

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

ESCUELA DE SISTEMAS

DISERTACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE SISTEMAS Y COMPUTACIÓN

DESARROLLO DE UNA METODOLOGÍA PARA CREAR VIDEOJUEGOS CON HERRAMIENTAS ÁGILES

XAVIER ALEJANDRO ORDÓÑEZ VÁSQUEZ

DIRECTOR: MSc. LUIS OSWALDO ESPINOSA VITERI

Quito, 2021

AGRADECIMIENTOS

A mi madre, Norma, todo te lo debo a ella, sus guías y el amor que me ha dado siempre. Es la persona más importante de mi vida y mi mayor ejemplo a seguir.

A mi padre, Javier, por siempre haber creído en mí y alentarme a seguir adelante sin importar lo difícil que se pueda tornar el camino.

A mis hermanas, Dominick y Samantha, mi admiración por ellas es inmensa y es solo superada por el amor que les tengo, son mis mejores amigas de toda la vida.

A mi tío, Andrés, quien ha estado conmigo siempre, me ha visto crecer y ha estado conmigo en todo momento orientándome y enseñándome a ser una mejor persona, cada palabra y momento que he compartido con él vive en mi mente y en mi corazón, agradezco cada día que este en mi vida.

A mis abuelos Norman y Tula, los extraño cada día, nunca olvidaré todo lo que hicieron y dieron por mí.

A Ana Lucía Boada, por todas las alegrías y éxitos en los que me ha acompañado y por todo el apoyo y comprensión que me ha brindado en los momentos difíciles, agradezco con todo mi ser que nos hayamos encontrado en esta vida.

A mis amigos, por todos los momentos en que nos hemos apoyado, y por todas las experiencias e historias vividas a lo largo de la carrera, no los olvidaré.

Al ingeniero Oswaldo Espinosa, por todas las enseñanzas y guías a lo largo de la carrera, las cuales han sido fundamentales para llegar hasta el final.

A Dios, por haberme dado la fuerza para continuar y no rendirme.

DEDICATORIA

Este trabajo está dedicado a todas las personas que creyeron en mí y siempre me alentaron a seguir adelante, a alcanzar mis sueños y a luchar por lo que más anhelo en la vida.

RESUMEN

El trabajo de disertación “desarrollo de una metodología para crear videojuegos con herramientas ágiles” tiene como objetivo principal, plantear una metodología de desarrollo ágil específicamente orientada a la creación de videojuegos, para esto se realiza una investigación exploratoria seleccionando una población de desarrolladores que hayan participado en la creación de videojuegos durante el periodo 2021-2022, específicamente en el Ecuador, se llevaron a cabo dos eventos de desarrollo de videojuegos durante dos años consecutivos. Inicialmente se realizó una encuesta para evaluar el trabajo de los equipos de desarrollo creando un videojuego mediante la utilización de metodologías agiles de desarrollo.

Posteriormente se procedió a establecer un modelo de evaluación con el objetivo de determinar en qué aspectos las metodologías de desarrollo presentaban debilidades mayores en la creación de un videojuego. Los resultados permitieron concluir que tanto Scrum como XP resultaron ser las metodologías óptimas para el desarrollo de videojuegos y a la vez las metodologías más aplicadas por parte de los equipos, siendo Scrum ligeramente más aplicada que XP.

Finalmente, con la selección de ambas metodologías se procedió a proponer una nueva metodología de desarrollo denominada AGP (Agile Game Programming) la cual está basada en los principios y prácticas tanto de Scrum como de XP, pero tomando en cuenta aspectos esenciales de un videojuego.

Palabras clave: Videojuego, Metodologías agiles, equipo de desarrollo, desarrollo de videojuegos.

ABSTRACT

The dissertation work "development of a methodology to create video games with agile tools" has as its main objective, to propose an agile development methodology specifically oriented to the creation of video games, for this purpose, an exploratory investigation is carried out selecting a population of developers who have participated in the creation of video games during the 2021-2022 period. Specifically, in Ecuador, two video game development events were held for two consecutive years. Initially, a survey was carried out to evaluate the work of the development teams creating a video game using agile development methodologies. Subsequently, an evaluation model was established with the aim of determining in which aspects the development methodologies presented major weaknesses in the creation of a video game.

The results allowed us to conclude that both Scrum and XP turned out to be the most optimal methodologies for the development of video games and at the same time the most applied methodologies by the teams, with Scrum being slightly more applied than XP. Finally, with the selection of both methodologies, a new development methodology called AGP (Agile Game Programming) was proposed, which is based on the principles and practices of both Scrum and XP but considering essential aspects of a video game.

Keywords: Videogames, agile methodologies, development team, game development.

Tabla de Contenido

Capítulo I: Introducción a los videojuegos 1
1.1 Introducción 1
1.2 ¿Qué es un videojuego? 2
1.3 Estructura de un videojuego 3
1.4 GameJams 4
1.5 Problemas en el desarrollo de videojuegos 5
1.6 Ingeniería de software aplicada a videojuegos 6
Capítulo II: Análisis de la literatura 9
2.1 Metodologías ágiles para el desarrollo de videojuegos 9
2.1.1 Videojuegos Serios 11
2.1.2 Desarrollo centrado en el jugador 11
2.1.3 Prototipo iterativo 12
2.1.2 Desarrollo incremental 12
2.2 Modelos de evaluación de metodologías 15
2.1.1 Modelo de evaluación según el manifiesto ágil 15
2.1.2 Modelo de evaluación según criterios comparativos 16
2.1.3 Modelo de evaluación por criterios y puntajes 17
Capítulo III: Metodología 18
3.1 Construcción de instrumento para encuestas 33
3.1.1 Diseño de Investigación 18
3.1.2 Tamaño de la muestra 19
3.1.3 Cálculo de la muestra 19
3.1.4 Métodos 20
3.2 Escenarios y participantes de la encuesta 21
3.2.1 Contexto 21
3.2.2 Población 21
3.2.3 Recursos 23
3.3. Propuesta de indicadores para evaluación de las metodologías ágiles 23
3.3.1 Descripción de los Indicadores 23
Capitulo IV: Resultados 28
4.1 Resultados de la aplicación de la encuesta 28
4.2 Resultados de la aplicación de la evaluación a metodologías de desarrollo identificadas 35
4.2.1 Valoración general y selección de metodologías 38
Capítulo V: Propuesta de Metodología Ágil para el desarrollo de videojuegos 39
5.1 Consideraciones generales 39

5.2 Marco Conceptual y Delimitación 41
5.3 Metodología 46
5.3.1 Principios de AGP 46
5.3.2 Estructura de AGP 48
5.3.3 Ciclo de vida de AGP 48
5.3.4 Roles de AGP 52
5.3.5 Etapas de desarrollo de AGP 56
5.3.5.1 Preproducción 56
5.3.5.2 Producción 72
5.3.5.3 Postproducción 83
Conclusiones y recomendaciones 87
6.1 Conclusiones 87
6.2 Recomendaciones 88
Referencias Bibliográficas 90
Anexos 93

Índice de figuras

Figura 1: Fases del diseño centrado en el usuario 11
Figura 2: Ciclo de desarrollo centrado en el usuario 11
Figura 3: Metodología del prototipo iterativo 12
Figura 4: Ejemplo de Valoración del primer postulado del manifiesto ágil 16
Figura 5: Ejemplo de criterios de evaluación establecidos y su peso respectivo 17
Figura 6: Utilización de métricas para medir el desempeño del videojuego 32
Figura 7: Enfoque de diseño en el que se centró el equipo 32
Figura 8: Colaboración entre los miembros para elaborar el documento de diseño 33
Figura 9: Características de la documentación generada 33
Figura 10: Pruebas realizadas para corrección de defectos 34
Figura 11: Ciclo de Vida de AGP 52
Figura 12: Ejemplo de Flujo de un videojuego 61
Figura 13: Características de la documentación generada 62
Figura 14: Pantalla de inicio del videojuego Call of duty Warzone 63
Figura 15: Pantalla de menú principal del videojuego Bioshock Infinite 63
Figura 16: Pantalla de menú de pausa del videojuego Minecraft 64
Figura 17: Pantalla de finalización del videojuego Metroid Dread 65
Figura 18: Borrador inicial de requerimientos del Game Backlog 66
Figura 19: Historia de usuario de AGP 68
Figura 20: Priorización de requerimientos de AGP 69
Figura 21: Arquitectura basada en componentes de un videojuego 71
Figura 22: Lógica de la arquitectura de un videojuego 71
Figura 23: Flujo de un Sprint en AGP 74
Figura 24: Burn-Down Chart 74
Figura 25: Ejemplo de ontología definida por los diseñadores del videojuego 76
Figura 26: Ejemplo de diagrama de clases del videojuego del género shooter 77
Figura 27: Ejemplo de diagrama de Jerarquía de Entidades de un videojuego 78
Figura 28: Función de asignación de rango en un videojuego 79
Figura 29: Prueba automatizada 80
Figura 30: Proceso de integración continua de Jenkins 81
Figura 31: Proceso de revisión por parte del jugador 84

Índice de tablas

Tabla 1: Indicadores de evaluación de metodologías 24
Tabla 2: Criterio Nro. 1. Interacción del equipo 25
Tabla 3: Criterio Nro. 2 Documentación 25
Tabla 4: Criterio Nro. 3 Adaptabilidad a los cambios 25
Tabla 5: Criterio Nro. 4 Ciclo de Vida 26
Tabla 6: Criterio Nro. 5 Ciclo de Vida 26
Tabla 7: Criterio Nro. 6 Historias de Usuario 26
Tabla 8: Criterio Nro. 7 Historias de Usuario 27
Tabla 9: Resultados de las preguntas de tipo Likert en la encuesta 28
Tabla 10: Resultados de las preguntas dicotómicas en la encuesta 30
Tabla 11: Resultados de las preguntas de frecuencia en la encuesta 31
Tabla 12: Metodologías utilizadas por los equipos de desarrollo 35
Tabla 13: Valoración general de las metodologías identificadas 38
Tabla 14: Fusión de prácticas de Scrum con prácticas de XP 48
Tabla 15: Prácticas de la Etapa de Preproducción 49
Tabla 16: Prácticas de la Etapa de Producción 50
Tabla 17: Prácticas de la Etapa de Postproducción 51
Tabla 18: Reunión de planificación de AGP 57
Tabla 19: Aspectos del GDD 58
Tabla 20: Formato para el concepto del videojuego como parte del GDD 60
Tabla 21: Formato para determinar la prioridad de una historia de usuario 69
Tabla 22: Sprint Backlog 72
Tabla 23: Tablero de tareas 73
Tabla 24: Estructura No. 1 del Sprint Review 82
Tabla 25: Estructura No. 2 del Sprint Review 83
Tabla 26: Reporte de errores 85

CAPÍTULO I: INTRODUCCIÓN A LOS VIDEOJUEGOS

1.1 Introducción

Los videojuegos se han posicionado como uno de los medios de entretenimiento más populares en la actualidad, dicha popularidad ha ido creciendo gracias a su difusión mediante plataformas orientadas a los jugadores, dichas plataformas han revolucionado la industria del videojuego aprovechando la globalización del internet. Sin embargo, el interés por los videojuegos no recae únicamente en su consumo, sino también en su creación y desarrollo. Lo cierto es que durante la última década varios programadores se han adentrado en el desarrollo independiente de videojuegos es decir que muchos videojuegos ya no son necesariamente desarrollados por compañías de renombre, de hecho, muchos programadores han optado por crear sus propios videojuegos y aprovechar las herramientas que existen actualmente para trabajar de manera colaborativa e incluso poder distribuir su videojuego.

El desarrollo de software cuenta con metodologías, estándares, buenas prácticas de programación y procesos que garantizan la calidad del producto final, de manera que los proyectos de desarrollo pueden tener éxito o fracasar dependiendo de la forma en que el equipo se desempeñe. Detrás de la creación de un videojuego está un equipo de desarrolladores que tiene como objetivo entregar un software de calidad y que pueda entrar al mercado. Desafortunadamente son varios los proyectos que han fracasado tanto por parte de desarrolladores pertenecientes a una compañía como por desarrolladores independientes, ya que el principal problema por el cual fracasan en sus proyectos, más allá de tratarse de un tema presupuestario, se trata de un completa desorganización en su trabajo, lógicamente varios equipos de desarrollo intentan guiarse a través de una metodología ágil, debido a su flexibilidad con los cambios drásticos sin embargo, y a pesar de esto, las metodologías actuales no contemplan todos los aspectos que incluye un videojuego, ya que si bien es cierto los videojuegos también manejan requerimientos, el enfoque que tiene este tipo de software no se centra únicamente en estos requerimientos, sino en entregar una experiencia de entretenimiento al usuario.

El mercado de los videojuegos es bastante exigente y competitivo, cada año se produce un alto número de títulos, razón por la cual los equipos desarrolladores tratan de trabajar adaptando una metodología ágil para la creación del videojuego, pero a pesar de esto se siguen presentando anomalías en distintas fases del desarrollo lo cual ha llevado al fracaso de varios videojuegos, entregando un producto deficiente, o incluso cancelando completamente el proyecto aun cuando su desarrollo ya había avanzado considerablemente. Es importante recalcar que este problema no es exclusivo de desarrolladores independientes, ya que existen muchos casos de estudios desarrolladores que igualmente registran proyectos que han fracasado.

Bajo este contexto se desarrollará el siguiente trabajo que consiste en un análisis de las metodologías ágiles existentes a través de un modelo de evaluación. Este modelo determinará que metodologías se adaptan mejor al desarrollo de videojuegos; en base a esto se realizará el planteamiento de una metodología de desarrollo ágil enfocada directamente a la creación de videojuegos.

1.2 ¿Qué es un videojuego?

La popularidad de los videojuegos se traduce al éxito que han logrado cosechar como medio de entretenimiento desde su creación hasta la actualidad. Hoy en día los videojuegos ocupan un aspecto relevante en la sociedad, pues los jugadores ya no solo ven a los videojuegos como un pasatiempo, ya que en torno al videojuego han surgido otro tipo de actividades como periodismo de videojuegos, e-sports y gamejams (Martínez, 2007).

Un videojuego es un software o programa informático en el cual un usuario o en este caso el jugador, interactúa directamente con un dispositivo, en el que se muestra una serie de imágenes y sonidos que captan su atención, la idea del videojuego como tal es presentarle un reto al jugador, dicho reto presenta distintos niveles de dificultad lo cual insta al jugador a tratar de superar esos retos, generando así un sentimiento de satisfacción en el jugador (Moncada Jiménez & Chacón Araya, 2012).

1.3 Estructura de un videojuego

Según, un videojuego posee una estructura interna sobre la cual se desarrolla una partida, esta partida se encuentra articulada en base a reglas y límites establecidos en cada situación, estas reglas y límites son las que generan una variedad en cuanto a configuración de escenarios se refiere, es decir, el jugador se va a encontrar ante distintos panoramas mientras avanza en la partida, esto le representa distintos retos, para los cuales deberá plantear diversas estrategias. El autor plantea la siguiente estructura:

Jugador. - El jugador establece una comunicación directa con el videojuego a través de la toma de decisiones.

Objetivos. - Los objetivos son comprendidos como un estado deseable por el jugador, estos objetivos varían dependiendo del progreso del jugador, sin embargo, estos objetivos igualmente se encuentran regidos por las reglas del entorno, para que el jugador consiga estos objetivos tienen que superar los retos que el juego ha planteado.

Reglas. - Las reglas limitan el comportamiento de los elementos en el videojuego y están diseñadas para mantener la lógica de este.

Recursos. - La interacción que tiene el jugador con el videojuego está ligada a través de elementos que puede utilizar dentro del juego para cumplir los retos.

Obstáculos. - El obstáculo es comprendido como un ente que influye directamente sobre el jugador y su progreso tratando de frenarlo.

Frontera. - Dentro del mundo jugable existen límites que conforman su estructura, y delimitan las zonas que el jugador puede explorar. Mediante estas limitantes, se establece la navegación que puede tener un jugador en cada zona.

Historia. - Los videojuegos suelen manejar premisas para ofrecer una experiencia más inmersiva. La historia dentro del videojuego permite mostrarle al jugador, de manera más detallada, las consecuencias de cada acción que desempeñe

Resultados. - El estado final de un nivel significa el éxito o fracaso que ha tenido el jugador tras haberse enfrentado a los retos que se le han presentado.

1.4 GameJams

El GameJam es un evento que se realiza cada año en distintas ciudades del mundo, aquí se reúnen varios desarrolladores, para poder diseñar y crear videojuegos propios en dos días, este tipo de eventos no son competitivos, sino que apelan al trabajo colaborativo para aumentar el aprendizaje, razón por la cual este evento no es exclusivo para programadores profesionales o seniors (Aurava et al., 2021).

Trabajo en Equipo

GameJam no representa únicamente una oportunidad para crear un videojuego, sino para adquirir habilidades de desarrollo y sobre todo aprender a trabajar en equipo ya que lógicamente, no todos los miembros que conformen un equipo tendrán los mismos conocimientos sobre programación. En este evento se apuesta por el aprendizaje STEAM (Ciencia, Tecnología, Ingeniería, Arte, Matemáticas) (Contreras-Espinosa & Eguia-Gomez, 2022).

Organización

Según Aurava et al. (2021), el equipo de trabajo, desarrolla durante dos días un prototipo de lo que vendría a ser el juego que quieren crear, este prototipo es construido basándose en una división de tareas entre los miembros, tomando en cuenta las habilidades y competencias de cada uno. El equipo adopta roles para desempeñar las tareas, estos roles corresponden a: programadores, diseñadores y artistas, esto demuestra que en la mayoría de los equipos existe una diversidad en cuanto a campos de acción.

1.5 Problemas en el desarrollo de videojuegos

La cancelación de un videojuego es el resultado de una serie de problemas que se dan durante el desarrollo del proyecto, estos problemas no se limitan únicamente a aspectos administrativos o financieros sino a un deficiente desarrollo por parte del equipo. Casos como el videojuego “Cyberpunk 2077” o “Scalebound” son solo uno de los tantos fracasos que se ven en la industria.

La especialización en el desarrollo de videojuegos deriva en una expansión de los roles existentes en la creación de este tipo de software. Estos nuevos roles ya no están basados únicamente en aspectos de la programación sino también en aspectos artísticos, y es que además de los programadores se adhieren al grupo los diseñadores, artistas y escritores (Llansó, 2014).

Según Mitre-Hernández et al. (2016), la industria del videojuego se encuentra incrementando sus capacidades de manera continua, lo que desemboca en procesos cada vez más ágiles, esto le representa al equipo un panorama que está en constante cambio y que demanda varias modificaciones. El mayor problema que sufren los equipos de desarrollo está relacionado con los roles, ya que, al tratarse de un videojuego, existen roles artísticos y técnicos, esta unión de ambos campos, provoca varias confusiones.

La comunicación entre los miembros se realiza mediante un documento de diseño, escrito en lenguaje natural, desafortunadamente este documento resulta muy estático, lo que conlleva a que las modificaciones se deban realizar en las especificaciones iniciales, y alterar un requerimiento cuando el proyecto ya se encuentra avanzado representa costos adicionales, que empiezan a mermar el presupuesto destinado al proyecto (Jarzębowicz & Weichbroth, 2021).

1.6 Ingeniería de Software aplicada a videojuegos

Según Acerenza et al. (2009), se han presentado casos en los cuales la adaptación de metodologías iterativas-incrementales han logrado llevarse a cabo con éxito, destacando principalmente la obtención de versiones jugables del software, de esa manera se podía inspeccionar el comportamiento del juego e implementar los cambios respectivos para actualizarlo y tener una mejora para la próxima iteración, evitando así cambios tardíos en los requerimientos.

De acuerdo con Llansó (2014), las múltiples adaptaciones de metodologías ágiles hacia el desarrollo de videojuegos han producido que no se pueda estandarizar ninguna de estas adaptaciones, ya que sus variaciones dependen de cada equipo de desarrollo, a pesar de esto, el proceso seguido por las adaptaciones resulta similar en varios casos salvo por ligeras excepciones. El autor plantea el siguiente proceso general de desarrollo de videojuegos.

Preproducción

En esta fase de desarrollo, se construye alrededor del 20% del videojuego sin embargo es a su vez la fase más importante, ya que aquí se concibe la idea general sobre la cual estará basado el videojuego, principalmente en esta fase se empiezan a crear prototipos de lo que se espera del videojuego como producto final, es decir durante esta fase se da el inicio conceptual del videojuego.

Producción

En esta fase ya empieza el desarrollo y la ejecución de tareas pertenecientes a cada rol. Durante el desarrollo de esta etapa producen todos los contenidos iniciales y finales del videojuego.

Postproducción

En esta fase de desarrollo el equipo ya puede considerar si el videojuego está completo y ha sido construido con todas las consideraciones iniciales, de ser así, se inicia la corrección de errores o bugs que el videojuego pueda tener, sin embargo, esta fase no está centrada únicamente en arreglar errores técnicos, sino también en equilibrar la jugabilidad.

Arquitectura de software de videojuegos

De acuerdo con Pi (2017), la aplicación de una arquitectura de software para desarrollar un videojuego le permite al equipo de desarrollo disminuir la complejidad de procesos de bajo nivel.

Un videojuego maneja entidades, que representan las partes lógicas del contenido, cada una de estas entidades, desempeña distintas tareas. Por sus constantes cambios, estas son modificadas o muchas veces eliminadas. En los casos documentados se ha evidenciado que la mayoría de los programadores que están desarrollando un videojuego optan por utilizar un paradigma orientado a objetos en el cual prefieren utilizar una composición dinámica en vez de jerarquías estáticas de clases (Llansó, 2014).

Para Gómez (2007), una manera en la que se puede implementar la jerarquía de clases es a través de la división de entidades en funcionalidades independientes que sea tratadas a manera de clases de manera que dichas clases puedan comunicarse a través de punteros, de forma que una entidad pueda hacer referencia a una serie de funcionalidades justamente a través de punteros entre clases.

Arquitectura Basada en componentes

Esta arquitectura se posiciona como una de las útiles y ventajosas para el desarrollo de videojuegos, lo cierto es que un videojuego al tener tantas entidades, lo que se busca con esta arquitectura es la capacidad de generar un conjunto de componentes que puedan ser utilizados por una sola entidad y que todas estas entidades se encuentren regidas por datos esenciales para el videojuego tales como: movimientos, puntuaciones, datos de personalización etc. (Gómez, 2007).

Para Muñoz-Arteaga (2013), la diversidad de componentes que existen en un videojuego hace que sea imperativo la utilización de una arquitectura de software, sobre todo por que varios de estos componentes están en constante comunicación durante la ejecución de cada nivel del videojuego, particularmente la existencia y aparición de múltiples clases en cada prototipo entregado en las distintas fases de desarrollo del videojuego conduce a que una arquitectura basada en componentes resulte ser la solución más óptima.

La lista de componentes funciona como un puerto de comunicación para receptar los mensajes del resto de componentes, este mensaje está conformado por un identificador y parámetros opcionales, la manera de comunicación que ejercen los componentes es a

través de las entidades que se encargaran de la propagación del mensaje a través de los métodos que manejen los componentes (Llansó, 2014).

CAPÍTULO II: ANÁLISIS DE LA LITERATURA

2.1 Metodologías ágiles para el desarrollo de videojuegos

Los videojuegos están en constante evolución, tratando de adaptarse a las exigencias de un mercado competitivo. Esto hace que los videojuegos cada vez sean más complejos de realizar y requieran de grandes equipos de desarrollo para poder crearlos, por consiguiente, la aplicación de metodologías ágiles de desarrollo se vuelve un asunto imperativo para entregar un videojuego que satisfaga las exigencias de la comunidad de jugadores y a la vez se entregue un producto de calidad.

