

# **Advertencia**

“Este documento contiene información desarrollada por la Escuela Superior de Cómputo del Instituto Politécnico Nacional, a partir de datos y documentos con derecho de propiedad y por lo tanto, su uso quedará restringido a las aplicaciones que explícitamente se convengan.” La aplicación no convenida exime a la escuela su responsabilidad técnica y da lugar a las consecuencias legales que para tal efecto se determinen. Información adicional sobre este reporte técnico podrá obtenerse en: La Subdirección Académica de la Escuela Superior de Cómputo del Instituto Politécnico Nacional, situada en Av. Juan de Dios Bátiz s/n Teléfono: 57296000, extensión 52000.

# Índice general

|  |           |
|--|-----------|
| <b>Advertencia</b>   | <b>I</b>  |
| <b>1. Introducción</b>                                       | <b>1</b>  |
| <b>2. Planteamiento del problema</b>                         | <b>2</b>  |
| 2.1. Contexto . . . . .                                      | 2         |
| <b>3. Marco teórico conceptual</b>                           | <b>3</b>  |
| 3.1. Videojuego. . . . .                                     | 3         |
| 3.1.1. ¿Qué es un videojuego? . . . . .                      | 3         |
| 3.1.2. Características del videojuego. . . . .               | 3         |
| 3.1.3. Clasificación del videojuego. . . . .                 | 4         |
| 3.1.4. Industria del videojuego. . . . .                     | 7         |
| 3.1.5. Industria del juego en México . . . . .               | 8         |
| 3.2. Desarrollo de videojuegos. . . . .                      | 10        |
| 3.2.1. Línea de producción de un videojuego. . . . .         | 10        |
| 3.2.2. Metodologías de desarrollo de videojuegos. . . . .    | 10        |
| 3.2.3. Software para el desarrollo de videojuegos. . . . .   | 17        |
| 3.3. Cultura. . . . .  | 22        |
| 3.3.1. ¿Qué es la cultura? . . . . .                         | 22        |
| 3.3.2. Características de la cultura . . . . .               | 22        |
| 3.3.3. Clasificación de la cultura. . . . .                  | 23        |
| 3.3.4. Importancia de la cultura. . . . .                    | 23        |
| 3.4. Gamificación . . . . .                                  | 24        |
| 3.4.1. ¿Qué es la gamificación? . . . . .                    | 24        |
| 3.4.2. Elementos del videojuego en la gamificación. . . . .  | 24        |
| 3.4.3. Tipos de jugadores. . . . .                           | 25        |
| 3.4.4. Intervalos de actividades de la gamificación. . . . . | 25        |
| 3.4.5. Tipos de diversión. . . . .                           | 26        |
| 3.4.6. Framework de diseño de la Gamificación . . . . .      | 27        |
| <b>4. Alcance del proyecto</b>                               | <b>28</b> |
| 4.1. Objetivos del proyecto . . . . .                        | 28        |
| 4.1.1. Objetivos generales . . . . .                         | 28        |
| 4.1.2. Objetivos específicos . . . . .                       | 28        |
| 4.2. Alcance del proyecto . . . . .                          | 28        |
| 4.3. Metodología de trabajo . . . . .                        | 29        |

|           |  |           |
|-----------|--|-----------|
| 4.4.      | Especificaciones de plataforma . . . . .           | 29        |
| 4.4.1.    | Hardware requerido . . . . .                       | 29        |
| 4.4.2.    | Software requerido . . . . .                       | 30        |
| 4.5.      | Productos esperados . . . . .                      | 31        |
| <b>5.</b> | <b>Trabajo realizado</b>                           | <b>32</b> |
| 5.1.      | Etapa de Preproducción . . . . .                   | 32        |
| 5.1.1.    | Primer Sprint Huddle de Preproducción . . . . .    | 32        |
| 5.1.2.    | Segundo Sprint Huddle de Preproducción . . . . .   | 32        |
| 5.1.3.    | Tercer Sprint Huddle de Preproducción . . . . .    | 33        |
| 5.1.4.    | Cuarto Sprint Huddle de Preproducción . . . . .    | 33        |
| 5.2.      | Etapa de producción . . . . .                      | 33        |
| 5.2.1.    | Primer Sprint Huddle de Producción. . . . .        | 34        |
| 5.2.2.    | Segundo Sprint Huddle de Producción. . . . .       | 34        |
| 5.2.3.    | Tercer Sprint Huddle de Producción. . . . .        | 34        |
| 5.2.4.    | Cuarto Sprint Huddle de Producción. . . . .        | 35        |
| 5.2.5.    | Quinto Sprint Huddle de Producción . . . . .       | 35        |
| 5.2.6.    | Sexto <i>Sprint Huddle</i> de Producción . . . . . | 36        |
| <b>6.</b> | <b>Resultados obtenidos</b>                        | <b>40</b> |
| <b>7.</b> | <b>Conclusiones</b>                                | <b>41</b> |
| <b>8.</b> | <b>Anexos</b>                                      | <b>46</b> |
| 8.1.      | Interfaces . . . . .                               | 46        |
| 8.2.      | Diseño de Personajes . . . . .                     | 46        |
| 8.3.      | Modelo de Datos . . . . .                          | 46        |
| 8.4.      | Control de adicción en el jugador . . . . .        | 46        |
| 8.5.      | Maquetas de niveles . . . . .                      | 46        |

# **Capítulo 1**

## **Introducción**

# **Capítulo 2**

## **Planteamiento del problema**

### **2.1. Contexto**

En México existe un problema de deficiencia en el sistema educativo, así lo señalan diferentes pruebas cuya finalidad son medir y comparar el nivel de conocimientos de los estudiantes mexicanos entre entidades estatales dentro del territorio nacional y con respecto a otros países. El Programa para la Evaluación Internacional de Estudiantes (P.I.S.A.) es una de las pruebas encargadas de medir el desempeño del sistema educativo de un país. Esta prueba es realizada por la Organización para la Cooperación y el Desarrollo Económico (O.C.D.E.). La O.C.D.E esta compuesta por 35 países, entre ellos México. La prueba P.I.S.A. tiene como objetivo evaluar hasta que punto los estudiantes cercanos a concluir la educación obligatoria han desarrollado los conocimientos y habilidades necesarios para la participación plena en la sociedad del saber[1], por lo que se realiza en jóvenes de 15 años en 72 países. En el 2016, México ocupó lugares 58, 55 y 56 de en la prueba PISA en materia de conocimiento científico, lectura y comprensión lectora y matemáticas respectivamente[2], lo que ubica el nivel de aprendizaje en México por debajo de la media internacional.

Anudado a las deficiencias del sistema educativo, México presenta un déficit en cuanto a libros libros leídos, esto se puede decir con base en diferentes estudios que determinaron la cantidad de libros leídos en México, de los cuales tres son los más citados por especialistas en el tema, éstos son:

- El realizado por la Organización de las Naciones Unidas (O.N.U.), en donde México tuvo una media de 2.8 libros leídos anualmente, ubicando a México en el penúltimo lugar de entre 108 países[3].
- La Encuesta Nacional de Lectura y Escritura de Conaculta en donde se obtuvo un promedio 5.3 libros al año, entre mexicanos mayores a 13 años[3].
- El más reciente Modulo de Lecitura (M.O.L.E.C.) del Instituto Nacional de Estadística y Geografía(I.N.E.G.I.). Con base en los resultados obtenidos durante la encuesta del 2018, el 76.4 % de la población mayor de 18 años alfabeto lee algún material considerado por el M.O.L.E.C; esta cifra presupone una disminución del 3.3 % con respecto al 2017. De este 76.4 % solo el 45.1 % tiene como material de lectura los libros, de los cuales el 40.8 % prefiere los libros de literatura[4].

# Capítulo 3

## Marco teórico conceptual

### 3.1. Videojuego.

En esta sección se define lo que es un videojuego, sus características, los tipos de clasificaciones, haciendo principal inac pie en las dos de las más importantes: la clasificación por su contenido y por su jugabilidad, finalizando con la el estado de la industria del videojuego a nivel internacional y en México.

#### 3.1.1. ¿Qué es un videojuego?

El grupo Carricay define al videojuego como: "una aplicación interactiva orientada al entretenimiento que, a través de ciertos mandos o controles, permite simular experiencias en la pantalla de un televisor, una computadora u otro dispositivo electrónico"[5].

#### 3.1.2. Características del videojuego.

Al igual que con otros productos tecnológicos, la evolución de los videojuegos ha sido vertiginosa, resultando complicado mencionar características comunes para todos los videojuegos. Sin embargo, en el libro "Marketing y videojuegos: Product placement, in-game, adevertising y advergaming" se menciona que existen seis características comunes en los videojuegos: Interactividad, entretenimiento, jugabilidad, simulación \virtualidad, inmersión y multiplataformidad [6]; a continuación se hace mención en que consisten cinco de las seis características, esto debido a que la última no se encuentra presente en todos los juegos y el mismo autor de la obra la menciona como una característica opcional a tomar en cuenta:

- **Interactividad:** En el artículo "Defining Virtual Reality: Dimensions Determining Telepresence" se define la interactividad como la capacidad de los usuarios para participar y modificar la forma y el contenido de un entorno mediado en tiempo real[7].
- **Entretenimiento:** en el artículo "Las Tecnologías del Entretenimiento: Pasado, Presente y Futuro", el entretenimiento "se asocia, usualmente, de hacer algo que nos divierte, algo que podemos hacer solos o con otros, para entretenernos o divertirnos, en nuestro tiempo libre, o tal vez, algo que nos relaje o que nos haga reír"[8].
- **Jugabilidad:** en el libro "Marketing y videojuegos: Product placement, in-game, adevertising y advergaming" se define la jugabilidad como "la relación que existe entre todas las acciones

reacciones e interacciones tanto del videojugador como el videojuego como entre los propios sistemas y subsistemas programados en el videojuego"[6].

- **Simulación \Virtualidad:** La simulación "se trata de una representación a medida cuyo objetivo nos permite interactuar y relacionarnos con lo representado según nuestros intereses"[6].
- **Inmersión:** Con base en el libro "La vida en la pantalla: La construcción de la identidad en la era de internet", la inmersión es un proceso psicológico que se produce cuando la persona deja de percibir de forma clara su medio natural al concentrar toda su atención en un objeto, narración, imagen o idea que le sumerge en un medio artificial [9]. Por su parte en la tesis "Libertad dirigida: Análisis formal del videojuego como sistema, su estructura y su avataridad", la inmersión se entiende como la coherencia de la ficción del juego y su aceptación por el jugador.[10]

### 3.1.3. Clasificación del videojuego.

Los videojuegos pueden ser clasificados con base a diferentes factores según lo que se desee saber de ellos. Las clasificaciones más populares son basándose en:

- **Contenido:** Esta clasificación evalúa desde la temática del videojuego, dificultad, el argumento(en caso de que el videojuego tenga), etc. y agrupa los juegos determinando el público para el que son aptos.
- **Jugabilidad:** Clasificación se basa en los puntos en común de la jugabilidad entre videojuegos y los clasifica en diferentes géneros; en esta clasificación un videojuego puede estar en dos géneros a la vez.
- **Plataforma:** Clasificación hecha con base en la plataforma para la que el juego se encuentra disponible tales como computadoras, móviles o consolas.
- **Cantidad de jugadores que el juego soporta al mismo tiempo:** Puede ser un solo jugador o múltiples jugadores.
- **Conectividad:** Esta clasificación se basa en si el juego necesita de conexión a Internet o no para funcionar o para algunas de sus funcionalidades.

En esta sección únicamente se va a profundizar en las clasificaciones por contenido y jugabilidad.

#### Clasificación por contenido.

Esta clasificación consiste en evaluar los contenidos y elementos de interactividad de los videojuegos y clasificarlos en los grupos de edad para los que son aptos [11]. El objetivo de esta clasificación es que los consumidores puedan comprar videojuegos de manera informada sobre el contenido de los juegos.

La necesidad de clasificar a los juegos por su contenido nació en 1994 luego de la controversia que trajo el estreno de Mortal Kombat ®, juego de peleas famoso por su alto contenido de violencia [12]; siendo este alto contenido de violencia lo que levantó la preocupación entre diferentes jefes de familia con respecto al tipo de contenido que sus hijos tenían acceso con los videojuegos. Actualmente existen diversas organizaciones se encargan de clasificar a los videojuegos por su contenido. A continuación se mencionan las más importantes:

- Organización de clasificación de entretenimiento informático (CERO, por sus siglas en inglés).
- Autocontrol en el Software de Entretenimiento (USK, por sus siglas en Aleman).
- Junta de Clasificación de Software de Entretenimiento (ESRB, por sus siglas en inglés).
- Información paneuropea del juego (PEGI, por sus siglas en inglés).

Cada una opera en determinada zona geográfica, esto debido a las diferencias culturales entre países. En algunos países, las instituciones gubernamentales son las encargadas de la clasificación de los videojuegos, tal es el caso de China y Rusia. Si bien no existe ninguna normatización internacional que obligue a los creadores de videojuegos a someter sus productos a esta evaluación(salvo por algunos países con este tipo de legislaciones), algunas distribuidoras como Nintendo®, Sony®, Microsoft®marcan como una condición obligatoria para todos aquellos que deseen publicar contenido en sus dispositivos [13].

Actualmente México esta afiliado a la ERSB, por lo que en la siguiente sección se profundiza en ésta; sin embargo, es importante mencionar que se desconoce si seguirá bajo este estándar ya que en el 2017, el Senado mexicano aprobó una nueva legislación en materia de clasificación de contenido de videojuegos [14].

### **Junta de Clasificación de Software de Entretenimiento.**

La ERSB fue fundada en 1994; su centro de operaciones se ubica en Estados Unidos y Canadá. Actualmente cuenta opera en Norteamérica y América latina.

Los juegos que han sido clasificados por la ERSB cuentan con la etiqueta (ver figura 3.1) en su empaque tal como se muestra en la figura 3.2.

