

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE  
TELECOMUNICACIÓN  
UNIVERSIDAD POLITÉCNICA DE CARTAGENA



Trabajo Fin de Estudios

**Implementación de metodologías Lean Manufacturing en el  
desarrollo de videojuegos**



AUTOR: Yoel De Sanctis Clemente  
DIRECTOR: Juan Ángel Pastor Franco  
CODIRECTOR: Diego Alonso Cáceres

Octubre / 2017

# ÍNDICE

0.	INTRODUCCIÓN .....	3	
0.1	OBJETIVOS.....	4	
0.2	ESTRUCTURA DEL TRABAJO DE FIN DE ESTUDIOS.....	5	
PARTE 1			
ESTUDIOS TEÓRICOS.....			6
1.	DISEÑO de VIDEOJUEGOS .....	7	
1.1	QUÉ SON LOS JUEGOS Y POR QUÉ JUGAMOS .....	7	
1.2	ANÁLISIS DEL VIDEOJUEGO Y SUS ELEMENTOS.....	10	
1.3	PROCESO DE DESARROLLO DE UN VIDEOJUEGO Y ACTORES QUE PARTICIPAN EN EL MISMO .....	28	
2.	METODOLOGÍAS LEAN .....	39	
2.1	LEAN MANUFACTURING.....	39	
2.2	LEAN STARTUP .....	49	
PARTE 2			
DESARROLLO DEL PROYECTO.....			60
3.	LEAN MANUFACTURING EN EL DESARROLLO DE VIDEOJUEGOS .....	62	
4.	LEAN STARTUP PARA UN ESTUDIO DE DESARROLLO DE VIDEOJUEGOS .....	68	
5.	DESARROLLO DEL PROTOTIPO JUGABLE.....	77	
5.1	ELECCIÓN DEL GAME ENGINE .....	77	
5.2	OTRO SOFTWARE A UTILIZAR .....	81	
5.3	DESARROLLO EN UNREAL ENGINE 4 .....	82	
5.3	RESULTADO FINAL .....	125	
PARTE 3			
CONCLUSIONES .....			126
6.	CONCLUSIONES .....	127	
6.1.	LÍNEAS FUTURAS.....	127	
6.2.	CONCLUSIONES FINALES .....	128	
6.3	BIBLIOGRAFÍA .....	129	
PARTE 4			
ANEXOS .....			131
Anexo 1.A. Bussiness Model Environment Canvas.....			132
Anexo 1.B. Bussiness Model Canvas.....			133
Anexo 1.C. Value Proposition Canvas.....			134
Anexo 2. Ejemplo de GDD.....			135
Anexo 3.1. Business Environment Model Map del proyecto .....			140
Anexo 3.2. Business Model Canvas del proyecto.....			141
Anexo 3.3. Hipótesis de la propuesta de valor.....			142
Anexo 3.4. Cuestionario para la elaboración de entrevistas .....			145
Anexo 4. Plan de Empresa .....			147

## 0. INTRODUCCIÓN

El mundo actual cambia a una velocidad vertiginosa. Gracias al crecimiento exponencial del progreso tecnológico, nuevos gadgets, tecnologías y estilos de vida que eran impensables hace unas décadas están influyendo en la manera en la que trabajamos, nos relacionamos, nos divertimos o pensamos. Y cada año, los progresos son más ágiles. Nuevos productos y servicios son lanzados, popularizados y olvidados o sustituidos con cada vez mayor celeridad. Y en medio de esta vorágine, se encuentran los videojuegos.

Desde que en la década de los 50 se creara el primer videojuego, nadie habría predicho lo exitosos que se convertirían más adelante. En los años 80, con la popularización de salones recreativos o arcades y los sistemas domésticos, se empezó a vislumbrar el tremendo potencial de estos “juegos”. Inicialmente (y aún hoy día por algunos sectores) considerados un juguete y rodeados de gran polémica en torno a la influencia que ejercían sobre los más jóvenes, los videojuegos han demostrado que llegaron para quedarse.

Su público es cada vez más grande y más heterogéneo <sup>1</sup>. La media de edad de los jugadores cada vez es mayor y seguirá aumentando. Las nuevas generaciones tienen completamente asimilada la existencia de mundos virtuales en los que poder jugar y vivir experiencias que no son posibles en la vida real. Y cuando estos jóvenes crezcan, una nueva generación de jugadores les seguirá, llegando el día en el que todas las personas se consideren “jugadores”.

A este avance increíble del modo de vida de miles de personas, le acompaña el crecimiento de la industria del videojuego. Desde la década del 2000, los videojuegos han pasado a generar más dinero que la industria del cine y la música juntas. En 2016, la industria del videojuego generó 91.000 millones de dólares<sup>2</sup> y las previsiones

---

<sup>1</sup> Why did ancient Egypt spend 3000 years playing a game nobody else liked? - Christian Donlan

<http://www.eurogamer.net/articles/2016-06-01-why-did-ancient-egypt-spend-3000-years-playing-a-game-nobody-else-liked>

<sup>2</sup> Market Brief — Year in Review 2016 | Superdata

<https://www.superdataresearch.com/market-data/market-brief-year-in-review/>

de crecimiento parecen no tener fin. También cabe destacar la popularización de los deportes electrónicos, llamados a convertirse en la próxima gran oferta de ocio y que ya mueven 892 millones de dólares al año.

Debido a esta rápida expansión de la industria y a la proliferación de equipos de desarrollo independientes, se ha hecho patente la necesidad de modernizar las técnicas que se habían venido utilizando hasta ahora para el desarrollo de software y la producción de experiencias audiovisuales interactivas, con el objetivo de mejorar la productividad y rentabilidad de los proyectos.

Una de estas nuevas metodologías, denominada Lean Manufacturing, es aplicada en procesos de fabricación en masa en industrias como la del automóvil con grandes beneficios. Sin embargo, muchos de sus principios pueden ser aplicados en el desarrollo y programación de aplicaciones software. Debido a ello, está ganando gran popularidad entre pequeñas empresas o startups. Este trabajo pretende pues, argumentar y demostrar los beneficios del uso de estas técnicas en el desarrollo de videojuegos, adecuando los desarrollos a su demanda real en beneficio tanto de los jugadores como de los desarrolladores y, en general, de la industria al completo.

## *0.1 OBJETIVOS*

Los objetivos de este trabajo son:

- Aprender y reconocer los principales elementos que forman parte del desarrollo de un videojuego, de manera que se puedan tener en consideración el mayor número de aspectos y áreas posibles.
- Conocer y aplicar metodologías *Lean*, como “Lean Manufacturing” o “Lean Startup”, así como sus ventajas sobre metodologías de trabajo tradicionales.
- Estudiar la viabilidad de uso de técnicas de Lean Manufacturing o “producción limpia” para el desarrollo de videojuegos.
- Ser capaz de desarrollar un prototipo jugable usando conceptos de prototipado y testeado rápido.

## *0.2 ESTRUCTURA DEL TRABAJO DE FIN DE ESTUDIOS*

Este Trabajo de Fin de Estudios se estructura en tres bloques principales.

En la primera parte, se explicarán los estudios teóricos realizados, divididos en dos grandes bloques, el referido a teoría y diseño de videojuegos y, por otra parte, el que trata acerca de metodologías Lean, como Lean Manufacturing y Lean Startup.

En la segunda parte, se aborda la aplicación directa de las metodologías Lean al diseño de videojuegos mediante la elaboración de un caso práctico, un prototipo jugable a través de todas sus fases de desarrollo y testeo. Se tratan y explican también todos los aspectos del desarrollo, así como las decisiones de diseño, de este videojuego en concreto desde su fase más temprana.

Finalmente, la tercera parte se compone de anexos consistentes en los propios documentos de diseño del videojuego, así como los cuestionarios usados en las fases de testeo y cualquier otro documento nacido de este proyecto.

A pesar de que el desarrollo real del trabajo se basa en un modelo iterativo, en el que todas las fases se entremezclan de cara a mejorar, detectar, y solucionar problemas tempranos lo antes posible, siguiendo esta estructura se pretende que la memoria sea lo más ordenada posible de cara a su mejor entendimiento.

PARTE 1

# ESTUDIOS TEÓRICOS

# 1. DISEÑO de VIDEOJUEGOS

En esta sección se tratará de explicar conceptos relacionados puramente con los videojuegos y su desarrollo, abarcando desde los aspectos psicológicos que hacen que jugar nos resulte placentero, hasta las áreas y actores implicados en el desarrollo de un título o la organización de un estudio de desarrollo.

## 1.1 QUÉ SON LOS JUEGOS Y POR QUÉ JUGAMOS

Es imposible entender las razones del éxito de los videojuegos y por qué son tan adictivos para tantas personas sin echar un vistazo hacia nuestro propio interior y a nuestro pasado.

Erraríamos si pensáramos que lo que conocemos como videojuego es una creación de las últimas décadas, que su concepto era algo desconocido para nuestros ancestros. Los juegos han acompañado al ser humano desde sus estadios de desarrollo más tempranos, constituyendo los videojuegos únicamente una evolución lógica permitida por los avances tecnológicos de la era de la información.

No serían muy diferentes tampoco las sensaciones que provocaría en nuestros antepasados el *Juego real de Ur*<sup>3</sup>, el popular *Senet* del Antiguo Egipto o incluso, más allá, los tableros de juego del Neolítico<sup>4</sup>, a las emociones que siente un joven jugando a *Mario Kart* o *League of Legends*. El acto de jugar ha estado presente en nuestras vidas sin importar ni el lugar ni el año en el que viviéramos. Si es así... ¿Constituye sólo una manera de perder el tiempo o tiene alguna utilidad más allá? ¿Qué es lo que define a un juego? ¿Qué lo hace divertido? Y lo que es más preocupante, si es algo fundamental para nuestro desarrollo, ¿Por qué se tiende a despreciar el juego en contraposición al trabajo conforme nos hacemos adultos?

---

<sup>3</sup> Museo Británico. Juego Real de Ur

[http://www.britishmuseum.org/research/collection\\_online/collection\\_object\\_details.aspx?objectId=8817&partId=1](http://www.britishmuseum.org/research/collection_online/collection_object_details.aspx?objectId=8817&partId=1)

<sup>4</sup> A Neolithic Game Board from 'Ain Ghazal, Jordan .Gary O. Rollefson

<http://www.jstor.org/stable/1357113>

Numerosos estudiosos han intentado encontrar una definición exacta del acto de jugar. Lo que en un principio parecería simple ha demostrado ser mucho más complicado de lo pudiera parecer, llegando a ser considerado como imposible de definir de manera absoluta. Debido a su amplitud, riqueza y a su inherencia al ser humano e, incluso, a otras especies animales, podemos decir que cualquier definición no dejaría de ser un mero acercamiento a las características del juego. Pese a ello, a lo largo de los años si se han podido reconocer ciertas características que nos ayudan a entender lo que define a un juego y por qué son tan importantes para las personas<sup>5</sup>.

La propiedad más obvia, quizás, sea que un juego **provoca emociones positivas**<sup>6</sup>. El juego es divertido, siendo una de sus funcionalidades básicas es el entretenimiento y el esparcimiento. Sin embargo, esto nos lleva a preguntarnos, la propia definición de lo que es la diversión y por qué lo que es divertido para un individuo puede ser aburrido para otro. Existen muchos tipos de diversión, cada uno con sus características propias y con distinta afinidad a cada persona<sup>7</sup>. Desde la satisfacción de explorar un espacio desconocido, el esfuerzo por reconocer patrones (la mente humana es propensa a ello, buscamos formas en las nubes o intentamos ver el futuro en posos de té), por poner orden en el caos, la satisfacción de sentirse poderoso o el empeño por la crianza y cuidado de alguien o algo a nuestro cargo, la diversión presente en los videojuegos satisface motivaciones humanas y nos ayuda a obtener felicidad.

Un juego es un **sistema ordenado**. Es necesario tener ciertas reglas consensuadas previamente para poder clasificar una actividad divertida como juego. Se establecen limitaciones, tanto espaciales como temporales y además se definen obstáculos que dificultan la consecución de un determinado **objetivo**. Estas normas crean una

---

<sup>5</sup>Man, Play and Games (1961) - Roger Caillois

[http://creativegames.org.uk/modules/Intro\\_Game\\_Studies/Caillois\\_Man\\_Play\\_Games\\_Chapters1\\_2-2001.pdf](http://creativegames.org.uk/modules/Intro_Game_Studies/Caillois_Man_Play_Games_Chapters1_2-2001.pdf)

<sup>6</sup> Why We Play Games: Four Keys to More Emotion in Player Experiences. Nicole Lazzaro

[http://twvideo01.ubm-us.net/o1/vault/gdc04/slides/why\\_we\\_play\\_games.pdf](http://twvideo01.ubm-us.net/o1/vault/gdc04/slides/why_we_play_games.pdf)

<sup>7</sup> Types of fun - Jonathan Winter

<http://www.managementexchange.com/hack/21-types-fun-whats-yours>

situación de incertidumbre placentera, que tienen lugar en una realidad alternativa, fomentando situaciones espontáneas en las que no se conoce el resultado. Además, se produce una retroalimentación, mediante la cual el sistema nos comunica que tal lo estamos haciendo, ya sea mediante puntos u otras técnicas, motivando una mejora en el jugador.

El juego **tiene valor por sí mismo**. El acto de jugar tiene valor intrínseco, ya sea mediante el desarrollo de habilidades motrices o cognitivas, tratándose de una actividad intrascendente y desinteresada. Cabe destacar el agente del videojuego como conector social, en especial con el auge del multijugador online, facilitando y potencian relaciones personales y la creación de comunidades. Es posible incluir en un juego aspectos formativos, potenciando también así los valores educativos.

Un juego es también una **actividad libre**. Todos los jugadores eligen jugar de manera voluntaria por sí mismos. Conocen y aceptan las reglas previamente pactadas y trabajan para lograr un objetivo que saben que es alcanzable.

Este último aspecto nos puede ayudar a entender la diferencia entre trabajo y juego. Quizás la depresión, un estado apático de baja autoestima, interés y actividad, funcione mejor como contrario al juego que el trabajo. Se podría resumir un juego como un conjunto de obstáculos innecesarios que superar. Para ello, nos esforzamos, centramos nuestra atención, nos implicamos y, en definitiva, *trabajamos* para conseguir un objetivo. Si aceptamos lo anterior, no hay mucha diferencia respecto a lo que entendemos propiamente por trabajo, salvo el hecho de que jugamos por voluntad propia y trabajamos, normalmente, por obligación. Necesitamos mantener nuestro nivel de vida, cumplir las expectativas de otra persona, los fallos nos pueden perjudicar directamente, en muchas ocasiones no nos sentimos satisfechos, etc.

Al contrario de lo que en un principio se pueda pensar, ya sea trabajo físico, mental, creativo, de búsqueda o una tarea laboriosa, al ser humano le apasiona trabajar. Si es así ¿por qué en nuestro día a día nos supone una carga? Simplemente porque no podemos escoger el trabajo apropiado en el tiempo y con el objetivo concreto, sin poder visualizar claramente a que estamos contribuyendo. Sin embargo, todo ello

cambia cuando el trabajo se parece al juego, cuando entendemos el sistema, entramos en él por voluntad propia y nos motiva a mejorar, a desarrollar nuestras habilidades.

Por todas estas razones, es fácil reconocer el beneficio potencial de los videojuegos en nuestras vidas. Pueden ayudar a nuestro desarrollo personal, a relacionarnos, a entender mejor el mundo, a potenciar nuestras fortalezas, a hacernos más felices y, en definitiva, a mejorar nuestra calidad de vida<sup>8</sup>.

## 1.2 ANÁLISIS DEL VIDEOJUEGO Y SUS ELEMENTOS

Un videojuego, al igual que otros medios audiovisuales, es la suma de muchos componentes. No se debe restar importancia a ninguno de ellos, pues si uno de ellos falla, puede lastrar gravemente el producto final y mermar su calidad.

Por ello, aunque siempre se destaque o flaquee en algunos puntos, todas las áreas que abarca un videojuego deben estar, idealmente, igualmente estudiadas, trabajadas y cuidadas.

### 1.2.1 Jugabilidad y mecánicas

Es el núcleo de un videojuego, su punto más diferenciador y el de mayor importancia. Para que un videojuego pueda ser considerado como tal debe de existir una **interacción** entre el jugador y el medio. Y para que esta interacción sea posible, deben haberse establecido ciertas reglas, ciertas mecánicas que permitan al jugador influir sobre el videojuego o ser influido por este. Estas mecánicas resultan, en mayor o menor medida, en una jugabilidad, la cual puede llegar a ser muy diferente entre un videojuego y otro.

Para agrupar y categorizar los videojuegos se utiliza el concepto de **género**, similar al del cine o de la literatura pero que, a diferencia de estos, en el videojuego se define desde dos enfoques: el género de la historia/mundo, definido por la narrativa (ciencia

---

<sup>8</sup> Jane McGonigal's Reality Is Broken: How Videogames Change the World

<https://www.thedailybeast.com/jane-mcgonigals-reality-is-broken-how-videogames-change-the-world>

ficción, fantasía, histórico, etc.), y el género jugable, el cual viene dado por las mecánicas y se entiende como más determinante. Algunos de los géneros típicos son los siguientes:

**Acción.** Se basan en la habilidad del jugador y en la capacidad de reaccionar a lo que ve en pantalla. Uno de los géneros más tempranos y con muchas variaciones. Entre ellas, podemos destacar los **Shooters** o juegos de disparos, uno de los géneros más populares en el que los jugadores lanzan proyectiles contra enemigos u otros jugadores y se centra en la puntería y agudeza visual. También encontramos juegos de **Lucha**, en los que se recrean combates, normalmente, cuerpo a cuerpo entre diferentes personajes.

**Aventura.** Estos videojuegos se centran en unos personajes y la trama que los hace avanzar. Aunque pueden compartir elementos con los juegos de acción, generalmente no son tan frenéticos y pueden llegar a incluir aspectos de toma de decisiones, gestión de objetos o resolución de puzzles. También es otro de los géneros más antiguos, destacando las **Aventuras gráficas**, donde los jugadores avanzan en la historia mediante la resolución de rompecabezas planteados por la propia trama.

Los géneros de acción y aventura son un tanto ambiguos, ya que se suelen entremezclar y formar géneros como los **Plataformas** (el personaje debe avanzar por un escenario superando obstáculos y/o enemigos), los de **Sigilo** (se centran en evitar un enfrentamiento directo con los enemigos), los **Survival Horror** (análogo al género de terror del cine) o los títulos de **Rol**, centrados en el progreso y evolución de un personaje, la personalización del mismo y una fuerte carga narrativa.

**Estrategia.** Toman su precedente en los juegos de tablero tradicionales, como el Ajedrez o el Go. Están enfocados a la planificación y a la gestión mediante el uso del intelecto. La mayoría de ellos tienen temática bélica, aunque también se encontramos títulos enfocados a la gestión de comercios o de imperios. Podríamos categorizarlos en dos grandes grupos, **Estrategia en tiempo real**, en la que todos los jugadores o actores del juego realizan sus acciones al mismo tiempo, o **Estrategia por turnos**, en la que las acciones se dividen en rondas bien definidas y la agilidad pasa a un

segundo plano frente a la capacidad de análisis.

**Simulación.** Se basan en recrear actividades del mundo real, centrándose en la creatividad y la capacidad de gestión del jugador. Muchos no tienen condiciones de victoria o derrota bien definidas y otros pueden tener una fuerte componente didáctica. Podemos encontrar **Simuladores de construcción y gestión**, en los que el jugador debe administrar, por ejemplo, una ciudad, un comercio, una granja, etc. **Simuladores de vida**, enfocados a controlar una vida artificial, o **Simuladores de Vehículos**, desde conducción hasta camiones, vehículos de combate o trenes.

Relacionado con la simulación encontramos los juegos **Deportivos** o los de **Carreras**, en los que, a pesar de recrear una actividad real, a menudo puede primar la diversión frente al realismo, incorporando aspectos más parecidos al subgénero **Arcade** de las primeras recreativas. Además, dentro de los videojuegos deportivos se pueden establecer tantas categorías como deportes existen en el mundo real.

Otros géneros que podríamos nombrar son los videojuegos de **Puzles**, los **Musicales**, los **Sandbox** o los **Educativos**. A pesar de ello, esta lista seguiría incompleta puesto que nuevos subgéneros aparecen día a día y seguirán haciéndolo conforme la industria se vaya desarrollando y la tecnología posibilite nuevas experiencias.

Independientemente de sus mecánicas y género, la jugabilidad de un videojuego debe ser atractiva para el jugador (no necesariamente divertida) y para que así sea debe estar sustentada por varios pilares. En caso de que no sea así, el juego puede llegar a no resultar satisfactorio para el jugador y le producirá hastío o desmotivación, por lo que conviene estudiar y cuidar cada uno de ellos.

#### [La dificultad o el reto](#)

Un juego es, básicamente, un reto o serie de retos a superar. Para que un videojuego sea divertido y satisfactorio, el objetivo a alcanzar debe de suponer algún tipo de **desafío** para el jugador, lo cual también se aplica a los juegos tradicionales (por ejemplo, un juego simple: tirar piedras para derribar latas). Estos desafíos no son solo la meta final del videojuego (por ejemplo, salvar el mundo), sino también la suma de

retos más pequeños, como resolver un puzle, saltar un obstáculo o vencer a un enemigo.

El tipo de retos a plantear al jugador pueden ser muy diferentes, existiendo relación con los tipos de diversión que se plantearon en el apartado anterior. Podemos encontrar retos de control de territorios, de predicciones en juegos de azar, de razonamiento espacial, supervivencia, construcción, destrucción, recolección, persecución o huida, comercio o completar un circuito o ruta. Es tarea del diseñador ensamblar estos retos de manera que su dificultad se adecue a las expectativas del jugador.

Escalar la dificultad de los retos es un aspecto complicado, pero necesario para construir experiencias absorbentes. Si se opta por una dificultad muy baja, el jugador se aburrirá y, si, por el contrario, la dificultad es muy elevada, el jugador se frustrará. Es relevante en este punto el concepto psicológico del “Flow” o “Flujo”<sup>9</sup>, un estado mental en el que una persona está totalmente concentrada e inmersa en una actividad concreta. Además de aplicaciones en espiritualidad, educación, trabajo y deportes su impacto en el diseño de videojuegos es directa.

En este estado de “Fluir” el jugador está totalmente implicado en el videojuego, toda su motivación, concentración y energía está enfocada en esta actividad para conseguir un objetivo determinado que le producirá placer. Se produce una pérdida de la percepción personal, la conciencia se reduce a la propia actividad y el jugador experimenta una pérdida del sentido temporal ante una situación en la que tiene cierto control y con la que está comprometido.

Para alcanzar este estado, se debe alcanzar un equilibrio perfecto entre la dificultad de la tarea realizada y la dificultad del individuo a la hora de su realización.

---

<sup>9</sup> Flow: The Psychology of Optimal Experience - Mihály Csíkszentmihályi

[http://www.ode.state.or.us/opportunities/grants/nclb/title\\_ia\\_basicprograms/schoolimprovement/transformation7flow.pdf](http://www.ode.state.or.us/opportunities/grants/nclb/title_ia_basicprograms/schoolimprovement/transformation7flow.pdf)

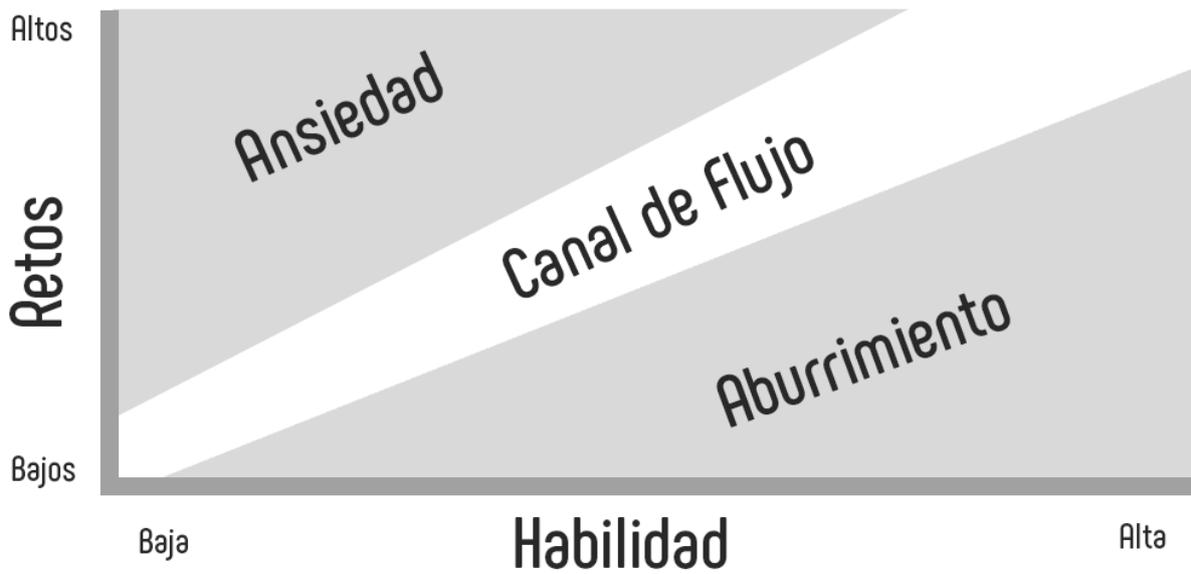


Figura 1. Diagrama del Flujo (A). Basado en *The Art of Game Design* de Jesse Schell

En el diagrama de la figura anterior, el eje vertical representa el nivel de los retos, mientras que el eje horizontal representa la habilidad del jugador. Las áreas en gris representan los estados negativos. En el área superior, la dificultad de los retos es demasiado elevada respecto a la habilidad del jugador, por lo que le produce ansiedad. En el área inferior, la habilidad del jugador es superior a la dificultad de los retos, por lo que el juego le produce aburrimiento. Entre estas dos áreas encontramos lo que denominamos el canal de flujo, es el área en la que la dificultad y la habilidad están en armonía y el jugador es trasladado al estado de *Flujo*. Los diseñadores de videojuegos deben, por tanto, intentar mantener al jugador en ese canal.

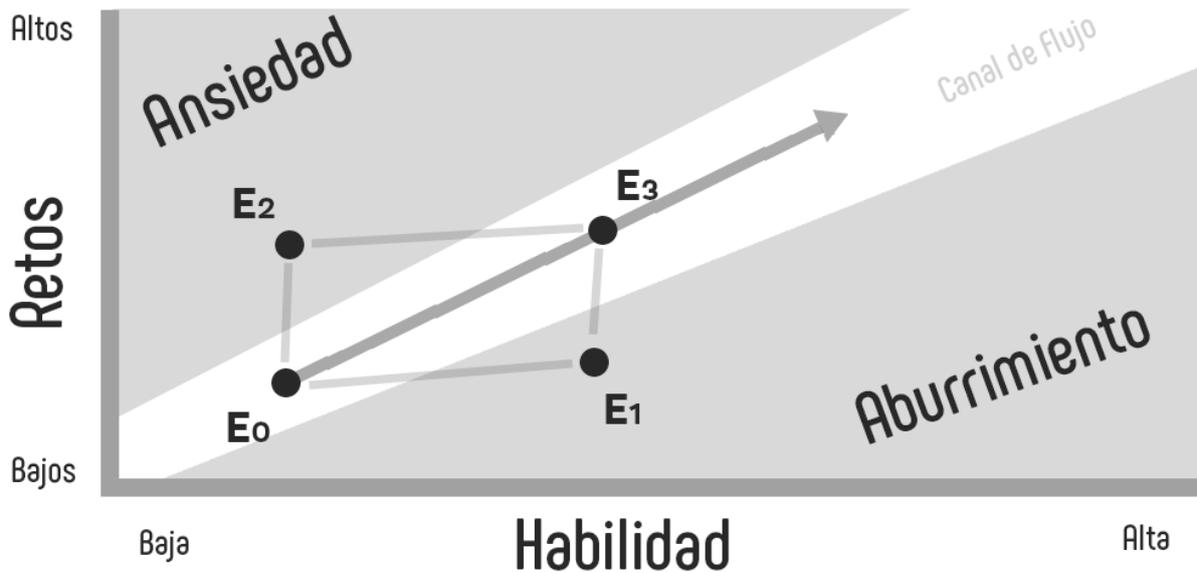


Figura 2. Diagrama del Flujo (B). Basado en *The Art of Game Design* de Jesse Schell

En la figura anterior, el jugador comienza en el estado  $E_0$ , su nivel de habilidad es bajo puesto que acaba de comenzar a jugar al videojuego. Para avanzar en el desarrollo del mismo, debe alcanzar el estado  $E_3$ . Si la dificultad del juego progresa demasiado lenta, alcanzará el estado  $E_1$ , en el que introducimos al jugador en el área del aburrimiento. Por otro lado, si la dificultad progresa muy abruptamente el jugador llegará al estado  $E_2$ , en el que sentirá ansiedad y pensará que el juego es demasiado difícil.

Tanto la ansiedad como el aburrimiento producen frustración en el jugador, y ese es un sentimiento que los diseñadores de videojuegos deben evitar a toda costa, ya que la frustración puede llevar al jugador a abandonar el videojuego. Por tanto, el progreso ideal sería el pasar directamente del estado  $E_0$  al  $E_3$  sin abandonar el canal de flujo. Extrapolando podríamos dibujar una línea que representa el progreso de la dificultad del juego, siendo la dificultad y la habilidad del jugador directamente proporcionales durante todo el transcurso del mismo.

Sin embargo, podemos ofrecer una mejor experiencia de juego si seguimos el patrón de la línea de la siguiente figura:

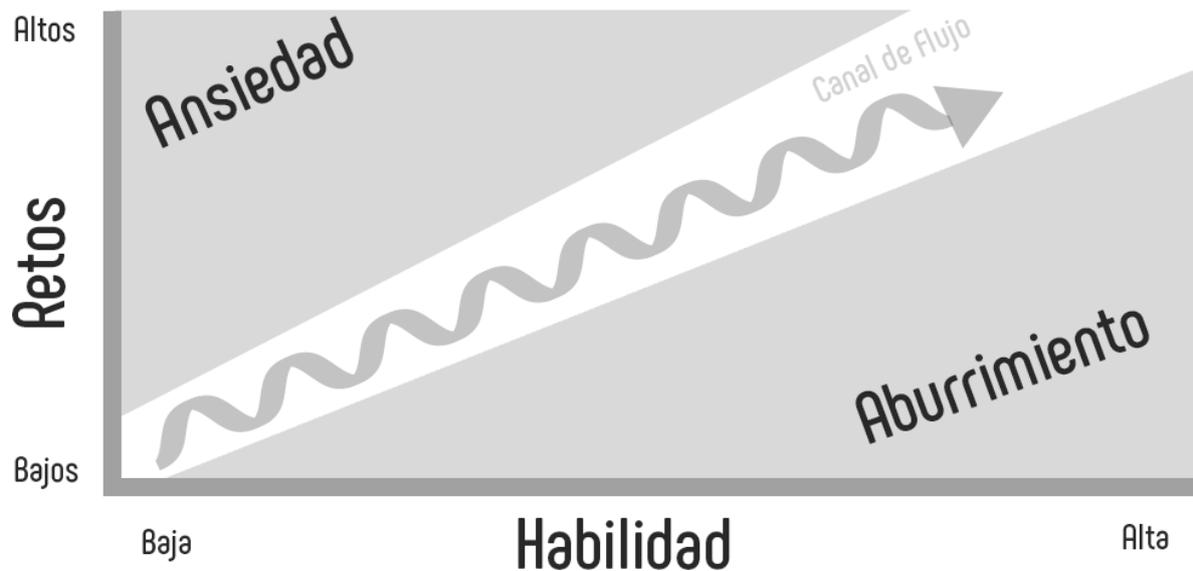


Figura 3. Diagrama del Flujo (C). Basado en *The Art of Game Design* de Jesse Schell

Así, la dificultad sigue progresando a buen ritmo dentro del canal, pero a la vez estamos ofreciendo al jugador una experiencia más satisfactoria. La dificultad incrementa y decrece ligeramente a lo largo del transcurso de la partida, ofreciendo momentos más calmados y otros en los que se requiere mayor atención por parte del jugador, manteniéndolo en tensión a la vez que se siente cómodo y se fomenta su sentido de progreso respecto al videojuego.

Balancear la dificultad del videojuego para que el jugador se mantenga en el área central del diagrama es muy complicado, y muchas veces se llega a pensar que la culpa es del propio jugador y su habilidad para mejorar y adaptarse a los nuevos retos. Sin embargo, esto es un error, ya que la responsabilidad final siempre recaerá en el desarrollador. Si añadimos a la ecuación que la interacción entre el jugador y el videojuego no es siempre la de *Jugador contra Juego*, si no que podemos tratar con *Juegos cooperativos*, *Jugador contra Jugador*, *Equipo contra Equipo*, *Competición multilateral* (todos contra todos) entre otros, la tarea se torna todavía más compleja.

¿Cómo podemos entonces ajustar la dificultad del videojuego? No hay una fórmula exacta, pero podemos hacer uso de técnicas como selectores de dificultad, en los que los propios jugadores escogen el nivel de dificultad del videojuego entre unos niveles predefinidos, técnicas de aleatoriedad (mediante la inclusión de elementos aleatorios

se introduce una incertidumbre que el jugador debe optimizar) o el uso de loops o ciclos de dificultad, en los que cuanto más avanza el jugador más retantes serán las situaciones a las que se enfrentará el jugador (produciendo un final en el que las condiciones de victoria no son conseguibles si se abusa de esta técnica). Finalmente, también son interesantes el uso de técnicas de ajuste de dificultad dinámico, en la que se analiza el desempeño del jugador y se adapta la dificultad del juego en consecuencia de manera dinámica. En el caso de videojuegos multijugador, se pueden establecer hándicaps o ventajas a los jugadores en función de su habilidad o, en juegos online, crear sistemas de emparejamiento complejos que busquen, idealmente, una partida balanceada en la que todos los jugadores tienen la misma probabilidad de vencer que de ser vencidos.

En cualquiera de estos casos, el feedback de los jugadores es una de las herramientas más potentes de la que los desarrolladores disponen a la hora de escalar la dificultad del videojuego correctamente, así como la de todos los elementos que lo componen, ya que si alguna mecánica hace la experiencia más fácil o difícil que otra estaríamos destruyendo el siguiente pilar.

### La posibilidad de elección

La posibilidad de elección es básica para la interacción entre el juego y el jugador. Las elecciones hacen que el juego sea más interesante, ya que cambian el transcurso de la experiencia y hacen que esta cambie para diferentes jugadores e incluso para distintas partidas del mismo jugador.

Las elecciones pueden ser analíticas, en las que se presentan al jugador varias posibilidades de manera clara sobre las que debe reflexionar para hacer una elección (por ejemplo, la elección de una de las dos rutas posibles para avanzar en un nivel o si quiere que su personaje sea un espadachín o un arquero). También pueden ser espontáneas, siendo decididas mediante la intuición y la habilidad antes que la premeditación (por ejemplo, lanzar un ataque o defenderse para preparar un contraataque).

De cualquier manera, estas elecciones deben de ser reales y deben tener un impacto tangible en el juego<sup>10</sup> e, idealmente, inmediato, favoreciendo así la retroalimentación. La habilidad del jugador se ve reflejada en este impacto, ya que cuanto más habilidoso es, más probable es que sus decisiones sean correctas y ofrezcan el resultado esperado.

Es común la presencia de una falsa libertad de elección, mediante la inclusión de diferentes opciones que no tienen ninguna consecuencia real en el transcurso del juego. Un ejemplo de ello son las usuales cadenas de dialogo en las que el jugador puede elegir como interaccionar con otro personaje pero que, no importa lo que haya escogido, siempre llegará al mismo punto final. Otro caso es la supuesta ausencia de elección; Se traslada al jugador a un entorno, supuestamente libre en el que cualquier acción diferente a la prefijada produce el mismo resultado o ni siquiera es posible.

El abuso de estos métodos de falsa libertad es perjudicial para el propio videojuego, ya que el jugador pierde la sensación de control, piensa que está siendo engañado y le produce frustración.

Otro caso de elección falsa está directamente relacionado con la dificultad. Si los resultados de determinadas elecciones producen consecuencias muy dispares, resultando una de las opciones claramente más beneficiosa que las demás en términos de dificultad, en realidad no hay elección posible ya que el jugador elegirá favorablemente la opción más provechosa. Este escenario puede verse claramente en el balance entre las armas de un videojuego de disparos o el poder de los diferentes personajes en un juego de lucha, ya que si uno de ellos sobresale respecto al resto los jugadores abusarán de esta mecánica y la posibilidad real de elección desaparecerá.

---

<sup>10</sup> Meaningful Choice in Games: Practical Guide & Case Studies - Brice Morrison

[https://www.gamasutra.com/blogs/BriceMorrison/20131119/204733/Meaningful\\_Choice\\_in\\_Games\\_Practical\\_Guide\\_Case\\_Studies.php](https://www.gamasutra.com/blogs/BriceMorrison/20131119/204733/Meaningful_Choice_in_Games_Practical_Guide_Case_Studies.php)

## **El cambio y la probabilidad**

Como ha sido comentado anteriormente, un videojuego no es un único reto sino la concatenación de muchos retos pequeños. Para que un videojuego sea atractivo es imperativo pues, que estos retos sean diferentes entre sí. Se hace necesario entonces la existencia de un cambio constante en el transcurso del juego, de manera que la cantidad de contenidos que se ofrecen son mayores y la rejugabilidad y la vida útil del título se ven potenciadas.

La sensación de repetición es otro punto que evitar por el diseñador de videojuegos, por lo que incluso la inclusión de pequeñas modificaciones que hagan que el jugador perciba que se encuentra ante una situación diferente es muy ventajosa. Estos ligeros cambios pueden ser la modificación de las estadísticas y el color de un enemigo, la inclusión de un sistema de trofeos o logros que planteen limitaciones o retos alternativos al jugador o la existencia de pequeñas rutas alternativas o áreas secretas en un mismo nivel.

Cabe destacar en este apartado las tecnologías de creación procedimental de contenidos, mediante las cuales los datos y recursos son generados de manera matemática en vez que de forma manual. A pesar de ello, una cantidad muy grande de contenidos puede abrumar al jugador e incluso aburrirlo si no son lo suficientemente diferentes. Si, por el contrario, son demasiado diferentes a las expectativas que tiene puestas en el juego, pueden tener efectos negativos.

Por otro lado, también es necesaria la presencia de la probabilidad en el videojuego. Si una acción siempre produce la misma reacción el juego se convierte en un acertijo, el cual, una vez resuelto, no produce más satisfacción. Esta probabilidad puede venir tanto de la propia habilidad del jugador (por ejemplo, el tanto por ciento de disparos acertados respecto a los totales) o de las propias mecánicas implementadas (un enemigo dispone de varios diferentes ataques que ejecuta sin un orden prefijado).

Esta aleatoriedad produce una sensación de incertidumbre que resulta en un estado de alerta por parte del jugador. Sin embargo, si esta aleatoriedad hace que el jugador

sienta que está siendo castigado injustamente, se estará aplicando de manera incorrecta.