Según Llopis (2017), el proceso para crear un videojuego es muy impredecible debido a los constantes cambios que tiene durante las distintas fases de desarrollo, por esta razón la mayoría de las compañías desarrolladoras han optado por adaptar metodologías ágiles, debido a su capacidad de adaptarse a constantes cambios.

De acuerdo con un estudio realizado por Morales et al. (2010), se han aplicado varias metodologías de desarrollo para crear un videojuego, siendo las más aplicadas las siguientes:

RUP

La metodología ágil RUP (Rational Unified Process) resulta ser más flexible y adaptativa, sin embargo, no resulta viable para el desarrollo de videojuegos debido a la constante y alta cantidad de documentación que requiere.

EssUP

La metodología EssUP (Essential Unified Process) se presenta como una opción viable debido a que integra prácticas esenciales basándose en tres campos; la madurez del proceso, los métodos ágiles y el proceso unificado, de esta forma se puede generar una separación de trabajo, entre al área enfocada a la parte funcional e ingenieril del videojuego con el área enfocada al apartado artístico.

OpenUP

El proceso de desarrollo OpenUP apuesta por incluir elementos que sean considerados únicamente como esenciales con el fin de que solo la información necesaria sea intercambiada entre en los miembros del proyecto, apostando fuertemente por la inclusión de los stakeholders como parte del proyecto, esta filosofía y modo de trabajo que pone especial énfasis en la comunicación resulta ventajosa para el desarrollo de videojuegos.

TSP

el TSP resulta como una de las metodologías óptimas para el desarrollo de videojuegos debido al balance que produce entre el producto, el proceso y el equipo de desarrollo, gracias a una clara definición de tareas para cada miembro en cada fase de desarrollo, de igual manera el registro de formas permite el seguimiento del proceso de una manera entendible.

MSF

El marco de trabajo MSF (Microsoft Solutions Framework) basa su funcionamiento en una serie de iteraciones apuntando a un refinamiento en cada ciclo, lo que permite entender los requerimientos de manera progresiva. Este marco de trabajo resulta entonces altamente adaptable a los requerimientos y entornos de varios tipos de proyectos, por esta razón encaja dentro del desarrollo de videojuegos debido a su alta adaptabilidad.

Scrum

SCRUM por otro lado aparece como marco de trabajo de tipo iterativo e incremental basado en un desarrollo por ciclos , estos ciclos se conocen como iteraciones , levantando historias de usuario en cada sprint, y manteniendo un orden en los requerimientos , para el desarrollo de videojuegos esto es especialmente útil ya que al final de cada iteración el equipo de desarrollo puede probar las características del juego y de esa manera conseguir retroalimentación de dichas características, tomando en cuenta lo que se debe mejorar.

2.1.1 VIDEOJUEGOS SERIOS

Un videojuego serio se entiende como un software que es desarrollado en base a estándares de calidad, para cumplir con estos estándares y entregar un videojuego de calidad se debe seguir un proceso de ingeniería de software, estos videojuegos serios están centrados en el jugador como usuario final. La razón principal para utilizar un diseño centrado en el usuario es entonces, poder enfocarse en las necesidades, limitaciones y lo deseado por el usuario (Garita Hernández et al., 2019).

Para conseguir esto se trabajan las siguientes fases:

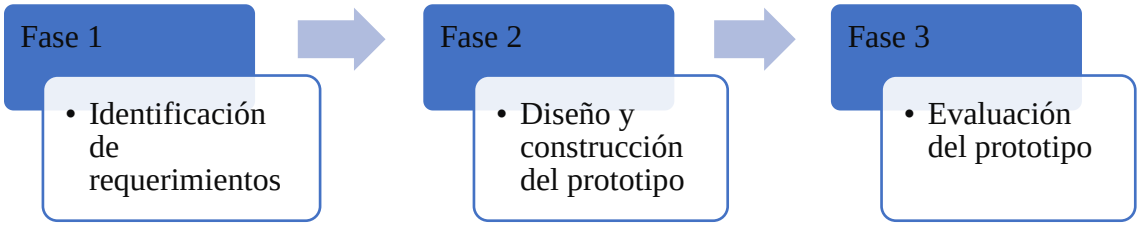


Figura 1: Fases del diseño centrado en el usuario

Fuente: (Garita Hernández et al., 2019)

2.1.2 DESARROLLO CENTRADO EN EL JUGADOR

Para Zapirain & Massa (2017), el desarrollo centrado en el usuario es un proceso cíclico bajo el cual las decisiones con respecto al diseño son dirigidas por el usuario para poder cumplir los objetivos que busca satisfacer el videojuego, para conseguir esto es necesario que la usabilidad del videojuego pueda ser evaluada de forma iterativa y poder implementar las mejores de manera incremental. El ciclo es descrito en la siguiente figura.

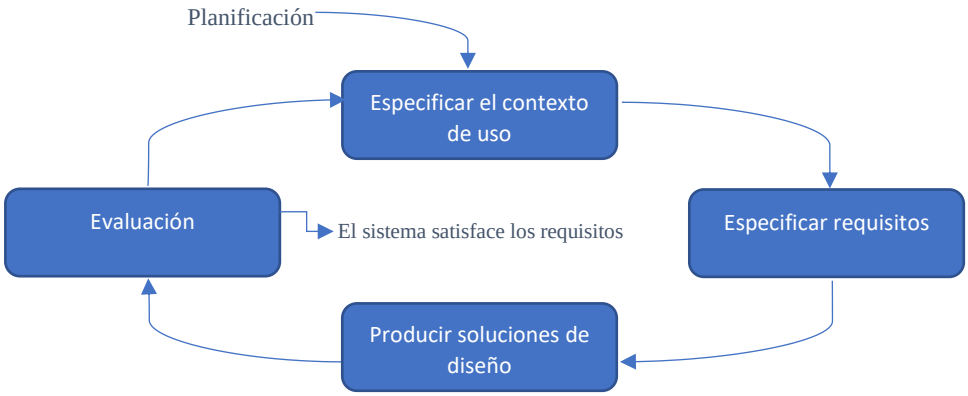


Figura 2: Ciclo de desarrollo centrado en el usuario.

Fuente: (Zapirain & Massa, 2017)

2.1.3 PROTOTIPO ITERATIVO

Para Iglesias (2011), el diseño centrado en el usuario está fundamentado en la creación de un prototipo que pueda ser utilizado durante todo el proceso de desarrollo, dicho prototipo debe ser refinado a través de pruebas iniciales que permitan identificar defectos, de esta manera poder obtener la retroalimentación necesaria para mejorar las historias de usuario y poder identificar posibles requerimientos emergentes, los cuales serían tomados en cuenta para las iteraciones posteriores. En la siguiente figura se establece el proceso de desarrollo del prototipo.

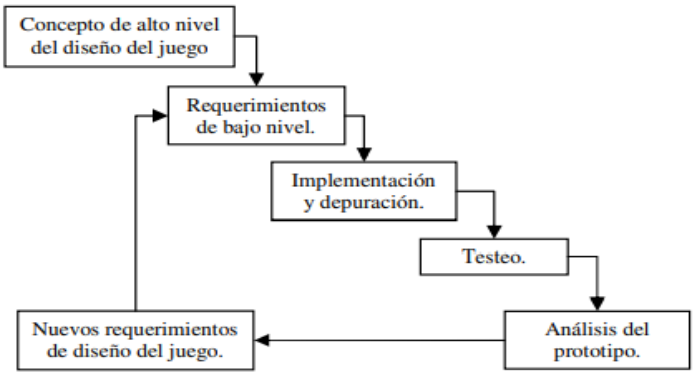


Figura 3: Metodología del prototipo iterativo.

Fuente: (Iglesias, 2011)

2.1.4 DESARROLLO INCREMENTAL

En el modelo incremental el videojuego puede irse desarrollando en partes incrementales a partir de un concepto en alto nivel del videojuego, mientras que la documentación del diseño del videojuego se debe ir realizando en paralelo con el proceso de desarrollo, de esta forma los desarrolladores y los diseñadores podrían evadir confusiones y conseguir un acuerdo sobre qué características van a ser implementadas y de qué manera (Iglesias, 2011).

Ciclo de Vida del desarrollo de videojuegos

Para Subhash & Maruthi (2007), el ciclo de vida que sigue el desarrollo de un videojuego es similar al de un software tradicional. Este ciclo de vida está compuesto por distintas fases que están basadas en varios factores, principalmente los factores más relevantes son los requerimientos tanto de software como de hardware y el tiempo de vida útil del videojuego. El ciclo de vida planteado por el autor es el siguiente:

Desarrollo de la historia. - En esta primera etapa se definen los elementos argumentales del videojuego, principalmente la historia central del videojuego, este elemento está desarrollado con base en la idea o concepto general del videojuego.

Desarrollo del guion. - En esta fase se toma la historia ya definida para empezar a construir el guion, este aspecto es relevante porque vuelve más consistente a la historia, especialmente en la actualidad.

Estudio de Viabilidad. - En esta etapa se evalúa la viabilidad del juego, esto ayuda a definir concretamente los requerimientos y las implicaciones que tendrá cumplir esos requerimientos, considerando los aspectos técnicos, legales y organizacionales. Los programadores entonces pueden definir y explicar las limitaciones del proyecto.

Demo Promocional. - En esta etapa se planifica la demostración promocional del videojuego, para mostrar al cliente la visión general del videojuego a través de una versión de prueba básica.

Diseño. - En esta etapa se inicia el diseño de los distintos elementos artísticos que componen al videojuego, enfocándose en el diseño de personajes, el diseño de los escenarios o fondos y el diseño de los elementos complementarios como los accesorios o decorados pertenecientes a los personajes o al ambiente del mundo virtual.

Diseño del Layout. - Esta etapa continua con el diseño, pero ahora enfocado a la construcción de los niveles, es decir aquí se idea la manera en que estarán dispuestos los obstáculos y los caminos o rutas que el jugador podrá seguir en el entorno del mundo virtual, así mismo se establecen las respectivas limitaciones del mundo virtual.

Modelado. - La etapa del modelado consiste en construir modelos que proporcionen información detallada sobre las representaciones de los personajes y los escenarios, incluyendo los detalles más finos, con el fin de lograr una representación llamativa.

Texturizado. - La texturización es una técnica bastante útil para proporcionar a las superficies tanto de los personajes como de los elementos que componen un escenario, esta técnica es generalmente aplicada a los diseños en 3D para dotar de realismo a los gráficos de un videojuego.

Animación. - Una vez los diseños ya se encuentren finalizados, ya empieza la etapa de animación, para construir los movimientos que van a tener los personajes.

Diseño de alto y bajo nivel. - En el diseño de alto nivel se establecen las relaciones que existen entre los módulos del proceso de desarrollo, estos son colocados a manera de flujo de datos para comprender el diseño del juego de manera general, mientras que el diseño de bajo detalla la lógica que existe detrás de los módulos definidos en el diseño de alto nivel.

Codificación. - La codificación es la etapa más esencial en el ciclo de vida, ya que esta fase además de estar enfocada en hacer que los juegos sean interactivos también implementa el diseño de la interfaz, para conseguir esto los equipos de desarrollo tienen a su disposición múltiples motores para crear videojuegos, tanto de tipo "open source" como de pago.

Pruebas y Depuración. - Para realizar la integración de todos los módulos primeramente se deben realizar pruebas a cada uno de ellos, para dejar a los módulos libres de errores, los equipos hacen uso de los motores de videojuegos para depurar y realizar múltiples pruebas a los módulos.

Integración. - En esta etapa se realiza la integración de todos los módulos después de que cada módulo haya superado la etapa de pruebas.

Pruebas del Juego. - En esta etapa final se prueba el juego para observar su comportamiento desde el inicio hasta el final, se podría comprender esta etapa como una segunda parte para comprobar que el videojuego está libre de errores y que su comportamiento es el esperado.

2.2 Modelos de Evaluación de metodologías ágiles

Los modelos de evaluación se establecen con el objetivo de comprobar si una metodología es óptima para responder a cambios repentinos y constantes que se producen en el desarrollo del software.

2.2.1 Modelo de evaluación según el manifiesto ágil

Según Mendes Calo et al. (2010), el modelo de evaluación consiste en la valoración de que metodologías ágiles logran cumplir con los postulados del manifiesto ágil, es decir que metodologías están sostenidas en las bases y principios del manifiesto ágil, de esta forma el modelo busca evaluar a la metodología tomando en cuenta cada postulado como un atributo, para su posterior medición. El procedimiento de evaluación es el siguiente:

- Se realiza una separación del postulado en dos partes, la primera parte representa una polaridad positiva mientras que la segunda parte corresponde a una polaridad negativa.
- Se establece una escala de calificación del 0 al 5 para la parte positiva siendo 5 el mayor puntaje, mientras que para la parte negativa se establece una escala del -5 al 0.
- Se establece una serie de criterios que son evaluados mediante la escala planteada.
- Una vez obtenidas ambas calificaciones, se realiza una suma de la parte positiva con la negativa para obtener la puntuación final, si el resultado final alcanza una puntuación de 5 o se acerca, se determina que la metodología cumple con dicho postulado.

P1	*Valorar al individuo y las interacciones del equipo por sobre el proceso y las herramientas*		
P1 .1	*Valorar al individuo y las interacciones*	P 1. 2	*Valorar el proceso y las herramientas*
valor	descripción	valo r	descripción
0	No define roles para individuos	-5	Define actividades, entregables, herramientas de desarrollo y de gestión
1	Clara definición de roles para individuos	-3	Define actividades, entregables y herramientas de desarrollo
2	Clara definición de roles y responsabilidades	-2	Define actividades y entregables
3	Clara definición de roles, responsabilidades y conocimientos técnicos	-1	Define actividades para cada iteración
5	Clara definición de roles, responsabilidades, conocimientos técnicos e interacciones entre miembros del equipo de trabajo	0	Define actividades para el proyecto pero no a nivel de cada iteración

Figura 4: Ejemplo de Valoración del primer postulado del manifiesto ágil

Fuente: (Mendes Calo et al., 2010)

2.2.2 Modelo de evaluación según criterios comparativos

Según un estudio realizado por Espinoza-Meza (2013), se puede evaluar cada metodología de desarrollo a través de la selección de una serie de filtros generales que se adecuan al desarrollo de la mayoría de proyectos. Los filtros contemplan los aspectos más esenciales y característicos de las metodologías ágiles, el autor plantea los siguientes filtros utilizados para evaluar que metodología resulta ser la más adecuada.

PROCESOS

- **Predicción y Agilidad:** Se analiza la incertidumbre del entorno y la inestabilidad de los requerimientos.
- **Frecuencia de Flujos de Trabajo:** Priorización de tareas en las fases de desarrollo.
- **Prácticas, Roles y Tareas:** Asignación de tareas de acuerdo con los roles.

EQUIPO

- **Número de personas:** Cantidad de miembros del equipo de desarrollo.
- **Clientes:** Grado de participación del cliente durante el desarrollo.

ORGANIZACIÓN

- **Criticidad:** Potencial daño que puede ocasionar un fallo en el sistema.
- **Tamaño del Proyecto:** Grado de complejidad del proyecto.

2.2.3 Modelo de evaluación por criterios y puntajes

Este modelo de evaluación propuesto por Cartuche (2015), consiste en evaluar las metodologías de desarrollo, pero a través de criterios previamente establecidos. Estos criterios tienen un peso o puntaje que permite determinar si la metodología cumple en mayor o en menor medida con el criterio establecido.

	Criterio de Evaluación	Peso (%)
1	Grado de Conocimiento	20
2	Soporte orientado a objetos	10
3	Adaptable a cambios	15
4	Basada en historias de usuario o casos de uso	10
5	Documentación adecuada	10
6	Facilita la integración de las etapas de desarrollo	10
7	Relación con UML	10
8	Desarrollo de software sobre cualquier tecnología	15
	TOTAL	100

Figura 5: Ejemplo de criterios de evaluación establecidos y su peso respectivo

Fuente: (Cartuche, 2015)

De acuerdo con la figura 5, se establecen una serie de criterios de evaluación, los cuales constan de un peso máximo, el escenario ideal consiste en que la metodología evaluada pueda obtener un puntaje alto en cada uno de los criterios establecidos. Una vez la metodología es evaluada completamente obtiene una puntuación final, que determina si la metodología resulta ser óptima o no.

Según Cartuche (2015), los criterios seleccionados corresponden a las características más notables de las metodologías de desarrollo agiles, adicionalmente en este modelo de evaluación se toma en cuenta como criterio, el nivel de dominio o conocimiento que tiene el equipo de desarrollo sobre la metodología en cuestión.

CAPÍTULO III: METODOLOGÍA

3.1 Construcción de instrumento para encuestas

El trabajo tiene como objetivo establecer una propuesta de metodología ágil para el desarrollo de videojuegos, para conseguir esto en primera instancia se recurrirá a una investigación cualitativa realizada a través de encuestas, los resultados de esta investigación servirán para determinar que metodologías de desarrollo presentan más fortalezas en el desarrollo de videojuegos y que metodologías presentan más debilidades ya que de esta manera se podrá definir que metodologías resultan más adecuadas, a partir de la identificación de fortalezas y debilidades se realizará un análisis comparativo de las metodologías para definir cuáles de ellas resultan más óptimas para el desarrollo de videojuegos.

El hecho de identificar las metodologías óptimas para la creación de videojuegos permitirá plantear la propuesta de metodología ágil tomando en cuenta las fortalezas de las metodologías ya existentes.

3.1.1 Diseño de Investigación

Investigación Cualitativa

La investigación está basada en los criterios de los programadores que se desenvuelven o han participado en el desarrollo de videojuegos, es decir la investigación está enfocada en analizar a través de preguntas puntuales la percepción de los desarrolladores sobre las metodologías ágiles en el desarrollo de videojuegos y un análisis de aspectos que resultan clave en el desarrollo de videojuegos.

Investigación Bibliográfica

El intento por adaptar metodologías agiles al desarrollo de videojuegos se encuentra documentado en varios casos en los que se evidencia cierto factor de éxito o de fracaso, por dicha razón la revisión y obtención de información bibliográfica de estos casos documentados servirá como material de apoyo para la investigación.

Investigación Exploratoria

La investigación posee exploraciones específicas en el entorno del desarrollo de videojuegos para estudiar el problema y definir una propuesta formal de una metodología ágil.

3.1.2 Tamaño de la muestra

La muestra se obtendrá a través de un muestreo intencional, de tipo no probabilístico, ya que se seleccionaron a los participantes de manera intencionada y de acuerdo con los criterios de inclusión especificados, se toma en cuenta una población finita para la muestra.

Debido al contexto de la pandemia y por motivos de seguridad, las encuestas se realizaron completamente de manera virtual a través de la plataforma Google Forms, para determinar el tamaño de la muestra a encuestar y considerando una población finita se aplicó la siguiente fórmula:

$$n = \frac{z^2 * p * q * N}{e^2(N-1) + z^2 * p * q}$$

N = Población

n = Muestra

p = Probabilidad a favor

q = Probabilidad en contra

z = Nivel de confianza

e = Error de muestra

3.1.3 Cálculo de la muestra

Para el trabajo de investigación se seleccionaron los participantes en el desarrollo de videojuegos de las dos últimas ediciones del Global Game-Jam Ecuador correspondientes al año 2020 y 2021, así como dos empresas de desarrollo de videojuegos establecidas en

el Ecuador, y analizando la muestra de acuerdo con los criterios de inclusión y exclusión propuestos se obtuvo la siguiente muestra.

z = 1.96 (Nivel de confianza del 95%)

p = 50% ⟶ 0,5

q = 50% ⟶ 1-p = 1 – 0,5 = 0,5

e = 5 ⟶ 0,05

N= 60

$$n = \frac{(1.96)^2 * 0{,}5 * 0{,}5 * 60}{(0{,}05)^2(60-1) + (1.96)^2 * 0{,}5 * 0{,}5}$$

El tamaño total de la muestra corresponde a 52 personas.

3.1.4 Métodos

De acuerdo con la muestra seleccionada para la investigación, se aplicaron métodos óptimos que permitan la consecución de los objetivos preestablecidos. El contexto de la pandemia ha conllevado a evaluar distintos tipos de métodos y técnicas para recopilar la información, y cumplir con los objetivos adaptándose a la modalidad en que trabajan las personas actualmente, partiendo de esta concepción se seleccionaron los siguientes métodos:

-Método inductivo: mediante este método fue posible sintetizar la información de los casos documentados para comprender el contexto y de qué manera trabajan los desarrolladores de videojuegos en el país.

-Método descriptivo: por este método fue posible analizar los procesos y distintas metodologías de desarrollo aplicadas por los equipos de desarrollo.

-Método analítico: de esta forma se pudo determinar los distintos aspectos que entran en juego a la hora del desarrollo de videojuegos

-Método sintético: mediante este método se pudo reorganizar los datos que se recopilaron acerca de los aspectos esenciales en el desarrollo de videojuegos, los cuales sirvieron para evaluar las distintas metodologías de desarrollo posteriormente.

3.2 Escenarios y participantes de la encuesta

3.2.1 Contexto

En el Ecuador, existen múltiples profesionales que han participado en el desarrollo de videojuegos o que actualmente se encuentran desarrollando un videojuego. El país actualmente cuenta con una reducida cantidad de empresas que se dedican al desarrollo de videojuegos, sin embargo, este tipo de desarrollo no está concentrado únicamente en un sector empresarial, ya que existen varios equipos de desarrollo que trabajan de manera independiente, ya sea en la participación de hackatons orientados a la creación de videojuegos o a través de plataformas virtuales tales como: Steam, Play Store, Itch.io, etc. Tomando esto en cuenta se ha considerado pertinente obtener la información de Profesionales en desarrollo de software, y que hayan participado en el desarrollo de un videojuego en los últimos 5 años, o que actualmente se encuentren participando en la creación de un videojuego.

3.2.2 Población

Para seleccionar la población participante de la investigación, se toman en cuenta los siguientes criterios de inclusión y exclusión.

Criterios de inclusión

-Profesionales o Preprofesionales que hayan participado en el área técnica del desarrollo.

-Formar parte de la industria o haber participado en una de las dos últimas ediciones del Global Game Jam Ecuador.

-Haber desarrollado y publicado un videojuego en los últimos 3 años.

Criterios de exclusión

-Profesionales o Preprofesionales enfocados únicamente en el área artística de un videojuego.

Proceso

La primera fase de la investigación empieza por la selección de la muestra de acuerdo a los criterios de inclusión y exclusión que fueron considerados previamente, tomando esto en cuenta se seleccionaron dos empresas ecuatorianas enfocadas al desarrollo de videojuegos, sin embargo como se estableció en un principio, la muestra no contemplaba únicamente el sector empresarial, sino también el sector independiente, por esta razón también fueron seleccionados desarrolladores que intervinieron en los juegos publicados en las dos últimas ediciones del Global Game Jam-Quito correspondientes al año 2020 y 2021. El Global Game Jam es un evento presencial sin embargo por el contexto de la pandemia que vivió el Ecuador durante el 2020 y el 2021, el evento se llevó a cabo de manera virtual utilizando principalmente la plataforma Discord.

La pandemia provocó que muchos de los equipos de desarrollo tanto en las empresas como en los equipos independientes tuviesen que reducir significativamente sus reuniones presenciales, tomando esto en cuenta, para poder contactar con los miembros seleccionados para participar en la encuesta, se lo hizo de manera virtual, contactados a través de canales de comunicación en línea para comprobar su interés en la investigación y su aporte de información de acuerdo a su experiencia desarrollando videojuegos.