La etiqueta que emite la ERSB esta compuesta por tres partes:

- **Categoría de clasificación:** Sugiere la edad adecuada para el público del videojuego, ver figura 3.1.
- **Descriptores de contenido:** Indica los elementos que motivaron la Categoría de clasificación o son motivo de preocupación para el consumidor.
- **Elementos interactivos:** Informan acerca de los aspectos interactivos de los productos, como recursos de memoria necesarios para su funcionamiento, uso de conexión a internet o cantidad de jugadores de manera simultanea.

### **Clasificación por jugabilidad.**

Esta clasificación es empleada por la industria para diferenciar los tipos de juegos y de jugadores. A continuación se presenta la clasificación propuesta en el libro "Juego. Historia, Teoría y Práctica del Diseño Conceptual de Videojuegos"[15]



Figura 3.1: Etiqueta de clasificación de la ERSB. Esta compuesta de tres partes: 1. Categorías de clasificación. 2. Descriptores de contenido. 3. Elementos Interactivos [Imagen](s/f) Recuperado de: <http://media.blizzard.com/global-video-player/ratings-png/wow-cataclysm-esrb-esmx.png>



Figura 3.2: Carátula de un juego con la etiqueta de clasificación ESRB [Imagen](s/f) Recuperado de: [https://vignette.wikia.nocookie.net/es.finalfantasy/images/1/1c/FF4PSP\\_NA\\_Caratula.PNG/revision/latest?cb=20110301012441](https://vignette.wikia.nocookie.net/es.finalfantasy/images/1/1c/FF4PSP_NA_Caratula.PNG/revision/latest?cb=20110301012441)

- **Juegos de acción:** Son juegos usualmente de temática violenta. El jugador lucha por su supervivencia, para ello se vale de armas o habilidades de combate. Los juegos de acción se subdividen en:
  - **Juegos de lucha:** Este tipo de juego involucra lucha cuerpo a cuerpo, con una fuerte influencia de las artes marciales.
  - **Juegos de disparos:** Son aquellos en donde el jugador mueve a su personaje para superar diversos escenarios en donde el jugador debe de hacer uso de su armamento para lograr completar el nivel.
  - **Juego de plataformas:** El jugador debe de controlar a un personaje con el que se desplazará saltando entre plataformas y esquivando todo tipo de obstáculos y enemigos.
- **Juegos de estrategia:** Para que el jugador logre sus objetivos en este tipo de juegos, éste debe de planear una estrategia, normalmente a largo plazo. La temática del juego de estrategia se relaciona más con la escala y la visión. Los juegos de estrategia usualmente involucran mapas de gran tamaño, visión sobre el campo, gestión de recursos, manejo de tropas, desarrollo de recursos, comercio e intercambio de recursos, modificadores del campo, control de territorio, etc.
- **Juegos de Rol:** Estos juegos tienen sus orígenes en los juegos de rol de mesa. La mecánica de los juegos de rol gira en torno a un grupo de héroes, con habilidades y progresión definidos; el grupo de héroes debe de trabajar coordinadamente para cumplir un objetivo; estos héroes pueden ser controlados por un solo jugador o por varios. El jugador deberá explorar un mundo de gran tamaño haciendo evolucionar a sus personajes y sus habilidades. En los juegos de rol, el uso y recolección de objetos tiene un gran peso en la capacidad de avance del jugador.
- **Videojuego de aventura:** Son parecidos a los juegos de Rol; con la peculiaridad de que tienen una progresión más lineal y no se hace tanto énfasis en los combates, siendo su eje principal la narrativa.
- **Videojuegos de deportes:** Son todos aquellos videojuegos que tratan sobre deportes que no involucren la conducción de un vehículo. Pueden ser juegos sobre fútbol, fútbol americano, tenis, etc.
- **Videojuegos de carreras de vehículos:** Son todos aquellos que se centran en las carreras con todo tipo de vehículos, mayoritariamente automóviles.
- **Videojuegos puzzle:** Este tipo de juego involucra la resolución de un problema a partir de la utilización de una serie limitada de recursos, por lo que si los recursos no se utilizan de la manera correcta el problema no podrá ser solucionado.

### 3.1.4. Industria del videojuego.

El videojuego como producto para masas nace en 1972 con el videojuego de Pong desarrollado por la empresa Atari, es así como nace las primeras máquinas recreativas, ver figura 3.3. En sus inicios los videojuegos se caracterizaban por el uso de pantallas estáticas, entornos bidimensionales y formas abstractas de juego [16]. A la aparición de las máquinas recreativas le seguirían las primeras consolas de mesa desarrolladas principalmente por Atari.



Figura 3.3: Primeras maquinas recreativas desarrolladas por Atari [Imagen](s/f) Recuperado de: <http://2.bp.blogspot.com/-98a06VymhA/VKJv2AQbNGI/AAAAAAAAXU/rASZPNmBMrg/s1600/first%2Barcades.jpg>

Durante los años subsecuentes la industria del videojuego gozaría de una alta tasa de recuperación; sin embargo la industria entraría en declive 1983, empezando el periodo conocido como la crisis de los videojuegos, misma que afectaría principalmente a las empresas estadounidenses y canadienses. Este periodo duraría hasta 1985 y provocaría una polarización del mercado en donde la industria japonesa de los videojuegos tomaría la ventaja frente a la estadounidense. En 1983 la compañía Nintendo lanza al mercado su primera consola, la Famicom o Nintendo Entertainment System(NES) como fue conocida en el resto del mundo [17]. Nintendo dio inicio a una nueva era para la industria del videojuego con el lanzamiento de títulos que se consagraron como clásicos de la industria y que al día de hoy aun se encuentran vigentes en el mercado tales como Super Mario Bros, The Legend of Zelda, Pokemón, etc (ver figura 3.4).

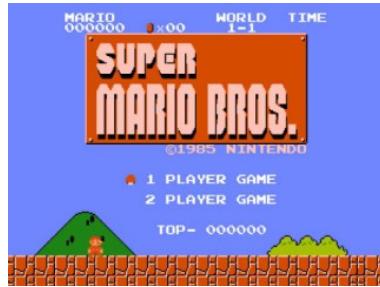
En las décadas posteriores, la industria del videojuego creció a un ritmo vertiginoso; no solo se diversificó el mercado con el nacimiento de nuevos géneros de videojuegos sino también las innovaciones tecnológicas de hardware permitieron crear cada vez mejores experiencias de juegos [17]. En menos de 20 años los videojuegos pasaron de entornos en dos dimensiones a entornos en tercera dimensión, del formato de almacenamiento en cartuchos al uso de CDs.

Anualmente la industria de los videojuegos genera miles de millones de dólares en ganancias, tan solo en el 2017 la industria generó 108.9 mil millones de dólares[18]. Estas ganancias fueron generadas un 25 % en Norteamérica, el 24 % en Europa, Medio Oriente y África, el 4 % en América Latina, siendo la zona con mayor ingresos Asia y la zona del Pacífico con el 47 % de ganancias generadas. En cuanto a la distribución de las ganancias con base a las plataformas, los teléfonos móviles destrozan a las consolas al generar el 32 % de las ganancias, mientras que las consolas han generado el 31 % de las ganancias, seguidas por las computadoras con el 23 %[18].

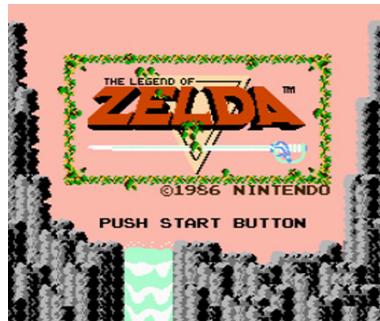
### 3.1.5. Industria del juego en México

La industria mexicana de videojuegos, al igual que la latinoamericana, es una industria emergente. Las empresas productoras de videojuegos mexicanas

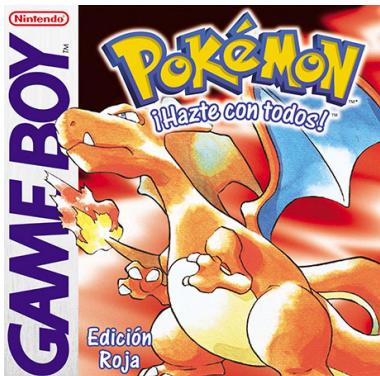
México cerró el 2017 con un consumo de videojuegos de 1.4 mil millones de dólares, lo que le valió el doceavo puesto en cuanto a consumo de videojuegos a nivel mundial y el primer lugar en América Latina. De acuerdo con los reportes de Unidad de Inteligencia Competitiva (CIU, por sus



(a) Super Mario Bros.



(b) The Legend of Zelda.



(c) Pokémon.

Figura 3.4: Super Mario Bros, The Legend of Zelda y Pokémon son algunas de las franquicias más importantes de la compañía japonesa Nintendo.[Imagen] (s/f) Recuperado de (respectivamente): <http://contenidos.enter.co/custom/uploads/2015/09/mario1.jpg>, [https://www.hiddentriforce.com/wp-content/uploads/2015/02/10106454\\_5124592\\_lz.jpg](https://www.hiddentriforce.com/wp-content/uploads/2015/02/10106454_5124592_lz.jpg), [https://vignette.wikia.nocookie.net/es.pokemon/images/d/db/Car%C3%A1tula\\_de\\_Pok%C3%A9mon\\_Rojo.jpg](https://vignette.wikia.nocookie.net/es.pokemon/images/d/db/Car%C3%A1tula_de_Pok%C3%A9mon_Rojo.jpg)

siglas en inglés) del 2016; en México hay 52.7 millones de jugadores, de los cuales el 71 % juega en dispositivos móviles[19]. Desafortunadamente y en contraposición con estas cifras el consumo de videojuegos en México está enfocado en juegos de origen extranjero.

## 3.2. Desarrollo de videojuegos.

En esta sección se describe el proceso de desarrollo del videojuego, hablando sobre los pasos que lleva dicho proceso; para después describir las metodologías de desarrollo que se emplean, algunas de las metodologías descritas en este apartado son originarias del desarrollo de software convencional pero son adaptadas al desarrollo de videojuegos por algunos estudios independientes; es importante mencionar que muchas de las metodologías de desarrollo de videojuegos son propiedad de las empresas que las utilizan y por lo tanto no son de carácter público por lo que no pudieron ser incluidas en este trabajo terminal. Para finalizar esta sección se hace mención del software empleado en el desarrollo de videojuegos, iniciando con el motor de juego, el cual se define y se explica a grandes rasgos su arquitectura.

### 3.2.1. Línea de producción de un videojuego.

Una línea de producción son los pasos o fases lógicos y secuenciales requeridos para obtener un producto. Los pasos que componen la línea de producción dependen del producto que se va a fabricar y de la empresa fabricante.

Existe una discrepancia en cuanto a que elementos tiene la línea de producción de un videojuego, siendo la planteada por la revista ING la más común. Esta línea consiste en tres etapas[20]:

- **Concepto:** Es la idea que origina a todo el juego a realizar. Esta puede ser una simple oración en la que se mencione el contexto del juego y su temática o bien puede ser un acuerdo de la compañía desarrolladora para hacer una secuela o una precuela de un videojuego ya existente[20].
- **Preproducción:** En esta etapa el equipo de producción redacta el documento de diseño del videojuego, define el argumento, los personajes y la jugabilidad. En esta etapa se determinan todas las limitantes técnicas y creativas que va a tener el proyecto[20].
- **Producción:** En esta etapa se desarrolla el juego: los artistas desarrollan todos los elementos visuales que se van a emplear, los programadores se encargan de implementar la lógica del juego y la jugabilidad establecida en la etapa de preproducción y el equipo de audio se encarga de generar todos los elementos de audio que conlleva el videojuego [20].
- **Postproducción:** En esta etapa el juego se considera casi terminado y es sometido a diferentes pruebas para medir su rendimiento, identificar todo tipo de errores y solucionarlos. También en esta etapa se intensifican las campañas de promoción para el juego[20].

### 3.2.2. Metodologías de desarrollo de videojuegos.

En esta sección se define lo que es una metodología de desarrollo de software, se mencionan tres metodologías de desarrollo de software que emplea la industria de los videojuegos y al final se menciona una metodología propia del desarrollo de videojuegos.

## ¿Qué es una metodología de desarrollo de software?

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software [21]. En palabras de Gacitúa: "Una Metodología impone un proceso de forma disciplinada sobre el desarrollo de software con el objetivo de hacerlo más predecible y eficiente. Una metodología define una representación que permite facilitar la manipulación de modelos, y la comunicación e intercambio de información entre todas las partes involucradas en la construcción de un sistema"[22].

### Metodología en cascada

La metodología de desarrollo en cascada o también conocida como modelo de vida lineal o básico, fue propuesta por Royce en 1970 y a partir de entonces ha tenido diferentes modificaciones. Sigue una progresión lineal por lo que cualquier error que no se haya detectado con antelación afectara todas las fases que le sigan provocando una redefinición en el proyecto y por ende un aumento en los costos de producción del sistema [23]. Esta metodología se divide en las siguientes etapas:

- **Análisis de los requisitos del software:** En esta etapa se recopilan los requisitos del sistema, se centra especialmente en toda aquella información que pueda resultar de utilidad en la etapa de diseño, tales como tipos de usuarios del sistema, reglas de negocio de la empresa, procesos, etc. En esta etapa se responde la pregunta de ¿Qué se hará?
- **Diseño:** Esta etapa se caracteriza por definir todas aquellas características que le darán identidad al sistema, tales como la interfaz gráfica, la base de datos, etc. Las características anteriormente definidas se obtendrán de la etapa de análisis. En esta etapa se respondería la pregunta de ¿Cómo se hará?
- **Codificación:** Terminada la etapa de diseño, lo siguiente es programar y crear todos los elementos necesarios para el funcionamiento del sistema.
- **Prueba:** Finalizada la decodificación se debe de probar la calidad del sistema. En este punto es importante resaltar que las pruebas no solo abarcan que se confirme que el sistema funcione, sino que también verifica que los usuarios puedan aprender a utilizarlo con facilidad, a su vez también se debe de verificar la seguridad de la información y los tiempos de respuesta del sistema.
- **Mantenimiento:** En esta última etapa se realizarán modificaciones al sistema, sin que esto necesariamente signifique que estos cambios se deban a errores de programación, puesto que esta etapa también abarca agregar nueva funcionalidad al sistema o, en caso de que trabaje con protocolos de estándar internacional, actualizar dichos protocolos [23].

