegos educativos, metodologías de desarrollo de software.

ABSTRACT

In the present work, a study of software development and serious games methodologies is done to determine a set of relevant criteria in the design of serious games that facilitates the creation of new development methodologies in this field. In the first instance, a systematic literature search of scientific publications related to the subject of study was used for which a set of inclusion and exclusion criteria was determined. Subsequently, the set of input methodologies was subjected to several analyzes using the qualitative method, also called "researcher opinion", and the quantitative method, called "weighted method"; which, in the synthesis process, converge in an algorithm that uses first-order mathematical logic for classification. Finally, based on the previous procedure, the necessary and sufficient criteria is established for the development of serious games.

KEYWORDS: video games, serious games, educational games, software development methodologies.

CAPITULO 1

1. INTRODUCCIÓN

Las Tecnologías de la Información (TI) han incursionado en casi todos los ámbitos de la vida, por lo tanto, no es extraño encontrar el uso de juegos serios como recurso pedagógico de apoyo en el proceso de enseñanza y aprendizaje. En particular, la utilización en el aula de clase de técnicas basadas en juegos serios, con el objetivo de favorecer el desarrollo de habilidades y destrezas, ya que estos permiten aprovechar los aspectos motivacionales que generan los juegos de video.

No obstante, el campo de estudio aún no ha sido abordado en su totalidad o al menos, de manera que se establezca una serie de criterios a considerar a la hora de diseñar y desarrollar juegos serios.

Uno de los principales inconvenientes es asegurarse de que el juego cumpla con los objetivos de aprendizaje en el campo que se desea fortalecer, pero sin olvidar el aspecto de entretenimiento.

Para diseñar juegos serios educativos que cumplan satisfactoriamente con su objetivo lúdico y pedagógico, es necesario previamente conocer cuáles son los aspectos fundamentales que deben ser considerados en su diseño para obtener un producto de calidad.

En este estudio, se plantea la posibilidad de solucionar la ausencia de estudios que hagan una comparación entre las metodologías de desarrollo de software y el diseño de juegos serios.

Uno de los caminos para lograrlo es a través de estudios comparativos, que permitan identificar las similitudes, diferencias y patrones entre los siguientes aspectos: las metodologías de desarrollo de software, sobre las cuales ya existe un fuerte fundamento teórico y conceptual; y las metodologías de desarrollo de juegos serios, que son una tendencia emergente donde todavía hay mucho por investigar.

Por ello la importancia de realizar un estudio de metodologías de desarrollo software y juegos serios que permitan visualizar los criterios claves utilizados para su construcción.

Este estudio, en proyectos futuros, servirá como base para establecer una serie de criterios claves para concebir y diseñar juegos serios educativos.

El documento se encuentra organizado de la siguiente manera: el Capítulo 1 presenta la problemática, objetivo general, objetivos específicos, hipótesis, justificación alcance y marco teórico; el Capítulo 2 se refiere a la metodología empleada; en el Capítulo 3 se detallan los resultados y un análisis de los mismos como parte de la discusión; y finalmente se exponen las conclusiones y el trabajo futuro.

1.1 Problema

En algunos ámbitos educativos se puede notar que aún se utilizan técnicas basadas en la transmisión de conocimientos de manera pasiva, por ejemplo, profesores que hacen uso solamente de técnicas de enseñanza con un marcador, pizarra y borrador, técnica educativa muy útil, pero que en muchos casos genera falta de motivación en los estudiantes y hasta en el mismo profesor.

Así también, existen estudios que afirman que, sin un correcto diseño del juego serio, el resultado puede ser negativo, por ejemplo, en (Zea, 2011), el autor considera que existen juegos serios con fines educativos que no motivan al estudiante en su aprendizaje, debido a que no permiten que los estudiantes experimenten un proceso de inmersión y motivación similar al que experimentan con los videojuegos comerciales, en algunos casos, los mismos usuarios han manifestado que desean una mejor historia o escenario.

Este inconveniente se ve inmiscuido en casi todas las metodologías de desarrollo de software, ya que si bien, muchas de ellas, sobre todo aquellas que siguen los principios ágiles, son amigables y adaptables a todo tipo de desarrollo de sistemas no detallan en su diseño, patrones que se enfoquen en el desarrollo de juegos serios, inclusive, no se cuenta con mucha información sobre metodologías orientadas específicamente a juegos serios, lo que permite abrir una gran brecha de investigaciones.

1.2 Objetivos e Hipótesis

Objetivo general

Identificar criterios clave para la creación de juegos serios educativos a partir del estudio de metodologías de desarrollo de software y juegos serios.

Objetivos específicos

- Estudiar el marco de referencia para la caracterización de las metodologías de desarrollo de software y de juegos serios.
- Analizar metodologías de desarrollo de software y de juegos serios con el fin de identificar patrones característicos distintivos.
- Inferir criterios para la elaboración de un perfil para la creación de una metodología de desarrollo de juegos serios.

Hipótesis

A partir de un estudio y un análisis documental, se pueden determinar patrones característicos distintivos de metodologías en juegos serios y así inferir criterios clave para la creación de juegos serios educativos.

1.3 Justificación y Alcance

El desarrollo de un juego serio, habitualmente se puede apoyar en marcos de trabajo o metodologías que permiten estructurar, planear y controlar dicho proceso, es decir, el uso de metodologías es importante, ya que, mediante un correcto análisis y diseño de la aplicación, se podría garantizar la satisfacción de los requerimientos del usuario, y más aún si se estudia metodologías propias de los juegos serios.

Es así que en Morales, Nava, Fernández y Rey (2010), los autores presentan algunos ejemplos de metodologías tradicionales, tales como: Waterfall Process, Essential Unified, Team Software Process, Scrum Framework, entre otros, que ayudan a guiar el proceso de desarrollo de juegos de video, permitiendo así verificar los criterios que éstas brindan al momento de implementar dichos juegos.

Por otro lado, existen autores que proponen metodologías específicas para el diseño de juegos serios (Mader, 2015; Tran, George & Marfisi-Schottman, 2010), las cuales detallan a su vez, aspectos necesarios a ser tomados en cuenta a la hora del desarrollo. Las metodologías ADGDP, Emergo, Meconesis y Barajas-Saavedra son algunos ejemplos.

El presente estudio propone la elaboración de un perfil de criterios para que facilite, a futuro, la creación de metodologías de diseño de juegos serios.

Los beneficiarios serán los desarrolladores, especialistas en Ingeniería de Software y profesionales del campo educativo, quienes podrán realizar sus respectivas actividades de manera eficaz y eficiente, además de lograr que el producto de software satisfaga las necesidades del cliente en conjunto con el cumplimiento de los objetivos lúdicos y pedagógicos.

Adicionalmente, esta investigación permitirá conocer las características y criterios de cada una de las metodologías estudiadas, permitiendo con ello, mejorar la concepción de lo que significa el campo de los juegos serios educativos, adaptarlos de mejor manera a las necesidades de los usuarios y clientes, y por ende expandir la utilidad de aplicar metodologías adecuadas a la hora de diseñar un juego serio, evitando proponer productos que no generen los beneficios para los que inicialmente fueron pensados, todo esto enmarcado en la línea de sistemas de información educativos.

1.4 Marco Teórico

En la siguiente sección, se puntualizan conceptos relevantes afines a la temática de estudio, y que permitirán marcar una diferenciación entre cada uno de ellos, para luego continuar con el estudio de las metodologías de desarrollo de software y de videojuegos, tomando en cuenta ciertas características importantes que servirán de punto de partida para los estudios posteriores.

Los juegos serios cada vez toman más fuerza en muchos de los planes programáticos educativos del aprendizaje escolar, incluso en entornos de personas con impedimentos auditivos, visuales, entre otros, por tal motivo se analizará a los videojuegos dentro del contexto pedagógico, así como las metodologías que se han venido utilizando como parte del desarrollo ágil y finalmente las metodologías que han sido pensadas propiamente en el desarrollo de juegos serios.

1.4.1 Juegos como recurso pedagógico

En el ámbito educativo, la evolución que se ha evidenciado en los procesos de enseñanza, conjuntamente con los avances tecnológicos ha ido constantemente en aumento.

Desde el hecho de que, en el pasado, y no hace muchos años, los métodos de enseñanza-aprendizaje consistían en marcadores de tiza líquida o tizas blancas, pizarras y algunas carteleras demostrativas de lado del maestro y cuadernos, hojas de papel, lápices y esferos del lado del estudiante.

En la actualidad, la cantidad de recursos con los que se cuenta para mejorar estos antiguos procedimientos han permitido resurgir en nuevas ideas de enseñanza.

El emplear el simple concepto de un juego, en cualquier contexto pedagógico, hace que un estudiante se muestre activo e interesado en la temática de estudio que se está abordando, sin embargo, no solo el contexto escolar es el único beneficiado, existen muchas otras tendencias que ahora utilizan los juegos serios para sus fines educativos.

A continuación, se detallan términos claves para el estudio, permitiendo de esta manera establecer un enfoque más claro de la investigación.

Juegos (Games)

Son sistemas formales de reglas que definen y a su vez restringen las acciones y el comportamiento del jugador. Un juego debe tener uno o varios objetivos, procedimientos y mecánicas que deben cumplirse. Adicionalmente, involucran experiencias emocionales que desafían a los jugadores a alcanzar sus objetivos (Ruiz-Rodríguez & Fernández-y-Fernández, 2007).

Juegos de video (video games)

Un videojuego se define como una aplicación de software en la cual uno o más jugadores toman decisiones controlando objetos y recursos del juego, en la búsqueda de su objetivo (Aleem, Capretz, & Ahmed, 2016).

Son aplicaciones de software que se instalan en dispositivos hardware como consolas de videojuegos, computadoras, dispositivos de mano y asistentes digitales personales (Aleem et al., 2016).

Zyda (2005) define a un videojuego como un concurso físico o mental, jugado acorde a un conjunto preestablecido de reglas específicas, con el objetivo de entretener o recompensar a los participantes. Además, lo establece como un conjunto de tres elementos básicos como historia, arte y software, concepto tomado de un diseñador de videojuegos, Bing Gordon, siendo la historia el escenario del videojuego, arte el aspecto estético visual, y software, referente al código con el cual va a ser desarrollado o plasmada la historia presentada en el videojuego.

Juegos Serios (serious game)

Al hablar de Serious Game (SG) algunos autores presentan sus definiciones, una de las más utilizadas por los investigadores en esta área es la propuesta por Zyda en su trabajo titulado *“From visual simulation reality to games”*, que se refiere a un concurso mental, jugado con una computadora de acuerdo a un conjunto de reglas específicas, que emplea el entretenimiento para promover la capacitación gubernamental o corporativa, educación, salud, política pública y objetivos estratégicos de comunicación (Zyda, 2005).

Así también Zyda (2005) explica que a más tres elementos que se considera en el videojuego (historia, arte y software) se toma en cuenta la parte seria cuyo aspecto es el pedagógico.

Por otro lado, (Gómez, 2007), describe a un (SG) como una herramienta de enseñanza-aprendizaje que emplea objetos de conocimientos, el cual involucra en su naturaleza y en

su uso, objetivos pedagógicos, didácticos y lúdicos que promueven el desarrollo de habilidades y aptitudes relacionadas con tareas profesionales específicas.

Software Educativo

Es un conjunto de programas cuyo objetivo principal es el de educar en algún ámbito específico o entrenar al usuario en alguna actividad específica (Acerenza et al., 2009).

Generalmente este tipo de software está desarrollado con fines específicos para un propósito, respondiendo a alguna problemática o temática que se quiera atacar.

En el campo del desarrollo de software en general, existe una gran cantidad de metodologías, procesos y frameworks de desarrollo que son utilizados para llevar a cabo la producción de varios sistemas o software funcionales, sin embargo, para seleccionar una metodología adecuada, se recomienda hacer un estudio de la naturaleza del proyecto (Duran & Fajardo, 2012).

A continuación, se detallan algunas de las metodologías de desarrollo de software.

1.4.2 Metodologías de desarrollo de software

Las metodologías de desarrollo de software son usadas para la planificación, estructuración y control del desarrollo de software, utilizando herramientas y mecanismos de buenas prácticas (Jiménez, 2012).

En el estudio fueron considerados algunos procesos ágiles y tradicionales para conformar el grupo de las metodologías de desarrollo de software, sin embargo, en la siguiente sección se estudiarán únicamente metodologías ágiles debido a las mayores facilidades que presenta en su estructura, sus prácticas y valores a la hora de desarrollar un videojuego.

Además, se describirán algunas metodologías específicas para el diseño de juegos serios, que son de utilidad para el análisis posterior.

Metodologías ágiles

Nacieron con la finalidad de superar las debilidades reales percibidas de la ingeniería de software convencional, como el exceso de documentación, el uso de una gran cantidad de procesos y herramientas en el diseño y desarrollo, la negociación apegada a un contrato formal y que debía respetarse y además el estricto cumplimiento de un plan para todo el curso del proyecto. Su enfoque apunta hacia la satisfacción del cliente, la entrega frecuente de software incremental, equipos pequeños, métodos informales, y la comunicación continua entre el cliente y los desarrolladores (Pressman & Maxim, 2019).

En el campo de los videojuegos e incluso algunos juegos serios, la mayoría de los que se han desarrollado, utilizan procesos de desarrollo ágiles, ya que presentan facilidades de adaptación en cuanto a tiempos, cronogramas y entregables, además de que cuentan con una alta participación del cliente.

La mayor fortaleza que presentan, es el fomento de la innovación, es decir estar abierto a los cambios, a mejoras, a proponer ideas y soluciones que beneficien el desarrollo de los videojuegos y sobre todo que satisfagan las necesidades del cliente a cabalidad.

En esta sección se describen algunas de las metodologías ágiles estudiadas, con sus respectivos ciclos de desarrollo.

Scrum

Scrum es un marco de trabajo que aborda problemas complejos adaptativos en el cual se pueden emplear varios procesos y técnicas (Schwaber & Sutherland, 2013).

Una de las fortalezas de este framework es que se alimenta con un enfoque interactivo e incremental para controlar de mejor manera los riesgos que aparezcan, además de que está basado en la teoría de control de procesos empírica, es decir en la experiencia y reforzar sobre aquello que ya se conoce.

En realidad, el éxito de Scrum es precisamente el trabajar como un todo, es decir empleando los equipos de Scrum, sus roles, eventos, artefactos y reglas asociadas, todo lo demás no es Scrum.

Por las facilidades y por los beneficios que presenta en el aspecto organizacional y en la gestión del producto, muchas de las prácticas son tomadas para otras metodologías y sobre todo en este campo de estudio, es decir el desarrollo de videojuegos.

Si se habla de una adaptación de Scrum al ámbito de estudio de videojuegos, las fases se detallan en tres, ya antes mencionadas: pre-juego, juego, post-juego (Orjuela & Rojas, 2008).

- Pre-juego: Esta fase está subdividida en dos; la fase de planificación y la fase de arquitectura. En la planificación se debe tener una lista de actividades por realizar conocido como “*backlog*”, un costo estimado y un cronograma.

Así mismo cuando el sistema haya iniciado, existe una etapa de conceptualización de ideas y después, una etapa de análisis.

En la fase de arquitectura se aplica un diseño de alto nivel, además de la creación de la arquitectura del sistema.

- Juego: Es el momento en donde se da paso al desarrollo de cada sprint¹. Nunca se pierde de vista aspectos como el tiempo, la calidad de los requerimientos, costos, entre otros.
- Post-juego: Existe una sub-fase conocida con el nombre de clausura. Es la etapa en donde se ajustan los últimos detalles para la entrega del software, añadiendo la documentación final, las respectivas pruebas y finalmente las entregas a los usuarios finales.

Programación Extrema

Como parte de los procesos ágiles, surge programación extrema (XP), siendo uno de los enfoques más utilizados dentro del desarrollo de software ágil. Sustenta su fundamento en

¹ Unidad de trabajo que permite llevar a cabo un requerimiento previamente definido.

cinco valores: comunicación, simplicidad, retroalimentación, valentía y respeto (Pressman & Maxim, 2019).

De los aspectos más importantes que considera esta metodología, es la programación en pares, las pruebas de aceptación, además se puede aclarar que es ideal si se tiene requisitos imprecisos y muy cambiantes, otra característica es la refactorización del código (Orjuela & Rojas, 2008).

El proceso de desarrollo está compuesto de cuatro actividades estructuradas: planeación, diseño, codificación y pruebas. Cada una de estas fases tiene un objetivo específico en el desarrollo y además involucra características y recursos propios (ver Figura 1).

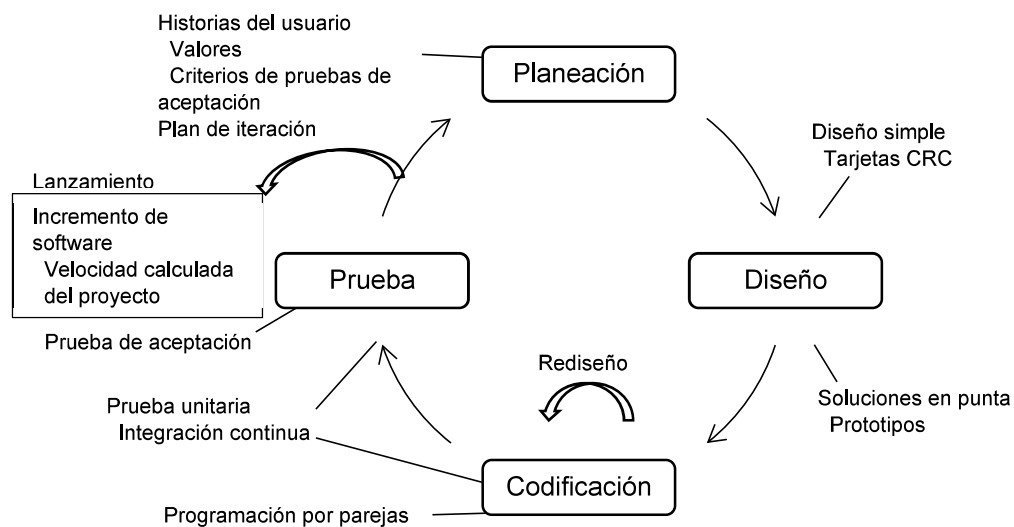


Figura 1 - Proceso de Desarrollo de XP
(Pressman & Maxim, 2019)

Fases del Proceso XP

- **Planeación:** Se define la idea de negocio del software y su funcionalidad. Los requerimientos son atendidos mediante historias de usuario, las mismas que son escritas por los clientes y colocadas en tarjetas.

De igual manera son los clientes los que asignan prioridades de implementación a estas historias de usuario.

Es importante la participación conjunta entre clientes y desarrolladores, ambos trabajan en la designación y agrupación de las historias de usuario que se implementarán para la siguiente entrega.

- **Diseño:** El diseño guiará la implementación de una historia de usuario. El principio que guía esta fase es hacerlo lo más sencillo posible, evitando sobrecargarlo de funcionalidad excesiva.

El artefacto que se utiliza en este proceso son las tarjetas CRC (clase-responsabilidad-colaborador) que facilita la concepción de una programación orientada a objetos.

- Codificación: El equipo desarrolla aquello que está reflejado en la historia de usuario y lo somete a pruebas unitarias que permite tener una ventaja sobre lo que debe implementar para pasar la prueba.

Una de las fortalezas de XP es la programación en parejas, que aporta algunos beneficios en el desarrollo de software como el brindar soluciones a problemas en tiempo real, asegurarse de que el código respete los lineamientos y estándares de codificación e incluso, cerciorarse de aminorar los errores que se puedan presentar en las pruebas unitarias. En otras palabras, es primordial garantizar la calidad del código.

- Pruebas: Cuando se ha pasado por las pruebas unitarias se procede con las pruebas de regresión y así lograr identificar anomalías a partir de lo que se modificó en las pruebas unitarias. Se da paso a las pruebas de integración y validación del sistema, dejando a veces al par de programadores a cargo de esto o a un grupo particular de desarrolladores.

Al final de todo el proceso, están las pruebas de aceptación, que son especificadas por el cliente. Consisten en una constatación de lo detallado en las historias de usuario y la funcionalidad que se presenta por parte del grupo de desarrollo.

Crystal

Son un conjunto de metodologías para el desarrollo de software, las personas, su cooperación, su comunicación e invención junto con la reducción al máximo del número de artefactos producidos son las principales características que las distinguen del resto de metodologías ágiles (Orjuela & Rojas, 2008).

Crystal basa su clasificación de acuerdo a dos enfoques o ejes:

- el número de personas en el proyecto; y
- la consecuencia de los errores (ver Figura 2).

Por esa razón surgen la familia de metodologías Crystal distinguidas por colores: Crystal Clear, Cristal Yellow, Crystal Orange, Crystal Blue, entre otras.

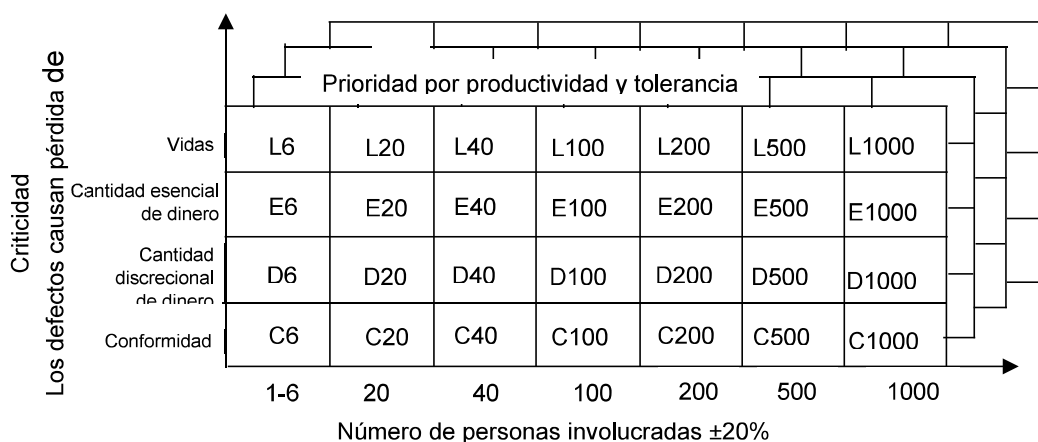


Figura 2 - Criticidad de Cristal
(Orjuela & Rojas, 2008)

Crystal no especifica un ciclo de desarrollo a seguir, permitiendo adoptar cualquier modelo de procesos o ciclo de desarrollo, pero sí propone una serie de lineamientos, políticas, propiedades, prácticas y roles que deberán ser seguidos a cabalidad.

Huddle

Es un proceso de desarrollo que está basado y fundamentado en Scrum, por ende, muchas de sus prácticas, recursos, roles y reuniones están sustentadas sobre Scrum, con la diferencia de que está orientado al desarrollo de videojuegos (Morales et al., 2010).