En el contexto empresarial se realizó una comunicación directa con el líder de cada proyecto para dialogar sobre el propósito de la investigación. Cabe destacar que se indicó claramente que la información obtenida de las respectivas encuestas se utiliza únicamente con fines investigativos, al tratarse de empresas pequeñas y de proyectos emergentes en el país los equipos de desarrollo se mostraron abiertos a participar en la investigación.

El procedimiento en el sector independiente fue el mismo, se contactó inicialmente con el líder de cada videojuego publicado oficialmente en los eventos del Global Game-Jam del año 2020 y el año 2021 para dialogar sobre la investigación e indicarles que la información obtenida tenía únicamente propósitos investigativos, de igual manera los equipos se mostraron interesados y abiertos a participar en la investigación.

3.2.3 Recursos

Para la elaboración de la investigación fueron necesarios los siguientes recursos:

Recursos Humanos

Debido a la pandemia, en el país se adoptó el teletrabajo como nueva modalidad para un porcentaje de instituciones en el país, en el caso de las dos empresas de desarrollo de videojuegos continuaron sus actividades de manera virtual, así mismo las dos últimas ediciones de los Global Game-Jam se realizaron de manera virtual a través de la plataforma Discord, de manera que para entablar una comunicación directa con los grupos que desarrollaron y publicaron sus videojuegos, se utilizaron netamente plataformas de comunicación principalmente Google Meets.

En el contexto de las empresas, al tratarse de empresas pequeñas con un personal reducido, el líder de cada proyecto comentó con el resto de los miembros del equipo, el propósito de la investigación y una vez se mostraron abiertos a participar se les envió las respectivas encuestas.

En el contexto de desarrollo independiente fue bastante similar ya que primeramente se estableció una comunicación con el líder del proyecto, seguidamente el líder del proyecto comentó el motivo de la investigación y una vez los miembros se mostraron abiertos a participar se les envió las respectivas encuestas.

3.3 Propuesta de indicadores para evaluación de las metodologías ágiles

Para poder identificar las fortalezas y debilidades que presentan las metodologías ágiles utilizadas para el desarrollo de videojuegos, y de acuerdo con las respuestas obtenidas mediante la encuesta realizada, se plantearon indicadores que permitan valorar la efectividad que tiene cada metodología en distintos aspectos del desarrollo del videojuego, los indicadores valoran aspectos importantes tanto en lo que se refiere a buenas prácticas de desarrollo de software como a desarrollo de videojuegos

3.3.1 Descripción de los Indicadores

Para poder evaluar la efectividad de cada metodología utilizada por los distintos equipos de desarrollo, se plantea un modelo de evaluación que está basado en el “Modelo de evaluación por criterios y puntaje” propuesto por (Cartuche, 2015), dicho modelo se encuentra explicado en el capítulo 2. Para este caso particular en el cual se van a evaluar

que metodologías de desarrollo son las óptimas para el desarrollo de videojuegos, se tomaron en cuenta una serie de criterios que están basados en las características más prominentes del desarrollo de videojuegos. Para esto, cada criterio es evaluado a través de un indicador, el cual consta de una valoración que permite determinar si la metodología presenta una mayor fortaleza o debilidad en un determinado aspecto, para esto caso se ha decidido utilizar un rango de valoración del 1 al 5, siendo "1" una puntuación muy baja, "2" una puntuación baja, "3" una puntuación media, "4" una puntuación alta y "5" una puntuación muy alta. A continuación, se muestra la tabla de criterios y sus respectivos indicadores.

Tabla I: Indicadores de evaluación de metodologías

Código	Criterios	Indicadores	Valoración
C1	Interacción del equipo	Nivel de comunicación	15
C2	Documentación	Nivel de precisión de la información	20
C3	Adaptabilidad a los cambios	Adecuación con los objetivos	15
C4	Ciclo de Vida	Eficacia del proceso de desarrollo	10
C5	Historias de Usuario	Nivel de precisión de los requerimientos	15
C6	Videojuego Final	Rendimiento del videojuego	15
C7	Desarrollo del Prototipo	Calidad del Prototipo	10

Elaborado por: Ordóñez, 2022

La encuesta fue realizada con el propósito de poder evaluar el desempeño de las metodologías seleccionadas por cada grupo para desarrollar los videojuegos respectivos, dicha encuesta está conformada por las siguientes preguntas divididas de acuerdo con cada criterio.

Tabla 2: Criterio Nro. 1 Interacción del equipo

Criterio C1 “Interacción del equipo”

Indicador	Nivel de precisión de la información	
Calificación	20 puntos	
Preguntas	Pregunta Nro. 8	Características de la documentación obtenida
	Pregunta Nro. 22	Cantidad de personas necesarias para la codificación del videojuego
	Pregunta Nro. 25	Apoyo en áreas ajenas
	Pregunta Nro. 21	Comunicación entre desarrolladores y artistas.

Elaborado por: Ordóñez, 2022

Tabla 3: Criterio Nro. 2 Documentación

Criterio C2 “Documentación”

Indicador	Nivel de comunicación	
Calificación	15 puntos	
Preguntas	Pregunta Nro. 4	Comunicación entre desarrolladores y artistas.
	Pregunta Nro. 7	Cantidad de personas necesarias para la codificación del videojuego
	Pregunta Nro. 15	Apoyo en áreas ajenas

Elaborado por: Ordóñez, 2022

Tabla 4: Criterio Nro. 3 Adaptabilidad a los cambios

Criterio C3 “Adaptabilidad a los cambios”

Indicador	Adecuación con los objetivos	
Calificación	15 puntos	
Preguntas	Pregunta Nro. 2	Capacidad de la metodología para responder a los cambios.
	Pregunta Nro. 6	Tiempo de respuesta ante los cambios presentados.
	Pregunta Nro. 20	Estimación de tiempos para cumplimiento de tareas

Elaborado por: Ordóñez, 2022

Tabla 5: Criterio Nro. 4 Ciclo de Vida

Criterio C4 “Ciclo de Vida”

Indicador	Eficacia del proceso de desarrollo	
Calificación	10 puntos	
Preguntas	Pregunta Nro. 5	Estimaciones del trabajo a realizar en las distintas etapas.
	Pregunta Nro. 12	Tiempo de respuesta ante los cambios presentados.
	Pregunta Nro. 13	Medición del trabajo mediante uso de métricas
	Pregunta Nro. 14	Filosofía de diseño
	Pregunta Nro. 19	Falencias en el proceso de desarrollo

Elaborado por: Ordóñez, 2022

Tabla 6: Criterio Nro. 5 Ciclo de Vida

Criterio C5 “Historias de Usuario”

Indicador	Nivel de precisión de los requerimientos	
Calificación	15 puntos	
Preguntas	Pregunta Nro. 3	Definición de requerimientos de manera clara y concisa.
	Pregunta Nro. 11	Frecuencia de reestructuración de requerimientos
	Pregunta Nro. 23	Gestión de las historias de usuario

Elaborado por: Ordóñez, 2022

Tabla 7: Criterio Nro. 6 Historias de Usuario

Criterio C6 “Historias de Usuario”

Indicador	Rendimiento del videojuego	
Calificación	15 puntos	
Preguntas	Pregunta Nro. 16	Grado de aceptación del videojuego final
	Pregunta Nro. 17	Actividades de validación y verificación
	Pregunta Nro. 18	Manejo y Corrección de defectos

Elaborado por: Ordóñez, 2022

Tabla 8: Criterio Nro. 7 Historias de Usuario

Criterio C7 “Desarrollo del Prototipo”

Indicador	Calidad del Prototipo	
Calificación	15 puntos	
Preguntas	Pregunta Nro. 9	Prototipo generado para realizar pruebas
	Pregunta Nro. 10	Mejoras para el próximo prototipo
	Pregunta Nro. 24	Revisión del software (Videojuego)

Elaborado por: Ordóñez, 2022

CAPÍTULO IV: RESULTADOS

Para determinar que metodologías resultan ser las óptimas para el desarrollo de videojuegos, se aplicó el instrumento establecido en el tercer capítulo. Una vez realizado este procedimiento, se inició con el respectivo análisis de datos para establecer una comparativa entre las metodologías de desarrollo y poder evaluarlas de acuerdo con los parámetros previamente establecidos.

4.1 Resultados de la aplicación de la encuesta

La información obtenida ha sido examinada mediante un análisis estadístico que permite visualizar los resultados con claridad a través de gráficos y tablas.

Las siguientes tablas hacen referencia a cada una de las preguntas que componen la encuesta, dichas preguntas se encuentran en el anexo. Los resultados fueron agrupados de acuerdo con las 3 estructuras de preguntas que existen en la encuesta, la tabla Nro. 9 Hace referencia a las preguntas de escala Likert, la tabla Nro. 10 Hace referencias a las preguntas dicotómicas mientras que la tabla Nro. 11 Hace referencia a las preguntas de frecuencia. Cabe recalcar que existen preguntas que no forman parte de las siguientes tablas pues manejan su estructura propia.

Tabla 9: Resultados de las preguntas de tipo Likert en la encuesta

Opciones	Totalmente de acuerdo												De acuerdo												Ni de acuerdo ni desacuerdo											
Pregunta	3	4	5	6	7	8	11	17	18	19	22	24	3	4	5	6	7	8	11	17	18	19	22	24	3	4	5	6	7	8	11	17	18	19	22	24
Scrum	25	60	25	0	5	15	15	10	0	5	0	0	70	30	35	20	75	50	50	75	70	45	90	75	0	5	25	40	10	20	20	5	20	25	5	5
Extremme Programming(XP)	14,2857	19	10	0	0	10	4,8	9,5	0	0	4,8	0	71	67	57	14	71	48	81	67	63	71	85	81	5	14	10	43	19	28	14	14	19	14	10	14
Rational Unified Process (RUP)	0	0	0	0	0	0	0	0	0	0	33	0	67	66	0	83	33	50	66	50	33	17	33	50	0	17	50	0	33	17	17	33	33	17	33	17
Rapid Application Development (RAD)	50	0	0	0	0	0	0	0	0	0	0	0	0	87	50	38	25	50	38	87	38	13	63	25	50	0	38	63	63	25	37	13	62	0	37	50
Total %	14,5	29	13	0	2	9	7	7	0	2	2	0	67	56	42	27	62	49	62	71	58	47	78	67	9	9	24	41	18	24	20	11	29	16	11	16

Opciones	En desacuerdo												Totalmente en desacuerdo												Peso										
Pregunta	3	4	5	6	7	8	11	17	18	19	22	24	3	4	5	6	7	8	11	17	18	19	22	24	3	4	5	6	7	8	11	17	18	19	22	23
Scrum	5	5	15	40	10	15	15	10	10	25	5	20	0	0	0	0	0	0	0	0	0	0	0	0	5	5	3	3	5	3	3	4	4	3	5	4
Extremme Programming(XP)	10	0	23	33	10	14	0	10	14	15	0	5	0	0	0	5	0	0	0	0	0	0	0	0	5	4	3	3	4	3	5	5	5	4	5	5
Rational Unified Process (RUP)	33	17	50	17	33	14	17	17	33	66	0	33	0	0	0	0	0	0	0	0	0	0	0	0	4	3	1	1	2	1	3	3	3	1	4	3
Rapid Application Development (RAD)	0	13	13	0	12	25	25	0	0	87	0	25	0	0	0	0	0	0	0	0	0	0	0	0	3	5	2	2	2	1	2	2	2	1	3	2
Total %	9	6	22	31	18	18	11	11	13	35	9	17	0	0	0	1	0	0	0	0	0	0	0	0												

Elaborado por: Ordóñez, 2022

Las pregunta Nro. 3 muestra que un 67,3 % de los encuestados indica que la metodología aplicada fue altamente efectiva para el desarrollo del videojuego, esta respuesta esta reforzada con las respuestas a las preguntas Nro. 3, Nro. 4 y Nro. 5 en donde se indica

que la metodología también fue efectiva para responder a los cambios y también permitió plantear con claridad los requerimientos funcionales y no funcionales, por otro lado en la pregunta Nro. 5 un 12,7% de los encuestados indican haber alcanzado una correcta comunicación entre los miembros del equipo.

El hecho de que los equipos de desarrollo hayan sido capaces de realizar una correcta ingeniería de requerimientos y que hayan podido adaptarse a los cambios que se suscitaron muestra que la metodología que aplicaron resulto efectiva, sin embargo, en la comunicación del equipo se muestra que aun cuando pudieron trabajar juntos, pudieron existir varios errores de comunicación entre los miembros del equipo.

En la pregunta 6, un 61,8% de los encuestados señalan que el equipo fue capaz de responder a los cambios en los tiempos adecuados, esto se contrasta con las preguntas Nro. 7, Nro. 11 en donde un 49,1% de los encuestados indica que el número de programadores fue suficiente para adaptar estos cambios al desarrollo del juego, no obstante, resulta llamativo el hecho de que solo el 61,8% de los equipos haya considerado suficiente la retroalimentación del jugador para encontrar defectos.

Estas respuestas muestran que los equipos desarrollo en su mayoría, han logrado distribuir correctamente las tareas de acuerdo con la cantidad de miembros del equipo y los roles que desempeñan, logrando así generar un prototipo continuo perfectamente funcional en cada iteración.

Para la pregunta Nro. 17 un 71% de los encuestados muestran haber alcanzado el objetivo que tenía con el videojuego cumpliendo las expectativas que había alrededor del mismo. En las siguientes preguntas Nro.18 y Nro. 19 las respuestas sobre la gestión de calidad del videojuego muestran que un 58,2% de los equipos mantuvo una correcta gestión de calidad del videojuego, esto se complementa con la pregunta Nro. 19 en la que los equipos muestran haber destinado una considerable cantidad de tiempo a correcciones de defectos.

Los bugs afectan en gran medida la experiencia que el jugador tiene con el videojuego y el hecho de que los equipos en su mayoría hayan logrado corregir defectos manteniendo una correcta gestión de calidad hace que el videojuego logre cumplir con las expectativas iniciales.

La siguiente tabla de tipo está compuesta por preguntas dicotómicas.

Tabla 10: Resultados de las preguntas dicotómicas en la encuesta

Opciones	Si			No			Peso		
Pregunta / Metodología	**10**	**16**	**21**	**10**	**16**	**21**	**10**	**16**	**21**
Scrum	90	80	45	10	20	55	5	5	3
Extremme Programming(XP)	71,4286	76	52	29	24	48	4	4	3
Rational Unified Process (RUP)	55	70	83	45	30	17	3	4	1
Rapid Application Development (RAD)	38	63	63	62	37	37	2	3	2
Total %	67	75	45	33	25	55			

Elaborado por: Ordóñez, 2022

De acuerdo con la tabla Nro. 10. La pregunta 10 el 67,3 % de los equipos fue capaz de entregar un prototipo al final de cada iteración, lo que demuestra un correcto y organizado proceso de desarrollo en la mayoría de los equipos, en contraste con esta cifra las respuestas de la pregunta 15 muestra que un 74,5 % de los usuarios participaron también en otras tareas ajenas a su rol lo cual es ventajoso si el equipo mantiene autoorganización. En la pregunta Nro. 21 justamente se puede notar que un 54,5 de los equipos tuvo que recurrir a Crunch para completar sus tareas.

El Crunch es una práctica negativa en el desarrollo de videojuegos provocando que el equipo caiga en una sobrecarga de tareas que puede terminar por saturar el proceso completamente.

La siguiente tabla está compuesta por preguntas dicotómicas.

Tabla 11: Resultados de las preguntas de frecuencia en la encuesta

	Porcentaje de participantes (%)											
Opciones	Baja			Media			Alta			Peso		
Pregunta / Metodología	**12**	**13**	**26**	**12**	**13**	**26**	**12**	**13**	**26**	**12**	**13**	**26**
Scrum	5	20	0	65	65	65	30	15	35	3	3	4
Extremme Programming(XP)	9,5238	10	5	71	71	62	19	19	33	3	3	4
Rational Unified Process (RUP)	0	67	0	33	33	50	67	0	50	2	2	3
Rapid Application Development (RAD)	0	0	0	38	38	62	62	62	38	2	4	4

Elaborado por: Ordóñez, 2022

La pregunta Nro. 12 muestra que apenas un 12,7% de los equipos pudo reutilizar código en mayor medida, esta práctica trae beneficios notables en el desarrollo de videojuegos, sin embargo, el hecho de que se evidencia esta práctica en un pequeño porcentaje de los equipos muestra que tuvieron inconvenientes a la hora de aplicar una arquitectura de software en el videojuego.

Las siguientes preguntas manejan su propia estructura, razón por las cual sus resultados están expresados en las siguientes figuras.

La siguiente figura muestra los resultados de la pregunta Nro. 14 los equipos muestran haberse guiado mayoritariamente por métricas de defectos con un 52,7% y de usabilidad con un 23,6%, esto es acertado ya que se enfocan en entregar un producto que escucha al jugador y que al mismo tiempo alcanza estándares de calidad.

Pregunta 14

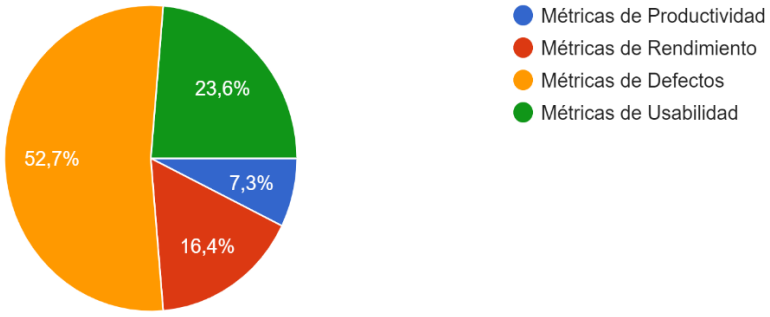


Figura 6: Utilización de métricas para medir el desempeño del videojuego.

Fuente: Encuesta realizada a profesionales o estudiantes que han participado en el desarrollo un videojuego

La siguiente figura muestra que un 70.9% de los encuestados se enfocaron un trabajar con un diseño centrado en el usuario, mientras que un 18.2% se enfocaron en un diseño centrado en el contenido, dejando así al 10.9% restante de encuestados que se enfocaron en un diseño centrado en la tecnología. Evidentemente el diseño centrado en el usuario resulta ser el más utilizado ya que por tratarse de un videojuego, la experiencia final que tenga el jugador es el factor que determina el éxito del videojuego.

Pregunta 15

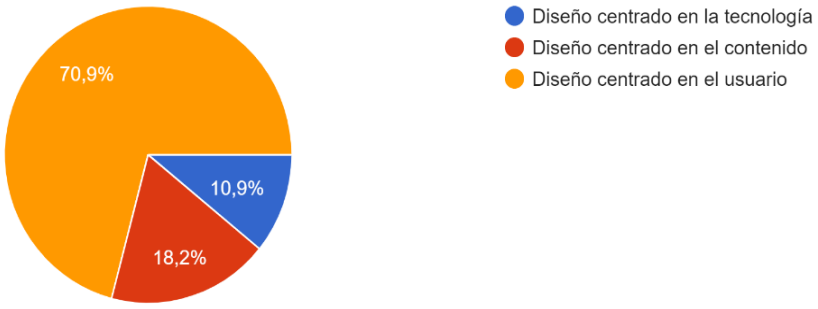


Figura 7: Enfoque de diseño en el que se centró el equipo.

Fuente: Encuesta realizada a profesionales o estudiantes que han participado en el desarrollo un videojuego

De acuerdo con la figura 8, un 58.2% de los encuestados respondieron que programadores y diseñadores participaron en la elaboración del documento de diseño, mientras que un 34.5% indica que el documento de diseño estuvo a cargo únicamente por programadores, en contraste, un 7.3% indicó que el documento de diseño estuvo a cargo de los miembros enfocados al área artística, estas respuestas indican que en la mayoría de los equipos existió una colaboración entre los miembros de ambas áreas haciendo así que el documento de diseño sea entendido por ambas partes.

Pregunta 23

¿Quiénes participaron en la creación del documento de diseño del videojuego?

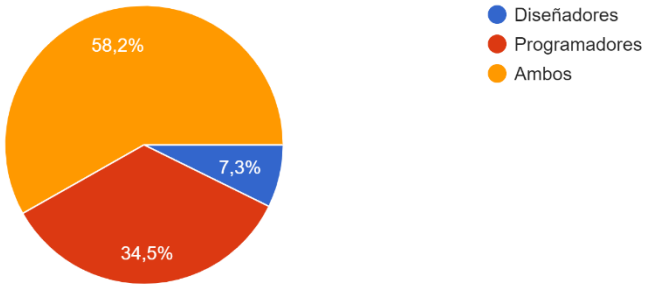


Figura 8: Colaboración entre los miembros para elaborar el documento de diseño.
Fuente: Encuesta realizada a profesionales o estudiantes que han participado en el desarrollo un videojuego

Pregunta 9

La documentación es un aspecto importante para el desarrollo del videojuego, por lo cual la figura 9 muestra la manera en que los equipos terminaron generando la documentación respectiva, se puede constatar que en un 40% de los encuestados consideraron a la documentación final como “Corta” pero al mismo tiempo “Ambigua.

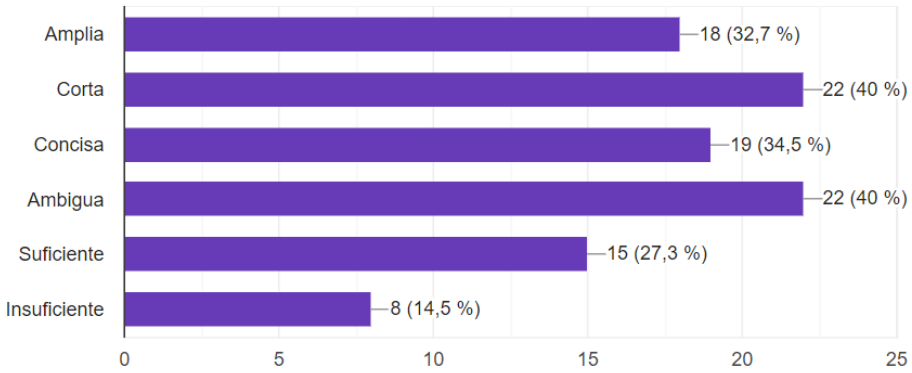


Figura 9: Características de la documentación generada.
Fuente: Encuesta realizada a profesionales o estudiantes que han participado en el desarrollo un videojuego

Estos resultados demuestran la manera en que cada equipo llevo a cabo el proceso de documentación, la ambigüedad de la documentación que se presentó en un alto número de equipos es el resultado de la participación en la creación de esta, por parte de los artistas y los programadores, lo que deriva en un documento de difícil comprensión para ambas partes.

La figura 10 indica que un 36.4% de los encuestados realizó pruebas de usabilidad, sucede de igual manera con las pruebas de aceptación en donde se observa que un 36.4% de los encuestados aplicaron estas pruebas a los prototipos, por otro lado, un 16.4% aplicó pruebas unitarias, y apenas un 10.9% aplicó pruebas de rendimiento. El hecho de que la mayoría de los equipos hayan aplicado pruebas de usabilidad a los prototipos implica que se enfocaron en trabajar de cerca con el usuario para que pueda entregar retroalimentación útil para el desarrollo.