### 1.2.2 *Narrativa*

La narrativa es, quizás, el segundo aspecto más importante de un videojuego. No es extraño confundir el concepto de narrativa como únicamente a la historia de un videojuego. Un juego no necesita estrictamente una trama, un argumento, pero siempre posee una narrativa. Desde un punto de vista cognitivo, la narrativa es una secuencia de hechos conectados, una serie de cambios de estado. Sirve para establecer un orden y para ayudarnos a organizar entender las cosas, puesto que, si no existiera la narrativa, todo sería caótico e imprevisible<sup>11</sup>.

Si atendemos a esa definición, hasta los juegos más sencillos y abstractos poseen una narrativa, la cual puede ser tan simple como, por ejemplo, un cuadrado moviéndose en pantalla mientras esquiva cuadrados más pequeños. Si nos encontráramos con un videojuego así, sería complicado, a simple vista, extraer una narrativa. Pero la tendría, ya que, si el cuadrado choca con alguno de los cuadrados pequeños, este es destruido y, por lo tanto, existe un cambio de estado. Además, dota al videojuego de un conflicto, un elemento necesario para lo que entendemos como historia.

A pesar de que existen multitud de juegos en los que su única narrativa consiste en un duelo entre el jugador y el propio videojuego, desde su inicio los videojuegos han sido una herramienta comunicativa, y, como tal, siempre han intentado contar cosas en mayor o menor medida.

Si volvemos al ejemplo anterior, aún establecido el conflicto la trama sería difícil de apreciar. Si dotáramos a este videojuego de un contexto, como una breve frase o imagen en la que se transmita al jugador que, de hecho, el cuadrado grande es una

---

11 ¿Qué es la narrativa en un videojuego?

<http://www.zehngames.com/articulos/la-narrativa-videojuego/>

nave espacial esquivando asteroides, haría más fácil abstraer un argumento e incluso entender mejor el propio juego (el peligro que representan los cuadrados pequeños).

En los primeros videojuegos, en los que sólo se podían presentar pocos píxeles en pantalla y utilizar mecánicas simples, se utilizaban técnicas parecidas, como las pegatinas que adornaban las primeras recreativas, para contar una historia. Estas historias solían ser bastante sencillas, y ha sido el desarrollo tecnológico que ha experimentado la industria el que ha permitido que los videojuegos planteen historias tan complejas como la de una novela o una película.

Su estructura es similar a la de otros medios; Tienen un argumento que se desarrolla como una secuencia de introducción, nudo y un desenlace, están ambientadas en un mundo real o imaginario, están protagonizadas por unos personajes que evolucionan durante el transcurso de la historia, etc. Sin embargo, algunas características causadas por la interactividad hacen a la narrativa del videojuego única.

Estos aspectos son los que son estudiados por la narratología, la rama que analiza al videojuego como una estructura narrativa clásica y como los elementos de la misma se adaptan al formato interactivo. Por otro lado, se define la ludología<sup>12</sup> como la rama que analiza la narrativa del videojuego a través de las mecánicas y la jugabilidad. Aunque no son incompatibles, estas dos vertientes suelen oponerse.

Esta confrontación entre la historia o el argumento que se nos cuenta frente a la jugabilidad del videojuego nos lleva a otros dos términos: la narrativa embebida y la narrativa emergente<sup>13</sup>. La narrativa embebida es el argumento premeditado, que se plantea tanto de manera explícita, (con cinemáticas, textos o diálogos), como implícita (elementos gráficos en el juego, acciones de personajes, etc.). Por otro lado, la narrativa emergente es aquella que surge de la propia experiencia del jugador al

---

12 What is ludology? A provisory definition.

<http://www.ludology.org/2001/07/what-is-ludolog.html>

13 GDC 2014: Jeremy Bernstein - "Stories, Meaning and Emergent Narrative: A Therapy Session"

<https://archive.org/details/GDC2014Bernstein2>

interactuar con el videojuego. Esta narrativa emergente puede estar debida a mecánicas que fomenten cierta actitud en el jugador o simplemente a coincidencias o comportamientos imprevistos.

De manera ideal, estas dos narrativas deben ser consistentes, ya que si una se opone a la otra se produce una disonancia ludonarrativa<sup>14</sup>, es decir, lo que nos cuentan es contrario a lo que nos hacen hacer (aunque esto no tiene que ser, necesariamente, negativo). Por ejemplo, en un videojuego en el que su argumento gire en torno a un mensaje pacifista, pero en el que el jugador deba eliminar enemigos continuamente.

Todos estos términos son jóvenes, ya que el estudio de la narrativa del videojuego es un nuevo campo que ha surgido gracias a la maduración del medio. El área de la narrativa es un aspecto en el que profundizar, algo que ya se está haciendo mediante el análisis y la innovación experimentada por numerosas propuestas en los últimos años, a las que podríamos categorizar como fuera de lo hasta ahora convencional.

### *1.2.3 Arte y apartado gráfico*

Como medio audiovisual, la representación artística en el videojuego comprende la manera de transmitir al jugador tanto la jugabilidad como la narrativa del título. La elaboración del arte de un videojuego comienza en las etapas tempranas del mismo, una vez establecidos los puntos en los que se sustentara el diseño de las mecánicas como el argumento del título.

Conforme han pasado los años y la tecnología ha permitido una mayor libertad creativa, a este proceso se han destinado cada vez más recursos y profesionales. En los inicios del videojuego, sólo podían ser representadas en pantalla formas simples, tras lo que llegaron la posibilidad de incluir más colores, un mayor número de píxeles, gráficos tridimensionales... Llegando a la actualidad, en la que la potencia gráfica de los ordenadores y videoconsolas va camino de permitir el fotorealismo.

---

14 Gameplay, narrative, and dissonance in 2017

<https://www.gamecrate.com/gameplay-narrative-and-dissonance-2017/16748>

Dado que las posibilidades son tan amplias, los aspectos gráficos de los videojuegos pueden diferir muchísimo, siendo complicado intentar categorizar o dividir estrictamente los mismos, ya que, como cualquier medio artístico, su propia naturaleza se basa en traspasar mediante el uso de la creatividad las fronteras establecidas.

Podríamos encontrar videojuegos con una representación bidimensional, los que simulan un aspecto artesanal, en el que los elementos se representan con objetos como cartón o tela, cel shading (con un aspecto de dibujos animados, pudiendo replicar aspectos inspirados en comics, acuarelas o distintos estilos artísticos) , el uso de collages (mezclando diferentes estilos artísticos) apartados visuales realistas o estilizados, abstractos, etc.

Independientemente del apartado gráfico que se escoja para un videojuego, este estar en concordancia con la jugabilidad y, sobre todo, con la narrativa (aunque si se pretende causar confusión al jugador, se puede jugar con este último principio).

Esta área merece la especial atención del desarrollador, ya que su pone la primera impresión del jugador; Antes de siquiera jugar a un videojuego, el público verá capturas o vídeos del mismo y, si estas imágenes no le agradan, el videojuego perderá la atención del jugador. Por ello, el arte visual del videojuego es un factor determinante para la venta del mismo.

#### *1.2.4 Sonido*

El apartado sonoro del videojuego es, quizás, el área más ignorada por los jugadores. Sin embargo, el sonido es uno de los principales responsables en proporcionar inmersión al jugador<sup>15</sup>, por lo que el esfuerzo en este campo es necesario.

---

15 Playing with your mind: The psychology of sound in video games  
<http://www.amplifon.ie/resources/playing-with-your-mind/>

Si pensamos en los cinco sentidos del cuerpo humano (vista, olfato, gusto, oído y tacto), en el videojuego no podemos contar ni con el gusto ni con el olfato. De los restantes sentidos, la vista y el oído son nuestra principal vía de información a la hora de percibir el mundo. Respecto al sentido de la vista, existe una limitación a la hora de transmitir información visual de manera realista, ya que, aunque tecnologías como la realidad virtual y las pantallas en tres dimensiones han ido cobrando fuerza, en la gran mayoría de ocasiones simplemente observamos una representación en dos dimensiones de un entorno tridimensional por medio de una pantalla, que en ningún momento es comparable a la información obtenida con nuestros propios ojos.

Quizás debido a que esa limitación está menos marcada en el sonido, puede llegar a ser más evocador que una imagen o vídeo. Probablemente resulte más fácil trasladar nuestra mente a la costa si escuchamos el sonido del oleaje, de las gaviotas y del viento que si simplemente vemos una imagen vídeo del mismo escenario.

Además de sumergir al jugador en el ambiente del videojuego, el sonido también le proporciona información. Mediante pequeñas pistas sonoras se puede persuadir al jugador para que realice una acción o informarle de que hay un secreto cerca de su posición. Si pensamos en géneros como la simulación, en el que es fundamental que el sonido sea igual al original que produce el objeto a simular, o en los juegos de disparos, en los que el ligero sonido de unos pasos cercanos o el caos de disparos y explosiones ponen en alerta al jugador, se puede ver claramente la importancia de los efectos de sonido.

A fin de cuentas, los sonidos juegan con nuestra mente y nos provocan emociones. Precisamente, en este último punto también entra en juego la banda sonora de un título. Las bandas sonoras han evolucionado casi a la par que los apartados visuales; Se ha evolucionado desde melodías repetitivas con tonos limitados a composiciones complejas que nada tienen que envidiar a grandes producciones cinematográficas, mereciendo incluso premios Grammy<sup>16</sup>.

---

16 «Journey» se convierte en el primer videojuego nominado a un Grammy por su banda sonora  
<http://www.fotogramas.es/Noticias-cine/Journey-se-convierte-en-el-primer-videojuego-nominado-a-un-Grammy-por-su-banda-sonora>

Un buen apartado sonoro puede convertir un buen juego en uno inolvidable, ya que nos pone en situación respecto a lo que ocurre en pantalla y pueden generar sensaciones que el jugador recordará. Así, además de en los juegos de género musical, que se basan por completo en este apartado, todo videojuego necesita de esfuerzos en este ámbito, el cual también está en constante evolución ya sea mediante la tecnología, como los sistemas de audio inmersivo o el sonido adaptativo, o simplemente, mediante la creatividad, como, por ejemplo, el uso de cráneos humanos para grabar los sonidos de un videojuego<sup>17</sup>.

#### 1.2.5 Tecnología

No se puede hablar de videojuegos sin tener presente la tecnología, ya que es ella la que los permite.

Mucho ha evolucionado la potencia de las videoconsolas y computadores actuales respecto a los primeros. La primera videoconsola comercial, la Magnavox Odyssey, apenas tenía 40 transistores e incluso el jugador era el encargado de llevar la cuenta de las puntuaciones, ya que la consola no tenía capacidad para ello. Por otro lado, hoy en día ya comparamos Teraflops al hablar de la potencia de una videoconsola.

Sin embargo, quedarse meramente en la potencia sería superficial, ya que tan importante como la capacidad de procesamiento es la manera en la que se hace uso de ella. Entran en juego los lenguajes de programación, las matemáticas y la algoritmia, la inteligencia artificial, las tecnologías de renderizado gráfico, la conexión a internet, la popularización de los smartphones y los propios periféricos que actúan como interfaz de comunicación bidireccional entre el usuario y el juego.

---

17 Audio Design Deep Dive: Using a human skull to create the sounds of Inside  
[www.gamasutra.com/view/news/282595/Audio\\_Design\\_Deep\\_Dive\\_Using\\_a\\_human\\_skull\\_to\\_create\\_the\\_sounds\\_of\\_Inside.php](http://www.gamasutra.com/view/news/282595/Audio_Design_Deep_Dive_Using_a_human_skull_to_create_the_sounds_of_Inside.php)

Las pantallas han evolucionado, permitiendo mayores resoluciones, imágenes en tres dimensiones, mayores rangos de colores... De igual manera, lo han hecho los controladores y los demás periféricos, con tecnologías como la vibración o el control por movimiento. Todos estos avances permiten que el videojuego sea un medio cada vez más complejo e inmersivo.

De la misma manera que las tecnologías actuales eran impensables hace unas décadas, de igual manera se abre un futuro lleno de posibilidades. Con el desarrollo de la realidad virtual o aumentada, la generación procedimental de contenidos y narrativa, la evolución de los dispositivos móviles o la globalización de los deportes electrónicos, imaginar un futuro en el que se produzca una migración masiva hacia mundos digitales multisensoriales <sup>18</sup> no es improbable.

#### 1.2.6 *El Game Feel*

Todos los elementos anteriores sirven para dar forma a lo que llamamos “*Game Feel*”, las sensaciones intangibles que nos produce un videojuego al jugarlo <sup>19</sup>.

Partimos de que al contrario que en la literatura o el cine, en el que la persona es un simple espectador de los acontecimientos, en el videojuego el jugador es el protagonista, vive la experiencia en primera persona. Esto ocasiona un hecho curioso, al jugar a un videojuego la persona extrapola su propia identidad al avatar del videojuego. Mientras que viendo una película podríamos decir “*¡Han matado al*

---

<sup>18</sup> A pesar de que, en el videojuego, como se ha citado anteriormente, sólo se transmite información mediante la vista, el oído y el tacto, ya hay estudios y desarrollos en el campo de las tecnologías de olor digital. Estas representan un reto importante ya que, al contrario que la luz o el sonido, los aromas son transmitidos por partículas en vez de ondas. Esto supone una dificultad añadida al poco control que se tiene al transmitir un olor, ya que factores como el calor, las corrientes de aire o la humedad afectan al mismo. En las últimas décadas ha habido numerosas propuestas comerciales, habiendo fracasado todas ellas. Sin embargo, estudios relacionados con la estimulación eléctrica de los receptores olfativos (<http://nghai.net/2016/contents/nghai16-hariri.pdf>), podrían suponer un avance hacia una era de experiencias virtuales multisensoriales.

<sup>19</sup> Game Feel: The Secret Ingredient - Steve Swink

[https://www.gamasutra.com/view/feature/130734/game\\_feel\\_the\\_secret\\_ingredient.php](https://www.gamasutra.com/view/feature/130734/game_feel_the_secret_ingredient.php)

*protagonista!*” en un videojuego la reacción más inmediata sería la de “*¡Me han matado!*”, a pesar de que, evidentemente, eso no es verdad.

En un juego de carreras muchos jugadores se inclinan a los lados al girar o aprietan más el gatillo para acelerar más deprisa de manera instintiva, aunque saben que no tiene ningún resultado. Sentimos los golpes que recibe nuestro personaje y nos tomamos las derrotas de manera personal. Cuando comentamos una partida a un amigo lo hacemos en primera persona, ya que sentimos que la aventura la hemos vivido nosotros mismos.

Estas sensaciones son provocadas por la respuesta del videojuego hacia nuestras acciones en el mundo real, como interaccionan los diferentes elementos del juego a consecuencia de haber pulsado un botón con cierto objetivo. Son importantes, pues, los periféricos usados para interactuar con el juego, ya sea mando o ratón y teclado, así como una asignación de controles que se siente natural para el usuario. Además, la respuesta a las pulsaciones o cambios de estado en estos controladores tiene que ser adecuada, así como su sensibilidad hacia ellos, ya sea por un correcto retardo o por la propia interacción física de los elementos del videojuego.

El modo en el que se siente el que un personaje realice un salto puede diferir mucho entre varios títulos. En algunos videojuegos puede parecer que el personaje flota, en otros sentirse más pesado y en otro totalmente erróneo si las físicas son extrañas. Acciones simples como que al arrastrar un objeto pesado el personaje avance más lento hacen que el jugador pueda imaginarse una sensación de esfuerzo.

Si además acompañamos a estas reacciones físicas elementos visuales o sonidos, estas sensaciones se pueden mejorar. Ralentizar ligeramente la ejecución del juego cuando alcanzamos a un enemigo o sacudir la imagen de la cámara del juego pueden hacer que los impactos se sientan más reales. Por otro lado, si el sonido de los pasos cuando el personaje anda sobre agua suena distinto al del personaje caminando sobre tierra, ayudamos a que ambos terrenos se sientan diferentes.

Lograr esta inmersión del jugador en nuestro mundo virtual no es sencillo, ya que se

trata de un área muy subjetiva. Es necesario que todos estos detalles estén relacionados con el contexto del juego y con, si se aplica, la realidad que se representa en el mismo. De nada sirve unas mecánicas de salto muy trabajadas si en el transcurso de la aventura esta acción sólo es necesaria en contadas ocasiones. El propio entorno, contexto y jugabilidad del videojuego deben fomentar que se realicen estas acciones. Por otro lado, un jugador quizás pueda imaginarse cómo se comporta y conduce un avión, pero no como se monta un dragón, por lo que en el primer caso las mecánicas se deben ajustar a unas expectativas preconcebidas.

Al igual que muchas otras áreas del estudio del videojuego, el “*Game Feel*” es un término joven que no llega al decenio; Una muestra más de que es ahora cuando el potencial como medio de los videojuegos se empieza a vislumbrar, a dedicar esfuerzos en desentrañar el porqué de las reacciones que provocan en las personas y en cómo se pueden canalizar esas reacciones en vistas a mejorar nuestra sociedad.

### 1.3 PROCESO DE DESARROLLO DE UN VIDEOJUEGO Y ACTORES QUE PARTICIPAN EN EL MISMO

La creación de un videojuego es una tarea cada día más compleja. Lejos quedaron los días en los que eran desarrollados por equipos de pocas personas y con presupuestos reducidos. En la mayoría de casos, desarrollar un videojuego ya no es sólo un hobby, es una actividad comercial.

En 2016 la industria del videojuego movió 99,6 billones de dólares, una tendencia al alza que no apunta a descender. En este marco, los presupuestos dedicados a desarrollar un videojuego pueden llegar a ser abrumadoramente altos y, junto a este aumento de fondos, un mayor número de profesionales se ven implicados en el proceso de creación y comercialización de estas experiencias jugables. Y conforme los estudios aumentan de tamaño, se hace necesario mejorar la gestión de los recursos humanos, así como optimizar el proceso de desarrollo del videojuego.

A pesar de ello, esta afirmación no es definitiva, ya que con el auge de los videojuegos

independientes o *indie* se posibilita a estudios con pocas personas y recursos a desarrollar sus propuestas mediante una mayor accesibilidad en herramientas, formación, software y medios de publicación para comercializarlas.

Independientemente del tamaño del estudio, podríamos dividir el proceso de desarrollo en cuatro fases: Concepto, Pre producción, Producción y Post producción o post lanzamiento.

### 1.3.1 *Concepto*

La primera etapa es simple, ya que simplemente se trata del nacimiento de una idea. Ya sea respecto al género jugable o a la ambientación, es necesario comenzar con un concepto, analizar su potencial y su posible desarrollo.

Este concepto puede ser una idea completamente original o partir de uno ya existente y desarrollarlo (aunque también hay sectores que encuentran motivador copiar directamente un concepto sin aplicarle ningún añadido reseñable).

Es extraño el caso en el que surja un concepto ideal a la primera. Mediante uso de técnicas de *brainstorming*, pensamiento lateral o cualquiera otra manera de focalizar la creatividad, se generan multitud de ideas, las cuales se descartan unas en favor de otras y se entremezclan hasta dar con un concepto con el suficiente potencial creativo y comercial para justificar el desarrollo de un videojuego a su alrededor.

Independientemente de todo esto, dejar constancia escrita de todo el futuro es necesario, tanto para evitar olvidar aspectos de la idea escogida como para rescatar conceptos descartados en un futuro.

### 1.3.2 *Pre producción*

Una vez establecido el concepto inicial, es necesario desarrollarlo y ponerlo a prueba lo antes posible. En esta fase comienzan a entrar en juego diseñadores, artistas, programadores, escritores y productores.

El eje de esta fase lo constituye el **Documento de Diseño**, el *Game Design Document* o GDD. Dicho documento es la base para el futuro desarrollo; Es una síntesis de todo lo que el videojuego va a ser; el propio concepto, la historia, las mecánicas a implementar, etc.

En esta fase, el GDD es muy flexible e iterativo, dado que ideas que creíamos que podrían funcionar demuestran no hacerlo o se encuentran mejores alternativas a ellas.

Durante la pre producción se detalla por escrito la jugabilidad del videojuego: el género, las acciones que el jugador podrá realizar, los controles, las características del avatar protagonista y como se relaciona con los demás actores del título, como objetos, enemigos, obstáculos y las características de todos ellos, etc.

Todas estas mecánicas tienen que ser evaluadas por el equipo artístico, los programadores y los productores para estudiar su viabilidad jugable, técnica, temporal y económica. Se hace necesaria el prototipado de estos conceptos para ayudar a analizarlos. Con ellos, se detectan errores, mejoras y cambios que serán aplicados al GDD en un proceso iterativo hasta que se dé con una propuesta sólida.

Se generan los recursos artísticos que más tarde servirán como guía y, si es necesario, se escribe el argumento del título, se detalla el universo, los personajes y las relaciones entre ellos. Se elabora el diseño de los niveles, personajes, menús, interfaces y actores del videojuego para obtener un esquema general del videojuego en su totalidad.

Con ello se acuerdan metas y objetivos, a la vez que se establece un calendario para las siguientes fases y las limitaciones presupuestarias del proyecto. Una vez contemplados y previstos todos los agentes y escenarios que intervendrán en el desarrollo del proyecto, se avanza la siguiente fase.

### 1.3.3 Producción

Durante esta fase un mayor número de profesionales trabajan en el videojuego. Partiendo de las pautas establecidas en el documento de diseño, equipos de artistas, programadores y diseñadores trabajan juntos para hacer real el videojuego plasmado en papel. Los productores, por su parte, se encargarán de que todos los equipos trabajen de manera coordinada, de que se cumplen los objetivos y la planificación temporal, etc. Se aseguran de que todo el trabajo se ajusta a lo previamente planificado.

Aunque idealmente el documento de diseño está terminado, en compañías menos organizadas puede no ser así, por lo que está abierta la posibilidad de volver a la mesa de diseño para arreglar detalles o aspectos que, bien por omisión o por limitaciones técnicas, no se pueden ejecutar según lo establecido. Lo mismo ocurre con conceptos que no funcionan según lo esperado, a pesar de que estos problemas se deberían haber detectado en la pre producción.

Se generan los recursos gráficos que se usarán en el videojuego, tanto en dos como tres dimensiones. Se realizan ilustraciones y bocetos que servirán para el posterior modelado y animación de los personajes y objetos del título. Se modela el mundo y los niveles del videojuego, así como se programa la jugabilidad y el desarrollo del mismo. Se graban y crean los efectos de sonido y se compone su banda sonora. Además, se comienza a trabajar en el marketing del videojuego.

Conforme se va finalizando el desarrollo, llega el testeo, en el que se detectan los posibles errores en la ejecución del juego y se solucionan los posibles problemas de rendimiento del mismo y cualquier comportamiento no deseado.

Una vez que el videojuego está listo para ser jugado, se lanza al mercado.

#### 1.3.4 *Post producción*

La fase final del desarrollo de un videojuego se produce cuando este llega a las manos de los jugadores. Durante esta fase se desarrollan principalmente dos actividades: el mantenimiento del videojuego y la comercialización del mismo.

Tras la producción de un título, es necesario su mantenimiento. En este, se solucionan todos los problemas y errores que no hayan sido detectados durante el desarrollo, sino mediante el feedback de los jugadores. Idealmente estos errores se solucionan con la máxima brevedad posible mediante parches. Además, actualmente es habitual aumentar la vida útil y comercial del videojuego a través de contenidos descargables, tanto de pago como gratuitos para el jugador.

Esta fase prácticamente no existía en el pasado, ya que una vez vendido un videojuego físico era imposible gestionar una actualización o reemplazo del producto. Este servicio post venta sólo ha sido posible mediante el desarrollo de internet.

Por otro lado, en referencia a la comercialización, es en esta fase cuando se recogen los frutos del trabajo realizado. El marketing del producto es crítico en esta fase, marcando muchas veces la barrera entre éxito y fracaso del videojuego independientemente de su calidad. Es hora de analizar el desempeño económico del proyecto, darlo por concluido y, si todo ha funcionado según lo previsto, comenzar con un nuevo desarrollo.

Pero quizás este procedimiento comience en poco tiempo a ser cosa del pasado; la tendencia hacia el videojuego como servicio comienza a vislumbrarse en el horizonte. Conforme se hacen más y más habituales las nuevas estrategias de productos a largo plazo, con contenidos que se amplían de forma recurrente en el tiempo, la popularización de los micro pagos, los videojuegos con suscripción, así como proyectos de juego por streaming o catálogo mediante tarifa mensual, se hace patente que la metodología de comercialización de los videojuegos está cambiando, así como la manera de entenderlos por parte de los jugadores.

### 1.3.5 Profesionales del videojuego

Este proceso es posible gracias al trabajo de equipos de profesionales multidisciplinares. Conforme más complejos y grandes son los desarrollos, mayor número de personas con perfiles muy diferentes se ven implicadas. Aunque existen más cargos, a continuación, se listan aquellos más relevantes durante el desarrollo de un videojuego.

#### **Diseñador**

Incluye tanto a directores del videojuego, como jefes de diseño o a los diseñadores a su cargo. Se encargan de generar las ideas, normas y objetivos del videojuego. Deben poseer muchas habilidades diferentes, normalmente una mezcla de conocimientos entre diferentes campos que conforman el videojuego final. Son los encargados de saber diferencias los juegos que funcionan de los que no.

Como otras áreas, los diseñadores cada vez están más diferenciados. Podemos encontrar diseñadores de niveles, de combate, del sistema, etc. Son supervisados por el director creativo, quien se asegura que la visión inicial del videojuego no se ve alterada. Finalmente, el principal responsable de un videojuego es su director, quien también pertenece al equipo de diseño.

#### **Programador**

Son los encargados de escribir el código que sustenta el videojuego. Crean y trabajan con los motores gráficos, los cuales permiten representar en pantalla los gráficos generados por otros equipos de desarrollo, se encargan de la creación de la inteligencia artificial de los actores del videojuego, configuran la cámara desde la que el jugador observará el mundo del juego, etc.

Al igual que los diseñadores, los profesionales de este perfil también tienden hacia la especialización en un campo determinado. Así, puede haber especialistas en físicas, en infraestructuras multijugador o incluso los que sólo trabajan en áreas como el sonido.

## **Artista**

Durante las primeras etapas de la industria, eran los propios programadores los encargados de realizar los gráficos de los videojuegos. Conforme pasaron los años, se hizo patente la necesidad de disponer de artistas para elaboración de los mismos y, con ellos, llegó una obvia mejora en la calidad visual de los títulos.

Hoy en día, los artistas de un videojuego tienen roles muy diferentes entre sí. Por un lado, algunos trabajan ilustrando *concept arts*, escenarios o storyboards del argumento o cinemáticas del título. También podemos encontrar artistas especializados en pixel art para videojuegos en dos dimensiones. Por otro lado, la elaboración de gráficos en tres dimensiones necesita de sus propios profesionales especializados en diferentes campos. Modeladores 3D, diseñadores de texturas, animadores o artistas técnicos que ayudan a los demás, etc.

Todos ellos son supervisados por el director de arte, el cual se encarga de asegurarse de que todo el recurso visual alcanza la calidad marcada, de que el estilo visual se mantiene homogéneo (a pesar de haber tantos artistas diferentes implicados) y se mantiene la visión artística general del proyecto.

## **Productor**

Los productores son aquellos que se encargan de supervisar el desarrollo del videojuego, gestionando el trabajo de los miembros del estudio. En su inicio, miembros del propio equipo de desarrollo asumían el rol de productor, pero dado que las tareas del rol se han ido expandiendo ha sido necesario profesionales que se dedicaran exclusivamente a este rol.

Este rol no es asumido por una sola persona, sino que es habitual que haya varios productores trabajando en un mismo proyecto, los que son supervisados a su vez por un productor ejecutivo. Dada la amplitud de tareas que abarcan, son ayudados por asistentes de producción y productores asociados.

Sus tareas pueden ir desde asuntos triviales, como gestionar la reparación de una avería en la oficina, hasta trabajos de importancia crítica, como asegurar los plazos

de entregas, gestionar el presupuesto del proyecto, organizar las sesiones de testeo o ser el puente comunicativo entre los diferentes equipos de desarrollo. Además, es habitual que los productores actúen como la cabeza visible del proyecto y sean los encargados de tratar con la prensa y con el público.

A diferencia de los productores externos, que vienen de parte de la editora y los inversores para verificar que se hace buen uso de los fondos que han sido destinados en el estudio, el equipo de producción lo constituyen empleados de este.

Aunque sus tareas son muy diferentes a la de programadores, diseñadores y artistas, el trabajo del productor es crítico, ya que son ellos los que gestionan el correcto desarrollo del proyecto y en muchas ocasiones son el puente comunicativo entre los distintos equipos. La influencia creativa y las tareas de los productores varían mucho de una empresa a otra, ya que las estructuras organizativas y las metodologías de trabajo pueden ser muy diferentes. Independientemente de que haya uno, ninguno o varios productores, el objetivo final es que el videojuego salga a tiempo y cumpla las expectativas puestas en el mismo.

### **Compositor**

Normalmente, el compositor no forma parte del estudio que desarrolla el videojuego, sino que es subcontratado para realizar la tarea. Esta externalización u *outsourcing* es más habitual en la banda sonora del videojuego, pero también puede darse en otras áreas como escritores, artistas o programadores ya sea como apoyo o para campos de trabajo específicos.

A pesar de que las bandas sonoras de los videojuegos cada vez son más cinematográficas, el proceso de composición es muy diferente debido a que normalmente, la música de un mismo nivel se repite una y otra vez, resultando en una melodía de ambiente continua, lo que supone además un ahorro tanto en memoria como en costes de composición. Con estas limitaciones, los compositores crean piezas que sigan siendo emotivas y emocionantes.

Un estudio puede optar por usar música bajo licencia o música original para su

videojuego. En el primer caso, el coste es menor, pero se pierde la exclusividad de la música. En el segundo caso, se suele pactar un precio por minuto de composición, el cual puede variar mucho entre compositores (un precio normal estaría en torno a los 200\$ por minuto).

En todos los casos, la música debe acompañar al escenario a presentar en pantalla de la mejor manera posible, así como potenciar las emociones que se pretenden generar en cinemáticas y otras escenas relevantes para el argumento del título.

### **Editor de sonido**

El editor o diseñador de sonido es el encargado de grabar y mezclar los efectos de sonido que se utilizarán en el videojuego.

Es un campo muy creativo, ya que en muchas ocasiones los sonidos que escuchamos en un videojuego o película provienen de objetos de lo más variopinto. En algunos casos, los sonidos deben ser acordes a la acción que los desencadena: un sonido tras una acción que beneficia al jugador deberá sonar positivo y, por el contrario, un sonido de una acción dañina deberá transmitir negatividad. Debido a este hecho y a que, no es extraño tener que crear sonidos de objetos o criaturas imaginarias, es un puesto que también puede llegar a ser complejo.

### **Escritor**

A pesar de que en la actualidad los escritores están comenzando a trabajar en las primeras etapas de un proyecto, hace unos años solía, al igual que los compositores, ser incorporados en las últimas fases del mismo.

Se encargan de reescribir la historia pensada por los diseñadores, de redactar los diálogos, las interfaces, manuales y tutoriales del videojuego para hacerlos más entendibles, etc.

Conforme la narrativa se ha ido haciendo más y más importante en el medio, su labor y su influencia creativa es mayor, ya que los videojuegos se van acercando cada vez más a historias interactivas, por lo que se hace patente la importancia de este rol, ya

que escribir para un videojuego es muy diferente a escribir una novela o un guion cinematográfico. Por todas estas razones, su influencia creativa en los proyectos cada vez es mayor.

## **Marketing**

Como vender un videojuego es casi tan importante como el propio juego en sí. Esta labor la lleva a cabo el equipo de Marketing, quien promociona y da a conocer el videojuego al público.

Trabajan con los medios de comunicación, como prensa, medios digitales o televisión y cine. Elaboran tráileres y diseñan las cajas o *packaging* del videojuego, así como cualquier otro material promocional. Además, en la sociedad actual es necesario disponer de profesionales que gestionen las redes sociales y comunidades que se formen en torno al videojuego, lo cual también entraría en las competencias de los profesionales de este sector.

A pesar de que muchos responsables de marketing ni siquiera juegan al videojuego que promocionan, es tarea de los desarrolladores asegurarse de que eso sea así, ya que puede marcar la diferencia a la hora de saber comunicar y vender la idea que hay detrás de un videojuego.

## **Editor**

Los editores son las entidades que invierten dinero en el videojuego y se encargan de su publicación y comercialización. Son los responsables de muchos campos profesionales diferentes, desde asesoramiento legal hasta los fabricantes del videojuego físico y la distribución del mismo en tiendas.

Proporcionan asistencia y medios a los estudios de desarrollo de la misma manera que gestionan y supervisan la producción del videojuego.

En la actualidad el sector de los videojuegos da empleo a muchos profesionales que pertenecen a categorías que no han sido listadas, como testers, control de calidad, prensa especializada, gestores de licencias e incluso comentaristas deportivos, etc.

En los años venideros esta lista de profesionales será mucho más amplia y los roles y tareas asignadas a ellos evolucionarán y serán diferentes. No es de extrañar que ante esta demanda de profesionales la oferta formativa en los diferentes campos del videojuego sea cada vez mayor ante una demanda creciente.

Pese a ello, aún es limitada y se hace necesario la creación de nuevos planes de estudio y titulaciones que formen a los profesionales que trabajarán en esta industria que cada día tiene más impacto en nuestro estilo de vida y nuestra sociedad.

## 2. METODOLOGÍAS LEAN

Las metodologías *lean* nacen de la necesidad de optimizar los procesos comerciales ante un mundo cada vez más globalizado en el que la competitividad en distintos sectores de negocio es altísima.

Variables como el *time to market*, el tiempo que tarda un producto desde es ideado hasta que se pone en venta, la sobreproducción o la gestión de productos defectuosos pueden ser determinantes para determinar el éxito de una propuesta de negocio.

Manejarlas todas ellas es complicado, ya que, dada su propia naturaleza, cualquier actividad comercial está rodeada de grandes cantidades de incertidumbre. Optimizar las gestiones y procesos e intentar reducir esa incertidumbre al máximo posible se hace necesario para lanzar negocios viables al mercado.

Con este objetivo, han sido planteadas numerosas metodologías de las cuales en este proyecto se estudiarán dos: *Lean Manufacturing* y *Lean Startup*.

### 2.1 LEAN MANUFACTURING

Los principios de Lean Manufacturing surgen a principios del siglo XX en la industria del automóvil<sup>20</sup>. En el auge de la producción industrial, el sector automovilístico comenzó a producir vehículos en masa en cadenas de ensamblaje cada vez más complejas con maquinaria cada vez más específica. Sin embargo, estos procesos producían un único modelo de producto, y los clientes demandaban un servicio personalizable, ya fuera la posibilidad de elegir el color del vehículo como sus características.

Ante esta demanda, se invirtió en maquinaria cada vez más grande, más potente y más rápida. A pesar de ello, los tiempos de producción seguían aumentando de la misma manera que lo hacía el inventario. Los tiempos entre que una pieza pasaba

---

20 A Brief History of Lean

<https://www.lean.org/WhatsLean/History.cfm>

de un proceso a otro también se alargaron conforme las partes del vehículo eran más complejas. Muchas fábricas siguieron utilizando la misma metodología que venían haciendo hasta ese momento, alimentando este problema en un ciclo que no era solucionado. Fue tras la Segunda Guerra Mundial, en un periodo de escasez de recursos, de inversión y de espacio, cuando el Grupo Toyota decide centrar su estrategia en afrontar este crucial problema en un sector muy competitivo.

Con el objetivo de optimizar al máximo los procesos y los recursos disponibles, se crean diferentes herramientas que darán forma al Toyota Production System (TPS), el que más tarde sería conocido por diferentes términos como World Class Manufacturing (WCM), Lean Sigma, Continuous Flow Manufacturing o Lean Manufacturing.

### *2.1.1 PRINCIPIOS DE LEAN MANUFACTURING*

Lean Manufacturing se basa en diferentes sistemas de producción, resultando la suma de ellos en esta metodología.

Sus principales objetivos son la satisfacción del cliente y que garantizar la rentabilidad de proveerla. Cualquier cosa que no aporte valor de cara al cliente se considera desperdicio, por lo que se debe cuestionar su lugar en el proceso.

Este cambio en la mentalidad tiene su origen en los propios trabajadores, ya que el factor humano es uno de los pilares fundamentales de Lean Manufacturing.

#### *2.1.1.1 El Kaizen o Mejora Continua*

El proceso de Mejora Continua tiene su origen en el concepto japonés Kaizen<sup>21</sup>. El Kaizen consiste en el autodesarrollo personal, en la creencia de que siempre hay margen de mejora. Este beneficio es sustentable y nunca viene de terceros, si no de uno mismo.

---

<sup>21</sup> What is Kaizen?

<http://leanmanufacturingtools.org/621/what-is-kaizen/>

De la misma manera, los procesos industriales nunca son perfectos y las operaciones que los componen deben ser mejoradas constantemente a través de la innovación y de su propia evolución.

Esta posibilidad de mejora es detectada por los propios trabajadores, por lo que es necesario que los objetivos y la metodología empresarial este en concordancia con la capacidad e inquietudes de sus empleados. Todos los actores del proceso de producción deben estar implicados en la filosofía de la mejora continua.

Es necesario estudiar el sistema en profundidad para conocer sus debilidades. Esto se hace aplicando sistemas de reingeniería de procesos y principios de la Teoría de las Restricciones (Theory of Constraints, TOC).

Según esta, en todo sistema productivo siempre existe, como mínimo, un cuello de botella. El conocimiento de este eslabón débil de la cadena es determinante, ya que las actuaciones que sirvan para mejorarlo repercutirán positivamente a todo el proceso en su totalidad. Si, por el contrario, se mejorara otro subproceso, este cambio no repercutiría al proceso final dado que este siempre está limitado por el cuello de botella.

Una vez detectado este eslabón, se podría optar por destinar más recursos para su mejora y, como hemos dicho, todo el sistema se vería beneficiado. Sin embargo, si analizamos su funcionamiento y exploramos otras vías de mejora, podríamos llegar a una solución en la que, destinando los mismos recursos que al principio se mejore de igual manera el rendimiento. Este principio es lo que se denomina la reingeniería de procesos: Se mira más allá de lo establecido y se replantea el propio funcionamiento del proceso para buscar vías de mejora mediante medidas que no habían sido planteadas antes, lo que podría resultar incluso en la propia sustitución de un proceso por un equivalente con mayor eficiencia.

Estas medidas sólo pueden darse en un ambiente de trabajo en el que exista una cultura hacia la mejora. Para ello, la propia organización y gestión del personal y del entorno de trabajo deberá adaptarse a este contexto.

### 2.1.1.2 Las 5S

El método de las 5S es un programa orientado a conseguir mejoras en el nivel de organización y limpieza del entorno de trabajo, aumentando así las condiciones de trabajo, la motivación de los empleados y la operatividad del proceso<sup>22</sup>. Estas cinco etapas son:

#### **Seiri (Clasificación)**

En esta primera fase, se identifican los elementos innecesarios para el proceso y son desechados, asegurándose de que no vuelven a aparecer. Tras analizar y clasificar los elementos según la frecuencia de su uso, lo que se utiliza menos de una vez por año es desechado (atendiendo también al coste de su reposición en caso de que sea necesario de manera anómala).

#### **Seiton (Organización)**

Consiste en decidir la manera en la que se situarán los elementos necesarios atendiendo a la clasificación realizada en la fase anterior. Se utilizan técnicas de comunicación visual, como delimitaciones con líneas, letreros, siluetas de colores...para que esta organización sea sencilla de interpretar por los empleados.