Su proceso de desarrollo consta de tres fases, las mismas que serán visualizadas en otras metodologías posteriormente (ver Figura 3).

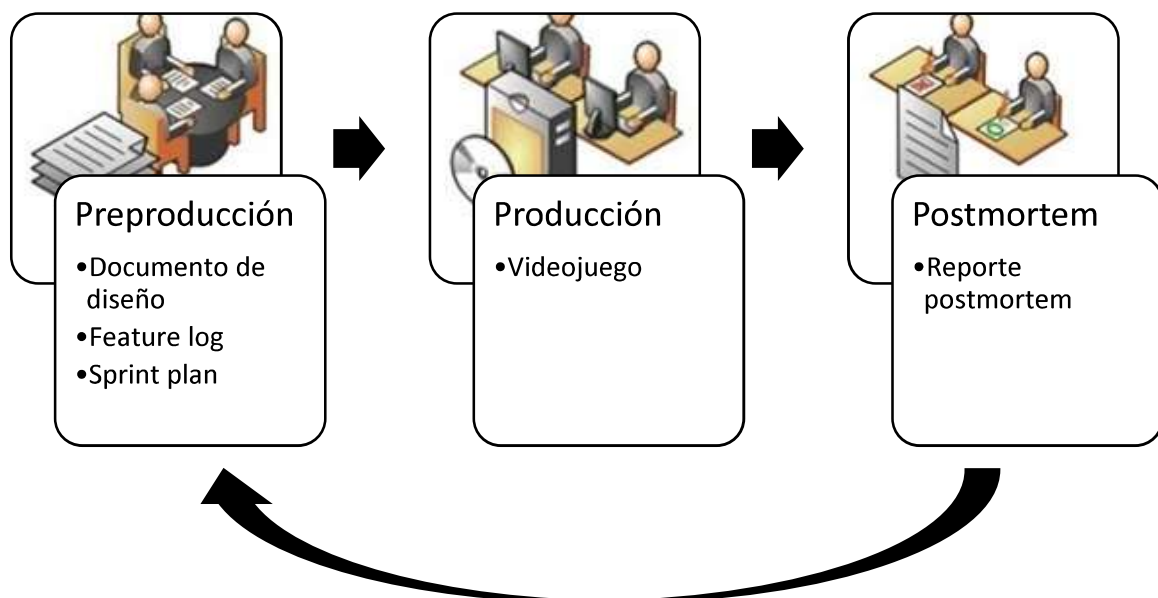


Figura 3 - Proceso de Desarrollo Huddle
(Morales et al., 2010)

- **Preproducción:** En un inicio se realiza un análisis del proyecto, donde se acepta o rechaza la idea inicial, de ser aceptada, se procede con la planificación de toda la fase de producción.

En esta fase resalta el *“Documento de Diseño”*, puesto que es aquí en donde se detallará la idea principal del videojuego de manera formal, además de otras particularidades de la propuesta. Cuando este artefacto no es aprobado, se lo debe modificar nuevamente hasta poder ser aprobado.

Si el artefacto es aprobado, se continúa con el *“Project Huddle”*, en el cual se debe contar con todas las partes interesadas en el proyecto, ya que aquí se realiza una planificación a futuro para la fase de producción, características del videojuego y tiempos estimados de cada sprint.

Finalmente, la idea del diseñador reposará sobre otros artefactos como son el *“Feature Log”* y el *“Sprint Plan”*.

- Producción: En esta fase se lleva a cabo toda la planificación anterior, es decir poner en marcha los diferentes requerimientos, e ir adoptando nuevos durante los sprints. La fase de producción se da inicio con el primer sprint, con el inicio del desarrollo del videojuego.

Es determinante en este punto realizar un *“Sprint Huddle”*, en el cual se reúnen todos los miembros del equipo, para analizar cada uno de los requerimientos, de esta manera se les asigna una prioridad para el proceso de desarrollo *“Sprint Backlog”*.

Cuando se ha cumplido el objetivo de un sprint, se continúa con las pruebas alfa, las cuales permiten analizar las características de lo que se implementó, además de detectar errores en la codificación e integración de recursos.

Una vez que se concluyó con la implementación de todos los sprints y sus respectivas pruebas alfa, se llega a obtener una versión beta del videojuego. Los miembros del equipo de desarrolladores se encargarán de realizar las pruebas finales sobre esta versión y así concluir con un producto final o *“Gold Master”*.

- Postmortem: Como resultado de todo el proceso de desarrollo se procede a establecer un reporte en el cual se detallan las actividades más importantes para el proyecto, tanto aquellas más efectivas, así como aquellas en las cuales se tuvo problemas, el objetivo es evitar lidiar con estos problemas nuevamente.

Una de las particularidades de este proceso es que en el artefacto *“Documento de Diseño”* si bien, hay temáticas generales que abarcan todos los juegos, hay otros que son particulares para cada juego como la jugabilidad, escenario y recursos.

Sum

Cuando se habla del desarrollo de videojuegos, normalmente el tiempo que se emplea para poner en producción un videojuego tiene una duración mayor a un año (Morales et al., 2010) . Sum es una metodología orientada al diseño de videojuegos que permite reducir ese tiempo.

Muchas de las características de esta metodología se derivan del común de las metodologías de desarrollo de software, sin embargo si presenta adecuaciones importantes, por ejemplo Acerenza et al., 2009):

- Existe una amplia participación del cliente
- Los principios de la metodología están fundamentados en SPEM (Software and Systems Process Engineering Metamodel Specification), con la ventaja de que no se mencionan prácticas específicas a seguir, en otras palabras, todo dependerá del proyecto que se realizará.
- Es flexible y adaptable respondiendo a las necesidades del proyecto, incluso puede ser combinada con otras metodologías.

Dentro de los roles y concretamente dentro del equipo de desarrollo, se cuenta con otros específicos, enfocados directamente al desarrollo de videojuegos, como artistas gráficos, artistas sonoros, diseñadores del juego.

El ciclo de vida está compuesto por 5 fases secuenciales y una transversal a todas ellas (ver Figura 4), estas fases son iterativas e incrementales. Si se intenta hacer una comparación con Scrum, Sum adapta sus fases a pre-juego, juego y post-juego.

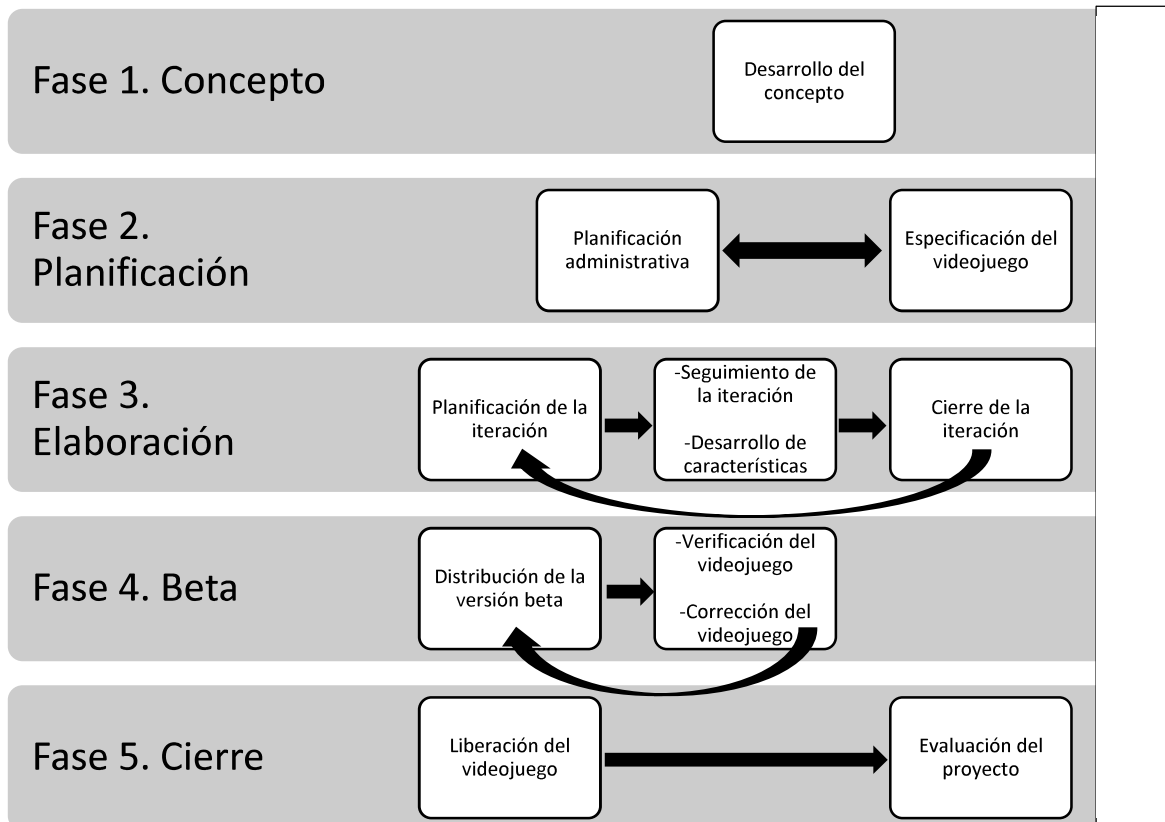


Figura 4 - Proceso de Desarrollo de Sum
(Acerenza et al., Mesa, et al., 2009)

Las fases del ciclo de vida se sintetizan a continuación:

- Fase 1 o Concepto: En esta fase se define la idea del juego, en donde se trata, además, aspectos de negocio, elementos del juego y otros aspectos técnicos.

Es propuesto por todos los roles involucrados en el proyecto, mediante reuniones y pruebas de concepto. Esta fase se da por finalizada, cuando el concepto del juego finalmente es aceptado y aprobado por todo el equipo.

- Fase 2 o Planificación: Se planifica las fases restantes del ciclo de desarrollo. Es vital elaborar un cronograma para esclarecer las actividades que se deben realizar. Adicionalmente se determina el equipo de desarrollo para la fase de elaboración.

Se necesita definir las características funcionales y no funcionales que tendrá el videojuego, acorde un presupuesto. Esta fase es flexible, puesto que está sujeto a cambios durante el avance del proyecto.

- Fase 3 o Elaboración: En esta fase se desarrolla el juego produciendo versiones ejecutables al final de cada iteración.

A su vez, esta fase se compone de tres sub-fases, en la primera, se definen los objetivos de implementación, las métricas de evaluación, tiempos de iteración, y las tareas necesarias para llevar a cabo dicha iteración.

En la segunda, se desarrollan las tareas descritas en la fase anterior, pero además se guarda un control de la iteración que permite comparar los objetivos planteados con lo que se ha logrado implementar.

En la tercera, se evalúa lo que se ha implementado con los objetivos trazados, siendo más fácil poder realizar cambios y tomar decisiones sobre el rumbo del proyecto.

- Fase 4 o Beta: Consiste en evaluar aspectos propios del videojuego, como por ejemplo, la jugabilidad², factor diversión, curva de aprendizaje, curva de dificultad y eliminar errores que se hayan detectado en fases anteriores.

Una versión beta, permite al equipo de desarrollo poder descubrir errores, fallas y enviar reportes de evaluación. Estos reportes son importantes para ser tomados en cuenta en las siguientes versiones del juego.

La fase termina cuando se ha logrado cubrir todas las novedades de los reportes en las distintas versiones del videojuego.

- Fase 5 o Cierre: Cuando se cuenta con la versión final y estable del videojuego se procede a entregarlo al usuario final, de acuerdo a los contratos y formatos establecido. Para el grupo de desarrollo es determinante discutir y estudiar todos los inconvenientes y problemáticas identificados y las soluciones propuestas. En otras palabras, se analiza las situaciones positivas y negativas que dejó la experiencia y se plantea mejoras de acuerdo a un proceso de retroalimentación.
- Gestión de Riesgos: Es una fase transversal a todas las fases descritas anteriormente y existe con el objetivo de poder mitigar y minimizar al máximo el impacto de problemas.

Para cada fase, se debe identificar probabilidades e impactos de ocurrencia, establecer mecanismos de monitoreo y elaborar estrategias de mitigación y planes de contingencia.

² Se refiere al intercambio continuo que se da entre la entrada del jugador y la respuesta del juego como un sistema cibernético.

Game-Scrum

Game-Scrum es una metodología que se nutre de otras dos, como es el caso de Scrum y de Extreme Programming, XP. La particularidad de esta metodología va más allá de la preocupación de lo referente al proceso de desarrollo en sí, para lo cual utiliza las prácticas de XP, sino que dirige sus esfuerzos hacia la gestión del proyecto y permite que todas las tareas que serán tratadas posteriormente sean consideradas y atendidas sin impedimentos (Barbosa & Godoy, 2010).

El proceso de desarrollo se destaca en tres fases: pre-producción, producción y post-producción.

- Fase de pre-producción: En esta fase se definen los objetivos del videojuego y constantemente el factor diversión.

Se utiliza el brainstorming, en el desarrollo de nuevas ideas, en la búsqueda del concepto y diseño del videojuego, además se planifica la fase de producción, con la ayuda de prototipos se permiten la visualización del factor diversión y las características requeridas para las pruebas.

Un artefacto determinante es el "*Documento de Diseño del Videojuego*", en el cual se describen cada uno de los aspectos, objetos, roles, comportamiento e interacciones del videojuego. Este documento incluye el desarrollo y pruebas del ciclo de desarrollo.

- Fase de producción: Se debe tener claro el alcance del proyecto. El documento de diseño debe traducirse en un backlog de productos. En cada iteración, los aspectos más importantes se dividen en pedazos más pequeños.
- Fase de post-producción: Cuando el juego haya sido desarrollado en la fase anterior, se procede con las pruebas correspondientes para garantizar la calidad del mismo y además la temática de diversión.

Posteriormente, se realizan postmortems para dar a conocer las experiencias, fortalezas, debilidades en el proceso de desarrollo utilizado, problemas ocurridos y sugerencias para mejorar.

Posiblemente Game-Scrum no brinde mayores aportes a la hora de desarrollar el videojuego, pero una de sus fortalezas es la de involucrar a todo el equipo del proyecto con la colaboración de sugerencias en torno a un mismo objetivo (Barbosa & Godoy, 2010).

Extreme Game Development

Es una metodología desarrollada y propuesta por la empresa Titus Interactive Studio. En esta metodología participan todo el equipo del proyecto, desde gerentes, productores, artistas, diseñadores y programadores (Barreno & Guaraca, 2013).

Es amigable, puesto que todas las actividades que se deben realizar durante el proceso de desarrollo son gestionadas por cada miembro del equipo de trabajo, permitiendo tener cierta libertad con el fin de pensar siempre en mejorar el juego.

Es una adaptación de XP. Nació respondiendo a las limitaciones que presenta XP, por ejemplo, facilitar la vida de los programadores, pero la concepción de desarrollar un

videojuego reúne muchas más características y disciplinas, justamente de ahí nace XGD (Barreno & Guaraca, 2013).

Al ser una metodología ágil y seguir principios ágiles, exige que los desarrolladores deban ofrecer una versión jugable del videojuego tan pronto como sea posible.

Su ciclo de vida consta de seis etapas (ver Figura 5), detalladas a continuación:

- **Exploración:** Se tiene como objetivo la primera entrega del videojuego, para eso se definen las historias de usuario que se involucran en esta primera entrega. Además, es competencia del equipo del desarrollo familiarizarse con las herramientas, tecnologías, entornos de desarrollo y prácticas que se utilizarán en el videojuego. Se empieza con las primeras aproximaciones sobre la idea del futuro del videojuego mediante el brainstorming.
- **Planificación:** El cliente define aquellas funcionalidades plasmadas en historias de usuario que serán entregadas con mayor prioridad, mientras que los desarrolladores realizan estimaciones sobre el esfuerzo que se requiere para llevar a cabo estos requerimientos. En esta fase también se estipula la entrega del cronograma con los respectivos acuerdos de contenido del primer entregable.
- **Iteración:** Es una fase que contiene varias iteraciones, debido a que para llegar a la constitución del videojuego se requieren varias. En la primera iteración se establece la idea del entregable de todo el videojuego y cuando se llega a la última, se continúa con el pase a producción. En cada plan de iteración se tratan aspectos como: historias de usuario no abordadas, velocidad del juego, pruebas de aceptación no superadas, básicamente, actividades inconclusas de la iteración anterior.
- **Producción:** En esta fase se evalúa la funcionalidad del videojuego antes de poder pasar al entorno del usuario final o el cliente. En esta etapa se pueden incluir más funcionalidades si es que previamente se detectó cambios.
- **Mantenimiento:** Se refiere a las tareas de soporte que se brinda al cliente cuando el videojuego se encuentra en la fase de producción. Extreme Game Development dentro de sus virtudes busca que mientras la primera versión se encuentra en producción, el videojuego se pueda mantener en funcionamiento, mientras se llevan a cabo otras iteraciones.
- **Muerte del Proyecto:** Cuando el cliente no tiene más funcionalidades para incluir en el videojuego, es decir cuando el cliente ha quedado satisfecho debido a que tanto los requisitos funcionales como los no funcionales fueron abarcados. Otra de las razones es en cambio, cuando el cliente no ha quedado conforme con los resultados esperados o simplemente cuando se carece de recursos para mantener el proyecto.

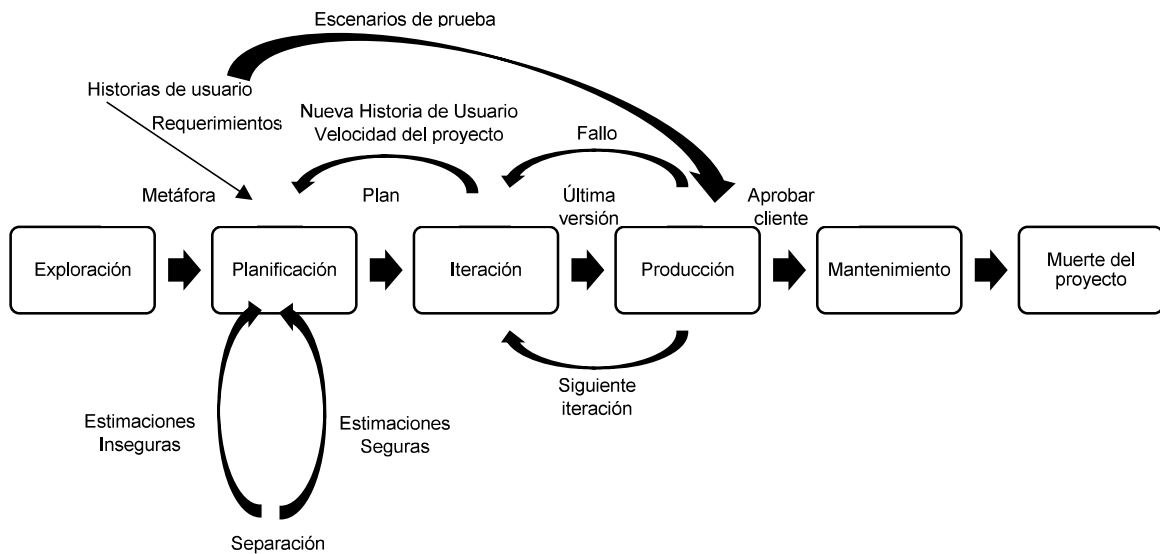


Figura 5 - Fases de un Proyecto Extreme Game Development
(Barreno & Guaraca, 2013)

Game Development Process

Cada vez se pretende ahondar más en el campo del desarrollo de videojuegos, juegos serios y software educativos, precisamente porque las investigaciones que se han realizado hasta el momento no se consideran completas.

Tal es el caso que la Academia Edumóvil, ha empleado sus esfuerzos en el desarrollo de aplicaciones para mejorar el proceso de enseñanza-aprendizaje de los niños de nivel básico haciendo uso de la tecnología móvil en el aula (Gerónimo-Castillo, Fernández-y-Fernández, & Ruiz-Rodríguez, 2008).

Se propone una nueva metodología que parte de GUP (Game Unified Process) y UCD (User-centered design), pero estos dos procesos detallan cuestiones únicamente de la parte técnica, más no de la problemática que ataca la Academia, además de que ambos procesos no incorporan dentro de su estudio elementos didácticos (Gerónimo-Castillo et al., 2008).

Entonces Game Development Process recoge de otros procesos y metodologías enfocados en los videojuegos, pautas y guías necesarias, además de GUP y UCD, constituyéndose de esta manera un nuevo proceso de desarrollo ad hoc.

Agrupar características de los enfoques ágiles, siendo un proceso iterativo, por ende, en cada ciclo o fase se recibe una retroalimentación de forma constante. Se basa en el ciclo de vida de la fuente (Gerónimo-Castillo et al., 2008) el mismo que es definido como incremental, iterativo y paralelo.

En la Figura 6, se muestran todas las fases del proceso de desarrollo mediante un diagrama de actividades donde además se visualiza la retroalimentación de un proceso a otro.

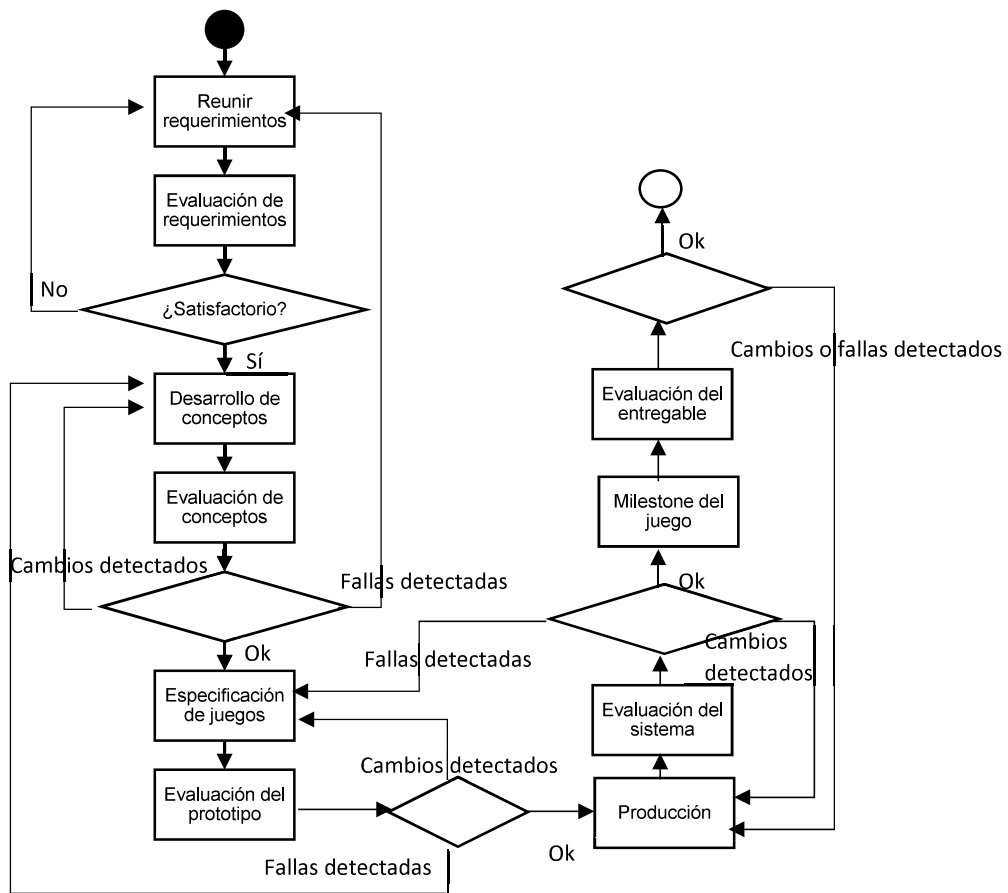


Figura 6 - Proceso de Desarrollo GDP
(Gerónimo-Castillo et al., 2008)

El proceso de desarrollo cuenta con las siguientes fases:

- **Recopilación de Requerimientos:** Se analiza las dificultades de aprendizaje conocido como “vacíos” que tienen los estudiantes, las asignaturas en donde existen estos problemas, asimismo, los datos recolectados y los requerimientos no funcionales. Con estos datos, se elabora un plan de proyecto y finalmente existe un proceso de evaluación que consiste en una evaluación de los requisitos funcionales.
- **Desarrollo del Concepto:** Se proponen soluciones a los problemas identificados, se define, además, la temática, el guion del videojuego, y los personajes principales.
- **Especificación del Juego:** Se realiza un planteamiento sobre el diseño general del videojuego, el cual va apoyado con la elaboración de prototipos, el resto de personajes se definen en esta fase, conjuntamente con aspectos técnicos, diagrama de navegación, métricas didácticas, sistemas de puntuación.