Pregunta 25

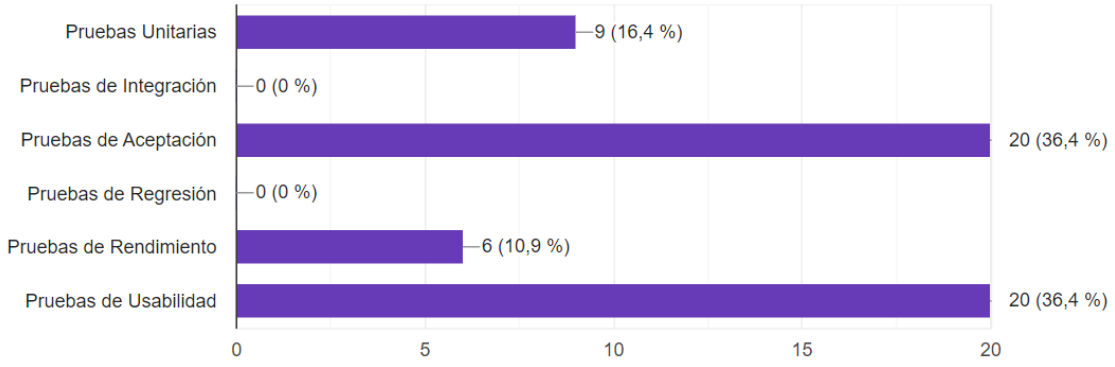


Figura 10: Pruebas realizadas para corrección de defectos.

Fuente: Encuesta realizada a profesionales o estudiantes que han participado en el desarrollo un videojuego

4.2 Resultado de la aplicación de la evaluación a metodologías de desarrollo identificadas.

El modelo de evaluación descrito en el tercer capítulo servirá para evaluar a las metodologías identificadas. Los resultados y valoraciones de esta evaluación son presentados a través de tablas y figuras que establecen una comparativa entre las metodologías identificadas, de esta forma se adquiere una valoración general de cada metodología y se podrá determinar cuáles metodologías resultan ser óptimas para el desarrollo de videojuegos.

Primeramente, se procedió a determinar que metodologías habían utilizado los equipos de desarrollo. Los resultados fueron los siguientes.

Tabla 12: Metodologías utilizadas por los equipos de desarrollo

Scrum	Extreme Programming(Xp)	Rational Unified Process(RUP)	Rapid Apliccation Development(RAD)		
36.4%	38.2%	12.7%	12.7%		

Las metodologías más utilizadas fueron Scrum y Extreme Programming, mientras que las metodologías RUP y RAD también fueron aplicadas, pero en menor medida, de manera que el modelo de evaluación contemplara evaluar estas 4 metodologías identificadas.

Cada pregunta ha sido evaluada de acuerdo con un peso del 1 al 5, como se mencionó en el capítulo III, estos pesos pueden encontrarse en las tablas del subcapítulo 4.1.

Primer criterio de evaluación

De acuerdo con la tabla Nro. 9 y la tabla Nro. 10. Las preguntas que evalúan este criterio corresponden a las preguntas Nro. 5, Nro. 8 y Nro. 16

La evaluación del primer criterio evidencia que, las metodologías Scrum y XP obtuvieron el puntaje más alto, lo que implica que el nivel de comunicación que tuvieron los miembros del equipo no es deficiente, pero es mejorable, mientras que para la pregunta Nro.16 se logra determinar que la distribución de miembros enfocados al área de desarrollo y al área artística fue correcta en el caso de las 4 metodologías. Tanto la metodología RUP como RAD obtuvieron puntajes por debajo de Scrum y XP, lo que significa que los equipos que trabajaron bajo estas metodologías tuvieron varios problemas de comunicación.

Segundo criterio de evaluación

De acuerdo con la tabla Nro. 9 y la tabla Nro. 11. Las preguntas que evalúan este criterio corresponden a las preguntas Nro. 9, Nro. 22, Nro. 23 y Nro. 26.

La evaluación del segundo criterio evidencia que las metodologías Scrum y XP le permitieron al equipo generar documentación más clara y concisa al igual que un documento de diseño completo cuya elaboración estuvo a cargo tanto de la parte técnica como de la parte artística.

En el caso de las preguntas Nro. 9 y Nro. 26 se puede constatar que el documento de diseño fue adecuado para el desarrollo de los videojuegos ya que no resultó en un documento demasiado extenso y que puede ser entendido por ambas partes del equipo de desarrollo.

Tercer criterio de evaluación

De acuerdo con la tabla Nro. 9 y la tabla Nro. 10. Las preguntas que evalúan este criterio corresponden a las preguntas Nro. 3, Nro. 7 y Nro. 21.
La evaluación del tercer criterio muestra que los equipos que trabajaron con las metodologías Scrum, XP y RUP pudieron adaptarse a los cambios que se presentaban constantemente durante el desarrollo, no obstante, de acuerdo con las respuestas de la pregunta Nro. 21 varios equipos no fueron capaces de evitar el Crunch.

Cuarto criterio de evaluación

De acuerdo con la tabla Nro. 9 y la tabla Nro. 11. Las preguntas que evalúan este criterio corresponden a las preguntas Nro. 6, Nro. 13, Nro. 14, Nro. 15 y Nro. 20.

La evaluación del cuarto criterio consta de dos aspectos, el primer aspecto este compuesto por las preguntas Nro. 6 y Nro. 13. Estas respuestas permitieron determinar que los equipos que trabajaron bajo las metodologías Scrum y XP tuvieron mayor eficacia en cumplir con las planificaciones.

El segundo aspecto de evaluación está constituido por las preguntas Nro. 14, Nro. 15 y Nro. 20, bajo esta evaluación se pudo determinar que para el caso de las 4 metodologías

que utilizaron los equipos, tanto las métricas de defectos como las métricas de usabilidad fueron las más aplicadas para monitorear el estado del videojuego.

Quinto criterio de evaluación

De acuerdo con la tabla Nro. 9 y la tabla Nro. 11. Las preguntas que evalúan este criterio corresponden a las preguntas Nro. 4, Nro. 12 y Nro. 24.
La evaluación del quinto criterio permite evidenciar que las metodologías Scrum y XP resultan ser las que obtuvieron una mejor valoración, seguidas de RAD y al final RUP, sin embargo, como muestra la pregunta Nro. 12, los equipos no fueron capaces de mantener una consistencia en los requerimientos.

Sexto criterio de evaluación

De acuerdo con la tabla Nro. 9. Las preguntas que evalúan este criterio corresponden a las preguntas Nro. 17, Nro. 18 y Nro. 19.
La evaluación del sexto criterio permite evidenciar que las metodologías Scrum y XP terminaron obteniendo puntajes mayores en comparación con RUP y RAD. Los equipos que trabajaron con las metodologías Scrum y XP indican que el videojuego obtenido al final del desarrollo cumplió con las expectativas la visión que cada equipo tenía del videojuego, no obstante, la pregunta Nro. 19 si permite constatar que solo los equipos que trabajaron basándose en la metodología XP pudieron darle una correcta calidad técnica al videojuego al haber sido capaces de detectar y corregir defectos antes de finalizar el videojuego.

Séptimo criterio de evaluación

De acuerdo con la tabla Nro. 9 y la tabla Nro. 10. Las preguntas que evalúan este criterio corresponden a las preguntas Nro. 10, Nro. 11 y Nro. 25.
Se puede evidenciar que los equipos que trabajaron con metodología RUP aplicaron casi en su totalidad pruebas de aceptación a los prototipos, mientras que para los equipos que trabajaron con las 3 metodologías restantes también aplicaron pruebas de aceptación mayoritariamente, aunque en menor grado, ya que también se puede ver que aplicaron pruebas unitarias. Por otro lado, la mayoría de los equipos que trabajaron con metodologías Scrum, XP y RUP lograron desarrollar prototipos funcionales al final de

cada iteración, no siendo el caso de los equipos que trabajaron mediante la metodología RAD ya que un porcentaje menor de ellos pudo desarrollar el prototipo funcional.

4.2.1 Valoración general y selección de metodologías

Las metodologías identificadas fueron evaluadas de acuerdo con cada uno de los criterios establecidos, cada uno de estos criterios permitió identificar los aspectos en donde las metodologías presentaban una mayor facilidad o dificultad para adaptarse al desarrollo de los respectivos videojuegos construidos por los distintos equipos de desarrollo a quienes se les realizaron las encuestas. Una vez obtenidos los resultados de cada uno de los criterios de evaluación, se realizó una valoración general para determinar cuáles son las metodologías más efectivas para la creación de videojuegos.

Tabla 13: Valoración general de las metodologías identificadas

Metodología	C1	C2	C3	C4	C5	C6	C7	Valoración Total
Scrum	11	17	13	6	12	11	8	78
XP	10	17	12	6	12	14	9	80
RUP	6	11	7	3	8	7	6	48
RAD	6	11	7	6	9	5	4	48

Elaborado por: Ordóñez, 2022

La valoración general de la tabla 52 muestra que las metodologías XP y Scrum obtuvieron las puntuaciones más altas, y en todos los criterios exceptuando el criterio C4, obtuvieron puntajes altos, mientras que las metodologías RUP y RAD obtuvieron una valoración general media. De esta manera se concluye que los equipos que trabajaron bajo las metodologías Scrum y XP tuvieron mejores resultados en el desarrollo de los respectivos videojuegos.

CAPÍTULO V: PROPUESTA DE METODOLOGÍA ÁGIL PARA EL DESARROLLO DE VIDEOJUEGOS

Adaptar una metodología de desarrollo puede resultar complejo cuando se trata de crear videojuegos. En los capítulos anteriores se pudo constatar como los equipos de desarrollo pudieron crear sus videojuegos basándose en distintas metodologías, sin embargo, aun cuando los equipos pudieron publicar sus videojuegos, estos no lograron satisfacer completamente la visión que tenía cada equipo sobre el videojuego final, esto se pudo evidenciar mediante le evaluación realizada a cada metodología, y en donde se pudo determinar los aspectos en donde cada metodología presentaba mayores dificultades, de igual manera se pudo determinar que metodologías habían resultado más óptimas para el desarrollo de videojuegos.

5.1 Consideraciones Generales

La razón por la cual se establece propuesta de una nueva metodología que está basada en otras metodologías es que las metodologías de desarrollo que fueron aplicadas por los equipos son eficaces en varios aspectos fundamentales para el desarrollo de videojuegos, de manera que establecer una nueva metodología sin tomar en consideración las metodologías ya existentes no resulta muy práctico debido a que estas metodologías ya presentan fortalezas en el desarrollo de videojuegos, como sucede justamente con las metodologías Scrum y XP.

Consideraciones del proceso de desarrollo

De acuerdo con los resultados obtenidos después de la aplicación de la evaluación, se pudo evidenciar que el desarrollo de cada uno de los videojuegos estuvo sujeto a constantes cambios, de ahí que sea imperativo la utilización de una metodología ágil con una respuesta eficaz a los cambios. Debido a esto la metodología está basada en un desarrollo incremental e iterativo, de manera que la metodología responde a los principios de un sistema incremental.

Para la creación y estructuración de la metodología, se toman en cuenta los criterios establecidos en el capítulo anterior, dichos criterios representan los aspectos fundamentales que debe tener la metodología, la metodología estará conformada 3 fases principales, dichas fases son las siguientes:

- Preproducción
- Producción
- Postproducción

Consideraciones del equipo de desarrollo

La comunicación y las interacciones del equipo de desarrollo son fundamentales para la consecución de un videojuego de calidad. La metodología toma en cuenta mayoritariamente las prácticas y principios de trabajo en equipo, así como distribuciones de las tareas y organizaciones de tiempo, para esto la metodología se basa en mayor medida en los valores de Scrum, en donde se establecen factores determinantes para la comunicación del equipo, para esto se adaptarán estos valores y principios al ambiente característico del desarrollo de un videojuego, se consideran los siguientes aspectos esenciales:

Consideraciones de las prácticas de la metodología

La metodología está basada en los principios de Scrum y XP, por esta razón, la metodología establece un conjunto de prácticas enfocadas con la organización y la gestión del trabajo, sin dejar de lado las prácticas enfocadas a la programación.

Según Kniberg (2007), Scrum y XP pueden unirse a través de prácticas comunes. Las prácticas de scrum le sirven al equipo para controlar aspectos de gestión y organización, mientras que las prácticas propias de XP servirán para enfocarse en el apartado de programación.

Finalmente, lo que se busca con la metodología es darles a los equipos de desarrollo una guía que les permita llevar la creación del videojuego de una forma estandarizada y que está basada en prácticas de ingeniería de software, de manera que puedan crear un videojuego que cumpla con las expectativas y la visión que se tiene sobre el mismo, pero que al mismo tiempo pueda ser considerado un software de calidad.

5.2 Marco conceptual y delimitación

Scrum

Este marco de trabajo provee a un equipo de desarrollo un conjunto de prácticas que les permitirán alcanzar su objetivo, a la vez que cumplen con los principios del agilismo, optimizando tiempo y costos. La consecución del objetivo se va logrando de manera incremental, es decir que el equipo no cosechará logros de manera inmediata, sino que, deben darse varias iteraciones o "sprints" para poder cumplir con el objetivo (Subra & Vannieuwenhuyse, 2018).

- **Scrum Máster:** Esta persona se encarga de actuar como un facilitador entre las partes interesadas en el proyecto y el equipo de desarrollo, eliminando todo tipo de obstáculos que impidan una correcta comunicación entre las partes interesadas.
- **Product Owner:** Esta persona es quien se comunica directamente con el cliente, para poder definir un listado de requerimientos, él debe encargarse de traducir todos estos requerimientos a documentos que el equipo de desarrollo pueda entender.
- **Equipo de desarrollo:** El equipo se encarga de llevar a cabo la ejecución de todas las tareas necesarias para cumplir con los requerimientos que el cliente ha establecido.

Deemer et al. (2009), plantea la siguiente serie de pilares fundamentales para la metodología:

Pilares de Scrum

- Compromiso: el equipo entero debe comprometerse a desempeñar todas las actividades requeridas para el sprint.
- Inspección: El equipo debe llevar a cabo siempre el sprint diario para poder revisar el avance del proyecto y como está trabajando el equipo.
- Adaptación: Los equipos deben saber organizar y administrar los cambios que se produzcan.

León (2014), sostiene que Scrum no es una metodología rígida y que sus procesos se van generando de acuerdo con adaptaciones, de manera que plantea los siguientes eventos;

Eventos de Scrum

- Planificación del Sprint: Estimación del tiempo que tomará cumplir las historias de usuario y las actividades necesarias para cumplir con el sprint.
- Scrum Diario: Reunión diaria que el equipo debe realizar y que no debe durar más de 15 minutos, para comunicar retroalimentación útil sobre el avance de las tareas.
- Revisión del Sprint: Sucede al final del sprint en donde el equipo debe encargarse de comprobar y validar las funcionalidades llevadas a cabo en el sprint.

Scrum en videojuegos

En la industria de videojuegos, se han adoptado muchas metodologías para crear videojuegos, en sus inicios, cuando los videojuegos eran menos complejos que ahora, varias empresas utilizaban metodologías más rígidas y tradicionales como la metodología en cascada, sin embargo, en los últimos años y debido al aumento en la complejidad de videojuegos, varias empresas empezaron a utilizar metodologías ágiles (Kristiadi et al., 2019).

Un estudio efectuado por Jondya et al. (2022), establece que Scrum esta entre las metodologías que mejor han logrado adaptarse al desarrollo de videojuegos, existiendo varios casos documentados en los últimos años.

Un modelo establecido por Kristiadi et al. (2019), establece la siguiente estructura para trabajar los Sprint del proyecto de manera que se puedan adecuar al desarrollo de videojuegos.

Preproducción del videojuego con scrum

La etapa de preproducción es una etapa de planificación, durante este intervalo de tiempo, el equipo debe generar documentos que contengan descripciones y especificaciones de como estará construido el videojuego.

Producción del videojuego con scrum

En esta etapa empieza el desarrollo de todos los requerimientos planteados en el Product Backlog, para esto se deben realizar el siguiente proceso.

- Diseño
- Desarrollo
- Pruebas

- Revisión

Postproducción

El videojuego ha sido probado y aceptado por todas las partes interesadas, por lo tanto, se procede con el lanzamiento del videojuego.

A juicio del autor, este modelo podría resultar útil si el equipo de desarrollo fuera amplio, más sin embargo si se trata de un equipo con pocos miembros, podría desencadenar en un entorno poco favorable, de ahí la importancia de la autoorganización del equipo, pues ellos son quienes determinan la estructura más conveniente del equipo de desarrollo, pudiendo reestructurarla de ser necesario.

Extreme Programming (XP)

Es una metodología de desarrollo ágil que le permite a un equipo de desarrollo producir un software de calidad de una manera ágil, XP a diferencia de Scrum se fundamenta principalmente en una serie de prácticas que se ejecutan simultáneamente para garantizar la calidad del software (Bahit, 2012).

De acuerdo con Meléndez et al. (2016), XP resulta una metodología especialmente útil para el desarrollo de videojuegos considerando que los roles que maneja se adecuan perfectamente al desarrollo común de un videojuego, planteando así los siguientes roles.

Roles de XP

- **Cliente:** Es quien establece y prioriza las funcionalidades que tendrá el software.
- **Programador:** Es quien debe encargarse de realizar la codificación de las historias de usuario planteadas por el cliente.
- **Tester:** Es quien está enfocado en realizar pruebas para encontrar defectos, de igual manera es responsable de proponer y aplicar herramientas de soporte.
- **Entrenador:** Es la persona que tiene vastos conocimientos en la metodología, de manera que actúa como el responsable del proceso, guiando a todo el equipo e instruyéndolos para que se cumplan las prácticas y principios de XP.
- **Consultor:** Actúa como un miembro externo al equipo de desarrollo ya que interviene en momentos puntuales del proceso de desarrollo para asesorar sobre algún tema en específico.

Jefe de proyecto: Se encarga de conseguir todos los recursos necesarios para lidiar con los problemas que se presenten, también es quien establece reuniones continuas entre el cliente y el equipo de desarrollo.

Extreme Programming aplicado a videojuegos

Según un estudio realizado por Schofield (2007), señala que mediante las prácticas de XP es posible permitirles a los programadores mantener un código flexible y que este sustentado en la simplicidad, permitiendo así que el equipo de desarrollo pueda adaptarse a los cambios a la vez que se mantenga el videojuego en una dirección positiva, además señala que las prácticas de XP resultan especialmente útiles para adaptarse a los cambios más comunes que pueden ocurrir en un videojuego. Dichas prácticas son las siguientes:

TDD

El control de defectos es importante para dejar al videojuego libre de bugs, por tal razón, esta práctica común de XP resulta ventajosa para el desarrollo de videojuegos, puesto que les permite a los programadores escribir pequeños pedazos de código que funcionan a manera de pruebas unitarias, las cuales describen como desean que actúe una funcionalidad, esto resulta especialmente útil debido a que en los videojuegos, existen múltiples elementos o componentes que tienen un mismo comportamiento.

Programación en pares

Esta práctica indica que dos programadores trabajan en un mismo computador para producir un código de mejor calidad, mientras un programador escribe y decide como será el código, el otro programador participa proponiendo ideas que ayuden a mejorar el código escrito a medida que lo va revisando, esto resulta especialmente útil en el entorno de desarrollo de videojuegos puesto que los programadores pueden aportar con nuevas ideas de diseño o mejoras que puedan agregarle innovación al videojuego.

Diseño Continuo

Esta práctica establece que el diseño del videojuego no es estático ni tampoco se generaliza antes de empezar a desarrollar el videojuego, sino que se parte de un diseño simple y bien definido, el cual va adquiriendo complejidad dependiendo de cómo avanza el desarrollo del videojuego, esto va ligado con la práctica de refactorizar el código y mantenerlo simple.

Participaciones activas

Esta práctica anima a que el cliente o las partes interesadas en el videojuego traten de participar en etapas más tempranas del desarrollo, es decir, que el cliente pueda utilizar el videojuego antes de que este llegue a su versión final, de esta manera resulta más fácil monitorear el avance del videojuego y comprobar que efectivamente se están cumpliendo las necesidades del cliente.

A juicio del autor, el hecho de que los diseñadores trabajen en conjunto con los programadores puede resultar ventajoso para que puedan comprender como se está dando vida a los personajes y entornos que han creado, por otro lado, la participación del cliente en etapas muy tempranas si bien es cierto puede ayudar a monitorear de mejor manera el avance del videojuego, el hecho de que solo el cliente sea quien juega al videojuego en estas etapas puede no otorgar la retroalimentación que un jugador con más experiencia si puede entregar y que le puede resultar más beneficiosa al equipo de desarrollo.

5.3 Metodología

Una propuesta de metodología ágil específica para el desarrollo de videojuegos surge con el propósito de mitigar estos inconvenientes que tuvieron los equipos de desarrollo, para esto se considera las metodologías mejor puntuadas en el capítulo 4, las cuales resultaron ser las metodologías Scrum y XP, de esta forma surge la metodología AGP (Agile Game Programming), esta metodología estará basada en los principios que manejan tanto Scrum como XP pero tomando en cuenta factores indispensables para el desarrollo de videojuegos.

Al igual que el resto de metodologías de desarrollo existentes, AGP busca facilitar el desarrollo de videojuegos a través de buenas prácticas de ingeniería de software, propiciando un ambiente de trabajo tal que todos los miembros del equipo puedan comunicarse correctamente y se pueda generar una correcta distribución y organización de tareas, cabe destacar que, dentro de AGP se toman en cuenta varios aspectos fundamentales para el desarrollo de software, pero adaptándolos al desarrollo de videojuegos, para conseguir esto, es imperativo que AGP también considere aspectos esenciales en el desarrollo de videojuegos, ya que como se ha visto a lo largo del presente trabajo, un videojuego, además de cumplir con una serie de requerimientos, también debe buscar entregar una experiencia de entretenimiento al usuario mientras cumple con la visión que tiene el equipo de desarrollo sobre el mismo.

5.3.1 Principios de AGP

Cuando se habla de desarrollo de videojuegos, se debe comprender que cada videojuego es un sistema complejo que responde a una serie de reglas y parámetros, de esta forma se entiende que construir un videojuego representa superar una serie de escenarios desafiantes para el equipo de desarrollo (McDaniel & Daer, 2016).

Mediante AGP se busca que el equipo de desarrollo pueda superar estos desafíos mediante la aplicación de procedimientos que estén basados en los valores de las metodologías Scrum y XP, los cuales son perfectamente aplicables al desarrollo de videojuegos.

Según Ahmad et al. (2014), XP toma en cuenta principios que favorecen a pequeños equipos que necesitan desarrollar un software en poco tiempo y el cual está en constante cambio, mientras que, Scrum provee una serie de principios que permiten establecer un liderazgo de proyecto necesario para que el equipo pueda tener una auto organización

óptima, de manera que, AGP adapta estos valores para direccionarlos al desarrollo de videojuegos, a continuación, se muestran dichos valores.

Autogestión: La facultad que tiene el equipo de poder tomar las decisiones que consideren más oportunas para la resolución de cada tarea, estas decisiones se relacionan con la planeación de cada iteración.

Autoorganización: La capacidad que tiene el equipo para designar la cantidad de miembros que van a participar en las distintas tareas de cada iteración. Los cambios constantes requieren una adaptación por parte del equipo por lo cual, de ser necesario, la estructura del equipo pueda cambiar, es decir que cada miembro del equipo pueda trabajar de manera dinámica y pueda trabajar en una tarea distinta en las iteraciones posteriores.

Compromiso: Cada miembro del equipo debe entender y aceptar el grado de responsabilidad que tiene con el proyecto.

Simplicidad: El videojuego debe ser conciso y cumplir con las necesidades específicas, el hecho de querer hacer al videojuego más llamativo puede terminar por incrementar la complejidad de este y que el equipo se vea superado e incapaz de cumplir con la visión central del videojuego.

Comunicación: La capacidad del equipo para poder entenderse a través de lenguaje natural y de la documentación generada.

Feedback: La capacidad del equipo para poder entregar prototipos funcionales que muestren información clara sobre las mejoras que deben realizarse en iteraciones posteriores.

Respeto: El equipo debe encontrarse abierto a nuevas ideas y escuchar posibles soluciones que los miembros del equipo puedan proponer.