Atendiendo a la clasificación anterior, se podría determinar que lo que se use menos de una vez al mes se aparta, se almacena. Lo que se usa menos de una vez a la semana se aparta a poca distancia, dentro de la oficina. Lo que se usa menos de una vez por día se deja en el puesto de trabajo, lo que se usa menos de una vez por hora tiene que estar al alcance de la mano, colocándose todo lo que se usa al menos una vez por hora directamente sobre el operario.

De cualquier manera, las reglas de ordenamiento y colocación de objetos tienen que ser obvias para los trabajadores.

---

<sup>22</sup> What Is 5S? - Matt Wastradowski

<https://www.graphicproducts.com/articles/what-is-5s/>

### **Seisō (Limpieza)**

La limpieza sistematizada se lleva a cabo una vez clasificado y ordenado el espacio de trabajo. Se identifican y eliminan las fuentes de suciedad, procurando que no vuelvan a aparecer. De esta manera se hace un mantenimiento constante de las herramientas y se mantienen en un estado operativo perfecto, previniendo así también posibles errores de funcionamiento y facilitando la inspección y detección de anomalías.

### **Seiketsu (Estandarización)**

Una vez detectada una anomalía, saber cómo actuar ante ella con rapidez es fundamental. De la misma manera, saber qué se debe de hacer ante una irregularidad en el puesto de trabajo es importante.

En esta fase, se establecen unas normas sencillas y conocidas por todo el equipo de trabajo, de manera que se estandarizan los procedimientos de actuación ante la aparición de situaciones imprevistas. Para ello, es necesario formar al personal en los estándares establecidos a la vez que mantener actualizada y documentada la lista de procedimientos estándar ante anomalías.

### **Shitsuke (Disciplina)**

La última etapa consiste en asegurarse de que las acciones realizadas se mantienen en el tiempo, al igual que se plantean campos de mejora de manera continua. Para ello, es necesario hacer un riguroso control del funcionamiento del sistema y comprobar que se está aplicando correctamente.

Los beneficios de la aplicación de las 5S son inmediatos. Por una parte, el empleado desarrolla un sentido de pertenencia al espacio de trabajo y se encuentra más motivado al ser implicado en la organización del mismo. Además, las condiciones de seguridad laboral son mejores, ya que en cuando se detecta una irregularidad esta es solucionada, evitando posibles futuros accidentes laborales. También, al aplicar un mantenimiento continuo a las herramientas y equipos se incrementa su vida útil y se

evitan largos periodos de reparaciones y mantenimiento. Todo ello lleva, en definitiva, a la elaboración de productos de mayor calidad.

#### 2.1.1.3 Gestión de la Calidad Total (TQM)

La necesidad de la aplicación de medidas de mejora continua en el rendimiento de los procesos y la optimización de los procesos internos ha generado el concepto de Gestión de la Calidad Total.

Con *Calidad Total* no solamente se abarca la calidad y eficiencia del producto final que llega al cliente, sino que nos referimos a la calidad de toda la cadena productiva y procesos implicados a la hora de ideas, fabricar y poner en venta un producto. Así, se busca la eficiencia total, la calidad perfecta a la primera, evitando defectos y detectando los problemas en su origen.

Se busca la eliminación total de todos los desperdicios<sup>23</sup> que se generan durante la producción. Podríamos clasificar los tipos de desperdicio como:

#### **Transporte**

El primer tipo de desperdicio consiste en el transporte de elementos, herramientas o productos de manera innecesaria entre subprocesos.

#### **Inventario**

Cualquier componente terminado o en proceso que no está siendo usado en cierto momento, bien porque sea innecesario o porque se utiliza en un subproceso que supone un cuello de botella y este no puede procesarlos al mismo ritmo que en el que se generan. Esta situación supone un desperdicio de espacio y estimula daños y obsolescencia en los componentes o productos.

---

23 The Seven Wastes | 7 Mudas

<http://leanmanufacturingtools.org/77/the-seven-wastes-7-mudas/>

### **Movimiento**

Se refiere al movimiento innecesario de personas y máquinas a la hora de realizar un proceso. Se hace necesario optimizar la organización espacial del lugar de trabajo para que los movimientos necesarios sean mínimos.

### **Espera**

Los tiempos de inactividad entre un proceso y el siguiente. También se incluyen las interrupciones en la producción debidas a cualquier causa.

### **Sobreproducción**

Uno de los desperdicios que más daño pueden hacer al negocio. Es causado por la elaboración del producto antes que su demanda. Esto, al igual que los desperdicios de inventario, lleva al almacenaje innecesario de productos que puede que nunca se lleguen a poner en venta. Los recursos generados a estos productos no producen beneficio.

### **Procesado extra**

Se considera procesado extra todo aquel trabajo causado por defectos en los productos, por errores de diseño, por movimiento, manejo de materiales innecesarios debidos a la sobreproducción o un inventario insuficiente y, en general, cualquier trabajo no contemplado en el proceso o repetido.

### **Defectos**

Otro desperdicio es el tiempo de trabajo y el esfuerzo dedicados a detectar y corregir defectos y características no deseadas.

### **Habilidades**

Se refiere a la infrautilización de las capacidades de los empleados o al talento inutilizado. También engloba a las tareas delegadas a empleados que no tienen el entrenamiento necesario para llevarlo a cabo de manera óptima.

Estos desperdicios son sólo una guía; haciendo uso de las herramientas de análisis de problemas y mejora continua, nuevos tipos de desperdicio pueden ser detectados en cierta actividad comercial. Por otra parte, también podríamos considerar como desperdicio cualquier característica del producto a la que se han destinado recursos que no sea demandada por los clientes, ya que al utilizar *Lean Manufacturing* el cliente es el extremo desde el que se genera la cadena de producción y no el fabricante

#### 2.1.1.4 El Flujo Tirado o Pull System

En *Lean Manufacturing*, el grado de eficiencia viene marcado por el cumplimiento de las expectativas del cliente. Esto supone un cambio respecto a la industria tradicional en la que el objetivo era producir y la eficiencia venía guiada por la capacidad productiva del proceso, independientemente de la demanda del cliente. Este proceso es denominado *Flujo Empujado*, ya que, sin importar la señal del cliente, el producto va siendo producido y almacenado hasta su puesta en venta. Esto puede ocasionar situaciones fuera de control y complicaciones en la logística del negocio.

Si, por el contrario, la producción comienza cuando el cliente demanda un producto y son las últimas etapas de la producción las que solicitan los componentes necesarios a la anterior, estaremos haciendo uso del Flujo Tirado.

A pesar de que las ventajas de este modelo de producción son fácilmente identificables, se hace necesario centrar esfuerzos en mejorar la cadena de comunicación y la transmisión de información desde el cliente a través de la cadena de producción hasta el propio proveedor de las materias primas.

Este proceso puede llegar a ser complicado ya que entran en juego factores como la logística de proveedores, los materiales necesarios, la logística interna de la empresa, la secuenciación de los subprocesos y las propias herramientas de comunicación entre los mismos.

El objetivo final de este proceso no es otro que ofrecer, no un producto, si no la solución que el cliente necesita cuando él quiere.

#### 2.1.1.5 Just In Time (JIT) y Kanban

Se entiende como *Just In Time*<sup>24</sup> la eliminación de cualquier desperdicio en el proceso de producción, desde las compras de los materiales y recursos hasta la comercialización del producto.

Se basa en que tanto los componentes intermedios como el producto final deben estar acabados justo cuando se necesitan y no antes (ni después). Además, de la misma manera, se debe asegurar la calidad del producto y el cumplimiento de las expectativas del cliente.

Mediante el uso de la filosofía Just In Time se reducen los niveles de stock, al producir justo la cantidad que el cliente demanda. Por otra parte, al reducir también los inventarios de los productos intermedios, se detectan los cuellos de botella más fácilmente y se hace más fácil mejorarlos. Mediante ello, se simplifican las tareas administrativas de gestión de recursos y el flujo de la producción es más equilibrado que en sistemas de *Flujo Empujado*.

Así, podemos considerar un flujo físico que tira del material a través de la línea productiva utilizando el tamaño de lotes de producción más pequeño posible. El ritmo de este flujo lo marca la propia velocidad de consumo de los clientes.

Como se ha indicado anteriormente, es necesario mejorar también la comunicación y gestión de las necesidades entre los diferentes subprocesos. Para ello, se utiliza la metodología Kanban.

**Kanban** es una metodología de comunicación visual para controlar el estado de diferentes procesos<sup>25</sup>. Básicamente, se trata de un sistema de señales que se utiliza

---

24 Just in Time (JIT) Production. What is Just in Time?

<http://leanmanufacturingtools.org/just-in-time-jit-production/>

25 What is Kanban?

<http://leanmanufacturingtools.org/kanban/>

a través de la cadena de producción, desde el extremo del cliente hasta las materias primas. El formato de estas etiquetas o tarjetas puede variar mucho y, en ocasiones, también van acompañadas o son sustituidas por un tablero Kanban, en el cual se representan de manera visual entre tarjetas las relaciones y dependencias entre procesos al igual que se indica su estado actual. El uso de estas tarjetas Kanban ayuda a los equipos a comprender rápidamente los detalles del trabajo a la vez que facilita la comunicación entre ellos.

Las seis reglas básicas de Kanban son las siguientes.

- 1) Los procesos posteriores recogen los materiales del proceso anterior indicados por el Kanban.
- 2) Los procesos anteriores producen los materiales en la cantidad y secuencia indicada por el Kanban.
- 3) Ningún material es transportado ni manipulado sin una autorización o tarjeta del Kanban.
- 4) Todos los componentes de la cadena de montaje deben tener una etiqueta Kanban.
- 5) Los componentes defectuosos no son enviados a los procesos posteriores. La cadena de producción se para hasta que se soluciona el problema.
- 6) Si el sistema tiene menos fases Kanban, es más sensible ante posibles problemas o equivocaciones en la utilización del sistema por parte de los empleados.

### *2.1.2 BENEFICIOS ESPERADOS Y COMO APLICARLO*

Cuando una empresa adopta la filosofía Lean, los beneficios pasan a experimentarse en todos los aspectos del negocio.

Por una parte, se incrementa la productividad, la calidad del producto final, las ventas y, por lo tanto, las ganancias y el valor total de la empresa. De la misma manera, se reduce el inventario, el plazo de entrega y los costes de producción.

A pesar de ello, la implantación de esta metodología no es algo que se pueda hacer de la noche a la mañana. Requiere de un proceso en el que, los principales implicados y comprometidos deben ser los miembros del equipo de dirección y promoción de la empresa. Esto en ocasiones puede ser muy difícil en negocios de antigüedad en los que la resistencia al cambio sea muy fuerte y los hábitos de trabajo no han variado a lo largo de su historia en el mercado. Este compromiso en la dirección de la empresa debe ser transmitido a todos los trabajadores, sin importar su posición jerárquica en la misma

Para ello, se debe invertir en la formación de los empleados e incluso modificar la organización interna de la empresa. Es inviable realizar estos cambios de manera inmediata, por lo que se puede optar por adoptar esta filosofía de manera secuencial por diferentes departamentos, analizando y comparando los resultados obtenidos por estos y evaluando cómo se comportan ante esta nueva dinámica de trabajo.

Sin duda, esto supone un coste a la empresa, ya que es necesario adoptar una nueva mentalidad, con las consecuencias que ello conlleva. Resulta fundamental que esta predisposición al cambio sea compartida por todos los empleados de la empresa, ya que la comunicación debe ser inmediata y los objetivos de todos los actores implicados deben estar alineados.

Aun así, por su propia naturaleza, una empresa nunca puede llegar a ser lean en su totalidad; Por una parte, se considera que siempre hay un margen de mejora sobre el que es posible actuar. Por otro lado, se basa en la demanda de los clientes, y estas cambian con el tiempo a un ritmo que no para de aumentar en la sociedad actual.

## 2.2 LEAN STARTUP

Debido a su carácter industrial, Lean Manufacturing evolucionó hasta formar la filosofía de Gestión Lean o Lean Management. Esta evolución está destinada a mejorar la eficiencia de empresas mediante el uso de la producción limpia en negocios de todos los sectores y tamaños (Lean Manufacturing también está más bien enfocada a empresas grandes).

De la aplicación de técnicas de Lean Management en empresas tecnológicas que buscaban introducir nuevos productos al mercado se desarrolló lo que hoy conocemos como Lean Startup. Desde su origen en 2008 en Silicon Valley, la popularidad de esta metodología y, a día de hoy, es utilizada por muchas startups de todo el mundo.

Así, con el mismo objetivo de aprovechar los recursos al máximo desechando despilfarros y de aumentar la productividad y la rentabilidad de los negocios, Lean Startup, sin embargo, está enfocado a las primeras fases del proyecto, en la génesis de la idea de negocio y en su proceso de desarrollo<sup>26</sup>.

Podríamos definir una startup como una organización temporal de personas que están buscando un modelo de negocio escalable de manera recurrente. Esto supone una diferencia fundamental respecto a empresas ya consolidadas, que ejecutan un modelo de negocio ya conocido.

Por lo tanto, manteniendo el centro de la propuesta comercial en el cliente, con la misma filosofía de que no se le ofrece un producto, sino que se le facilita una solución a un problema, Lean Startup pretende reducir la incertidumbre de una propuesta de negocio en lugar de centrarse tanto en optimizar el proceso de producción.

Esto se lleva a cabo mediante la supresión de prácticas que no contribuyen al producto final, el diseño de una propuesta de valor potente de manera ágil y siendo el feedback de los clientes una parte integral del proceso, para asegurarse que de no se invierte tiempo y recursos en productos que nadie quiere. Se hace uso del aprendizaje validado y la experimentación científica a la hora de lanzar los negocios, sustituyendo el concepto de mejora continua por aprendizaje continuo.

Así, se trabaja en un ciclo iterativo con tres fases: Construir, Medir y Aprender. En la primera fase, se convierten las ideas en productos, mientras que en la segunda se

---

26 The Lean Startup Methodology  
<http://theleanstartup.com/principles>

estudia la respuesta de los clientes. Finalmente, con el conocimiento adquirido se toman las medidas necesarias, tanto mejorando el producto como cambiando la propuesta comercial por completo. Este es el concepto de **Pivote**, que se define como *“una corrección estructurada de la ruta de trabajo diseñada para validar una nueva hipótesis fundamental sobre el producto o la estrategia comercial”*.

Hemos hablado de la existencia de una idea comercial y de unas hipótesis, pero ¿cómo se generan estas?

### 2.2.1 EL DISEÑO DEL MODELO DE NEGOCIO

Toda actividad comercial comienza con una idea. Muchas empresas fracasan directamente porque su idea de negocio no tiene valor para los clientes que habían previsto.

Encontrar una oportunidad de negocio sostenible en el tiempo puede ser una tarea muy complicada. La creatividad, núcleo de esta primera etapa, siempre debe estar subordinada a las necesidades de los clientes. Se debe pensar en los problemas que se quieren solucionar y en las personas que los padecen.

Tras este análisis y proceso creativo, se obtiene una idea, la cual se tiene que empezar a estudiar y desarrollar. Esto se hace con la ayuda de unas herramientas conocidas como Business Model Environment Map (BMEM), Business Model Canvas (BMC) y Value Proposition Canvas (VPC). Aunque no estaban incluidas en la primera definición de Lean Startup, hoy en día constituyen una parte fundamental de este marco metodológico.

Estas herramientas apuestan, al igual que Kanban, por el lenguaje visual para facilitar el análisis y la comprensión de los conceptos que se presentan. En ellas, se representan diversos aspectos de la propuesta de negocio a diferentes niveles.

### 2.2.1.1 Business Model Environment Map

Esta herramienta se encarga de facilitar a los emprendedores el conocimiento del entorno en el que se moverá su propuesta de negocio.

De manera visual y estructurada, se representan aspectos que pueden suponer restricciones u oportunidades en la comercialización de la propuesta, haciendo más sencillo el establecer patrones y asociaciones facilitando la generación de nuevas ideas. El BMEM está organizado en cuatro áreas, una representación del mismo se puede observar en el **Anexo 1.A.** (*Extraído de Knowledge.li y basado en Generación de Modelos de Negocio, por Alexander Osterwalder y Yves Pigneur*)

Estas áreas son:

#### **Fuerzas del Mercado**

Son los problemas claves de los clientes del sector del proyecto, como los problemas que los afectan y las soluciones que esperan. Se listan también sus hábitos de consumo y cualquier dificultad existente para llegar a ellos. Se detallan los segmentos de clientes, así como las necesidades específicas de cada uno de ellos.

#### **Tendencias Clave**

Principales tendencias del sector, desde cambios en el marco legal del sector hasta innovaciones regulatorias o tendencias sociales y culturales.

#### **Fuerzas de la industria**

Se trata de los agentes externos al proyecto que resultan clave para el sector. Se incluyen los competidores, los proveedores de materias primas o tecnología y cualquier entidad que actúe en la cadena de valor del proyecto.

#### **Tendencias Macroeconómicas**

Se listan también las tendencias económicas globales, desde la disponibilidad y coste de los recursos hasta las condiciones del mercado o el acceso a inversión.

### 2.2.1.2 Business Model Canvas

El lienzo del modelo de negocio es una herramienta visual que permite diseñar, describir y analizar de forma ágil los diferentes actores que intervienen en un modelo de negocio. Se divide en 9 módulos divididos en 4 categorías: Clientes, Oferta, Infraestructura y Viabilidad económica. Se puede encontrar una plantilla del mismo en el **Anexo 1.B.** (*Extraído de Strategyzer.com y Generación de Modelos de Negocio, por Alexander Osterwalder y Yves Pigneur*). Los nueve módulos son:

#### **1) Segmentos de clientes**

En este módulo se definen los distintos tipos de clientes a los que se enfoca la propuesta comercial. Se describen los segmentos de mercado, así como se clasifican según criterios geográficos, demográficos, psicográficos, socioeconómicos y conductuales.

#### **2) Propuesta de valor**

En este módulo se aborda, en un nivel de profundidad menor al del *Value Proposition Canvas*, los elementos que aportan valor a la propuesta comercial. Se especifican las razones por las que el producto es superior al de la competencia y cualquier aspecto que lo dote de valor añadido.

#### **3) Canal de distribución**

Se especifican también los canales de comercialización, distribución y venta de la propuesta comercial. Se indican todos los actores de la cadena de distribución, desde equipos internos hasta socios comerciales.

#### **4) Relación con los clientes**

Es necesario establecer también unas vías de comunicación con los clientes y decidir qué relación se va a tener con ellos. Se pueden plantear estrategias de captación y fidelización de clientes, así como las herramientas que serán usadas para establecer estos canales de comunicación.

## **5) Fuentes de ingresos**

Se detallan las fuentes de ingresos, las vías por la que la propuesta comercial obtendrá beneficios. Pueden ser por pagos puntuales, como ventas, alquileres o cuotas de uso o pagos recurrentes, como los recibidos por cuotas de suscripción. Un modelo de negocio no tiene por qué limitarse a una sola fuente de ingresos.

## **6) Recursos clave**

Se recogen y listan todos los recursos necesarios para llevar a cabo la propuesta comercial. Estos pueden ser físicos, económicos, intelectuales, recursos humanos, de publicidad, etc. Todos estos recursos deberán estar a disposición de la empresa para poder ejecutar su plan de negocio.

## **7) Actividades clave**

En este módulo se analizan las actividades necesarias para llevar a cabo la propuesta comercial. Estas actividades varían mucho de una empresa a otra según el sector, pueden abarcar desde procesos de producción hasta servicios de asistencia al cliente o gestión de plataformas online.

## **8) Socios clave**

Se tienen en cuenta también todos los socios clave sin los que el funcionamiento del modelo de negocio se vería afectado. Se incluyen en este módulo proveedores de productos o de servicios, alianzas comerciales, acuerdos de colaboración, etc.

## **9) Estructura de costes**

Finalmente, se calculan y analizan todos los factores que suponen un coste para la ejecución del modelo de negocio, ya vengan por la necesidad de unos recursos clave, los costes asociados a las actividades clave u ocasionados por los socios clave.

### 2.2.1.3 Value Proposition Canvas

El Value Proposition Canvas es una ampliación extraída del propio Business Model Canvas. Engloba tanto al módulo de Segmentos de clientes como al de Propuesta de valor, detallando en mayor profundidad como se crea el valor añadido de cara a los clientes. Estos dos módulos, con tres submódulos cada uno, se enfrentan y comparan para llegar a un *encaje*, estado alcanzado cuando los clientes se ilusionan con la propuesta de valor ya que se adapta perfectamente a sus necesidades y problemas. Se puede encontrar una plantilla del mismo en el **Anexo 1.C.** (*Extraído de Strategyzer.com y Diseñando la propuesta de valor, por Alexander Osterwalder y Yves Pigneur*)

En primer lugar, se observa a los clientes y se les intenta comprender. Esto se especifica en el módulo de **Perfil o segmento del cliente**. Este módulo se divide en Trabajos, Frustraciones y Alegrías.

#### **Trabajos del cliente**

Describen aquello que el cliente intenta resolver en su vida personal, laboral o en cualquier otro ámbito.

#### **Alegrías**

Describen los resultados que los clientes esperan conseguir y los beneficios que buscan.

#### **Frustraciones**

Describen los malos resultados, los riesgos o los obstáculos que les afectan a la hora de tratar de resolver el problema

Por otro lado, en el **Mapa de la propuesta de Valor** se describe con mayor detalle las características de la propuesta de valor del modelo de negocio. Este módulo se divide en Productos y Servicios, Aliviadores de Frustraciones y Creadores de Alegrías.

#### **Productos y Servicios**

Un listado de todos los Productos y Servicios a partir de los que se construye la propuesta de valor.

## **Aliviadores de Frustraciones**

Describen cómo los productos y servicios aplacan o disminuyen las frustraciones de los clientes.

## **Creadores de alegrías**

Se indican como los Productos y Servicios consiguen los resultados que el cliente espera y crean los beneficios que busca.

Cuando el Mapa de Valor coincide con el Perfil de Cliente, se considera que se alcanza un encaje y la oferta de la propuesta de valor encaja con las demandas de los clientes.

### [2.2.1.4 Identificar los problemas. Planteamiento y validación de hipótesis](#)

Una vez analizado y estudiado la supuesta viabilidad de la propuesta comercial, es momento de entrar en el ciclo planteado por Lean Startup.

Sin embargo, hay un paso más que debemos hacer antes siquiera de empezar a trabajar en la idea. Uno de los principios del desarrollo de clientes es la necesidad de salir a la calle, de contactar con los potenciales clientes para encontrar hechos que sustenten la idea de negocio, ya que dentro del propio espacio de trabajo no se encontrará ninguno.

En la fase actual, nuestra idea se basa en meras conjeturas sin ninguna garantía sobre su viabilidad, por lo que se asumiría un gran riesgo si las tomáramos como correctas. Para ello, se comenzará planteando una lista de hipótesis, suposiciones de algo que puede o no ser posible basados en la información previa de la que disponemos. Esta lista de hipótesis puede ser extensa, abarcando campos como el modelo de negocio o los propios clientes.

Una vez elaborada esta lista de hipótesis, se deberá clasificar todas ellas en función de su incertidumbre y de la criticidad para la viabilidad del proyecto. A mayor criticidad e incertidumbre, la necesidad de probarla con rapidez será mayor.

Así, una vez realizada esta lista de hipótesis clasificadas según su importancia será necesario contactar con posibles clientes u otros actores implicados en el modelo de negocio para probarlas. Es necesario resaltar que, en este punto, solamente estamos poniendo a prueba las suposiciones que hemos hecho respecto a un problema, por lo que cualquier referencia a la solución que se tiene ideada para resolver el mismo debe ser evitada (este punto se abordará en la siguiente fase del proyecto). En estas entrevistas, sólo se deberá comprobar la validez de las hipótesis redactadas en torno a un problema, las dificultades que tienen para resolverlo y los beneficios que esperan.

Puede darse el caso de que no obtengamos los resultados que esperábamos. Esto, pese a lo que pudiera parecer en un principio, no es negativo, ya que sirve para evitar esfuerzos dedicados a una idea que nadie quiere. Será el momento de volver a meditar sobre el problema y sus posibles soluciones, realizar ajustes en la propuesta o incluso descartar la idea totalmente.

Una vez se llegue a una idea cuyas bases quedan validadas por esta primera prueba, es momento de pasar a la siguiente etapa y comenzar a desarrollar la solución propuesta al problema que se ha determinado.

### 2.2.2 PRODUCTO MÍNIMO VIABLE

Tras validar el problema y ajustar el modelo de negocio, es necesario comprobar la validez de la solución planteada. Siguiendo la filosofía lean, esta validación se debe hacer de manera ágil y evitando cualquier despilfarro.

Surge el concepto de Producto Mínimo Viable (PMV). Un Producto Mínimo Viable no es más que *un experimento que permite a la startup recolectar, con el menor esfuerzo posible, la máxima cantidad de conocimiento validado* sobre sus potenciales clientes y la efectividad de su producto o servicio. Este producto mínimo no es siquiera un prototipo de la solución, es simplemente “*algo*” representativo que pueda servir para abstraer la funcionalidad básica de la solución propuesta.

Se puede optar por realizar una presentación, diagrama o un storyboard que aborde el proceso de interacción entre los clientes y la solución propuesta. También se puede realizar un vídeo o realizar un experimento interactivo con materiales tan básicos como el cartón u aplicaciones para simular el funcionamiento de la solución final.

De cualquier manera, este Producto Mínimo Viable debe ser realizado con la mínima inversión, tanto en trabajo como en recursos, posible y de manera rápida y ágil.

Una vez construido el PMV, es hora de diseñar experimentos que sirvan para estudiar los efectos de la solución propuesta en los clientes potenciales.

Estos experimentos pueden ser muy variados. Se pueden realizar nuevas entrevistas donde se enseña a los clientes el producto mínimo desarrollado, se pueden exponer los conceptos al público general y estudiar sus reacciones e incluso se pueden diseñar campañas de financiación o poner la solución a disposición del mercado en una escala reducida y con funcionalidades limitadas.

En cualquier caso, atendiendo a la fidelidad del PMV respecto a la solución final, se deberán analizar y medir los datos obtenidos mediante el comportamiento de los clientes potenciales. Se deben estudiar el interés y la relevancia de la solución, así como las prioridades y preferencias de los clientes. Esto se realiza mediante la descripción detallada de los experimentos a realizar a la vez que se establecen indicadores que nos sirvan para evaluar el resultado de manera objetiva y rigurosa.

De la misma manera que en la etapa anterior, si la solución no obtiene la respuesta esperada, será necesario reevaluarla, pivotar o incluso desechar la idea. Si la respuesta queda validada, se podrá seguir avanzando en el desarrollo del proyecto y comenzar a iterar hasta encontrar el modelo de negocio de la entidad.

### 2.2.3 CREAR – MEDIR - APRENDER

Una vez identificada la supuesta validación de la solución por parte del cliente, será necesario ponerla a prueba de manera más concienzuda. Se entra en un ciclo de aprendizaje continuo en el que, tras plantear nuevas hipótesis y documentar el aprendizaje que hemos obtenido, se siguen realizando ajustes en el Business Model Canvas y se vuelven a poner a prueba de manera iterativa.

Es importante encontrar a los **early adopters**, a los clientes que, se ven tan afectados por el problema al que se pone solución, que son capaces de obviar las barreras iniciales que tiene el producto (defectos o un alto coste, por ejemplo). Ellos actuarán como evangelizadores de la solución para un mayor número de clientes, que ofrecen mayor resistencia.

Se debe encontrar también el **Segmento Mínimo Viable** (SMV) de la solución, el segmento de mercado lo suficientemente grande como para sustentar la propuesta comercial en sus etapas iniciales a la vez que es lo suficientemente pequeño como para poder controlarlo. Cuando las necesidades de este SMV coincidan con las soluciones del MPV, se pueden comenzar a realizar experimentos más realistas (e incluso comenzar a realizar actividad comercial) de los que se obtendrá un feedback muy importante.

Se seguirá en este ciclo de prototipado y aprendizaje hasta que se produzca el encaje perfecto entre el problema y la solución, sustentado todo ello por métricas y hechos comprobados. La duración de esta etapa es incierta, dado que a cada propuesta comercial puede costarle más o menos llegar a un modelo de negocio rentable y sostenible, llegando gran parte de ellas a fracasar totalmente.

En el caso de que nuestro modelo de negocio responda positivamente a este ciclo, será hora de *“abandonarlo”* y comenzar a trabajar en el apartado empresarial y en el plan de negocio del proyecto. Idealmente, se seguirán utilizando los principios de la filosofía lean en las siguientes etapas del proyecto comercial hasta cuando ya haya encontrado su modelo comercial e implantado como empresa.

PARTE 2

## DESARROLLO DEL PROYECTO

Tras la fase de estudio y documentación, se pasó a la etapa de poner en práctica lo aprendido. En primer lugar, se estudió la viabilidad del uso de Lean Manufacturing en empresas de desarrollo de videojuegos. Aunque se concluye que, efectivamente, principios de esta filosofía pueden ser aplicados para mejorar la eficiencia de un estudio de desarrollo, esto no se pudo poner en práctica sin la participación de una organización de personas.

Dentro del programa Santander Yuzz 2017 de la Universidad Politécnica de Cartagena, se trabajó en el uso de las metodologías Lean Startup para la creación de una empresa de desarrollo de videojuegos, concluyendo con la elaboración de un Plan de Empresa de la propuesta comercial.

Finalmente, para abordar el desarrollo y diseño de videojuegos, se comenzó a realizar un prototipo haciendo uso de Unreal Engine 4, cubriéndose así las tres áreas de estudio planteadas en el proyecto

### *3. LEAN MANUFACTURING EN EL DESARROLLO DE VIDEOJUEGOS*

A pesar de su inherente carácter industrial, muchos aspectos de esta metodología pueden ser aplicados a empresas de otros sectores.

Principios como la Mejora Continua, el Just In Time o el Flujo tirado de la producción por parte del cliente se ajustan perfectamente a la dinámica del desarrollo de software, aunque su ejecución es diferente a la manera en la que se aplican en otras industrias.

En primer lugar, la asimilación de la filosofía Kaizen de mejora continua no encuentra ningún obstáculo ya que, como se ha indicado, independientemente de las medidas que se tomen, el primer paso en este proceso es la interiorización personal por parte de los trabajadores de la misma, el conocimiento de que siempre existe margen de mejora en nuestras vidas y en nuestro trabajo.

Los beneficios de la implantación de aspectos de la herramienta de las 5S a la hora de organizar y mantener el espacio de trabajo tampoco son difíciles de apreciar. Es, quizás, el enfoque que sitúa al cliente o consumidor en el centro del proceso comercial el que más beneficioso puede resultar a las empresas, especialmente con la envergadura que están alcanzando muchos proyectos de la industria.

No es de inusual que muchos proyectos se mantengan en secreto hasta etapas avanzadas de su desarrollo o que pequeños estudios realicen una propuesta creativa teniendo únicamente en cuenta sus criterios. ¿Qué pasa si, una vez lanzado el producto, este no es lo que los clientes esperan? Y, ¿qué ocurre si, directamente, a nadie le interesa el videojuego?

Estas preguntas pueden ser críticas para el éxito comercial de un videojuego y la propia supervivencia del estudio de desarrollo. Si a la hora de poner en marcha un proyecto, no se tiene en cuenta al cliente, la viabilidad económica del mismo estará rodeada de grandes cantidades de incertidumbre y de desconocimiento respecto a que apartados dedicar los recursos del estudio. Podríamos decir que se están

dedicando recursos y tiempo a apartados que no aportan valor ninguno a la propuesta para el cliente.

Se podría considerar este hecho como la generación de una gran cantidad de desperdicios, los cuales deben ser eliminados a toda costa según los principios de *Lean Manufacturing*. Aunque el desarrollo de aplicaciones, software y videojuegos no está exento de deshechos de las otras categorías listadas anteriormente, esta es probablemente la mayor fuente de desperdicios en los proyectos.

Si se sitúa al cliente en el punto de partida del desarrollo, se pueden evitar situaciones en las que las expectativas de los desarrolladores y los jugadores no están alineadas. Por ejemplo, puede darse el caso en el que el equipo de desarrollo realice cinco versiones diferentes a la hora de mejorar el agua del videojuego cuando a partir de la segunda versión el jugador se da por satisfecho y nuevas mejoras no le proporcionan ningún valor añadido. Los recursos dedicados a realizar esas tres versiones redundantes podrían haberse dedicado a elaborar otras características que el jugador esperaba y se quedaron fuera del producto final por falta de tiempo.

Pese a que puede parecer obvio y sencillo este enfoque al cliente, en el mercado de los videojuegos puede llegar a ser muy complicado. Los videojuegos, a diferencia de gran mayoría de las otras industrias, se desarrollan en fases muy marcadas. En otros productos de software, se puede comenzar a comercializar un producto con la funcionalidad mínima e ir mejorándolo en siguientes versiones. No se restringen a fases de concepto, pre producción, producción y testeo, sino que estas se mezclan y se trabaja en ellas de manera paralela en mayor o menor medida.

Por otro lado, la mayoría de los videojuegos consisten en una sola versión final que se tarda años en conseguir. Esta realidad puede ser muy perjudicial por muchos motivos. Quizás el de más peso sea que se pospone la fase de testeo al final de la producción, cuando realizar modificaciones y corregir errores puede llegar a ser más costoso y en ocasiones imposible. Se trabaja con fases de pre producción largas en las que se planifican con gran detalle todos los aspectos del proyecto, cuando, sin embargo, la información más valiosa proviene de la propia ejecución del mismo, pudiendo incluso la planificación estar sustentada en presunciones incorrectas.

Para adaptar la metodología Lean Manufacturing al sector del videojuego, sería necesario derribar las barreras y límites existentes entre las diferentes fases de desarrollo, lo cual, en muchas ocasiones no es posible. Por un lado, a la hora de comercializar un videojuego, este debe tener una cantidad mínima de contenido. Este contenido se suele generar en la fase de producción una vez descubierta la jugabilidad del título durante la preproducción. Podría decirse que los recursos elaborados para un videojuego dependen de las mecánicas del mismo. Si tras realizar un gran número de recursos estas mecánicas cambian, los recursos deberán ser modificados o sustituidos, ocasionando costes adicionales al proyecto.

Además, a la hora de buscar inversión o una editora que publique en videojuego, se suele exigir a las desarrolladoras conceptos detallados del futuro videojuego en el que quizás inviertan recursos. En caso de que sea así, el equipo de desarrollo tendrá libertad limitada a la hora de salirse de los conceptos presentados anteriormente.

Pese a todo, los beneficios de las metodologías ágiles y lean son tan amplios que, aunque no sea en todas las etapas del proyecto, merece la pena su implantación por lo menos en alguna fase del desarrollo.

Hay que destacar la tendencia actual del videojuego como servicio y la popularización de campañas de crowdfunding y sistemas de early-access, mediante los cuales los usuarios tienen acceso a versiones inacabadas de un videojuego previo pago, ayudando al desarrollador a obtener fondos para continuar con el desarrollo del título a la vez que obtienen grandes cantidades de feedback de los jugadores.

Además de esta necesidad de comunicación con el cliente, si se tiene en cuenta el proceso de desarrollo de las características del videojuego, puede ser observado que, debido a las dependencias de unas tareas respecto a otras, se puede establecer una analogía con un proceso industrial y modelar el desarrollo como uno.

Así, por ejemplo, para realizar un elemento del videojuego, el equipo de arte 3D necesita las ilustraciones generadas por el equipo de arte 2D, quienes a su vez necesitan las indicaciones del diseño del nivel realizadas por los desarrolladores. Si

profundizamos en cada una de estas áreas, se podrían establecer aún más relaciones de dependencia entre los diferentes actores implicados. De cualquier manera, se pueden establecer unas etapas en el estado de cada subelemento del videojuego: desde que se elabora el concepto, el guion, se diseña y se crean bocetos que más tarde se modelan en tres dimensiones para luego ser animados hasta que se le agregan los sonidos asociados a él y se testea su funcionamiento dentro del juego y se realizan los ajustes necesarios.

El funcionamiento de este proceso de desarrollo no es tan diferente a la manera en la que se crea un componente en una fábrica mientras va pasando por diferente maquinaria y procesos. Atendiendo a esto, Kanban constituye una potente herramienta para ayudar a la comunicación entre los diferentes departamentos o equipos del estudio a la vez que puede hacer el desarrollo más ágil.

Mediante el uso de tableros Kanban se facilita la visualización del flujo de trabajo actual. Pongamos como ejemplo que un nivel pasa por seis etapas diferentes antes de considerarse terminado: Concepto, Arte 2D, Arte 3D, Programación, Sonido y Testeo. Si un videojuego consta de tres niveles, el estado del proyecto se podría representar de la siguiente manera.



Figura 4. Tablero Kanban (A)

Así, se puede ver de manera fácil el estado de un elemento del videojuego. También se puede observar otro hecho. Si todas las etapas tardan el mismo tiempo en realizarse, cuando el equipo de programación comience a trabajar en el primer nivel, el equipo de diseño ya no tendrá más trabajo. Por su parte, los departamentos de sonido y de testeo ni siquiera habrán comenzado a trabajar. Estas fracciones de tiempo en el que un equipo no está trabajando constituyen un despilfarro que se debe de evitar.

Se podría aplicar otro principio de *Lean Manufacturing*, el trabajo con los lotes de producción más pequeños posibles. Si un nivel tiene un área subacuática y otra en una caverna, el equipo de sonido podría ir creando los recursos de la primera área mientras los otros equipos aún trabajan en el área de la caverna. Por ello, los niveles se podrían dividir en diferentes secciones, siendo la unidad de trabajo más pequeña. Así, si cada nivel se divide en tres áreas diferentes podríamos encontrarnos con el tablero de la siguiente figura.



Figura 5. Tablero Kanban (B)

Esta situación luce mejor que la de la primera figura, ya que prácticamente todos los equipos se encuentran trabajando. Pero se puede detectar un nuevo problema: los equipos de *Arte 2D* y, especialmente, *Arte 3D*, están ocasionando un cuello de botella en la producción, por lo que es necesario tomar medidas en esta etapa. Estos cuellos

de botella serán más fácilmente detectables cuanto más pequeñas sean las unidades de trabajo.

Este cuello de botella podría estar causado por falta de recursos en esta etapa. O puede que simplemente esté causado por la generación de cualquier tipo de desperdicio. Quizás el problema sea simplemente que los artistas realizan más trabajo del necesario o porque debido a problemas de comunicación cometen más errores. Sea como sea, será el momento de aplicar la filosofía de la mejora continua para solucionar el problema.

En un proceso ideal, todos los departamentos tardan el mismo tiempo en realizar una unidad de trabajo. En la realidad esto ocurre así de manera natural en rara ocasión. Se puede optar por *forzar* esta situación mediante el uso de mecánicas de timeboxing, mediante las cuales se fija un tiempo máximo para conseguir un objetivo. De esta manera, todos los equipos de la cadena de desarrollo dedican el mismo tiempo como máximo a cierta unidad de trabajo. De esta manera, se ayuda a la priorización y a la toma de decisiones mientras que se aumenta la productividad, la creatividad y la eficiencia de los equipos.

Sin embargo, establecer este tiempo máximo es complicado, ya que no puede ser tan poco que la calidad del trabajo se vea comprometida y no puede ser tanto que se llegue al punto en el que se deja de ofrecer valor de cara al jugador. Para ello, mediante ajustes iterativos y con la ayuda del feedback del cliente, se debe establecer este punto en el que el valor generado y el coste temporal se encuentran en sintonía.