Algunos de los inconvenientes que presenta son:

- No refleja el proceso de desarrollo real.
- Tiempos largos de desarrollo.
- Poca comunicación con el cliente.
- Revisiones de proyecto de gran complejidad.

## Metodología en Scrum

Desarrollada por Ikujiro Nonaka e Hirotaka Takeuchi a principios de los 80's, esta metodología le debe su nombre a la formación scrum de los jugadores de ruby. Scrum es una metodología eficaz para proyectos con requisitos inestables que demandan flexibilidad y rapidez, esto principalmente a su naturaleza iterativa e incremental [24].

Scrum parte de la visión general que se desea que el producto alcance; a partir de esta visión se inicia la división del proyecto en diferentes módulos. Scrum implementa una jerarquía entre los módulos en donde los módulos de mayor jerarquía son los que se desarrollaran al inicio del proyecto o durante las primeras iteraciones (sprint). Cada sprint tendrá una duración de hasta seis semanas a lo máximo [25].

Durante el proceso de desarrollo del sprint, el equipo tendrá reuniones diarias en donde se definirán metas diarias para lograr completar el objetivo del sprint. Estas reuniones deberán de ser de corta duración (no más de quince minutos) y recibirán el nombre de scrum diario. Al final de cada sprint, el equipo contará con un módulo funcional que el cliente podrá utilizar sin que el sistema este completado.

Cada sprint se compone de las siguientes fases:

- **Concepto:** se define a grandes rasgos las características del producto y se asigna a un equipo para desarrollarlo.
- **Especulación:** Con la información del concepto se delimita el producto, siendo las principales limitantes los tiempos y los costes. Esta es la fase más larga del sprint. En esta etapa se desarrolla basándose en la funcionalidad esperada por el concepto.
- **Exploración:** El producto desarrollado se integra al proyecto.
- **Revisión:** Se revisa lo construido y se contrasta con los objetivos deseados.
- **Cierre:** Se entrega el producto en la fecha programada, esta etapa no siempre significa el fin del proyecto; en ocasiones marca el inicio de la etapa de mantenimiento [26].

Uno de los principales componentes de la metodología scrum son los roles, es decir el papel que cada integrante del equipo desempeñara durante el proceso de desarrollo. Los roles se dividen en dos grupos:

- **Cerdos :** Son los que están comprometidos con el proyecto y el proceso de Scrum.
  - **Product owner:** Es el jefe del proyecto y por lo tanto es quien toma las decisiones. Esta persona es quien conoce más del proyecto y las necesidades del cliente. Es el puente de comunicación entre el cliente y el resto del equipo.
  - **Scrum Master:** Se encarga de monitorear que la metodología y el modelo funcionen. Es quien toma las decisiones necesarias para eliminar cualquier inconveniente que pueda surgir durante el proceso de desarrollo.
  - **Equipo de desarrollo:** Estas personas reciben el objetivo a cumplir del Product owner y cuentan con la capacidad de tomar las decisiones necesarias para alcanzar dicho objetivo.

- **Gallinas:** Personas que no participan de manera directa en el desarrollo, sin embargo, su retroalimentación da pie a la planeación de los sprints.
  - **Usuarios:** Son quienes utilizarán el producto.
  - **Stakeholders:** Son quienes el proyecto les aportará algún beneficio. Participan en las revisiones del sprint.
  - **Manager:** Toma las decisiones finales. Participa en la selección de objetivos y en la toma de requerimientos[25].

## Metodología de Programación extrema

La metodología de programación extrema o metodología XP(por sus siglas en inglés) fue desarrollada por Kent Beck en 1999 basándose en la simplicidad, la comunicación y la retroalimentación de código. Es una metodología de desarrollo ágil y adaptable, soporta cambios de requerimientos sobre la marcha. Su principal objetivo es aumentar la productividad y minimizar los procesos burocráticos, por lo que el software funcional tiene mayor importancia que la documentación [27].

XP se fundamenta en doce principios que se agrupan en cuatro categorías. A continuación, se hará mención de estos principios:

- **Retroalimentación:**
  - **Principio de pruebas:** Se define el periodo de pruebas de funcionalidad del software a partir de sus entradas y salidas como si se tratara de una caja negra.
  - **Planificación:** El cliente o su representante definirá sus necesidades y sobre ellas se redactará un documento, el cual servirá para establecer los tiempos de entregas y de pruebas del producto.
  - **Cliente in-situ:** El cliente o su representante se integrarán al equipo de trabajo con la finalidad de que participen en la planeación de tareas y en la definición de la funcionalidad del sistema. Esta estrategia se implementa para minimizar los tiempos de inactividad entre reuniones y disminuye la documentación a redactar.
  - **Pair-programming:** Se asignan parejas de programadores para desarrollar el producto. Esto generará mejores resultados en menores costos.
- **Proceso continuo en lugar de por bloques.**
  - **Integración continua:** Se implementan progresivamente las nuevas características del software. Esta integración no se hace de manera modular ni planeada.
  - **Refactorización:** La eliminación de código duplicado o ineficiente les permite a los programadores mejorar sus propuestas en cada entregable.
  - **Entregas pequeñas:** Los tiempos de entregas son cortos y permiten la evaluación del sistema bajo escenarios reales.
- **Entendimiento compartido.**
  - **Diseño simple:** El programa que se utiliza en los entregables es aquel que tenga la mayor simplicidad y cubra las necesidades del cliente.

- **Metáfora:** expresa la visión evolutiva del proyecto y define los objetivos del sistema mediante una historia.
  - **Propiedad colectiva del código:** Todos los programadores son dueños del programa y de las responsabilidades del programa. Un programa con muchos programadores trabajando en él es menos propenso a errores.
  - **Estándar de programación:** Se define la estructura que tendrá el programa a la hora de ser escrito, esto para dar la impresión de que una sola persona trabajo en él.
- **Bienestar del programador.**
- **Semana de 40 horas:** Se minimizan las jornadas de trabajo excesivas para garantizar el mejor desempeño del equipo [28].

Tal como se puede observar XP, es una metodología fuertemente orientada hacia los miembros del equipo, su bienestar, la interacción entre ellos y en su aprendizaje.

### Metodología Huddle

Huddle es una metodología creada por el Instituto de Ingeniería y Tecnología de Universidad Autónoma de Ciudad Juárez. Huddle recibe su nombre por las reuniones que se realizan en el fútbol americano antes de cada jugada. Su funcionalidad se basa en la metodología Scrum, con la diferencia de que está orientada al desarrollo de videojuegos. De naturaleza ágil, resulta óptimo para equipos multidisciplinarios de 5 a 10 personas; es iterativa, incremental y evolutiva [29].

Huddle se divide en tres etapas:

- **Preproducción:** Consiste en la planeación del juego. En esta etapa se redactará el documento de diseño; este documento contendrá la idea general del juego, su escritura deberá de ser tal que todos los miembros del equipo pueden entenderlo y darse una idea de cómo será el juego una vez que se haya terminado. En esta etapa se definirá el argumento del juego, sus personajes, el género del juego, sus mecánicas, la música, los efectos de sonido, los efectos especiales y su funcionalidad. Huddle proporciona plantilla para realizar este documento, dejando la posibilidad de modificarlo según el equipo considere oportuno.
- **Producción:** Es la etapa más larga y de mayor importancia. Su organización se basa totalmente en la organización iterativa e incremental de Scrum; es decir se harán reuniones diarias en donde se discutirán los objetivos de la iteración. Antes de finalizar cada Sprint, el módulo se someterá a diferentes pruebas para garantizar su funcionalidad. Cuando un Sprint finaliza, se realiza una reunión en la que los elementos del equipo discuten las decisiones tomadas y analizan cuáles fueron las decisiones y acciones más eficientes para retomarlas y desechar aquellas que atrasen al proyecto. Al finalizar esta etapa el equipo contará con las versiones alfa y beta del juego.
- **Postmorten:** En esta etapa se discuten todos los puntos positivos y negativos del proyecto. En esta evaluación se redactará un documento que permita a futuros proyectos efectuar planes de acción más efectivos [29].

### El metodo Cerny

Esta metodología fue desarrollada por Mark Cenry, arquitecto del Play Station 4 de Sony y desarrollador de juegos como Jak and Daxter y Crash Bandicoot. El método Cenry es un método para el diseño industrial de videojuegos de plataforma y acción. Se basa en dar una fuerte libertad creativa al equipo de desarrolladores durante la etapa de preproducción con la finalidad de explorar la viabilidad del producto; bajo este enfoque se desarrolla un primer prototipo y si de este no se obtiene un producto que pueda resultar entretenido para los jugadores el proyecto se descarta y se inicia uno nuevo utilizando lo aprendido en el anterior.

El método Cenry se centra en cuatro conceptos:

- **Preproducción vs. Postproducción.** La etapa de preproducción es la más importante. Esta etapa es definida como una etapa caótica y para solventarla se debe de integrar un equipo con los mejores diseñadores, programadores y artistas, estos pasaran a ser los líderes de sus equipos en la fase de producción [30].

En la preproducción se desarrolla una serie de prototipos que en la etapa de producción serán las pautas a seguir para los niveles finales del juego. Es importante entender que en la preproducción no se realiza el juego sino se diseñan y pulen las mecánicas del mismo. Lo anterior se logra bajo el enfoque de las **3 C's: personaje(Character) , camara(camera) y control (controles de movimiento)**. De igual forma en esta etapa se define: La apariencia del juego, que tecnologías se utilizaran durante el desarrollo [30].

Para que esta etapa tenga un producto exitoso el equipo debe aprovechar las oportunidades e ideas que vayan surgiendo sin centrarse en las limitaciones tecnológicas ni en el contexto del proyecto.

Al término de esta fase el equipo tendrá un primer demo del juego con la calidad suficiente como para ser publicado y un macro documento de diseño.

- **“Publishable” First Playable.** El demo de la etapa de preproducción debe de contar con dos niveles de alta calidad. Estos niveles deben de incluir:
  - Todo el comportamiento del personaje jugable
  - Todo el comportamiento de los enemigos y obstáculos,
  - La tecnología de base y todas las funcionalidades tanto locales como globales.
  - La implementación de toda la capa artística.
  - Las mecánicas totalmente definidas.
- **Macro vs Micro Design.** Este concepto hace referencia a la documentación que acompaña al videojuego. La documentación se divide en:
  - **El macro diseño:** Se desarrolla durante la etapa de preproducción. Consta de a lo más cinco páginas, contiene información sobre el personaje, sus movimientos y mecánicas

más exóticas, de igual forma contiene información sobre los niveles tales como su tamaño, estructura, progresión y contenido; finalmente este documento cuenta con un gráfico o tabla de resumen sobre sus contenido. En este documento se puede agregar arte conceptual del juego y la historia del mismo. La ventaja del uso del macro diseño es que permite una mejor planeación y seguimiento de las actividades ya que los líderes de los equipos saben totalmente a que punto se desea llegar con el juego.

- **El micro diseño:** Se realiza durante la etapa de producción. Esta documentación sigue la estructura de documentos de diseño más clásicos al incluir: scripts, mapas de niveles, diseño y comportamiento de los enemigo, descripción de los obstáculos, descripción del gameplay especial, etc.
- **Gameplay testing:** Se deben de realizar pruebas exhaustivas de jugabilidad y desempeño. Las pruebas son de vital importancia ya que estas le dirán al equipo de desarrollo que tan rápido el jugador aprende las mecánicas y se inmeciona en el juego. Sin embargo, el equipo debe de evitar la pruebas de focalización como pedir la opinión de los teasers ya que no aportan nada al juego. En lugar de estas pruebas se deben de realizar las que involucren al teaser en el juego, es decir hacerlo jugar el juego mientras se observa sus reacciones a este, dichas reacciones se deben de cuantificar y analizar los resultados.

El metodo Cerny busca desmentir los mitos del desarrollo de videojuegos, a continuación se presentan dichos mitos:

- “**Se puede planear y planificar un horario para la creación de un juego**”. Cerny plantea que esta idea no es compatible con el desarrollo de un videojuego al crear presiones en el equipo y limitar la creatividad del mismo.
- “**Se productivo significa no tirar nada**”. Cerny sostiene que en la etapa de preproducción es valido desechar tantos prototipos como sea posible ya que no son el juego en sí sino peldaños para obtener el juego.
- “**La tecnología de vanguardia es importante por lo que es lo primero que debe de desarrollarse**”. Con base en la experiencia de Cerny, al desarrollar primero la tecnología antes que el diseño del juego es que se esta limitando el potencial del juego al potencial de la tecnología.
- “**La revisión constante del proyecto es esencial para su buena gestión**”. De igual forma Cerny plantea que esta idea contradice la naturaleza caótica de la etapa de preproducción, por lo que es imposible de realizar.
- “**La versión alfa es la primera versión jugable**”. En palabras de Cerny: sin importar que la versión alfa o beta sea la primera versión jugable, esta debe de ser de suficiente calidad como para que se pueda publicar.
- “**Un proyecto cancelado es sinónimo de de una mala planeación o un mal equipo**”. Cerny propone que no necesariamente, si durante la etapa de preproducción el equipo no logra desarrollar un demo de alta calidad significa que el proyecto no es viable y por lo tanto es un buen momento para descartar el proyecto.

- “Entre más definida este la versión inicial del proyecto mayor probabilidad de éxito o a mayor detalle el documento de diseño, mejor para el proyecto”. Cenry sostiene que este principio es erróneo ya que emplear un largo periodo de tiempo únicamente en la documentación es una perdida de tiempo. Cenry hace la analogía con planear una ruta detallada sin antes conocer el terreno, puede que funcione pero en caso contrario todo el esfuerzo y recursos habrán sido en vano.
- “Si quieres ser exitoso escucha tu consumidor”. Cenry sostiene que es bueno hacerle caso al cliente sin embargo no en lo que éste dice sobre el producto, en lugar de eso son más fiables sus reacciones mientras juega el producto [30].