Coraje: Capacidad para tomar las mejores decisiones frente a escenarios desfavorables.

Liderazgo: Capacidad para dirigir el equipo, encontrando las mejores formas para aprovechar las habilidades e ideas de los miembros del equipo y tomando decisiones que permitan el éxito del proyecto.

5.3.2 Estructura de AGP

Las prácticas que establece Scrum están orientadas a la gestión y administración de las tareas que se deben completar, mientras que XP provee al equipo una serie de procesos y técnicas enfocadas al apartado técnico, de esta forma se considera que Scrum le sirve al equipo de desarrollo para poder definir que prácticas deben seguir, mientras que XP le indica al equipo como debe llevar a cabo dichas prácticas (Salazar & Tovar & Linares & Lozano & Valbuena, 2018).

Tabla 14: Fusión de prácticas de Scrum con prácticas de XP

Ceremonia	Prácticas de XP
Planeación del Sprint	Marcha Sostenible
Ejecución del Sprint	Programación por pares Integración continua Propiedad colectiva del código Pruebas
Scrum diario	Espacio de Trabajo

Fuente: Salazar & Tovar & Linares & Lozano & Valbuena, 2018

Elaborado por: Ordóñez, 2022

Las prácticas de AGP están basadas en la fusión de las prácticas de Scrum y XP, cabe recalcar que se adaptan las ceremonias de Scrum para la gestión y organización de las tareas de cada sprint, mientras que las prácticas de XP son utilizadas para indicar como los equipos deben llevar a cabo dichas ceremonias.

El proceso para desarrollar un videojuego puede ser visto desde tres etapas generales, las cuales son: preproducción, producción y postproducción. Estas 3 etapas son ampliamente aceptadas en el desarrollo de los videojuegos, por lo cual AGP estará basada en este proceso en cuestión, cabe destacar que dentro de cada etapa se tomarán en cuenta las ceremonias de Scrum y las prácticas de XP como se mencionó anteriormente.

5.3.3 Ciclo de vida de AGP

Preproducción

Esta etapa servirá para que el equipo pueda empezar la planeación del sprint y se pueda definir la visión central del videojuego, por lo tanto, a continuación, se establecen las prácticas que se llevarán a cabo durante esta etapa.

Tabla 15: Prácticas de la Etapa de Preproducción

Ceremonias	**Prácticas**
Planificación del Sprint	**Ritmo sostenible:** Las tareas que se realicen a diario deben ser correctamente distribuidas, es decir que el tiempo que se requiera para cumplir las tareas debe ser equitativo, no se permite días con sobrecarga de tareas.
	Equipo multidisciplinar: El equipo completo debe estar presente en las reuniones de planificación del sprint, de igual manera se debe mantener una comunicación constante entre ambas partes durante todo el proceso de desarrollo.
	Espacio de trabajo: Se debe ir monitoreando el trabajo del equipo a través del control de tareas que se están ejecutando y las que se encuentran pendientes, para esto se aplica un burndown chart y un tablero de notas (físico o digital)
	Especificación de requerimientos: La definición de los requerimientos debe estar documentada tal que la visión central y las características fundamentales del videojuego estén claramente definidas y comprendidas tanto por el equipo de desarrollo como por el cliente, en este caso el cliente es el jugador. Para esto, se deben definir los siguientes documentos. • GDD (Game Design Document) • Game Backlog
	Arquitectura de alto nivel: Se define la estructura global que debe tener un videojuego tomando en cuenta componentes, patrones de diseño y características básicas.

Elaborado por: Ordóñez, 2022

Producción

En esta fase el equipo se dedica a ejecutar todas las tareas que fueron planeadas de acuerdo con la visión general y los requerimientos que se hayan definido en la etapa anterior, de igual manera en esta fase se produce una refinación de los requerimientos y del prototipo en general, por lo tanto, a continuación, se establecen las prácticas que se llevarán a cabo durante esta etapa.

Tabla 16: Prácticas de la Etapa de Producción

Ceremonias	**Prácticas**
Ejecución del sprint	**Código Colectivo:** El código fuente debe ser compartido con todo el equipo a través del uso de repositorios. Cada miembro del equipo debe tener acceso a estos repositorios.
	Integración Continua: Los cambios deben ser implementados con la mayor brevedad posible. Es imperativo que el equipo se encuentre actualizando constantemente las ramas del repositorio y realizando pruebas para comprobar que los componentes actualizados se integren sin afectar el funcionamiento general del sistema. La metodología insta a la utilización de herramientas para proyectos ágiles que automaticen el despliegue del sistema para que este se realice continuamente, cada vez que un nuevo cambio es insertado, de esta forma se optimiza considerablemente el tiempo que el equipo requiere para realizar pruebas posteriores a los cambios implementados.
	Pair Programing: Se insta a que se realice programación en parejas para disminuir el número de defectos y sobre todo permitir que los miembros del equipo en el área técnica puedan comprender de mejor manera el funcionamiento general del sistema desde las bases.
	Pruebas: El prototipo debe estar sometido constantemente a pruebas unitarias que permitan recopilar retroalimentación para la corrección de errores o la implementación de mejoras.
Revisión del Sprint	**Refinamiento del Game Backlog:** Actualización continua de los requerimientos del Game Backlog. Para esto, todo el equipo debe

	revisar el grado de detalle de cada requerimiento y adaptarlo de acuerdo con los nuevos cambios que hayan surgido, cabe destacar que de igual manera se debe actualizar la priorización de los requerimientos cada vez que se añadan más detalles.
Reuniones diarias	El equipo debe mantener reuniones diarias para monitorear el avance de las tareas y el proyecto, así como notificar acerca de algún posible inconveniente.

Elaborado por: Ordóñez, 2022

Postproducción

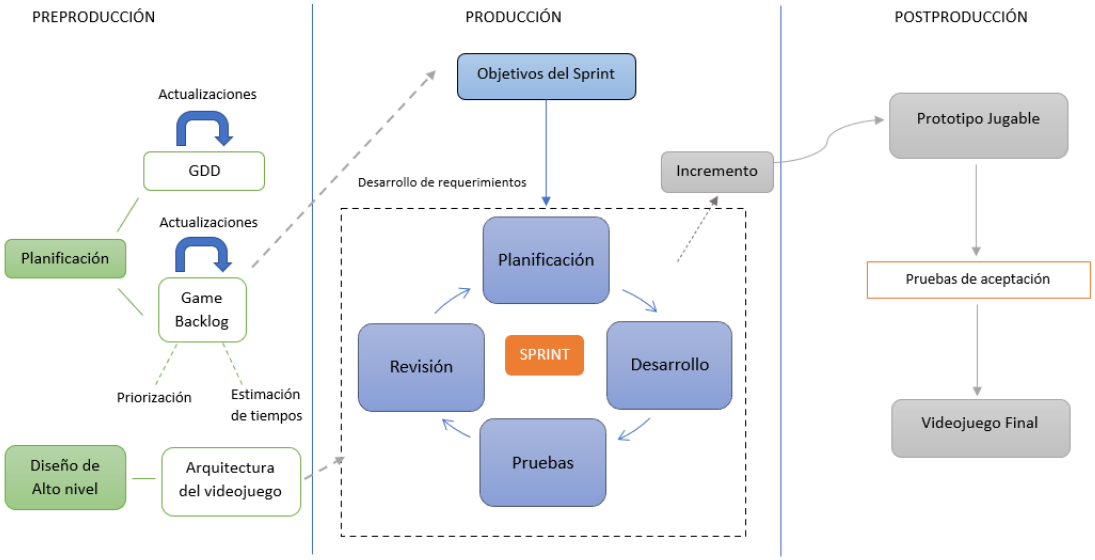
En esta etapa el equipo ya tiene una versión final del videojuego, de manera que, a través de esta versión, pueden constatar el funcionamiento general del videojuego y determinar si ha cumplido con las expectativas o se le pueden realizar ciertas mejoras, por lo tanto, a continuación, se establecen las prácticas que se llevarán a cabo durante esta etapa

Tabla 17: Prácticas de la Etapa de Postproducción

Ceremonias	**Prácticas**
Refinamiento	**Refinamiento de la versión final:** El videojuego debe contar con una versión beta que permita verificar y comprobar el comportamiento general del mismo, para esto, la versión beta debe ser abierta a un selecto grupo de jugadores que puedan otorgar retroalimentación entorno a la versión actual del videojuego. Una vez el videojuego logra cumplir con los objetivos, se procede al lanzamiento oficial.

Elaborado por: Ordóñez, 2022

Figura 11: Ciclo de Vida de AGP.



Elaborado por: Ordóñez, 2022

5.3.4 Roles de AGP

Scrum y XP manejan una serie de roles definidos que para el entorno del desarrollo de videojuegos puede resultar muy rígido, sobre todo porque no se toman en cuenta los roles enfocados al apartado artístico. AGP se inspira mayoritariamente en los roles que maneja Scrum, pero tomando en cuenta también nuevos roles complementarios que son bastante comunes en los equipos de trabajo que desarrollan un videojuego.

AGP consta de dos tipos de roles globales, los roles principales y los roles externos, los cuales están compuestos de la siguiente manera.

Roles principales

- AGP Máster (Líder del proyecto)
- Equipo de desarrollo
- Game Owner

Roles externos

- Jugador
- Stakeholder

Los roles principales fungen como los miembros que son indispensables para la construcción del videojuego, mientras que los roles externos funcionan mayoritariamente

como miembros que intervienen en momentos puntuales del desarrollo del videojuego y que realmente no están involucrados en el proyecto desde su etapa inicial.

AGP Máster

Al tratarse de una metodología nueva, es trabajo del AGP máster instruir al equipo sobre cómo aplicar esta metodología y cómo manejarla durante todo el proceso de creación del videojuego. El AGP máster debe guiar a ambas partes del equipo a través a los principios de la metodología con el fin de cumplir los objetivos que se hayan planteado.

Funciones

- Facilitar la comunicación entre el equipo de desarrollo y el Game Owner.
- Facilitar la comunicación entre ambas partes del equipo.
- Responder y actuar rápidamente ante los posibles cambios que se puedan producir durante el desarrollo.
- Mantener al equipo motivado.
- Monitorear el desempeño del equipo y el cumplimiento de las tareas de acuerdo con las estimaciones realizadas.
- Mantener un alto grado de disponibilidad para atender a cualquier conflicto que pueda presentar el equipo y que impida el avance con normalidad del proceso de desarrollo.
- Mantener neutralidad y mostrar apoyo a ambas partes del equipo por igual.
- Mantenerse actualizado sobre herramientas ágiles que puedan facilitar y agilizar el trabajo del equipo.

Game Owner

El Game Owner es quien actúa como un intermediario entre el Stakeholder (cliente) y el AGP Máster, indicándole las funcionalidades que debe incluir el videojuego, así como la visión central que se tiene del mismo, es decir, que es lo que se espera del videojuego. Para esto el Game Owner es quien define el Game Backlog y establece una priorización de dichas funcionalidades. Es imperativo que exista una comunicación activa con el AGP Máster, pues desde aquí se plantea e inicia como tal la creación del videojuego. Es importante tener en cuenta que un Game Owner debe tener un conocimiento elevado

sobre videojuegos de manera que pueda agregar valor al producto final desde sus etapas iniciales.

Funciones

- Comunicar con claridad la visión del Stakeholder al AGP Máster.
- Priorizar los requerimientos y mantenerlos actualizados
- Innovar los requerimientos a través de mejoras posibles que le permitan al videojuego adquirir originalidad y una identidad.
- Revisar constantemente los prototipos del videojuego para monitorear el avance de este y analizar posibles mejoras.
- Receptar retroalimentación de los jugadores a través del prototipo y analizar el nivel de aceptación de los jugadores con el videojuego.
- Realizar estimaciones de tiempo realistas para el cumplimiento de los requerimientos, tomando en cuenta las posibles mejoras.

Equipo de desarrollo

El equipo de desarrollo es el núcleo del proyecto y quienes se van a encargar de traducir los requerimientos y objetivos planteados por el Game Owner a un videojuego final que además de ser un software de calidad pueda posicionarse como un producto innovador en el mercado de los videojuegos. El equipo de desarrollo está compuesto por dos partes, una parte enfocada al apartado técnico del videojuego y otra parte enfocada al apartado artístico del videojuego. La estructura del equipo se compone de la siguiente manera.

Programadores

Son los encargados de codificar e implementar las funcionalidades requeridas para construir la lógica del videojuego.

Funciones

- Traducir los requerimientos establecidos y la visión central del videojuego al GDD.
- Realizar Pair programming esporádicamente para mejorar el código.

- Mantener los repositorios de código actualizados y correctamente estructurados.
- Realizar pruebas unitarias que permitan verificar las funcionalidades
- Realizar una integración continua a través de herramientas ágiles.

Diseñadores

Son los encargados de construir el entorno del videojuego.

Funciones

- Participar activamente en la creación del GDD.
- Mantener una constante comunicación con los programadores.
- Crear los escenarios y los personajes del videojuego.

Compositores

Encargados de producir las pistas musicales y los efectos de sonido del videojuego.

Guionistas

Encargados de escribir la historia que le da sentido a la aventura del videojuego.

Stakeholder

El stakeholder es la persona externa al equipo de desarrollo que se encuentra interesada en el éxito del videojuego y que apoya al equipo mediante distintas formas, generalmente financiando el videojuego, para esto, el stakeholder es quien comunica al Game Owner lo que desea conseguir con el videojuego.

Funciones

- Escuchar abiertamente la retroalimentación del Game Owner.
- Mantenerse actualizado sobre el mercado de los videojuegos y los intereses que muestran los jugadores cada día.
- Definir un alcance realista del videojuego.

Jugadores

Los jugadores son los encargados de probar el prototipo y realizar un análisis desde el aspecto jugable para encontrar posibles defectos. La interacción entre los jugadores y el videojuego como tal permite encontrar defectos que no son fácil de localizar.

Funciones

- Analizar el comportamiento de los elementos que intervienen en el videojuego a través de una rápida jugabilidad.
- Entregar retroalimentación sobre los defectos encontrados en el prototipo.

5.3.5 Etapas de desarrollo de AGP

El ciclo de vida de la metodología AGP consiste en las 3 etapas generales que existen en el desarrollo de videojuegos, sin embargo, AGP utiliza las ceremonias de Scrum con el objetivo de ejecutar un sprint en estas 3 etapas generales. La preproducción del videojuego consiste en la planificación del Sprint en donde intervienen el Game Owner y el AGP máster para definir como se llevará a cabo el sprint. La etapa de producción por otro lado ya consiste en la ejecución de todo lo planificado anteriormente, aquí es en donde intervienen el equipo de desarrollo en cumplir con los requerimientos planteados a través del desarrollo del sprint. La etapa de postproducción por otra parte ya consiste en la realización de pruebas a los prototipos desarrollados y la entrega de un release al stakeholder.

5.3.5.2 Preproducción

La etapa de preproducción corresponde a todo el aspecto de planificación y definición de las bases sobre las cuales estará sustentado todo el proyecto. De acuerdo con la figura 34, el proceso empieza por el stakeholder indicando los requerimientos que debe tener el videojuego, sin embargo, lo que distingue en este caso al videojuego de un software más tradicional es el hecho de entregar un videojuego que más halla de cumplir con una serie de requerimientos, sea capaz de entregar una experiencia innovadora al jugador, por esta razón el stakeholder debe mostrar un amplio conocimiento sobre el mercado de los videojuegos y saber expresar la visión que tiene entorno al videojuego. Una vez este proceso es completado y el Game owner logra recopilar toda esta información, se dan las reuniones con el equipo para empezar a planificar todo y construir los documentos que servirán de guía para el resto del desarrollo.

Reunión de planificación

En AGP, las reuniones son cruciales para el éxito del proyecto, por esta razón debe existir una correcta coordinación entre los miembros del equipo y definir reuniones en las que todos los miembros del equipo tengan disponibilidad para participar.

En esta reunión consiste en conocer todos los requerimientos y la visión general del videojuego que ha sido establecida por el stakeholder en conjunto con el game owner, mismo quien será el encargado de mostrar el Game backlog al equipo de desarrollo, a partir de este punto, comienza la repartición de las distintas tareas que se deben ejecutar para cumplir los requerimientos, con las respectivas fechas establecidas y las personas que ejecutaran las tareas. Se toman en cuenta los siguientes aspectos.

Tabla 18: Reunión de planificación de AGP

Participantes	Game Owner y Equipo de desarrollo
Objetivo	Distribuir las tareas y determinar los tiempos requeridos para completarlas
Tiempo de reunión	El tiempo recomendable es de 3 horas, sin embargo, depende de la organización del equipo. Es importante recalcar que las reuniones deben tratar asuntos puntuales. Aspectos adicionales pueden tratarse en las reuniones diarias.
Documentos generados	• Game Backlog • GDD (Game Design Document) • Diseño de la arquitectura.

Elaborado por: Ordóñez, 2022

GDD (Game Design Document)

El documento de diseño de un videojuego es fundamental para que el equipo logre comprender como debe estar construido el videojuego más allá del aspecto técnico, y es que mediante este documento se pueden definir todos los aspectos esenciales del

videojuego. El documento como tal no posee un formato específico para su creación sin embargo si debe incluir todo lo que se detalla a continuación.

Tabla 19: Aspectos del GDD

Concepto del Videojuego	Explicación general de lo que trata el videojuego
Rol del jugador	El papel que tiene el jugador dentro del videojuego
Gameplay	Forma particular de jugar al videojuego, puede entenderse de acuerdo con 3 puntos esenciales. **Perspectiva:** como el jugador percibe y comprende el entorno del videojuego. **Modelo de interacción:** influencia del jugador en el entorno del videojuego, generalmente a través de un avatar. **Modalidad:** Un jugador o varios jugadores de manera simultánea.
Género	Temática del videojuego
Plataforma	Equipo en donde se va a ejecutar el videojuego.
Audiencia	Público objetivo del videojuego, aquí se determina la clasificación del control parental que tendrá el videojuego.
Entorno/Mundo del videojuego	Descripción del entorno del videojuego.
Flujo del videojuego	Manera en que el jugador avanza en el videojuego.

Elaborado por: Ordóñez, 2022

De acuerdo con los aspectos revisados anteriormente, se procede a establecer el documento de diseño de la metodología AGP.

Propuesta del videojuego

La propuesta de un videojuego es el punto de inicio en el cual se establece de manera general una breve descripción de los detalles más relevantes que tendrá el videojuego. Este apartado es fundamental pues aquí se expresa la visión general del videojuego. Los requerimientos planteados en el Game Backlog pueden ser vistos como objetivos que se deben ir completando paulatinamente para que el videojuego posea una baja cantidad de defectos y su comportamiento sea normal, sin embargo, en la propuesta del videojuego se establece la visión central del videojuego, y esta visión es la razón de ser del videojuego, ya que aquí se determina cual es la experiencia que se busca entregar al jugador. Cabe mencionar que este documento debe estar escrito en lenguaje natural de manera que todos los miembros sean capaces de comprenderlo.

Descripción: Este apartado debe ser conciso, sin entrar en detalles muy específicos y plasmando la idea central del videojuego, se deben responder las siguientes preguntas.

- ¿Qué distingue este videojuego de los demás?
- ¿Qué rol deseo que el jugador desempeñe en el videojuego?
- ¿Qué experiencia deseo crear en la comunidad de jugadores a través de este juego?
- ¿Cómo será la interacción del jugador con el entorno del videojuego?

Alcance: Es de suma importancia que se defina el alcance que tendrá el videojuego y sus respectivas limitaciones. El equipo debe procurar construir un videojuego de acuerdo con sus capacidades, de lo contrario, podrían incurrir en una planeación poco realista del videojuego y terminar por no cumplir con todo lo que se estableció.

Motivación: En este apartado del equipo debe establecer la razón por la cual desean construir el videojuego y que esperan conseguir de él.

Objetivo: Los requerimientos funcionales y no funcionales, sirven como objetivos incrementales que se irán cumpliendo paulatinamente, sin embargo, aquí se plantea el objetivo central del videojuego que vendría a ser la misión que tiene el equipo.

Características: Descripción de los aspectos clave que le otorgan originalidad al videojuego, este apartado es fundamental para determinar la innovación que presentará el videojuego al público.

Concepto del videojuego

Este apartado es más específico y una vez se han definido todos los aspectos pertenecientes a la "descripción del videojuego", se debe empezar por establecer el concepto central.

Tabla 20: Formato para el concepto del videojuego como parte del GDD

	Concepto del videojuego					
	Trama del videojuego: De que trata el videojuego y que rol desempeña el jugador dentro del mismo.					
	Mecánicas					
	Controles	Reglas	Patrón	Comportamiento		
	Combinación de botones para ejecutar la acción	Condiciones que permiten o evitan la ejecución de determinadas mecánicas	Serie de variables que intervienen en el entorno durante la ejecución de las mecánicas.	Respuesta del entorno y los NPC´S, a las mecánicas ejecutadas		
1						
2						
…						
	Personajes					
	Nombre	Categoría	Rol	Diseño	Acciones	
	ID del personaje	Personaje principal/NPC aliado/NPC enemigo	Papel que desempeña en el entorno del videojuego	Bosquejo del personaje	Mecánicas que desempeña el personaje	
1						
2						
…						
	Niveles/Mapas					
	Nombre	Descripción	Objetos	NPC´S	Diseño	Música
	ID del nivel o mapa	Breve presentación del entorno perteneciente al nivel	Items y objetos característicos del lugar	Personajes que se encuentran en el nivel	Bosquejo general del escenario	Descripción de la musicalización del nivel
1						
2						
…						

Elaborado por: Ordóñez, 2022

Línea de tiempo

Este apartado debe comprenderse como el flujo del videojuego y como el jugador va marcando ese avance de manera progresiva.

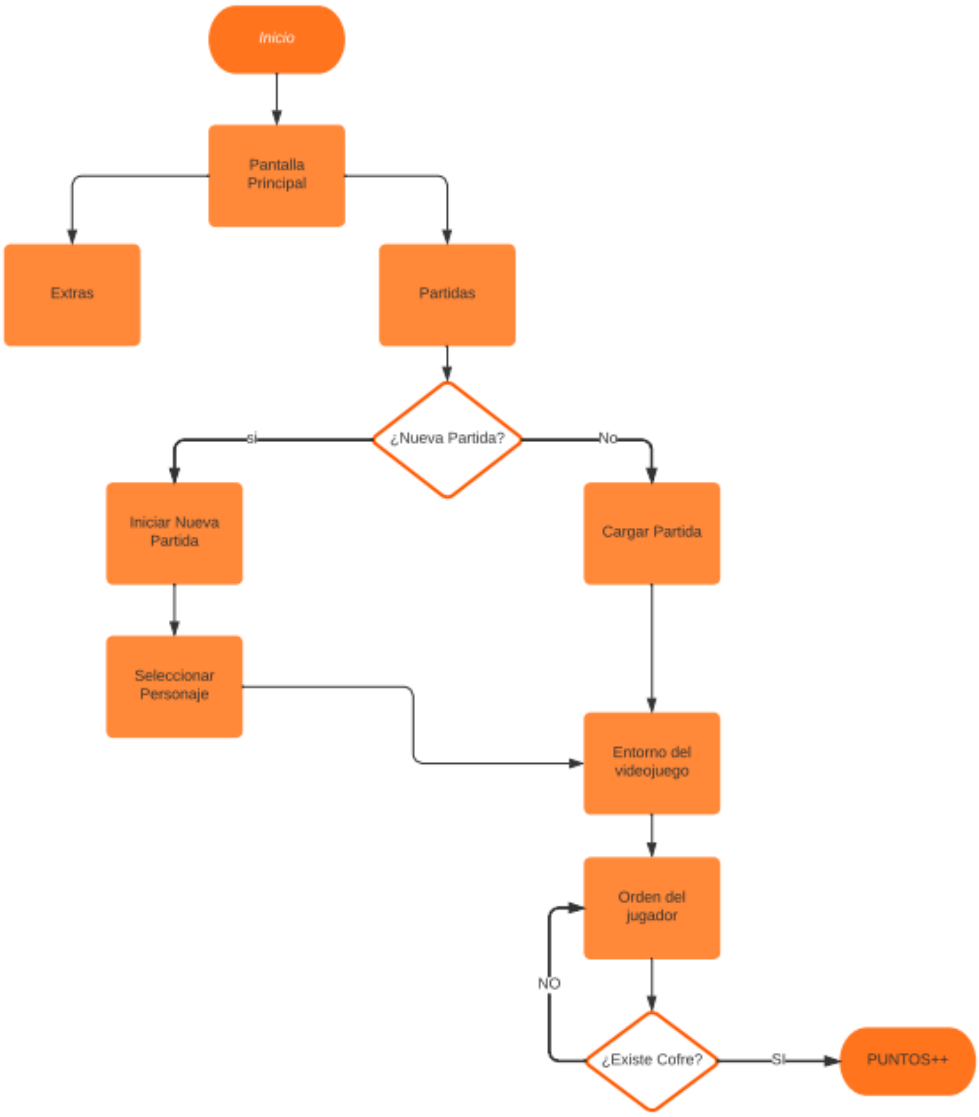


Figura 12: Ejemplo de Flujo de un videojuego.