Así, mediante la aplicación de esta metodología en el desarrollo de videojuegos, en conjunción con cualquier otra herramienta que sirva para eliminar otros desperdicios que se generen en el trabajo del día a día, se consigue ofrecer el producto de la calidad que el cliente espera de manera más rápida, ya que el trabajo es más eficiente, todo ello aprovechando los recursos disponibles al máximo. A pesar de las dificultades que pueda conllevar en un principio, es recomendable que los estudios se adapten a este marco, especialmente con la competitividad creciente que está experimentando el sector.

#### **4. LEAN STARTUP PARA UN ESTUDIO DE DESARROLLO DE VIDEOJUEGOS**

Entre enero y junio de 2017 se participó en el programa Santander Yuzz de emprendimiento con la propuesta de un estudio de desarrollo de videojuegos. Durante esos meses, se desarrolló la idea siguiendo la filosofía de Lean Startup, culminando con la elaboración de un Plan de empresa.

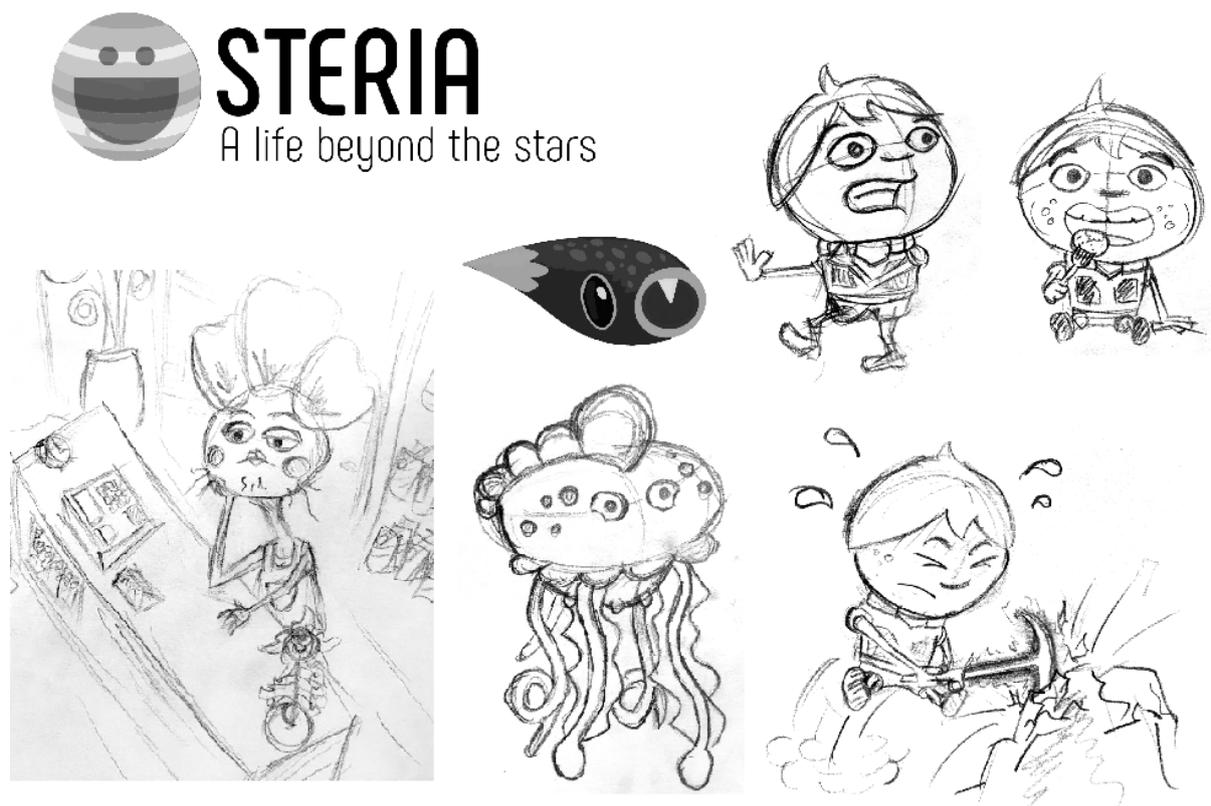
Se participó con la siguiente propuesta:

*"Desarrollo de un videojuego en Unreal Engine para PC. Dicho videojuego pertenece a la categoría de Life Simulator, género con una base de jugadores extensa y que no se está cubriendo en PC (sí en otros dispositivos, como Smartphones y consolas, portátiles, sobre todo). Los pocos títulos del género en PC cuentan con gran éxito (como Los Sims, Second Life o Stardew Valley). Steria (Título provisional) recibe sus mayores fuentes de inspiración de sagas como Animal Crossing, Harvest Moon o Los Sims. En Steria los jugadores son transportados a un planeta extraterrestre inexplorado, en el que tendrán que aprender a reconocer y usar los recursos y materias primas disponibles (mediante minería, pesca, agricultura...), convivir con los demás habitantes del asentamiento, decorar y construir un hogar y experimentar sorpresas día a día (el tiempo en Steria transcurre a la misma velocidad que el tiempo real). En el videojuego se hace especial énfasis en la experiencia de descubrimiento, coleccionismo, personalización y relaciones sociales. Respecto a este último punto, cabe destacar que se pretende invertir grandes esfuerzos en que la IA de los habitantes controlados por el ordenador simule personalidades reales. Es decir, los jugadores deberán desarrollar, poco a poco, su relación con los demás personajes del juego, quienes tienen diferentes aficiones, lloran, se enfadan, recuerdan experiencias pasadas y tienen sus motivos que les han llevado a formar parte de la expedición. Además, ya que los jugadores pueden personalizar su apariencia, el mobiliario de su casa, el aspecto de su pueblo... para reforzar la creación de comunidad y el apartado social, se plantea la inclusión de características multijugador online. Finalmente, me gustaría destacar que el proyecto forma parte de mi Trabajo de Fin de Grado titulado "Implementación de metodologías Lean Manufacturing en el*

*desarrollo de videojuegos". Ya que durante el desarrollo se usarán y estudiarán técnicas similares, considero que la formación recibida por el programa Yuzz podría ser valiosa."*

Durante las primeras sesiones se trataron asuntos relacionados con la creatividad, mediante técnicas como el gamestorming o el brainstorming y el trabajo en equipo. En esta primera fase, se debía desarrollar y buscar los puntos fuertes de la idea. Se realizó un *Game Design Document* (GDD) y algunos bocetos para presentar este proyecto, el cual se puede encontrar en el **Anexo 2**.

En este documento, se explican de manera general aspectos del videojuego como la narrativa, jugabilidad, mecánicas, personajes, escenarios, etc.



*Figura 6. Algunos bocetos realizados para este proyecto.*

Tras ello, se pasó a la fase de diseño del modelo de negocio. Para ello, se elaboraron los BMEM, BMC y se analizó el tamaño de mercado.

Se pueden encontrar el Business Model Environment Map y el Business Model Canvas en los **Anexos 3.1 y 3.2** respectivamente. *(Plantillas extraídas de Knowledge.li y Strategizer.com. Basadas en Generación de Modelos de Negocio, por Alexander Osterwalder y Yves Pigneur)*

Respecto al mapa de entorno, se analizaron los cuatro segmentos que lo componen, con los siguientes aspectos a considerar:

**Fuerzas del Mercado:** En primer lugar, se destaca la tendencia hacia un público más cada vez más heterogéneo, con mayor porcentaje de mujeres a la vez que aumenta la edad media de los jugadores<sup>27</sup>. Además, dado que los perfiles abarcan a un público más diverso y más adulto, los jugadores dan más importancia que nunca a la narrativa y al videojuego como medio audiovisual comparable al cine<sup>28</sup>.

**Tendencias Clave:** Algunas de las tendencias clave que suponen una oportunidad de negocio pueden ser el éxito de los estudios independientes, que desarrollan juegos con presupuestos limitados y que pueden llegar a ser más rentables que las propuestas de las grandes empresas del sector. También la popularización de los deportes electrónicos y los servicios de vídeo en streaming, así como el alto uso de smartphones como dispositivo de juego y la introducción de nuevas tecnologías, como la Realidad Virtual<sup>29</sup>.

**Fuerzas de la Industria:** Se listan diversos actores del sector a tener en cuenta a la hora de lanzar un negocio en la industria, como los editores, otros grandes

---

<sup>27</sup> The new faces of Gaming. Ipsos Connect. 2017

<https://www.ipsos.com/ipsos-mori/en-uk/new-faces-gaming>

<sup>28</sup> Why non-violent games are taking over an industry? IGN.com

<http://www.ign.com/articles/2014/04/24/why-non-violent-games-are-taking-over-an-industry>

<sup>29</sup> 16 Trends that Will change the games industry. The Guardian

<https://www.theguardian.com/technology/2015/jul/23/16-trends-that-will-change-the-games-industry>

desarrolladores, las fabricantes de hardware, los medios de comunicación o las plataformas de distribución online.

Tendencias Macroeconómicas: Respecto a las tendencias económicas de la industria, algunas de estas ya se han mencionado a lo largo de esta memoria. Por una parte, los beneficios que genera la industria del videojuego son mayores año tras año. También es importante resaltar el impacto que están teniendo herramientas de crowdfunding para la financiación de proyectos de estudios de desarrollo pequeños. Por otro lado, destacar la mayor accesibilidad a la tecnología y la disponibilidad de más vías de formación en el sector.

En referencia al Business Model Canvas, se plantearon los siguientes puntos en cada una de las secciones del mismo.

**Segmentos de clientes:** Los clientes de la propuesta comercial serán los jugadores de más de 7 años. Los límites geográficos no están muy marcados, al tratarse de un negocio online en gran parte. Sin embargo, por cuestiones de idioma y culturales, el público objetivo es mayoritariamente europeo, con un nivel adquisitivo medio. También, debido a la dificultad que conlleva llegar a los jugadores más jóvenes, en una primera etapa se centrarían los esfuerzos en atraer al público mayor de 16 años, sin diferencias de sexo.

**Propuesta de valor:** La principal propuesta de valor que se propone es la de anteponer calma antes que conflicto, que la experiencia de juego sea tranquilizadora para el jugador potenciando mecánicas de exploración, recolección y personalización. Por otro lado, se apuesta por la transparencia y el feedback desde las etapas tempranas del proyecto.

**Canal de distribución:** Los canales por los que los clientes pueden adquirir nuestro productos y servicios son, en primer lugar, plataformas de distribución online como Steam, GOG o Humble Store. Existe la posibilidad de conseguir distribución física de los productos, pero su probabilidad es reducida.

**Relación con los clientes:** La comunicación con los clientes se llevará por redes sociales y, de manera más importante, por comunidades online, ya sea en foros propios como en canales de terceros, como Reddit.com o Discord <sup>30</sup>.

**Fuentes de ingresos:** Los beneficios de la propuesta comercial se obtendrán, principalmente, mediante ventas. Se plantea también la posibilidad de diversificar el negocio y ofrecer soluciones de gamificación a terceros.

**Recursos clave:** Los recursos clave necesarios para llevar a cabo el proyecto son, por una parte, recursos humanos para formar el estudio de desarrollo y, por otra, los equipos informáticos, software y requisitos de espacio para poder llevar a cabo el trabajo.

**Actividades clave:** Obviamente, la actividad más crítica consiste en el diseño y desarrollo de videojuegos. Sin embargo, esta tiene que estar acompañada de las herramientas para comunicarse con los jugadores, así como de la promoción de los títulos, ya sea mediante publicidad online o la asistencia a ferias del sector.

**Socios clave:** Sería muy beneficioso para el proyecto el ingreso en asociaciones de la industria como la Asociación Española de Videojuegos (AEVI). Por otro lado, para la distribución de los videojuegos será necesaria una relación con editoras, empresas de hardware y videoconsolas (para poder lanzar los títulos en sus plataformas) así como con las otras plataformas de distribución digital.

**Estructura de costes:** Finalmente, en referencia a los costes, estos provienen de varias fuentes. En primer lugar, se contemplan los sueldos de los trabajadores del estudio, así como los generados por los recursos materiales y administrativos. Por otro lado, tanto la distribución, como el marketing y otros aspectos subcontratados suponen generan una cantidad importante de gastos.

---

<sup>30</sup> How to build an active and engaged indie game community with discord

<https://blog.discordapp.com/how-to-build-an-active-and-engaged-indie-game-community-with-discord-4a5be74d822e>

En referencia al tamaño de mercado, se usó la herramienta TAM, SAM y SOM. Estas siglas hacen referencia a:

**Total Addressable Market (TAM):** Se refiere al tamaño total del mercado al que se dirige el negocio. Sirve para analizar como de grandes son las oportunidades y el potencial de las mismas si en etapas posteriores el proyecto pivota y se dirige a otra área.

**Serviceable Available Market (SAM):** Se refiere al mercado que se puede servir con el modelo de negocio y recursos del proyecto actual. Se refiere al potencial global del modelo de negocio.

**Serviceable Obtainable Market (SOM):** Sirve para valorar el potencial del proyecto a medio plazo con una estimación más realista, dependiendo de los recursos y los clientes que podrían alcanzarse.

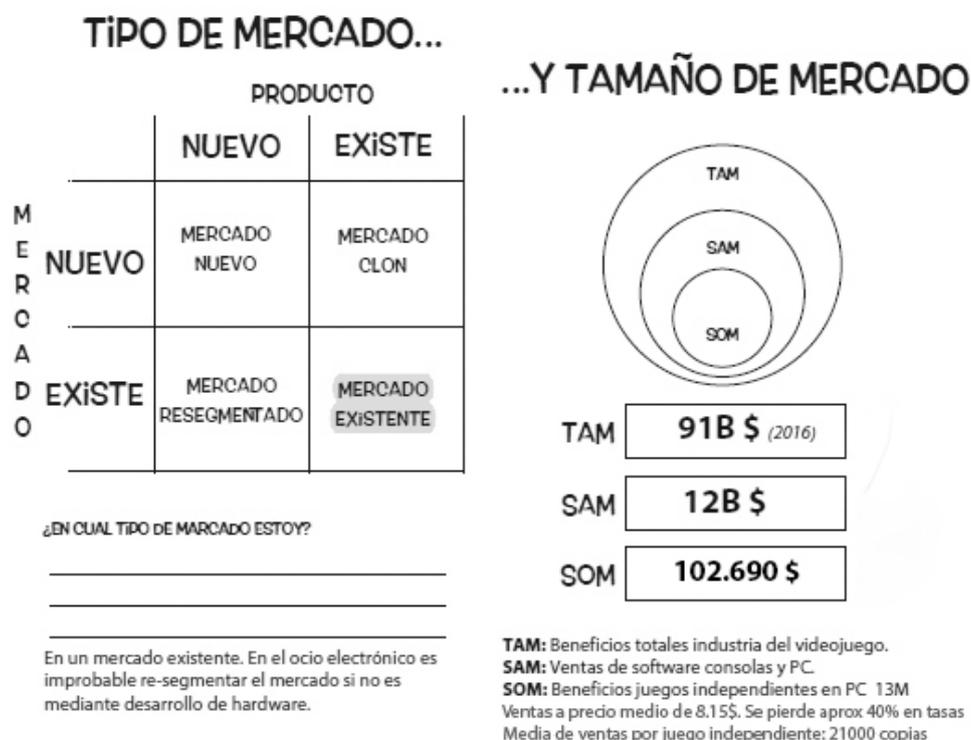


Figura 7. Tipo de mercado y TAM, SAM, SOM del mismo. Plantilla por Santander Yuzz

Llegados a este punto, se comenzó a poner las ideas a prueba. Para ello, se redactó una lista de hipótesis a validar tras lo cual se diseñó un cuestionario para la elaboración de entrevistas. Estos documentos se pueden encontrar en los **Anexos 3.3 y 3.4**. (*Plantillas del listado de hipótesis por Santander Yuzz*).

Estas hipótesis se plantean desde dos puntos de partida: La propuesta de valor y el Segmento de Clientes. A grandes rasgos, no vienen a ser más que una redacción más detallada de lo indicado en el Business Model Canvas, a la vez que se plantean de manera que puedan ser demostradas o desmentidas una vez puestas a prueba.

Antes de la realización de estas entrevistas, se estudió la complejidad del proyecto que se había planteado. Debido a la jugabilidad propuesta, el proceso de desarrollo sería costoso y, a su vez, la experiencia de juego se prolongaba en el tiempo por lo que realizar sesiones de testeo con posibles jugadores sería más complicado. Se produjo entonces un **pivote**, el modelo de negocio sería el mismo pero el videojuego a desarrollar en este proyecto de ejemplo sería otro distinto. Se optó por realizar un videojuego de plataformas con, en mayor o menor medida, los mismos principios y objetivos que el primer título propuesto. Este proyecto sería denominado como Mapu y la Luz Perdida.

Fueron realizadas un número reducido de entrevistas (12) y, aunque algunas de las hipótesis recibieron resultados positivos, se habrían necesitado datos más concluyentes para seguir adelante con el proyecto. A pesar de ello, para seguir con el ritmo temporal del programa Yuzz, se siguió trabajando en el mismo proyecto.

Una vez validado el problema, era turno de validar la solución. Para ello, se realizó un Producto Mínimo Viable. Este consistió en un portal web en el que se explicaba a grandes rasgos la dinámica y argumento del título y, a modo de experimento, se incluiría una llamada a la acción consistente en la suscripción a una lista de distribución de correo para medir el interés generado.

De manera paralela, se realizó un pequeño proyecto en *Unreal Engine 4* en el que, mediante recursos y modelos ajenos o incluidos en el contenido por defecto, se

compuso una escena que serviría para elaborar recursos gráficos que acompañaran a la página web. Este portal web es accesible mediante la siguiente url:

<http://yoeldsc.es/mapu/>



*Figura 8. Portada del portal web desarrollado*

Es necesario indicar que el experimento simplemente se diseñó, no se difundió ni se puso en marcha ni mediante un contador de clicks en el botón de suscripción a la lista de distribución de correos ni la propia existencia de la misma.

Se realizaron unas pocas entrevistas informales para recibir feedback sobre este portal web, al mismo tiempo en el que se entraba en la fase final del programa Santander Yuzz. En esta última etapa, se debía redactar un Plan de empresa para los primeros años de la propuesta comercial. Así, se nombró al estudio de desarrollo que sirvió como caso de estudio como Lemon Jellyfish. Este Plan de Negocio se puede encontrar en el **Anexo 4**.

En el Plan de Negocio se abordan temas como el Plan de Marketing del proyecto, análisis DAFO del mismo, el plan de operaciones o el plan de inversiones. Todos estos apartados se encuentran explicados en dicho documento, por lo que no serán replicados en esta sección de la memoria y se recomienda su lectura.

La experiencia recibida mediante la participación en el programa Santander Yuzz fue muy positiva para el proyecto. Pese a ello, debido a los requisitos temporales del mismo, algunos aspectos deberían analizarse en mayor profundidad a la hora de lanzar una propuesta comercial real al mercado, sirva esta memoria de guía para ello.

## *5. DESARROLLO DEL PROTOTIPO JUGABLE*

Finalmente, para terminar los contenidos a cubrir por este Trabajo de Fin de Estudios, se comenzó a desarrollar un prototipo jugable del videojuego propuesto mediante el uso de metodologías de Lean Startup.

Se debía tomar una decisión importante para la ejecución del proyecto: La elección del motor gráfico.

### 5.1 ELECCIÓN DEL GAME ENGINE

El primer paso para comenzar el desarrollo era la elección de las herramientas y el software que se utilizarían. Entre ellas, el motor gráfico era la más importante. Un motor gráfico.

Un motor gráfico o motor de videojuego es un framework orientado al desarrollo de videojuegos. Incluye funcionalidades como el sistema de renderizado de gráficos, el motor de físicas, animación, inteligencia artificial, interfaces de red y la programación del título, entre otras cosas.

Es uno de los factores más costosos de desarrollar, por lo que habitualmente se suelen reutilizar o adaptar el mismo motor gráfico para distintos videojuegos. Conforme se van desarrollando, van surgiendo nuevas versiones del motor e incluso nuevos motores diferentes. Elaborar un motor para el videojuego del proyecto era totalmente inviable. Afortunadamente, en la actualidad existen muchos motores disponibles para la comunidad, seguramente estando este hecho relacionado con la proliferación de pequeños estudios de desarrollo sin capacidad para desarrollar sus propios motores. Así, mediante suscripciones, compra única o un porcentaje de los beneficios generados por el producto final, cualquiera puede disponer de herramientas potentes que faciliten el desarrollo de sus propuestas.

Se estudió el uso de los siguientes motores:

### **GameMaker Studio (YoYo Games)**

GameMaker Studio es un motor de desarrollo de juegos, principalmente, en dos dimensiones. Destacan su facilidad de uso, ya que está enfocado a desarrolladores que no necesariamente tengan conocimientos de programación. Proporciona una interfaz en la que el comportamiento se configura con diferentes acciones que simplemente se utilizan mediante *drag & drop*, repartidas en bibliotecas. Los usuarios más avanzados pueden configurar y crear nuevas bibliotecas mediante un lenguaje de programación propio llamado GML (Game Maker Language). También incluye soporte para código programado en C++, Delphi o Pascal. La calidad final de los videojuegos depende mucho del equipo de desarrollo; a pesar de que esta facilidad de uso puede repercutir en limitaciones técnicas, se pueden alcanzar resultados sobresalientes. Es por esta razón por lo que su popularidad está creciendo y la disponibilidad de documentación es amplia. Se puede descargar y consultar más información a través de su página web ( <https://www.yoyogames.com/> ).

### **Unity 5 (Unity Technologies)**

Es quizás el principal precursor de esta tendencia a la democratización del acceso a la tecnología de desarrollo de videojuegos. Desde su primera versión en 2004, Unity se ha alzado como el entorno de desarrollo más usado y popular de la industria, gracias a esta visión de un motor accesible, potente y multiplataforma. La programación se realiza mediante C#, a pesar de que soporta lenguajes como Unity Script (basado en Javascript) y Boo (inspirado en Python), debido a su bajo uso, estos dos lenguajes han dejado o están en proceso de dejar de recibir soporte.

Es fácil de usar y, debido a la gran comunidad de desarrolladores que utilizan la herramienta, la oferta a documentación y contenidos formativos es amplísima. A pesar de que las últimas versiones han mejorado mucho, su capacidad gráfica y el rendimiento del motor son sus puntos débiles. En algunos casos la no disponibilidad del código fuente crea una menor flexibilidad.

Su uso es gratuito, pero si el estudio recibe más de 100.000\$ anuales o a recibido esa cantidad a lo largo de otro periodo temporal, deberá acogerse a uno de los modelos de suscripción a la herramienta. Se puede descargar a través de <https://unity3d.com/es>

### **Unreal Engine 4 (Epic Games)**

Unreal ha sido un motor de referencia desde su lanzamiento en 1998. Debido a su potencia y a su política de licencias, fue utilizado por gran cantidad de empresas del sector. En 2011 se publicó una versión gratuita de Unreal Engine 3, no llegando a la cuarta versión del motor con un modelo gratuito y multiplataforma, similar a Unity, hasta 2015.

Destacan su potencia, su estabilidad y rendimiento, así como la alta variedad de herramientas y funcionalidades. Su curva de aprendizaje es más alta que la de otros motores, como Unity. Su lenguaje de programación es C++, aunque también cuenta con un sistema de script visual denominado "*Blueprint*". Este lenguaje está totalmente integrado con el código C++, pudiendo combinarse ambos a la vez que se pueden convertir scripts en un lenguaje al otro de forma automática. Su objetivo permitir desarrollos más rápidos permitiendo prototipar sin necesidad de programar usando C++.

Su uso también es gratuito hasta que un producto alcanza los 3000\$ de beneficio, cuando comienza a aplicarse una tasa del 5% de los beneficios obtenidos. Puede ser descargado desde <https://www.unrealengine.com/>

### **CryEngine V (Crytek)**

Motor desarrollado por Crytek cuya primera versión se lanzó en 2002. Desde entonces, ha sido considerado un referente gráfico y ha sido usado por multitud de desarrolladores bajo licencia. Soporta C++, C# y Lua como lenguajes de programación. Su curva de aprendizaje es la más alta de las de los motores listados.

A finales de 2016 lanzó su última versión, CryEngine V. Esta versión adoptó un modelo completamente gratuito, sin tasas de suscripción ni de porcentajes de ventas

independientemente de las ganancias de los proyectos. Obtiene sus ganancias mediante un porcentaje de las ventas del Marketplace de recursos, programas de membresía por acceso a formación, soporte y otras ventajas, así como del licenciamiento para grandes empresas con necesidades especiales. Se puede descargar a través de su página web <https://www.cryengine.com/>

Por su rendimiento, potencia, su posición en la industria y su punto medio en cuestiones de dificultad de aprendizaje, se decidió usar **Unreal Engine 4** en su versión 4.16.3 para elaborar el prototipo del videojuego.

Independientemente de ello, es necesario destacar positivamente este cambio en la industria, ya que hace 10 años esta disponibilidad de herramientas no existía, siendo necesaria una gran cantidad de recursos para comenzar a desarrollar un videojuego, recursos de los que muchos diseñadores, programadores y artistas no disponían.

Esta tendencia a la libre disponibilidad de motores gráficos no parece que vaya a cambiar en los años venideros, especialmente cuando otros actores importantes de la industria están lanzando propuestas como **Lumberyard**, un motor en beta derivado de CryEngine, modificado por Amazon con integración completa con Amazon Web Services (lo que lo hace una opción muy interesante para juegos con enfoque online) o el venidero **Source 2** de Valve, aún sin fecha de lanzamiento pero del que se ha anunciado que se acogerá a un modelo completamente gratuito mientras los videojuegos desarrollados en el mismo se publiquen a través de Steam, el canal de distribución digital más importante en la actualidad, propiedad de Valve

## 5.2 OTRO SOFTWARE A UTILIZAR

Adicionalmente a Unreal Engine, se debían utilizar otras herramientas para elaborar diferentes recursos para el prototipo. En primer lugar, era necesario software para la elaboración de los recursos 3D. Es necesario señalar que la elaboración de gráficos en 3D es compleja, ya que consta de muchos procesos, como el modelado poligonal, sculpting, elaboración de texturas, animación, etc. que hacen necesario el uso paralelo de varias herramientas de software.

En primer lugar, como herramienta de modelado 3D se consideraron dos softwares: Maya y Blender.

**Maya** forma parte de la oferta de Autodesk en materia de modelado y animación en 3D. Es el programa más utilizado en la industria. Su modelo de suscripción tiene un coste de 1.936€ anuales, aunque existen versiones gratuitas para estudiantes. Se puede adquirir a través de la web de Autodesk ( <https://www.autodesk.es/> )

Por otro lado, **Blender** es un programa multiplataforma también dedicado a las mismas funcionalidades que Maya. Aunque la curva de dificultad a la hora de aprender a utilizar es más elevada, debido a su carácter Open Source ha gozado de gran aceptación y su popularización va en aumento, con más desarrolladores trabajando para mejorar el software, el cual cada vez es más potente. Se puede descargar a través de su página web ( <https://www.blender.org/> ).

Debido a este carácter Open Source y a que ya tenía un poco de experiencia con **Blender**, decidí utilizar este software para el proyecto.

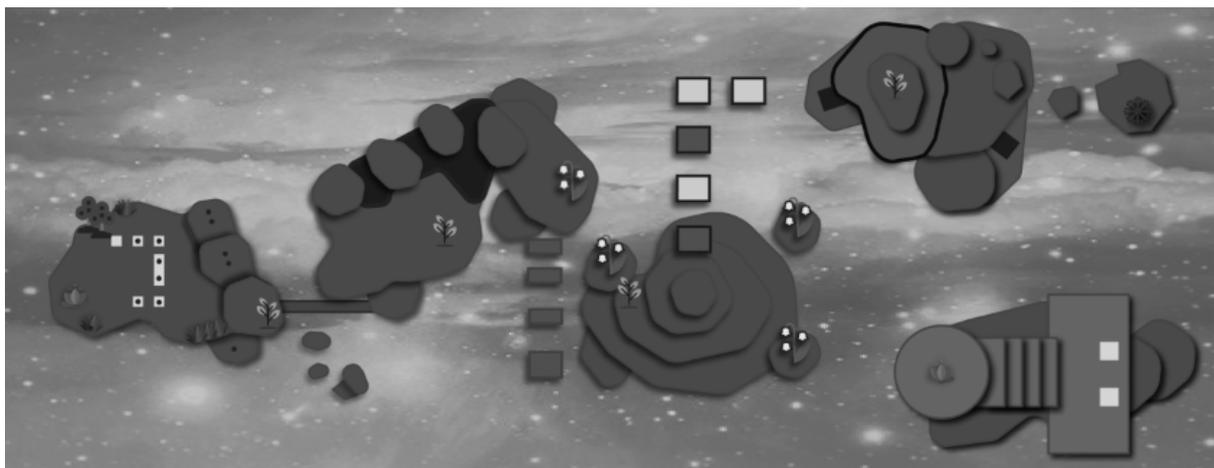
Por otro lado, para la elaboración de recursos gráficos bidimensionales como texturas se utilizó **Adobe Photoshop** y **Adobe Illustrator**, dada mi experiencia con estos programas. Además, para la elaboración de materiales (serán tratados más adelante), también se utilizó de manera puntual **Substance Designer**, de la suite de Allegorithmic, accesible sin necesidad de suscripción gracias a una licencia de estudiante.

Por otro lado, aunque no se trabajó en profundidad en esta área, para la generación y edición de recursos de audio fueron utilizados **Garageband**, desarrollado por Apple y distribuido de manera gratuita, y **Adobe Audition**, el cual, al igual que Photoshop e Illustrator, es accesible mediante suscripción al programa **Adobe Creative Cloud**. Se puede adquirir una licencia de esta suite a través de <http://www.adobe.com/es/creativecloud.html>

### 5.3 DESARROLLO EN UNREAL ENGINE 4

El desarrollo de este primer prototipo se desarrolló entre agosto y septiembre de 2017.

En primer lugar, el esquema del nivel propuesto era el que se muestra en la siguiente figura.



*Figura 9. Boceto del nivel de prototipo a desarrollar.*

Este primer concepto se respetó en gran medida, aunque se introdujeron algunos cambios a la hora de diseñar el nivel de prueba en Unreal Engine. Este nivel se compondría de cinco secciones interconectadas. El objetivo del jugador es, simplemente, llegar a la última de ellas. Para ello, podría hacer varias acciones, como saltar, agacharse o lanzar pulsos luminosos que servirían para interactuar con otros objetos. Las acciones que el jugador podría realizar se ampliaron, como se verá más adelante.

Respecto al desarrollo del nivel, en la primera de las secciones no se plantearía ningún reto en especial, ya que serviría a modo de toma de contacto. El jugador sólo debería saltar para llegar a niveles altos del terreno y avanzar a la siguiente sección. En esta se introducía un riesgo, ya que si el jugador fallaba el salto entre las plataformas caería entre las zarzas y debería volver a empezar. Al final de esta sección, debería activar unas plataformas flotantes para llegar a la siguiente etapa del nivel. En esta tercera etapa el jugador debería escalar y esquivar obstáculos en una sección más vertical que las anteriores. Durante el camino, debería activar tres interruptores para que activara el camino a la siguiente sección.

En esta cuarta sección, el jugador debería llegar a un lago pasando a través de una cueva. Allí, habría diversas plataformas. En la más alta de ellas, habría un diente de león que el jugador usaría para llegar flotando a la quinta sección, en la que se encontraría la meta tras una puerta que debería abrir.

Algunas de estas cosas cambiaron, bien para añadir otras características o por cuestiones técnicas (como la parte de la cueva, dado que la cámara no respondía bien en interiores). A pesar de ello, el esquema del nivel se mantuvo similar a lo planteado inicialmente.

El proceso de desarrollo fue iterativo; se fueron mejorando secciones anteriores, corrigiendo errores, añadiendo funcionalidades no planteadas en un inicio, sustituyendo elementos antiguos que no funcionaban correctamente, etc.

Pese a ello, en esta memoria se presentarán los elementos desarrollados de manera ordenada, divididos por secciones que no tienen por qué representar necesariamente el flujo de trabajo.

### 5.3.1 CONFIGURACIÓN Y ORGANIZACIÓN DEL PROYECTO

Una de las primeras acciones consistía en establecer la organización de los elementos del directorio de recursos y la del *World Outliner*, el árbol de los elementos que se incluyen en el nivel.

Respecto a los recursos, se optó por utilizar siete directorios: Actores, Blueprints, Niveles, Materiales, Modelos, Partículas y Sonido.

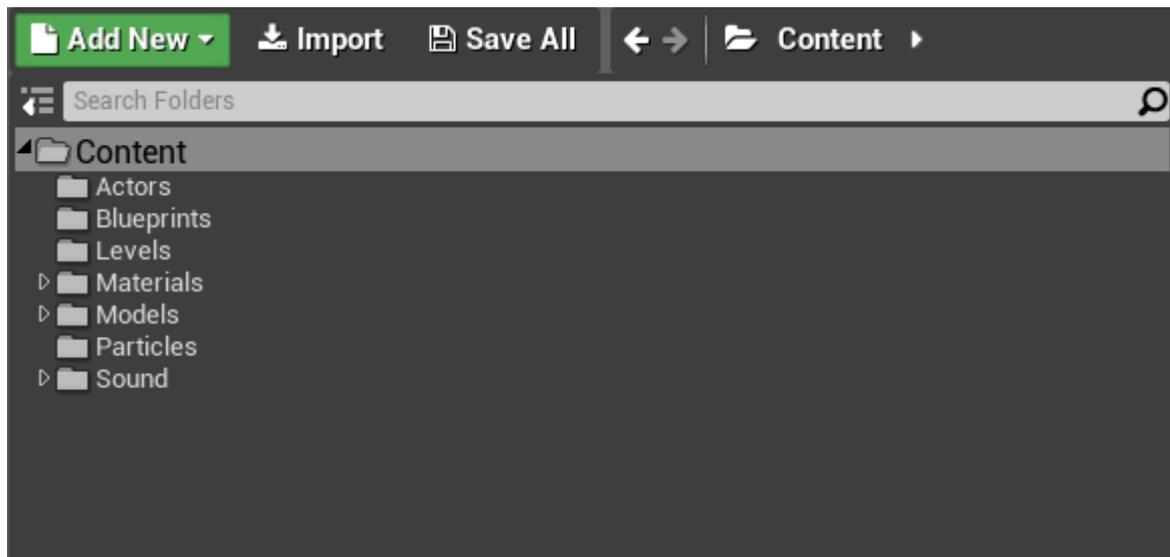


Figura 10. Directorios para organizar los recursos del proyecto.

Los recursos fueron organizados de esa manera dado que se alcanzó un número moderado de los mismos. Prácticamente todos son explicativos; en Sonido se encontraban los recursos de audio, en Partículas los sistemas de partículas elaborados, en Modelos los modelados 3D y en Materiales tanto texturas como materiales realizados a partir de ellas. En relación con los 3 directorios restantes, Actores, Blueprints y Niveles, su uso fue, quizás más confuso dado que en naturaleza son similares. Se pretendía que en Niveles estuvieran los recursos asociados al desarrollo de los niveles, como interfaces de usuario, secuencias de animación, los propios niveles y otros scripts. En Blueprint estarían los scripts desarrollados que más tarde serían implementados por los Actores del directorio del mismo nombre. Al final esto no fue así, dado que los scripts se definían en los propios actores y el uso de estos directorios no fue ordenado, por lo que es un punto a revisar en siguientes

etapas del proyecto.

Por otra parte, en la organización de los elementos del nivel o World Outliner se optó por dividirlos en seis directorios: Los cinco correspondientes a cada sección y uno adicional para elementos generales, como la música de fondo y volúmenes de post procesado. A modo de ejemplo para clarificar este apartado, en los recursos se definiría el comportamiento de un objeto, encontrándose todas las instancias del mismo utilizadas en el nivel en el World Outliner.

Uno de los primeros pasos a la hora de elaborar un proyecto es la creación de un *Game Mode*. En este script, se definen las reglas básicas del videojuego, como el número de jugadores, como se gestionan las pausas o las transiciones entre niveles.

Además, es necesario establecer este *Game Mode* como modo por defecto a la vez que se selecciona el personaje que controlará el jugador por defecto.

También fue configurado el mapeo de controles con los que interactuaría el jugador. Así, se definieron 4 acciones: Salto, Pulso/Interactuar, Agarrar y Agacharse. Adicionalmente, se establecieron los controles para mover al personaje y para girar la cámara. Estos controles se definieron tanto para teclado y ratón como para mando. Se puede ver la asignación de botones en la siguiente figura

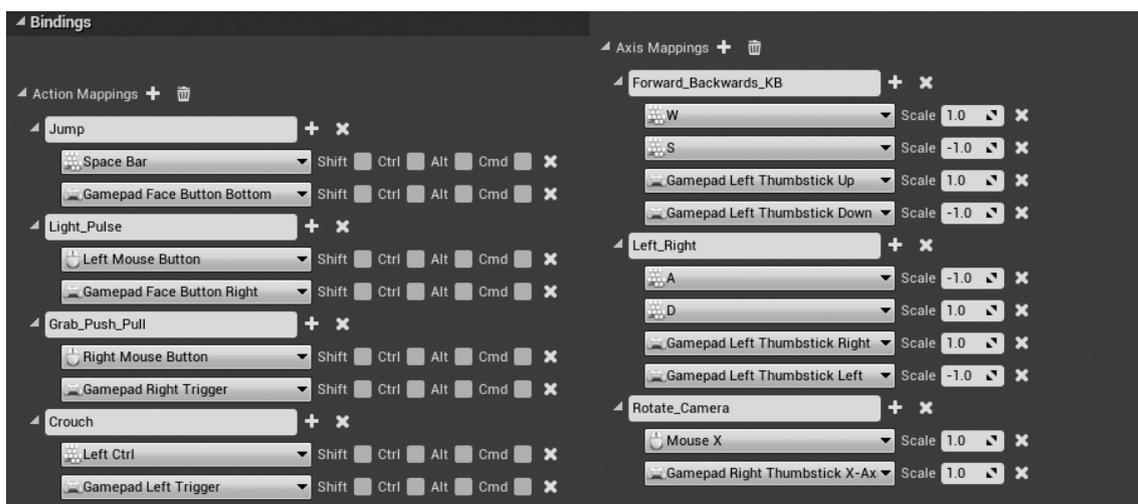


Figura 11. Asignación de controles del prototipo

Finalmente, también fueron creados tres canales de colisión personalizados: ActivableItem, GrabableObject y PushableObject. Estos canales servirán para definir actores con reacciones específicas a ciertas acciones del jugador. Su funcionamiento se explicará más adelante.

### 5.3.2 PROGRAMACIÓN DE SCRIPTS

En la primera fase del proyecto, se comenzó a trabajar mediante programación en C++ a través de Microsoft Visual Studio.