En la parte final de esta fase, se realizan pruebas conocidas con el nombre de “*Pruebas del Mago de Oz*”, las mismas que son realizadas por los estudiantes y evaluadas por los desarrolladores.

- **Producción:** Existe un trabajo de identificación de componentes reutilizables, una vez realizadas las pruebas con los estudiantes, es factible realizar mejoras en las interfaces y funcionalidades del juego, en otras palabras, realizar ajustes, en la evaluación de esta fase, el videojuego se somete a pruebas unitarias de integración y usabilidad, además de administrar versiones de componentes y módulos.
- **Hito del Juego:** Como es de conocimiento, la metodología está empleada para un software educativo para niños, por este motivo, es necesario definir o redefinir, en caso de que ya existan, políticas de uso sobre el software.

También se actualiza la base de conocimientos (componentes y documentación), se genera un reporte de productos terminados en proceso o defectuosos.

1.4.3 Metodologías de juegos serios

Zea (2011) manifiesta que los procesos de desarrollo orientados hacia los videojuegos, resurgieron a partir del movimiento liderado por Michael- Chen debido a que anteriormente se podían considerar como peligrosos, puesto que no presentaban contenidos de inmersión apropiados.

En la actualidad, a pesar de que se tiene un mayor uso de estas metodologías, incluso en diferentes áreas, el tema requiere de un estudio a mayor profundidad, por el hecho de que aún no se superan ciertos vacíos en la adopción de criterios como su base, para establecer metodologías con un alto grado de confianza que se puedan seguir.

En esta sección se estudian las metodologías más importantes en el campo de juegos serios para identificar las principales fortalezas y aún debilidades y así poder cumplir los objetivos de este estudio.

Proceso de Ágil para el Desarrollo de Juegos Serios (ASGDP)

ASGDP es un proceso ágil para el desarrollo de juegos serios que combina las líneas de investigación de Extreme Programming (XP) y de Diseño Centrado en el Usuario (Cano, González, Collazos, Arteaga & Zapata, 2015).

Se encuentra integrado por las siguientes etapas (ver Figura 7): objetivos pedagógicos, escenario pedagógico-juego, diseño de juegos y técnicas, implementación y pruebas (Cano et al., 2015). A continuación, se detalla cada una de las etapas:

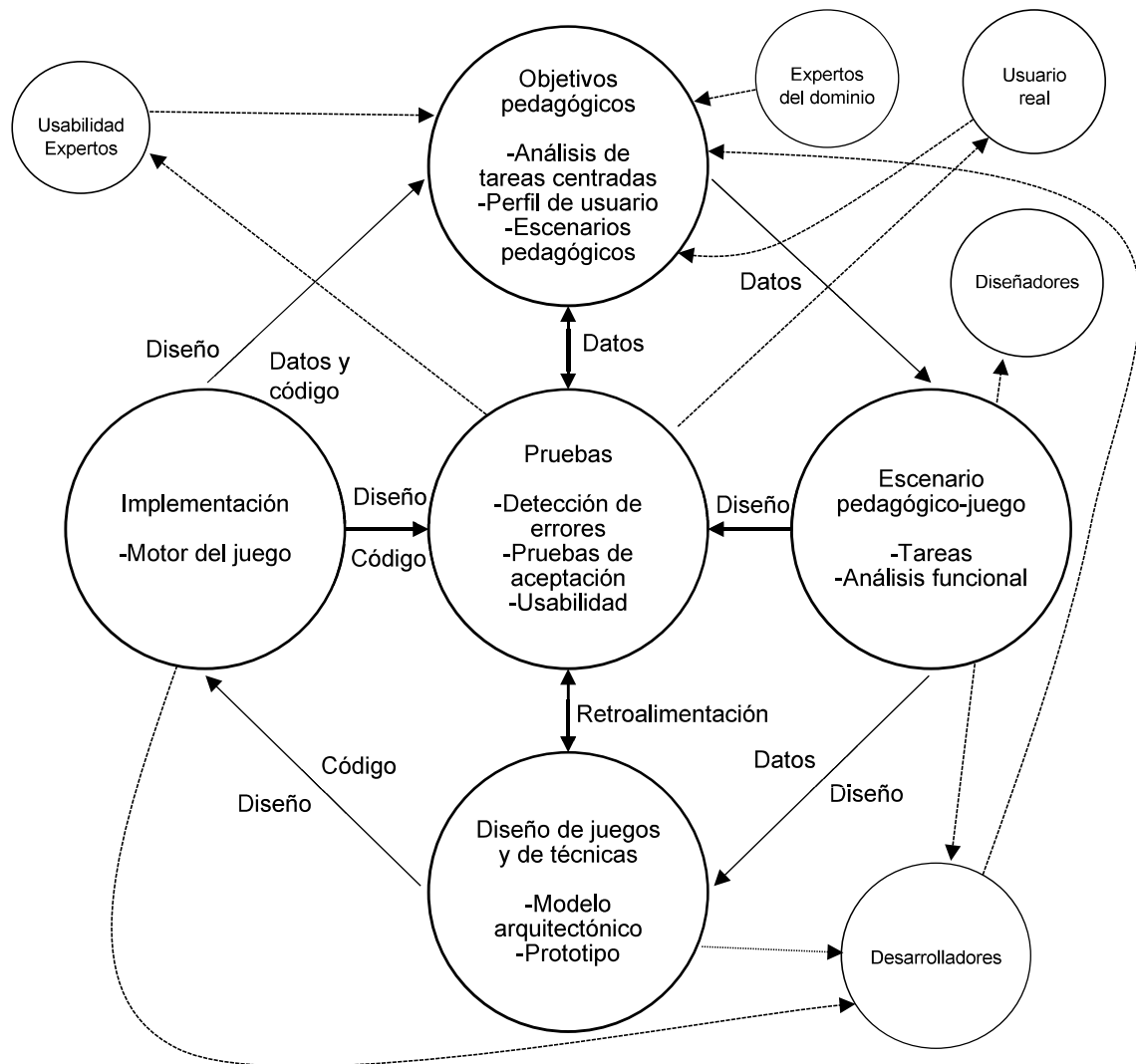


Figura 7 - Proceso de Desarrollo ASGDP
(Cano et al., 2015)

- **Objetivos pedagógicos:** En esta etapa, se determinan los objetivos pedagógicos que se construyen como referencia educativa, se consideran las habilidades, el conocimiento y las habilidades relacionadas con el usuario.

Se consideran, como parte de esta etapa, los siguientes cinco principios de aprendizaje: (1) los estudiantes participan en la resolución de problemas del mundo real, (2) la experiencia previa relevante para los estudiantes, (3) la instrucción muestra lo que se debe aprender, en lugar de repetir la información, (4) se requiere que el estudiante use un nuevo conocimiento o habilidad para resolver un problema, y (5) se alienta a los estudiantes a integrar el nuevo conocimiento o habilidad en su vida diaria.

- **Escenario pedagógico-juego:** En esta etapa, se tienen en cuenta dos escenarios: el escenario pedagógico y el escenario del juego, a través de una discusión, las partes interesadas establecen un conjunto de aspectos para conectar el modelo pedagógico con el juego, teniendo en cuenta los objetivos pedagógicos que se integrarán en la

historia y que apoyan el establecimiento de reglas y tareas de aprendizaje para interactuar con los objetivos del sistema.

- **Diseño de juegos y de técnicas:** En esta etapa, se define una interfaz gráfica de usuario principal del juego, donde se incluyen actividades como: definición del tema, elementos de interacción, personajes, estrategias de aprendizaje, canales de comunicación, variables de entrada y salida del sistema, y niveles de dificultad, entre otros.

El efecto psicológico que los diferentes colores pueden tener en los niños también interfiere en el diseño de la interfaz, de modo que los colores como el verde, representa la armonía y la sensación de calma; el azul, produce tranquilidad; y el rojo, estimula la acción, la energía y la vitalidad.

- **Implementación:** En esta etapa, todo el trabajo para desarrollar el software del juego se agrupa según un modelo arquitectónico.

Las características son definidas entre todos los miembros del equipo, se selecciona un motor gráfico y los componentes de trabajo son identificados.

- **Pruebas:** En la fase de prueba, la resolución de problemas del sistema se realiza de forma iterativa hasta que se pueda garantizar la usabilidad y accesibilidad del juego.

Las pruebas realizadas deben ser extensas y detalladas, ya que son instrumentales en el trabajo que se está realizando, en este momento es necesario llevar a cabo todas las funciones del sistema.

En esta etapa, participan todos los miembros del equipo, que son responsables de documentar la detección de errores a corregir. Se realizan dos tipos de pruebas: la calidad del software y la evaluación de la usabilidad con los usuarios finales. Para el efecto, es importante garantizar el cumplimiento de los objetivos por medio de una evaluación.

Emergo

Emergo es una metodología de desarrollo enfocada al desarrollo de juegos serios, basada en RUP ("Rational Unified Process") y en el método ADDIE (Nadolski et al., 2008).

Se encuentra integrado por las siguientes etapas (ver Figura 8): análisis, diseño, desarrollo, implementación y evaluación. A continuación, se explica cada una a mayor detalle (Nadolski et al., 2008):

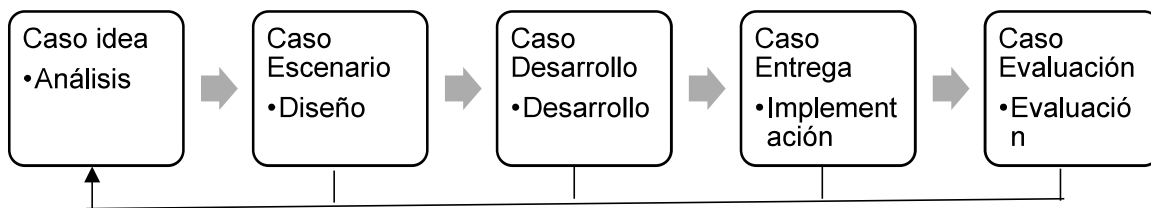


Figura 8 - Proceso de Desarrollo Emergo
(Nadolski et al., 2008)

- **Análisis:** Al principio, los desarrolladores deben considerar varios problemas relacionados con el caso previsto.

Se discute si el equipo del proyecto obtiene más información y más conciencia sobre el caso. Aquí se responderá a varias preguntas: ¿por qué se necesita?, ¿para quién se entiende?, ¿qué habrá en él?, ¿cómo se estructurará? Una imagen realista de posibilidades e imposibilidades surge antes de comenzar el caso diseño y desarrollo.

Al responder, un subconjunto apropiado de las preguntas, el equipo proporciona una descripción global del caso previsto.

- **Diseño:** La fase de diseño debe dar como resultado un documento de escenario detallado a través de los pasos intermedios, escenario marco y escenario ingredientes.

Cada paso proporciona más detalles.

- **Desarrollo:** Durante esta fase, el kit de herramientas EMERGO se utiliza para la entrada de datos con el escenario detallado proporcionando orientación, la entrada de datos no necesita experiencia en casos específicos, pero esto rara vez ocurre.

Se necesitan pruebas intensas y cíclicas durante la entrada de datos, que a menudo identifican problemas que solo pueden ser resueltos rápidamente por expertos en casos, por lo tanto, el diseño y los equipos de desarrollo necesitarán una superposición considerable.

Además, la entrada de datos y las pruebas son interdependientes y pueden requerir experiencia específica. Es importante acordar un procedimiento de trabajo concreto y un sistema de gestión de contenido es indispensable para una eficiente gestión de versiones de todos los “activos de casos digitales”³.

- **Implementación:** Durante la implementación, ciertos estudiantes y maestros acceden al caso.

³ Los activos de casos digitales se refieren al conjunto de recursos que son empleados en cada una de las fases, también conocidas como casos, dentro de la metodología Emergo.

La entrega del caso supone lo siguiente:

- un alumno puede elegir un tema en su entorno de estudio (portal del alumno);
- un profesor puede elegir en su entorno de trabajo (portal del profesor);
- la entrada de datos para el caso se ha verificado y está bien; y
- la gestión de ejecución de caso se ha utilizado para preparar la ejecución del caso que se lanzará.

Si se cumplen todos los aspectos anteriormente mencionados, entonces el caso puede ser "publicado".

Ciertos estudiantes y profesores pueden elegir el caso a través de sitio Web de Emergo utilizando sus datos de autorización.

- **Evaluación:** Se evaluará si el caso cumple con las demandas iniciales definidas durante el análisis: es ¿Es el uso real acorde con el uso esperado? Un claro enfoque de evaluación antes de lanzar el caso debe hacerse explícito en un plan de evaluación.

En principio, en la fase de evaluación, las respuestas a las preguntas en la fase de análisis ahora se pueden verificar.

Meconesis

Meconesis es una metodología para la concepción de juegos serios para niños con discapacidad auditiva (Cano, Arteaga, Collazos, González & Zapata, 2016).

Hace uso de Rational Unified Process (RUP); Concurrent Task Trees (CTT), para el modelado de interacciones; UML, en el caso de los diagramas de clase; IMS-LD, para la descripción de escenarios; y BPMN, referente a la notación de los procesos (Cano et al., 2016).

Se encuentra integrado por las siguientes fases (ver Figura 9): análisis, preproducción, producción y postproducción. Los modelos (ver Figura 9) que lo conforman son análisis, usuario, objetivos pedagógicos, tareas, escenarios y validación.

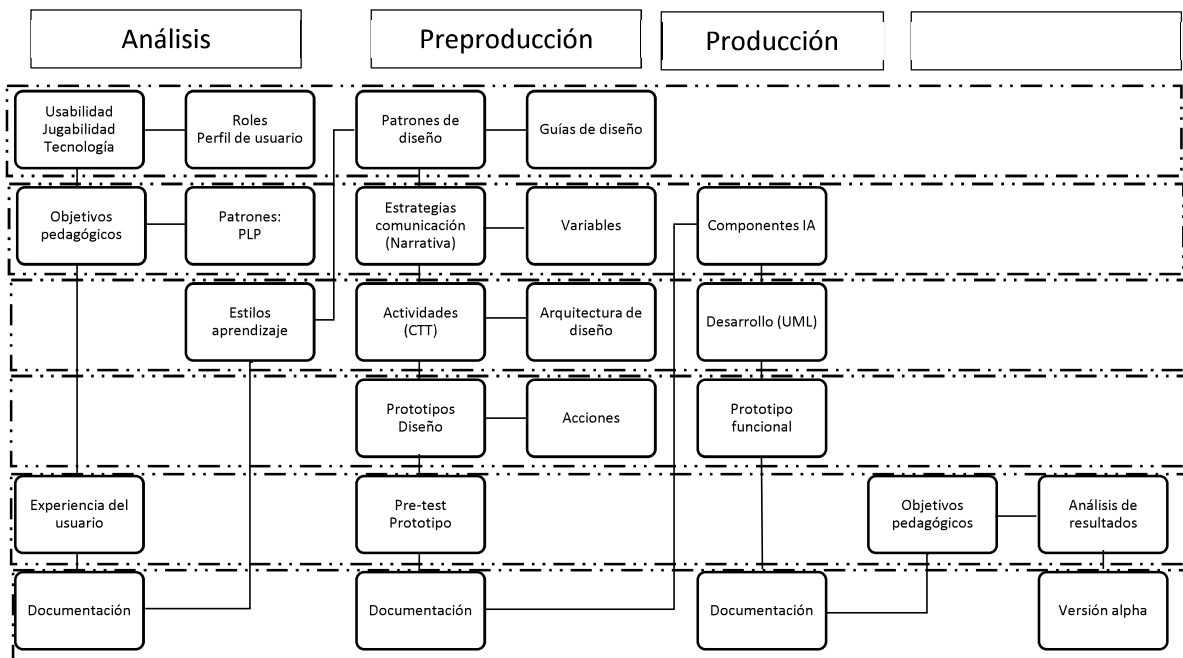


Figura 9 - Proceso de Desarrollo Meconesis
(Cano et al., 2016)

Se encuentra conformado por las siguientes fases:

- **Análisis:** Es el punto de partida del análisis de requerimientos. Se consideran los requerimientos en los escenarios lúdico y pedagógico, la participación del usuario final y la metodología de Diseño Centrado en el Uso.
- **Pre-producción:** Se relaciona con las guías de diseño para niños con discapacidad auditiva, los patrones de software para su implementación y el diseño de la interfaz de juego. Los factores involucrados son los estudios observacionales las estrategias lúdicas y pedagógicas, los escenarios lúdicos y pedagógicos, la implementación, y la comunicación y validación. Como parte del contenido del juego, se encuentran los modelos de tareas y escenarios.
- **Producción:** También conocida como implementación. Se orienta más al desarrollador y se integran los componentes necesarios para el funcionamiento del juego serio.
- **Post-producción:** En esta fase, se hace una evaluación del juego serio. Para esto, se propone un modelo de evaluación, considerando al experto y al usuario final, en su rol de evaluadores. Además, se valida el alcance de los objetivos pedagógicos, establecidos en la fase de análisis. Las 4 facetas que lo integran son las siguientes: objetivos pedagógicos, interacciones, problemas y progresión, y finalmente, condiciones de utilización.

Barajas Saavedra

La metodología desarrollada por Barajas, Álvarez, Muñoz, Santaolaya y Collazos (2014) se basa en los paradigmas de ingeniería de software y en el desarrollo a gran escala de recursos de aprendizaje digitales.

En la gráfica Figura 10, se pueden apreciar las etapas y actividades que conforman este proceso.

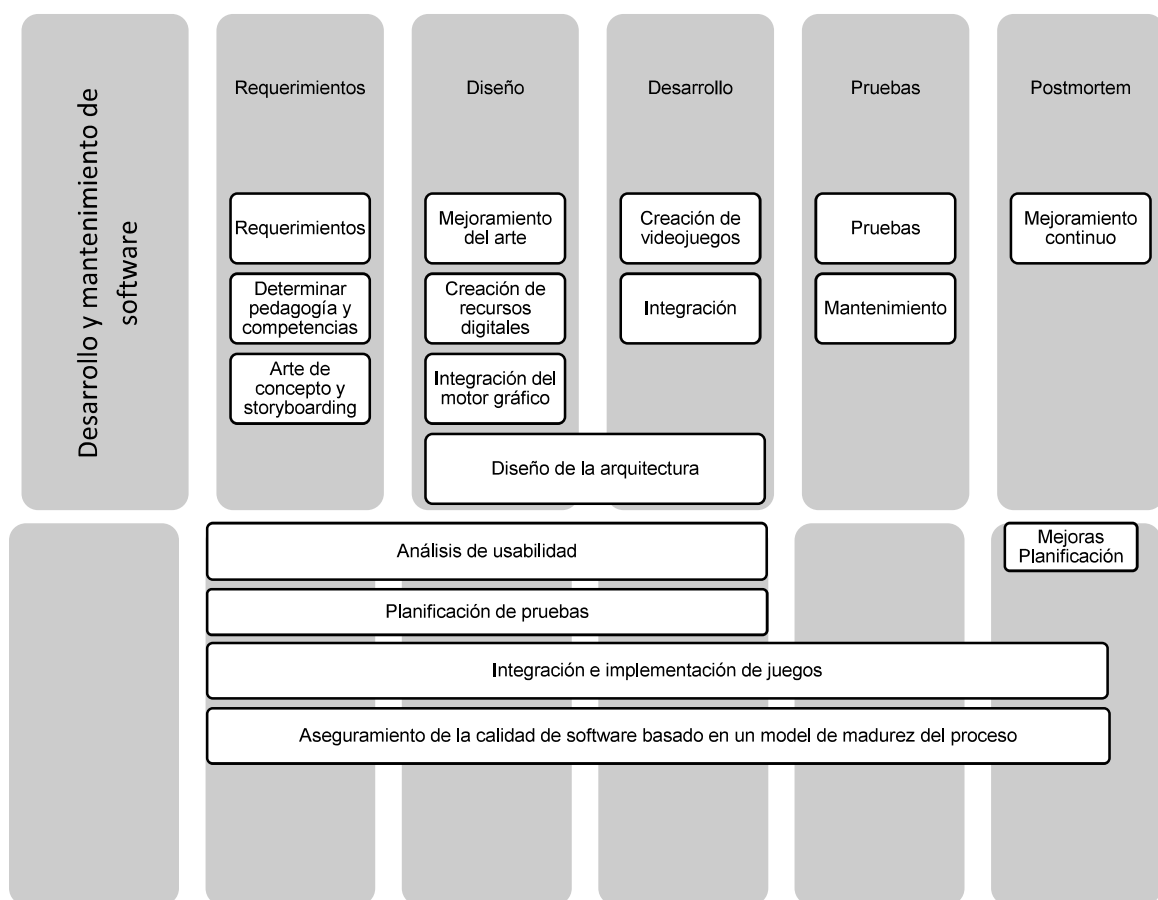


Figura 10 - Proceso de Desarrollo Barajas Saavedra
(Barajas et al., 2014)

- Etapa de requerimientos: El objetivo es establecer metas que cubran el juego; para establecer los mecanismos pedagógicos, a través de los cuales el conocimiento será transferido a los estudiantes; para determinar las competencias y las áreas de conocimiento que deben ser cubiertas; y para crear storyboard⁴ y arte conceptual.