Elaborado por: Ordóñez, 2022

El flujo de un videojuego varía dependiendo de la complejidad y extensión de este. El ejemplo de la figura 12, es un patrón común en la mayoría de los videojuegos. El equipo de desarrollo debe construir un diagrama de acuerdo con sus necesidades.

Interfaces

Las interfaces funcionan como el front-end del videojuego, y es que mediante estas interfaces se organizan todas las opciones que un jugador puede controlar más allá de las mecánicas propias de cada personaje, dichas opciones controlan una serie de aspectos esenciales que dependerán de las preferencias de cada jugador, estas opciones van desde controlar aspectos técnicos como el brillo o el contraste del videojuego, hasta opciones estéticas o jugables como aumentar o disminuir la dificultad de un videojuego, entre otras varias opciones que se plantean de acuerdo a la complejidad del videojuego.

Las pantallas deben ser intuitivas y seguir un orden que no confunda al jugador. Las opciones deben estar distribuidas de acuerdo con el siguiente orden de pantallas.

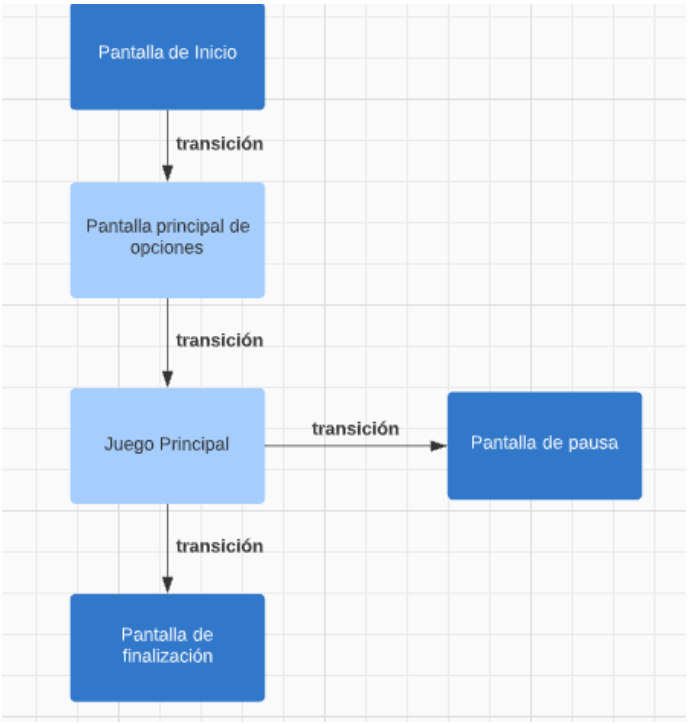


Figura 13: Orden de pantallas en un videojuego.

Elaborado por: Ordóñez, 2022

Pantalla de Inicio

En la pantalla de inicio debe incluirse la portada del videojuego, esta portada es la que causa la primera impresión del jugador y lo que inicia la inmersión del jugador.



Figura 14: Pantalla de inicio del videojuego Call of duty Warzone.

Pantalla principal de opciones

En esta pantalla se coloca el menú principal de opciones, el jugador debe poder configurar aspectos estéticos del videojuego antes de proceder a jugar, en este apartado se controlan los siguientes aspectos.

- Configuraciones visuales (Brillo, contraste, luminosidad, etc.)
- Configuraciones de audio (Control de idioma, aumento de sonido ambiental, etc.)
- Configuraciones de jugabilidad (Controles, dificultad, gestión de partidas, etc.)



Figura 15: Pantalla de menú principal del videojuego Bioshock Infinite.

Pantalla de pausa

El jugador puede requerir realizar cambios durante la partida, por esta razón se debe implementar una pantalla de opciones que pueda desplegarse en cualquier momento según el jugador lo desee, cabe destacar que esta pantalla de opciones siempre es accesible aun si el videojuego es en modalidad on-line, la diferencia radica en que, al ser un videojuego de un jugador, el entorno se detiene, mientras que en un videojuego on-line, el menú se despliega sin que el entorno se detenga. Esto depende netamente de las decisiones de diseño que el equipo tome.



Figura 16: Pantalla de menú de pausa del videojuego Minecraft.

Generalmente este menú maneja las mismas opciones que las del menú principal, pero también adiciona nuevas opciones.

Pantalla de finalización

Esta pantalla simplemente sirve para mostrar al jugador que ha fallado o ha sido vencido por la IA del juego o por otro jugador. Generalmente maneja dos opciones, una para repetir la misión y volver al juego, u otra para abandonar el videojuego.



Figura 17: Pantalla de finalización del videojuego Metroid Dread.

Aclaraciones finales del GDD

El documento de diseño del videojuego no es un documento estático ya que el videojuego puede enfrentar cambios constantes en cada uno de los elementos, por esta razón el equipo es importante que ambas partes del proyecto participen en la creación de este documento, de forma que se pueda llegar a un consenso siempre que el GDD deba ser alterado.

Game Backlog

El Game backlog es definido por el Game Owner en conjunto con el AGP Máster, este documento debe incluir todas las historias de usuario que logren plasmar claramente los requerimientos que planteó el stakeholder. El documento puede estar sujeto a cambios constantes, sin embargo, lo ideal es que el stakeholder logre comunicar de forma definitiva lo que desea del videojuego, de manera que no se presente una elevada cantidad de cambios que terminen por retrasar los tiempos de entrega, por esta razón, es imperativo que el stakeholder y el Game Owner tengan conocimientos sobre el mercado de videojuegos actualmente.

El documento debe ser construido de acuerdo con el siguiente orden y siguiendo los parámetros definidos a continuación.

Se debe definir un objetivo primordial, que indica que es lo que se espera conseguir del videojuego.

- Los requerimientos deben ser claros y concisos, no caer en ambigüedades.
- Cada requerimiento debe estar argumentado en una corta frase.
- Los requerimientos deben estar escritos en lenguaje natural.

- Los requerimientos deben ser comprobables, es decir se debe poder monitorear su avance en cada incremente del producto.
- Cada requerimiento debe tener un alcance realista.
- Se deben realizar estimaciones del tiempo que llevará cumplir cada requerimiento.

Borrador del Game Backlog

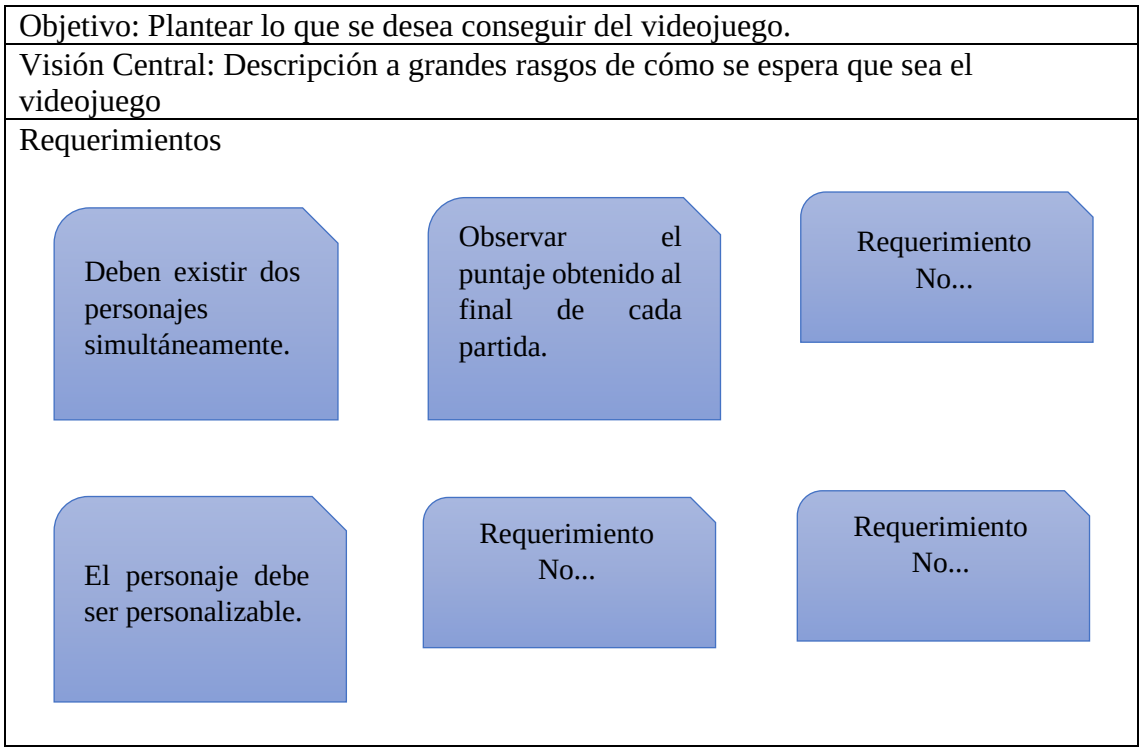


Figura 18: Borrador inicial de requerimientos del Game Backlog,

Elaborado por: Ordóñez, 2022

Clasificación de requerimientos

Un software posee tanto requerimientos funcionales como no funcionales, esto sucede de igual manera para un videojuego, por tanto, es importante analizar el borrador planteado anteriormente, determinando así el tipo de requerimientos que se desea del videojuego.

Requerimientos funcionales

Requerimientos enfocados a la interacción del jugador con el videojuego, es decir todas las acciones que ejecuta un videojuego en respuesta a las ordenes enviadas por el jugador.

Ejemplo: “El avatar del jugador debe ser personalizable”

Requerimientos no funcionales

Requerimientos enfocados a características propias del videojuego, aquí se consideran los siguientes aspectos:

- Rendimiento
- Usabilidad
- Seguridad
- Normas Legales

Ejemplo: “El videojuego debe correr en todo a momento a 60 fps (fotogramas por segundo)”

Inicialmente se parte indicando los requerimientos que se desea, no necesitan tener un nivel de detalle muy alto, ya que servirán para que el Game Owner pueda comprender de mejor manera cual es la visión del videojuego y las necesidades que tiene el Stakeholder. Posteriormente se debe traducir este borrador a historias de usuario que ya estén bien definidas, para esto es importante considerar la siguiente estructura.

Traducción del borrador

En el borrador se plantean las necesidades del Stakeholder, sin embargo, estas necesidades son planteadas de una manera muy informal. El paso siguiente para poder definir los requerimientos correctamente, consiste en detallar y especificar cada una de estas necesidades en historias de usuario que contemplen la siguiente estructura.

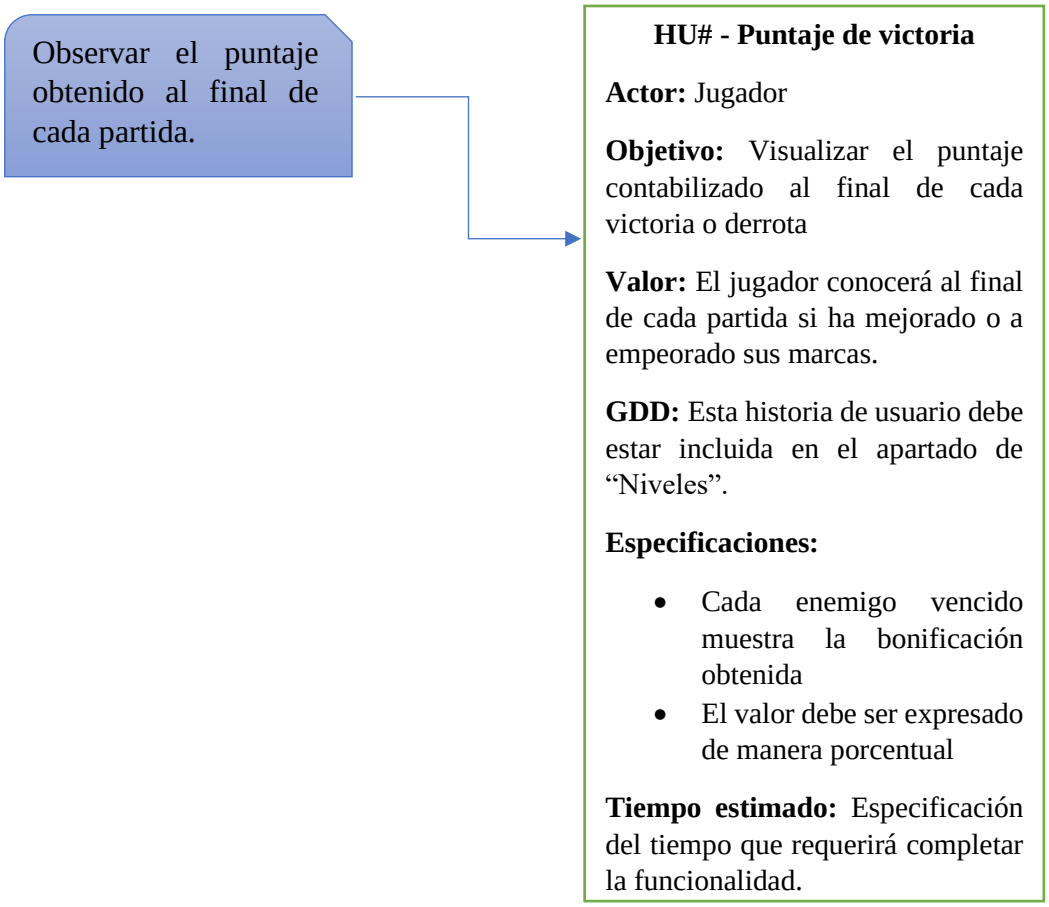


Figura 19: Historia de usuario de AGP.

Elaborado por: Ordóñez, 2022

Priorización de las historias de usuario

Una vez se han definido y especificado cada una de las historias de usuario con sus respectivas estimaciones de tiempo, lo que sigue es ordenar estas historias de usuario en base a la prioridad que se le asigne a cada una.

Para priorizar cada historia de usuario, se debe considerar que tan factible y que tan imprescindible es la funcionalidad para el videojuego, de esa manera se puede realizar un

orden desde una alta prioridad hasta una baja prioridad. Se propone la siguiente escala acompañada de un esquema para determinar la prioridad del requerimiento.

- 1: bajo
- 2: medio
- 3: alto

Tabla 21: Formato para determinar la prioridad de una historia de usuario.

Deseable	Viable	
3	3	
2	3	Prioridad Alta
3	2	
2	2	Prioridad Media
3	1	
2	1	
1	1	Prioridad Baja

Elaborado por: Ordóñez, 2022

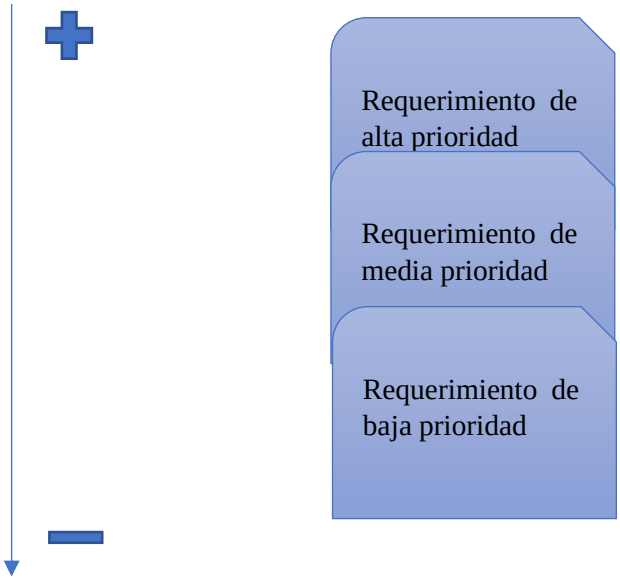


Figura 20: Priorización de requerimientos de AGP.

Elaborado por: Ordóñez, 2022

Arquitectura del videojuego

El Game Backlog y el GDD son documentos que le ayudan al equipo a definir los objetivos que deben cumplir y cuál es la visión central del videojuego, sin embargo, un videojuego al ser un sistema complejo requiere también de una descripción detallada de la arquitectura con la que se piensa crearlo.

Limitaciones Arquitectónicas

Antes de proceder a definir la arquitectura de un videojuego, siempre se debe determinar una serie de restricciones que permitan definir las limitaciones que tendrá el videojuego, de esta manera se puede diseñar una arquitectura realista.

Componentes de un videojuego

Un videojuego posee múltiples entidades que se comunican entre sí, manteniendo una leve dependencia entre ellas. Estas entidades funcionan como módulos que cumplen una función perteneciente a un apartado del videojuego. Las más comunes son las siguientes:

- Música
- Jugabilidad
- Cinemáticas
- Interfaz
- Gestor Principal
- Inteligencia artificial
- Datos

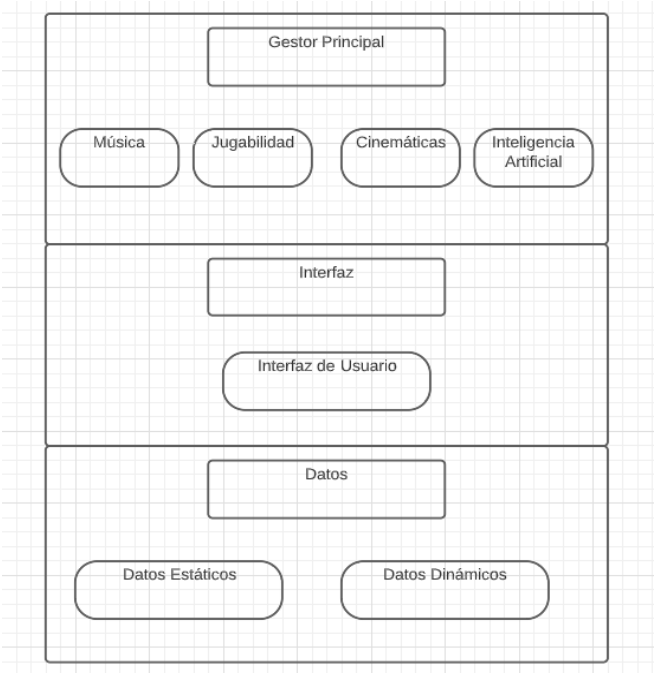


Figura 21: Arquitectura basada en componentes de un videojuego.

Elaborado por: Ordóñez, 2022

Este es un modelo arquitectónico que engloba los componentes más comunes y esenciales de los videojuegos, en cada parte de la estructura están organizados los componentes de acuerdo con 3 capas, la primera capa principal, la segunda capa de interfaz y la tercera capa de datos. La siguiente figura establece la lógica de la arquitectura planteada.

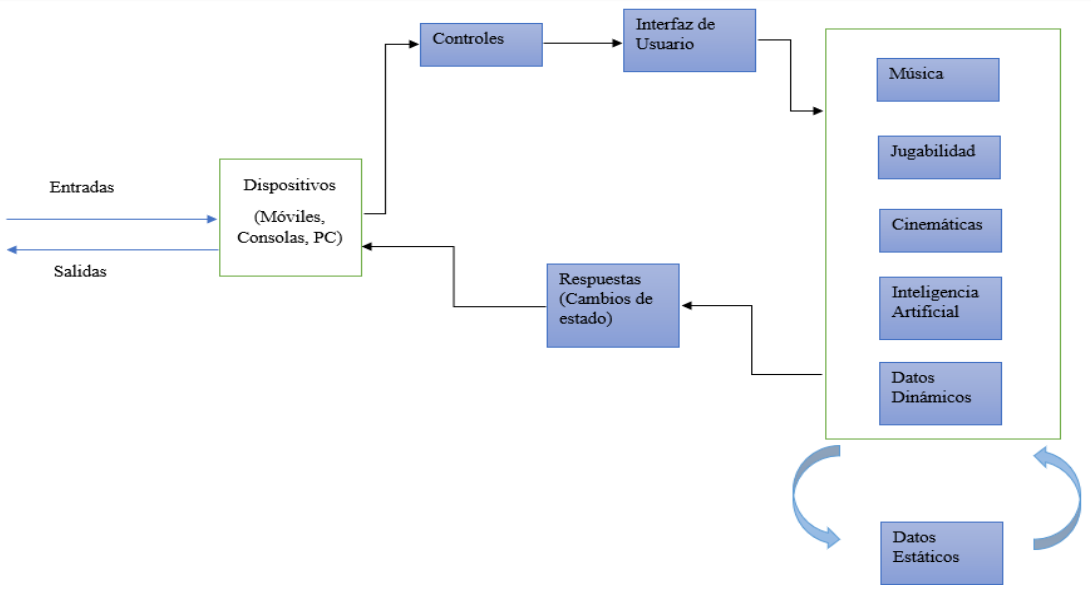


Figura 22: Lógica de la arquitectura de un videojuego.

Elaborado por: Ordóñez, 2022

5.3.5.2 Producción

La etapa de producción consiste en el desarrollo del videojuego a partir de los documentos que se plantearon en la etapa anterior, durante esta etapa se van a ejecutar todos los sprints necesarios para ir generando incrementos del videojuego.

Planeación del Sprint

El Game Backlog define todos los requerimientos que habrán de cumplirse durante el proceso de desarrollo, sin embargo, estos requerimientos se van cumpliendo de manera incremental a medida que se ejecutan varios sprints, por esta razón es necesario realizar una planificación previa a cada sprint que el equipo vaya a desempeñar, de esta forma se puede realizar una correcta distribución de las tareas.

Sprint Backlog

El Sprint Backlog le permite al equipo distribuir las tareas que se llevarán a cabo en una determinada iteración para cumplir con el objetivo del sprint. El Sprint Backlog que propone la metodología Scrum se funciona correctamente para el desarrollo de un videojuego. Se detalla su estructura a continuación.

Tabla 22: Sprint Backlog

Requerimientos (Identificador del requerimiento)	Tareas	Encargado	Tiempo estimado	Área Correspondiente
(Ordenados de acuerdo con la prioridad establecida en el Game Backlog)	(Tareas que deben llevarse a cabo para cumplir el requerimiento)	(Persona que debe desempeñar una determinada tarea)	(Cantidad de tiempo que se requiere para completar la tarea)	(Programación/Arte/ Ambas)
HU01	Renderizar el escenario B	Xavier	3 días	Ambas
HU01				
HU01				
HU02				
HU02				
HU05				
HU05				
HU05				

Elaborado por: Ordóñez, 2022

Dentro del sprint backlog se incluyen todas las tareas necesarias para cumplir con el requerimiento, lo que implica tanto tareas de programación como tareas de arte, según el requerimiento así lo requiera. Es decir, existe un solo sprint backlog para todo el equipo.