Un ejemplo de código desarrollado en esta fase es el siguiente:

```
“OpenDoor.h”

#pragma once

#include "CoreMinimal.h"
#include "Components/ActorComponent.h"
#include "Engine/TriggerVolume.h"
#include "OpenDoor.generated.h"

UCLASS( ClassGroup=(Custom), meta=(BlueprintSpawnableComponent) )
class BUILDINGSCAPE_API UOpenDoor : public UActorComponent
{
    GENERATED_BODY()

public:
    // Sets default values for this component's properties
    UOpenDoor();

protected:
    // Called when the game starts
    virtual void BeginPlay() override;

    void OpenDoor();
    void CloseDoor();

public:
    // Called every frame
    virtual void TickComponent(float DeltaTime, ELevelTick TickType,
    FActorComponentTickFunction* ThisTickFunction) override;

private:
    UPROPERTY(VisibleAnywhere)
    float OpenAngle = 90.0f;

    UPROPERTY(EditAnywhere)
    ATriggerVolume* PressurePlate;
    UPROPERTY(EditAnywhere)
    float DoorCloseDelay=1.f;
    float LastDoorOpenTime;
```

```

    AActor* ActorThatOpens;
    AActor* Owner;

};

“OpenDoor.cpp”

#include "OpenDoor.h"
#include "Gameframework/Actor.h"

// Sets default values for this component's properties
UOpenDoor::UOpenDoor()
{
    // Set this component to be initialized when the game starts, and to be ticked
    every frame. You can turn these features
    // off to improve performance if you don't need them.
    PrimaryComponentTick.bCanEverTick = true;
}

// Called when the game starts
void UOpenDoor::BeginPlay()
{
    Super::BeginPlay();

    ActorThatOpens = GetWorld()->GetFirstPlayerController()->GetPawn();
    Owner = GetOwner();
}

void UOpenDoor::OpenDoor()
{
    Owner->SetActorRotation(FRotator(0.0f, OpenAngle, 0.0f));
}

void UOpenDoor::CloseDoor()
{
    Owner->SetActorRotation(FRotator(0.0f, 0.0f, 0.0f));
}

// Called every frame
void UOpenDoor::TickComponent(float DeltaTime, ELevelTick TickType,
FACTORCOMPONENTTICKFUNCTION* ThisTickFunction)
{
    Super::TickComponent(DeltaTime, TickType, ThisTickFunction);

    if (PressurePlate->IsOverlappingActor(ActorThatOpens))
    {
        OpenDoor();
        LastDoorOpenTime = GetWorld()->GetTimeSeconds();
    }

    if (GetWorld()->GetTimeSeconds()-LastDoorOpenTime > DoorCloseDelay)
    {
        CloseDoor();
    }

    // ...
}

```

Figura 12. Script OpenDoor en C++

No se va a analizar en detalle este código, ya que forma parte de las primeras tomas de contacto con Unreal y no forma parte del proyecto final. Este script es un componente que puede ser asignado a diferentes actores de un nivel, cuya finalidad es la apertura de una puerta al pasar por encima de un volumen. Así, se establecen algunas variables, como el ángulo de apertura, el propio volumen y el actor que lo activa. Estas variables pueden ser directamente modificadas por el editor de Unreal Engine. Así, cuando el actor pasa por encima del volumen asignado, se llama a la función `OpenDoor()`, que hace rotar la puerta.



*Figura 13. Funcionamiento del Script OpenDoor*

Este comportamiento se podría replicar utilizando Blueprint con el siguiente gráfico:

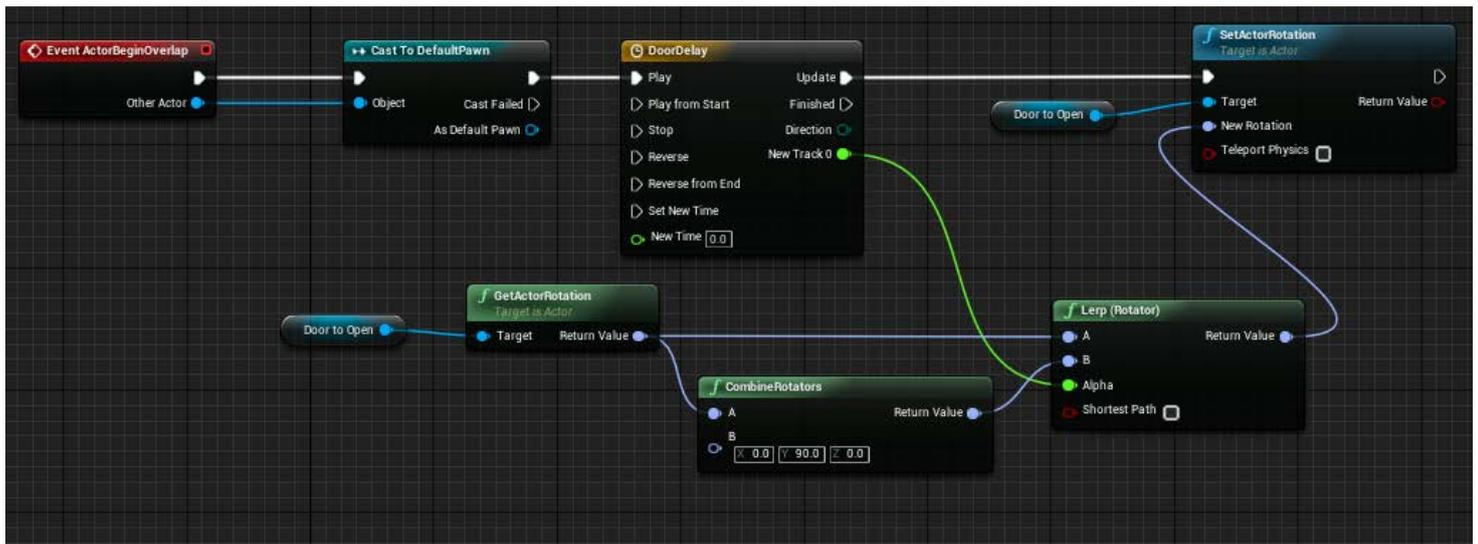


Figura 14. Blueprint con comportamiento similar a OpenDoor

A grandes rasgos, en los dos primeros módulos se comprueba que el jugador se ha situado sobre el volumen. Tras ello, se establece un retardo para que la apertura de la puerta sea progresiva, lo cual se consigue tras modificar la rotación en el eje Y de la misma.

La disponibilidad de una herramienta como Blueprint supone una tremenda ventaja a la hora de prototipar e incluso desarrollar scripts finales. Su funcionalidad es muy amplia e incluso se pueden definir nuevos módulos personalizados usando C++.

A la hora de definir un actor o elemento, en primer lugar, se les asignan a estos diversos componentes, como un modelo 3d, volúmenes, inteligencia artificial, etc. Además, también se le pueden definir variables de diversos tipos. Tras ello, se puede definir el constructor del mismo, así como su comportamiento en el gráfico de eventos, donde utilizaremos Blueprint.

Este lenguaje de scripting visual permite programar comportamientos de manera rápida y eficaz, por lo que tras barajar ambas opciones se decidió por utilizar este sistema. En este prototipo, se desarrollaron 15 scripts de relativa sencillez, teniendo algunos de ellos scripts hijos derivados. No se analizarán todos ellos, sino aquellos más interesantes o que sirvan para explicar algunos módulos de especial utilidad. En

algunos casos, sólo se mostrarán partes del script, ignorando aquellas que sean redundantes para la explicación del funcionamiento general.

### 5.3.2.1 PlatformGameMode

Este script ya ha sido mencionado anteriormente. En el prototipo, en este Game Mode se declaraban 6 variables. Una de ellas correspondía con la posición de respawn (**Spawn Transform**), a la que el jugador volvería tras ser eliminado. Las demás eran contadores; por un lado, uno de tiempo y por otro las correspondientes a los dos tipos de coleccionables a recolectar por el jugador (Pétalos y Frutas). Así, para cada uno de ellos se definía una variable que consistía en la suma total de los presentes en el nivel y otro que contabilizaría los que había conseguido el jugador (tenemos pues las variables Petal Counter, Fruit Counter, Total Petals y Total Fruits, de tipo int todas ellas).

Además de ello, incluye la funcionalidad de que, tras ser eliminado, el jugador pueda volver al juego desde una posición determinada. Esta funcionalidad está desarrollada de la siguiente manera:

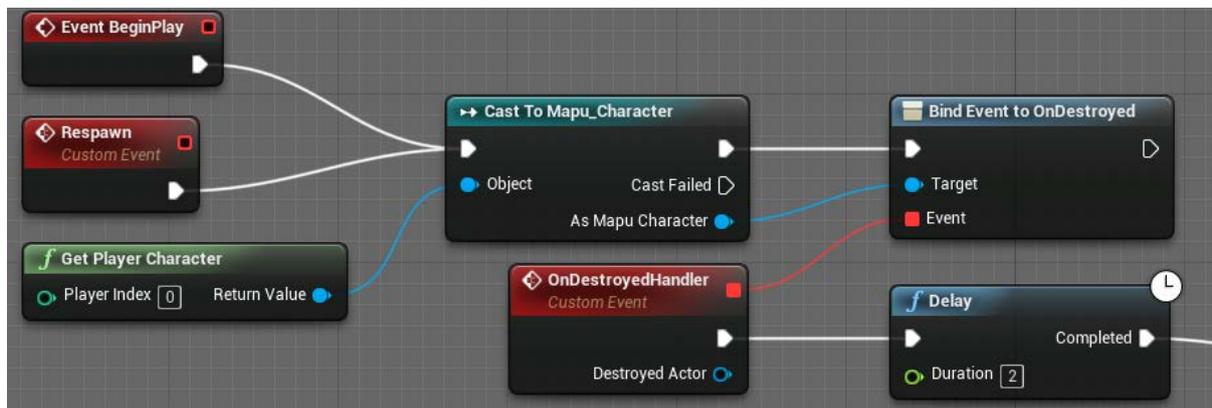


Figura 15 (A). Blueprint PlatformGameMode

En este diagrama se presentan varios nodos importantes. En primer lugar, los **Eventos** representados por un recuadro rojo. Su uso es el mismo que el de otros lenguajes, son nodos que comienzan la ejecución de ciertas funciones cuando son llamados desde el hilo principal del videojuego. Se utilizan para dar respuesta a

acciones que ocurren en la ejecución del juego. Por defecto, hay definidos numerosos eventos, como `BeginPlay`, cuando se inicia la partida, `Tick`, que se ejecuta en cada ciclo de ejecución del juego o eventos que sirven para manejar colisiones o eventos producidos por los periféricos de entrada. Adicionalmente, el desarrollador puede definir sus propios eventos personalizados.

También se utiliza otro concepto fundamental, el **Casting**. Mediante el Casting, se comprueba si un objeto de entrada es de una clase específica. Si es así es posible acceder y editar sus variables y funciones, así como activar eventos. Constituye así la principal vía de comunicación entre scripts. Además, se pueden especificar comportamientos diferentes según si el casteo tiene éxito o no.

Antes de analizar el diagrama, es necesario aclarar algunos conceptos acerca del funcionamiento de los módulos. De manera general, en el lado izquierdo de cada módulo se encuentran las señales de entrada, estando las de salida situadas a la derecha. Distintos tipos de variables son representadas por figuras de diferente color o forma, así, las variables de tipo **int** son representadas mediante un círculo color verde azulado, las de tipo **float** mediante uno de color verde oliva, los **booleanos** mediante uno rojo, los **objetos** mediante uno azul claro, etc. Entre esos símbolos hay que destacar el representado por una flecha blanca, que representa el **pin de ejecución**. Es decir, marca el flujo en el que se ejecuta el algoritmo. Si a la entrada de un nodo, este pin no está conectado, este nodo nunca se ejecutará. Por el contrario, si el pin de salida no está conectado, tras realizar la funcionalidad del nodo actual, el algoritmo se dará por finalizado.

Así, en esta primera parte de `PlatformGameMode`, podemos observar que, tras el evento `BeginPlay` y `Respawn`, se intenta realizar un casteo del personaje que está controlando el jugador como `Mapu_character` (esta clase se analizará en un apartado posterior). Dado que, como se ha indicado en el apartado 5.3.1 (Configuración y organización del proyecto), se estableció el peón por defecto como uno de esta clase, el casteo tendrá éxito.

A continuación, se hace uso de un manejador de eventos (**Event Dispatcher**) para llamar a una subrutina de código cuando se produzca un evento determinado. Podríamos decir que estos manejadores sirven como mensajeros entre diferentes scripts, avisan de que un determinado evento a ocurrido en uno de ellos.

Mediante el nodo “Bind Event to OnDestroyed”, cuando en el script del personaje principal (Mapu Character) se activa su evento OnDestroyed, en PlatformGameMode se inicia el evento *OnDestroyedHandler*.

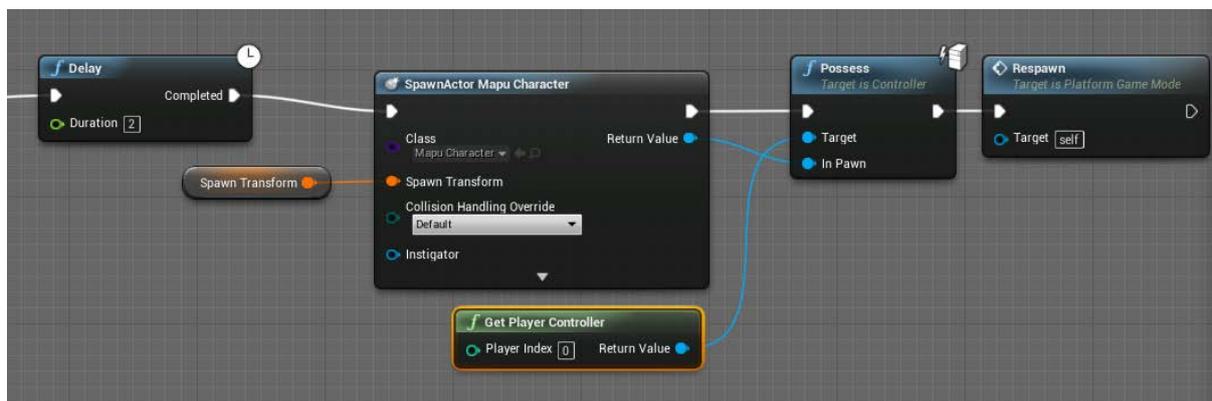


Figura 15 (B). Blueprint PlatformGameMode

Tras un retardo de dos segundos (nodo **Delay**), se genera un nuevo actor del tipo *Mapu\_Character* en la posición indicada por la variable *Spawn Transform* mediante el nodo **SpawnActor**. Tras ello, se asigna el controlador del jugador a este nuevo peón (nodo **Possess**) y se llama al evento *Respawn* (Figura 15 A), para volver a configurar la escucha del evento *OnDestroyed* de este nuevo peón.

### 5.3.2.2 CollectPetal

A lo largo del escenario del nivel de prototipo se encuentran diversos pétalos que el jugador debe recoger, a modo de **coleccionables** o **puntuación**. El comportamiento de estos pétalos viene determinado por el siguiente script, en el cual se diferencian tres funcionalidades.

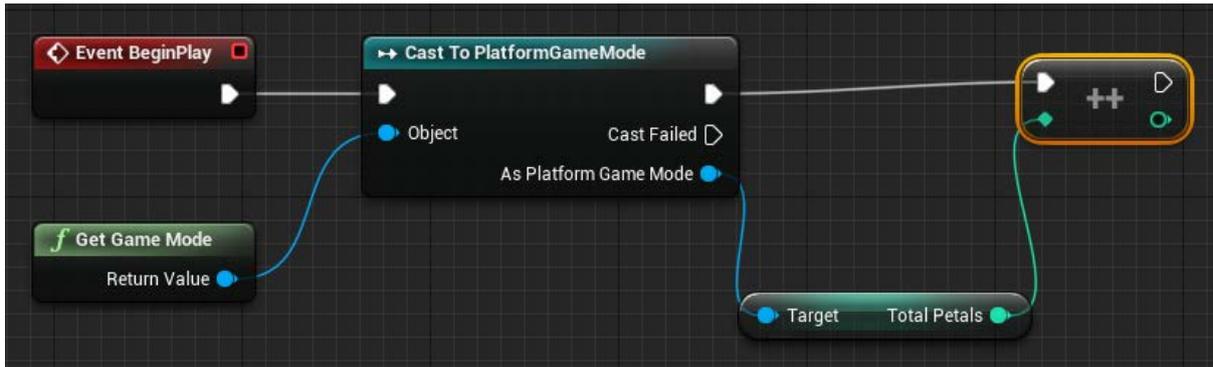


Figura 16 (A). Blueprint CollectPetal

Lo primero que se hace, al iniciar el juego (*Event BeginPlay*) es realizar un casting hacia *PlatformGameMode* e incrementar su variable *Total Petals*. Esto se hace en todas las instancias de *CollectPetals* presentes en el nivel, por lo que se calcula su número total de manera automática. Esta manera de proceder es útil a la hora de añadir o eliminar nuevos pétalos del nivel, ya que no se tendrá que cambiar de manera manual el valor de la variable *Total Petals*.

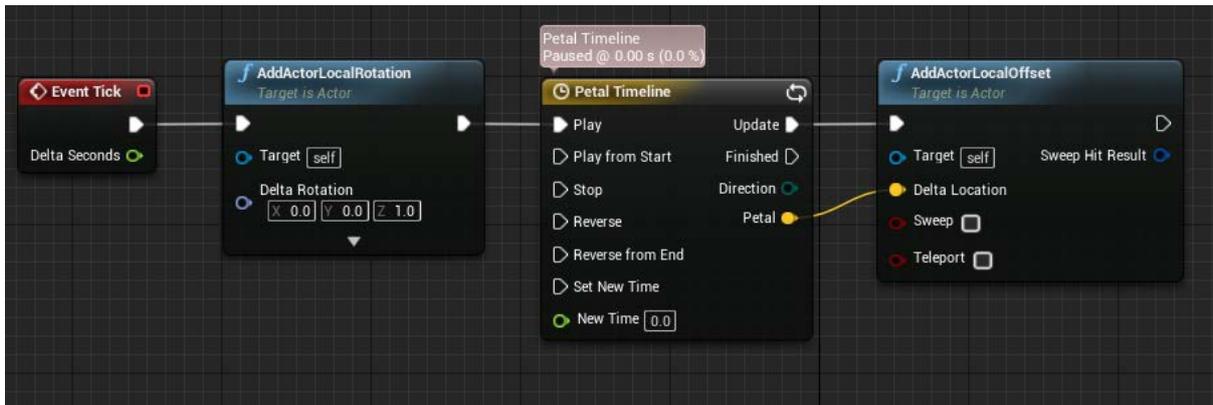


Figura 16 (B). Blueprint CollectPetal

Además, también se define otro comportamiento interesante. En cada ciclo de ejecución del proyecto (*Event Tick*), el actor rota un grado en el eje Z. Adicionalmente, se le aplica un offset en el mismo eje (altura) determinado por una línea temporal (*Timeline*) denominada *Petal Timeline*. Este *Timeline* se configura de la siguiente manera:

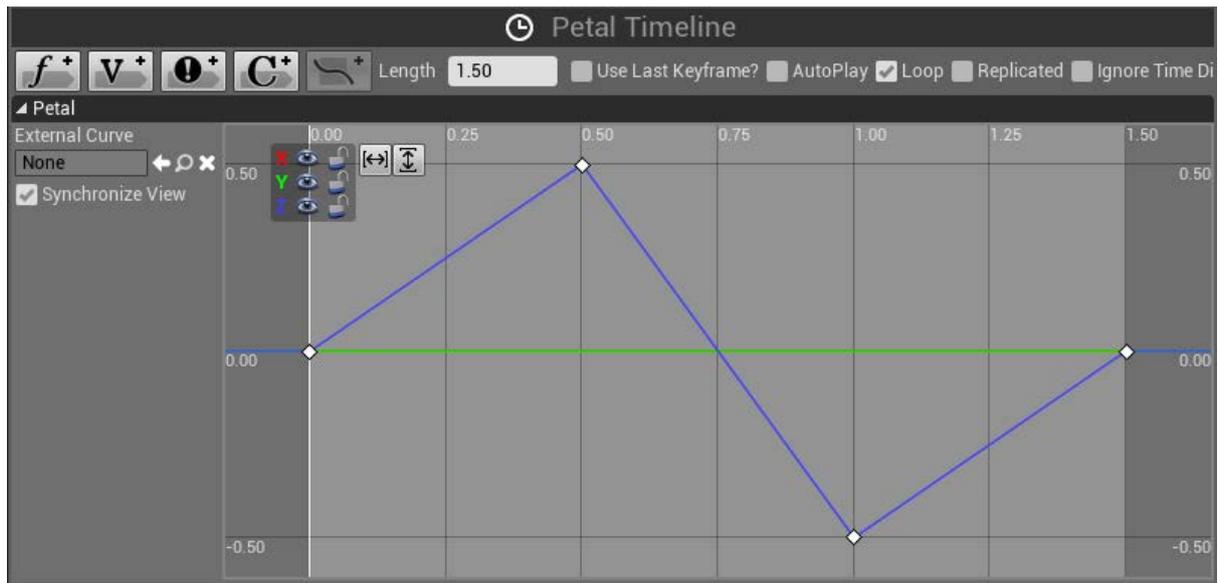


Figura 16 (C). Blueprint CollectPetal. Petal Timeline

Así, durante un período de 1.5 segundos se fluctúa entre un offset de  $\pm 0.5$  en el eje Z, mientras el eje X e Y permanecen constantes.

Finalmente, el script *CollectPetal* también define el comportamiento más interesante, cuando el personaje controlado por el jugador colisiona con un pétalo, este se destruye a la vez que se incrementa el valor de la variable *Petal Counter* definida en *PlatformGameMode* y se reproduce un sonido. Esto queda definido mediante el siguiente diagrama.

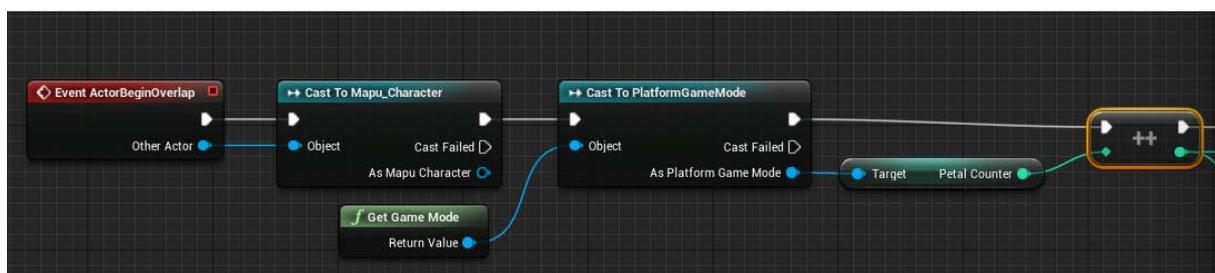


Figura 16 (D). Blueprint CollectPetal.

En este gráfico se puede apreciar un nuevo comportamiento del nodo **Cast To**. En el primer caso, sólo lo utilizamos para comprobar que el evento **ActorBeginOverlap** ha sido desencadenado por un actor de la clase *Mapu\_Character*. Si el casting tiene éxito, hacemos otro para referenciar a *PlatformGameMode* e incrementar *Petal*

*Counter*. El nodo **++** además de incrementar y almacenar la variable dada como entrada, permite utilizar el resultado de esta operación como salida. Así, se llega a la última sección del script.

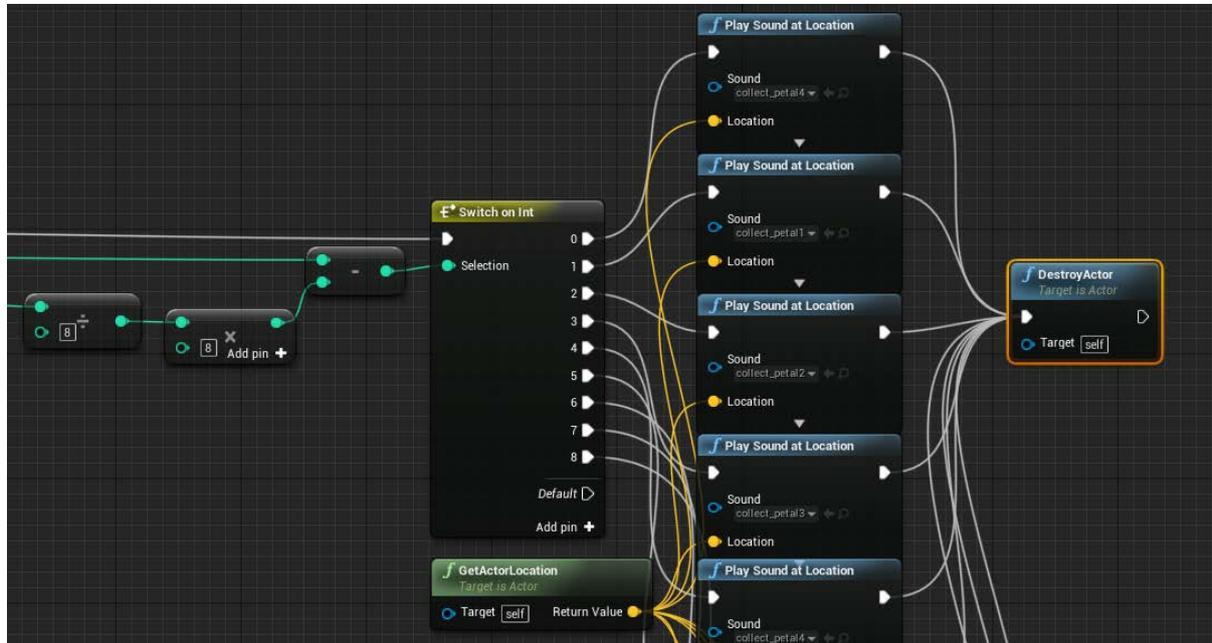


Figura 16 (E). Blueprint CollectPetal.

Esta sección puede resultar confusa. En primer lugar, se realizan operaciones matemáticas para determinar en qué lugar de una secuencia de ocho ha sido recogido el pétalo. Tras determinar este número, se entra a una estructura **Switch** en la que, según esta posición, se reproduce un sonido de entre una secuencia de ocho notas diferentes previamente determinada. Tras ello, se destruye el actor que implementa *CollectPetal*.

De este script surgen varios hijos en los que el valor del pétalo es diferente. Además, el script *CollectFruit* presenta un comportamiento similar.

### 5.3.2.3 LifeFlower

El Blueprint de la figura ha sido modificado para simplificar su explicación, dado que adicionalmente a lo aquí descrito también incluye efectos de sonido, estando el funcionamiento de estos explicados más adelante.

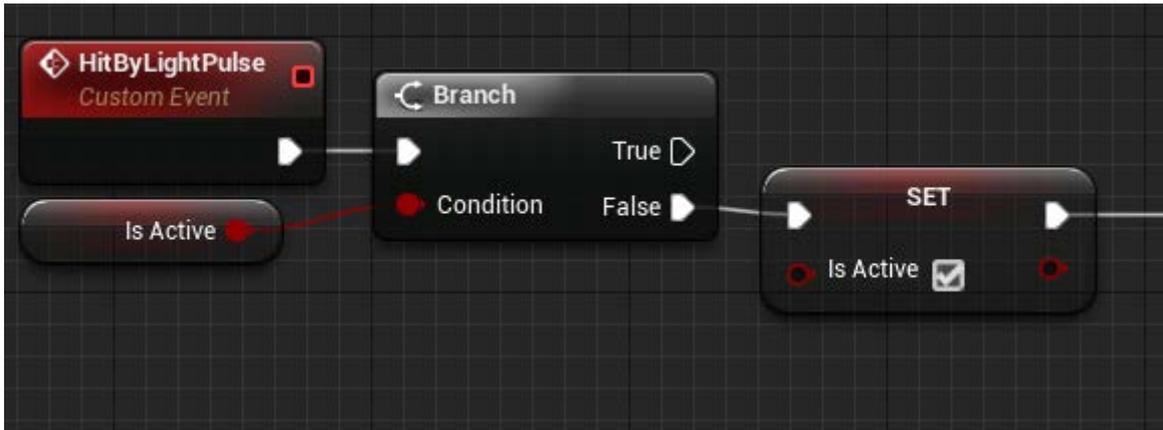


Figura 17 (A). Blueprint LifeFlower

En primer lugar, se define un evento personalizado denominado *HitByLightPulse*. Cuando se activa este evento, se comprueba si el actor ya ha sido activado mediante la variable booleana *IsActive*, definida de manera previa. Para ello, se utiliza el nodo **Branch**, el cual actúa como un **IF**. Si el actor no ha sido activado previamente, se activa ahora mediante el módulo **SET**.

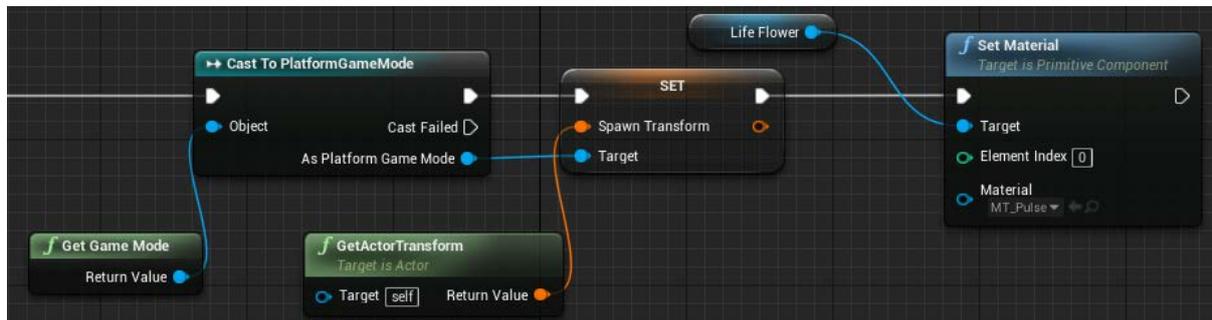


Figura 17 (B). Blueprint LifeFlower

Tras ello, se realiza nuevamente un casting a *PlatformGameMode* y se modifica la variable *SpawnTransform*. Finalmente, se cambia el material de *LifeFlower* referenciado con el índice 0 mediante el nodo **Set Material**. En este caso, la esfera que flota sobre la flor se ilumina, como se puede apreciar en la siguiente figura.



Figura 17 (C). Life Flower. Inactiva a la izquierda y activa a la derecha

Este script hace que estas flores actúen como **checkpoint**, tras ser activadas, si el jugador muere, renacerá en esta posición en vez de en la posición inicial.

Antes de explicar este script, es necesario mostrar los componentes que lo forman.

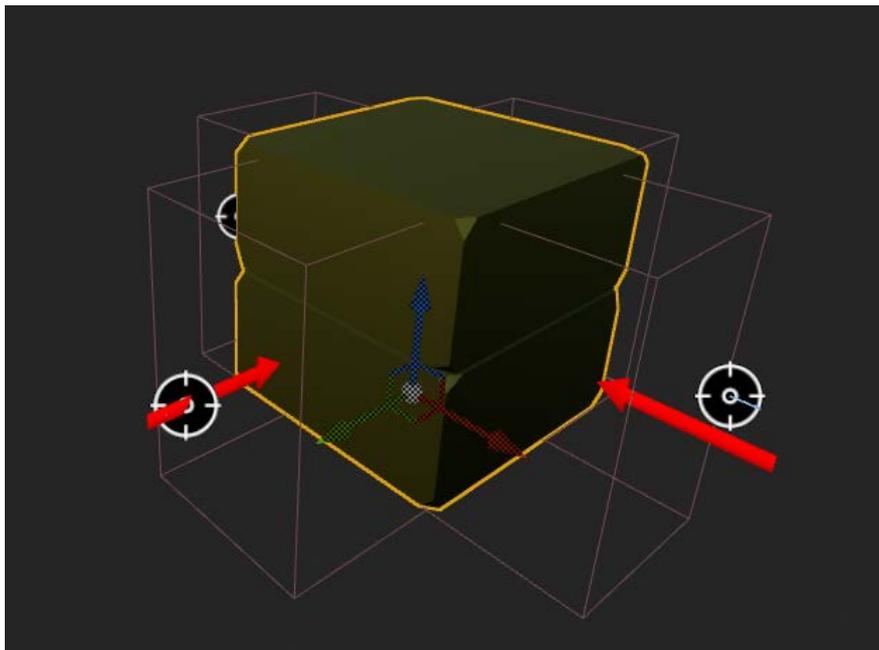


Figura 18 (A). Componentes de Pushable Box

Como se puede apreciar en la figura, en primer lugar, encontramos un modelo tridimensional, una malla poligonal o **mesh**, como es denominada. Este **mesh** consiste en una suerte de columna cúbica. En cada uno de los lados de este cubo, encontramos un **Volumen de colisión**, que se activará cuando un actor pase por

encima del mismo. También, cada lado cuenta con un componente **Flecha**, que sirve para indicar un vector dirección y un componente hijo que se utilizará para transportar al jugador a dicha posición cuando comience a empujar la caja.

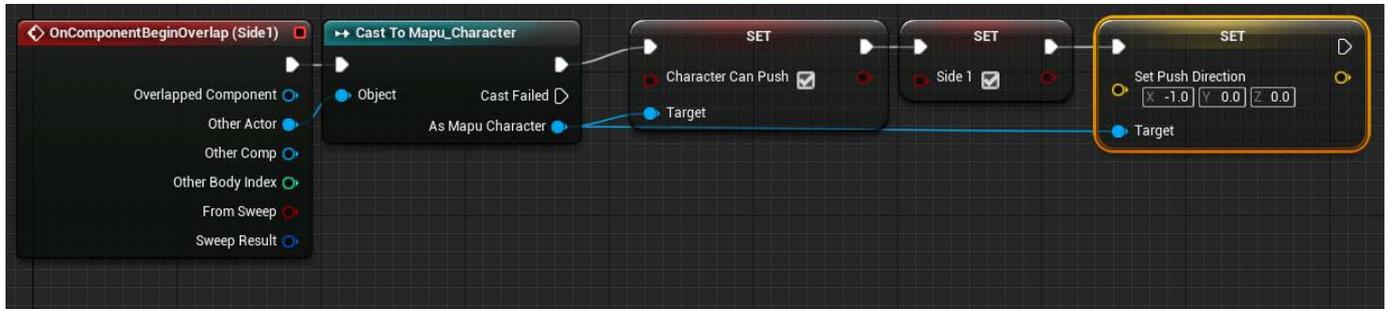


Figura 18 (B). Blueprint PushingBox

De igual manera que en *CollectPetal*, en primer lugar, comprobamos que se ha producido una colisión con *Mapu\_Character*. Sin embargo, en esta ocasión no se utiliza *ActorBeginOverlap*, sino que se utiliza ***OnComponentBeginOverlap***, específicamente en el componente *Side1* (uno de los volúmenes de colisión). Así, no se comprueba la colisión con el actor en su conjunto, sino solamente con uno de los componentes que lo forman. Tras un casting positivo, se activa la variable *CharacterCanPush* de *Mapu\_Character*, lo que desencadenará un comportamiento especificado en este blueprint. Asimismo, se activa la variable *Side1* en este actor y se establece la dirección de empuje en *Mapu\_Character*.

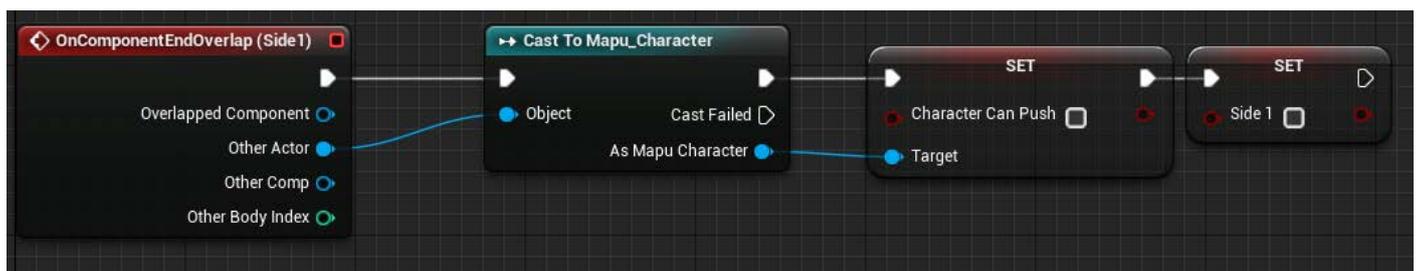


Figura 18 (C). Blueprint PushingBox

De manera contraria, cuando el actor abandona el volumen, se activa el evento ***OnComponentEndOverlap*** y se revierten los cambios realizados. Destacar que estos dos gráficos sólo definen el comportamiento de una de las caras del cubo, por lo que ha de repetirse para los tres lados restantes.

### 5.3.2.5 LightWaterFont

Este actor actúa a modo de **interrupor**. No se va a analizar el comportamiento completo de este script dado que, aunque es distinto y más complejo se compone principalmente de nodos ya tratados. Además, tiene asociada una variable *LightFontID* mediante la cual distintas instancias de este actor activan diferentes partes de la rutina. Lo más diferenciable de este componente consiste en lo siguiente:



Figura 19. Blueprint LightWaterFont

Este Una vez activado este actor (mediante un evento *HitByLightPulse*, de manera similar al script *LifeFlower*) y comparado su id mediante un nodo *switch*, encontramos este fragmento. A estas fuentes, se les referencia como parámetro una **secuencia** a activar y un sonido a reproducir.

Se explicará más tarde como se crea y configura una secuencia, las cuales se tratan básicamente de ciclos de animación mediante los cuales se definen cambios en las propiedades de otros elementos del nivel.

En el prototipo, por ejemplo, al activar una de estas fuentes, se activa la secuencia que hace que una plataforma comience a moverse, permitiendo al jugador avanzar y alcanzar nuevas áreas.

### 5.3.2.6 JumpingMushroom

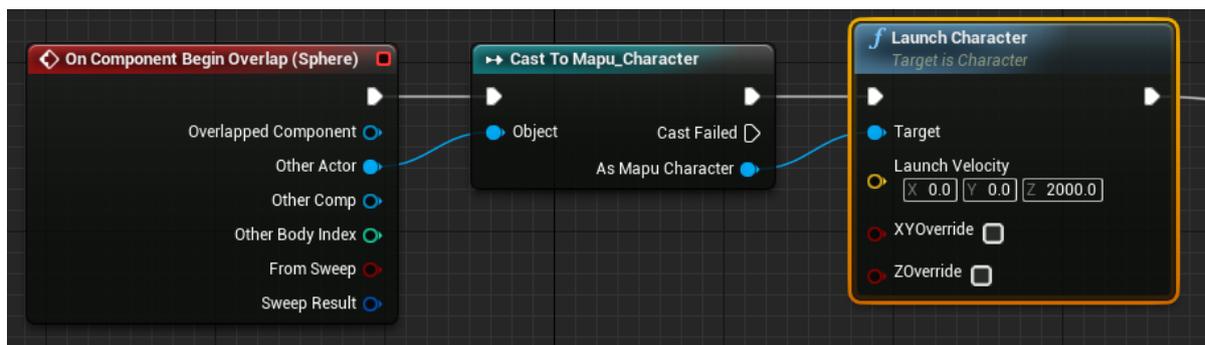


Figura 20(A). Blueprint Jumping Mushroom

El comportamiento de este script es sencillo. Mediante el uso del nodo **Launch Character**, cuando el jugador colisiona con el volumen denominado *Sphere*, es lanzado hacia arriba.



Figura 20(B). Cuando el jugador salta sobre la seta, es propulsado hacia arriba

### 5.3.2.7 Mapu Character

Este script es el más complejo del prototipo. Como se puede ver en la siguiente figura, en él se definen los comportamientos relacionados con el movimiento, el salto, el control de la cámara y las interacciones con otros actores.