La principal desventaja de este método que que no siempre se obtiene un juego de este por lo que no cualquier tipo de estudio de videojuegos puede implementarlo al no ser segura la obtención de un producto y por ende de ganancias. Otra desventaja es que durante la etapa de preproducción la mayoría del equipo de trabajo se encuentra desempleado, por lo que no sería un método de desarrollo recomendable para equipos de desarrollo que a penas inician en la industria.

### 3.2.3. Software para el desarrollo de videojuegos.

En este apartado se habla del software que comúnmente se emplea para el desarrollo de videojuegos, empezando por el motor de juego, la definición del motor de juego, la arquitectura del motor de juego, los motores de juego más usados en el mercados; para finalizar con una lista del software auxiliar que se usa para generar los elementos visuales y auditivos que componen al juego.

#### Motor de juego.

El motor de juego, también conocido como Game Engine, parte del concepto de reutilización; es decir, es posible generar juegos a partir de un código base y común mediante una separación adecuada de los componentes fundamentales, tal como visualización de gráficos, control de colisiones, físicas, entrada de datos etc [31]; esto permite a quienes trabajen en un juego puedan centrarse en todos aquellos detalles que hacen al juego único.

#### Arquitectura del motor

Los motores de juego se basan en una arquitectura estructurada a capas. Por lo que las capas de nivel superior dependen directamente de las de nivel inferior [32] A continuación se mencionarán las capas que componen al motor de juego junto a una breve descripción de la capas.

- **Hardware:** esta capa se relaciona con la plataforma sobre la que se ejecutará el juego. Existen motores gráficos orientados hacia una sola plataforma (dispositivos móviles, consolas caseras, computadoras o consolas portátiles, etc.) y existe motores multiplataforma que permiten el desarrollo simultáneo de un juego para diferentes plataformas (cross-platform) [32].
- **Drivers:** Esta capa garantiza la correcta gestión de determinados dispositivos (tarjeta gráfica, tarjeta de sonido, etc.) haciendo uso de software de bajo nivel [32].
- **Sistema Operativo:** Esta capa garantiza la comunicación de los procesos que se ejecutan en el sistema operativo y los recursos de la plataforma asociada con el juego [32].

- Kits de desarrollo de software y middleware: Un Kit de desarrollo de software(SDK, por sus siglas en inglés) son todas aquella herramientas que le permiten al programador desarrollar aplicaciones informaticas para una plataforma determinada [33]. Mientras que un middleware es software que se sitúa entre un sistema operativo y las aplicaciones que se ejecutan en él. Básicamente, funciona como una capa de traducción oculta para permitir la comunicación y la administración de datos en aplicaciones distribuidas [34].
- **Capa independiente de la plataforma:** Esta capa aísla las capas dependientes de la plataforma para la que se va a desarrollar el juego, de las capas superiores que son estándares e independientes de la plataforma [32].
- **Subsistemas principales:** Esta capa esta compuesta sub sistemas que vinculan a todas aquellas utilidades o bibliotecas de utilidades que dan soporte al motor de juegos. Tal como:
  - Biblioteca matemática.
  - Estructuras de datos y algoritmos.
  - Gestión de memoria.
  - Depuración y logging [32].
- **Gestor de recursos:** Esta capa es responsable de generar una interfaz de comunicación unificada para acceder a las distintas entidades de software que componen el motor de juego, como por ejemplo las escenas, los sonidos o los objetos de juego [32].
- **Motor de rendering:** Renderizado (render en inglés) es un término usado en computacion para referirse al proceso de generar una imagen foto realista desde un modelo 3D [35]. Esta capa tiene una gran importancia, debido a la naturaleza gráfica del videojuego. El enfoque más utilizado para implementar esta capa es utilizando una arquitectura multi-capa [32].
- **Herramientas de depuración:** Esta capa se encarga de depurar y optimizar el motor de juego para obtener un mejor rendimiento [32].
- **Motor de Física:** Esta capa se encarga de gestionar la detección de colisiones, su determinación y la posterior respuesta que tendrá el juego ante dicha colisión.
- **Interfaces de usuario:** Esta capa tiene como objetivo ofrecer una abstracción de las interacciones del usuario con el juego y de tratar todos los eventos de salida, es decir la retroalimentación que el juego le da al usuario[32].
- **Networking y multijugador:** Esta capa permite que el juego sea capaz de soportar diferentes jugadores de manera simultanea, ya sea que se encuentren de manera local (es decir en una misma plataforma sin conexión a internet) o de manera online (haciendo uso del internet)[32].
- **Subsistema de juego:** Esta capa permite la creación de las mecánicas de juegos; es decir es capa soporta la implementación de un lenguaje de programación, comúnmente de alto nivel, para definir el comportamiento de todos aquellos elementos que componen el juego, como enemigos, cámaras, obstáculos, etc [32].
- **Audio:** Esta capa proporciona al moto la capacidad de utilizar archivos de audio para garantizar una mejor experiencia al usuario [32].

- **Subsistemas específicos de juego:** En esta capa se implementan todos aquellos módulos que proporcionen una identidad al sistema y por lo tanto son únicos[32].

### Motores gráficos existentes en el mercado.

En este apartado se mencionaran los principales motores de juego que existen en la industria, de igual manera se hará mención de sus principales características.

- **Unity3D:** Actualmente Unity es el motor grafico más utilizado en la industria.
  - **Sistema operativo:** Microsoft ver 10,8, 7(solo 64 bits); MacOs ver X 10.9 en adelante.
  - **CPU:** Soporte para el conjunto de instrucciones SSE2.
  - **GPU:** Tarjeta gráfica con DX9 (modelo de shader 3.0) o DX11 con capacidades de funciones de nivel 9.3.
  - **Memoria RAM:** Depende de la complejidad del proyecto.
  - **Desarrollo para plataforma:** Cross-platform.
  - **Orientado a 2D/3D:** 2D y 3D.
  - **Lenguaje de programación que soporta:** `#C`, javaScript, Boo.
  - **Tipo de Licencia:** Maneja tres tipos de licencia, dos de pago y uno gratuito. [36]
- **UnrealEngine:** Considerado por algunas revistas especialistas en videojuegos como el motor de juego más potente.
  - **Sistema operativo:** Microsoft ver 10,8, 7(solo 64 bits); macOS 10.13 High Sierra y Ubuntu 15.04.
  - **CPU:** SQuad-core Intel or AMD, 2.5 GHz or faster (Para Windows), Quad-core Intel, 2.5 GHz or faster(Para Mac y linux).
  - **Tarjeta de vídeo:** DirectX 11 compatible graphics card (Para Windows), Metal 1.2 Compatible Graphics Card(Para Mac) y NVIDIA GeForce 470 GTX or higher with latest NVIDIA binary drivers(Linux).
  - **Memoria RAM:** 8GB (Microsoft y Mac) y 16GB (Linux).
  - **Desarrollo para plataforma:** Cross-platform.
  - **Orientado a 2D/3D:** 2D y 3D.
  - **Lenguaje de programación que soporta:** C++.
  - **Tipo de Licencia:** licencia de pago pero se debe de pagar el 5 por ciento de las regalías cuando el juego sea publicado. [37]
- **CryEngine:** Considerado por algunas revistas especialistas en videojuegos como el motor de juego más potente.
  - **Sistema operativo:** Microsoft ver 10,8, 7(solo 64 bits y 32 bits).
  - **CPU:** Intel Dual-Core min 2GHz (Core 2 Duo and above) o AMD Dual-Core min 2GHz (Phenom II X2 and above).

- **Tarjeta de vídeo:** NVIDIA GeForce 450 series o AMD Radeon HD 5750 series or higher (minimum 1 GB dedicated VRAM GDDR5).
- **Memoria RAM:** 4GB.
- **Desarrollo para plataforma:** Cross-platform.
- **Orientado a 2D/3D:** 2D y 3D.
- **Lenguaje de programación que soporta:** C++, #C y Lua.
- **Tipo de Licencia:** Licencia gratuita pero ofrece planes de pago para capacitación. [38]

## Software auxiliar

Además de los motores gráficos el proceso de desarrollo de videojuegos necesita diferentes herramientas auxiliares para la creación de todos aquellos elementos que se necesiten poner dentro del juego, sea personajes, música, fondos, efectos de sonido, etc. A continuación, se mostrará una lista de aplicaciones y páginas web que fungen como herramientas auxiliares en el desarrollo de videojuegos:

- Creación de Sprites (Solo juegos 2D) o texturas.
  - Adobe Photoshop.
    - Descripción: Aplicación de diseño y tratamiento de imágenes. Con esta aplicación se pueden crear ilustraciones e imágenes 3d. Su capacidad de manejo de imágenes secuenciales la hacen de gran ayuda en la generación de imágenes de bloques de animación para los sprites de juegos 2D, así como su compatibilidad con Adobe Illustrator facilitan la vectorización de sprites.
    - Requerimientos mínimos en Windows:
      - ◊ Procesador Intel Core 2 o AMD Athlon 64 processor de 2 GHz.
      - ◊ Sistema operativo Microsoft Windows 7, Windows 8.1, o Windows 10.
      - ◊ 2 GB de RAM.
      - ◊ Espacio de 2.6 GB en el disco duro para instalacion en 32 bits; o 3.1 GB para sistemas de 64 bits.
      - ◊ Pantalla de 1024 x 768 con 16-bit de color y 512 MB de VRAM [39].
  - Adobe Ilustrator.
    - Descripción: Esta aplicación de gráficos vectoriales permite crear logotipos, íconos, dibujos, tipografías e ilustraciones para ediciones impresas, la web, vídeos y dispositivos móviles. Su sistema de vectorización de imágenes permite crear sprites de mejor calidad. Es una buena herramienta para la creación de botones o iconos para la GUI de juegos.
    - Requerimientos mínimos en Windows:
      - ◊ Procesador Intel Pentium 4 or AMD Athlon 64 processor
      - ◊ Sistema operativo Microsoft Windows 7, Windows 8.1, o Windows 10
      - ◊ 1 GB de RAM para 32 bits; 2 GB de RAM para 64 bit
      - ◊ 2 GB libres en el disco duro.
      - ◊ Pantalla de 1024 x 768, 1GB de VRAM [40].

- AutoDesk SketchBook.
  - Descripción: Herramienta de diseño, más orientada hacia artistas que hacia diseñadores. Es una herramienta de gran utilidad en la creación de arte conceptual para el juego y el diseño de personajes. También posee una herramienta que permite la creación de imágenes secuenciales para bloques de animación. Tiene una total compatibilidad con Adobe Photoshop, por lo que se pueden exportar proyectos desde AutoDesk SketchBook sin el temor de perder detalles de diseño. Su principal ventaja es que se encuentra disponible para dispositivos móviles (Android e IOS) y computadoras (Windows y MAC), cuenta con tres tipos de licencias: la gratuita (tiene funcionalidad limitada), la de pago (por un único pago se cuenta con varias herramientas de diseño) y la pro (Suscrición mensual que ofrece la total funcionalidad de la aplicación y permite utilizar toda funcionalidad tanto en dispositivos móviles como en computadoras ).
  - Requerimientos mínimos en Windows:
    - ◊ Sistema operativo Windows 7 SP1 (32 bit, 64 bit), Windows 8/8.1 (32 bit, 64 bit), o Windows 10.
    - ◊ Procesador de 1 GHz Intel o AMD CPU.
    - ◊ 1GB de Memoria.
    - ◊ 256 MB de tarjeta gráfica con soporte de OpenGL 2.0[41].
- Modelos 3D y animación 3D.
  - Blender.
    - Descripción: Aplicación de modelado y animación 3D de licencia libre. Se encuentra disponible para Windows, Linux y macOS. Blender permite la exportación de modelos, paquetes de animación y escenarios enteros a motores gráficos como Unity3D.
    - Requerimientos mínimos:
      - ◊ CPU de 32-bit dual core
      - ◊ 2Ghz con soporte a SSE2.
      - ◊ 2 GB de memoria RAM.
      - ◊ Pantalla de 24 bits 1280 x 768.
      - ◊ OpenGL 2.1 Compatible con gráficos y con 512 MB RAM [42].
  - Maya.
    - Descripción: Es un software de renderización, simulación, modelado y animación 3D. Maya ofrece un conjunto de herramientas integrado y potente, que puede usar para crear animaciones, entornos, gráficos de movimiento, realidad virtual y personajes. Se encuentra disponible para Windows, Linux y macOS.
    - Requerimientos mínimos:
      - ◊ Procesador de varios núcleos de 64 bits Intel o AMD con el conjunto de instrucciones SSE4.2.
      - ◊ 8 GB de RAM.
      - ◊ 4 GB de espacio libre en disco para la instalación [43].

- Edición y creación de sonido.
  - Ardour
    - Descripción: Software que permite grabar, editar y mezclar audio. Su público objetivo son ingenieros de audio, compositores, músicos y editores de soundtracks. Se encuentra disponible para Mac, Windows y Linux. Posee soporte para plug-ins.
    - Requerimientos mínimos para Linux:
      - ◊ Cualquier procesador de 32 o 64 bits Intel.
      - ◊ Cualquier distribución de Linux con un kernel más actual al 2.3 y libc versión 2.25
      - ◊ 2GB de RAM.
      - ◊ Espacio mínimo de 350MB en el disco duro [44].

### 3.3. Cultura.

En esta sección se hablará de la cultura, primeramente definiéndola, para después mencionar sus principales características, los tipos de cultura que hay, finalizando con su importancia de la cultura.