⁴ Un storyboard es la secuencia de ilustraciones que permiten previsualizar una historia o animación.

- Etapa de diseño: Sus objetivos son crear todos los recursos digitales que necesita el motor de juego para la creación del videojuego. Estos digitales los recursos incluyen: ilustraciones 2D, modelos 3D, mapas, objetos, Materiales, superficies, etc., sonidos y música; y para crear juego motor si es necesario.
- Etapa de desarrollo: Se refiere a la creación del juego, lo cual incluye eventos, shader e IA; como parte del diseño del juego; y la integración de todos los elementos anteriores con menús, opciones, etc.
- Etapa de pruebas: En esta etapa se prueba el videojuego en los siguientes aspectos: absorción de conocimiento, técnico, usabilidad, utilidad; para obtener estadísticas de eficiencia; y para mantener el juego.
- Etapa postmortem: Se busca analizar toda la información del proceso y del producto recogido durante el proceso de desarrollo para mejorar el futuro desarrollos.

1.5 Conclusiones del Capítulo 1

Las metodologías de desarrollo de software han sido utilizadas sobre todo para los videojuegos cuyo factor primordial es la diversión, debido a que en su ciclo de vida presentan diferentes consideraciones que pueden ser adaptables para sus fines respectivos.

Por otro lado, las metodologías ágiles como Scrum se han utilizado principalmente para el desarrollo de juegos serios, presentando varios inconvenientes porque en algunos casos carecen de entretenimiento, de objetivos pedagógicos o ambos.

Sin duda las metodologías que se abordaron dentro de la sección de juegos serios, ya se preocupan en sus diferentes etapas, de controlar que se lleve de la mano el aprendizaje, la diversión y además un control o evaluación del progreso del usuario o jugador, permitiendo tener un seguimiento constante de los avances del jugador.

Otra de las problemáticas que se pudo identificar es que de por sí, un videojuego común conlleva varias dificultades en su fase de concepción, en su fase de captura de requisitos, diseño, y desarrollo principalmente, la razón es que se involucran otros contextos ajenos al desarrollo solamente, es decir los aspectos técnicos y artísticos.

En los juegos serios, además de estas trabas, se cuenta con dificultad de incorporar las temáticas de estudio y los diferentes tipos de diseños como sugiere Orjuela Duarte (Orjuela & Rojas, 2008).

A pesar de rescatar muchos aspectos positivos de las metodologías enfocadas en los juegos serios, se puede notar que no existe un proceso definido para el diseño de dichas metodologías, incluso el número de procesos de desarrollo que se encontró para realizar el respectivo estudio fue muy bajo respecto de las metodologías de desarrollo de software,

lo que indica que la literatura aún es escasa en este ámbito, de aquí es que surge la iniciativa de realizar el presente estudio.

A continuación, se detallará una tabla con aspectos determinantes en cada una de las metodologías estudiadas (ver Tabla 1).

Tabla 1 – Resumen de metodologías de desarrollo de juegos serios y de software

Metodologías de estudio	Campo de aplicación	¿Aplica al desarrollo de juegos serios?	Ventajas	Desventajas
Emergo (Nadolski et al., 2008)	Desarrollo de juegos serios	Sí	Provee de una metodología y un kit de herramientas para el desarrollo de juegos serios en la educación superior.	Al tratarse de una metodología nueva, los resultados que demuestran su efectividad son limitados.
Doodle (McMahon, 2009)	Desarrollo de juegos serios	Sí	Es una de las pocas metodologías de juegos serios que se enfoca en promover la experticia en diseñadores novatos.	Su implementación en un ambiente comercial e industrial es un aspecto pendiente.
Edos (Tran et al, 2010)	Desarrollo de juegos serios	Sí	Permite generar escenarios adaptables y flexibles para juegos serios, considerando los objetivos pedagógicos, escenarios pedagógicos y la interacción humano-computador.	No se cuenta con la plantilla de escenario apropiada para cada tipo de juego serio.
Esd (Çağatay, 2012)	Desarrollo de juegos serios	Sí	El desarrollador, los expertos del dominio y el usuario final participan de manera activa en cada una de las etapas del ciclo de desarrollo.	El uso de esta metodología puede llevar más tiempo en comparación a otras basadas en el ciclo de vida de desarrollo de software.

Barajas-Saavedra (Barajas et al., 2014)	Desarrollo de juegos serios	Sí	Es una metodología que hace énfasis en la descomposición basada en competencias para la producción de juegos serios.	El contenido del juego serio podría confundir a los chicos si la metodología no se implementa de manera adecuada.
Asgdp Agile serious game development process (Cano et al., 2015)	Desarrollo de juegos serios	Sí	Es una de las pocas metodologías de desarrollo de juegos serios que se basa en metodologías de desarrollo ágil.	Se requieren más estudios para evaluar la efectividad de los juegos serios resultantes de la metodología en grupos de niños.
Meconesis Metodología para la concepción de juegos serios para niños con discapacidad auditiva (Cano et al., 2016)	Desarrollo de juegos serios	Sí	Esta metodología es usada para el diseño de juegos serios para niños con discapacidades auditivas, usando el enfoque interacción humano-computador.	La metodología todavía requiere validación en lo referente a la eficiencia y experiencia de aprendizaje.
Crystal (Cockburn, 2001)	Desarrollo de software y videojuegos	No	Presenta una clasificación por colores, adaptable al equipo, tomando en cuenta el número de integrantes.	Limita al máximo el número de recursos utilizados y artefactos producidos.
PU Proceso Unificado (Jacobson & Booch, 2001)	Desarrollo de software	No	Es un proceso que puede ser adaptable a diferentes tipos de proyectos y organizaciones.	Recoleta las mejores prácticas de metodologías tradicionales, por ende es un proceso pesado.
Open UP (Balduino, 2007)	Desarrollo de software	No	Los elementos Open UP, otorgan una ventaja en el intercambio de información entre los equipos de desarrollo.	Al incluir contenido fundamental en el desarrollo, no es recomendable para proyectos grandes.

GDP Game Development Process (Gerónimo-Castillo, Fernández-y-Fernández, & Ruiz-Rodríguez, 2008)	Desarrollo de Software Educativo	No	Es una metodología utilizada para el desarrollo de software educativo en la enseñanza escolar.	No se aprecian desventajas significativas con respecto al respecto de metodologías.
GUP Game Unified Process (Gerónimo-Castillo et al., 2008)	Desarrollo de videojuegos	No	Es una propuesta eficiente para el desarrollo de videojuegos que documenta estrictamente cada paso.	No se aprecian desventajas significativas con respecto al respecto de metodologías.
Metodologías Ágiles (Orjuela & Rojas, 2008)	Desarrollo de software	No	Son adaptables a diferentes tipos de software, incluidos los videojuegos.	Al tener principios ágiles, muchos aspectos en desarrollo, quedan a criterio de los miembros del proyecto.
SUM (Acerenza et al., 2009)	Desarrollo de videojuegos	No	Aunque presenta un ciclo de desarrollo definido, puede ser modificado y adaptable a las distintas realidades.	Es concebida solo para proyectos pequeños con duración menor a 1 año.
Game-Scrum (Barbosa & Godoy, 2010)	Desarrollo de videojuegos	No	Permite la colaboración de investigadores, tal es así que su razón de ser se basa en unir varias sugerencias en búsquedas del mismo objetivo.	No se aprecian desventajas significativas con respecto al respecto de metodologías.

Huddle (Morales et al., 2010)	Desarrollo de videojuegos	No	Es un proceso adecuado para equipos multidisciplinarios	Sus equipos de trabajo solo admiten un máximo de 5 integrantes. Además se requiere un alto conocimiento de cada uno de ellos sobre el proceso.
Metodología genérica para videojuegos (Petrillo & Pimenta, 2010)	Desarrollo de videojuegos	No	El ciclo de desarrollo está pensado en las particularidades que aborda el desarrollo de videojuegos.	Se requiere capacitación constante del personal en herramientas, prácticas que se utilizan.
Waterfall Process (Morales et al., 2010)	Desarrollo de software	No	Proceso más usado en la industria de videojuegos.	Documentación exhaustiva. Retrasos en la obtención del producto final por correcciones de errores en etapas anteriores en el ciclo de desarrollo.
RUP Rational Unified Process (Martínez, 2011)	Desarrollo de software	No	Es un proceso adaptable a las necesidades del equipo.	Exceso de documentación y uso de recursos.
EGD Extreme Game Development (Barreno & Guaraca, 2013)	Desarrollo de videojuegos	No	Es una metodología pensada no solo para programadores sino también para diseñadores, artistas, gerentes, entre otros.	No se aprecian desventajas significativas con respecto al respecto de metodologías.
Scrum (Schwaber & Sutherland, 2013)	Desarrollo de software y videojuegos	No	Entrega de un producto funcional al final de cada sprint.	Para llevar a cabo el desarrollo de un videojuego se debe contar con experiencia en trabajos en equipo.

MSF Microsoft Solution Framework (Ze, 2014)	Desarrollo de software	No	Es un framework adaptativo a las necesidades del cliente. Podría ser adaptable al desarrollo de videojuegos.	No se aprecian desventajas significativas con respecto al respecto de metodologías.
EUP Essential Unified Process (Hui, Yan, Quanyu, & Zhiwen, 2015)	Desarrollo de software	No	Establece un conjunto de prácticas esenciales en todo proceso de desarrollo.	No se aprecian desventajas significativas con respecto al respecto de metodologías.
XP Extreme Programming (Pressman & Maxim, 2019)	Desarrollo de software	No	Favorable cuando existen requisitos imprecisos y muy cambiantes.	La familiarización de los programadores con alguna nueva herramienta puede conllevar mayor tiempo en los plazos de obtención del producto. Proceso más costoso.

1.6 Resumen del Capítulo 1

En este capítulo se presentó brevemente el contexto en el cual se desenvuelve el estudio, la problemática, las justificaciones de lo que se solucionará y desembocará en una alternativa confiable para determinación de metodologías de desarrollo de juegos serios en base a los criterios que se rescaten del análisis, permitiendo así, establecer los objetivos que orientarán la investigación.

A su vez, se logró conceptualizar algunas terminologías que serán pieza fundamental en el resto del proceso de estudio.

Además, se pudo realizar una descripción detallada de algunas metodologías destacadas tanto para el desarrollo de software como para el desarrollo de juegos serios y determinar qué aspectos son los más relevantes para utilizarlos en el Capítulo 2.

CAPITULO 2

2. METODOLOGÍA

Para el desarrollo de la investigación se tomó en cuenta la metodología de investigación “Design-Science” de March y Smith (1995) y Simon, (1996).

La ciencia del diseño es fundamentalmente un paradigma de resolución de problemas que permite innovar definiendo las ideas, las prácticas, las capacidades técnicas y los productos a través de los cuales el análisis, diseño, implementación, administración y uso de los sistemas de información se pueden llevar a cabo de manera efectiva y eficiente (Denning, 1997; Tschritzis, 1997).

El principio fundamental de la investigación del enfoque “Ciencia del diseño” es que el conocimiento y la comprensión de un problema de diseño y su solución se adquieren construyendo un artefacto (Gregor & Hevner, 2013; Hevner, 2004). En este caso, el artefacto resultante corresponde al conjunto de criterios, que a su vez servirán de base para, en un proyecto posterior, la construcción de la metodología de diseño de juegos serios.

Siguiendo el enfoque de Hevner, March, Park y Ram (2004), el diseño es considerado como la producción de un artefacto viable en forma de modelo, método e instanciación. La relevancia del problema se enfoca en desarrollar soluciones basadas en tecnología y en lo que respecta a la evaluación de diseño, la utilidad, calidad y eficacia tienen que ser rigurosamente demostradas con métodos de evaluación bien ejecutados. El entregar contribuciones claras y verificables se conoce como la relevancia del problema, cuando se hace referencia al rigor de la investigación, la aplicación de métodos rigurosos se hace tanto en la construcción como en la evaluación del diseño de artefactos. Otro aspecto a considerar es el diseño como un proceso de búsqueda, lo cual consiste en buscar los medios disponibles para llegar a los resultados deseados, finalmente la comunicación de la investigación se refiere a la presentación de manera efectiva del estudio realizado a la audiencia a la que va dirigido.

Para cumplir los objetivos del proyecto, se construirá, como artefacto, un conjunto de criterios que permitan adquirir el entendimiento del problema de diseño y su solución.

Se toman en cuenta dos aproximaciones en paralelo mediante una combinación de un método cualitativo y cuantitativo (Carrión, Santórum, Aguilar & Pinaida, 2019), en el primero, se realiza un análisis de los datos obtenidos a partir de una revisión sistemática de la literatura Bardin (1977), y posteriormente, los datos resultantes deben ser sometidos a un análisis cuantitativo (Barwise, 1977), para poder realizar inferencias en cuanto a los criterios claves de diseño.

A continuación, se realiza una revisión sistemática de literatura relacionada con las metodologías para el desarrollo de juegos serios y metodologías para el desarrollo de software.

Enseguida se efectúa una lectura detallada de los artículos científicos para la identificación de criterios para cada uno de los grupos, en base a cada criterio, se definirán los “conceptos” y “¿qué se busca?”. Únicamente los criterios estrechamente relacionados con

el desarrollo de software serán retenidos para su codificación y categorización como parte del análisis cualitativo. Este proceso está formado por dos alternativas: un análisis del investigador, donde se definirá la justificación de cada criterio respondiendo a la pregunta ¿qué se busca?, y una aplicación del método ponderado, usando el cálculo de la frecuencia del término, la frecuencia normalizada, la frecuencia inversa del documento y la multiplicación $ntf \times idf$. Una vez identificada la lista de criterios, como parte del análisis cualitativo y cuantitativo, se identificará cuáles entran en la categoría de necesarios y suficientes (Barwise, 1977), haciendo uso de lógica matemática.

2.1 Proceso General

Antes de ejecutar el proceso general, se procedió a obtener los criterios base de documentos claves en el desarrollo de juegos serios y de software. Para esto, se tomaron en consideración los estándares de la familia ISO 25000, referente a los requerimientos y evaluación de calidad de sistemas y software; el libro “Serious games: games that educate, train and inform” de Michael y Chen (2006) y el libro “Level up: the guide to great video game design” de Rogers (2010).

Se realizó una búsqueda sistemática de literatura relacionada con las metodologías para el desarrollo de juegos serios y metodologías para el desarrollo de software. Para la definición de criterios que sirvieron como base para los estudios posteriores se realizó un análisis cualitativo de las distintas metodologías previamente diferenciadas y separadas en Mendeley, para luego elaborar una tabla de criterios, con el fin de poder establecer más adelante si estas metodologías contemplan estos criterios en su estructura.

Los estudios que fueron tomados en cuenta para la presente investigación respetaron los siguientes criterios de inclusión:

- 1) publicaciones sobre metodologías de desarrollo de software y/o juegos serios en conferencias o revistas indexadas, disponible en algunos de los repositorios digitales: ACM, IEEE, JSTOR, ScienceDirect, Scopus, Springer, Web of Science y el motor de búsqueda de Google Scholar;
- 2) publicaciones realizadas entre noviembre de 2008 y junio de 2019; y
- 3) las publicaciones deben contener algunas de las siguientes palabras claves: serious game, game development process, development process design, agile software development, software development methodology, development methodology.

Los criterios de exclusión fueron los siguientes:

- 1) publicaciones con inconsistencias, por ejemplo, las citas no coinciden con sus respectivas referencias; y
- 2) publicaciones no vigentes o desactualizadas, por ejemplo, aquellas cuyo contenido fuese refutado por nuevas investigaciones.

El Criterio se define como un parámetro para clasificar el diseño de juegos serios, usando diferentes subclasificaciones. En otras palabras, se toma en cuenta diferentes aspectos que debe tener una metodología de diseño como: categoría, nivel, capas, propósito, orientación, herramientas, proceso, aplicaciones, jugador, usuarios, evaluación y

comunicación entre diseñadores y expertos (Abdelali, Mateu, Imma, Fatiha, & Mohammed, 2016).

No obstante, para el presente trabajo en el análisis de las diferentes metodologías de desarrollo de videojuegos, el enfoque no se centra solamente en el diseño, sino que examina características del resto de fases del ciclo de desarrollo del software.

Este conjunto de criterios se ha obtenido de diferentes fuentes, en cuanto a parámetros afines al desarrollo de software y a los aspectos previamente analizados (Abdelali et al., 2016).

Por otro lado, en lo referente a la calidad del producto de software nos basamos en algunos criterios de la familia de estándares ISO 25000.

Es importante mencionar las cuatro dimensiones afines al tema de definición de criterios de diseño para juegos serios, es decir, se tomarán en cuenta los criterios de cada dominio (Carrión, Santórum, Pérez & Aguilar, 2017). Se los detalla a continuación:

- Desarrollo de Juegos Serios Educativos
- Personas con habilidades cognitivas no estándar
- Desarrollo centrado en el usuario
- Desarrollo participativo

Otro aspecto determinante, fue definir el objetivo de cada criterio, respondiendo a la pregunta ¿Qué se busca durante el análisis?

Posteriormente, para uno de los dos análisis en los cuales se divide la investigación, se requirió escribir los sinónimos de cada uno de los criterios tanto en inglés como en español, debido a que, en la documentación recabada, priman estos dos idiomas.

Para facilitar la comprensión y lectura, hemos establecido una codificación asociada al proceso metodológico. A continuación, se detalla la codificación usada para cada una de las tablas que conforman el proceso usado en la metodología. Para esto, se manejan 3 niveles (ver Tabla 2): siglas, índices y códigos.

Tabla 2 - Codificación de las Tablas Usadas en la Metodología

Siglas		Índices		Códigos	
Elemento	Descripción	Elemento	Descripción	Elemento	Descripción
PG	Proceso general	E	Entrada	CDM	Conjunto de documentos de las metodologías
PA1	Proceso antecedente 1	A	Actividad	CCB	Conjunto de criterios base
PA2	Proceso antecedente 2	S	Salida	CCC	Conjunto de Criterios Clasificados
PC	Proceso consecuente			MJ	Matriz de justificación
				MB	Matriz booleana

				MAR	Matriz de análisis reducida
				MFA	Matriz de frecuencia absoluta
				MFN	Matriz de frecuencia normalizada
				MFI	Matriz de frecuencia inversa
				MNTFIDF	Matriz NTF-IDF
				MI	Matriz de instanciación
				MC	Matriz de comportamiento
				ME	Matriz de emparejamiento

La estructura de la codificación se muestra a continuación con un ejemplo demostrativo (ver Tabla 3) para el conjunto de criterios base que forman parte de las entradas del proceso general.

Tabla 3 - Estructura de la Codificación

Elemento	Sigla	Índice	Código	Codificación
Conjunto de criterios base que forman parte de las entradas del proceso general	PG	E	CCB	PG-E-CCB

Como parte de las entradas, se tienen los documentos de las metodologías a evaluar para el desarrollo de juegos serios y de software (ver Tabla 4).

Tabla 4 - Conjunto de Documentos de las Metodologías (PG-E-CDM)

Metodologías de desarrollo de JS	Metodologías de desarrollo de software
Documento de Metodología 1	Documento de Metodología 1
Documento de Metodología 2	Documento de Metodología 2
Documento de Metodología n	Documento de Metodología n

La otra entrada (ver Tabla 5) muestra la estructuración de la tabla base, que contiene las dimensiones, los criterios, definición, el objetivo de búsqueda y los sinónimos.

Tabla 5 - Conjunto de Criterios Base (PG-E-CCB)

Dimensiones	Criterios	Definición	Explicación ¿Qué se busca?	Sinónimos
Dimensión 1 (D1)	Criterio 1 D1	Definición 1 D1	Explicación 1 D1	Sinónimos 1 D1
	Criterio 2 D1	Definición 2 D1	Explicación 2 D1	Sinónimos 2 D1
	Criterio n D1	Definición n D1	Explicación n D1	Sinónimos n D1
Dimensión 2 (D2)	Criterio 1 D2	Definición 1 D2	Explicación 1 D2	Sinónimos 1 D2
	Criterio 2 D2	Definición 2 D2	Explicación 2 D2	Sinónimos 2 D2
	Criterio n D2	Definición n D2	Explicación n D2	Sinónimos n D2
Dimensión 3 (D3)	Criterio 1 D3	Definición 1 D3	Explicación 1 D3	Sinónimos 1 D3
	Criterio 2 D3	Definición 2 D3	Explicación 2 D3	Sinónimos 2 D3
	Criterio n D3	Definición n D3	Explicación n D3	Sinónimos n D3
Dimensión 4 (D4)	Criterio 1 D4	Definición 1 D4	Explicación 1 D4	Sinónimos 1 D4
	Criterio 2 D4	Definición 2 D4	Explicación 2 D4	Sinónimos 2 D4
	Criterio n D4	Definición n D4	Explicación n D4	Sinónimos n D4

Una vez obtenidas las tablas anteriores, se prosigue con la elaboración del proceso, el cual parte de un conjunto de documentos sobre los que se eligieron los criterios base y sobre los cuales se van a extraer las metodologías a evaluar; este proceso tiene un análisis cualitativo y cuantitativo (Carrión et al, 2019).

A continuación, se divide el proceso en las siguientes partes (ver Figura 11): proceso antecedente 1 (opinión del investigador), proceso antecedente 2 (análisis ponderado) y el proceso consecuente.

Cada subproceso es trabajado individualmente hasta obtener la matriz booleana, de ahí en adelante, se tiene un proceso consecuente, el mismo que es común para los dos procesos antecedentes.

Bajo el proceso consecuente, se podrá categorizar a los criterios como necesarios y suficientes, suficientes y no necesarios, necesarios y no suficientes y ninguno de los anteriores.