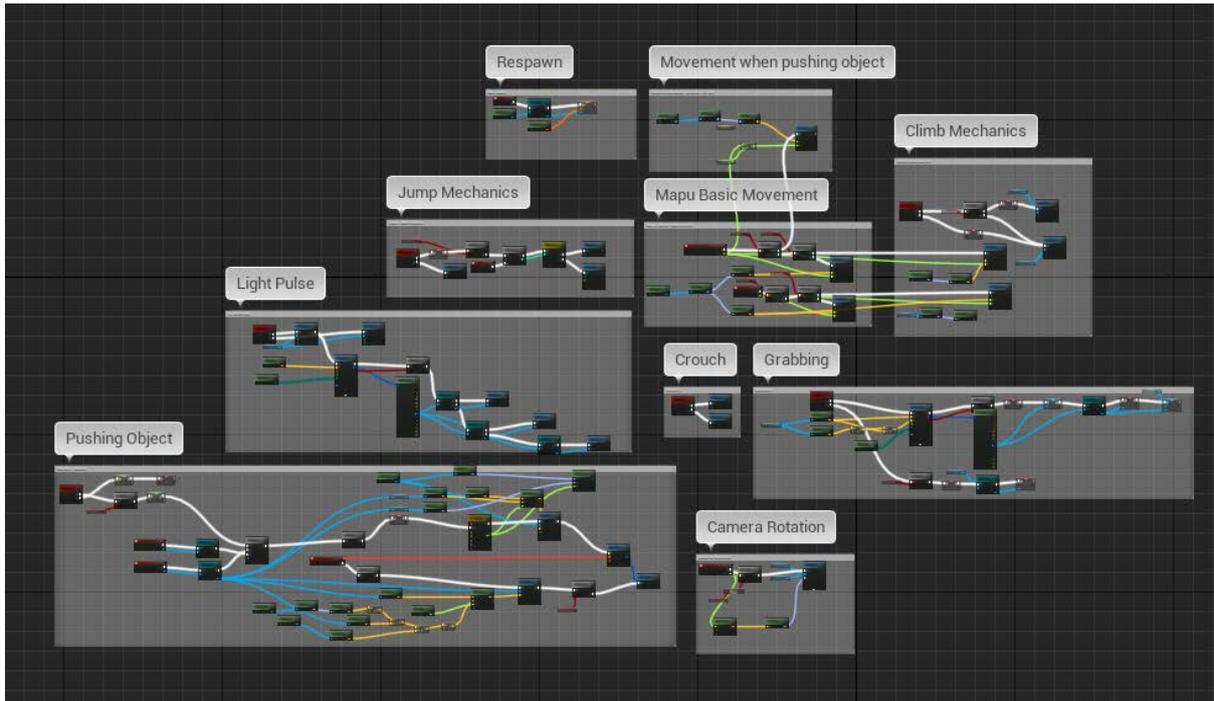


Figura 21. Blueprint Mapu Character

No se van a analizar todos los gráficos, sólo algunos de ellos compuestos por nodos interesantes.

### 5.3.2.7.1 Mapu Basic Movement

El funcionamiento de este *script* es el siguiente:

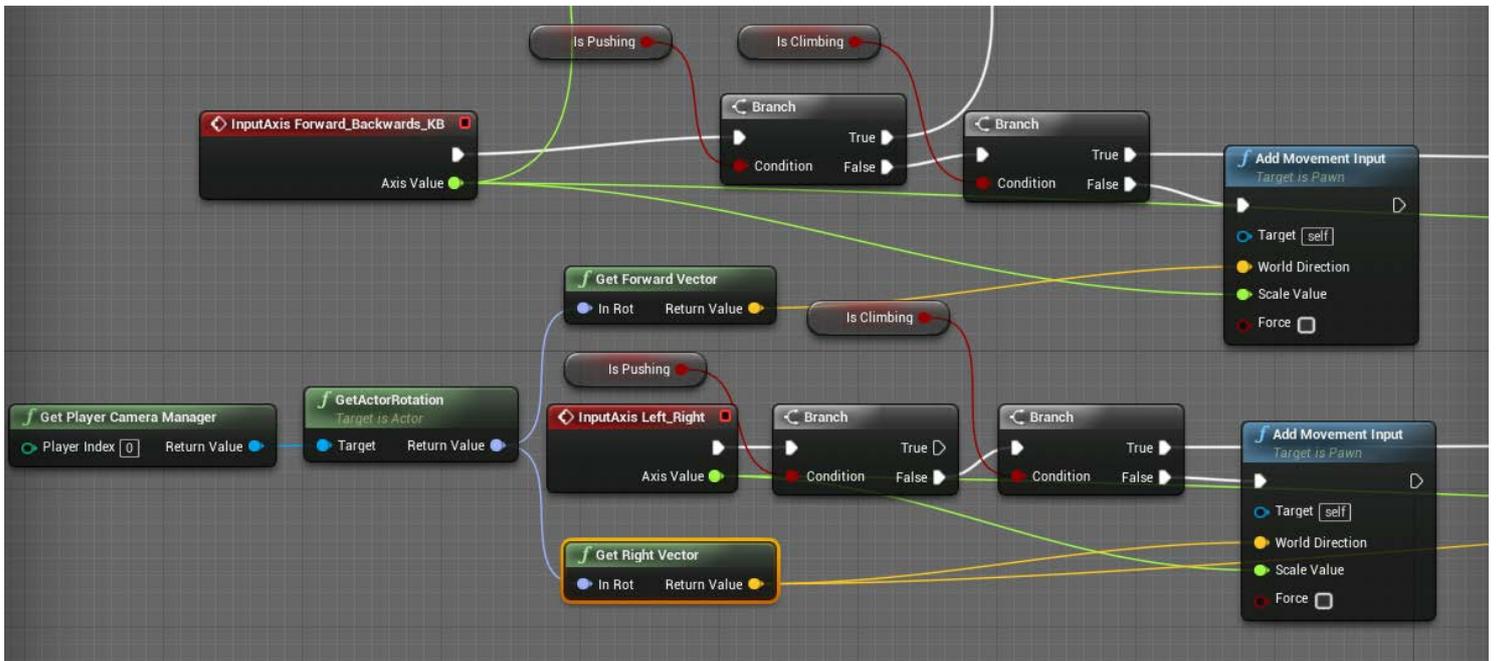


Figura 22. Blueprint Mapu Character. Mapu Basic Movement

Destacar, en primer lugar, los eventos **InputAxis Forward\_Backwards** e **InputAxis Left\_Right**. Estos eventos se activan cuando el jugador pulsa alguno de los botones asignados a esas acciones (según lo especificado en la sección 5.3.1, *Configuración y Organización del proyecto W y S* en el teclado y el eje vertical del **Joystick izquierdo** en mando para *Forward\_Backwards* y las teclas **A** y **D**, así como el eje horizontal del mismo *Joystick para Left\_Right*). Tras comprobar que el jugador no está empujando un objeto o escalando, se llama al nodo **Add Movement Input**. Como su nombre indica, hace que el personaje manejado por el jugador se mueva una cantidad, en este caso, definida en el propio *mapping* de los controles.

Además, este nodo necesita el parámetro de entrada de la dirección de movimiento. Este vector se consigue a través de los nodos **Get Forward Vector** y **Get Right Vector**. Es necesario destacar que estos vectores están unidos al vector rotación (**GetActorRotation**) de el manejador de la cámara (**Get Player Camera Manager**). Esto es así dado que el personaje se mueve, no según su dirección relativa sino mediante la dirección de la cámara del prototipo.

En el gráfico, se puede observar que, cuando se está empujando o escalando, la respuesta a los eventos es diferente.

### 5.3.2.7.2 Jump Mechanics

Las mecánicas de salto se especifican de la siguiente manera:

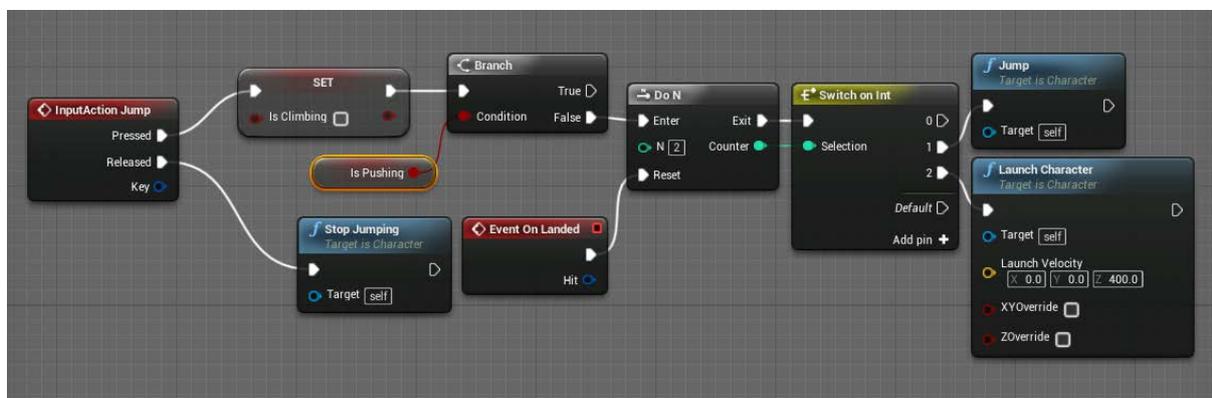


Figura 23. Blueprint Mapu Character. Jump Mechanics

Se puede observar que en este caso el evento que se ejecuta es del tipo ***InputAction*** y difiere a los eventos que controlaban el movimiento, que eran de tipo ***InputAxis***. Su comportamiento es diferente, en vez un eje con valores distintos en sus extremos, en este evento se establecen los estados ***Pressed*** y ***Released***. Así, cuando se pulsa el botón de salto (Barra espaciadora en teclado y botón inferior en mando), se activa el evento ***InputAction Jump***.

En primer lugar, se puede observar que tras pulsar el botón se establece la variable booleana ***IsClimbing*** como *false*. Esto ocasiona que, si se está escalando y se salta, el jugador deja de escalar. También se puede ver que, si el personaje está empujando un objeto, no se le permite saltar, ya que la rutina de salto sólo se ejecuta si la variable ***IsPushing*** es *false*.

Tras ello, la ejecución sigue a través del nodo ***Do N***. Este módulo permite el paso del hilo de ejecución al siguiente nodo un número de veces especificado por la variable de entrada N. En este caso, N es igual a dos. El contador se resetea cuando el jugador vuelve a tocar el suelo, mediante ***Event On Landed***. El valor del contador de Do N pasa a un nodo ***Switch*** que lleva, en caso de que el contador sea uno, al nodo ***Jump***, el cual ejecuta la acción de salto según viene definida en el componente de movimiento del actor. Si N es 2, se ejecuta el nodo ***Launch Character***, que ya se vio anteriormente.

Jugablemente, este comportamiento se traduce en una mecánica de ***doble salto***. Así, el jugador puede saltar una segunda vez mientras está en el aire, teniendo este salto menor impulso.

El mencionado componente de movimiento del personaje o ***Character Movement Component***, es un elemento que se genera por defecto para actores controlables y que maneja variables como la física, la velocidad, la gravedad que afecta al personaje, el peso del mismo, la potencia del salto, la capacidad de movimiento durante el salto, etc. Dado que se definen muchas más variables de las que se han usado en este prototipo, no se cubrirá su funcionamiento en profundidad en esta memoria.

### 5.3.2.7.3 Climb Mechanics

En esta sección se define la funcionalidad que permite que el jugador escale algunas superficies.

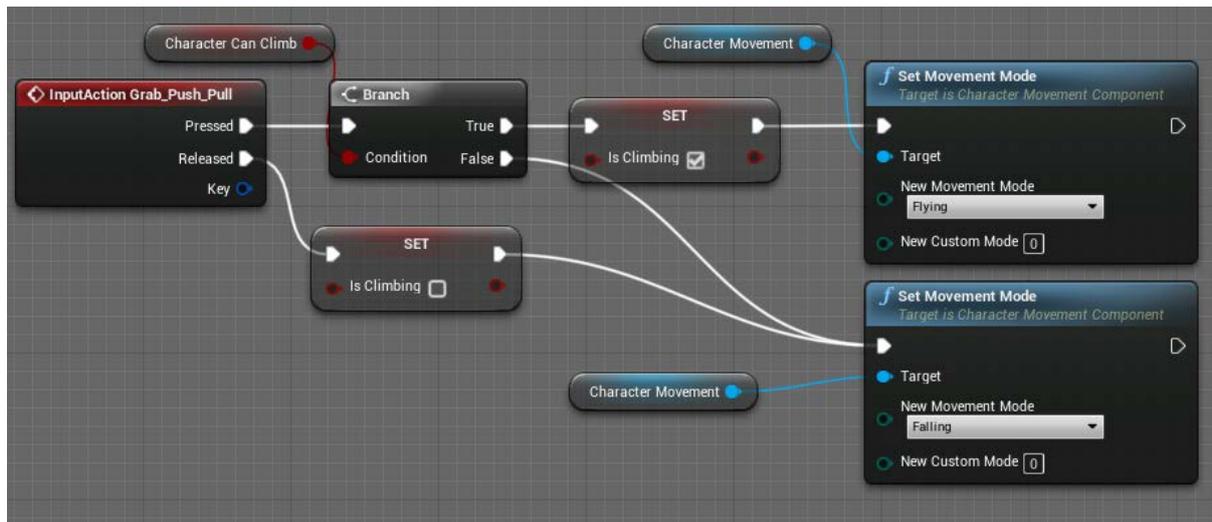


Figura 24(A). Blueprint Mapu Character. Climb Mechanics

Así, tras pulsar el botón de agarre (Click derecho de ratón o gatillo derecho en mandos) se activa el evento **InputAction Grab\_Push\_Pull**. Este botón de acción ha sido denominado así ya que es utilizado para varias acciones.

En este caso, se comprueba el valor de la variable **Character Can Climb**. Esta variable es, por defecto, *false*. Son los **objetos que pueden ser escalados** los que la configuran como true, de manera similar a la activación de la variable *Character Can Push* por parte de *Pushing Box* (de la sección 5.3.2.4 PushingBox).

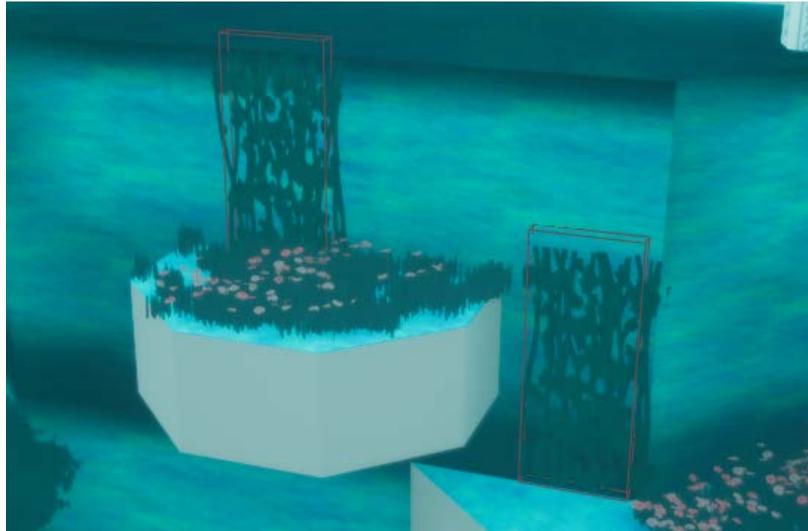


Figura 24(B). La variable *Character Can Climb* se activa cuando el jugador se sitúa en el volumen delimitado por líneas rojas en estas hiedras

Así, una vez comprobado el valor de esta variable se configura la variable booleana *Is Climbing* como true. Esto se hace para limitar el movimiento del personaje, como se vio anteriormente.

Tras ello, se utiliza el nodo **Set Movement Mode** para cambiar el modo de movimiento del jugador a **Flying**. Una de las características de Character Movement Component es que define varios modos de movimiento, como **Walking**, **Flying**, **Swimming** u otros modos personalizados. En este caso, se configura como *flying* debido a que el modo por defecto, *Walking*, no permite el movimiento en el eje Z como consecuencia de eventos del tipo *InputAxis*.

De manera análoga, cuando se deja de pulsar el botón de acción, se configura la variable *Is Climbing* con false y se vuelve al modo de movimiento normal (*Falling*, que cambiará automáticamente a *Walking* cuando se toque de nuevo el suelo).

El mencionado movimiento en el eje Z se logra con la siguiente parte del gráfico:

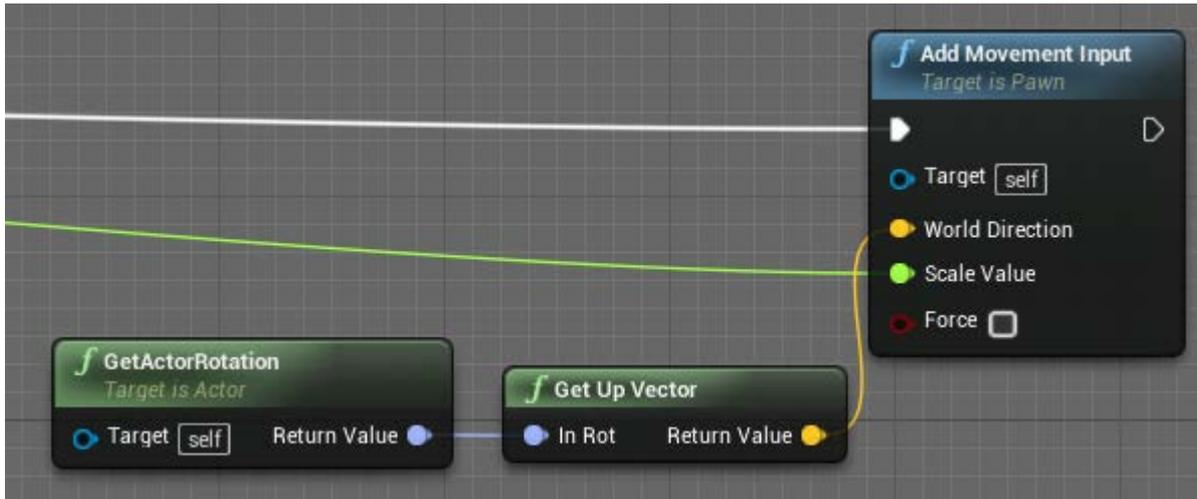


Figura 24(C). Blueprint Mapu Character. Climb Mechanics

Esta parte está directamente conectada a la comprobación del estado de la variable *Is Climbing* de la sección del movimiento básico (ver figura 22). Así, el comportamiento es similar a lo presentado en aquella sección, pero se sustituye el nodo *Get Forward Vector* por el nodo **Get Up Vector**. Además, en este caso se obtiene el vector de dirección del propio personaje, no de la cámara (aunque funcionaría de igual manera).

Así, si mientras esta escalando el jugador pulsa W el personaje se moverá hacia arriba, mientras que si pulsa S se moverá hacia abajo.

#### 5.3.2.7.4 Grabbing

Finalmente, se analizará el funcionamiento de parte del script que permite al personaje agarrar otros objetos, el cual incluye otros módulos interesantes.

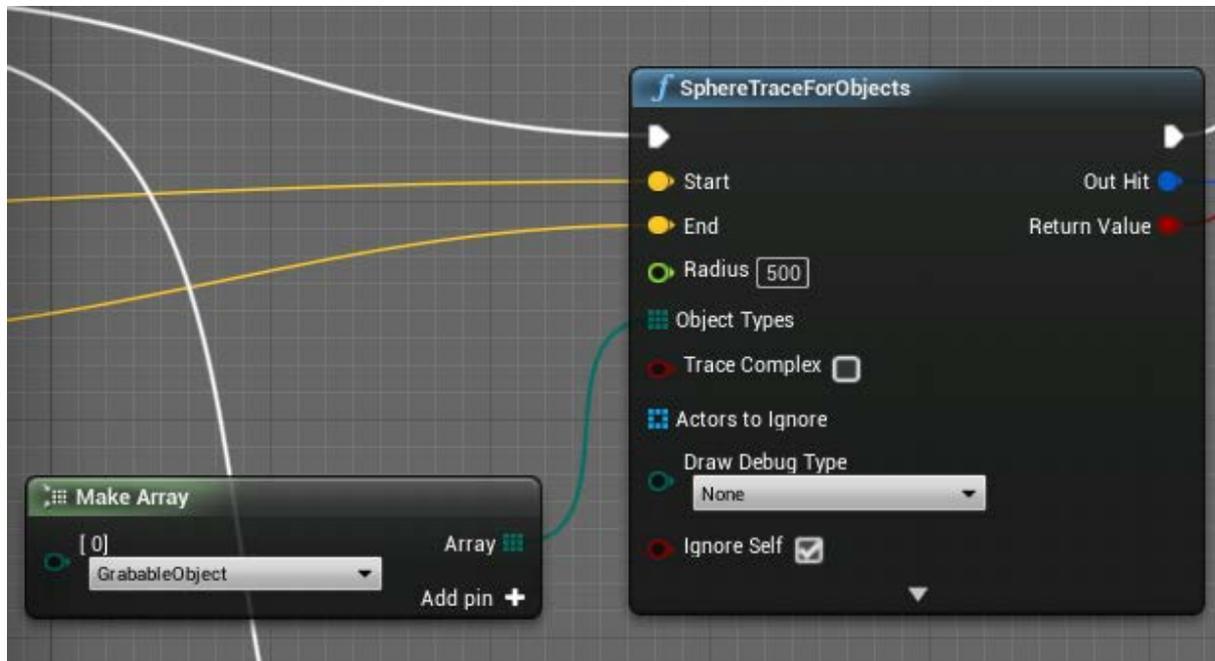


Figura 25(A). Blueprint Mapu Character. Grabbing

El módulo ***SphereTraceForObjects*** es llamado cuando el jugador pulsa las teclas que activan el evento ***InputAction Grab\_Push\_Pull***, también usado en las mecánicas de escalada. Este nodo realiza un barrido esférico en la línea resultante de los vectores de entrada ***Start*** y ***End***. Devuelve como ***Out Hit*** el primer objeto que encuentre. Además, si se le especifica como argumento un array de tipos de objetos, sólo se activará con objetos de ese tipo.

Se utiliza el nodo ***Make Array*** para pasarle la lista de tipos que se pretende que hagan reaccionar al barrido. En este caso, sólo queremos que sea activado con objetos del tipo ***GrabableObject***, tipo que se definió en la sección 5.3.1, Configuración y organización del proyecto.

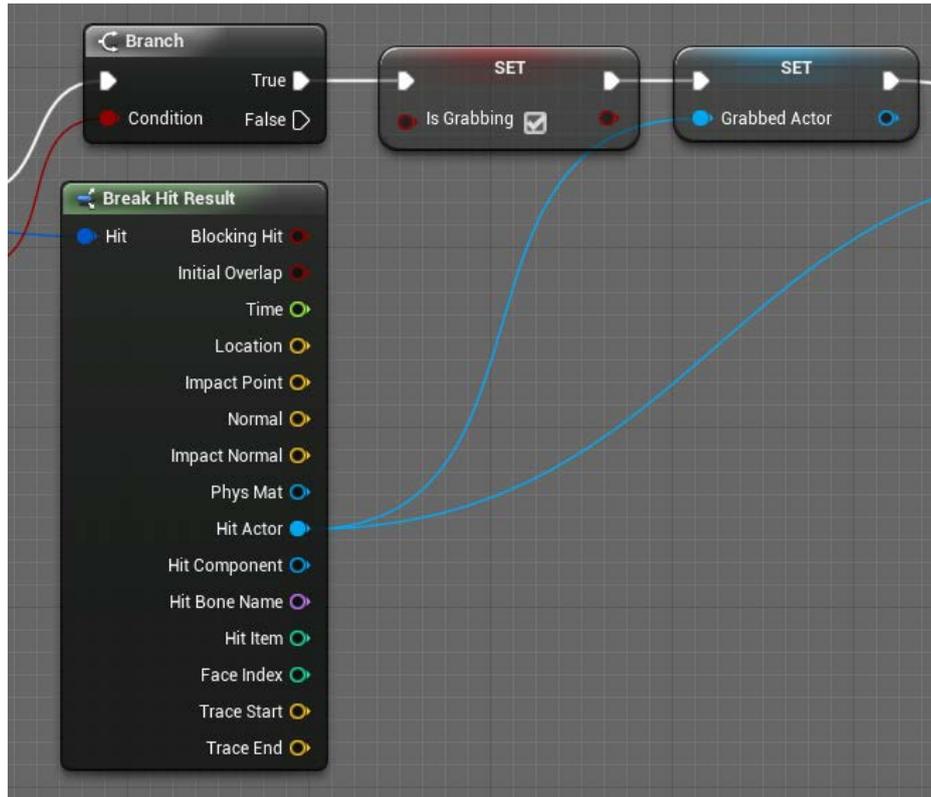


Figura 25(B). Blueprint Mapu Character. Grabbing

Así, cuando este barrido da resultado, *SphereTraceForObjects* devuelve tanto la estructura encontrada como una variable booleana que indica si ha tenido o no éxito.

Se comprueba esta variable mediante un nuevo *Branch* y se configura la variable *Is Grabbing* como *true*.

Se introduce aquí el nodo ***Break Hit Result***, el cual sirve para acceder a datos como el momento en el que se encontró el objeto, la localización del impacto, el componente impactado o el propio objeto impactado. En nuestro caso, se define la variable ***Grabbed Actor*** como el ***Hit Actor*** resultante de *SphereTraceForObjects*.

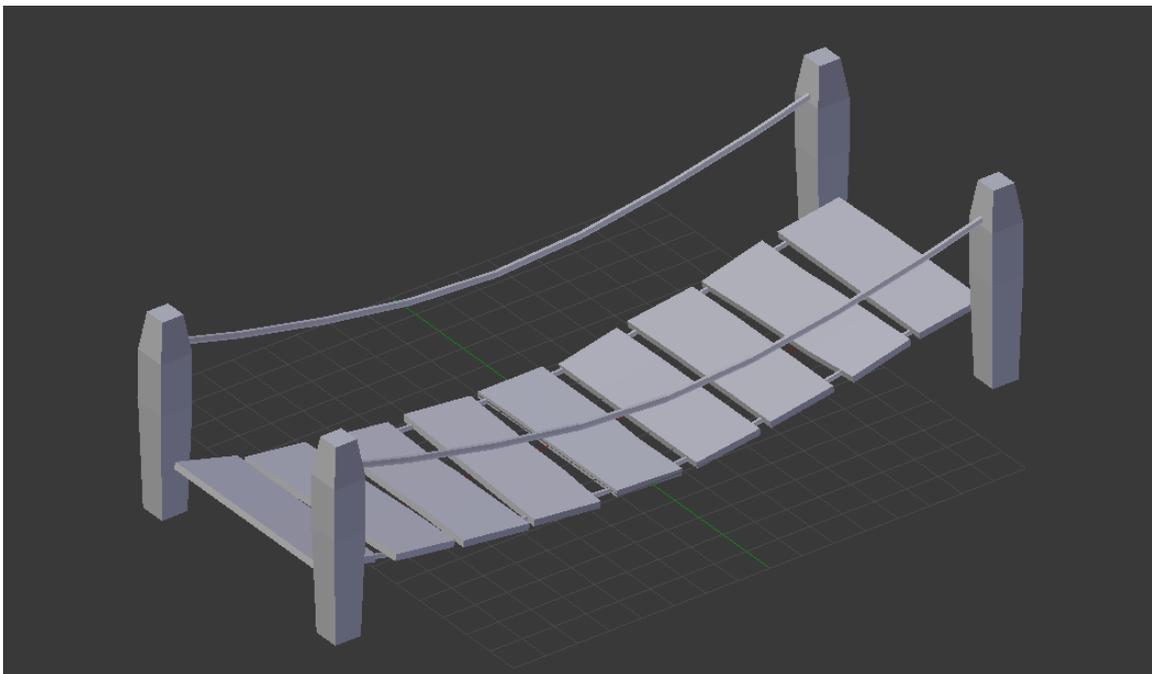
Aunque se han elaborado más scripts de los aquí analizados, gran parte del funcionamiento de los mismos viene dado por los módulos recogidos en esta memoria. Dado que resultaría redundante, no se analizarán todos ellos.

### 5.3.3 RECURSOS TRIDIMENSIONALES

Desde que la potencia de los computadores lo permitió, los gráficos tridimensionales han formado parte de la industria del videojuego. Y conforme esta potencia ha ido aumentando, los modelos en tres dimensiones han ganado sofisticación y, a la vez, complejidad. Hoy en día, al hablar de 3D se abarcan muchos campos, especializándose muchos profesionales sólo en uno de ellos. En esta memoria sólo se cubrirán algunos conceptos básicos, dado que tampoco se ha trabajado con ellos en profundidad.

#### 5.3.3.1 MODELADO 3D

El primer paso de este proceso consiste en el modelado poligonal de los elementos que más tarde se integrarán en el prototipo. Para ello, como se ha mencionado antes, se utilizó la herramienta **Blender**. No se elaboraron modelos muy complejos, se optó por el uso de elementos de bajo poligonaje, tanto por inexperiencia con esta disciplina como para optimizar el tiempo de trabajo.



*Figura 26. Modelo 3D de un puente colgante, presente en el prototipo.*

El funcionamiento de **Blender** no será cubierto en esta memoria, dado que el proceso de elaborar estos gráficos se acerca más a otras artes plásticas que a una ciencia exacta. Básicamente, se trata de, partiendo de polígonos básicos como planos, cilindros o cubos, modificar las distintas caras o vértices de las figuras hasta conseguir una malla poligonal que represente cierto objeto. Esto puede llegar a suponer un proceso costoso y complicado, por lo que la curva de dificultad en su aprendizaje suele ser alta.

Destacar en este campo la popularización de herramientas de **escultura digital**. En ellas, el modelo 3D no se genera mediante la extrusión o división de polígonos, por ejemplo, sino que se modela una figura como si fuera *arcilla digital*. De igual manera, esto se traduce en una malla poligonal. En un principio se experimentó con el uso de *ZBrush*, una herramienta de escultura digital, pero su uso se descartó.

Es necesario señalar que, a la hora de modelar objetos tridimensionales para videojuegos, se debe tener en cuenta el factor del **rendimiento** y la **optimización**, ya que una mayor carga poligonal afectará negativamente al rendimiento del videojuego.

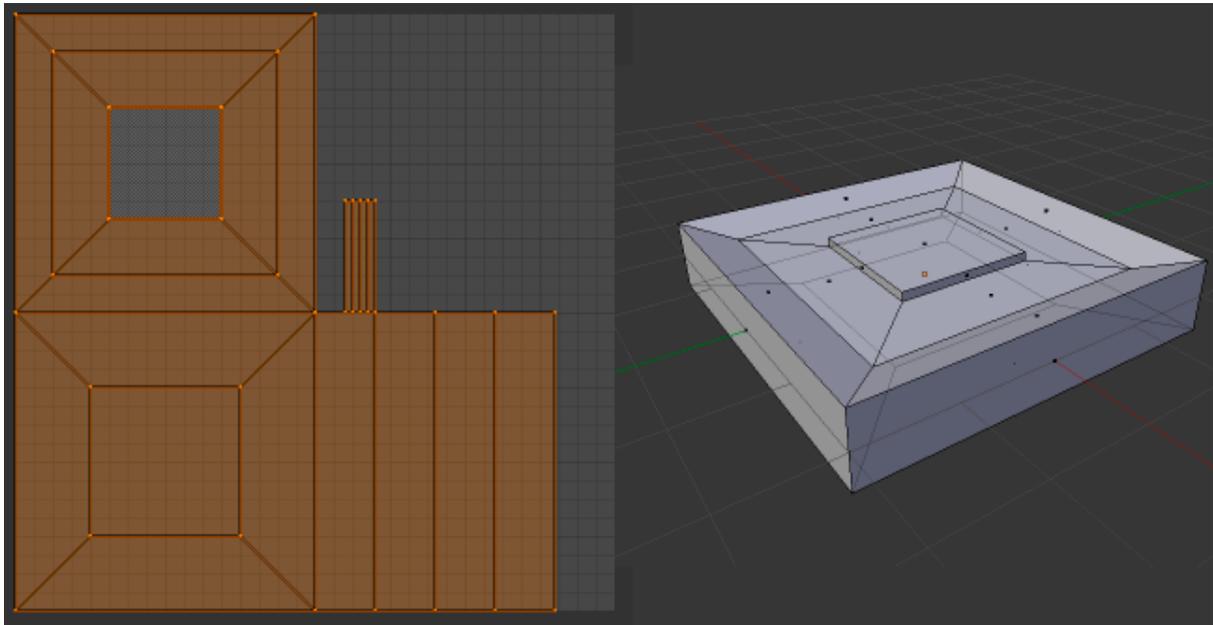
Respecto a este punto, dentro de Unreal Engine se hizo uso de **LODs** (Level of Details) para algunos elementos como la hierba. Mediante esta técnica, se puede asignar diferentes modelados al mismo objeto. Así, cuando un objeto se encuentra, se utilizan modelados más sencillos que cuando está cerca del jugador, de manera que mejora el rendimiento y la distancia impide apreciar la diferencia visual.

#### 5.3.3.2 UV EDITING

Tras este proceso de modelado 3D se obtiene una figura tridimensional, con mayor o menor detalle, que ha de pasar por un proceso de **UV Editing**. Este proceso consiste en el proyectar una imagen en dos dimensiones a la figura tridimensional.

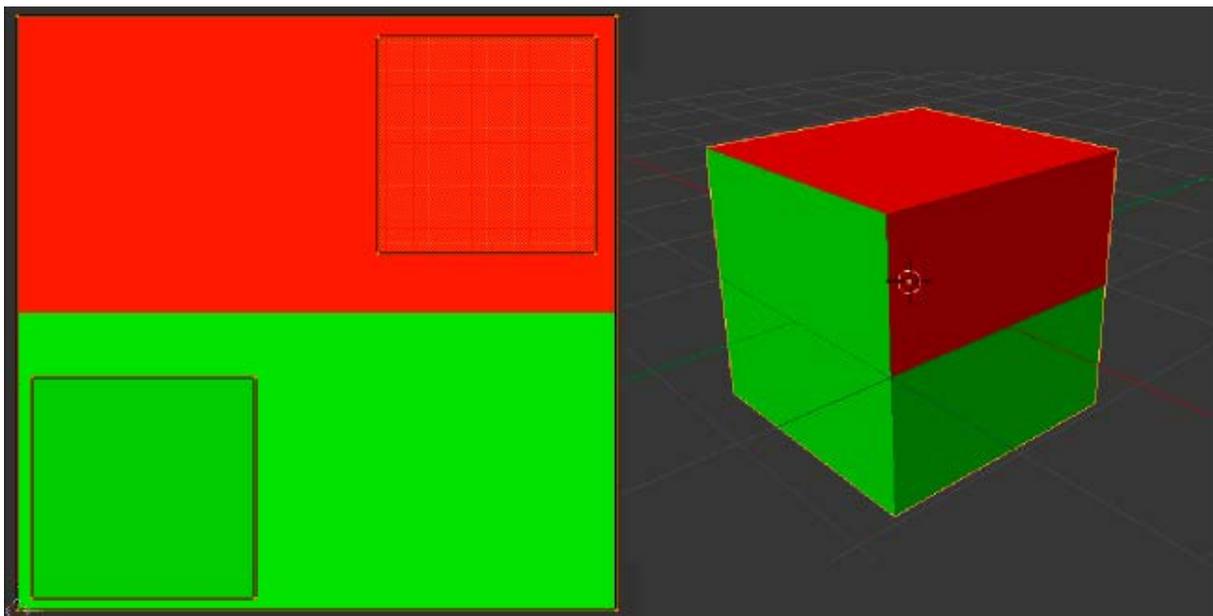
Es un proceso necesario para el posterior texturizado el modelo. Básicamente, se seleccionan las diferentes caras formadas por la estructura poligonal y se le asigna

un área determinado en un plano bidimensional. Esto se puede apreciar mejor en la siguiente imagen.



*Figura 27(A). A la izquierda, mapeado UV de la malla representada a la derecha*

Así, una vez asignado un material o textura a este modelo, se proyectarán los gráficos bidimensionales indicados por este mapeo UV a las caras de la malla tridimensional correspondientes. Este comportamiento se puede observar en el siguiente ejemplo.



*Figura 27(B). Ejemplo de UV Mapping*

Se ha realizado un UV Mapping de ejemplo para el cubo de la figura. Así, también se ha creado una textura consistente en dos colores, rojo y verde.

Se puede observar que, a la cara de la derecha del cubo, se le ha asignado un mapeo correspondiente a la textura en su conjunto. Por otro lado, a las otras dos caras visibles, se les han asignado áreas más pequeñas, una correspondiente con la zona verde y otra correspondiente con la zona roja.

En este caso, el mapeo es sencillo dado que el modelo tiene pocas caras. Sin embargo, cuando más complejo es el modelo, este proceso de mapeo también se hace más complicado.

#### 5.3.3.3 TEXTURE MAPPING

Este es, quizás, uno de los campos que más complejidad ha adquirido a lo largo de los años. Originalmente consistía únicamente en la extrapolación de un recurso en dos dimensiones hacia una superficie tridimensional, como el proceso seguido en la figura anterior, 27(B).

Sin embargo, la necesidad de ofrecer gráficos con mayor detalle al mismo tiempo que se optimiza el rendimiento ha llevado a que actualmente se trabaje con multitud de texturas complementarias. Así, mientras que en su origen el proceso constituía solamente lo que hoy se conoce como **diffuse mapping**, actualmente se trabaja con muchas más capas que aportan nuevos comportamientos y detalles a la hora de renderizar el objeto tridimensional.