#### 3.3.1. ¿Qué es la cultura?

La Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (UNESCO) define la cultura como “el conjunto de los rasgos distintivos, espirituales y materiales, intelectuales y afectivos que caracterizan a una sociedad o un grupo social. La cultura engloba, además de las artes y las letras, los modos de vida, los derechos fundamentales al ser humano, los sistemas de valores, las tradiciones y las creencias; de igual forma la cultura da al hombre la capacidad de reflexionar sobre sí mismo. Es ella la que hace de los seres específicamente humanos, racionales, críticos y éticamente comprometidos. A través de ella discriernen los valores y se efectúan opciones. Es por medio de ella que el hombre se expresa, toma conciencia de sí mismo, se reconoce como un proyecto inacabado, pone en cuestión sus propias realizaciones, busca incansablemente nuevas significaciones, y crea obras que lo trascienden[45]”. Bajo esta definición se puede entender a la cultura como un construcción humana que le da identidad a los individuos.

#### 3.3.2. Características de la cultura

Con base en Puja Modal la cultura está compuesta de las siguientes características[46]

- **La cultura es adquirida:** La cultura se aprende no es una característica biológica inherente al nacer.
- **La cultura es social:** La cultura se adquiere como producto de las interacciones humanas. Sin una sociedad no puede existir una cultura.
- **La cultura es transmisiva:** La cultura se transmite entre individuos, tiene un flujo dinámico y nunca permanece constante.

- **La cultura llena necesidades:** La cultura puede llenar diferentes necesidades humanas como la moral, la solidaridad y la convivencia.
- **La cultura es compartida:** La cultura no es una posesión de un solo individuo sino es algo que comparte una gran mayoría de una población en un espacio determinado.
- **La cultura es idealista:** La cultura conglomerá las ideas, valores y normas del grupo dominante de una sociedad y los maneja como si todos tuvieran los mismo valores e ideas.
- **La cultura es acumulativa:** La cultura no se crea en cortos periodos de tiempo sino es la suma de los ideales, creencias y normas que generacionalmente se van creando.
- **La cultura es adaptable:** La cultura se adapta a diferentes cambios y se modifica.
- **La cultura es variable:** No es absoluta, cada sociedad tiene su propia cultura.
- **La cultura es organizada:** La cultura esta organizada por diferentes conjuntos de culturas; estos se unen de manera ordenada para formar un todo, la cultura de la sociedad.
- **La cultura es comunicativa:** La cultura se basa en símbolos y en como estos símbolos son comunicados entre los individuos de una sociedad.

### 3.3.3. Clasificación de la cultura.

Existen diferentes tipos de clasificación de la cultura, sin embargo en esta sección se presentará la clasificación por definición, ya que esta clasificación es la que aborda el tipo de cultura sobre la que trabaja el presente trabajo terminal. A continuación se presentan los tipos de cultura según la clasificación por definición:

- **Tópica:** Esta clasificación consiste en una lista de tópicos o categorías, tales como organización social, religión, seguridad, empleo, economía, etc [47].
- **Histórica:** Esta clasificación hace referencia a la herencia social. Es la relación que tiene la sociedad con su pasado [48].
- **Cultura mental:** Este tipo de cultura engloba todos aquellos hábitos o costumbres que diferencian a un individuo o un conjunto de individuos del resto. Este tipo de cultura se puede entender como la idiosincrasia de una población [47].
- **Cultura estructural:** Es el conjunto de símbolos, valores, creencias y conductas reglamentadas y relacionados entre sí[48].
- **Cultura simbólica:** Conforma todas aquellas reglas, canales modos de comunicación que existen entre los individuos de una sociedad[47].

### 3.3.4. Importancia de la cultura.

La cultura es importante ya que ella:

- Determina la estructura del pensamiento, lo que influye en las percepciones, los valores y el comportamiento [49].

- Permite la construcción de piezas artísticas e históricas que sirven como testimonio del pasado [50].
- Da unidad y sentido de pertenencia[51].
- Permite la convivencia entre individuos [49].
- Regula el comportamiento humano[49].
- Permiten el crecimiento y recreación del individuo [50].

## 3.4. Gamificación

### 3.4.1. ¿Qué es la gamificación?

La gamificación es comúnmente definida como el uso de elementos del juego y el pensamiento basado en juego en entornos no relacionados con el juego para aumentar el compromiso o modificar el comportamiento[52]. Otra definición para gamificación es la empleada en el libro Gamification by Desing, en la cual la gamificación es “un proceso relacionado con el pensamiento del jugador y las técnicas de juego para atraer a los usuarios y resolver problemas”[53].

Es importante comprender que en primera instancia el principal objetivo de la gamificación es influir en el comportamiento de las personas al producir un sentimiento de dominio y autonomía.

Actualmente muchas empresas utilizan principios de gamificación en sus sitios web o en sus aplicaciones móviles para mejorar la experiencia de sus usuarios y conseguir que estos pasen más tiempo en sus aplicaciones o sitios web. Un ejemplo del uso de la gamificación en una aplicación móvil es la aplicación de Duolingo, en esta aplicación el jugador pude aprender diferentes idiomas a partir de dinamicas que involucran la escritura, lectura y audición del idioma.

### 3.4.2. Elementos del videojuego en la gamificación.

Tal como lo dice su definición la gamificación toma elementos del videojuego para cumplir sus objetivos, en este apartado se detallan tres elementos de la gamificación que son tomados del videojuego:

- **Dinámicas:** Dentro de los videojuegos las dinámicas hacen referencia a la interacción de los objetivos y acciones del jugador con las reglas del juego, es decir la jugabilidad [54]; un ejemplo de dinámicas puede ser un sistema de diálogos en el cual el jugador debe de obtener información para resolver diferentes acertijos.
- **Mecánicas:** Las mecánicas son las reglas del juego, dependen directamente de las dinámicas[54]. Retomando el ejemplo anterior del sistema de diálogos, sus respectivas mecánicas serían que se tenga un tiempo limite para elegir las respuestas del dialogo, que se disponga de cuatro opciones de respuestas a cada dialogo, etc.
- **Componentes:** Son los últimos elementos en ser elegidos y tienen que ser coherentes con las dinámicas y las mecánicas o de lo contrario la aplicación no tendrá cohesión. Un ejemplo de elementos: avatares, logros, insignias, batallas contra jefes, colecciónables, contenido desbloqueable, niveles, puntos, equipos, tablas de posiciones, misiones, etc [52].

### 3.4.3. Tipos de jugadores.

Para diseñar correctamente una experiencia con gamificación es necesario delimitar quienes van a usar el producto, que habilidades tienes, etc. El sociólogo Richard Bartle desarrollo cuatro tipos de perfiles de jugadores para describir los jugadores del juego World of Warcraft; por su escalabilidad esta segmentación de jugadores ha sido ampliamente utilizada para clasificar a los jugadores sin importar el juego.

La clasificación de Bartle se basa en un modelo de cuatro jugadores puestos sobre dos ejes:

- El eje horizontal que describe si el jugador esta más enfocado en el mundo del juego o en otros jugadores.
- El eje vertical que representa la afinidad del jugador por actuar o por interactuar.

De estos dos ejes se generan cuatro tipos de jugador:

- **Triunfador:** Son todos aquellos jugadores que prestan mayor atención a su estatus y sus puntos dentro del juego. Les gusta mostrar su progreso a sus amigos. Pueden ser motivados por medallas, por incrementos de nivel (ya sea de su personaje o sus armas) dentro del juego. Se calcula que al menos el 10 % de los jugadores son del tipo Triunfador.
- **Explorador:** Los jugadores que son de este tipo desean explorar todo el mapa y descubrir todos los secretos posibles. Para los exploradores el descubrimiento es la recompensa y su mayor motivación es lo que esta oculto en el mundo del videojuego y no alardear con sus amigos sobre sus descubrimientos. Al igual que el tipo triunfador, el 10 % de los jugadores son de este tipos.
- **Socializador:** Este tipo de jugador encuentra su motivación en la interacción con otros jugadores dentro del juego. El jugador socializador preferirá un juego que le permita colaborar con otros jugadores para superar retos como equipo. La mayoría de los jugadores pertenecen a este tipo.
- **Asesino:** Los jugadores de tipo asesino son parecidos a los de tipo triunfador: disfrutan de obtener la victoria y de progresar dentro del juego; sin embargo lo que los diferencia del tipo triunfador es que ellos disfrutan más de ver a otros perder. Buscan ser los mejores en el juego y para ello deben de derrotar a todos los demás, siendo esa su principal motivación. Menos del 1 % de los jugadores pertenece a este tipo de jugadores [55].

### 3.4.4. Intervalos de actividades de la gamificación.

Lograr el compromiso hacia un comportamiento o idea es el principal objetivo de la gamificación y para conseguir dicho objetivo son necesarios dos tipos de bucles en el diseño de la aplicación:

- Bucles de compromiso.
- Bucles de inicio y progresión.

En los siguientes apartados se profundizara en dichos bucles con el fin de que dichos conceptos queden claros al lector.

### Bucles de compromiso

El bucle de compromiso reafirma el compromiso del jugador para que continúe mostrando el comportamiento que se desea crear o modificar, este tipo de bucle ata al jugador al juego una vez que ha entrado. En el bucle de compromiso se encuentran los siguientes tres elementos:

- Motivación.
- Acción.
- Retroalimentación[52].

Estos tres elementos se deben de encontrar siempre presentes, por lo que es importante que se repitan de manera indefinida. Estos tres elementos deben de ser ejecutados en el orden en el que se mencionan ya que si no hay motivación es imposible que el jugador pueda pasar a la acción y después de la acción debe de existir la retroalimentación o de lo contrario la motivación no se podrá mantener. La implementación de estos elementos depende directamente del tipo de jugador objetivo. Por ejemplo para un jugador de tipo explorador su motivación será descubrir los secretos del mundo del juego, por lo que sus acción será explorar el mundo del juego y su retroalimentación será descubrir un secreto dentro del juego.

### Bucle de inicio y progresión

Este tipo de bucle tiene como objetivo asegurarse que el jugador pueda progresar. Para lograr este el sentimiento de progreso en el jugador se necesitan dos tipos de bucles:

- **Bucle de inicio:** Este bucle se presenta la primera vez que el jugador juega por primera vez. Este bucle permite al jugador familiarizarse con todas las mecánicas y conceptos del juego.
- **Bucle de progresión:** En este bucle se presentan retos que superan las habilidades iniciales del jugador por lo que éste deberá de mejorar para poder progresar. Este reto tiene como objetivo medir cuánto el jugador ha evolucionado. Un ejemplo de este bucle dentro de un juego es un enemigo jefe, en un principio el jugador no podrá derrotarlo pero después de un tiempo de luchar contra él, el jugador podrá derrotarlo dejando una sensación de progresión en este [52].

### 3.4.5. Tipos de diversión.

La diversión es uno de los elementos principales sobre los que se apoya la gamificación para generar experiencias de usuario únicas. En trabajo “Why we play games: Four keys to more emotion without story” se propone la existencia de cuatro tipos de diversión que los videojuegos utilizan para generar experiencias de juego que logran enganchar a sus jugadores:

- **Diversión difícil.** Este tipo de diversión es la que se da cuando el jugador logra superar retos dentro del juego. Los jugadores disfrutan de este tipo de diversión cuando pueden ver que tan buenos son en el juego, superan múltiples objetivos o requieren más estrategia para ganar en lugar de suerte. Las emociones que despierta este tipo de diversión es de éxito o frustración[56].

- **Diversión fácil.** Este tipo de diversión viene de descubrir el mundo del juego. Para lograr esto el juego se vale de elementos que despierten la curiosidad del jugador. El jugador disfruta de este tipo de diversión cuando explora nuevos mundos, se siente uno con el personaje, logra descubrir misterios del mundo o de la narrativa del juego[56].
- **Diversión seria (estados alterados).** Esta diversión se vale de como la partida puede afectar el estado de ánimo del jugador. Esta diversión se logra cuando: El jugador evita el aburrimiento, se siente mejor consigo mismo, aclara su mente al completar un nivel. En este tipo de diversión el juego pasa a ser un factor de desahogo personal[56].
- **El factor persona.** Con este tipo de diversión el juego se convierte en un medio de convivencia con otras personas en donde la interacción con otros jugadores es el máximo factor de disfrute y se encuentra por sobre el juego[56].

Dado que la gamificación toma elementos del videojuego y los pone en entornos ajenos a éste, estos cuatro tipos de diversión también pueden ser empleados en el diseño de experiencias de gamificación.

#### 3.4.6. Framework de diseño de la Gamificación

A continuación se presenta el framework de diseño creado por el profesor Kevin Werbach de la Universidad de Pensilvania. El framework del profesor Werbach está compuesto por seis pasos:

- **Definir el objetivo del negocio.** Este paso consiste en definir el objetivo que persigue la gamificación, no se debe de confundir con el comportamiento objetivo. En este punto de contesta la pregunta ¿Cómo beneficiará la implementación de la gamificación al proyecto?
- **Definir el comportamiento objetivo.** Aquí se define el comportamiento que se quiere modificar o inculcar al usuario para que se cumpla el objetivo del negocio. Este comportamiento debe quedar definido en cualidades cuantificables que permitan su verificación.
- **Describir a los jugadores.** Una aplicación exitosa no se enfoca en un solo tipo de jugador sino logra incorporar dinámicas atractivas para los cuatro tipos de jugadores.
- **Diseñar los intervalos de actividades.** En este paso se intercalan los lapsos de compromiso de la aplicación y los lapsos de progresión del comportamiento. Es importante que se realice una buena implementación de dichos intervalos, pues de esto depende que el usuario utilice o no la aplicación por largos períodos de tiempo.
- **No olvidar la diversión[52].**

# **Capítulo 4**

## **Alcance del proyecto**

### **4.1. Objetivos del proyecto**

En esta sección se habla de los objetivos, tanto generales como específicos, que persigue el presente trabajo terminal.

#### **4.1.1. Objetivos generales**

- Fomentar la cultura Mexica entre jóvenes mayores de 13 años a través de un videojuego.

#### **4.1.2. Objetivos específicos**

- Realizar una investigación sobre la cultura Mexica.
- Diseñar un videojuego con bases históricas y mitológicas.
- Diseñar e implementar una narrativa que permita la difusión de la cultura Mexica.
- Comprender el funcionamiento del motor de juego elegido.
- Entender el funcionamiento de un juego de plataforma básico.