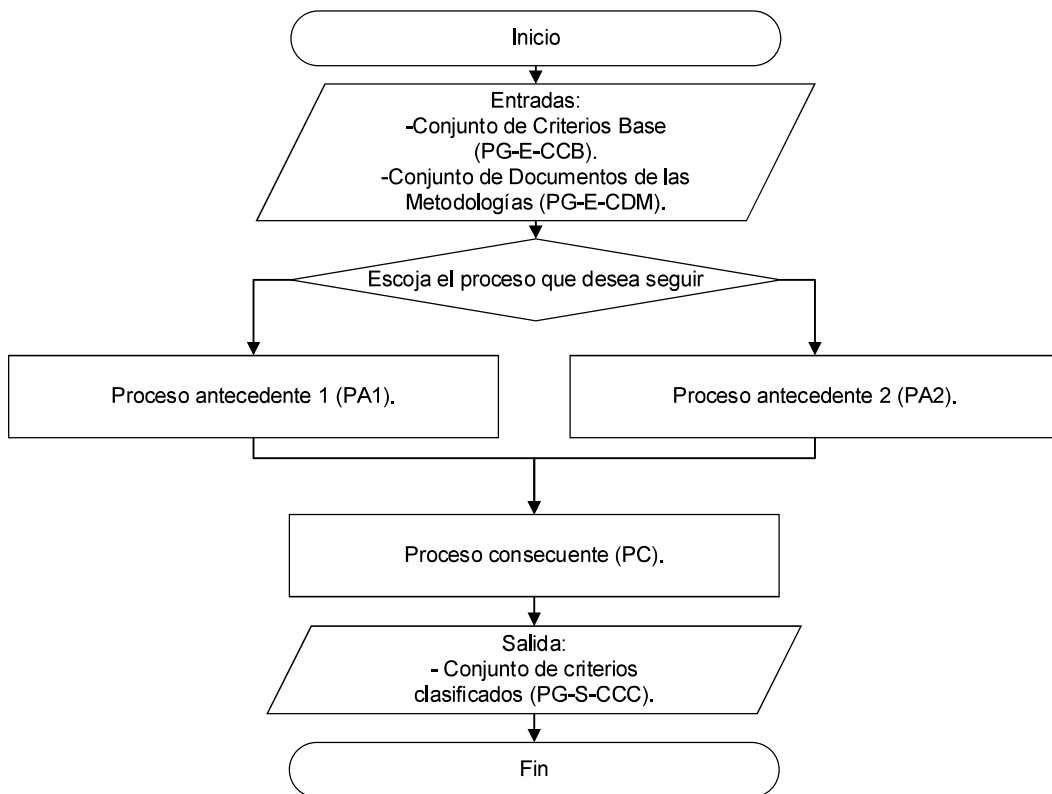


Figura 11 - Diagrama de Flujo del Proceso General (PG)
Elaborado por los autores

2.1.1 Proceso Antecedente 1 (Opinión del investigador)

El objetivo de este proceso es la obtención de la matriz booleana en base a la opinión del investigador, a continuación, se presenta el proceso en el diagrama de flujo (ver Figura 12) que tiene como entradas el conjunto de documentos de las metodologías a evaluar (ver Tabla 4) y los criterios base (ver Tabla 5).

Tomando la Tabla 4 y la Tabla 5 como punto de referencia, el siguiente paso consiste en formar la matriz de justificación, posteriormente la matriz booleana (ver Figura 12).

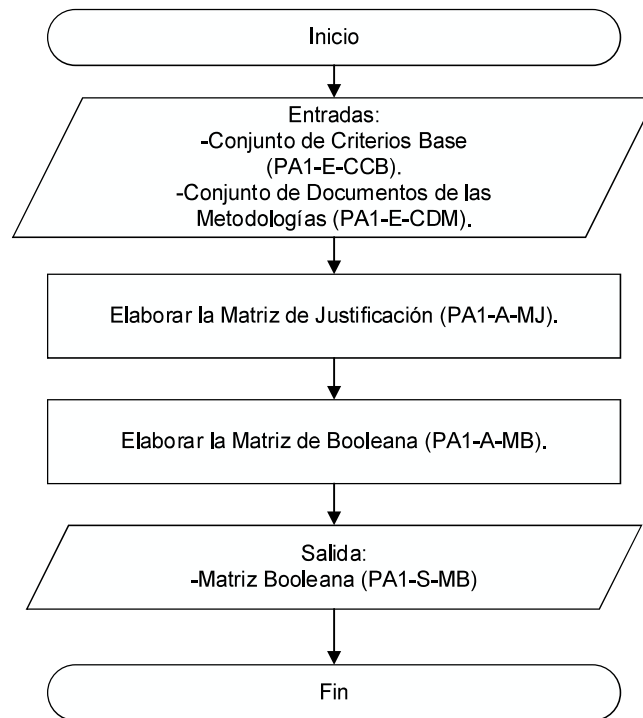


Figura 12 - Diagrama de Flujo del Proceso Antecedente 1
Elaborado por los autores

Matriz de Justificación

Una vez que se cuenta con los criterios ya definidos y establecidos, se procede a la lectura de cada uno de los documentos relacionados con las metodologías de desarrollo de juegos serios y software, en este proceso es estrictamente necesario responder la pregunta “¿Qué se busca?”.

En caso de responder las preguntas que enmarcan el cuestionamiento general “¿Qué se busca?”, se coloca la justificación correspondiente en la matriz, acorde a la porción de texto en la que, a opinión del investigador, el criterio está inmiscuido, caso contrario, se coloca la justificación “No se especifica en el proceso de desarrollo (N/E)” lo que implica que no se pudo responder la pregunta objetiva (ver Tabla 6).

Además, se coloca la referencia correspondiente, es decir la fuente de información sobre la metodología estudiada, de donde se extrajeron las justificaciones pertinentes.

Cada metodología tendrá su respectiva matriz de justificación, es decir para cada campo de estudio, metodologías de desarrollo de juegos serios y metodologías de desarrollo de software.

Tabla 6 - Matriz de Justificación (PA1-A-MJ)

Dimensiones	Criterios	Justificación	Bibliografía
Dimensión 1 (D1)	Criterio 1 D1	Justificación o (N/E)	Bibliografía 1
	Criterio 2 D1	Justificación o (N/E)	
	Criterio n D1	Justificación o (N/E)	
Dimensión 2 (D2)	Criterio 1 D2	Justificación o (N/E)	
	Criterio 2 D2	Justificación o (N/E)	
	Criterio n D2	Justificación o (N/E)	
Dimensión 3 (D3)	Criterio 1 D3	Justificación o (N/E)	
	Criterio 2 D3	Justificación o (N/E)	
	Criterio n D3	Justificación o (N/E)	
Dimensión 4 (D4)	Criterio 1 D4	Justificación o (N/E)	
	Criterio 2 D4	Justificación o (N/E)	
	Criterio n D4	Justificación o (N/E)	

Matriz Booleana

La matriz booleana se llena con la información de la matriz de justificación (ver Tabla 6).

Si en cada metodología, cada criterio pudo responder la pregunta objetivo y además está sustentada o justificada, tanto con la porción de texto y con la bibliografía correspondiente, se coloca el valor booleano "1", por el contrario, si la pregunta objetiva, no fue respondida y a su vez, el criterio fue justificado con la frase "No se especifica en el proceso de desarrollo", se coloca el valor booleano "0".

En otras palabras, si a criterio del investigador, existe justificación de existencia del criterio en el documento, se coloca 1, caso contrario, 0.

La matriz booleana, ya presenta un orden, primero, diferenciando los campos de análisis y segundo mostrando los valores de 1s y 0s para todos los criterios en todas las metodologías analizadas, (ver Tabla 7).

Tabla 7 - Matriz Booleana del Proceso Antecedente 1 (PA1-A-MB)

METODOLOGIAS CRITERIOS (C)	Metodologías de desarrollo de JS			Metodologías de desarrollo de software		
	Documento de Metodología 1	Documento de Metodología 2	Documento de Metodología n	Documento de Metodología 1	Documento de Metodología 2	Documento de Metodología n
Criterio 1 (C1)	0 ∨ 1	0 ∨ 1	0 ∨ 1	0 ∨ 1	0 ∨ 1	0 ∨ 1
Criterio 2 (C2)	0 ∨ 1	0 ∨ 1	0 ∨ 1	0 ∨ 1	0 ∨ 1	0 ∨ 1
Criterio 3 (C3)	0 ∨ 1	0 ∨ 1	0 ∨ 1	0 ∨ 1	0 ∨ 1	0 ∨ 1
Criterio n (Cn)	0 ∨ 1	0 ∨ 1	0 ∨ 1	0 ∨ 1	0 ∨ 1	0 ∨ 1

Cuando se ha llegado a tener la matriz booleana bajo el método de opinión del investigador, se debe continuar con el proceso consecuente.

2.1.2 Proceso Antecedente 2 (Análisis Ponderado)

El análisis Ponderado, a diferencia del proceso anterior, ya no cuenta con subjetividad, al contrario, tiene un sustento matemático que lo respalda.

Está basado en la ponderación de términos para la recuperación de textos.

En varias ocasiones, un texto puede ser catalogado como relevante o de mucha importancia, gracias al peso que tiene cada uno de sus términos en el documento.

Este término que irá asociado a la importancia de una palabra, una frase o cualquier unidad de indexación utilizada para identificar el contenido de un texto (Singhal, Salton, Mitra, & Buckley, 1996).

Para establecer la importancia de un término en un texto, se siguen las pautas propuestas en (Singhal et al., 1996), las mismas que son las siguientes:

- La frecuencia absoluta del término (tf), en otras palabras, el número de veces que un criterio o su sinónimo, se repite en todo el documento de estudio. Los valores de ocurrencia del criterio original deberán ser sumados a los valores de ocurrencia de sus sinónimos, en caso de existir.
- La frecuencia del término normalizada (ntf) permite mitigar la anomalía de observar frecuencias de términos más altas en documentos más grandes, donde tiende a repetirse la misma palabra varias veces.
- La frecuencia inversa del documento (idf), que hace relación a la aparición del término o criterio en todos los documentos de estudio.
- El producto entre la frecuencia del término normalizada y la frecuencia inversa del documento (nft-idf) que determina la relevancia del término en un documento, respecto del conjunto de documentos analizados.

Al igual que en el proceso anterior, el análisis ponderado, durante su proceso trabaja con distintas matrices, hasta llegar a la matriz booleana y luego continuar con el proceso consecuente (ver Figura 13).

Es necesario, hacer énfasis, en los motivos de interpretación de resultados, que se obtendrán matrices booleanas utilizando la moda, el promedio y la varianza, para luego determinar cuál de ellas, provee resultados más coherentes.

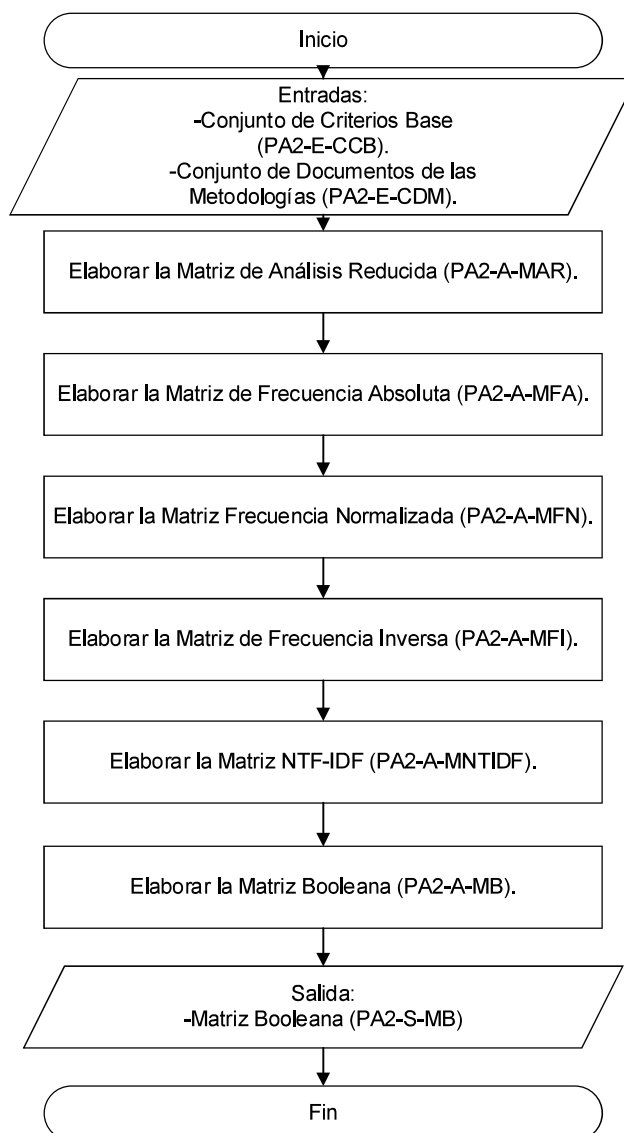


Figura 13 - Diagrama de Flujo del Proceso Antecedente 2
Elaborado por los autores

Matriz de análisis reducida

Se puede únicamente separar y seleccionar aquello que es necesario para esta parte de la investigación, es decir agrupar los criterios y sus respectivos sinónimos (ver Tabla 8).

Tabla 8 - Matriz de Análisis Reducida (PA2-A-MAR)

Criterios	Sinónimos
Criterio 1 (C1)	Sinónimos del Criterio 1 (Sinónimo 1, Sinónimo 2, Sinónimo n) (S1)
Criterio 2 (C2)	Sinónimos del Criterio 2 (Sinónimo 1, Sinónimo 2, Sinónimo n) (S2)
Criterio 3 (C3)	Sinónimos del Criterio 3 (Sinónimo 1, Sinónimo 2, Sinónimo n) (S3)
Criterio n (Cn)	Sinónimos del Criterio 4 (Sinónimo 1, Sinónimo 2, Sinónimo n) (S4)

Matriz de Frecuencia Absoluta

Luego de que se conocen cada uno de los criterios con sus respectivos sinónimos, es momento de contabilizarlos, uno por uno, en cada documento, es decir, esta matriz se llena con los valores numéricos que resultan de las veces que el criterio y sus sinónimos son detectados en cada texto de estudio, (ver Tabla 9). Asimismo, esta matriz está ordenada por los artículos que responden a las metodologías de juegos serios y a las metodologías de desarrollo de software.

Tabla 9 - Matriz de Frecuencia Absoluta (PA2-A-MFA)

METODOLOGIAS CRITERIOS (C) + SINÓNIMOS (S)	Metodologías de desarrollo de JS			Metodologías de desarrollo de software		
	Documento de Metodología 1	Documento de Metodología 2	Documento de Metodología n	Documento de Metodología 1	Documento de Metodología 2	Documento de Metodología n
C1 + S1	$tf_{t,d}$	$tf_{t,d}$	$tf_{t,d}$	$tf_{t,d}$	$tf_{t,d}$	$tf_{t,d}$
C2 + S2	$tf_{t,d}$	$tf_{t,d}$	$tf_{t,d}$	$tf_{t,d}$	$tf_{t,d}$	$tf_{t,d}$
C3 + S3	$tf_{t,d}$	$tf_{t,d}$	$tf_{t,d}$	$tf_{t,d}$	$tf_{t,d}$	$tf_{t,d}$
Cn + Sn	$tf_{t,d}$	$tf_{t,d}$	$tf_{t,d}$	$tf_{t,d}$	$tf_{t,d}$	$tf_{t,d}$

El contenido de la tabla puede ser representado en lenguaje matemático. El conjunto T (1) representa el conjunto de documentos de los que se van a extraer las metodologías a evaluar. El conjunto D (2) representa el conjunto de documentos de los que se van a extraer las metodologías a evaluar.

$$T = \left\{ t_i \mid t_i \in \left\{ \begin{array}{l} \text{Términos identificados en los documentos} \\ \text{de las metodologías para el desarrollo} \\ \text{de juegos serios y de software} \end{array} \right\} \wedge i \in \mathbb{Z}^+ \right\} \quad (1)$$

$$D = \left\{ d_j \mid d_j \in \left\{ \begin{array}{l} \text{Documentos de las metodologías evaluadas} \\ \text{para el desarrollo de juegos serios} \\ \text{y de software} \end{array} \right\} \wedge j \in \mathbb{Z}^+ \right\} \quad (2)$$

Matriz de Frecuencia Normalizada

Posteriormente, se procede a construir la matriz de frecuencia normalizada (ver Tabla 10). La expresión $ntf(t_i, d_j, a)$, se obtiene de la división de la frecuencia absoluta de un criterio de estudio en el documento y la mayor frecuencia de un criterio en el mismo documento.

Dicha expresión (3) se representa de la siguiente forma:

$$ntf(t_i, d_j, a) = a + (1 - a) \left(\frac{tf_{t,d}}{tf_{\max}(d_j)} \right) \quad (3)$$

El valor para la constante propuesta es $a = 0,5$, por lo que en este caso, también se toma como referencia este valor. La constante a representa un smoothing o almohadilla que suaviza a la función de frecuencia, la misma que es utilizada como una de las técnicas de normalización, concretamente la normalización tf máxima. La expresión $tf_{t,d}$, representa la frecuencia normalizada para cada documento, mientras que $tf_{\max}(d_j)$, representa la frecuencia absoluta máxima en el documento de estudio (Singhal et al., 1996).

Tabla 10 - Matriz de Frecuencia Normalizada (PA2-A-MFN).

	Metodologías de desarrollo de JS			Metodologías de desarrollo de software		
METODOLOGIAS CRITERIOS (C) + SINÓNIMOS (S)	Documento de Metodología 1	Documento de Metodología 2	Documento de Metodología n	Documento de Metodología 1	Documento de Metodología 2	Documento de Metodología n
C1 + S1	$ntf_{t_i,d_j,a}$	$ntf_{t_i,d_j,a}$	$ntf_{t_i,d_j,a}$	$ntf_{t_i,d_j,a}$	$ntf_{t_i,d_j,a}$	$ntf_{t_i,d_j,a}$
C2 + S2	$ntf_{t_i,d_j,a}$	$ntf_{t_i,d_j,a}$	$ntf_{t_i,d_j,a}$	$ntf_{t_i,d_j,a}$	$ntf_{t_i,d_j,a}$	$ntf_{t_i,d_j,a}$
C3 + S3	$ntf_{t_i,d_j,a}$	$ntf_{t_i,d_j,a}$	$ntf_{t_i,d_j,a}$	$ntf_{t_i,d_j,a}$	$ntf_{t_i,d_j,a}$	$ntf_{t_i,d_j,a}$
Cn + Sn	$ntf_{t_i,d_j,a}$	$ntf_{t_i,d_j,a}$	$ntf_{t_i,d_j,a}$	$ntf_{t_i,d_j,a}$	$ntf_{t_i,d_j,a}$	$ntf_{t_i,d_j,a}$

Matriz de Frecuencia Inversa

Para la construcción de la matriz de frecuencia inversa (ver Tabla 11), la expresión $idf(t_i, D)$ se obtiene del logaritmo de la división del número total de documentos analizados y el número de documentos del criterio de estudio.

Dicha expresión (4) se representa de la siguiente forma:

$$idf(t_i, D) = \log\left(\frac{n(D)}{df_t}\right) \quad (4)$$

Para esta expresión matemática, $n(D)$ representa el número total de documentos a analizar, df_t , es el número total de documentos en el que dicho criterio aparece en el documento.

Tabla 11 - Matriz de Frecuencia Inversa (PA2-A-MFI).

METODOLOGIAS CRITERIOS (C) + SINÓNIMOS (S)	Metodologías de desarrollo para JS			Metodologías de desarrollo de software		
	Documento de Metodología 1	Documento de Metodología 2	Documento de Metodología n	Documento de Metodología 1	Documento de Metodología 2	Documento de Metodología n
C1 + S1	$idf_{t_i,d_j,a}$	$idf_{t_i,d_j,a}$	$idf_{t_i,d_j,a}$	$idf_{t_i,d_j,a}$	$idf_{t_i,d_j,a}$	$idf_{t_i,d_j,a}$
C2 + S2	$idf_{t_i,d_j,a}$	$idf_{t_i,d_j,a}$	$idf_{t_i,d_j,a}$	$idf_{t_i,d_j,a}$	$idf_{t_i,d_j,a}$	$idf_{t_i,d_j,a}$
C3 + S3	$idf_{t_i,d_j,a}$	$idf_{t_i,d_j,a}$	$idf_{t_i,d_j,a}$	$idf_{t_i,d_j,a}$	$idf_{t_i,d_j,a}$	$idf_{t_i,d_j,a}$
Cn + Sn	$idf_{t_i,d_j,a}$	$idf_{t_i,d_j,a}$	$idf_{t_i,d_j,a}$	$idf_{t_i,d_j,a}$	$idf_{t_i,d_j,a}$	$idf_{t_i,d_j,a}$

Matriz NTF-IDF

A continuación, se procede a construir la matriz ntf-idf (ver Tabla 12). Esta matriz, es el resultado de la multiplicación de $ntf(t_i, d_j) * idf(t_i, D)$.

En base a lo detallado, la función normalizada (5) se representa de la siguiente forma:

$$ntf\ idf(t_i, d_j, a, t_i, D) = ntf(t_i, d_j, a) * idf(t_i, D) \quad (5)$$

Tabla 12 - Matriz NTF-IDF (PA2-A-MNTFIDF).

METODOLOGIAS CRITERIOS (C) + SINÓNIMOS (S)	Metodologías de desarrollo de JS			Metodologías de desarrollo de software		
	Documento de Metodología 1	Documento de Metodología 2	Documento de Metodología n	Documento de Metodología 1	Documento de Metodología 2	Documento de Metodología n
C1 + S1	$ntfidf_{t,d}$	$ntfidf_{t,d}$	$ntfidf_{t,d}$	$ntfidf_{t,d}$	$ntfidf_{t,d}$	$ntfidf_{t,d}$
C2 + S2	$ntfidf_{t,d}$	$ntfidf_{t,d}$	$ntfidf_{t,d}$	$ntfidf_{t,d}$	$ntfidf_{t,d}$	$ntfidf_{t,d}$
C3 + S3	$ntfidf_{t,d}$	$ntfidf_{t,d}$	$ntfidf_{t,d}$	$ntfidf_{t,d}$	$ntfidf_{t,d}$	$ntfidf_{t,d}$
Cn + Sn	$ntfidf_{t,d}$	$ntfidf_{t,d}$	$ntfidf_{t,d}$	$ntfidf_{t,d}$	$ntfidf_{t,d}$	$ntfidf_{t,d}$

Luego de obtener la Tabla 12, se procede a construir la matriz booleana (ver Tabla 13). Antes, se destaca que una vez que se obtuvieron los resultados en la matriz de frecuencias normalizadas (ver Tabla 12), se procede a calcular el promedio y la varianza, con el objetivo de trabajar con varios resultados, posteriormente.

Tabla 13 - Matriz Booleana (PA2-A-MB)

	Metodologías de desarrollo de JS	Metodologías de desarrollo de software
--	----------------------------------	--

METODOLOGIAS CRITERIOS (C) + SINÓNIMOS (S)	Documento de Metodología 1	Documento de Metodología 2	Documento de Metodología n	Documento de Metodología 1	Documento de Metodología 2	Documento de Metodología n
C1 + S1	0 ∨ 1	0 ∨ 1	0 ∨ 1	0 ∨ 1	0 ∨ 1	0 ∨ 1
C2 + S2	0 ∨ 1	0 ∨ 1	0 ∨ 1	0 ∨ 1	0 ∨ 1	0 ∨ 1
C3 + S3	0 ∨ 1	0 ∨ 1	0 ∨ 1	0 ∨ 1	0 ∨ 1	0 ∨ 1
Cn + Sn	0 ∨ 1	0 ∨ 1	0 ∨ 1	0 ∨ 1	0 ∨ 1	0 ∨ 1

Dicho de otra forma, a partir de aquí, se trabajará con 2 resultados para la matriz booleana, es decir una matriz booleana para el promedio y otra matriz booleana para la varianza.