Desarrollo del Sprint

La ejecución de cada una de las tareas que se definieron en el sprint backlog se procede a la ejecución de las tareas que se definieron en el sprint backlog. Monitorear el progreso de las tareas determina la agilidad con la que el equipo está trabajando, por esta razón, es importante llevar además de un sprint backlog, un tablero que permite ir monitoreando el avance de cada tarea y el tiempo que está llevando realmente ejecutar esa tarea en relación con el tiempo que se había estimado. El método más ágil para monitorear las tareas es través de un tablero mediante el cual los miembros tengan acceso al mismo todo el tiempo y puedan mover las notas de forma dinámica según la tarea se vaya completando.

Tabla 23: Tablero de tareas

Requerimiento	Tareas	Pendiente	En desarrollo	Completado
HU01	T01 T04	T04		T01
HU02	T02		T02	
HU03	T03 T06		T06 T03	

Elaborado por: Ordóñez, 2022

Flujo del Sprint

Cada sprint que se ejecute durante el proceso de desarrollo seguirá el siguiente orden.

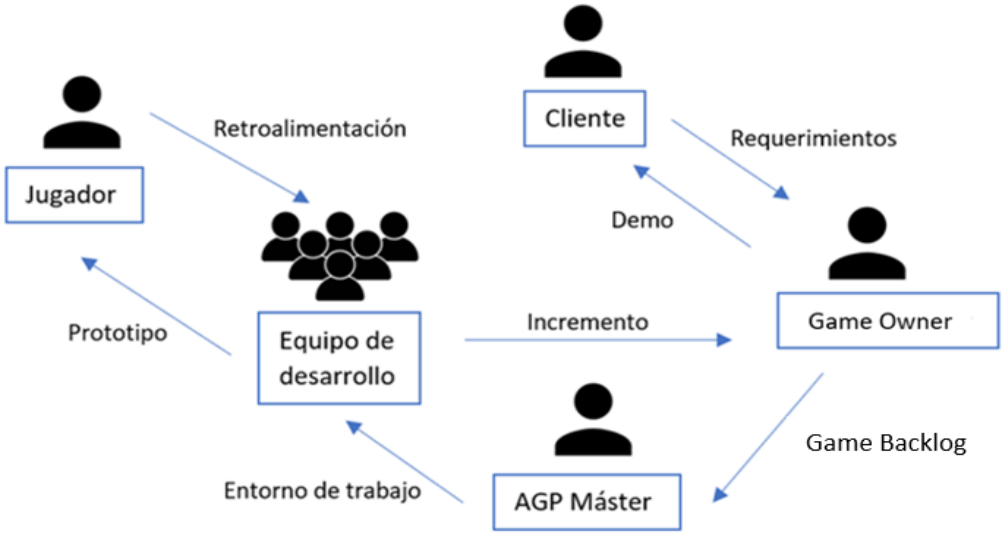


Figura 23: Flujo de un Sprint en AGP.

Elaborado por: Ordóñez, 2022

Burn-Down Chart

Para que el equipo alcance un mayor grado de exactitud en la estimación de tiempos, el burn-down chart permite establecer una comparativa exacta entre el tiempo estimado con el tiempo que ha llevado completar un sprint.

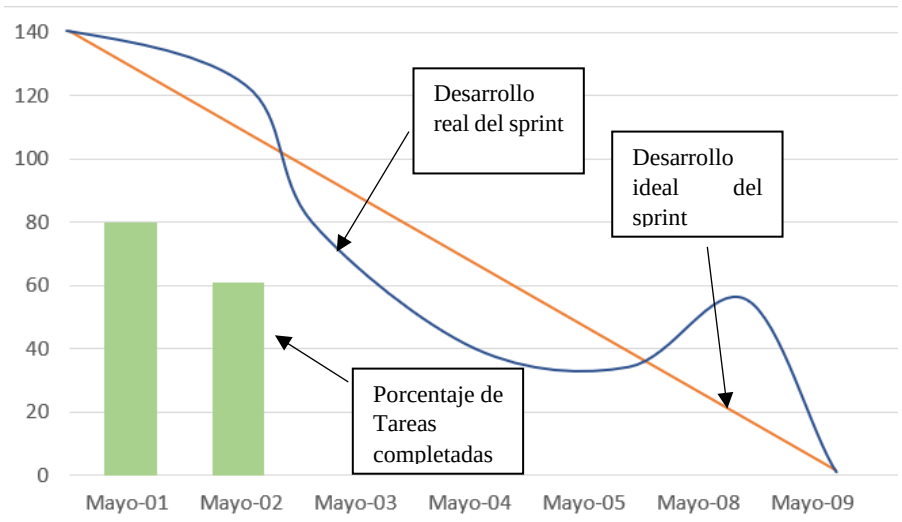


Figura 24: Burn-Down Chart

Elaborado por: Ordóñez, 2022

En el Burn-down chart, se colocan en el eje X, los días que requerirá completar una tarea de acuerdo con las estimaciones anteriores, mientras que en el eje Y se colocan la cantidad de horas estimadas para completar las tareas que conforman el sprint.

Tareas interdisciplinarias

El equipo de desarrollo desempeña varias tareas con el objetivo de cumplir con cada requerimiento, de acuerdo con la tabla 57, el sprint backlog posee un apartado que hace referencia al "Área" de la tarea, esto quiere decir que si bien es cierto un requerimiento puede enfocarse netamente a la parte técnica o a la parte artística de un videojuego, existen también varios requerimientos que dependen de ambas partes para conseguirlos.

La figura 30, muestra que mayoritariamente, miembros del equipo participaron también en tareas que no correspondían específicamente a su área, esta práctica es ventajosa para los equipos, siempre y cuando mantengan un orden en las tareas que deben desempeñar. Para todos estos requerimientos en los cuales deben participar miembros de ambas partes, se debe seguir el siguiente procedimiento.

Un estudio propuesto por Llansó (2014), establece que una forma eficaz en la que se ha logrado un entendimiento entre programadores y diseñadores es través del uso de ontologías. En los videojuegos, es común la utilización de entidades para expresar los distintos elementos de un videojuego, estas entidades pueden ser avatares, npc's, o algún otro elemento del entorno que responda a las acciones de un jugador.

El problema más recurrente sucede cuando programadores y diseñadores trabajan en una misma funcionalidad sin antes haber llegado a un consenso, en este punto, se establecen los siguientes parámetros a tomar en cuenta, para poder definir como funcionarán las entidades en el videojuego, según el siguiente modelo sugerido por Llansó (2014).

- Los diseñadores definen las acciones de cada entidad
- Las entidades y las relaciones entre ellas deben estar descritas en un diseño de alto nivel.
- Las entidades deben tener una descripción detallada de sus atributos.
- Se debe indicar a que funcionalidad pertenece cada una de las entidades.

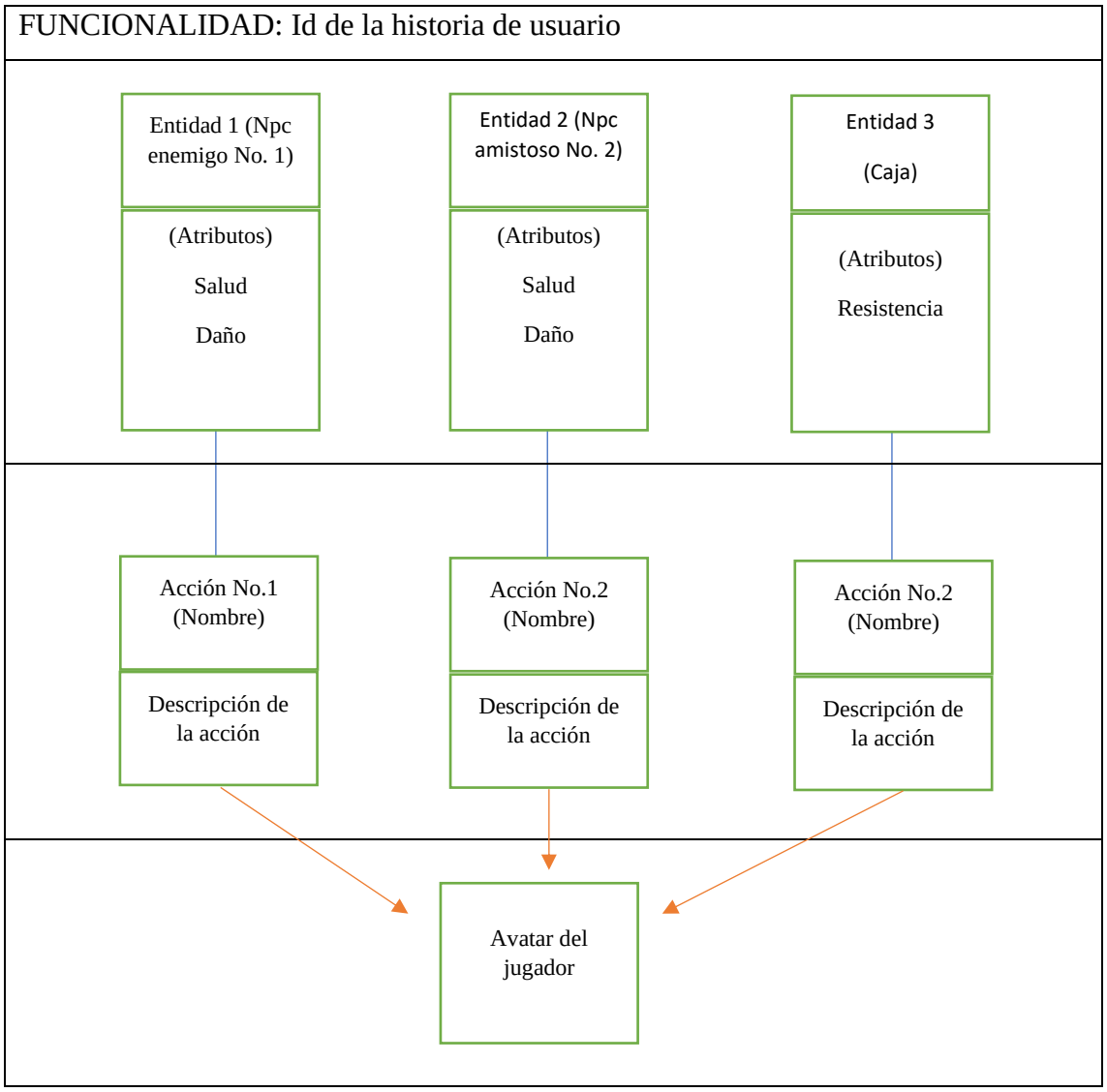


Figura 25: Ejemplo de ontología definida por los diseñadores del videojuego.

Elaborado por: Ordóñez, 2022

La figura 25, es el modelo planteado por los diseñadores, en este modelo están descritos a grandes rasgos las entidades que va a tener un determinado nivel, mediante esta representación ontológica, los programadores pueden traducir este modelo y convertirlo en diagramas un diagrama de clases que puede ser comprendido por todos los programadores.

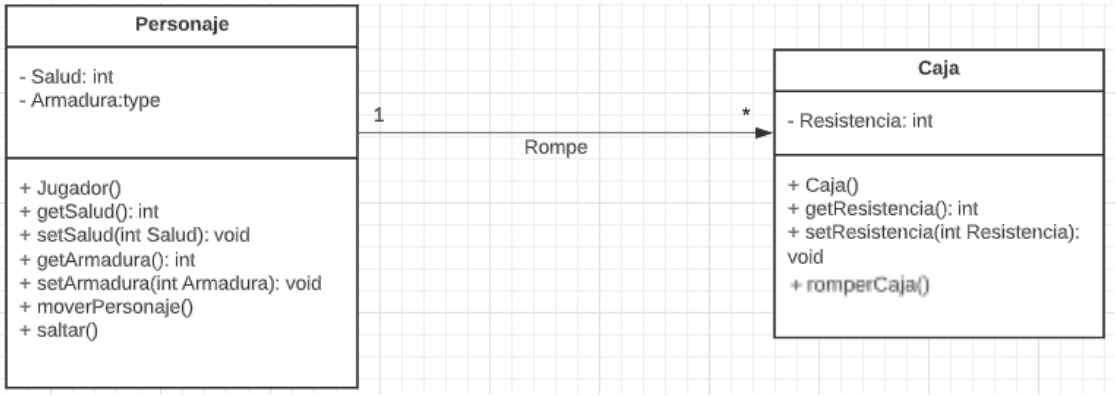


Figura 26: Ejemplo de diagrama de clases del videojuego del género shooter.

Elaborado por: Ordóñez, 2022

La figura 26, muestra la traducción del modelo ontológico planteado por los diseñadores a un diagrama de clases sencillo que le permite a los programadores estructurar correctamente las entidades al código fuente.

AGP establece el siguiente procedimiento para cumplir con requerimientos que dependan de ambas partes del equipo.

1. Los diseñadores deben presentar un modelo ontológico de las entidades que tendrá el videojuego

2. Los programadores deben reunirse con los diseñadores y verificar el modelo presentado.

3. Los programadores toman el modelo planteado y los traducen mediante lenguaje UML.

4. Los diagramas planteados deben ser compartidos con el resto del equipo enfocado al apartado técnico.

5. Una vez los diagramas han sido planteados, los diseñadores empiezan a construir las entidades, este proceso de construcción se resume en las siguientes actividades.

- Diseño de personajes
- Texturización
- Renderización

6. Los programadores vuelven a reunirse con los diseñadores para la tarea de animación.

Estructura del código

Una vez los programadores inician la codificación correspondiente a cada una de las funciones, deben tratar de mantener una jerarquía de entidades a nivel de código. Por ejemplo, si se trata de un grupo de Npc´s la jerarquía debería seguir un orden similar a la siguiente figura.

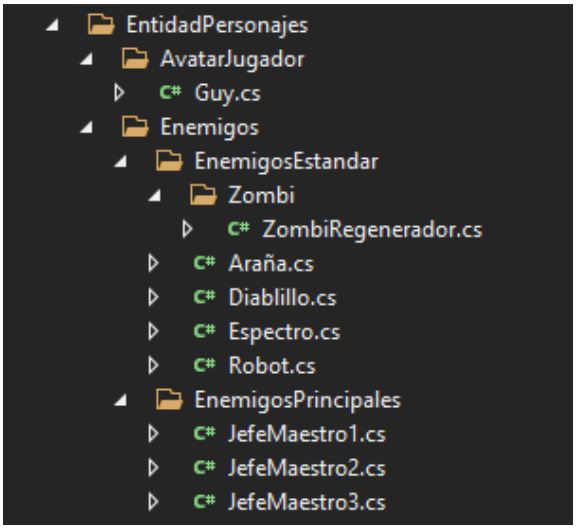


Figura 27: Ejemplo de diagrama de Jerarquía de Entidades de un videojuego

Elaborado por: Ordóñez, 2022

Código Colectivo

La metodología promueve una participación dinámica de todos los miembros del equipo, esto quiere decir que, si un miembro del equipo o varios miembros del equipo se encuentran teniendo alguna dificultad para completar una tarea, los otros miembros deben destinar un porcentaje de su tiempo para ayudar a solventar la dificultad que impide el cumplimiento de dicha tarea, esto también se aplica, las los cambios que puedan surgir en el proyecto, para agilizar la adaptación del videojuego a un nuevo cambio que pueda surgir, es necesario que todo el equipo, tenga acceso al código fuente de todo el videojuego, para esto se deben considerar los siguientes puntos esenciales.

- Desistir de almacenar los códigos únicamente de manera local, utilizar repositorios en línea.
- Crear una rama para cada miembro del equipo.
- El AGP Máster es quien debe controlar la rama principal y actualizarla posterior a cada cambio realizado.

- Realizar comentarios en todo momento de los cambios que se hayan realizado, sobre todo cada vez que se efectué un ‘commit’.

Pair Programing

De acuerdo con la complejidad de la tarea, dos programadores pueden dedicarse a completarla y recurrir a sesiones de pair programming, sin embargo, AGP, establece que estas sesiones deben darse de preferencia si una tarea se torna compleja para un solo programador, es necesario que otro programador programe sesiones de pair programming para ayudar a completar la tarea.

Pruebas

La agilidad con la que se realizan las pruebas es vital para el desarrollo del prototipo, y es que, al finalizar la programación de las entidades, el equipo debe comprobar su comportamiento en el entorno que han planteado los diseñadores. Para conseguir que el equipo pueda realizar múltiples pruebas y añadirle robustez al videojuego, la metodología plantea una serie de pruebas unitarias automatizadas.

La automatización de las pruebas ayuda a que el equipo pueda economizar el tiempo destinado a las pruebas. Los motores gráficos o los IDE’S actuales poseen librerías que permiten automatizar pruebas, tal es el caso de visual studio y su framework NUnit que permite escribir nuestras propias pruebas para poder automatizarlas y aplicarlas cuando sea requerido, de igual manera otros motores gráficos como Unreal permiten realizar una automatización de pruebas.

```
{   public string Rango (int EnemigosAbatidos, int Itemsncontrados , int sum )
    {
        sum = 0;
        String rank;
        sum += EnemigosAbatidos * 10;
        sum += Itemsncontrados * 20;

        if (sum > 1000)
        {
            rank = "A";
        }
        else if (sum > 2000)
        {
            rank = "B";
        }
        else
            rank = "C";

        return rank;
    }
}
```

Figura 28: Función de asignación de rango en un videojuego

Elaborado por: Ordóñez, 2022

La figura 28, muestra una función que determina el rango que un jugador obtiene al final de cada nivel, esta funcionalidad resulta típica en muchos videojuegos que determinan un rango de un jugador de acuerdo con un puntaje obtenido. Así como esta función existen varias otras funciones que el equipo de desarrollo puede plantear y que al ser funciones que desencadenan acciones importantes, siempre deben someterse a pruebas constantes.

```
public class DoomGuy2
    {
        [test]
            public void TestRango()
        {
            //INPUTS
            var NuevoRango = new MedidorRango();
            var EnemigosAbatidos= 35;
            var ItemsEncontrados = 4;

            var ResultadoEsperado = ((EnemigosAbatidos * 10) + (ItemsEncontrados
* 20));

            var RangoFinal = NuevoRango.Rango(EnemigosAbatidos,
ItemsEncontrados);
            Assert.That(NuevoRango,IsEqualTo(ResultadoEsperado));
        }
    }
```

Figura 29: Prueba automatizada

Elaborado por: Ordóñez, 2022

De acuerdo con la figura 29, en la primera sección de la prueba se deben ordenar todos los parámetros de entrada. En la segunda sección se invoca a la función que se desea evaluar, mientras que en la sección final o en la "salida" se realiza una comparativa entre el resultado obtenido de la función, con el resultado deseado, este resultado representa el comportamiento esperado de la entidad.

Integración continua

Las pruebas que realiza el equipo durante esta etapa son vitales para evitar defectos que lleguen hasta el prototipo final del videojuego. Para complementar esta buena práctica y agilizar las pruebas que el equipo realicen, AGP alienta al equipo a aplicar una integración continua de cada uno de los módulos que se vayan probando, de esta manera, el proceso de pruebas se agiliza notablemente.

La metodología establece que el equipo de desarrollo recurra a la integración continua para optimizar el tiempo que requiere realizar pruebas e integrar todo el código que los miembros del equipo vayan generando, ya que realmente varios procesos de actualizaciones o inserciones de código a una rama del repositorio resultan ser procesos automáticos que puede ser automatizados. Existen múltiples softwares que permiten realizar una integración continúa dependiendo del repositorio de código que el equipo haya seleccionado. El equipo de desarrollo debe seguir el siguiente proceso para lograr una integración continua.

1.Aplicar un software para integración continua.

2.Realizar la automatización de pruebas correspondiente a la funcionalidad.

3.Realizar un commit cada vez que se hayan producido cambios en el código fuente de un módulo.

4.Diariamente se debe realizar un merge en el repositorio.

5.Asegurarse de estar trabajando en la versión más actualizada del videojuego.

AGP hace énfasis en el último punto del proceso, pues a través de un software de integración continua, es posible automatizar tanto las pruebas como el proceso de integración, cada vez que un miembro del equipo realice un commit.

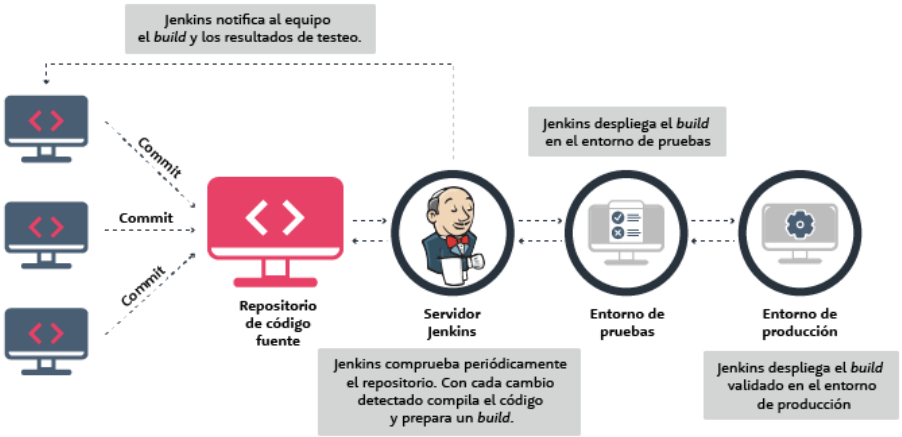


Figura 30: Proceso de integración continua de Jenkins

Fuente: (Sentrio, 2021)

Revisión del Sprint

Una vez el equipo ha cumplido con las tareas que fueron asignadas en el plan del sprint, incluyendo las respectivas pruebas de la funcionalidad en cuestión, el equipo completo debe realizar una reunión para revisar el incremento generado, de esta forma se podrá determinar si la funcionalidad cumple o no con las expectativas, de ser un caso de éxito, el requerimiento adquiere un estado de "cumplido" en el Game Backlog, caso contrario el requerimiento es reestructurado para volver a trabajar en él, sin embargo el equipo de desarrollo debe tomar en cuenta que si el requerimiento adquiere una complejidad que frena al equipo considerablemente para poder cumplir con los otros requerimientos, lo recomendable sería eliminar dicho requerimiento.

Estructura de la reunión

El objetivo central de la revisión es corroborar el funcionamiento del incremento presentado, para esto el equipo debe tomar en cuenta la siguiente estructura del sprint.

Tabla 24: Estructura No. 1 del Sprint Review,

Sprint #
Estatus del Videojuego
Descripción del avance del videojuego: Se da una retroalimentación de la funcionalidad y la importancia para el videojuego en general
Aspectos mejorables: Características que puedan mejorarse.
Sprints requeridos para completar el videojuego: % de avance del videojuego.
Objetivo del Sprint Descripción: El equipo determina si el objetivo se cumplió satisfactoriamente
Contratiempos El equipo describe los inconvenientes que se presentaron durante el sprint.

Elaborado por: Ordóñez, 2022

Tabla 25: Estructura No. 2 del Sprint Review,

Demo del Videojuego
Demostración de la funcionalidad
Posibles defectos: El equipo también debe evaluar el comportamiento del incremento, incluso antes de que lo pruebe el Stakeholder.
Ligeras Mejoras: Listado de mejoras que podrían implementarse.
Actualizaciones rápidas: En caso de haber establecido mejoras el equipo, tiene que destinar un tiempo a implementarlas (las mejoras deben ser ligeras)
Puntos para considerar en el siguiente Sprint: Esta no es una planeación del siguiente sprint, el equipo debe anotar aspectos importantes a considerar para el siguiente sprint.