### **4.2. Alcance del proyecto**

El presente trabajo terminal tendrá:

- Funcionalidad de un solo usuario.
- Contener diez niveles, uno introductorio y nueve situados en el inframundo Mexica.
- Contar con sprites originales.
- Contar con un sistema de guardado, para salvar el progreso del jugador.
- Contener cinematicas que cuentan una historia original.
- Funcionar en dispositivos Android con los requerimientos expuestos en la sección 4.4.

- Contener un nivel que permite repetir los niveles ya completados.

El presente trabajo terminal no realizará:

- Enseñar historia.
- Realizar microtransacciones.
- Soportar múltiples jugadores.
- Contar con música original, creada especialmente para el juego.

### 4.3. Metodología de trabajo

La metodología de trabajo elegida es Huddle. Como se mencionó en el apartado 3.2.2, Huddle es una metodología orientada a videojuegos y una de sus principales bondades que ofrece la naturaleza iterativa de Scrum con el agregado de cubrir la línea de producción de un videojuego (ver apartado 3.2.1).

El principal motivo por el que se eligió huddle, fue que es una metodología orientada a videojuegos; por lo que su documentación y sistema de trabajo cubre las necesidades de un proyecto de esta naturaleza y no es necesario hacer adaptaciones drásticas de la metodología tal como se tendrían que hacer si se hubiera elegido alguna de las metodologías orientadas a desarrollo de software como hubiera sido Scrum o programación extrema.

Para consultar el cronograma de actividades del Trabajo Terminal, consultar el anexo ??.

### 4.4. Especificaciones de plataforma

En esta sección se listarán todos aquellos dispositivos de hardware y licencias de software que se necesitan para el desarrollo del videojuego.

#### 4.4.1. Hardware requerido

En esta sección se mencionan los dispositivos de hardware empleados en el desarrollo del sistema y los dispositivos de prueba del juegos. Estos dispositivos son con los que se contaban a la hora de iniciar el Trabajo Terminal y no son sustituibles por motivos de presupuesto.

##### Computadoras para desarrollo

- Computadora DELL Inspiron 15.
  - Procesador Intel Core i3-4005U.
  - CPU de 1.70 GHz de 64 bits.
  - Memoria ram de 8GB.
- Lenovo G40.

- Intel Core i3 4005U CPU 1.7 KHz de 64 bits.
- Memoria ram de 8GB.
- Tarjeta gráfica AMD Radeon R5 235 de 1GB

### Dispositivos móviles de prueba

- Dispositivo de prueba 1:
  - Versión 5.2
  - Modelo Huawei TAG-L13
  - CPU MediaTek MT6753 1,50 GHz
  - IPS TFT 16M colors 720 x 1280 px (5,00) 294 ppi
  - RAM 2GB
- Dispositivo de prueba 2:
  - Versión 7.0
  - Modelo ASUS X008DC
  - CPU MediaTek Quad Core Processor
  - GPU Mali T720
  - RAM 3GB LPDDR3
  - PANEL 5.2-inch HD(1280 x 720) IPS display 75 por ciento screen-to-body ratio 400nits brightness

#### 4.4.2. Software requerido

##### Motor de juego

Como motor de juego se optó por Unity 3D en su versión 5.6.2.f1 como motor de juego en su licencia libre ya que no se cuenta con los fondos necesarios para contratar las versiones de pago. Los motivos por los que se eligió Unity 3D, son los que se presentan a continuación:

- Curva de aprendizaje rápida.
- Comunidad de desarrolladores activa.
- Permite gestionar trabajos 2D y 3D, esto permitirá escalar el juego a 3D a futuro si alguien desea retomar el proyecto.
- Codificación basada en el paradigma de programación orientada a objetos.
- Requerimientos técnicos de instalación dependientes del proyecto por lo que no exige una computadora de gran costo.
- Capacidad de desarrollo en múltiples plataformas, lo que permite la escalabilidad futura del proyecto hacia nuevas plataformas en caso de que alguien desee retomarlo.

A fin de garantizar la generación de los archivos apk de juego fue necesaria la instalación de Android Studio versión 2.3.3 y java en su versión 8u111.

### Creación de sprites

Para la creación de sprites se eligieron dos softwares Corel Draw X5 y Adobe Photoshop. El primero se eligió para la vectorización de los sprites, ya que es de fácil uso, no requiere tantos recursos como Adobe Ilustrator y permite importar archivos a Adobe Photoshop para su posterior coloreado.

Tal como se mencionó en el párrafo anterior, el objetivo de Adobe Photoshop dentro de este trabajo terminal es colorear los sprites. El motivo para emplear Adobe Photoshop es que permite la edición de imágenes y su optimizado para hacer sprites que requieran menores tiempos de renderizado y menor espacio de almacenamiento sin sacrificar significativamente la calidad de la imagen.

### Interfaz gráfica de usuario

Para los iconos de los botones que controlan al usuario, se descargó una colección de botones del sitio web pixelsticky, este sitio web permite la descarga y utilización de diferentes iconos bajo la licencia de CC0 o de dominio público.

## 4.5. Productos esperados

Los productos esperados se dividirán en dos, siendo los primeros los que se entregarán al término de Trabajo Terminal 1 y los segundos los que se entregarán al finalizar Trabajo Terminal 2.

- Productos esperados al finalizar TT1:
  - Documento de diseño.
  - Guión literario del juego.
  - Storyboard del juego.
  - Documentación de la propuesta de diseño del juego.
  - Maquetado de los niveles 1, 2, 3, 4.
  - Niveles 1 y 2 terminado.
  - Reporte técnico.
- Productos esperados al finalizar TT2:
  - Documentación del juego actualizada.
  - Maquetado de los niveles 5, 6, 7, 8, 9 y 10.
  - Nivel 3, 4, 5, 6, 7, 8, 9 y 10 terminados.
  - Reporte técnico.

# Capítulo 5

## Trabajo realizado

En este capítulo se habla sobre el trabajo realizado durante la etapa del Trabajo terminal 1 y trabajo terminal 2. El contenido de este capítulo se estructura con base en los sprints de la metodología huddle.

### 5.1. Etapa de Preproducción

En esta sección se hablarán a manera de resumen de la etapa de preproducción desarrollada durante Trabajo Terminal 1.

#### 5.1.1. Primer Sprint Huddle de Preproducción

Antes de iniciar el diseño del juego se realizó un trabajo de investigación sobre la cultura azteca. Esta investigación abarcó:

- **La sociedad Mexica:** su historia tradiciones y clases sociales.
- **Mitología mexica:** Dioses, mito de los cinco soles, mito de la creación del hombre del maíz, el Mictlán.
- **Historia de la Malinche:** Historia del personaje antes y después de la llegada de los españoles.

Durante la etapa de investigación se seleccionó la información histórica que sería relevante y útil para la narrativa del juego y el diseño de su jugabilidad. Para la investigación histórica de esta etapa se consultaron libros, códices, páginas de internet, artículos de investigación e incluso se visitaron museos como el templo mayor.

#### 5.1.2. Segundo Sprint Huddle de Preproducción

En esta etapa se redactaron las primeras secciones del documento de diseño del juego Yolotl. Primeramente, se inició con la idea concepto del juego. Para algunos juegos la mecánica es la primera es ser definida; no obstante, por la naturaleza del juego como herramienta de transmisión de cultura, Yolotl nació con su historia. La historia de Yolotl pasó por diferentes etapas de diseño en el que se vio modificada, pero manteniendo algunos elementos y agregando otros.

En la etapa del concepto también se definieron la plataforma para la que sería el juego: dispositivos móviles con sistema operativo Android 5.2; mientras que la plataforma de desarrollo sería el motor de juego Unity3D. Paralelamente a la preproducción se inicia el desarrollo de un primer demo que permita familiarizarse con la herramienta de Unity3D.

Una vez teniendo la idea concepto se definió la visión del juego y sus mecánicas. En cuestión de las mecánicas el enfoque por el que se optó fue el de mantener el juego con mecánicas simples y familiares para aquellos jugadores que ya habían tenido alguna experiencia con algún juego de plataformas, sin descartar algunos detalles que le dieran identidad al juego en cuanto a su jugabilidad.

Con la historia, la visión y la mecánica definidas se procedió a definir los estados del juego, diseñar las interfaces gráficas de navegación y de interacción con el personaje. Para ver las interfaces ver anexo 8.1.

### **5.1.3. Tercer Sprint Huddle de Preproducción**

En el tercer Sprint se definieron la cantidad de niveles y en qué consistiría cada uno, de igual forma se definieron los objetivos de cada nivel, la recompensa a obtener una vez, los enemigos a vencer, las cinemáticas que fungirían como transiciones entre nivel.

Al mismo tiempo que se definieron los niveles se definieron y detallaron los personajes tanto a nivel narrativo como a nivel de jugabilidad, definiendo habilidades para los enemigos, los niveles en los que parecerían y sus acciones dentro de la historia. Para esta parte se trató de obtener la mayor fidelidad posible a los mitos y códices. Si se desea profundizar en los diseños de personajes ver el anexo 8.2.

### **5.1.4. Cuarto Sprint Huddle de Preproducción**

En el cuarto sprint se terminó de definir el argumento del juego al escribir el guión de la historia. En este sprint también se definieron elementos de ambientación para el juego tales como la música de fondo, los efectos de sonido, los efectos de sonido.

De igual forma, en este sprint se definieron las armas de los personajes, los ítems; estos elementos se definieron tanto a nivel de comportamiento como a nivel visual. Al igual que con los personajes se buscó que las armas, tanto en comportamiento como en diseño, se mantuvieran lo más fiel posible a los mitos y leyendas de donde se basaron.

Con el cuarto sprint se finalizó la etapa de preproducción, obteniendo así un documento de diseño lo suficientemente detallado como para iniciar el diseño del juego a nivel de ingeniera.

## **5.2. Etapa de producción**

En esta sección se hablará del trabajo realizado durante la etapa de producción del juego, en esta sección los cuatro primeros sprints serán presentados a manera de resumen, ya que fueron

desarrollados durante los meses correspondientes al trabajo terminal 1. Todos los sprints se mencionen después del cuarto sprint se tratarán a detalle, ya que corresponden al periodo de desarrollo de Trabajo Terminal 2.

### 5.2.1. Primer Sprint Huddle de Producción.

En este sprint se realizó un análisis del documento de diseño y se definieron las clases y el modelo bajo el que funcionaría el juego a nivel de programación. Haciendo uso del paradigma orientado a objetos se propone emplear tres tipos de clases:

- **Actores:** Son las clases que modelan a los enemigos, los ítems, los coleccionables, los check-points y al jugador.
- **Controladores:** Son las clases encargadas de gestionar la partida y la navegación entre interfaces. Estas clases desencadenan eventos conforme a las acciones de las clases actoras. Estas clases también son las encargadas de verificar que se cumplan las reglas de los niveles.
- **Auxiliares:** Estas clases ayudan al funcionamiento de los actores y los controladores. Estas clases también se encargan de vincular datos con las clases controladoras como efectos de sonido, música, datos para la progresión entre niveles.

El modelo planteado permitió reutilizar parte del demo generado durante la etapa de preproducción.

En el primer Sprint de Producción también se crean los sprites del primer nivel utilizando la herramienta de modelado en 3D Blender.

### 5.2.2. Segundo Sprint Huddle de Producción.

En este sprint se maqueta la sección del mercado del primer nivel del juego. Y se procede a trabajar en los sprites. Durante este sprint se inicia la integración del código del primer prototipo con el nuevo modelo del juego. Al finalizar este sprint se determina la no viabilidad del modelado en 3D de los sprites por cuestiones de tiempos; en consecuencia, se descarta este método para generar los sprites y se inicia el desarrollo de los sprites a partir de otras técnicas de animación más tradicionales.

### 5.2.3. Tercer Sprint Huddle de Producción.

En este sprint se inicia el desarrollo de los sprites con Adobe Photoshop y Corel Draw. A la par se inicia la maquetación de la etapa de selva del nivel uno. Una vez terminados los sprites referentes al nivel uno estos se integran al código permitiendo tener un segundo prototipo con la siguiente funcionalidad:

- Control de personaje por medio de la GUI.
- Transiciones entre interfaces.
- Personaje seguible que aparece en el primer nivel funcional.
- Funcionamiento básico del controlador de diálogos.

### 5.2.4. Cuarto Sprint Huddle de Producción.

Durante el cuarto sprint se desarrollaron los sprites referentes a los obtaculos de plataformas, objetos de fondo. Procediendo a redactar el reporte técnico referente al trabajo terminal 1 y la preparación de la posterior exposición.

### 5.2.5. Quinto Sprint Huddle de Producción

En este sprint se trabajaron sobre las observaciones realizadas por los sinodales durante la exposición del Trabajo Terminal 1.

#### Modelo de datos

La primera observación en atender fue el modelo de datos del juego, dicho modelo de datos se realizó utilizando un modelo entidad relación de base de datos (Ver Anexo 8.3) ya que al modelarse de esta forma hace escalable el juego si se deseará en algún futuro emplear una base de datos para mejorar el almacenamiento de datos y el manejo de más usuarios para ofrecer un modo online. El modelo de datos está basado en el modelo de clases y contiene únicamente a las clases actoras. Toda entidad actora se define como una especialización de una entidad base llamada GameObject, esta entidad está definida por como su identificador y por otras entidades como GameObjectPosition, Level, Tag, AnimationMachine, entre otros.