El contenido de la tabla puede ser representado en lenguaje matemático (6-14) de la siguiente manera.

$$X = \left\{ x_i \mid x_i \in \left\{ \begin{array}{l} \text{Criterios identificados en las metodologías} \\ \text{para el desarrollo de juegos serios y de software} \end{array} \right\} \wedge i \in \mathbb{Z}^+ \right\} \quad (6)$$

$$X = T \quad (7)$$

$$Y = \left\{ y_j \mid y_j \in \left\{ \begin{array}{l} \text{Metodologías evaluadas para el} \\ \text{desarrollo de juegos serios y de software} \end{array} \right\} \wedge j \in \mathbb{Z}^+ \right\} \quad (8)$$

$$Y = D \quad (9)$$

$$h(\text{ntf idf}(t_i, d_j, k, t_i, D), k) = \text{El valor de } \text{ntf idf}(t_i, d_j, k, t_i, D) \text{ es mayor o igual a } k. \quad (10)$$

$$f: X \times Y \rightarrow \{0,1\} \quad (11)$$

$$f(x_i, y_j) = h(\text{ntf idf}(t_i, d_j, k, t_i, D), k) \quad (12)$$

$$[f(x_i, y_j)]_{ij} = \begin{cases} 1, & \text{if } f(x_i, y_j) \text{ is true} \\ 0, & \text{otherwise} \end{cases} \quad (13)$$

$$A = \begin{bmatrix} [f(x_1, y_1)]_{11} & [f(x_1, y_2)]_{12} & \dots & [f(x_1, y_n)]_{1n} \\ [f(x_2, y_1)]_{21} & [f(x_2, y_2)]_{22} & \dots & [f(x_2, y_n)]_{2n} \\ \dots & \dots & \dots & \dots \\ [f(x_m, y_1)]_{m1} & [f(x_m, y_2)]_{m2} & \dots & [f(x_m, y_n)]_{mn} \end{bmatrix} \quad (14)$$

Donde **A** es la matriz de evaluación.

2.1.3 Proceso Consecuente (Obtención de Criterios clave)

Las secciones 2.2.1 y 2.2.2, trabajan individualmente hasta obtener una misma matriz booleana al final de sus respectivos procesos.

En la presente sección, se tratará un proceso consecuente, en el cual se unen los dos procesos antecedentes, lo que se trata de evidenciar es la obtención de criterios relevantes, para ello, previamente se elaboran matrices de instanciación, matrices de comportamiento y finalmente las matrices de emparejamiento (ver Figura 14).

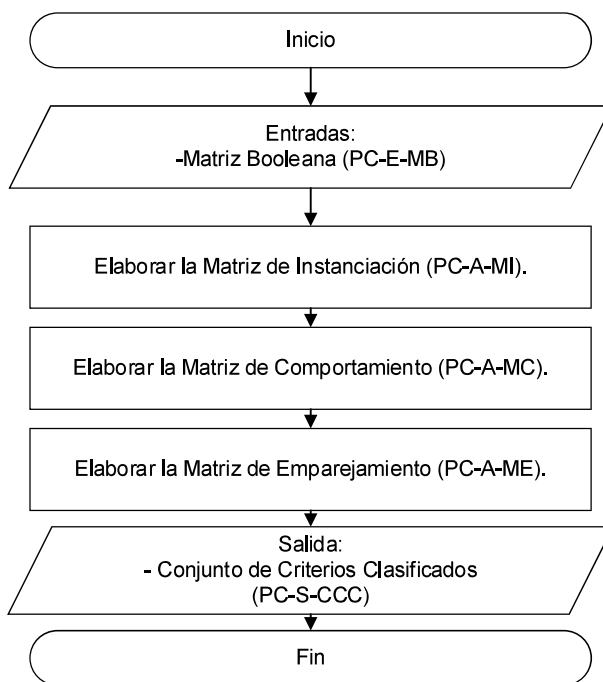


Figura 14 - Diagrama de Flujo del Proceso Consecuente
Elaborado por los autores

Matriz de Instanciación

Esta matriz busca primordialmente, determinar cuán relevante es un criterio en las metodologías estudiadas, debido a que se evalúa si el mismo, aparece o no en los documentos de estudio. La estructuración de esta matriz está conformada de la siguiente forma (ver Tabla 14):

- Por cada uno de los criterios definidos previamente, se elabora una matriz independiente.
- En la primera columna se coloca el número de la metodología que está siendo abordada.
- En la segunda columna, se colocan valores correspondientes a 0s o 1s, donde se explica si este criterio aparece o no en cada una de las metodologías analizadas.
- En la tercera columna, se identifica el contexto de estudio, en este caso, para juegos serios se tiene el valor 1 y si no, un valor de 0. Para utilizar el apoyo de lógica matemática, se define los contextos de la siguiente forma:
 - ✓ p, para juegos serios
 - ✓ q, para aquello que no involucra a juegos serios.

Tabla 14 - Matriz de Instanciación (PC-A-MI)

Numero de Metodología	Criterio analizado	Número de criterio analizado
1	$0 \vee 1$	$0 \vee 1$
2	$0 \vee 1$	$0 \vee 1$
3	$0 \vee 1$	$0 \vee 1$
n	$0 \vee 1$	$0 \vee 1$

El contenido de la tabla puede ser representado en lenguaje matemático (15-18) de la siguiente manera.

$$g: Y \rightarrow \{0,1\} \quad (15)$$

$$g(y_j) = \text{La metodología } y_j \text{ es una metodología para el desarrollo de juegos serios.} \quad (16)$$

$$[g(y_j)]_{j2} = \begin{cases} 1, & \text{if } g(y_j) \text{ is true} \\ 0, & \text{otherwise} \end{cases} \quad (17)$$

$$B_{i'} = \begin{bmatrix} a_{11} & [g(y_1)]_{12} \\ a_{21} & [g(y_2)]_{22} \\ \dots & \dots \\ a_{n1} & [g(y_n)]_{n2} \end{bmatrix} \quad (18)$$

Donde $B_{i'}$ es la matriz de instanciación por cada criterio y a_{ij} representa a los elementos de la matriz A .

Matriz de Comportamiento

Esta matriz se explica en tres columnas (ver Tabla 15). El objetivo de esta matriz es preparar la información para determinar los criterios clave.

Tabla 15 - Matriz de Comportamiento (PC-A-MC)

Tablas de Verdad		Criterio de Evaluación			
El criterio de estudio esté presente en los documentos "P"	La metodología evaluada es una metodología para el desarrollo de juegos serios "Q"	Criterio 1	Criterio 2	Criterio 3	Criterio n
1	1	$0 \vee 1$	$0 \vee 1$	$0 \vee 1$	$0 \vee 1$
1	0	$0 \vee 1$	$0 \vee 1$	$0 \vee 1$	$0 \vee 1$
0	1	$0 \vee 1$	$0 \vee 1$	$0 \vee 1$	$0 \vee 1$
0	0	$0 \vee 1$	$0 \vee 1$	$0 \vee 1$	$0 \vee 1$

La primera sección corresponde a las tablas de verdad o valores lógicos, y a su vez, está dividida en dos columnas.

La primera columna, se evalúa si un criterio dado, forma parte o no de una metodología dada, en la segunda columna, se evalúa si una metodología dada es o no una metodología para el desarrollo de juegos serios.

Para lo descrito anteriormente, se utilizan valores booleanos, 0s o 1s, para indicar “sí” en el caso del valor “1” y “no” en el caso del valor “0”.

Los valores que se colocan por defecto en estas dos columnas pertenecen a los valores de las tablas de verdad de la lógica matemática (Patricia, Marulanda, Augusto, & Torres, n.d.).

En la segunda sección de la Tabla 15, se colocan uno a uno los criterios que se están analizando. Los valores booleanos que se colocan en esta sección, por cada criterio, pertenecen a la asociación del par de valores que se obtienen y el par de valores que se obtienen en la primera sección.

Expresado de manera diferente, si cualquiera de las cuatro asociaciones o pares dadas por defecto en la Tabla 15, (0,0), (0,1), (1,0), (1,1) existen también como asociaciones en la Tabla 16, se coloca el valor 1, caso contrario el valor 0.

El contenido de la tabla puede ser representado en lenguaje matemático (19-22) de la siguiente manera.

$$R = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \quad (19)$$

$$P: \text{proposición de entrada} \quad (20)$$

$$[P] = \begin{cases} 1, & \text{if } P \text{ is true} \\ 0, & \text{otherwise} \end{cases} \quad (21)$$

$$C_i = \begin{bmatrix} [\exists b'_{i1} \exists b'_{i2} (b'_{i1} = r_{11} \wedge b'_{i2} = r_{12})]_{11} \\ [\exists b'_{i1} \exists b'_{i2} (b'_{i1} = r_{21} \wedge b'_{i2} = r_{22})]_{21} \\ [\exists b'_{i1} \exists b'_{i2} (b'_{i1} = r_{31} \wedge b'_{i2} = r_{32})]_{31} \\ [\exists b'_{i1} \exists b'_{i2} (b'_{i1} = r_{41} \wedge b'_{i2} = r_{42})]_{41} \end{bmatrix} \quad (22)$$

Donde C_i representa a la columna criterio de la matriz de comportamiento, donde $b'_{i'ij}$ representa a cada uno de los elementos de la matriz $B_{i'}$ y r_{ij} representa a cada uno de los elementos de la matriz R .

El vector C'_i (23) es una alternativa al vector C_i , ya que permite obtener los mismos resultados trabajando directamente con las funciones $f(x_i, y_j)$ y $g(y_j)$, sin considerar las matrices A , $B_{i'}$ y R .

$$C'_i = \begin{bmatrix} [\exists y_j (\neg f(x_i, y_j) \wedge \neg g(y_j))]_{11} \\ [\exists y_j (\neg f(x_i, y_j) \wedge g(y_j))]_{21} \\ [\exists y_j (f(x_i, y_j) \wedge \neg g(y_j))]_{31} \\ [\exists y_j (f(x_i, y_j) \wedge g(y_j))]_{41} \end{bmatrix} \quad (23)$$

Matriz de Emparejamiento

Para obtener la matriz de emparejamiento, en primer lugar, es necesario hacer uso de los patrones lógicos que denotarán posteriormente la relevancia de un criterio, a su vez, permitirán caracterizarlo como necesario y suficiente, suficiente y no necesario, necesario y no suficiente, no necesario y no suficiente y ninguna de los anteriores (ver Tabla 16).

A continuación, se explicará cada uno de las condiciones que se pueden obtener:

- Necesaria y no suficiente (ver Figura 15): Cuando un criterio está presente en todas las metodologías referentes a juegos serios y en al menos una de las metodologías de desarrollo de software. (En todos los P, y al menos en un Q).
- Necesario y suficiente (ver Figura 16): Cuando el criterio está presente en todas las metodologías referentes a juegos serios y en ninguna de las metodologías de desarrollo de software. (En todos los P, y en ningún Q).
- Suficiente y no necesaria (ver Figura 17): Cuando el criterio está presente en al menos una de las metodologías referentes a juegos serios y en ninguna de las metodologías de desarrollo de software. (En al menos un P, y en ningún Q).
- No necesario y no suficiente (ver Figura 18): Cuando el criterio está presente en al menos una de las metodologías referentes a juegos serios y de igual manera, al menos en una de las metodologías de desarrollo de software. (En al menos un P, y en al menos un Q).
- Ninguno: Cuando el emparejamiento de criterios no corresponde a ninguna de las clasificaciones anteriores.

Tabla 16 - Tabla de Patrones Lógicos

Valores lógicos (P) Metodologías Juegos Serios	Valores lógicos (Q) Metodologías de desarrollo de software.	Condición necesaria y no suficiente	Condición suficiente y no necesaria	Condición necesaria y suficiente
p	q	$q \rightarrow p$	$p \rightarrow q$	$p \leftrightarrow q$
1	1	1	1	1
1	0	1	0	0
0	1	0	1	0
0	0	1	1	1

Dados los siguientes conjuntos:

A: Conjunto de metodologías para el desarrollo de videojuegos.

B: Conjunto de metodologías para el desarrollo de juegos serios.

A través de teoría de conjuntos, es posible realizar este análisis de la siguiente manera:

Dado el siguiente conjunto:

C: Conjunto de metodologías que tienen al menos un criterio necesario.

Se tiene que cumplir una de las siguientes condiciones:

- Si el conjunto B está incluido en el conjunto A, el conjunto C está incluido en el conjunto A y el conjunto B está incluido en el conjunto C (24); entonces, los elementos que integran el conjunto C son los Criterios Necesarios y No Suficientes (ver Figura 15).

$$B \subset A, C \subset A \text{ y } B \subset C \quad (24)$$

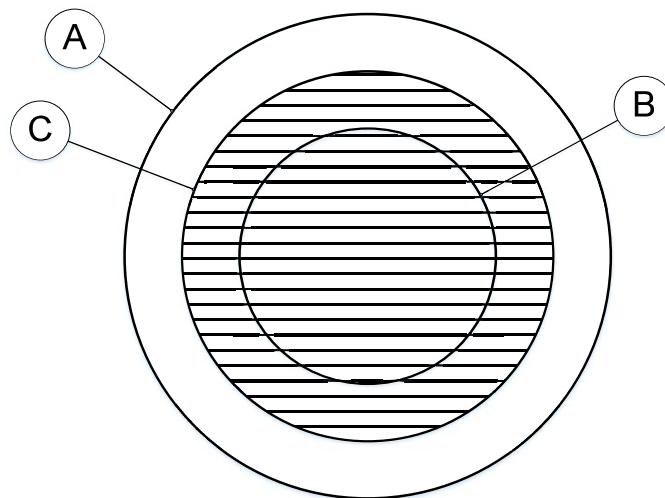


Figura 15 - Representación Gráfica de Criterios Necesarios y No Suficientes.
Elaborado por los autores

- Si el conjunto B está incluido en el conjunto A, el conjunto C está incluido en el conjunto A y el conjunto C es igual al conjunto B (25); entonces, los elementos que integran el conjunto C son los Criterios Necesarios y Suficientes (ver Figura 16).

$$B \subset A, C \subset A \text{ y } C = B \quad (25)$$

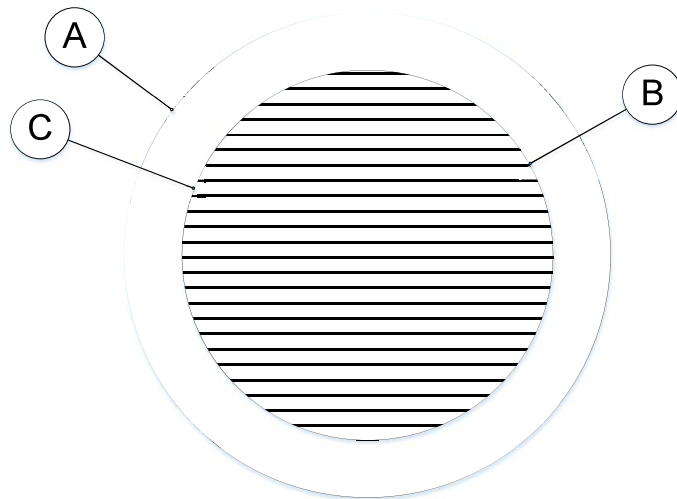


Figura 16 - Representación Gráfica de Criterios Necesarios y Suficientes

En cambio, si se sustituye el conjunto C anteriormente mencionado por el siguiente:

C: Conjunto de metodologías que tienen al menos un criterio suficiente.

Se tiene que cumplir una de las siguientes condiciones:

- Si el conjunto B está incluido en el conjunto A, el conjunto C está incluido en el conjunto A y el conjunto C está incluido en el conjunto B (26); entonces, los elementos que integran el conjunto C son los Criterios Suficientes y No Necesarios (ver Figura 17).

$$B \subset A, C \subset A \text{ y } C \subset B \quad (26)$$

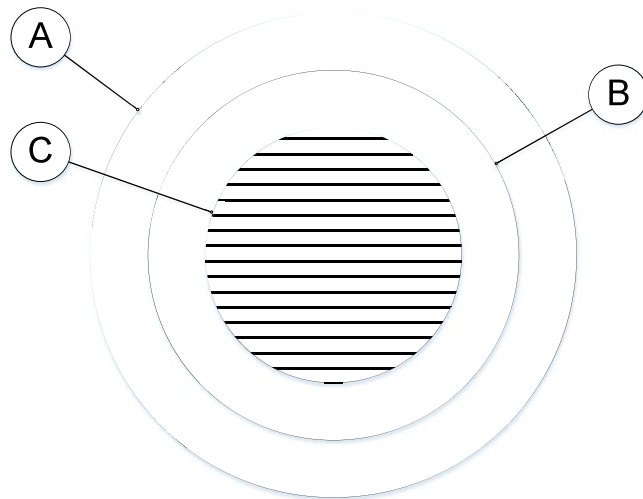


Figura 17 - Representación Gráfica de Criterios Suficientes y No Necesarios

- Si el conjunto B está incluido en el conjunto A, el conjunto C está incluido en el conjunto A, el conjunto B no está incluido en el conjunto C y el conjunto C no está incluido en el conjunto B (27); entonces, los elementos que integran el conjunto C son los Criterios No Suficientes y No Necesarios (ver Figura 17).

$$B \subset A, C \subset A, B \not\subset C \text{ y } C \not\subset B \quad (27)$$

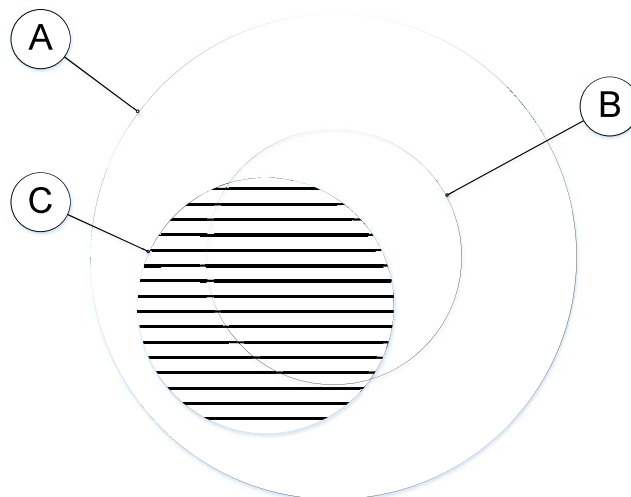


Figura 18 - Representación Gráfica de Criterios No Suficientes y No Necesarios

Por consiguiente, teniendo los valores de la Tabla 15 y la tabla de patrones lógicos (Tabla 16), si los valores de una tabla y otra se corresponden entre sí, se puede concluir en una de las condiciones anteriores (ver Tabla 17).

Tabla 17 - Matriz de Emparejamiento (PC-A-ME)

Criterios	Criterio 1	Criterio 2	Criterio 3	Criterio n
	$0 \vee 1$	$0 \vee 1$	$0 \vee 1$	$0 \vee 1$
	$0 \vee 1$	$0 \vee 1$	$0 \vee 1$	$0 \vee 1$
	$0 \vee 1$	$0 \vee 1$	$0 \vee 1$	$0 \vee 1$
Correspondencia Patrón Lógico	Necesario y no suficiente	Necesario y no suficiente	Necesario y no suficiente	Necesario y no suficiente
No necesario y suficiente	No necesario y suficiente	No necesario y suficiente	No necesario y suficiente	
Necesario y suficiente	Necesario y suficiente	Necesario y suficiente	Necesario y suficiente	
Ninguno	Ninguno	Ninguno	Ninguno	

El contenido de la tabla puede ser representado en lenguaje matemático (28) de la siguiente manera.

$$D = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} \quad E = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} \quad F = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (28)$$

Donde D representa el vector para que una condición sea necesaria y no suficiente, E para que sea no necesaria y suficiente y F para que se necesaria y suficiente.

$$W = \left\{ w \mid w \in \left\{ \begin{array}{l} \text{Condición necesaria y no suficiente,} \\ \text{Condición suficiente y no necesaria,} \\ \text{Condición necesaria y suficiente} \end{array} \right\} \right\} \quad (29)$$

$$f: V \times V \times V \times V \rightarrow W \quad (30)$$

$$M(C_i, D, E, F) = \begin{cases} \text{Condición necesaria y no suficiente, } & \text{if } C_i = D \\ \text{Condición no necesaria y suficiente, } & \text{if } C_i = E \\ \text{Condición necesaria y suficiente, } & \text{if } C_i = F \end{cases} \quad (31)$$

Donde M es la función de emparejamiento (29-31), encargada de determinar el tipo de criterio.

2.2 Conclusiones del Capítulo 2

En función de todo lo anteriormente expuesto, se puede concluir que el uso de diagramas de flujo y de tablas permite explicar a detalle cada uno de los pasos que conforman el algoritmo utilizado en la metodología.

De esta forma, los lectores que no se encuentren familiarizados con los conceptos matemáticos podrán tener una visión general del presente estudio y replicar el experimento para reafirmar o refutar los resultados.

2.2 Resumen del Capítulo 2

El presente capítulo indica de manera detallada la metodología utilizada en este proyecto de investigación, que consta de un proceso general, subdivido en tres subprocesos: proceso antecedente 1, referente a la opinión del investigador; el proceso antecedente 2, relativo al análisis ponderado, el cual hace uso de lógica matemática; y el proceso consecuente, el cual nos permite obtener las salidas finales, es decir, los criterios clasificados como necesarios y suficientes.

CAPITULO 3

3. EJECUCIÓN, RESULTADOS Y DISCUSIÓN

3.1 Ejecución de la metodología

Análogamente a los puntos expuestos en la metodología que se tomó como referencia, se explicará el proceso que se siguió en este estudio.

El tema general del trabajo puntualiza un estudio sobre metodologías de desarrollo de software en general, por un lado, y, por otro lado, un estudio de metodologías de desarrollo de juegos serios, mediante el cual, se pueda concluir en una serie de criterios clave de diseño de juegos serios educativos.

A partir de la delimitación del tema, se continúa con la formulación de criterios de búsqueda (keywords), los mismos que fueron ingresados en bibliotecas digitales, para posteriormente conseguir los artículos, documentos, libros o referencias bibliográficas que hagan hincapié a estas claves de búsqueda.

Las claves de búsqueda que se utilizaron para la obtención de documentos fueron las siguientes: serious games, development methodologies, agile methodologies, development of video games, educational games, serious game design, traditional methodologies.

Para la organización de documentos se utilizó la herramienta Mendeley, en donde a partir de obtención de documentos en el paso anterior, se ordenó la información en base a los criterios de búsqueda antes definidos. En este caso, se elaboró dos directorios:

- Metodologías de desarrollo de software, en donde se almacenó, todos los documentos que referenciaron a procesos, metodologías, frameworks ágiles, tradicionales, e híbridos, tanto para el desarrollo de software y el desarrollo de videojuegos.
- Metodologías de desarrollo de juegos serios, en donde se almacenó, todos los documentos que referenciaron a metodologías, procesos de desarrollo de juegos serios y software educativos.
- Las metodologías extraídas y estudiadas para ambos enfoques (ver Tabla 18), fueron recabadas en base al análisis exhaustivo de la documentación encontrada, seleccionándolas en base a la información que aportan.