Podemos analizar el aspecto del siguiente objeto:

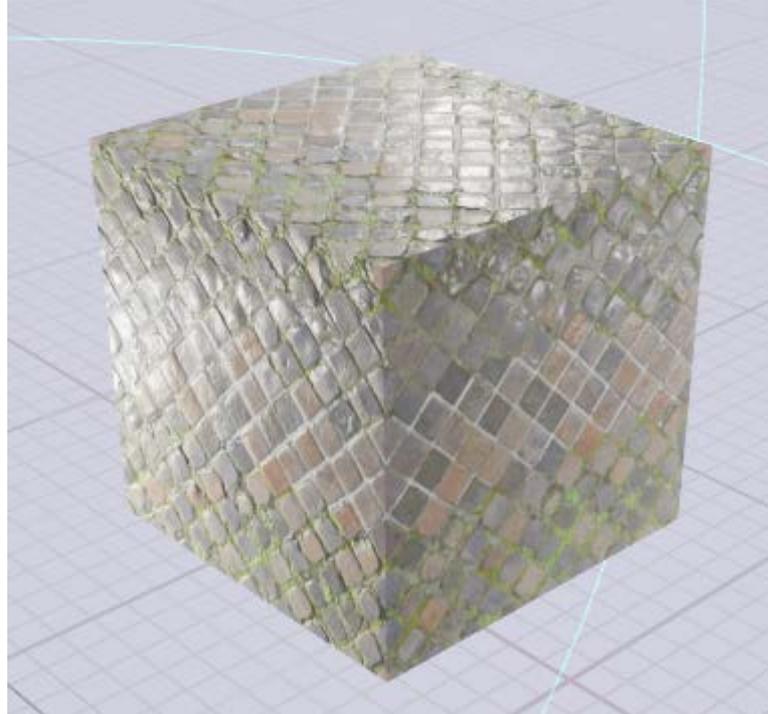
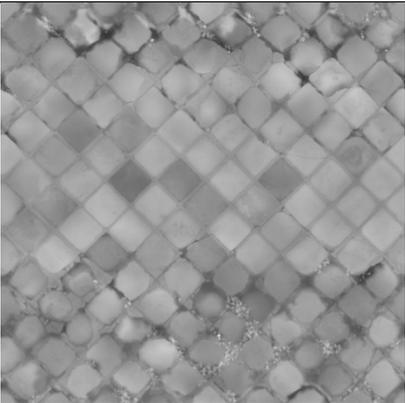
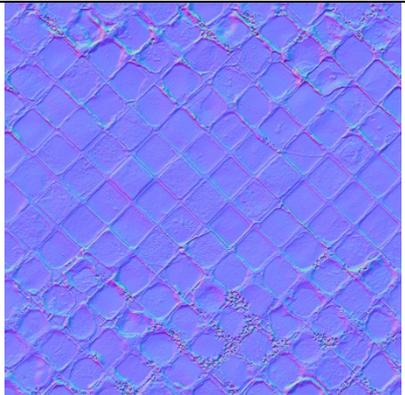
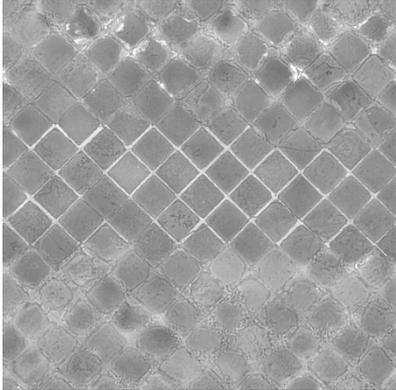
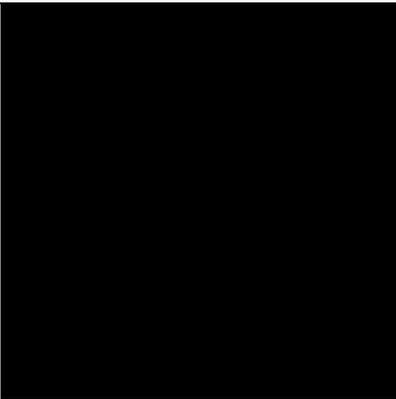
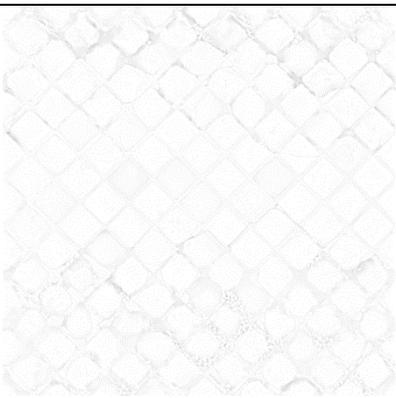


Figura 28. Ejemplo de material. Autor del material: Allegorithmic. Substance Source

Se puede ver como, a pesar de ser simplemente un cubo, la superficie del mismo tiene un aspecto complejo, pudiendo apreciarse detalles del adoquinado que dan una sensación de volumen más rica que la que aporta simplemente la malla poligonal. Ese volumen, la manera en que se comporta cada parte tras ser expuesta a fuentes de luz y otros aspectos, vienen definidos por diferentes texturas o **mappings**. El conjunto de todas ellas es lo que se conoce como material, que será tratado en el siguiente apartado. Algunos de estos mapas se explicarán en la siguiente tabla, sirviendo el material de la figura 28 como ejemplo.

	<p><b>Diffuse map</b></p> <p>Contiene la información básica referente al color del modelo.</p>
	<p><b>Height map</b></p> <p>Se trata de una imagen en escala de grises en la que se representa un desplazamiento en la dirección normal a la cara del polígono correspondiente. El negro representa la altura mínima, mientras que el blanco representa la altura máxima. Funciona indicando donde se generarán las sombras del modelo sin incrementar la carga poligonal del mismo.</p>
	<p><b>Normal map</b></p> <p>Su comportamiento es parecido al del Height map, pero en este caso se utiliza una imagen RGB en la que cada uno de los canales representan los ejes X, Y y Z. Se utilizan para añadir gran cantidad de detalles sin incrementar la carga poligonal.</p>

	<p><b>Roughness map</b></p> <p>Define irregularidades en la superficie, las cuales reflejarán la luz en distintas direcciones. Así, el negro representa superficies suaves y el blanco superficies ásperas.</p>
	<p><b>Metallic map</b></p> <p>Mediante una imagen en escala de grises se define qué áreas del <i>diffuse map</i> deben ser interpretadas como metálicas, siendo estas reflectantes. El negro (0) representa superficies no metálicas, mientras que el blanco (1) representa superficies totalmente metálicas. En este ejemplo, dado que no hay partes metálicas la textura es negra por completo.</p>
	<p><b>Ambient Occlusion map</b></p> <p>Mediante una escala de grises, define qué cantidad de la luz ambiental es accesible a cada punto de la superficie. Crea un sombreado suave, especialmente útil en áreas cóncavas del modelo.</p>

Estos son algunos de las texturas más importantes a la hora de diseñar un material, sin embargo, hay muchas más como el ***Opacity map***, ***Specular map***, ***Glossiness map***, ***Displacement map*** o ***Reflection map***.

Uno de los métodos de aumentar el nivel de detalle de un modelo con baja carga poligonal, es la realización del mismo modelo con alta carga poligonal y exportar esa

información como un *Normal map* que será utilizado en el primer modelo. En resumen, estos mapas de texturas constituyen una herramienta útil a la hora de añadir más detalles y efectos a la composición visual de cualquier medio tridimensional.

#### 5.3.3.4 MATERIALES

Un material consiste en un conjunto de texturas que sirven para definir el aspecto y comportamiento de una superficie. Durante la elaboración del prototipo se siguieron varios procesos. En la mayoría de casos, se utilizó Adobe Illustrator y Adobe Photoshop para elaborar las texturas, las cuales luego fueron importadas a Unreal Engine 4. En algunos materiales, estas texturas fueron creadas haciendo uso de Substance Designer, de Allegorithmic. Finalmente, también se crearon materiales directamente mediante el editor de materiales del propio Unreal Engine, que proporciona algunos efectos interesantes.

##### 5.3.3.4.1 MATERIALES A PARTIR DE PHOTOSHOP

El primer proceso que analizar es el flujo de trabajo desde **Adobe Photoshop/Illustrator** a Unreal Engine. Así, mediante herramientas de diseño gráfico se elaboran las texturas que más tarde se aplicarán a los objetos. Por ejemplo, las texturas de la siguiente figura se aplicaron en el modelo que se vio en la Figura 27(A).

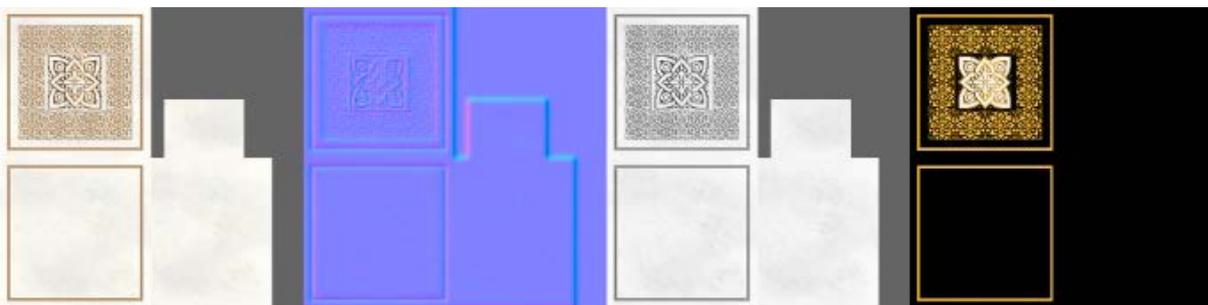


Figura 29(A). Texturas realizadas mediante Adobe Illustrator y AdobePhotoshop

Estas texturas serán importadas en Unreal Engine y serán utilizadas en su editor de materiales, el cual tiene un funcionamiento similar a Blueprints.

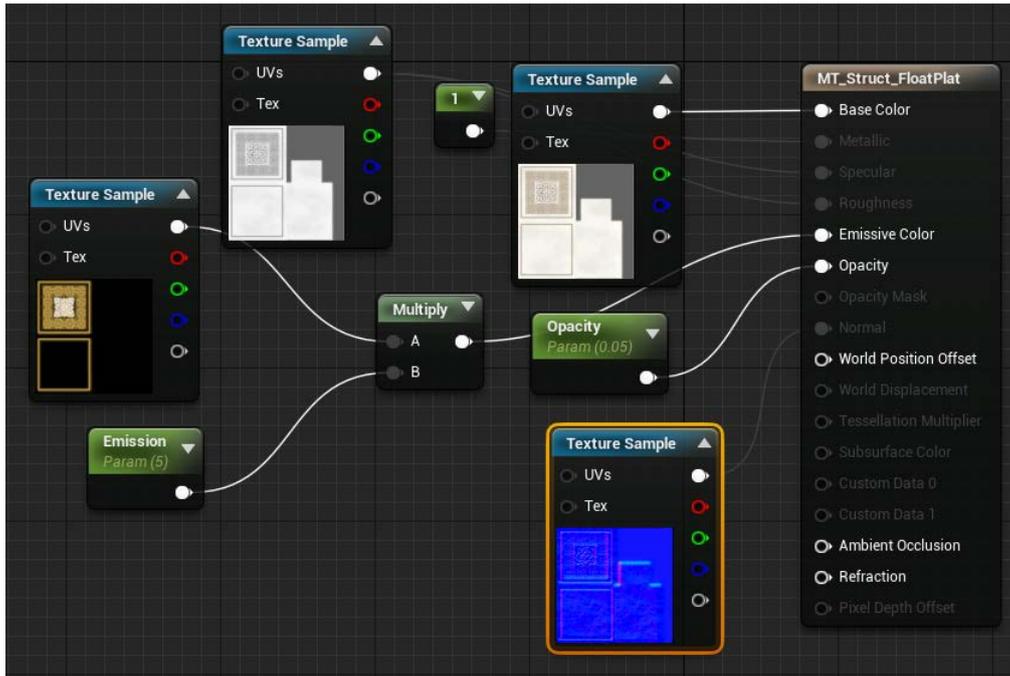


Figura 29(B). Material definido por las texturas anteriores

Este proceso de texturizado puede ser más o menos complejo dependiendo de la complejidad del modelo tridimensional y de la propia textura. El material que viene definido por el gráfico anterior sólo es utilizado para un modelo tridimensional en concreto, por lo que se podría considerar que estos materiales específicos suponen un esfuerzo y coste temporal razonable, aunque la textura a aplicar, como en el caso anterior, sea relativamente sencilla.



Figura 29(C). Otros objetos texturizados mediante esta técnica

Para algunas texturas de carácter más orgánico se utilizó **Substance Designer**. En este software, mediante un sistema de nodos se pueden definir materiales mediante un flujo de texturas y efectos que permiten la rápida modificación de elementos de la cadena, viéndose estos reflejados en el resultado final de manera automática. Se utilizó este software para elaborar materiales como hierba o rocas.

Así, mediante nodos que deforman, duplican, superponen gradientes, introducen repeticiones aleatorias, etc. se pueden conseguir materiales de gran calidad a los que se les puede hacer variaciones y ajustes de manera cómoda y sencilla.



*Figura 30(A). Material de césped elaborado mediante Substance Designer*

Por ejemplo, el material correspondiente al terreno sobre el que está situado la roca de la imagen viene determinado por el siguiente gráfico.

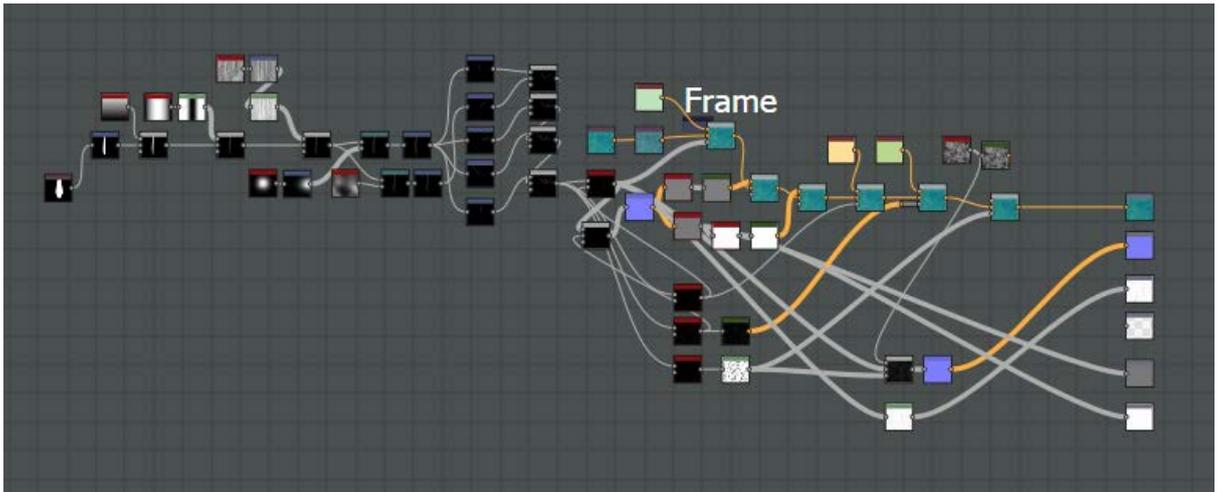


Figura 30(B). Gráfico del material elaborado mediante Substance Designer

No se va a entrar en el funcionamiento en profundidad del programa, pero, como se puede intuir en la imagen, a la hora de elaborar materiales complejos el proceso puede ganar dificultad. Por ello, sólo se utilizó este flujo de trabajo en materiales puntuales.

#### 5.3.3.4.3 MATERIALES MEDIANTE UNREAL ENGINE 4

Independientemente del origen de las texturas, estas tuvieron que ser importadas en Unreal Engine para formar el material. En algunos casos en los que las texturas de los objetos consistían en colores planos, estos materiales se pudieron crear directamente desde el editor de materiales del motor.

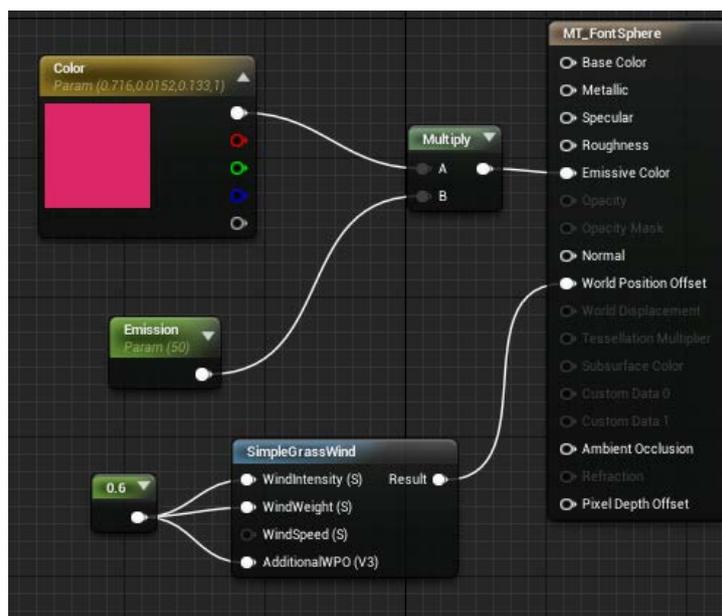


Figura 31(A). Ejemplo de material sencillo elaborado en Unreal Engine 4

En la figura anterior se puede observar como, a partir de elementos como colores o valores numéricos se puede definir un material sin usar texturas elaboradas en herramientas externas. Elementos como colores o valores numéricos se pueden parametrizar para crear nuevos materiales, instancias del material original en las que se puedan cambiar esos valores fácilmente sin tener que definir un nuevo gráfico.



*Figura 31(B). Materiales que hacen uso de SimpleGrassWind*

Además, Unreal Engine también permite el uso de efectos visuales interesantes. Por ejemplo, uno de ellos es **SimpleGrassWind**, el cual simula una deformación causada por el viento (del que se pueden definir varios parámetros). Este efecto fue utilizado en las hojas de los árboles, en las flores o en el césped. Otros nodos pueden ocasionar una variación de color en función de la posición del objeto (útil en árboles y arbustos), la superposición de texturas, las cuales se desplazan con el tiempo (usado en el agua, para dar sensación de superficie irregular), o el uso de texturas animadas.

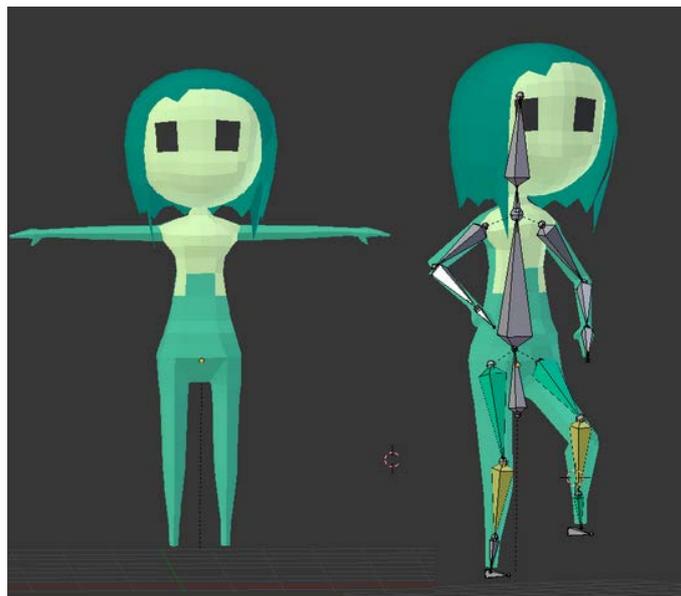
En resumen, se hicieron uso de multitud efectos para que la presentación visual del prototipo fuera más rica y causara mejores sensaciones.

### 5.3.3.5 RIGGING

El último punto que se trató en el desarrollo del proyecto fue el Rigging. Este proceso es el paso previo a la animación de modelos en 3D.

Consiste en la creación y asignación de un esqueleto a un modelo 3D, que servirá para deformar el mismo. Debido a limitaciones temporales, este punto quedó sin concluir, no llevando a cabo tampoco, por tanto, el proceso de animación.

El modelo con esqueleto que se elaboró es el que se puede ver en la siguiente figura. El esqueleto lo componen las estructuras grises que se pueden observar a la izquierda.



*Figura 32. Modelo en 3D con Rigging*

A pesar de que este modelo es apto para animación, su funcionamiento es mejorable. Debido a la dificultad del proceso, este modelo no se implementó en el prototipo, siendo uno de los apartados pendientes.

#### 5.3.4 SISTEMAS DE PARTÍCULAS

Fueron realizados dos sistemas de partículas dentro del prototipo. Un sistema de partículas podría definirse como una simulación de un fenómeno físico, como lluvia, nieve, explosiones o rayos. Podrían considerarse como los *efectos especiales* de un videojuego.

Así, en este caso de estudio se elaboró un efecto de niebla y un efecto de luciérnagas, los cuales se pueden observar en la siguiente figura.

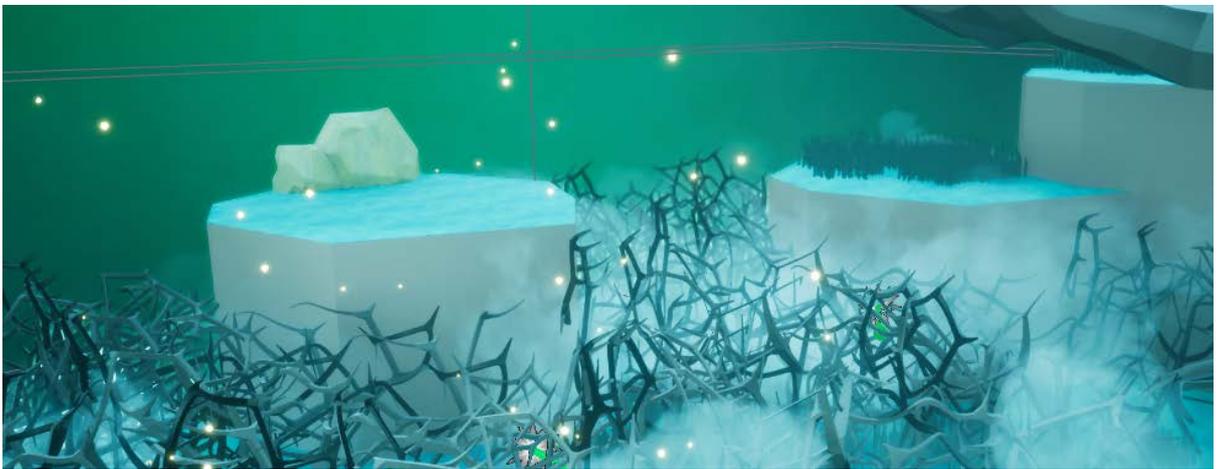


Figura 33. Sistemas de partículas

Estos sistemas de partículas se generan a través de otro editor propio dentro de Unreal Engine. En este, se configura un generador de partículas de manera que se definen variables como la ratio de generación de partículas, el tiempo que pasa hasta que desaparecen, su rotación, su velocidad y dirección de movimiento, etc.

De igual manera que los apartados anteriores, su complejidad puede llegar a ser alta.

### 5.3.5 SONIDOS

Respecto al apartado sonoro del prototipo, aunque se utilizó **GarageBand** y **Adobe Audition** para la elaboración de algunos efectos de sonido, como los que suenan tras activar un mecanismo, tras coger un pétalo o la música de fondo del menú mostrado al finalizar el nivel, la mayoría de los sonidos presentes en el prototipo provienen de bibliotecas de efectos de sonido o, en el caso de la música de fondo, de bibliotecas musicales.

Esto se realizó así dado que, como se trató con anterioridad en esta memoria, el sonido resulta fundamental a la hora de proporcionar inmersión en el videojuego. Es preferible el uso de sonidos de archivo antes que la ausencia de los mismos, aunque la grabación de sonidos propios es un paso necesario en etapas futuras del proyecto.

### 5.3.6 OTROS ELEMENTOS UTILIZADOS

Otros elementos importantes desarrollados mediante Unreal Engine 4 que son necesarios listar son los volúmenes, los BSPs, las secuencias y las interfaces.

En el apartado de scripting se ha tratado ya el funcionamiento de un volumen de colisión, que es aquel que desencadena una respuesta cuando otros actores del nivel entran en este volumen. Sin embargo, en el prototipo se han utilizado otros objetos de esta categoría, como **volúmenes de post procesado** o el volumen de muerte. Así, el volumen de post procesado permite variar el aspecto visual de una escena al superponer a la imagen capturada por la cámara de juego diversos efectos o parámetros propios de una cámara cinematográfica. Por otro lado, el volumen de muerte, como su nombre indica, destruye a los actores que entran en su espacio.



Figura 34. Volumen de post procesado que varía el aspecto visual de la cámara bajo el agua

Otra herramienta importante fueron los BSP (**Binary Space Partitioning**). Se tratan de modelos tridimensionales que pueden ser editados dentro del propio engine. Podríamos considerarlos como pinceles poligonales, ya que sirven para moldear formas geométricas con un nivel limitado de complejidad. Esta capacidad de edición repercute negativamente en el rendimiento y los recursos que consumen. Así, su uso es importante en la etapa de prototipado, ya que permiten modelar la distribución general de niveles y elementos del mapeado de manera rápida antes de elaborar los modelados detallados del terreno, permitiendo así testear como se ve afectada la jugabilidad antes de destinar recursos a estas tareas.

En el prototipo, todos los terrenos están modelados mediante **BSPs**, por lo que tras comprobar mediante etapas de testeo que la distribución es buena y que resulta jugablemente atractiva, sería necesario modelar mallas poligonales para sustituir el terreno actual.

Por otra parte, también fueron citadas anteriormente las **secuencias**. Mediante las mismas, se pueden definir secuencias de animación de los disantos elementos que componen un nivel. Así, haciendo uso de una *línea temporal* y *keyframes*, se puede modificar el valor de propiedades de los mismos. En este prototipo, fueron usadas varias secuencias para lograr que diversas plataformas tuvieran movimiento.

Finalmente, y aunque no se profundizó demasiado en su uso, se elaboró una **interfaz** que se mostraría cuando el jugador terminara el nivel. En ella, se acceden a las variables del blueprint *PlatformGameMode* y se muestran el número de pétalos y frutas que ha recolectado el jugador, así como el tiempo que ha tardado en completar el nivel.

### 5.3 RESULTADO FINAL

El resultado final de este desarrollo se puede observar en el vídeo alojado en la siguiente URL:

<https://youtu.be/TyiRYXMqzww>

PARTE 3

## CONCLUSIONES

## 6. CONCLUSIONES

### 6.1. LÍNEAS FUTURAS

Como ya se ha comentado a lo largo de la última parte de la memoria, hay numerosos puntos de actuación, tanto en el desarrollo de videojuego como en el apartado comercial.

El primero de ellos y el más inmediato es **la organización de sesiones de testeo** para medir las reacciones de jugadores ante el prototipo. Como se ha comentado en las secciones dedicadas a las metodologías lean, el *feedback* generado por estas sesiones resulta crítico para seguir avanzando con el proyecto.

Sobre el prototipo, prácticamente se podría decir que lo desarrollado en este proyecto no llega a ser siquiera el **punto de partida** de un videojuego comercial. Numerosos aspectos deberían ser mejorados y tratados para pretender medir su potencial. Se han mencionado algunos de ellos, como el modelado y animación del personaje principal, la elaboración de los modelos del terreno, la inclusión de música y sonidos personalizada, etc. Todo esto supone una tarea enorme, que difícilmente puede ser llevada a cabo por una sola persona, por lo que uno de los siguientes pasos sería la **búsqueda de socios y colaboradores**.

Finalmente, atendiendo al hipotético carácter comercial de la propuesta, se deberían tener en consideración aspectos tratados en el **Plan de Negocio** (Anexo 4). En este documento se establecen algunas pautas y fases para seguir adelante con el proyecto tras el establecimiento de un equipo de desarrollo.

## 6.2. CONCLUSIONES FINALES

En este proyecto se ha pretendido dar una visión global de lo que supone hoy en día el desarrollo de un videojuego. En sus orígenes, estos eran desarrollados solamente por informáticos y programadores y, en muchos casos, esta visión aún se mantiene en la actualidad. Ahora, desarrollar un videojuego es una tarea compleja y multidisciplinar, en la que además de los numerosos aspectos creativos, también entra en juego la gestión comercial.

A la hora de la verdad, cada estudio de desarrollo trabaja según unos procesos y metodologías que les permiten funcionar de una manera organizada. Sería ingenuo afirmar con rotundidad que las herramientas lean aquí planteadas deberían ser implantadas en todos ellos, dado que no se puede comparar este caso de estudio a una actividad comercial real. Si bien es cierto que estas herramientas generan multitud de beneficios, comenzar a utilizarlas en empresas con una metodología ya establecida requiere de un esfuerzo y unos costes asociados al mismo. De la misma manera, a la hora de lanzar un nuevo negocio, también se debe invertir tiempo en estudiar los principios de *lean manufacturing* y *lean startup* para su correcta aplicación. Además, en cada equipo de trabajo y entorno comercial surgen particularidades que deben ser tratadas.

Todo ello supone un obstáculo al comenzar a implantar esta filosofía en una empresa. Sin embargo, en el panorama comercial actual, en el que minimizar riesgos, ser rápido y eficiente constituyen factores críticos a la hora de lanzar un negocio, metodologías como las propuestas en este proyecto pueden suponer la diferencia entre el éxito y el fracaso o, si se tiene en consideración que el 80% de las startups creadas no sobreviven al tercer año<sup>31</sup>, entre un fracaso controlado o un fracaso desolador.

Estos pequeños fracasos generan el éxito de futuros proyectos, por lo que las filosofías *lean* son de gran interés para cualquier persona con mentalidad emprendedora y con proyectos en cualquier sector, incluido el de los videojuegos.

---

31 ¿Por qué motivos fracasan las startups? – Josep Arroyo

<http://blogs.lavanguardia.com/emprendedor-en-serie/2016/09/25/por-que-motivos-fracasan-las-startups-50290/>

### 6.3 BIBLIOGRAFÍA

- [1] The Art of Game Design: A Book of Lenses  
CRC Press. ISBN: 978-1466598645
- [2] Game Feel: A Game Designer's Guide to Virtual Sensation .  
Steve Swink, CRC Press. ISBN: 978-0123743282
- [3] Level Up! The Guide to Great Video Game Design. Scott Rogers, John Wiley & Sons Inc. ISBN:  
978-1118877166
- [4] Theory of Fun for Game Design. Raph Koster, O'Reilly Media. ISBN: 978-1449363215
- [5] AI for Game Developers. David M. Bourg - Glenn Seemann, O'Reilly Media. ISBN: 978-0596005559
- [6] Reality Is Broken. Jane McGonigal, Random House UK. ISBN: 978-0099540281
- [7] Game Programming Patterns, Lightning Source Inc. ISBN: 978-0990582908
- [8] Homo ludens. Johan Huizinga. Alianza editorial. ISBN: 9788420608532
- [9] LEAN MANUFACTURING. EXCELENCIA EN LAS OPERACIONES. La mejora continua.  
LeanSis, Personas Procesos Productividad  
<http://www.leansisproductividad.com/wp-content/uploads/LeanSis-Productividad-La-Mejora-Continua.pdf>
- [10] Beyond Scrum: Lean and Kanban for Game Developers  
Clinton Keith, Gamasutra.  
[http://www.gamasutra.com/view/feature/132241/beyond\\_scrum\\_lean\\_and\\_kanban\\_for\\_.php?page=1](http://www.gamasutra.com/view/feature/132241/beyond_scrum_lean_and_kanban_for_.php?page=1)
- [11] Unreal Engine Documentation  
<https://docs.unrealengine.com/latest/INT/>
- [12] Playing with your mind: The psychology of sound in video games  
<http://www.amplifon.ie/resources/playing-with-your-mind/>
- [13] Will smell ever come to smartphones?  
<https://www.newyorker.com/tech/elements/is-digital-smell-doomed>

[14] Metodología Lean Manufacturing: Qué es y cómo implementarla en tu empresa

<https://leanmanufacturing10.com/>

[15] Defining Lean Manufacturing

<http://leanmanufacturingtools.org/34/lean-manufacturing-definition-2/>

[16] El método Lean Startup

Eric Ries. DEUSTO S.A. EDICIONES. ISBN: 9788423409495

[17] Diseñando la propuesta de valor: Como crear los productos y servicios que tus clientes están esperando.

Alexander Osterwalder. DEUSTO S.A. EDICIONES. ISBN: 9788423419517

[18] Generación de Modelos de negocio

Alexander Osterwalder. DEUSTO S.A. EDICIONES. ISBN: 9788423427994

[19] Flow: The Psychology of Optimal Experience

Mihály Csíkszentmihályi. ISBN 978-0061339202

[20] Man, Play and Games. Roger Caillois. ISBN-13: 978-0252070334

[21] El método Lean Startup: Cómo crear empresas de éxito utilizando la innovación continua.

Eric Ries. ISBN: 9788423409495

[22] Learn To Code by Making Games - The Unreal Engine Developer Course. Ben Tristem

<https://www.udemy.com/unrealcourse/learn/v4/overview>

[23] Itinerario Yuzz. UniMOOC

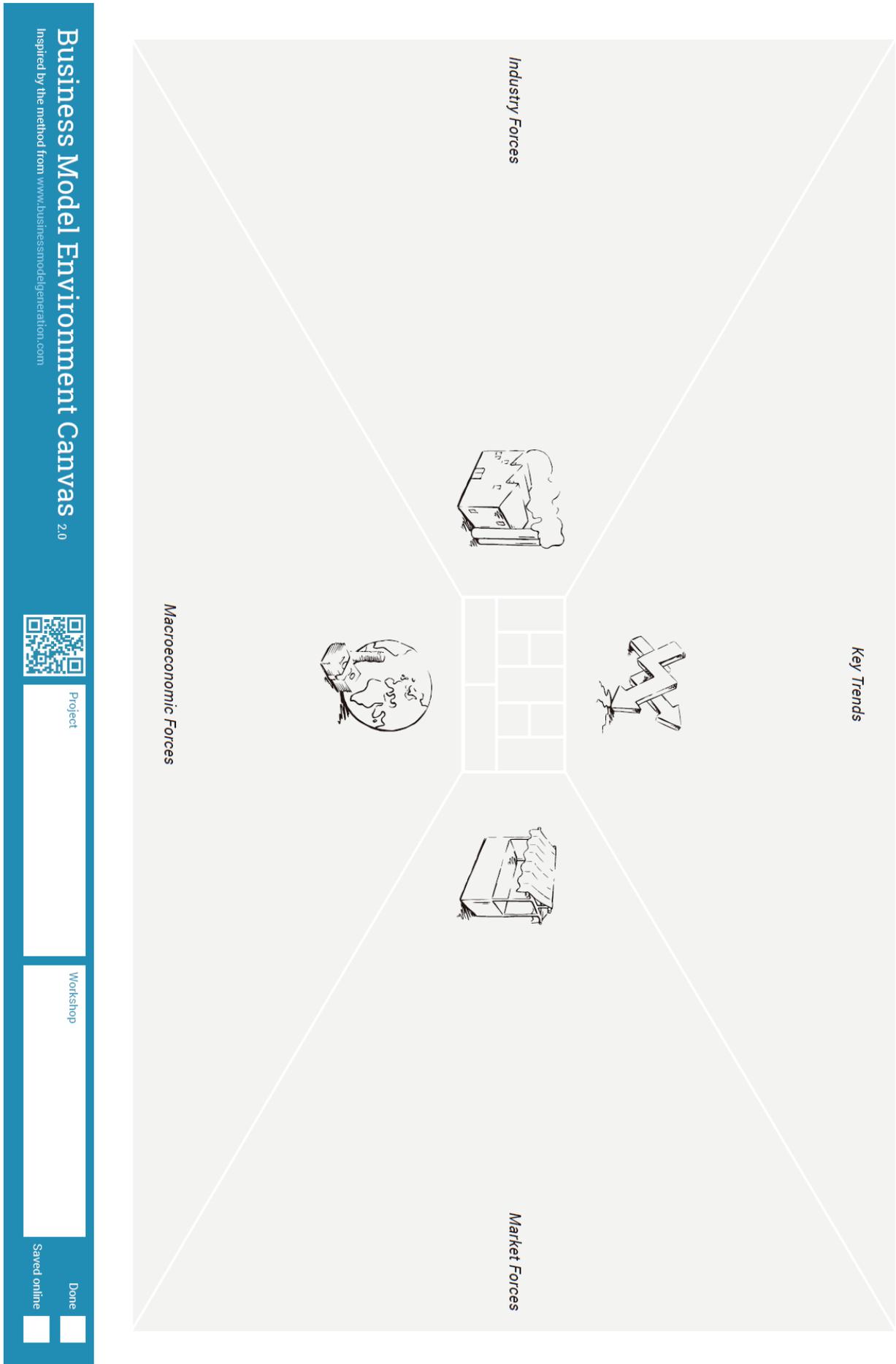
<https://learn.unimooc.com/student/courses/course?course=yuzz>

PARTE 4

## ANEXOS

## Anexo 1.A. Business Model Environment Canvas.

(Extraído de Knowledge.li y basado en Generación de Modelos de Negocio, por Alexander Osterwalder y Yves Pigneur)



# Business Model Canvas

Diseñado para:

Diseñado por:

Et: 1º, 2º, 3º, 4º

Iteración: 1ª

## 8 Socios clave



¿quienes son nuestros socios clave?  
¿quienes son nuestros suministradores clave?  
¿que recursos clave vamos a adquirir de nuestros socios?  
¿que actividades clave realizan los socios?  
¿cual es el valor que nos aportan?  
¿que actividades clave realizan los socios?  
¿que actividades clave realizan los socios?  
¿que actividades clave realizan los socios?

## 7 Actividades clave



¿que actividades clave requiere nuestro modelo de valor?  
¿que recursos claves de distribución?  
¿que recursos claves de canales?  
¿que recursos claves de actividades?  
¿que recursos claves de actividades?  
¿que recursos claves de actividades?  
¿que recursos claves de actividades?

## 2 Propuestas de valor



¿que valor estamos creando?  
¿cual de los problemas de nuestro cliente vamos a resolver?  
¿que beneficios de productos y servicios ofrecemos a cada segmento de cliente?  
¿que beneficios de productos y servicios ofrecemos a cada segmento de cliente?  
¿que beneficios de productos y servicios ofrecemos a cada segmento de cliente?  
¿que beneficios de productos y servicios ofrecemos a cada segmento de cliente?  
¿que beneficios de productos y servicios ofrecemos a cada segmento de cliente?

## 4 Relaciones con clientes



¿que tipo de relación espera que establezcamos y mantengamos cada uno de nuestros segmentos de cliente?  
¿que tipo de relación espera que establezcamos y mantengamos cada uno de nuestros segmentos de cliente?  
¿que tipo de relación espera que establezcamos y mantengamos cada uno de nuestros segmentos de cliente?  
¿que tipo de relación espera que establezcamos y mantengamos cada uno de nuestros segmentos de cliente?  
¿que tipo de relación espera que establezcamos y mantengamos cada uno de nuestros segmentos de cliente?  
¿que tipo de relación espera que establezcamos y mantengamos cada uno de nuestros segmentos de cliente?

## 1 Segmentos de cliente



¿que clientes estamos creando valor?  
¿quienes son nuestros clientes más importantes?  
¿quienes son nuestros clientes más importantes?

## 6 Recursos clave



¿que recursos clave requiere nuestro modelo de valor?  
¿que recursos claves de distribución?  
¿que recursos claves de canales?  
¿que recursos claves de actividades?  
¿que recursos claves de actividades?  
¿que recursos claves de actividades?  
¿que recursos claves de actividades?

## 3 Canales



¿a través de que canales quieren ser contactados nuestros segmentos de cliente?  
¿cómo se contactamos ahora?  
¿cómo se contactamos ahora?

## 9 Estructura de costes



¿quienes son los costes más importantes relacionados a nuestro modelo de negocio?  
¿quienes son los costes más importantes relacionados a nuestro modelo de negocio?  
¿quienes son los costes más importantes relacionados a nuestro modelo de negocio?  
¿quienes son los costes más importantes relacionados a nuestro modelo de negocio?  
¿quienes son los costes más importantes relacionados a nuestro modelo de negocio?  
¿quienes son los costes más importantes relacionados a nuestro modelo de negocio?

## 5 Fuentes de ingresos



¿que fuentes de ingresos estamos generando?  
¿que fuentes de ingresos estamos generando?