#### Control de adicción en el jugador

La segunda observación sobre la que se trabajo fue como disminuir la adicción del jugador al videojuego Yolotl. Esta observación dio lugar a una investigación sobre la adicción a los videojuegos ya que antes de proponer alguna solución se debía conocer cómo se definía, las causas y las consecuencias de la adicción al videojuego. Al final de la investigación se pudieron formular tres posibles soluciones para evitar la adicción del jugador; sin embargo, dado que este tópico no estaba en la planeación original del proyecto y por las implicaciones que conllevaban cada una de las soluciones se decidió únicamente describir las soluciones y sus implicaciones sin desarrollar ninguna de las tres. A continuación, se describen a manera de resumen las soluciones (nuevamente si se dese a profundizar en la investigación realizada y las soluciones se puede consultar el Anexo 8.4):

- **Notificación de confirmación para continuar la partida.** Esta solución propone que el juego solicite la confirmación del usuario para continuar una vez que éste ha detectado que el jugador ha estado jugando durante un tiempo prolongado como una hora.
- **Control paterno.** El juego le envía un formulario al tutor del jugador por medio de un correo electrónico. En este formulario el tutor podrá decidir cuanto tiempo al día la aplicación podrá estar abierta.
- **Sistema de vidas.** El jugador tiene una cantidad de vidas limitadas. Cada vez que el jugador ingresa a un nivel o muere dentro de uno y reinicia la partida se gasta una vida. Para recuperar vidas el jugador deberá de esperar un determinado tiempo.

## Modelo de negocios

Una vez terminados los puntos anteriores se procedió a diseñar el posible plan de negocios que se emplearía para el desarrollo y comercialización del juego. (Falta integrar el plan de negocios)

### 5.2.6. Sexto Sprint Huddle de Producción

En este sprint se definieron las nuevas estrategias a seguir para agilizar y optimizar el desarrollo del juego y el desarrollo de los *sprites* faltantes del juego.

#### Las nuevas estrategias de desarrollo

Para el sexto *sprint* y teniendo como base la experiencia de desarrollo los anteriores prototipos, quedaba claro que se necesitaba diseñar una nueva estrategia que permitiera agilizar el desarrollo del juego sin comprometer la calidad del mismo. La naturaleza iterativa de *Huddle* permite controlar la extensión del desarrollo y la asignación de tareas entre miembros del equipo, pero al estar orientada a niveles producía que la programación se tuviera que pausar cada vez que se iniciaba un nuevo nivel para diseñar la maqueta del nivel y para el desarrollo de los *sprites*. Por tal motivo se propuso un nuevo enfoque a *Huddle* orientar su naturaleza iterativa al desarrollo de los componentes del juego, una vez que estos estén completados y probado su funcionamiento de manera individual, se inicia la construcción de los niveles. Bajo este nuevo enfoque primeramente se crean las maquetas de todos los niveles, seguido de la creación de todos los *sprites*, para después programar los assets que correspondían a los actores, después se implementan las clases controladoras que comparten todos los niveles, como el controlador del personaje, el controlador de la barra de vida, etc.; con los controladores comunes funcionando se crea una escena base que incluye el personaje jugable y la interfaz de juego, sobre esta escena se construyen los niveles del juego. El anterior enfoque es una pequeña abstracción del Método *Centry* descrito en la sección 3.2.2.

Ya definido el nuevo plan de trabajo, se procedió a diseñar las maquetas para los niveles. Para el diseño de maquetas se siguió utilizando la plantilla creada durante Trabajo Terminal 1. Si se desea ver las maquetas de los niveles, estas se encuentran en el anexo 8.5.

#### Actualizando el *software* de desarrollo

Paralelamente al diseño de las maquetas de los niveles, fue liberada la versión 2017.3.1f de *Unity3D*. Esta versión incluye herramientas que agilizan la creación de niveles como el uso de:

- **Tilemap:** Herramienta para el mapeado de niveles. Esta herramienta facilita la creación de mapas al crear una malla sobre la que se arrastran diferentes *Sprites* que se hayan importado previamente al tilemap (ver figura 5.1). En la sección () se profundizará su funcionamiento.
- **Cinemachine:** *Asset* que permite controlar la cámara de la escena, con este *asset* se le puede indicar que objeto se desea que la cámara siga y se puede asignar un área que limitara el movimiento de la cámara (ver figura 5.3). *Cinemachine* se descarga directamente desde la tienda de *assets* de *Unity* y fue desarrollado por los ingenieros de *Unity*, lo que significa que no genera conflictos o no requiere de configuraciones extras al proyecto para importar. En la sección () se profundizará su funcionamiento.

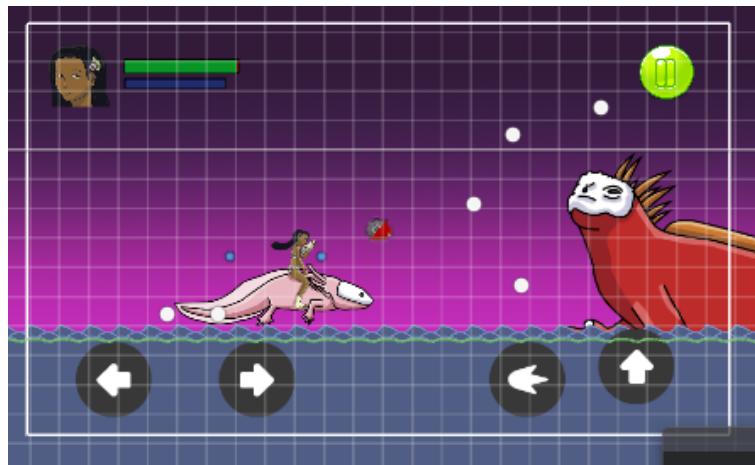


Figura 5.1: Vista de la escena cuando se tiene un *GameObject* de tipo *Tilemaps* para la construcción de niveles

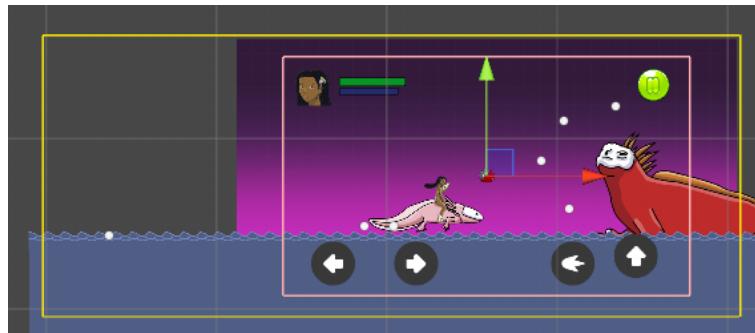


Figura 5.2: Vista de la escena cuando se tiene un *GameObject* de tipo *Tilemaps* para la construcción de niveles

- ***Sprite Packer***: Si bien no es una herramienta para construcción de niveles o un *asset*, esta herramienta es una de las más útiles que se agregó a la nueva versión de *Unity* ya que, como su nombre lo indica, permite el empaquetado de *sprites* (ver figura). Empaquetar los *sprites* es una práctica que optimiza el renderizado de objetos, ya que el controlador de gráficos de *Unity* realiza una sola llamada por paquete cuando renderiza los objetos y con esa única llamada renderiza todos los objetos de la escena que se encuentren en ese paquete; si los *sprites* no se encontraran dentro de un paquete el controlador de gráficos de *Unity* haría una llamada por cada *sprite*.

Por el impacto que tendrían las nuevas herramientas de la versión de *Unity*, se propuso utilizarla en lugar de la versión 5.6.2f1. Antes de actualizar la versión de *Unity* se investigó si el proyecto sufriría algún impacto negativo como falta de compatibilidad de componentes por la diferencia de versiones. Al comprobar que existía una total compatibilidad entre ambas versiones en cuanto a trasladar un proyecto de la versión 5.6.1f a la versión 2017.3.1f. Se determinó que la nueva versión de *Unity* sería la que se emplearía para el resto del desarrollo del juego.

### Creación de los *sprites* faltantes

Lo siguiente a realizarse durante el sexto *sprint* fueron los *sprites*, durante las modificaciones que se definieron en Trabajo Terminal 1 fue la utilización de un *software* de animación en dos

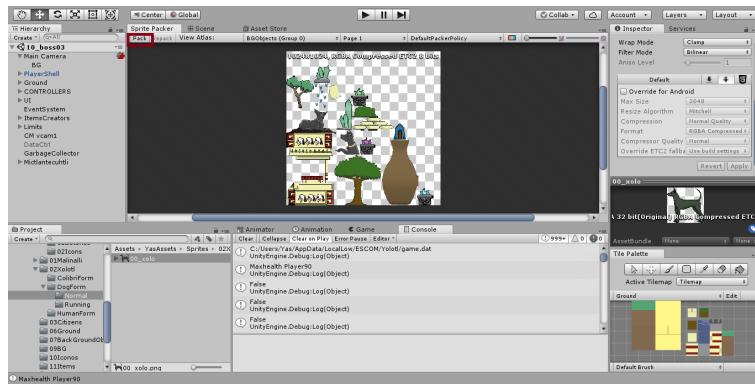


Figura 5.3: Vista de la pestaña del *Sprite Packer*.

dimensiones para generar los *sprites* restantes; sin embargo, el cambio de *software* para generar los *sprites* fue descartado, esto debido a que se adquirió una nueva tableta digitalizadora que agilizó la creación de *sprites*. Para Trabajo Terminal 2 se dibujaron y digitalizaron más de 100 *sprites*. Para mejorar la experiencia visual del jugador se animaron *sprites* que en los primeros demos eran estáticos como es el caso de los fantasmas del segundo nivel de la sección de plataformas (ver figura 5.4).

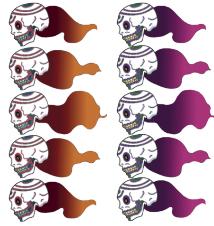


Figura 5.4: Bloques de animación para el enemigo de tipo fantasma.

Otros cambios en cuanto el aspecto visual del juego fue la integración de nuevos *sprites* para el personaje jugable, los nuevos *sprites* incluyen la caracola que *Malinalli* (ver figura 5.5) emplea para atacar y que se obtiene al final del primer nivel de la sección de selva, estos *sprites* para *Malinalli* son utilizados únicamente en los niveles posteriores al primer nivel para darle sentido a la narrativa; para el segundo nivel se hizo algo parecido, los *sprites* del personaje jugable fueron sustituidos por *Malinalli* montando un ajolote (ver figura 5.6), este cambio se hizo para que lo que el jugador vea dentro del nivel sea coherente con la narrativa propuesta y se mejore la inmersión del juego.

En lo que se refiere a los Jefes de cada nivel, no solo se crearon sus respectivos *sprites*, también fue necesario la creación de los *sprites* referentes a sus ataques, para el caso particular de *Mictlantecuhtli* se dibujaron 30 *sprites* tanto para la animación del personaje como para la animación de sus respectivos ataques (ver figura 5.7). Para el diseño de la interfaz gráfica de usuario(*GUI* por sus siglas en inglés) se emplearon *sprites* de las páginas *Kenney.nl* y *Game Art 2D*. Es importante aclarar que la creación de *sprites* pudo haber sido sustituida utilizando paquetes de *sprites* que existen en la red y que son de licencia libre; sin embargo, con la creación de *sprites* propios para el juego

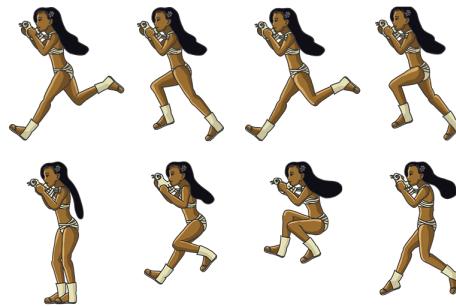


Figura 5.5: Bloques de animación para *Malinalli* posterior a que ella obtiene la caracola.

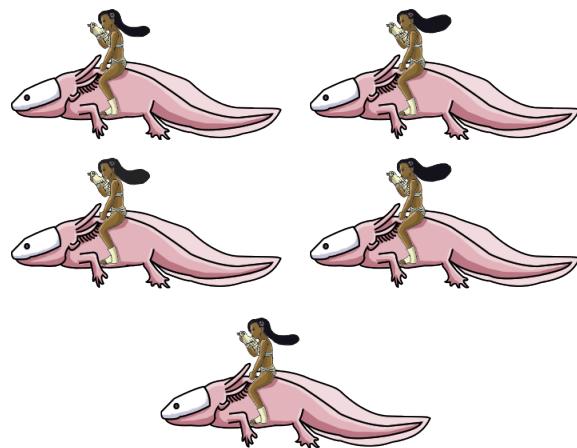


Figura 5.6: Bloques de animación para *Malinalli* montando al ajolote del segundo nivel del juego.

se consigue crearle una identidad visual propia al juego, esto permite que el jugador se identifique con mayor facilidad con el personaje y tenga una mejor asociación con el mundo y la historia que se le presenta dentro del juego [cita]. Si se desea ver a profundidad los sprites que se crearon se puede consultar el anexo 8.2.



Figura 5.7: Bloques de animación para *Mictlantecuhtli*, jefe final del juego.