Tabla 18 – Ejemplo - Conjunto de Documentos de las Metodologías (PG-E-CDM) (ver Anexo 1)

N°	Metodologías de desarrollo de juegos serios	N°	Metodologías de desarrollo de software
1	Asgdp (Cano et al., 2015)	8	Huddle (Morales et al., 2010)
2	Doodle (McMahon, 2009)	9	Sum (Acerenza et al., 2009)
3	Edos (Tran et al, 2010)	10	Game-Scrum (Barbosa & Godoy, 2010)
4	Emergo (Nadolski et al., 2008)	11	XP (Pressman & Maxim, 2019)
5	Esd (Çağatay, 2012)	12	Scrum (Schwaber & Sutherland, 2013)
6	Meconesis (Cano et al., 2016)	13	Crystal (Cockburn, 2001)

7	Saavedra (Barajas et al., 2014)	14	Extreme Game Development (Barreno & Guaraca, 2013)
		15	Game Unified Process (Gerónimo-Castillo et al., 2008)
		16	Metodologías Ágiles (Orjuela & Rojas, 2008)
		17	PU (Proceso Unificado) (Jacobson & Booch, 2001)
		18	RUP (Martínez, 2011)
		19	Waterfall Process (Morales et al., 2010)
		20	Essential Unified Process (Hui et al., 2015)
		21	Open UP (Balduino, 2007)
		22	Microsoft Solution Framework (Ze, 2014)
		23	Game development process (Gerónimo-Castillo et al., 2008)
		24	Metodología genérica para videojuegos (Petrillo & Pimenta, 2010)

En lo que respecta al desarrollo de la práctica, se puntualizó dividirla en base a dos aspectos:

- Enfoque 1: Se utilizaron todas las metodologías recabadas para ambos contextos (ver Tabla 18), es decir, se trabajó con un total de 24 metodologías.
- Enfoque 2: Se utilizaron las 7 metodologías para juegos serios, pero únicamente las metodologías de desarrollo de software y no las de videojuegos para el contexto de metodologías de desarrollo de software, con un total de 16 metodologías. En la Tabla 18, se muestran coloreadas las metodologías que fueron descartadas para este enfoque.
- A continuación, se muestra la conformación de la tabla base (ver Tabla 19), seleccionando un ejemplo de la tabla obtenida en el análisis (ver Anexo 1- Matriz de análisis).

Tabla 19 – Ejemplo – Conjunto de Criterios Base (PG-E-CCB) (ver Anexo 2)

Dimensiones	Criterios	Explicación ¿Qué se busca?	Sinónimos
Desarrollo de Juegos Serios Educativos	Roles	Si como parte del trabajo de desarrollo de los JS, se definen roles y funciones dentro del equipo. ¿Qué roles se proponen? ¿Qué funciones desempeñan?	Equipo de desarrollo
	Objetivos Pedagógicos	¿Se establecen objetivos pedagógicos para el desarrollo del juego serio? ¿Cómo se establecen los objetivos pedagógicos y escenarios pedagógicos a la hora de desarrollar un juego serio educativo?	Objetivos educativos, objetivos para el aprendizaje
	Gamificación	Con este criterio se busca identificar si durante desarrollo del juego serio educativo, se especifican mecanismos de gamificación, como se los define, diseña, incluye, que actividades son llevadas a cabo para ello, que entregables.	Ludificación, juguetización, lúdico, incrustación, actividades divertidas, elementos divertidos, dinámico, mecanismos de aprendizaje y entretenimiento, tratamiento del juego

En la Tabla 19, se visualiza la dimensión en la cual está centrada el estudio, en este caso el desarrollo de juegos serios educativos, los criterios, roles, objetivos pedagógicos y gamificación, además lo que se desea buscar en los documentos de estudio de cada uno de estos criterios y los sinónimos para cada uno de estos términos.

3.1.1 Enfoque 1

Proceso Antecedente 1 (Opinión del Investigador)

Como parte de los resultados, en el proceso antecedente 1, basado en la opinión del investigador, se elabora la matriz de justificación (ver Tabla 20).

Tabla 20 - Ejemplo Enfoque 1– Matriz de Justificación (PA1-A-MJ) (ver Anexo 3)

Dimensiones	Criterios	Justificación	Bibliografía
Desarrollo de juegos serios educativos	Objetivos pedagógicos	Dentro de los pilares conceptuales fundamentales está el aspecto educativo. Dentro de este aspecto se considera el modelo pedagógico y las estrategias didácticas a las cuales debe responder el software. En la etapa de diseño se encuentra el diseño educativo, aquel que involucra los objetivos pedagógicos. De igual manera se elabora un modelo pedagógico que vaya alineado con los procesos de enseñanza-aprendizaje y cumplir los objetivos trazados para el usuario final. Por otro lado, el modelo pedagógico debe estar estructurado de manera que se pueda acoplar a la implementación del software educativo en aspectos como los contenidos, el ambiente, o el contexto. Dentro de estos contenidos se contemplan más subcontenidos como aspectos metodológicos, aspectos organizacionales, estrategias de aprendizaje.	(Orjuela & Rojas, 2008)

A continuación, se elabora la matriz booleana como parte del análisis del investigador (ver Tabla 21). En color verde, se encuentran resaltados los documentos relacionados al desarrollo de software; en cambio, en color amarillo, se encuentran resaltadas las metodologías referentes al desarrollo de juegos serios.

La numeración presentada en la matriz booleana (ver Tabla 21) corresponde a los códigos asignados a cada una de las metodologías presentados en la Tabla 18.

Tabla 21 – Ejemplo Enfoque 1 - Matriz Booleana del Proceso Antecedente 1 (PA1-A-MB) (ver Anexo 5).

M. C.	1	2	3	4	5	6	7	8	9	10	11	12	13	...	24
10	1	1	1	1	1	1	1	0	0	0	0	0	0	...	0

Proceso Antecedente 2 (Análisis ponderado)

Como parte de los resultados, en el proceso antecedente 2, basado en el análisis ponderado, se comienza elaborando la matriz de análisis reducida (ver Tabla 22).

Tabla 22 - Ejemplo Enfoque 1- Matriz de Análisis Reducida (PA2-A-MAR) (ver Anexo 4)

Criterios	Sinónimos
Objetivos pedagógicos	Objetivos educativos, objetivos para el aprendizaje.

Posteriormente, se elabora la matriz de frecuencia absoluta (ver Tabla 23) para el criterio “Objetivos pedagógicos”.

Tabla 23 - Ejemplo Enfoque 1- Matriz de Frecuencia Absoluta (PA2-A-MFA) (ver Anexo 6)

M. C.	1	2	3	4	5	6	7	8	9	10	11	12	13	...	24
10	5	14	17	10	15	47	35	0	0	0	0	0	0	...	0

A continuación, se construye la matriz del producto $ntf*idf$ (ver Tabla 24).

Tabla 24 - Ejemplo Enfoque 1– Matriz NTF-IDF (PA2-A-MNTFIDF) (ver Anexo 6)

M. C.	1	2	3	4	5	6	7	8	9	10	11	12	13	...	24
10	0,0393	0,2341	0,1516	0,0823	0,1384	0,4123	0,2230	0	0	0	0	0	0	...	0

Finalmente, como parte del proceso antecedente 2 se elabora la matriz booleana (ver Tabla 25).

Tabla 25 - Ejemplo Enfoque 1– Matriz Booleana del Proceso Antecedente 2 (PA2-A-MB) (ver Anexo 6)

M. C.	1	2	3	4	5	6	7	8	9	10	11	12	13	...	24
10	1	1	1	1	1	1	1	0	0	0	0	0	0	...	0

Proceso Consecuente (Opinión del investigador)

Como parte de los resultados, en el proceso consecuente correspondiente a la opinión del investigador, se muestra un ejemplo para el criterio “Objetivos pedagógicos”. Este proceso inicia con la matriz de instanciación (ver Tabla 26).

Tabla 26 - Ejemplo Enfoque 1– Matriz de Instanciación de la Opinión del Investigador (PC-A-MI). (ver Anexo 5)

Objetivos pedagógicos	Documento	Contexto
0	0	P (Conjunto metodología para el desarrollo de juegos serios)
0	0	
...	...	
0	0	
1	1	Q (Documentos enfocados en metodologías para el desarrollo de juegos serios) (Conjunto de metodologías que no son para el desarrollo de juegos serios)
1	1	
1	1	
1	1	
1	1	
1	1	
1	1	

A continuación, se elabora la matriz de comportamiento (ver Tabla 27).

Tabla 27 - Ejemplo Enfoque 1– Matriz de Comportamiento de la Opinión del Investigador (PC-A-MC) (ver Anexo 5)

Tablas de Verdad		Criterio de Evaluación
El criterio de estudio esté presente en los documentos “P”	La metodología evaluada es una metodología para el desarrollo de juegos serios “Q”	Objetivos pedagógicos
0	0	1
0	1	0
1	0	0
1	1	1

Finalmente, se elabora la matriz de emparejamiento (ver Tabla 28) para el objetivo pedagógico.

Tabla 28 - Ejemplo Enfoque 1– Matriz de Emparejamiento de la Opinión del Investigador (PC-A-ME) (ver Anexo 5)

Criterios	Objetivos pedagógicos
	1
	0
	0
Correspondencia Patrón Lógico	Necesario y suficiente (ver Tabla 16)

Dicho patrón se compara con los valores de la tabla de verdad (ver Tabla 16):

En la siguiente matriz (ver Tabla 29), se muestra la clasificación de todos los criterios como necesarios o suficientes, de acuerdo a los análisis previos.

Tabla 29 – Ejemplo Enfoque 1- Conjunto de Criterios Clasificados por la Opinión del Investigador (PG-S-CCC). (ver Anexo 5)

Criterios y Sinónimos	Opinión del Investigador	Criterios y Sinónimos	Opinión del Investigador	Criterios y Sinónimos	Opinión del Investigador
Metodología	Necesario y no suficiente	Gameplay	Necesario y no suficiente	Técnicas aprendizaje	Ninguno
Ciclo de vida	Ninguno	Dispositivo de visualización	No necesario y no suficiente	Avatar	Ninguno
Proceso ágil	No necesario y no suficiente	Dispositivo de entrada o interacción	Ninguno	Diseño centrado en el Contexto de Uso	Ninguno
Recursos	No necesario y no suficiente	Retroalimentación desde producto de software	No necesario y no suficiente	Diseño aborda experiencia del Usuario	Ninguno
Escenario o historia	No necesario y no suficiente	Ajuste de dificultad	Ninguno	Usuarios finales	Ninguno
Prototipaje	No necesario y no suficiente	Usabilidad	Ninguno	Expertos	No necesario y no suficiente
Lenguaje de modelado	No necesario y no suficiente	Accesibilidad	Ninguno	Retroalimentación desde usuarios	No necesario y no suficiente
Pruebas	Necesario y no suficiente	Elementos visuales	Ninguno	Proceso Iterativo e incremental	Ninguno
Roles	Necesario y no suficiente	Elementos sonoros	Ninguno	Juego Lingüístico	Ninguno
Objetivos Pedagógicos	Necesario y suficiente	Retención corto plazo o largo plazo	Ninguno	Técnica Creativas	Ninguno

Gamificación	Necesario y suficiente	Test inicial	Ninguno	Mecanismo de consenso, acuerdo y representación	Ninguno
Calidad del Producto	No necesario y no suficiente	Ayuda Contextual	Ninguno		
Calidad de uso	No necesario y no suficiente	Adaptabilidad	Ninguno		

Proceso Consecuente (Análisis Ponderado)

Como parte de los resultados, en el proceso consecuente, se muestra un ejemplo para el criterio “Objetivos pedagógicos” y para el análisis ponderado (ver Tabla 30).

Tabla 30 - Ejemplo Enfoque 1 – Matriz de Instanciación del Método Ponderado (PC-A-MI) (ver Anexo 6)

Objetivos pedagógicos	Documento	Contexto
0	0	P (Documentos enfocados en metodologías para el desarrollo de software)
0	0	
...	...	
0	0	
1	1	Q (Documentos enfocados en metodologías para el desarrollo de juegos serios)
1	1	
1	1	
1	1	
1	1	
1	1	
1	1	

A continuación, se elabora la matriz de comportamiento (ver Tabla 31).

Tabla 31 - Ejemplo Enfoque 1– Matriz de Comportamiento del Método Ponderado (PC-A-MC) (ver Anexo 6)

Tablas de Verdad		Criterio de Evaluación
El criterio de estudio esté presente en los documentos “P”	La metodología evaluada es una metodología para el desarrollo de juegos serios “Q”	Objetivos pedagógicos
0	0	1
0	1	0
1	0	0
1	1	1

Finalmente, se elabora la matriz de emparejamiento (ver Tabla 32) para el objetivo pedagógico.

Tabla 32 - Ejemplo Enfoque 1– Matriz de Emparejamiento del Método Ponderado (PC-A-ME) (ver Anexo 6)

Criterios	Objetivos pedagógicos
	1
	0
	0
Correspondencia Patrón Lógico	Necesario y suficiente (ver Tabla 16)

En la siguiente matriz (ver Tabla 33), se muestra la clasificación de todos los criterios como necesarios o suficientes, de acuerdo a los análisis previos.

Tabla 33 - Ejemplo Enfoque 1- Conjunto de Criterios Clasificados por el Método Ponderado (PG-S-CCC) (ver Anexo 6)

Criterios y Sinónimos	Método Ponderado Promedio	Criterios y Sinónimos	Método Ponderado Promedio	Criterios y Sinónimos	Método Ponderado Promedio
Metodología	Ninguno	Gameplay	No necesario y no suficiente	Técnicas aprendizaje	Ninguno
Ciclo de vida	No necesario y no suficiente	Dispositivo de visualización	No necesario y no suficiente	Avatar	No necesario y no suficiente
Proceso ágil	No necesario y no suficiente	Dispositivo de entrada o interacción	No necesario y no suficiente	Diseño centrado en el Contexto de Uso	Ninguno
Recursos	Ninguno	Retroalimentación desde producto de software	No necesario y no suficiente	Diseño aborda experiencia del Usuario	No necesario y no suficiente
Escenario o historia	No necesario y no suficiente	Ajuste de dificultad	Suficiente y no necesario	Usuarios finales	No necesario y no suficiente
Prototipaje	No necesario y no suficiente	Usabilidad	No necesario y no suficiente	Expertos	No necesario y no suficiente
Lenguaje de modelado	No necesario y no suficiente	Accesibilidad	Suficiente y no necesario	Retroalimentación desde usuarios	No necesario y no suficiente
Pruebas	No necesario y no suficiente	Elementos visuales	No necesario y no suficiente	Proceso Iterativo incremental	Ninguno
Roles	No necesario y no suficiente	Elementos sonoros	No necesario y no suficiente	Juego Lingüístico	Ninguno
Objetivos Pedagógicos	Necesario y suficiente	Retención corto plazo o largo plazo	Ninguno	Técnica Creativas	Ninguno

Gamificación	Suficiente y no necesario	Test inicial	No necesario y no suficiente	Mecanismo de consenso, acuerdo y representación	Ninguno
Calidad del Producto	No necesario y no suficiente	Ayuda Contextual	Ninguno		
Calidad de uso	No necesario y no suficiente	Adaptabilidad	Suficiente y no necesario		

3.1.2 Enfoque 2

Proceso Antecedente 1 (Opinión del Investigador)

Como parte de los resultados, en el proceso antecedente 1, basado en la opinión del investigador, se elabora la matriz de justificación (ver Tabla 34).

Tabla 34 – Ejemplo Enfoque 2 – Matriz de Justificación (PA1-A-MJ) (ver Anexo 3)

Dimensiones	Criterios	Justificación	Bibliografía
Desarrollo de juegos serios educativos	Objetivos pedagógicos	Dentro de los pilares conceptuales fundamentales está el aspecto educativo. Dentro de este aspecto se considera el modelo pedagógico y las estrategias didácticas a las cuales debe responder el software. En la etapa de diseño se encuentra el diseño educativo, aquel que involucra los objetivos pedagógicos. De igual manera se elabora un modelo pedagógico que vaya alineado con los procesos de enseñanza-aprendizaje y cumplir los objetivos trazados para el usuario final. Por otro lado, el modelo pedagógico debe estar estructurado de manera que se pueda acoplar a la implementación del software educativo en aspectos como los contenidos, el ambiente, o el contexto. Dentro de estos contenidos se contemplan más subcontenidos como aspectos metodológicos, aspectos organizacionales, estrategias de aprendizaje (Orjuela & Rojas ,2008).	(Orjuela & Rojas ,2008)

A continuación, se elabora la matriz booleana como parte del análisis del investigador (ver Tabla 35). En color verde, se encuentran resaltados los documentos relacionados al desarrollo de software; en cambio, en color amarillo, se encuentran resaltadas las metodologías referentes al desarrollo de juegos serios.

Tabla 35 – Ejemplo Enfoque 2 - Matriz Booleana del Proceso Antecedente 1 (PA1-A-MB) (ver Anexo 7)

M. C.	1	2	3	4	5	6	7	8	9	10	11	12	13	...	24
10	1	1	1	1	1	1	1	0	0	0	0	0	0	...	0

Proceso Antecedente 2 (Análisis Ponderado)

Como parte de los resultados, en el proceso antecedente 2, basado en el análisis ponderado, se elabora la matriz de análisis reducida (ver Tabla 36).

Tabla 36 - Ejemplo Enfoque 2 - Matriz de Análisis Reducida (PA2-A-MAR) (ver Anexo 4)

Criterios	Sinónimos
Objetivos pedagógicos	Objetivos educativos, objetivos para el aprendizaje.

Posteriormente, se elabora la matriz de frecuencia absoluta (ver Tabla 37) para el criterio “Objetivos pedagógicos”.

Tabla 37 – Ejemplo Enfoque 2 - Matriz de Frecuencia Absoluta (PA2-A-MFA) (ver Anexo 8)

M. C.	1	2	3	4	5	6	7	8	9	10	11	12	13	...	24
10	5	14	17	10	15	47	35	0	0	0	0	0	0	...	0

A continuación, se construye la matriz del producto ntf x idf (ver Tabla 38).

Tabla 38 - Ejemplo Enfoque 2– Matriz de NTF-IDF (PA2-A-MNTFID) (ver Anexo 8)

M. C.	1	2	3	4	5	6	7	8	9	10	11	12	13	...	24
10	0,0264	0,1571	0,1017	0,0552	0,0929	0,2766	0,1496	0	0	0	0	0	0	...	0

Finalmente, como parte del proceso antecedente 2 se elabora la matriz booleana (ver Tabla 39).

Tabla 39 – Ejemplo Enfoque 2 – Matriz Booleana del Proceso Antecedente 2 (PA-A-MB) (ver Anexo 8)

M. C.	1	2	3	4	5	6	7	8	9	10	11	12	13	...	24
10	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Proceso Consecuente (Opinión del Investigador)

Como parte de los resultados, en el proceso consecuente, se muestra un ejemplo para el criterio “Objetivos pedagógicos” y para el análisis ponderado (ver Tabla 40).

Tabla 40 – Ejemplo Enfoque 2 – Matriz de Instanciación de la Opinión del Investigador (PC-A-MI) (ver Anexo 7)

Objetivos pedagógicos	Documento	Contexto
0	0	P (Documentos enfocados en metodologías para el desarrollo de software)
0	0	
...	...	
0	0	
1	1	Q (Documentos enfocados en metodologías para el desarrollo de juegos serios)
1	1	
1	1	
1	1	
1	1	
1	1	

A continuación, se elabora la matriz de comportamiento (ver Tabla 41).

Tabla 41 - Ejemplo Enfoque 2 – Matriz de Comportamiento de la Opinión del Investigador (PC-A-MC) (ver Anexo 7)

Tablas de Verdad		Criterio de Evaluación
El criterio de estudio esté presente en los documentos “P”	La metodología evaluada es una metodología para el desarrollo de juegos serios “Q”	Objetivos pedagógicos
0	0	1
0	1	0
1	0	0
1	1	1

Finalmente, se elabora la matriz de emparejamiento (ver Tabla 42) para el objetivo pedagógico.

Tabla 42 - Ejemplo Enfoque 2– Matriz de Emparejamiento de la Opinión del Investigador (PC-A-ME) (ver Anexo 7)

Criterios	Objetivos pedagógicos
	1
	0
	0
Correspondencia Patrón Lógico	Necesario y suficiente (ver Tabla 16)

Dicho patrón se compara con los valores de la tabla de verdad (ver Tabla 16):

En la siguiente matriz (ver Tabla 43), se muestra la clasificación de todos los criterios como necesarios o suficientes, de acuerdo a los análisis previos.

Tabla 43 – Ejemplo Enfoque 2- Conjunto de Criterios Clasificados por la Opinión del Investigador (PG-S-CCC) (ver Anexo 7)

Criterios y Sinónimos	Opinión del Investigador	Criterios y Sinónimos	Opinión del Investigador	Criterios y Sinónimos	Opinión del Investigador
Metodología	Ninguno	GamePlay	Necesario y no suficiente	Técnicas aprendizaje	Ninguno
Ciclo de vida	Ninguno	Dispositivo de visualización	No necesario y no suficiente	Avatar	Ninguno
Proceso ágil	No necesario y no suficiente	Dispositivo de entrada o interacción	Ninguno	Diseño centrado en el Contexto de Uso	Ninguno
Recursos	Ninguno	Retroalimentación desde producto de software	No necesario y no suficiente	Diseño aborda experiencia del Usuario	Ninguno
Escenario o historia	No necesario y no suficiente	Ajuste de dificultad	Ninguno	Usuarios finales	Ninguno
Prototipaje	No necesario y no suficiente	Usabilidad	Ninguno	Expertos	No necesario y no suficiente
Lenguaje de modelado	No necesario y no suficiente	Accesibilidad	Ninguno	Retroalimentación desde usuarios	No necesario y no suficiente
Pruebas	Necesario y no suficiente	Elementos visuales	Ninguno	Proceso Iterativo e incremental	Ninguno
Roles	Necesario y no suficiente	Elementos sonoros	Ninguno	Juego Lingüístico	Ninguno

Objetivos Pedagógicos	Necesario y suficiente	Retención corto plazo o largo plazo	Ninguno	Técnica Creativas	Ninguno
Gamificación	Necesario y suficiente	Test inicial	Ninguno	Mecanismo de consenso, acuerdo y representación	Ninguno
Calidad del Producto	No necesario y no suficiente	Ayuda Contextual	Ninguno		
Calidad de uso	Ninguno	Adaptabilidad	Ninguno		

Proceso Consecuente (Análisis Ponderado)

Como parte de los resultados, en el proceso consecuente, se muestra un ejemplo para el criterio “Objetivos pedagógicos” y para el análisis ponderado (ver Tabla 44).