Elaborado por: Ordóñez, 2022

5.3.5.3 Post Producción

La etapa final del videojuego ocurre cuando todos los requerimientos planteados en el Game Backlog fueron desarrollados e integrados correctamente. El equipo de desarrollo tiene que presentar un prototipo jugable que vaya a ser probado tanto por el jugador como por el Stakeholder. Este prototipo debe ser sometido a pruebas de sistema para comprobar su funcionamiento global. El prototipo debe ser evaluado por el jugador o "tester de videojuegos" que se encargará de jugar al videojuego y entregar la retroalimentación respectiva.

Pruebas de Sistema

Este tipo de pruebas sirven para comprobar el comportamiento del videojuego a nivel global, la idea de estas pruebas no es solo encontrar posibles defectos que alteren la experiencia de juego, sino también poder idear y añadir posibles mejoras que le beneficien al videojuego.

Aspectos para evaluar

Jugabilidad: El jugador debe probar el videojuego y comprobar que las mecánicas son correctas, es decir que los movimientos y acciones de cada uno de los elementos del videojuego funcionan y responden correctamente a todas las ordenes enviadas por el jugador.

Interfaz: El jugador debe comprobar que el entorno y los personajes mantienen estabilidad, es decir que la estructura tanto de los personajes como del entorno y todo lo que lo compone, debe estar correctamente animado y debe mantenerse en visión del jugador.

Música: El jugador comprueba los efectos de sonido del videojuego, revisando que estos no estén desfasados o que se interrumpan súbitamente.

Flujo de revisión

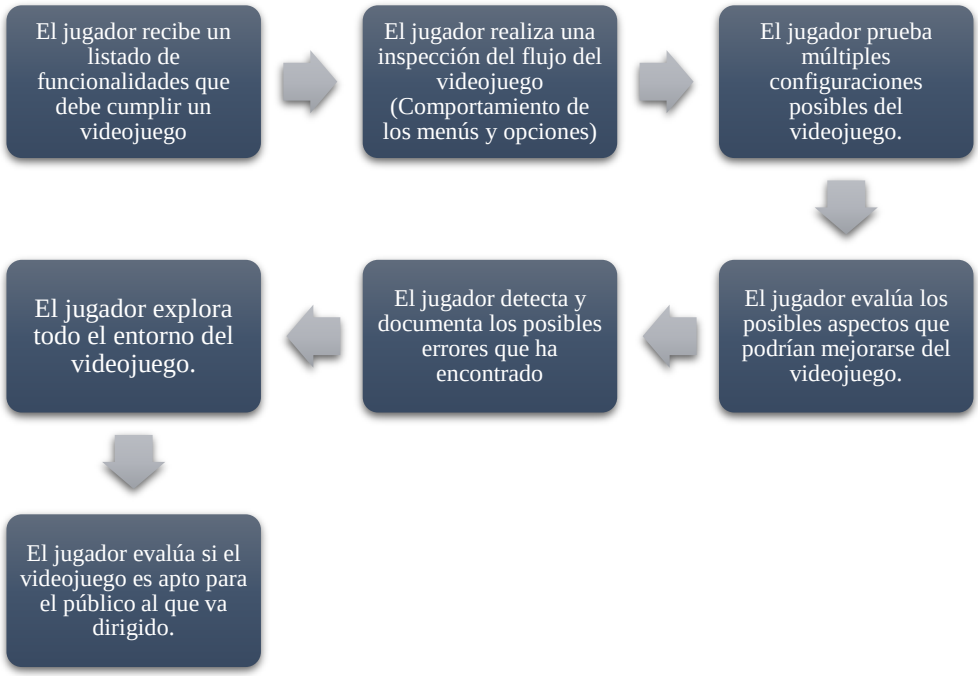


Figura 31: Proceso de revisión por parte del jugador

Fuente: Ordóñez, 2022

Existen distintos tipos de defectos dentro de un videojuego, muchos de ellos realmente no afectan en gran medida la experiencia del jugador, y pueden ser corregibles con parches posteriores, estos son los errores que el equipo de desarrollo debe considerar mayoritariamente como errores graves y que deben corregirse de inmediato.

Crash: Este error provoca que el videojuego deje de funcionar y el dispositivo expulsa al usuario de la aplicación indicando un mensaje de error, este defecto es el más grave de todos pues detiene por completo la experiencia de juego, y puede significar fuertes críticas para el equipo de desarrollo.

Congelamiento: Este error provoca que el videojuego se detenga por pocos segundos.

Tasa de cuadros: Este error corresponde al rendimiento del videojuego y provoca que un juego se ralentice constantemente, sobre todo en entornos con muchos elementos, este error sucede comúnmente cuando la capacidad del dispositivo no tiene la suficiente capacidad para correr escenas puntuales del videojuego.

Reporte de Errores

Tabla 26: Reporte de errores

Tipo de Error	Interfaz/Jugabilidad/Interfaz etc.
Resumen	Defecto encontrado y problema que causa.
Secuencia	Acciones que desencadenan ese defecto.
Comportamiento ideal	Acciones deseadas.
Nivel	Zona o nivel donde se presenta el defecto.
Prioridad	Alta/Media/Baja

Elaborado por: Ordóñez, 2022

Una vez se han realizado todas las pruebas, el equipo debe destinar un tiempo a solucionar estos defectos. El escenario ideal consiste en una actualización del videojuego corrigiendo estos defectos, por esta razón AGP hace énfasis en realizar una correcta integración durante la etapa de producción, pues cada defecto que no se haya corregido durante el desarrollo y siga avanzando provocará efectos adversos más graves, lo que le representa al equipo retrasar el proyecto debido a la complejidad en solucionar dicho defecto, el equipo debe propiciar a llegar al prototipo final con la menor cantidad de defectos posibles, y los pocos defectos que se encuentren puedan solucionarse rápidamente.

El prototipo final es enviado al stakeholder a través del Game Owner, quien finalmente prueba el videojuego y determina si el videojuego cumple con las expectativas planteadas.

Beta Cerrada

El tester de videojuegos le ayuda al equipo de desarrollo a encontrar defectos y a proponer mejoras que pueden beneficiarle al videojuego, sin embargo, una vez que el stakeholder ha aprobado el videojuego, AGP promueve el lanzamiento del videojuego a través de una Beta Cerrada, esto quiere decir que el videojuego es lanzado para un selecto grupo de jugadores que tendrán un acceso anticipado al videojuego, de esta manera se puede determinar con mayor exactitud el grado de aceptación que tiene el videojuego frente a la comunidad de jugadores.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- El modelo de evaluación por criterios y puntajes es efectivo para comparar metodologías de acuerdo con una serie de parámetros establecidos.
- Se ha planteado una metodología ágil que contempla los aspectos más esenciales de un videojuego sin dejar de lado los principios básicos de Ingeniería de software.
- Scrum y XP se fusionan manteniendo la filosofía de que Scrum señala los eventos que hay que seguir y XP señala como se deben llevar a cabo dichos eventos, lo cual ocurre debido a su alta capacidad para adaptarse a entornos de trabajo diversos.
- Las metodologías de desarrollo le permiten al equipo poder responder a los cambios que se producen constantemente en la creación de un videojuego.
- El prototipo es fundamental para el éxito del videojuego, en el, se logra detectar varios fallos e idear posibles mejoras.
- El líder del proyecto es quien asegura la comunicación entre todos los miembros del equipo de manera que ambas partes logren trabajar de manera asíncrona sin presentar inconvenientes mayores.
- Un videojuego está compuesto por varios profesionales en distintos campos, lo cual resulta necesario pues aun cuando los programadores puedan desempeñar otro tipo de actividades alternas no implica que mantengan un alto dominio sobre estas.
- La utilización de ontologías permite que ambas partes del equipo puedan comprender la estructura del videojuego a pesar de desempeñar tareas diferentes.
- La priorización de los requerimientos permite comprender el grado de complejidad de cada uno de ellos y poder estimar tanto el tiempo como el esfuerzo que será necesario para cumplirlos.
- Los errores de diseño del videojuego conllevan a que durante la implementación se produzcan muchas dificultades o los resultados no sean los esperados.

- Los problemas de comunicación se dan en un entorno multidisciplinario debido a la ambigüedad de la información generada en el proceso de documentación, la cual no es comprendida correctamente por ambas partes del equipo.

Recomendaciones

- El equipo siempre debe buscar innovar en el videojuego que están creando a través de mejoras continuas, sin embargo, tampoco deben descuidar la estructura principal del videojuego y los rasgos característicos que tiene.
- La definición de una arquitectura de software para crear el videojuego es de extrema importancia, el equipo debe esforzarse en definir la arquitectura que tendrá su videojuego durante la planificación del proyecto.
- Los miembros del área técnica deben destinar la cantidad de tiempo que consideren necesaria para trabajar con los miembros del área artística, la comprensión de una funcionalidad debe estar claramente comprendida por ambas partes, aun si esto significa reuniones de mayor duración.
- Las participaciones del jugador no deben limitarse enteramente a la fase de pruebas, ya que lo más recomendable es que el jugador pueda participar incluso desde las etapas de diseño, puesto que su retroalimentación le permite al equipo tomar en cuenta la usabilidad del videojuego, que es un factor clave.
- El equipo debe tratar de definir claramente los requerimientos en las fases iniciales, ya que, en el refinamiento, solo se busca mejorar y actualizar el requerimiento mas no alterarlo completamente, esto último debe evitarse.
- Independientemente de la metodología que un equipo de desarrollo aplique, el proceso de desarrollo del videojuego debe estar basado en tres fases fundamentales: preproducción para toda la planeación del proyecto, producción para el desarrollo de las tareas planeadas y postproducción para el lanzamiento del videojuego.
- Durante el desarrollo de un videojuego, el equipo de desarrollo no debe separar completamente la parte artística de la parte técnica, ambas partes del equipo deben encontrarse en constante comunicación para que no existan malentendidos.
- El equipo de desarrollo siempre debe mantener autoorganización para que puedan determinar la estructura del equipo que más se adecue a su estilo de trabajo.

- Todas las partes interesadas en el proyecto deben mantenerse actualizados sobre cómo está funcionando el mercado de los videojuegos día tras día.
- El equipo de desarrollo debe determinar un alcance realista del videojuego, tomando en cuenta sus recursos y sus capacidades, para no enfrentarse a una posible cancelación del proyecto.
- Es recomendable que el equipo ponga una atención especial en las pruebas que se realicen tanto durante la etapa de producción como de postproducción, puesto que los errores técnicos pueden afectar gravemente la experiencia del usuario.
- Es recomendable que el líder trate de instruir al equipo sobre incógnitas que puedan presentarse sobre la metodología.
- El compromiso es el pilar fundamental de la metodología de desarrollo, cada miembro del equipo debe comprometerse a aprender y mejorar en todo momento.

Referencias Bibliográficas

Acerenza, N., Coppes, A., Mesa, G., Viera, A., Fernández, E., Laurenzo, T., & Vallespir, D. (2009). *Una metodología para desarrollo de videojuegos: versión extendida*.

Meléndez, S., Gaitan, M., Pérez, N. (2016). *METODOLOGIA ÁGIL DE DESARROLLO DE SOFTWARE PROGRAMACION EXTREMA.* Universidad nacional autonoma de nicaragua, Managua

Aurava, R., Meriläinen, M., Kankainen, V., & Stenros, J. (2021). Game jams in general formal education. *International Journal of Child-Computer Interaction*.

Bahit, E. (2012). *Scrum y eXtreme Programming para Programadores*. Buenos Aires

Cartuche, E. (2015). *Desarrollo de un Videojuego para la Enseñanza-Aprendizaje de la Historia de los Sitios Turísticos de la Ciudad de Loja*. Universidad Nacional de Loja

Espinoza-Meza, A. (2013). *MANUAL PARA ELEGIR UNA METODOLOGÍA DE DESARROLLO DE SOFTWARE DENTRO DE UN PROYECTO INFORMÁTICO*. Universidad de Piura

Garita Hernández, F. A., Lizano Madriz, F., & Cordero Esquivel, C. M. (2019). *Metodologías para el Desarrollo de Videojuegos Serios: Una Revisión de Literatura Methodologies for the Development of Serious Video Games: A Literature Review*.

Iglesias, A. (2011). *Desarrollo de Videojuegos*. Universidad Nacional de Luján.

Kniberg, H. (2007). *Scrum y XP desde las trincheras : como hacemos scrum*. InfoQ

Kristiadi, D. P., Sudarto, F., Sugiarto, D., Sambera, R., Warnars, H. L. H. S., & Hashimoto, K. (2019). Game Development with Scrum methodology. *2019 International Congress on Applied Information Technology, AIT 2019*. Institute of Electrical and Electronics Engineers Inc.

Llansó, D. (2014). *Metodología ontológica para el desarrollo de videojuegos (TESIS*

DOCTORAL). UNIVERSIDAD COMPLUTENSE DE MADRID

Llopis, J. A. (2017). *Aplicando técnicas de Ingeniería del Software al desarrollo de Videojuegos:Ogame, un caso práctico. UNIVERSIDAD DE ALMERIA*

León, J. (2014). App Web Scrum. EPSEVG

Martinez, R. (2007). *Videojuegos, cultura y jóvenes*. Colexio Profesional de Xornalistas de Galicia

Sentrio. (2021). *Ejemplo de proceso de integración continua (CI) utilizando Jenkins*. Retrieved from: https://sentrio.io/blog/que-es-jenkins/

Mendes Calo, K., Estevez, E., & Fillottrani, P. (2010). *A Quantitative Framework for the Evaluation of Agile Methodologies*. Journal of Computer Science & Technology; vol. 10, no. 2

Mitre-Hernández, H. A., Lara-Alvarez, C., González-Salazar, M., & Martín, D. (2016). Decreasing rework in video games development from a software engineering perspective. *Advances in Intelligent Systems and Computing*, *405*, 295–304.

Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2009). The Scrum Primer. España :Scrum Training Institute

Moncada Jiménez, J., & Chacón Araya, Y. (2012). *El efecto de los videojuegos en variables sociales, psicológicas y fisiológicas en niños y adolescentes* (Vol. 21). Universidad de Costa Rica, San José

Morales, G., López, C., Fernández, L., & Rey, M. (2010). *PROCESOS DE DESARROLLO PARA VIDEOJUEGOS*. Universidad Autónoma de Ciudad Juárez

Navarro, V. (2012). *LIBERTAD DIRIGIDA: ANÁLISIS FORMAL DEL VIDEOJUEGO COMO SISTEMA, SU ESTRUCTURA Y SU AVATARIDAD*. (Tesis Doctoral). UNIVERSITAT ROVIRA I VIRGILI

Schofield, B. (2007). Embracing Fun: Why Extreme Programming is Great for Game Development. Retrieved from https://www.gamedeveloper.com/design/embracing-fun-why-extreme-programming-is-great-for-game-development

Subhash, K., & Maruthi, R. (2007). Lifecycle for Game Development to Ensure Enhanced Productivity. *International Journal of Innovative Research in Computer*

and Communication Engineering (An ISO, 3297

Subra, J.-P., & Vannieuwenhuyse, A. (2018). *Scrum: un método ágil para sus proyectos*. Barcelona: Ediciones ENI

Zapirain, E. A., & Massa, S. M. (2017). *Videojuegos en Serio Creando Serious Games para aprender jugando*. Universidad Nacional de Mar Del Plata

Jondya, A. G., Anwar, H., Hermawan, M., & Deru, T. (2022). *Developing "EMBER": A Fantasy Puzzle Adventure Game with An Approach to Agile by Applying Game-Scrum Methodology.* Bina Nusarta University

Muñoz-Arteaga, J., Gutierrez, R., Alvarez, F. (2013). *Arquitectura de Software para Juegos Serios con Aspectos Culturales: Caso de Estudio en un Videojuego para Formulas Temperatura*. Universidad Autónoma de Aguscalientes

Gómez, M. (2007). *Arquitectura y metodología para el desarrollo de sistemas educativos basados en videojuegos* . Universidad Complutense de Madrid

Jarzębowicz, A., & Weichbroth, F. (2013). *A Qualitative Study on Non-Functional Requirements in Agile Software Development*. Gdańsk University of Technology

Contreras-Espinosa, R., & Eguia-Gomez, F. (2022). *Game Jams as Valuable Tools for the Development of 21st-Century Skills.* Polytechnic University of Catalonia

ANEXOS

Anexo 1. Encuesta sobre metodologías de desarrollo en la creación de videojuegos

Encuesta sobre Desarrollo de Videojuegos

*Obligatorio

Protección de Datos

El responsable de la Encuesta Xavier Alejandro Ordóñez Vásquez, recoge estos datos a través de Google Forms con fines netamente académicos, al marcar la casilla accede a participar en la encuesta.

1. Acepta participar en la encuesta *

Selecciona todos los que correspondan.

- [ ] Si

Salta a la pregunta 2

2. ¿Cuál de las siguientes metodologías aplicó para el desarrollo del videojuego? *

Marca solo un óvalo.

- ◯ Scrum
- ◯ Extreme Programming(XP)
- ◯ Rational Unified Process (RUP)
- ◯ Rapid Application Development (RAD)
- ◯ Adaptative Software Development (ASD)
- ◯ Team Software Process(TSP)
- ◯ Otro: _____

3. ¿Considera que la metodología fue efectiva para responder a los cambios que se presentaron durante el desarrollo? *

Marca solo un óvalo.

- Totalmente de acuerdo
- De acuerdo
- Ni de acuerdo ni desacuerdo
- En desacuerdo
- Totalmente en desacuerdo

4. ¿Se logró definir con claridad los requerimientos funcionales y no funcionales? *

Marca solo un óvalo.

- Totalmente de acuerdo
- De acuerdo
- Ni de acuerdo ni desacuerdo
- En desacuerdo
- Totalmente en desacuerdo

5. ¿La comunicación que tuvieron los miembros del equipo enfocados en el área de desarrollo con los miembros del equipo enfocados en el área artística fue clara y comprensible para ambas partes? *

Marca solo un óvalo.

- Totalmente de acuerdo
- De acuerdo
- Ni de acuerdo ni desacuerdo
- En desacuerdo
- Totalmente en desacuerdo

6. ¿Resultó dificultoso ejecutar las planificaciones establecidas? *

Marca solo un óvalo.

- Totalmente de acuerdo
- De acuerdo
- Ni de acuerdo ni desacuerdo
- En desacuerdo
- Totalmente en desacuerdo

7. Cuándo se presentaron cambios importantes en el desarrollo, ¿Considera que el equipo fue capaz de responder a esos cambios en los tiempos establecidos? *

Marca solo un óvalo.

- Totalmente de acuerdo
- De acuerdo
- Ni de acuerdo ni desacuerdo
- En desacuerdo
- Totalmente en desacuerdo

8. ¿Considera que el número de programadores presentes en el área de desarrollo fue suficiente? *

Marca solo un óvalo.

- Totalmente de acuerdo
- De acuerdo
- Ni de acuerdo ni desacuerdo
- En desacuerdo
- Totalmente en desacuerdo

9. Señale las opciones. La documentación obtenida durante el desarrollo fue: *

Selecciona todos los que correspondan.

- [ ] Amplia
- [ ] Corta
- [ ] Concisa
- [ ] Ambigua
- [ ] Suficiente
- [ ] Insuficiente

10. ¿Le fue posible al equipo generar un prototipo funcional al final de cada iteración? *

Marca solo un óvalo.

- ◯ Si
- ◯ No

11. ¿La retroalimentación recibida por parte del jugador en cada prototipo fue suficiente para encontrar y corregir errores? *

Marca solo un óvalo.

- ◯ Totalmente de acuerdo
- ◯ De acuerdo
- ◯ Ni de acuerdo ni desacuerdo
- ◯ En desacuerdo
- ◯ Totalmente en desacuerdo

12. Durante el desarrollo, ¿Con qué frecuencia se producían cambios en los requerimientos previamente establecidos? *

Marca solo un óvalo.

- ◯ Baja
- ◯ Media
- ◯ Alta

13. En cada iteración con qué frecuencia se daba la reutilización de código para otros componentes del videojuego (NPC, Mecánicas, Parámetros de puntaje, etc....) *

Marca solo un óvalo.

- ◯ Baja
- ◯ Media
- ◯ Alta

14. Seleccione cuál de las siguientes métricas tuvo más peso para el equipo durante todo el ciclo de vida del videojuego *

Marca solo un óvalo.

- ◯ Métricas de Productividad
- ◯ Métricas de Rendimiento
- ◯ Métricas de Defectos
- ◯ Métricas de Usabilidad

15. ¿Cuál de los siguientes enfoques de diseño manejó durante el ciclo de vida del videojuego? *

Marca solo un óvalo.

- ◯ Diseño centrado en la tecnología
- ◯ Diseño centrado en el contenido
- ◯ Diseño centrado en el usuario

16. ¿Los miembros del equipo especializados en una actividad específica, desempeñaron también otras actividades ajenas a su área durante el desarrollo del videojuego? *

Marca solo un óvalo.

- ◯ Si
- ◯ No

17. ¿Considera que el producto final cumplió con las expectativas esperadas? *

Marca solo un óvalo.

- Totalmente de acuerdo
- De acuerdo
- Ni de acuerdo ni desacuerdo
- En desacuerdo
- Totalmente en desacuerdo

18. ¿Considera que la gestión de calidad del videojuego fue efectiva? *

Marca solo un óvalo.

- Totalmente de acuerdo
- De acuerdo
- Ni de acuerdo ni desacuerdo
- En desacuerdo
- Totalmente en desacuerdo

19. ¿El tiempo destinado a realizar pruebas de los prototipos y corrección de los defectos fue suficiente? *

Marca solo un óvalo.

- Totalmente de acuerdo
- De acuerdo
- Ni de acuerdo ni desacuerdo
- En desacuerdo
- Totalmente en desacuerdo

20. ¿En qué fase de desarrollo considera que el equipo presentó más inconvenientes? *

Marca solo un óvalo.

- ◯ Planificación
- ◯ Análisis
- ◯ Diseño
- ◯ Implementación
- ◯ Pruebas

21. ¿El Equipo tuvo que recurrir a Crunch para terminar de completar las tareas asignadas? *

Marca solo un óvalo.

- ◯ Si
- ◯ No

22. ¿Considera que el documento de diseño logro tomar en cuenta los aspectos más esenciales del videojuego? *

Marca solo un óvalo.

- ◯ Totalmente de acuerdo
- ◯ De acuerdo
- ◯ Ni de acuerdo ni desacuerdo
- ◯ En desacuerdo
- ◯ Totalmente en desacuerdo

23. ¿Quiénes participaron en la creación del documento de diseño del videojuego? *

Marca solo un óvalo.

- ◯ Diseñadores
- ◯ Programadores
- ◯ Ambos

24. ¿El equipo pudo gestionar correctamente las historias de usuario durante el ciclo de vida del videojuego, a pesar de los cambios que se presentaron durante el desarrollo? *

Marca solo un óvalo.

- ◯ Totalmente de acuerdo
- ◯ De acuerdo
- ◯ Ni de acuerdo ni desacuerdo
- ◯ En desacuerdo
- ◯ Totalmente en desacuerdo

25. Para la validación y verificación de los prototipos, ¿Cuáles de estas pruebas utilizó mayoritariamente? *

Selecciona todos los que correspondan.

- ☐ Pruebas Unitarias
- ☐ Pruebas de Integración
- ☐ Pruebas de Aceptación
- ☐ Pruebas de Regresión
- ☐ Pruebas de Rendimiento
- ☐ Pruebas de Usabilidad

26. ¿Con que frecuencia se expandía el documento de diseño? *

Marca solo un óvalo.

- ◯ Baja
- ◯ Media
- ◯ Alta