# **Capítulo 6**

## **Resultados obtenidos**

# **Capítulo 7**

## **Conclusiones**

# Bibliografía

- [1] O. para la Cooperación y el Desarrollo Económico, “Pisa en español.” [Online]. Available: <http://www.oecd.org/pisa/pisaenEspaol.htm>
- [2] E. periódico Sociedad Educación, “Resultados del informe pisa (por países y por comunidades).” [Online]. Available: <http://www.animalpolitico.com/2016/09/reforma-educativa-recorte-presupuesto-2017/>
- [3] J. R. Huerta, “Qué leemos los mexicanos y por qué.” [Online]. Available: <http://contenido.com.mx/2017/03/leemos-los-mexicanos/>
- [4] I. N. de Estadística y Geografía, “Módulo sobre lectura (molec). principales resultados febrero 2018.” [Online]. Available: [http://internet.contenidos.inegi.org.mx/contenidos/productos/prod\\_serv/contenidos/espanol/bvinegi/productos/nueva\\_estruc/promo/resultados\\_molec\\_feb18.pdf](http://internet.contenidos.inegi.org.mx/contenidos/productos/prod_serv/contenidos/espanol/bvinegi/productos/nueva_estruc/promo/resultados_molec_feb18.pdf)
- [5] G. Carricay. ¿qué son los videojuegos? [Online]. Available: <https://medium.com/grupo-carricay/qu%C3%A9-son-los-videojuegos-d640dc6aa84>
- [6] J. M. Pereño, Marketing y videojuegos: Product placement, in-game, advertising y advergaming. ESIC, 2010.
- [7] J. Steuer, “Defining virtual reality: Dimensions determining telepresence,” Journal of Communication, vol. 42, no. 4, pp. 73–93, 1994.
- [8] P. M. Fabrizio Lamberti, Andrea Sanna. Las tecnologías del entretenimiento: Pasado, presente y futuro. [Online]. Available: <https://www.computer.org/web/computingnow/archive/february2015-spanish>
- [9] S. Turkle, La vida en la pantalla: La construcción de la identidad en la era de internet. Paidos Iberica, 1997.
- [10] V. M. Navarro, “Libertad dirigida: Análisis formal del videojuego como sistema, su estructura y su avataridad.” Ph.D. dissertation, Universitat Rovira i Virgili, 2013.
- [11] ESRB. About esrb. [Online]. Available: <http://www.esrb.org/about/>
- [12] M. Alfaro. La esrb y por qué es importante para los videojuegos. [Online]. Available: <http://mx.blastingnews.com/ocio-cultura/2016/02/la-esrb-y-por-que-es-importante-para-los-videojuegos-00770243.html>
- [13] A. Miguez. Pegi ¿cómo se clasifican los videojuegos? [Online]. Available: <https://juegosadn.eleconomista.es/pegi-como-se-clasifican-los-videojuegos-ar-2067/>

- [14] M. Flores. Aprueba senado clasificación de videojuegos en México. [Online]. Available: <http://www.levelup.com/noticias/420095/Aprueba-Senado-clasificacion-de-videojuegos-en-Mexico>
- [15] B. L. Barinaga, Juego. Historia, Teoría y Práctica del Diseño Conceptual de Videojuegos. Alesia, 2010.
- [16] Ó. Pérez Latorre et al., Análisis de la significación del videojuego. Fundamentos teóricos del juego, el mundo narrativo y la enunciación interactiva como perspectivas de estudio del discurso. Universitat Pompeu Fabra, 2010.
- [17] S. Belli and C. López Raventós, “Breve historia de los videojuegos,” Athenaea Digital. Revista de pensamiento e investigación social, no. 14, 2008.
- [18] Newzoo. 2017 global games market report. [Online]. Available: <https://newzoo.com/insights/trend-reports/newzoo-global-games-market-report-2017-light-version/>
- [19] N. Pix. Egs 2016- la actualidad del marco de videojuegos en México. [Online]. Available: <https://nacionpix.com/2016/10/04/egs-2016-mercado-de-videojuegos-en-mexico/>
- [20] IGN. The game production pipeline: concept to completion. [Online]. Available: <http://www.ign.com/articles/2006/03/16/the-game-production-pipeline-concept-to-completion?page=1>
- [21] B. A. Rafael Menéndez. Metodologías de desarrollo de software. [Online]. Available: <http://www.um.es/docencia/barzana/IAGP/Iagp2.html>
- [22] G. Bustos, “Métodos de desarrollo de software: El desafío pendiente de la estandarización. software,” Theoria, vol. 12, no. 1, pp. 23–42, 2003. [Online]. Available: <http://www.redalyc.org/articulo.oa?id=29901203>
- [23] C. B. Jurados, Diseño Ágil con TDD. España: SafeCreative, 2010.
- [24] M. T. Gallego. Metodología scrum. [Online]. Available: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>
- [25] J. S. Ken Schwaber. La guía definitiva de scrum: Las reglas del juego. [Online]. Available: <http://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-es.pdf>
- [26] J. P. Alexander Menzinsky, Gertrudis López. Scrum manager. [Online]. Available: [http://www.scrummanager.net/files/sm\\_proyecto.pdf](http://www.scrummanager.net/files/sm_proyecto.pdf)
- [27] M. P. Esteso. Programación extrema: Qué es y principios básicos. [Online]. Available: <https://geekytheory.com/programacion-extrema-que-es-y-principios-basicos>
- [28] U. U. Bolivariana. Programación extrema. [Online]. Available: [http://ingenieriadesoftware.mex.tl/52753\\_xp---extreme-programing.html](http://ingenieriadesoftware.mex.tl/52753_xp---extreme-programing.html)
- [29] C. E. N. L. Gerardo Abraham Morales Urrutia. Proceso de desarrollo para videojuegos. [Online]. Available: [erevistas.uacj.mx/ojs/index.php/culcyt/article/download/299/283](http://erevistas.uacj.mx/ojs/index.php/culcyt/article/download/299/283)
- [30] M. Cerny. D.i.c.e. summit 2002 - mark cerny. Youtube. [Online]. Available: <https://www.youtube.com/watch?v=QOAW9ioWAvE>

- [31] J. Ward. What is a game engine? [Online]. Available: [https://www.gamecareerguide.com/features/529/what\\_is\\_a\\_game\\_.php](https://www.gamecareerguide.com/features/529/what_is_a_game_.php)
- [32] C. M. A. David Vallejos Fernández, Desarrollo de videojuegos: Arquitectura del motor. España: Universida de Castilla - La Mancha, 2012.
- [33] Webopedia. Sdk - software development kit. [Online]. Available: <https://www.webopedia.com/TERM/S/SDK.html>
- [34] M. Azure. ¿qué es middleware? [Online]. Available: <https://azure.microsoft.com/es-mx/overview/what-is-middleware/>
- [35] 3dcadportal. Render-rendering-renderizado. [Online]. Available: <http://www.3dcadportal.com/rendering.html>
- [36] Unity. Requisitos del sistema para la unity versión. [Online]. Available: [https://unity3d.com/es/unity/system-requirements?\\_ga=2.28270718.372269580.1509904500-1621154898.1498793137](https://unity3d.com/es/unity/system-requirements?_ga=2.28270718.372269580.1509904500-1621154898.1498793137)
- [37] UnrealEngine. Hardware and software specifications. [Online]. Available: <https://www.unrealengine.com/en-US/what-is-unreal-engine-4>
- [38] CryEngine. Características. [Online]. Available: <https://www.cryengine.com/features>
- [39] Adobe. Requisitos del sistema | versiones de enero de 2018 y octubre de 2017 de photoshop cc (19.x). [Online]. Available: <https://helpx.adobe.com/es/photoshop/system-requirements.html>
- [40] ——. Requisitos del sistema de la versión de marzo de 2018 y octubre de 2017 de illustrator cc (22.x). [Online]. Available: <https://helpx.adobe.com/es/illustrator/system-requirements.html>
- [41] Autodesk. Requisitos del sistema de autodesk sketchbook. [Online]. Available: <https://knowledge.autodesk.com/es/search-result/caas/sfdarticles/sfdarticles/ESP/System-requirements-for-Autodesk-SketchBook-Pro.html>
- [42] Blender. Requirements. [Online]. Available: <https://www.blender.org/download/requirements/>
- [43] Maya. Requisitos del sistema de autodesk maya. [Online]. Available: <https://knowledge.autodesk.com/es/support/maya/learn-explore/caas/sfdarticles/sfdarticles/ESP/System-requirements-for-Autodesk-Maya.html>
- [44] Ardour. System requirements. [Online]. Available: <https://ardour.org/requirements.html>
- [45] A. H. Núñez, “Además de presupuesto, ¿qué le falta a la cultura en méxico?” [Online]. Available: <https://cuadrivio.net/ademas-de-presupuesto-que-le-falta-a-la-cultura-en-mexico/>
- [46] P. Mondal. Culture: Characteristics and classifications of culture sociology. [Online]. Available: <http://www.yourarticlelibrary.com/culture/culture-characteristics-and-classifications-of-culture-sociology/6223>
- [47] E. de Clasificaciones. Tipos de culturas. [Online]. Available: <http://www.tiposde.org/ciencias-sociales/78-tipos-de-cultura/>

- [48] E. mundo de Tehuacan. Tipos de cultura y cultura híbrida. [Online]. Available: <http://www.elmundodetehuacan.com/index.php/opinion/opinion-conten-ini/28997-Tipos-de-cultura-y-cultura-h%C3%ADbrida>
- [49] S/A. Importancia de la cultura. [Online]. Available: <https://www.importancia.org/cultura.php>
- [50] A. García. Importancia del arte y la cultura. [Online]. Available: [http://www.el-nacional.com/noticias/columnista/importancia-del-arte-cultura\\_79568](http://www.el-nacional.com/noticias/columnista/importancia-del-arte-cultura_79568)
- [51] J. Mireles. La importancia de preservar nuestra cultura. [Online]. Available: <http://www.ruizhealytimes.com/cultura-para-todos/la-importancia-de-preservar-nuestra-cultura>
- [52] P. van den Boer. Introduction to gamification. [Online]. Available: <https://cdu.edu.au/olt/lresources/downloads/whitepaper-introductiontoggamification-130726103056-phpapp02.pdf>
- [53] G. Zichermann and C. Cunningham, Gamification by design: Implementing game mechanics in web and mobile apps. ".o'Reilly Media, Inc", 2011.
- [54] NewVoiceMedia. What are game mechanics and dynamics? [Online]. Available: <https://www.newvoicemedia.com/online-guides/the-ultimate-guide-to-gamification/what-are-game-mechanics-dynamics>
- [55] R. D. Janaki Kumar, Mario Herger. Bartle's player types for gamification. [Online]. Available: <https://www.interaction-design.org/literature/article/bartle-s-player-types-for-gamification>
- [56] N. Lazzaro. (2004) Why we play games:four keys to more emotion without story. [Online]. Available: [http://www.xeodesign.com/xeodesign\\_whyweplaygames.pdf](http://www.xeodesign.com/xeodesign_whyweplaygames.pdf)

# **Capítulo 8**

## **Anexos**

En este capítulo se encuentran todos los anexos que se mencionaron en los capítulos anteriores.

### **8.1. Interfaces**



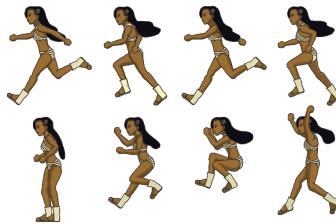
Figura 8.1: a nice plot

### **8.2. Diseño de Personajes**

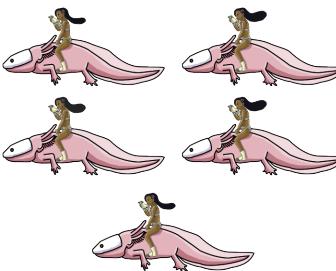
### **8.3. Modelo de Datos**

### **8.4. Control de adicción en el jugador**

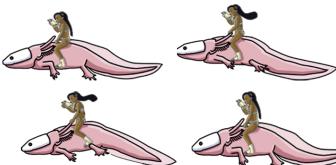
### **8.5. Maquetas de niveles**



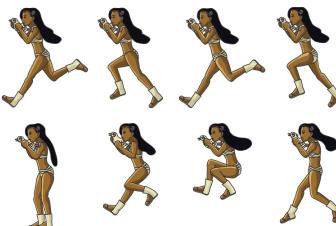
(a) Sprites Malinalli para el primer nivel.



(b) Sprites Malinalli de nado para el segundo nivel.



(c) Sprites Malinalli de salto para el segundo nivel.



(d) Sprites Malinalli para los niveles posteriores al segundo nivel.

Figura 8.2: Sprites del personaje jugable (Autoria propia)

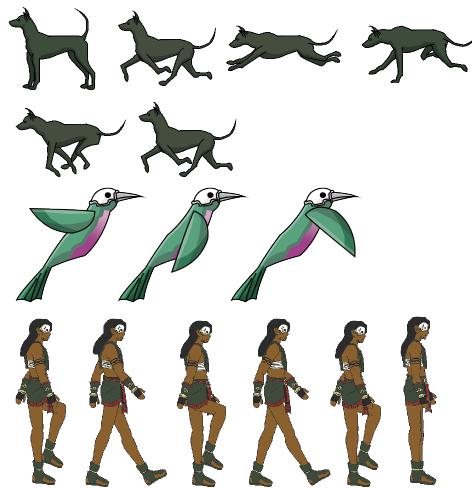


Figura 8.3: Sprites para las diferentes formas que toma Xólotl a lo largo del juego.



Figura 8.4: Sprites para los enemigos normales del juego.

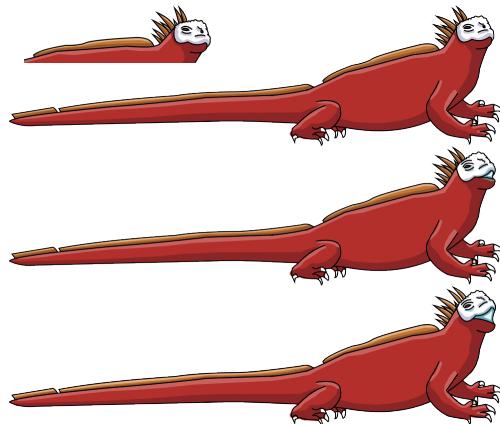


Figura 8.5: Sprites de Xochitonal.



Figura 8.6: Sprites de Itzpapálotl.



Figura 8.7: Sprites de Tlazolteotl.

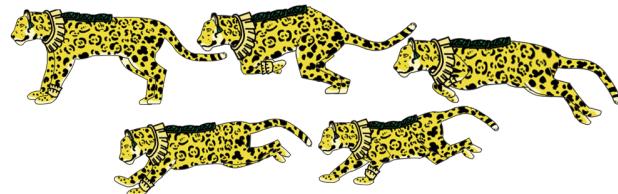


Figura 8.8: Sprites de Tepeyollotl.



Figura 8.9: Sprites de Mictlantecuhtli.

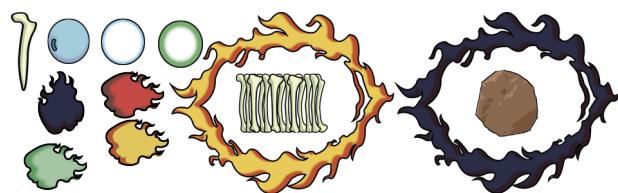


Figura 8.10: Sprites de los ataques de los personajes.