Tabla 44 - Ejemplo Enfoque 2 – Matriz de Instanciación del Método Ponderado (PC-A-MI) (ver Anexo 8)

Objetivos pedagógicos	Documento	Contexto
0	0	P (Documentos enfocados en metodologías para el desarrollo de software)
0	0	
...	...	
0	0	
1	1	Q (Documentos enfocados en metodologías para el desarrollo de juegos serios)
1	1	
1	1	
1	1	
1	1	
1	1	
1	1	

A continuación, se elabora la matriz de comportamiento (ver Tabla 45).

Tabla 45 - Ejemplo Enfoque 2– Matriz de Comportamiento del Método Ponderado (PC-A-MC) (ver Anexo 8)

Tablas de Verdad		Criterio de Evaluación
El criterio de estudio esté presente en los documentos “P”	La metodología evaluada es una metodología para el desarrollo de juegos serios “Q”	Objetivos pedagógicos
0	0	1
0	1	0
1	0	0
1	1	1

Finalmente, se elabora la matriz de emparejamiento (ver Tabla 46) para el objetivo pedagógico.

Tabla 46 - Ejemplo Enfoque 2 – Matriz de Emparejamiento del Método Ponderado (PC-A-ME) (ver Anexo 8)

Criterios	Objetivos pedagógicos
	1
	0
	0
Correspondencia Patrón Lógico	Necesario y suficiente (ver Tabla 16)

Dicho patrón se compara con los valores de la tabla de verdad (ver Tabla 16):

Como parte de los resultados, en el proceso consecuente, los procesos fueron clasificados como se muestran a continuación (ver Tabla 47).

Tabla 47 - Ejemplo Enfoque 2- Conjunto de Criterios Clasificados por el Método Ponderado (PG-S-CCC) (ver Anexo 8)

Criterios y Sinónimos	Método Ponderado Promedio	Criterios y Sinónimos	Método Ponderado Promedio	Criterios y Sinónimos	Método Ponderado Promedio
Metodología	Ninguno	Gameplay	Suficiente y no necesario	Técnicas aprendizaje	Ninguno
Ciclo de vida	No necesario y no suficiente	Dispositivo de visualización	No necesario y no suficiente	Avatar	Suficiente y no necesario
Proceso ágil	No necesario y no suficiente	Dispositivo de entrada o interacción	Suficiente y no necesario	Diseño centrado en el Contexto de Uso	Ninguno
Recursos	Ninguno	Retroalimentación desde producto de software	No necesario y no suficiente	Diseño aborda experiencia del Usuario	No necesario y no suficiente
Escenario o historia	No necesario y no suficiente	Ajuste de dificultad	Suficiente y no necesario	Usuarios finales	No necesario y no suficiente
Prototipaje	No necesario y no suficiente	Usabilidad	No necesario y no suficiente	Expertos	No necesario y no suficiente
Lenguaje de modelado	No necesario y no suficiente	Accesibilidad	Suficiente y no necesario	Retroalimentación desde usuarios	Ninguno
Pruebas	No necesario y no suficiente	Elementos visuales	Suficiente y no necesario	Proceso Iterativo e incremental	Ninguno
Roles	No necesario y no suficiente	Elementos sonoros	Suficiente y no necesario	Juego Lingüístico	Ninguno

Objetivos Pedagógicos	Necesario y suficiente	Retención corto plazo o largo plazo	Ninguno	Técnica Creativas	Ninguno
Gamificación	Suficiente y no necesario	Test inicial	No necesario y no suficiente	Mecanismo de consenso, acuerdo y representación	Ninguno
Calidad del Producto	Ninguno	Ayuda Contextual	Ninguno		
Calidad de uso	Suficiente y no necesario	Adaptabilidad	Suficiente y no necesario		

3.2. Resultados y Discusión

Enfoque 1

En el enfoque 1, se obtuvo como criterio necesario y suficiente los “Objetivos pedagógicos”. Esto significa que este criterio se encuentra presente en todas las metodologías de desarrollo de juegos serios y se identifica en otras metodologías.

Dado un conjunto de metodologías de desarrollo, siempre que en alguna de estas se haga mención a los “Objetivos pedagógicos” está será condición **suficiente** para concluir que dicha metodología está enfocada en el desarrollo de juegos serios.

A su vez, al ser una condición **necesaria**, se puede inferir que no existirá ninguna metodología enfocada en el desarrollo de juegos serios que no haga mención a dicho criterio.

Los criterios que entran en la categoría de suficientes y no necesarios, en el enfoque 1, son la gamificación, el ajuste de dificultad, la accesibilidad y la adaptabilidad.

Al ser suficientes, significa que la sola presencia de alguno de estos criterios en alguna de las metodologías estudiadas permite inferir con un nivel de seguridad razonable que la metodología analizada está enfocada en el desarrollo de juegos serios.

En cambio, al ser criterios no necesarios, significa que existirán algunas metodologías que no tendrán estos criterios y que, de igual forma, podrán ser clasificados como metodologías para el desarrollo de juegos serios.

Con respecto al método ponderado basado en varianza, se obtuvieron resultados similares.

Las diferencias radican en que la gamificación entra en la categoría de necesario y suficiente; mientras que, la adaptabilidad ya no es considerada como suficiente y no necesaria.

En la Tabla 48, se puede apreciar a mayor detalle lo antes mencionado.

Tabla 48 - Validación de los Resultados del Enfoque 1 con la Varianza para los Criterios Necesarios o Suficientes (ver Anexo 9)

Criterios y Sinónimos	Análisis completo			
	Número de veces que aparece el criterio promedio	Método Ponderado Promedio	Número de veces que aparece el criterio varianza	Método Ponderado Varianza
Objetivos Pedagógicos		necesario y suficiente		necesario y suficiente
Gamificación	6/7	suficiente y no necesario		necesario y suficiente
Ajuste de dificultad	1/7	suficiente y no necesario	1/7	suficiente y no necesario
Accesibilidad	3/7	suficiente y no necesario	3/7	suficiente y no necesario
Adaptabilidad	3/7	suficiente y no necesario		no necesario y no suficiente

Enfoque 2

En el enfoque 2, se obtuvo como criterio necesario y suficiente los “Objetivos pedagógicos”.

Esto significa que este criterio se encuentra presente en todas las metodologías de desarrollo de juegos serios y en ninguna de las metodologías que no sea para el desarrollo de juegos serios.

Dado un conjunto de metodologías de desarrollo, siempre que en alguna de estas se haga mención a los “Objetivos pedagógicos” está será condición **suficiente** para concluir que dicha metodología está enfocada en el desarrollo de juegos serios.

Los criterios que entran en la categoría de suficientes y no necesarios, en el enfoque 2, son la gamificación, calidad de uso, gameplay, dispositivos de entrada o interacción, ajuste de dificultad, el ajuste de dificultad, la accesibilidad, la adaptabilidad y el avatar.

Con respecto al método ponderado basado en varianza, se obtuvieron resultados similares. Las diferencias radican en que la gamificación y el gameplay entran en la categoría de necesario y suficiente; mientras que, la adaptabilidad y el avatar ya no son consideradas como suficientes y no necesarios.

En la Tabla 49, se puede apreciar a mayor detalle lo antes mencionado.

Tabla 49 - Validación de Resultados del Enfoque 2 con la Varianza para los Criterios Necesarios o Suficientes (ver Anexo 9)

	Análisis No completo			
Criterios y Sinónimos	Número de veces que aparece el criterio promedio	Método Ponderado Promedio	Número de veces que aparece el criterio varianza	Método Ponderado Varianza
Objetivos Pedagógicos		necesario y suficiente		necesario y suficiente
Gamificación	5/7	suficiente y no necesario		necesario y suficiente
Calidad de uso	2/7	suficiente y no necesario		no necesario y no suficiente
GamePlay	4/7	suficiente y no necesario		necesario y suficiente
Dispositivo de entrada o interacción	2/7	suficiente y no necesario		no necesario y no suficiente
Ajuste de dificultad	1/7	suficiente y no necesario	1/7	suficiente y no necesario
Accesibilidad	2/7	suficiente y no necesario	3/7	suficiente y no necesario
Elementos visuales	2/7	suficiente y no necesario	2/7	suficiente y no necesario
Elementos sonoros	3/7	suficiente y no necesario	4/7	suficiente y no necesario
Adaptabilidad	2/7	suficiente y no necesario		no necesario y no suficiente
Avatar	3/7	suficiente y no necesario		no necesario y no suficiente

3.3 Conclusiones del Capítulo 3

El estudio fue dividido en dos enfoques con el objetivo de poder identificar las diferencias que ambos arrojan en sus resultados.

Estos resultados indican que los criterios que fueron catalogados como necesarios y suficientes y suficientes y no necesarios para el enfoque1, son en número, menor a los mostrados en el enfoque 2.

La razón es que, al no considerar las metodologías de desarrollo de software tradicional y ágil para el segundo análisis, se logra descartar criterios que solo mantienen estas

metodologías y a su vez, mantener, ratificar y fortalecer, aquellos criterios que mantienen solo las metodologías de videojuegos y juegos serios.

En otras palabras, cuando se habla de videojuegos y juegos serios, se pueden notar criterios comunes, debido a que ambos responden a aspectos como la diversión, la jugabilidad, etc., por otro lado, las metodologías de desarrollo de software no consideran estos aspectos ya que están pensada para varios tipos de software y no específicamente para los videojuegos.

En los dos análisis se pudo obtener como criterios imprescindibles, a los objetivos pedagógicos y la gamificación.

Como se analizó en este documento, muchos juegos serios carecen de entretenimiento siendo más académicos, existen otros, cuyos objetivos de aprendizaje no llegan al estudiante, convirtiéndose en un videojuego tradicional, en otros casos, juegos serios que no tienen el seguimiento adecuado del participante ni un modelo de aprendizaje que permita evaluar los avances del estudiante.

Respondiendo a esta problemática, se considera que todo proceso de desarrollo de debe involucrar al contexto pedagógico y de gamificación.

La estructuración del análisis en dos enfoques, permitió obtener un criterio más y que a la hora de diseñar un juego serio es muy importante, el criterio “gameplay”. Este criterio pudo ser encontrado gracias a una mejora en el nivel de sensibilidad para la detección de criterios, al implementar la varianza como parte de los cálculos.

El factor interacción estudiante-software requiere ser incorporado en cualquier metodología de diseño de juegos serios, debido a que esto permitirá que el jugador capte su atención y concentración en el juego, a la vez que se mantiene entretenido y aprende según los objetivos de estudio.

El análisis nombrado como método ponderado, permitió brindar resultados más confiables, por el hecho de que se utilizaron algunos conocimientos matemáticos, lo que además sugiere que se pueda usar diferentes métodos estadísticos que soporte y aporte con mejoras a la metodología aquí desarrollada, no obstante, el método nombrado como opinión del investigador, al ser un poco más subjetivo, no cayó en el campo de lo absurdo, incluso sirvió de base para las confrontaciones entre ambos procesos.

3.4 Resumen Capítulo 3

Finalmente, en este capítulo se muestra una aplicación resumida de la metodología del presente proyecto.

Para esto, se desarrollaron cada uno de los pasos de manera detallada con el criterio “objetivos pedagógicos”, al final de cada uno de los procesos, se muestra una tabla de resumen general que incluye a todos los criterios y cómo estos fueron clasificados.

CAPITULO 4

4. CONCLUSIONES

En el presente trabajo, se realizó un estudio de las metodologías de desarrollo de software y juegos, con la finalidad de estudiar su marco de referencia para su caracterización, analizarlas para la identificación de patrones característicos distintivos e inferir criterios para la elaboración de un perfil para la creación de nuevas metodologías de desarrollo de juegos serios.

Para lograr demostrarlo, se realizó una búsqueda sistemática de literatura que permitió determinar el conjunto de metodologías que formarían parte del estudio, se propuso una metodología como parte del análisis y se clasificó finalmente a los criterios identificados como necesarios y suficientes, necesarios y no suficientes, suficientes y no necesarios, y no necesarios ni suficientes.

Para validar este trabajo, se repitió el experimento varias veces, usando diferentes herramientas y en distintos escenarios, obteniéndose resultados similares.

El conjunto de criterios debidamente clasificados, obtenidos al final de la presente investigación revelan los aspectos más relevantes que se deben considerar al momento de crear nuevas metodologías enfocadas en el desarrollo de juegos serios.

Las metodologías que fueron consideradas para el estudio permitieron dividirlo, de tal manera que se obtuvo dos enfoques, como se pudo evidenciar, gracias a la caracterización de cada una de ellas mediante la estructuración de la tabla base. Gracias a esto, los criterios que se obtuvieron en los resultados, producto de la aplicación de la metodología resultaron ser coherentes para un correcto diseño de juegos serios.

Los criterios más importantes a considerar para el desarrollo de juegos serios son los objetivos pedagógicos, la gamificación y el gameplay. La relación de cada uno de estos criterios con las metodologías para el desarrollo de juegos serios se puede apreciar en la siguiente tabla (ver Tabla 50):

Tabla 50 – Relación entre los criterios y las metodologías de desarrollo de juegos serios

Criterios necesarios y suficientes	Metodologías para el desarrollo de juegos serio
Objetivos pedagógicos	Es el punto partida para el desarrollo de juegos serios. Están orientados a satisfacer las necesidades de aprendizaje de los usuarios finales.
Gamificación	Hace posible el aprovechamiento de las características de los videojuegos tradicionales en contextos no lúdicos, como el de los juegos serios.
Gameplay	Es indispensable para que el videojuego pueda existir como tal. Determina como se dan las interacciones entre el jugador y el juego serio.

5. REFERENCIAS BIBLIOGRÁFICAS

- Abdelali, S., Mateu, S., Imma, B., Fatiha, E., & Mohammed, B. (2016). Improving serious game design through a descriptive classification: A comparison of methodologies. *Journal of Theoretical and Applied Information Technology*, 92(1), 130–143.
- Acerenza, N. Coppes, A., Mesa, G., Viera, A., Fernández, E., Lorenzo T. & Vallespir, D. (2009). Una Metodología para Desarrollo de Videojuegos. (Asse), 171–176.
- Aleem, S., Capretz, L. F., & Ahmed, F. (2016). Game development software engineering process life cycle: a systematic review. *Journal of Software Engineering Research and Development*, 4(1), 6. <https://doi.org/10.1186/s40411-016-0032-7>
- Balduino, R. (2007). Introduction to OpenUP (Open Unified Process). Organization, 1–9. Retrieved from <https://eclipse.org/epf/general/OpenUP.pdf>
- Barajas, A., Álvarez, F., Muñoz, J., Santaolaya, R. & Collazos, C. (2014). A serious game development process using competency approach: Case Study: Elementary School Math. *Proceedings of the XV International Conference on Human Computer Interaction (Interacción '14)*, Article 99, 1–9. <https://doi.org/10.1145/2662253.2662352>
- Barbosa, E. F., & Godoy, A. (2010). Game-Scrum: An Approach to Agile Game Development.
- Bardin, L. (1977). *L'Analyse de contenu*, France: Presses Universitaires de France.
- Barreno, G. A. A., & Guaraca, M. G. S. (2013). Adaptación De Las Metodologías Ágiles Scrum Y Extreme Game Development En Una Metodología Para Desarrollo De Videojuegos En Android. *Caso Práctico: Desarrollo De Un Videojuego*. 33–34.
- Barwise, J. (1977). *Handbook of mathematical logic*, Amsterdam, Netherlands: North Holland.
- Çağatay, M. (2012). A methodological approach for serious game software development: an application for language disorders. Atilim University.
- Cano, S., Arteaga, J., Collazos, C., González, C. & Zapata, S. (2016). Towards a Methodology for Serious Games Design for Children with Auditory Impairments. *IEEE Latin America Transactions*, 14(5), 2511 – 2521. <https://doi.org/10.1109/TLA.2016.7530453>
- Cano, S., González, C., Collazos, C., Muñoz, J. & Zapata, S. (2015). Agile Software Development Process Applied to the Serious Games Development for Children from 7 to 10 Years Old. *International Journal of Information Technologies and Systems Approach*, 8(29), 64-79. <https://doi.org/10.4018/IJITSA.2015070105>
- Carrión, M., Santórum, M., Aguilar J. & Pinaida, A. (2019). Study to Infer Key Criteria for the Design of Serious Games. 2019 International Conference on Information

Systems and Software Technologies (ICI2ST), 63-70.
<https://doi.org/10.1109/ICI2ST.2019.00016>

- Carrión, M., Santórum, M., Pérez, M. & Aguilar, J. (2017). A Participatory Methodology for the Design of Serious Games in the Educational Environment. 2017 Congreso Internacional de Innovación y Tendencias en Ingeniería (CONIITI), 1-6.
<https://doi.org/10.1109/CONIITI.2017.8273363>
- Cockburn, A. (2001). Agile Software Development.
- Denning, P. (1997). A New Social Contract for Research, Communications of the ACM, 40(2), pp. 132-134.
- Duran, S. K., & Fajardo, M. A. (2012). Capítulo 3: Afrontando los retos. In GINDEV: Evaluación de la metodología Scrum en el desarrollo de videojuegos independientes (pp. 28–50). Puebla.
- Gerónimo-Castillo, G., Fernández-y-Fernández, C. A., & Ruiz-Rodríguez, R. (2008). Evolución del proceso de desarrollo de videojuegos en la iniciativa académica EDUMÓVIL. 7th Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería Del Conocimiento 2008, JIISIC 2008, 163–170. Retrieved from <http://www.scopus.com/inward/record.url?eid=2-s2.0-84883121023&partnerID=tZOtx3y1>
- Gómez, M. (2007). Buenas Prácticas en la Creación de Serious Games (Objetos de Aprendizaje Reutilizables). Post-Proceedings Del IV Simposio Pluridisciplinar Sobre Diseño, Evaluación y Desarrollo de Contenidos Educativos Reutilizables, SPDECE, 19–21. Retrieved from <http://spdece07.ehu.es/actas/S?nchez.pdf>
- Gregor S. & Hevner, A. (2013). Positioning and presenting Design Science Research for maximum impact, MIS Quarterly, 37(2), pp. 337-355.
- Hevner, A., March, S., Park J. & Ram, S. (2004). Design Science in Information Systems Research, MIS Quarterly, 28(1), pp. 75-105.
- Hui, Y., Yan, Y., Quanyu, W., & Zhiwen, C. (2015). Compare Essential Unified Process (EssUP) with Rational Unified Process (RUP). Proceedings of the 2015 10th IEEE Conference on Industrial Electronics and Applications, ICIEA 2015, 472–476.
<https://doi.org/10.1109/ICIEA.2015.7334159>
- Jacobson, I. & Booch, J. R. (2001). The Unified Process. (June), 96–103.
- Jiménez, J. H. (2012). Metodologías híbridas para desarrollo de software México. (Cic), 5. Retrieved from http://www.iis.org/CDs2012/CD2012IMC/CICIC_2012/PapersPdf/CB153YB.pdf
- Mader, S. (2015). Le “game design” de jeux thérapeutiques : Modèles et méthodes pour la conception du gameplay. Modélisation et simulation. Conservatoire national des arts et métiers – CNAM.
- March, S. & Smith, G. (1995). Design and natural science research on information technology, Decision Support Systems, 15, pp. 251-266.

- Martínez, A. (2011). Guía a Rational Unified Process. 15. Retrieved from https://www.researchgate.net/publication/268005509_Guia_a_Rational_Unified_Process
- McMahon, M. (2009). Using the DODDEL model to teach serious game design to novice designers.
- Michael, D. & Chen, S. (2006). Serious Games: games that educate, train and inform.
- Morales, G., Nava, C., Fernández, L., & Rey, M. (2010). Procesos de desarrollo para videojuegos. CULCyT, 36(37), 25–39. Retrieved from <http://dialnet.unirioja.es/servlet/articulo?codigo=3238114&info=resumen&idioma=SPA>
- Morales, O. A. (2015). Fundamentos de la Investigación Documental y la Monografía. En Manual para la elaboración y presentación de la monografía (Norelkys Espinoza y Ángel Rincón, Editores). Mérida, Venezuela: Grupo Multidisciplinario de Investigac. (January). Retrieved from https://www.researchgate.net/publication/237611564_FUNDAMENTOS_DE_LA_INVESTIGACION_DOCUMENTAL_Y_LA_MONOGRAFIA/download
- Nadolski, R., Hummel, H., Van den Brink, H., Hoefakker, R., Slootmaker, A., Kurvers, H. & Storm, J. (2008). EMERGO methodology and toolkit for efficient development of serious games in higher education. Simulation & Gaming, 39(3), 338–352. <https://doi.org/10.1177/1046878108319278>
- Orjuela, A. & Rojas, M. (2008). The Methodologies of Agile Development like an Opportunity for the Engineering of Educative Software. 5(2).
- Patricia, L., Marulanda, O., Augusto, S., & Torres, C. (n.d.). Fundamentos de la lógica matemática.
- Petrillo, F., & Pimenta, M. (2010). Is Agility out there? Agile Practices in Game Development.
- Pressman, R. & Maxim, B. (2019). Software engineering: a practitioner's approach (9th ed.). McGraw-Hill Education.
- Rogers, S. (2010). Level up! The guide to great video game design.
- Ruiz-Rodríguez, R., & Fernández-y-Fernández, C. A. (2007). Towards a Design Process for Didactic Game Development: experiences and proposals of the Edumóvil project. (November), 7.
- Schwaber, K., & Sutherland J. (2013). La Guía de Scrum. 1, 21. Retrieved from <http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Spanish.pdf#zoom=100>
- Simon, H. (1996). The Sciences of the Artificial, Cambridge: MIT Press.
- Singhal, A., Salton, G., Mitra, M., & Buckley, C. (1996). Document length normalization. Information Processing and Management, 32(5), 619–633. [https://doi.org/10.1016/0306-4573\(96\)00008-8](https://doi.org/10.1016/0306-4573(96)00008-8)

- Tran, C., George, S., & Marfisi-Schottman, I. (2010). EDoS: An authoring environment for serious games design based on three models, 4th European Conference on Games Based Learning ECGBL2010, Copenhagen, Denmark, 21-22, pp. 393-402.
- Tsichritzis, D. (1997). The Dynamics of Innovation, in *Beyond Calculation: The Next Fifty Years of Computing*.
- Ze, S. E. (2014). Microsoft Solutions Framework (MSF) for Agile Software Development.
- Zea, N. P. (2011). Metodología para el diseño de videojuegos educativos sobre una arquitectura para el análisis del aprendizaje colaborativo. Editorial de la Universidad de Granada.
- Zyda, M. (2005). From visual simulation to virtual reality to games. *Computer*, 38(9), 25–32. <https://doi.org/10.1109/MC.2005.297>