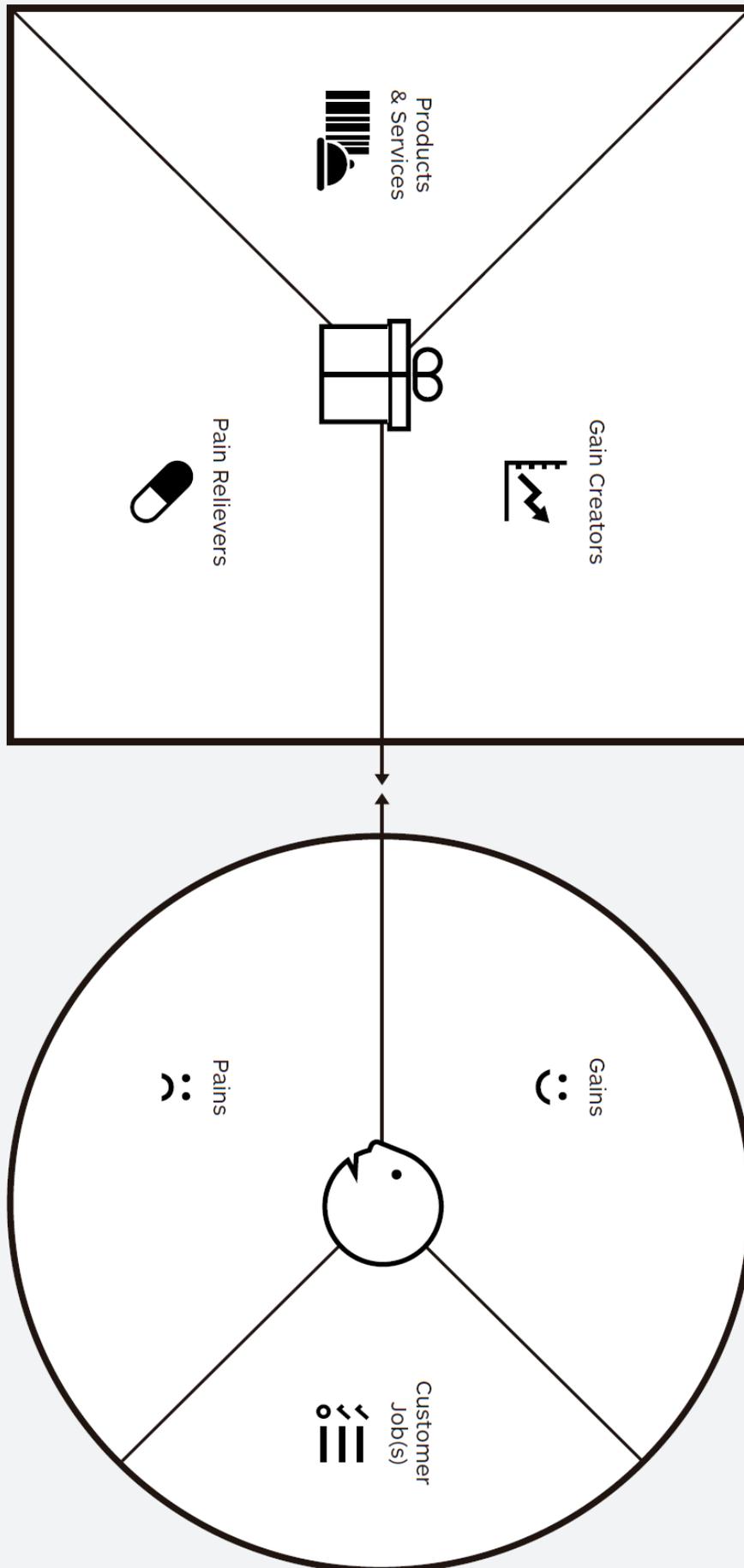
## 9 Estructura de costes



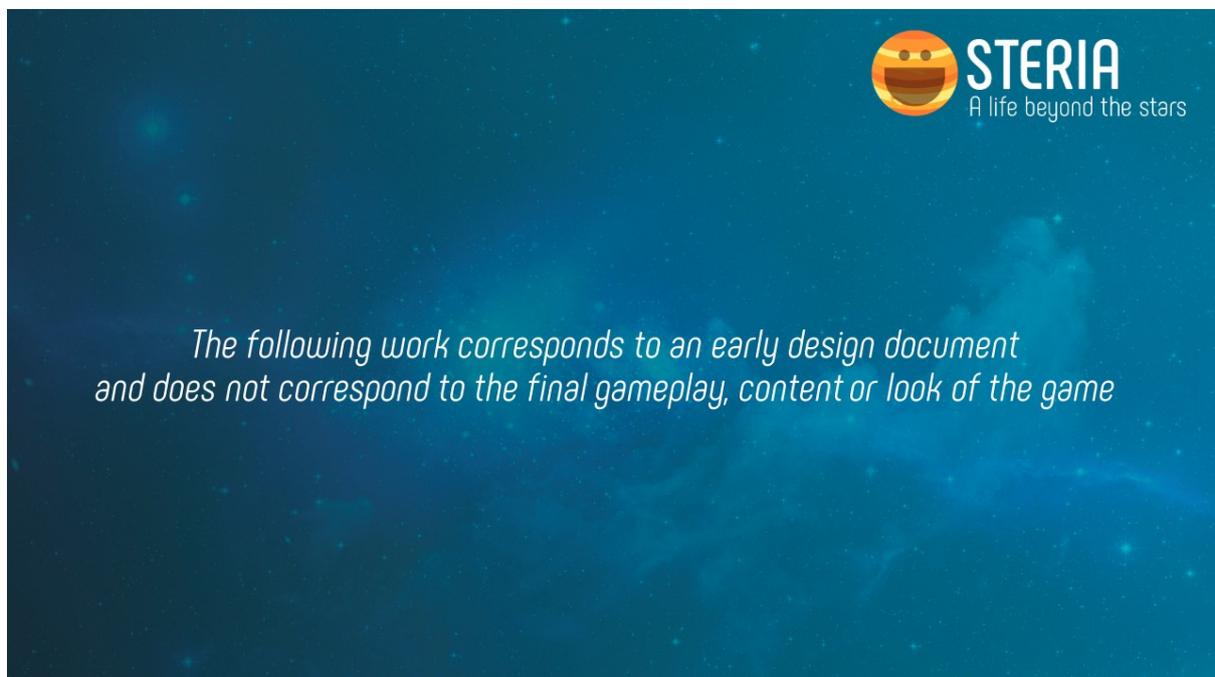
¿quienes son los costes más importantes relacionados a nuestro modelo de negocio?  
¿quienes son los costes más importantes relacionados a nuestro modelo de negocio?  
¿quienes son los costes más importantes relacionados a nuestro modelo de negocio?  
¿quienes son los costes más importantes relacionados a nuestro modelo de negocio?  
¿quienes son los costes más importantes relacionados a nuestro modelo de negocio?  
¿quienes son los costes más importantes relacionados a nuestro modelo de negocio?

*Anexo 1.C. Value Proposition Canvas.*

*(Extraído de Strategyzer.com y Generación de Modelos de Negocio, por Alexander Osterwalder y Yves Pigneur).*



*Anexo 2. Ejemplo de GDD*



## STERIA A LIFE BEYOND THE STARS



**Story overview** At an alternate future, the known civilizations are at the border of collapse. In such chaos, a new program aims to establish new colonies in a far away galaxy. You are part of those who adventure into the unknown, in search for a new future. You must learn to tame the wilderness, building a place you can call "home".

**Game overview** *Steria A life beyond the stars* is a life simulator that puts the player in an distant planet filled with surprises. He must help to establish a community of diverse alien individuals and develop relationships with them. Exploring, farming, crafting, building, recollecting, customising his house... Discovery awaits at every corner!

## THE CHARACTERS



**Player** The main character is the only human present at the expedition. He/She is customisable. At the start of the game players can choose his name, sex and appearance (hairstyle, face, skin, eye and hair colour...). Some of that options can be changed later. In addition, the player will find (or create) a wide array of clothing he can equip, being able to play express his/her creativity.

**Settlers** In *Steria* the player is placed in a space colony that is populated with other settlers. They are his neighbours and each one of them has a unique personality and background. The players colony will have a random selection from a bigger list of possible neighbours, as they all can not live in the same colony. Neighbours come and go, depending on their relationships with the player.

## THE CHARACTERS

**Visitors and special characters** In addition, there are certain characters that are present in every colony. Those characters have special functions, as shopkeepers, scientists, the mayor...

And last, the colony can be visited by special guests at random or fixed events. These characters are rare and usually provide rewards to the players.

**Creatures** There are, of course, native creatures of the planet the player can interact with. Some of them can be tamed and can be taken care of to obtain certain resources or bonuses. Others can be hunted via fishing or trapped to obtain food and to be studied.

But some creatures won't be as easy to deal with and can be dangerous...

## GAMEPLAY

**The world** The game is placed in an extraterrestrial planet which hides a lot of secrets. The player must learn how this new world works: which resources he can make use, how to cultivate and grow native plants and creatures... He will play in a world that feels strange and alive, enjoying a sense of Discovery.

The main action takes place in the colony, where all the settlers live. Each one of them has a house that the player can visit. There are buildings as shops, laboratories, the command center...

The colony is placed in a prosperous environment, near forests, crossed by a river that reaches a nearby coast.

Additionally, the player can be part of expeditions, exploring different biomes (alien forests, deserts...) filled with new resources and creatures not present in the colony.

## GAMEPLAY *cont.*

**The house** The player is given a small house, a private place he can customise placing new furniture and upgrading to have more space and functionalities. In his house the player can place his storages, to keep all the items and resources he has gathered.

There are a huge variety of furniture, some of them being part of collections that share the same theme and style. The player can buy new furniture or even craft it. There are some furniture wich has special functions, as allowing the player to craft certain items, to cook or to sleep and regain energy.

## EXPERIENCE

**Time** The colony's time passes at the same rate than the real. An ingame hour correspond to its real life equivalent, meaning that if the player plays the game at night, the world time would be place at night too. The game calendar is based on the real one too; Seasons change and special events occur during certain days of the week or at special dates.

**Basic needs** The player characters does have some basic needs as hunger, rest, higiene or social interactions that need to be fullfilled. Doing so the player wins certain bonuses for example, if the character eats, he gain a strength bonus, making easier to chop down tres or mining ores. But doing these actions deplete player's energy, that can be recovered by drinking, sitting or sleeping.

**A relaxing experience** The ultimate aim of the game is to provide a virtual place where the player can play an alternative life, finding new adventures, with non fixed objetives nor missions, at a calm speed and satisfiying the player's creativity and curiosity.

*Steria* is a world the player can visit in short sessions everyday.

## OTHER MECHANICS AND MODES

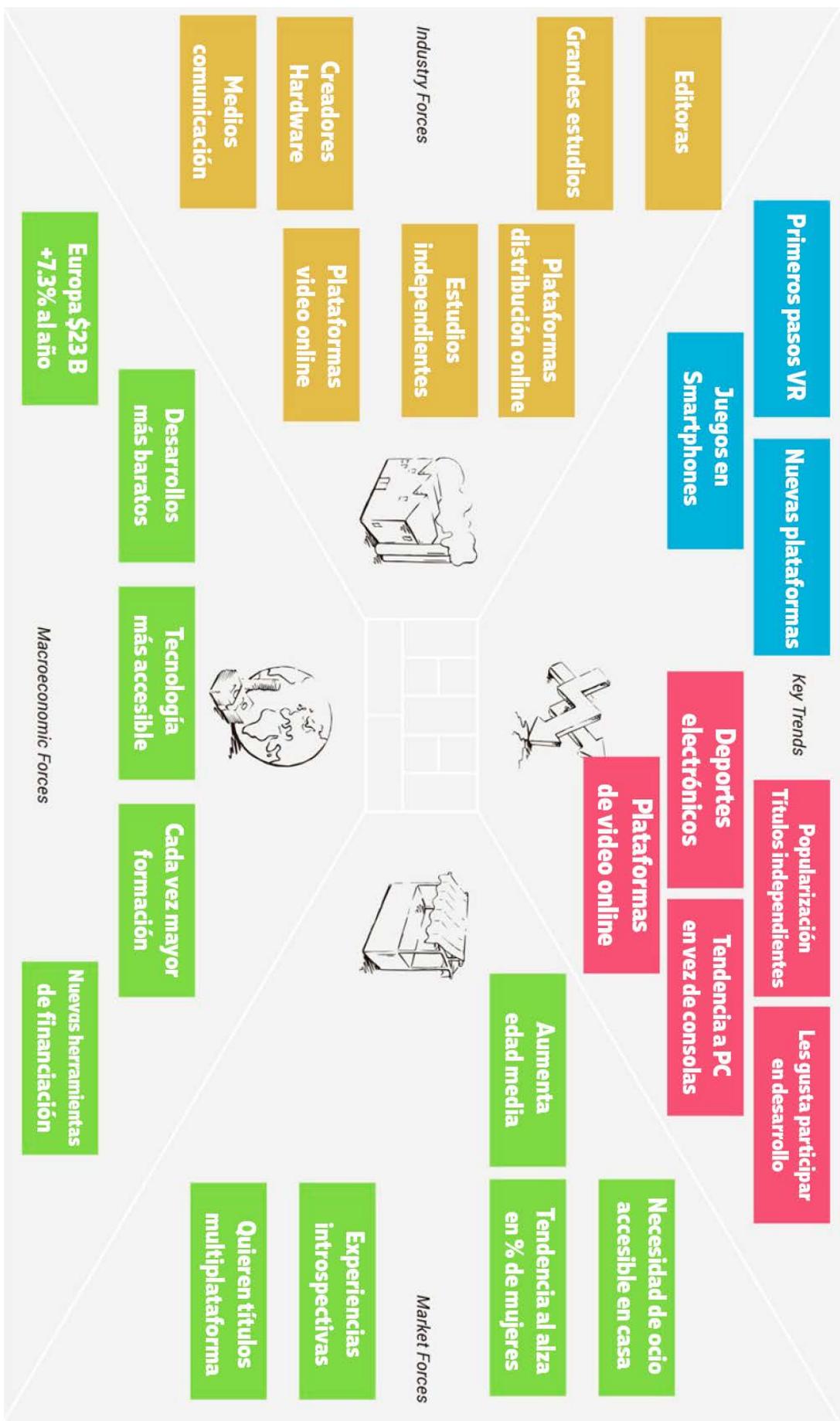
**Recollecting** The world is filled with resources the player can collect. He can mine ores in order to obtain minerals, chop down trees to get wood, plant vegetables to grow them, dig to find hidden treasures or artifacts or go fishing to catch strange water creatures. Completing collections are an important part of the gameplay.

**Crafting** But being able to gather resources means nothing if you can not make something with them. The players can use these to create new objects, clothing, furniture, food and other special items.

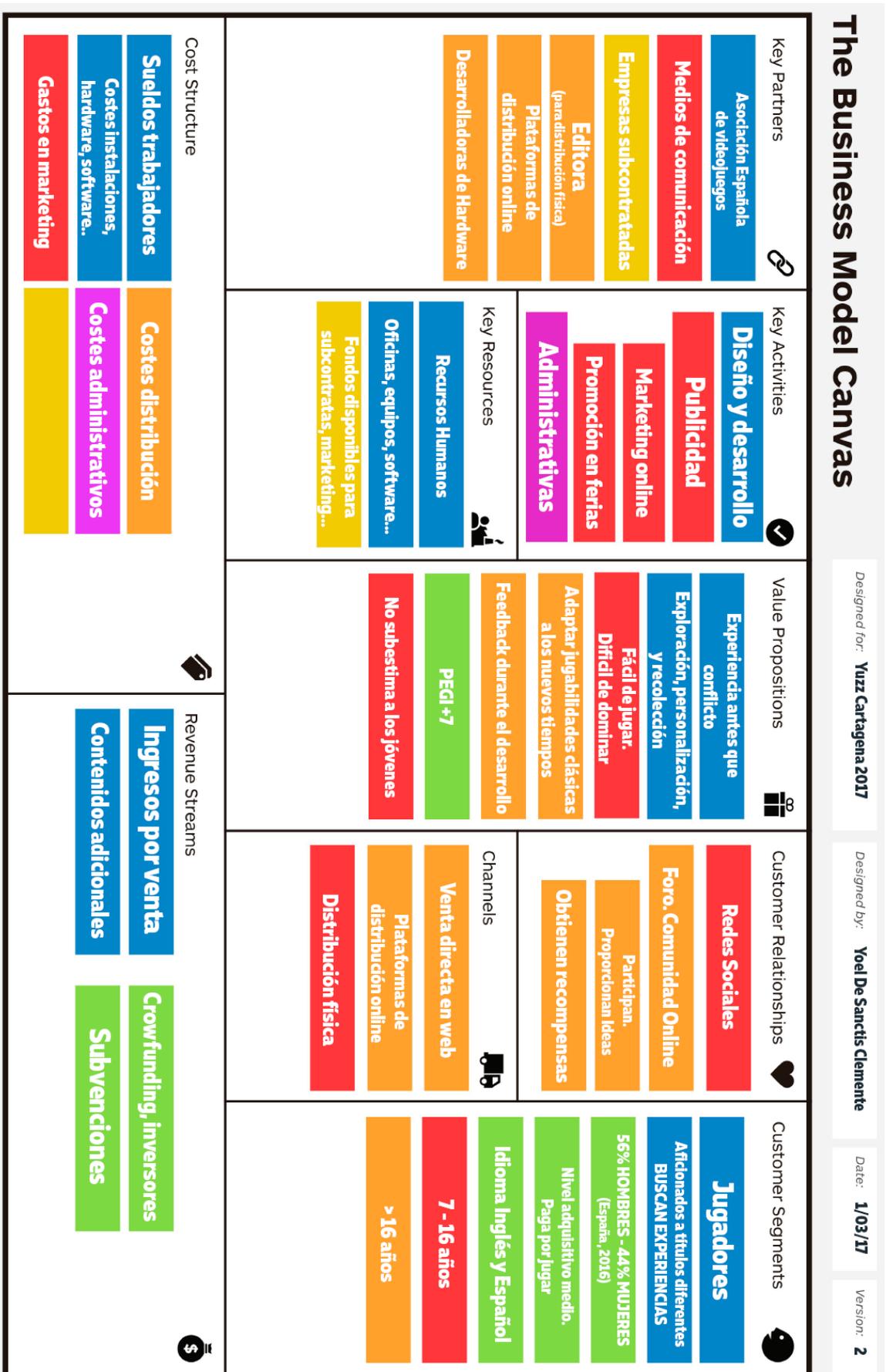
**Economy** Selling the collected or produced goods is necessary to earn credits, the coin the player can use to buy new items, upgrade his equipment or to being able to access to new content.

**Multiplayer** As an ambitious objective, a multiplayer mode would allow the player to visit friend's colonies, to trade items, to play expeditions together or to share experiences.

Anexo 3.1. Business Environment Model Map del proyecto



## Anexo 3.2. Business Model Canvas del proyecto



### Anexo 3.3. Hipótesis de la propuesta de valor

Supuestos de Negocio	
1. Creo que mi cliente es ...	Jugador, sin grupo demográfico predefinido. Juega en PC o consola y le gustan propuestas de muchos tipos y géneros. Aprecia los títulos independientes.
2. Creo que los distintos tipos de cliente que tengo son ...	Dependiendo de la edad, podría dividirlos en clientes jóvenes y adultos, con diferentes motivaciones que le hagan llegar a mi producto.
3. Creo que lo principal que tiene que resolver mi cliente es ...	Ocio en casa. Poder disfrutar de experiencias que le transmitan emociones y que le hagan disfrutar mientras juega.
4. Creo que mi cliente también tiene una necesidad/un problema en ...	Videojuegos que le producen estrés en vez de relajarle.
5. Creo que mi cliente expresa su necesidad/problema de forma ... (latente/pasiva/activa/visionaria)	De forma pasiva. Juega a lo que conoce, a lo que más publicidad recibe. Termina dedicando su tiempo a otras formas de entretenimiento más pasivas, como el cine o la lectura.
6. Creo que actualmente mi cliente soluciona/satisface su necesidad mediante ...	La prensa digital, en busca de nuevos juegos que le puedan satisfacer. Ve vídeos <u>gameplay</u> en plataformas online como <u>twitch</u> o <u>youtube</u> .
7. Creo que la importancia que le da mi cliente a la resolución del problema es ... (alta, baja...)	Le da una importancia media. Aunque cada vez hay mayor reconocimiento, todavía trata a los videojuegos como un medio de segunda categoría frente al cine o la televisión.
8. Creo que para mi cliente es especialmente doloroso ...	La falta de tiempo para jugar un juego. Aprecia la posibilidad de partidas cortas.
9. Creo que conozco a mi cliente ... (nada, poco, mucho...)	Bastante. Creo entender las razones por la que las personas se ven atraídas a los videojuegos y por qué algunas dejan de jugar con los años.
10. Creo que me falta por conocer en mi cliente ...	Su punto de vista respecto a videojuegos más tranquilos, en los que prime la experiencia, la exploración del entorno y la narrativa. ¿Le resultaría aburrido?
11. Creo que el tamaño de mi mercado es ...	Ciertamente no es un mercado de masas. Adicionalmente, en el sector de los videojuegos independientes el éxito o el fracaso de un título es bastante impredecible.
12. Creo que el decisor de compra es ...	Que lo que vea en pantalla le guste, que le llame la atención. El cliente se vería atraído a la propuesta y querrá disfrutar de la experiencia, tras lo que comenzará a comentarla en sus círculos sociales (personales u online) generando nuevos clientes interesados.

Supuestos de Negocio	
13. Creo que quien influencia a mi cliente es ...	Prensa online, análisis, otros clientes y generadores de contenido audiovisual online.
14. Creo que quien puede "sabotear" mi propuesta es ...	Publicadores de malas críticas.
15. Creo que quien prescribe a mi cliente es ...	Comunidad de jugadores. Se tiende a seguir la corriente.
16. Otros aspectos sobre mi cliente que no conozco (jobs to be done, pains, gains ...)	Si considera los videojuegos únicamente como una manera de matar el tiempo o, por el contrario, cree que pueden ser algo más. Incluso dentro de la industria hay críticos a la consideración de los videojuegos como arte, infravalorando el medio.

Supuestos de Usuario	
15. Creo que el que utilizará mi servicio será ... (¿quién es nuestro usuario?)	El cliente, el jugador.
16. Creo que mi usuario utilizará mi producto/servicio ... (ciclo de vida, trabajo/casa, momento del día/mes ... )	En casa, por las tardes/noches, después del trabajo y durante los fines de semana.
17. Creo que cuando usa mi producto/servicio se resuelve ...	La evasión de la vida diaria. La búsqueda de entretenimiento.
18. Creo que para mi usuario las características del producto/servicio más relevantes son ...	La accesibilidad y el hecho de que el producto le pueda llegar a implicar emocionalmente.
19. Creo que conozco a mi usuario ... (nada, poco, mucho...)	Ni mucho ni poco. Aunque los juegos producen una reacciones ampliamente estudiadas en los seres humanos, cada persona es un mundo.

<b>Supuestos de Negocio</b>	
1. Creo que la principal necesidad o problema de mi cliente es ...	Ocio en casa. Poder disfrutar de experiencias que le transmitan emociones y que le hagan disfrutar mientras juega.
2. Creo que además mi cliente tiene estas otras necesidades/problemas ... (ordenar por importancia)	Búsqueda de juegos que le relajen y le hagan pensar. Que sean accesibles.
3. Creo que mi cliente resuelve actualmente este problema/necesidad ...	A veces opta por otros medios, como la literatura o el cine.
4. Creo que otras barreras pueden ser ...	Que se tratará de un título humilde con poco impacto mediático en principio. Que el impacto en el jugador no sean el esperado.
5. Creo que la mayor satisfacción de mi cliente al resolver ese problema/necesidad es ...	Que cuando llegue a casa tenga ganas de jugar a un determinado título. Que al completarlo se quede con buen sabor de boca.
6. Creo que otras satisfacciones pueden ser ...	Que mientras juegue se evada de sus problemas y sirva para relajarse.
7. Creo que los aspectos menos valorados por mis clientes de los productos/servicios sustitutivos son ...	Duración escasa. Poca rejugabilidad.
8. Creo que las 10 funcionalidades principales de mi producto/servicio son ... (ordenadas según su valor para el cliente)	Experiencia relajante. Placer visual. Originalidad. Rejugabilidad. Narrativa. Exploración y coleccionables. Nostalgia. Personalización. Accesibilidad. Participación en desarrollo.
9. Creo que puedo eliminar o reducir las barreras (pains) del cliente ...	Siendo transparente y sincero durante el desarrollo.
10. Creo que puedo aumentar la satisfacción (gain) de mi cliente ...	Si a pesar de conocerlo el producto le sorprende mientras juega.

Creemos que, ofreciendo una apuesta diferente y relajada, atractiva visualmente y con ambición narrativa a los jugadores conseguiremos repercusión mediática, buenas opiniones del producto y aseguraremos la rentabilidad del desarrollo. Sabremos que es verdad cuando los posibles clientes confirmen las presunciones formuladas en este documento y puedan llegar a mostrar interés en la propuesta

### Anexo 3.4. Cuestionario para la elaboración de entrevistas

## ENTREVISTA DE PROBLEMA

Nº	Fecha	Nombre
----	-------	--------

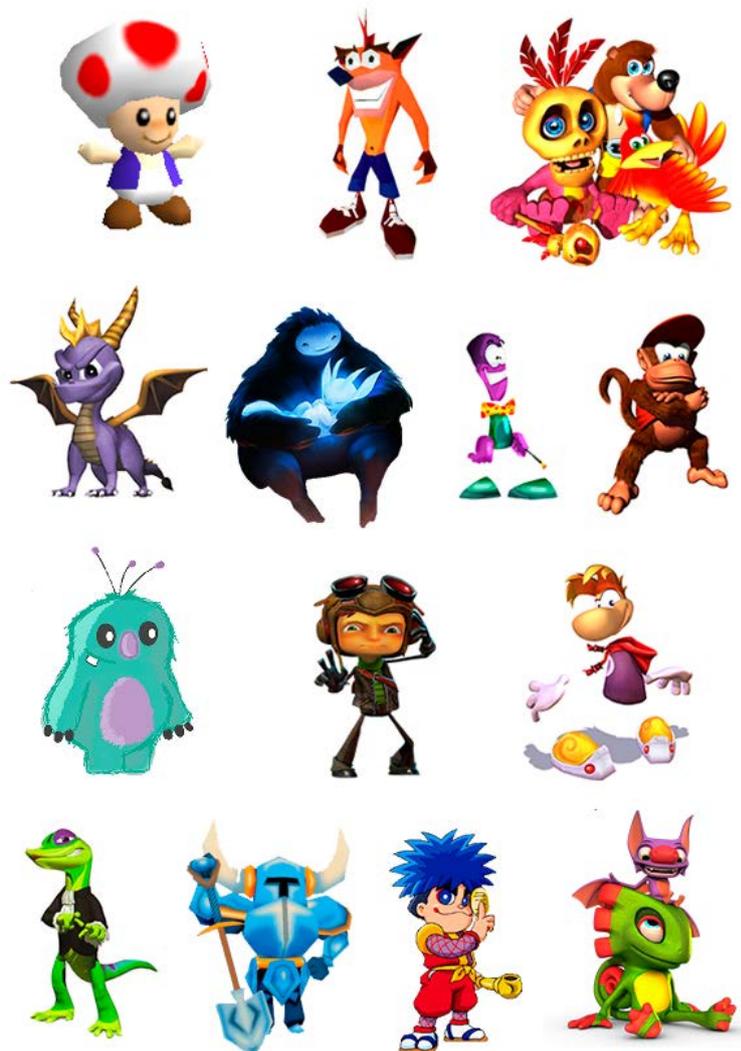
1 ¿Cuál es su edad? ¿Cuál es su sexo?

2 ¿Cuáles son sus hábitos respecto a los videojuegos? Cuantas veces juega a la semana, qué plataformas utiliza para jugar, qué títulos está jugando en este momento, cuánto gasta en videojuegos, se encuentra informado de las últimas noticias...

3 Géneros favoritos. ¿Por qué? ¿Qué aprecia en un videojuego? (Exploración, historia, coleccionables, aspecto visual...) ¿Le gustan los juegos de plataformas?

4 Videojuegos de los que tenga un buen recuerdo o que le hayan marcado de alguna manera. ¿Por qué?

5 ¿Reconoce a estos personajes? ¿Cuáles le gustan? ¿Qué le transmiten?



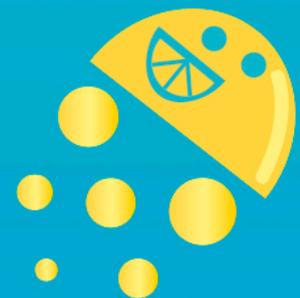
**6** Razones por las que juega. ¿Juega menos de lo que le gustaría? ¿Por qué?

**7** ¿Tiene dificultades para encontrar videojuegos que le llamen la atención? ¿Qué debe tener un videojuego para que le guste? ¿Echa de menos algunos aspectos de los juegos clásicos? ¿Por qué?

**8** ¿Alguna vez se ha puesto nervioso o sentido estresado jugando a un videojuego? ¿Por qué?

# Plan de empresa

**Lemon**  
Jellyfish



**Yoel De Sanctis Clemente**

Yuzz Cartagena 2017

# Índice

Identificación del proyecto .....	3
La idea y la propuesta de valor.....	3
El modelo de negocio .....	4
El equipo promotor .....	4
Proceso de validación.....	5
Plan de Marketing .....	6
Delimitación del producto o servicio y del mercado .....	6
Análisis del entorno en el que se desarrolla el proyecto .....	7
Análisis DAFO .....	8
Información interna del proyecto.....	9
Objetivos del Plan de Marketing.....	10
Estrategia .....	10
Estrategia corporativa .....	10
Estrategias de marketing.....	10
Plan de acción de marketing .....	11
Producto.....	11
Precio.....	11
Distribución .....	12
Promoción y publicidad .....	12
Personal, procesos y entorno físico.....	13
Plan de operaciones.....	13
Productos.....	13
Procesos.....	14
Plan de recursos humanos.....	15
Estructura legal de la empresa .....	15
Plan de Inversiones y ubicación.....	16
Plan de inversión inicial.....	17
Plan de financiación.....	18
Previsión de ventas.....	18
Gastos de explotación .....	19
Cuenta de pérdidas y ganancias provisional .....	20
Calendario de ejecución y conclusiones .....	21
Conclusiones.....	21

## Identificación del proyecto

Lemon Jellyfish es un estudio independiente de desarrollo de videojuegos afincado en Cartagena. En un mercado en auge como es el sector de los videojuegos, un público cada vez más heterogéneo, sin apenas barreras de sexo u edad, dedica más tiempo que nunca al ocio digital. Y, sin embargo, los videojuegos no han alcanzado aún todo su potencial, con aplicaciones claras en la educación, la salud, la productividad personal y en la empresa, aparte del habitual ocio, que no están desarrolladas.

En Lemon Jellyfish creemos que el videojuego no es tan sólo el presente, si no un futuro lleno de posibilidades, un futuro del que queremos formar parte.

### La idea y la propuesta de valor

El primer proyecto de Lemon Jellyfish es *Mapu and the Lost light*, un título plataformas en 3D desarrollado en Unreal Engine para PC (con la intención de portearlo a consolas domésticas más adelante).

*Mapu and the Lost Light* comenzó siendo un simulador de vida en una colonia espacial denominado *Asterial*, con una estética *cartoon* para todos los públicos, en el que se iban a tratar temas como la concienciación medioambiental, la importancia de las relaciones sociales y la vida en comunidad, así como ciertos aspectos de divulgación científica. Todo ello sería tratado en un título en el que el jugador se vería envuelto en un mundo virtual en el que podría jugar a su ritmo, relacionarse con potentes inteligencias artificiales que simularían a otros colonos extraterrestres, descubrir y dar uso responsable a los recursos presentes en el lejano planeta, así como a su flora y fauna y, en definitiva, experimentar una vida virtual en un mundo rico e inmersivo.

Dada la complejidad de la propuesta y el alto tiempo que llevaría el realizar un prototipo jugable que representara fielmente las mecánicas del título, así como la dificultad que se encontraría en la fase de testeo (ya que el tiempo del juego pasaba a la misma velocidad que el tiempo real), se decidió apostar por un título con una jugabilidad más directa y en el que los temas serían tratados a través de la narrativa.

En *Mapu and the Lost Light* el jugador se pone en la piel de Mapu, un pequeño espíritu de la naturaleza que, viviendo sólo en un mundo fracturado y en decadencia, ha perdido la ilusión por la vida y, con ella, los poderes que antaño iluminaban la tierra. Todo ello cambia cuando Lokko, una pequeña luciérnaga, se cruza en su camino y juntos emprenden una aventura para recuperar el brillo de Mapu y volver a iluminar el mundo.

Para ello, Mapu y Lokko contarán con habilidades relacionadas con la luz, mediante las cuales podrán utilizar la vegetación a su favor y hacerse camino entre misteriosos enemigos. Todo ello en un título en el que se fomentará la exploración, la aventura y la creatividad.

## El modelo de negocio

Lemon Jellyfish compete en el sector de los videojuegos, el cual actualmente se encuentra en crecimiento en España, habiendo cada vez mayor cantidad de programas de fomento, subvenciones e inversiones.

En este ecosistema, Lemon Jellyfish apuesta por el videojuego como medio de comunicación en el que no sólo importa el entretenimiento, sino la pasión inherente al ser humano por las historias profundas, con significado, razón de ser y que emocionen. Mientras que este aspecto se da por hecho en medios artísticos como el cine, en los videojuegos en la mayoría de ocasiones se sigue dejando de lado en favor de propuestas vacuas con las que cada vez el público es más crítico.

A medida que el público cada vez es más heterogéneo (en España el último año, de 15 millones de jugadores el 44% son mujeres y el 46% mayores de 25 años <sup>1</sup>), se demandan propuestas cada vez más diferentes. Es allí donde Lemon Jellyfish trabaja, centrando sus esfuerzos en la innovación narrativa, jugable y educativa.

Para ello uno de los pilares lo constituirá la participación colectiva, haciendo especial hincapié en desarrollos transparentes en los que el feedback del público sea muy importante, aportando ideas, sugerencias y nuevas propuestas que puedan ser tenidas en cuenta. Creemos que los videojuegos pueden constituir un gran impacto en la construcción de un mundo mejor. Es por ello que Lemon Jellyfish trabaja junto y para la sociedad.

Finalmente, los ingresos se obtendrán principalmente de la venta de títulos, centrándose en medios digitales sin dejar de lado la búsqueda de distribuidores físicos.

<sup>1</sup> Informe 2016 AEVI (Asociación Española de Videojuegos)

<http://www.aevi.org.es/aevi-publica-adelanto-los-datos-del-sector-del-2016-videojuego-facturo-espana-1-163-millones-euros/>

## El equipo promotor

Por el momento, Lemon Jellyfish es un proyecto de Yoel De Sanctis Clemente, Ingeniero Técnico en Telemática por la Universidad Politécnica de Cartagena. Complementando la faceta técnica, Yoel trabaja gracias a una beca de Iniciación a la Investigación en el Centro de Producción de Contenidos Digitales, realizando tareas de apoyo a proyectos audiovisuales de la UPCT.

Ha formado parte del Filmcamp con Pablo Berger y Curso de fotografía cinematográfica por la escuela de cine del mediterráneo La Base, cursos presenciales de

Marketing Digital por el programa Google Actívate, cursos de programación en Android, Python o Unity así como cursos de videomarketing o ilustración infantil.

Además, ha realizado labores de voluntariado, trabajando como Monitor de Tiempo Libre, dando charlas de concienciación a menores acerca del uso seguro de Internet o, especialmente, como miembro del equipo Coordinador de la Teleco LAN Party, evento anual celebrado en la UPCT en el que ha llevado a cabo la labor de coordinador de Imagen y Comunicación desde 2013.

Finalmente, ha desarrollado tareas de diseño gráfico y web de manera freelance.

## Proceso de validación

Se ha seguido la metodología Lean Startup en el desarrollo del proyecto. Actualmente se encuentra en fase de creación de un pequeño prototipo que sirva para reforzar el mínimo producto viable, ya creado, de cara a comenzar a realizar entrevistas de solución de mayor fidelidad.

Tal y como se comentó en la sección anterior, el título comenzó siendo otro distinto, que se desechó por la mayor dificultad para realizar las fases de testeo, ya que estas consumirían más tiempo. Pese a ello, gran parte de la propuesta de valor planteada en un inicio se mantienen en el nuevo desarrollo.

El primer paso que se dio fue la creación del canvas del modelo de negocio y el mapa de entorno, los cuales encuentran anexos a este documento. Se establecieron las principales líneas a seguir y aspectos a tener en cuenta, destacando la propuesta de valor de Lemon Jellyfish.

La empresa se centrará en ofrecer propuestas jugables que maximicen la experiencia frente al conflicto, queremos crear títulos que se centren en la exploración y fomenten la creatividad del jugador. Creemos que el mercado está saturado de propuestas estresantes que, aparte de vacuas y sin significado, ponen al jugador en situaciones complicadas generando estrés en vez de relajarlo en su tiempo libre.

También apostamos por títulos para todos los jugadores que no infravaloren a los jugadores más nóveles. Partiendo de la premisa “Fácil de jugar, difícil de dominar”, se plantearán argumentos con cierta complejidad que puedan ser disfrutados por un público heterogéneo.

Otro de los aspectos de la propuesta de valor, especialmente enfocado al público mayor de 16 años, es la creación de una comunidad online en la que el feedback durante el desarrollo sea constante, facilitando la posibilidad de testear y aportar ideas o sugerencias a cualquier interesado.

En la siguiente fase, se plantearon las hipótesis destacando entre ellas la necesidad entre los jugadores de ocio electrónico relajante, con un ritmo marcado por el usuario y

no el juego, en los que prime la exploración libre y la narrativa. También se redactaron hipótesis relacionadas con los clientes, su relación con la industria, las vías por las que se informan de las novedades...

Así pues, se redactó un modelo de entrevista de problema, el cual se sacó a la calle para validar las hipótesis. Pese a obtener en principio buenos resultados, se detectó que en el sector del videojuego se obtiene un feedback mucho más fiable si se acompañaba la entrevista de problema con algún boceto, idea o prototipo concreto, ya que, debido a las innumerables ofertas de productos de ocio disponibles en casa y que el uso de las mismas se suele limitar en el tiempo, pese a coincidir en la hipótesis de que muchas de ellas generan estrés, por sí sólo no constituye un problema que influya negativamente en la vida del consumidor.

Así pues, tras ello se procedió a la creación de un producto mínimo viable consistente en un portal web ([www.yoeldsc.es/mapu](http://www.yoeldsc.es/mapu)) en el que se explicará con más detalle la propuesta de Mapu and the lost light. Además, contará con la posibilidad de suscribirse a una lista de distribución de correo, midiendo así el interés suscitado.

Actualmente se está preparando la siguiente fase, que consistirá en la actualización del contenido de la web con recursos gráficos a partir de un prototipo y descripciones detalladas del título. Con esto, se creará una encuesta online de respuesta libre dividida en dos partes, una primera parte enfocada a detectar el problema y una segunda parte enfocada a la solución. Este cuestionario será distribuido por medios online (foros nacionales e internacionales) con objeto de obtener unos resultados más fiables y poder llegar al público de la propuesta de manera más directa.

## Plan de Marketing

### Delimitación del producto o servicio y del mercado

Lemon Jellyfish se dedica al desarrollo de videojuegos (CNAE 5821.- Edición de videojuegos), ofreciendo a sus clientes un producto que cumpla sus necesidades de ocio. Líneas futuras de trabajo se constituyen con proyectos de gamificación como servicio a empresas e instituciones educativas.

En 2016, el mercado de videojuegos en su conjunto generó 99.6 billones de dólares mundialmente. Dentro de esa cantidad, encontramos 41 billones en el mercado de los videojuegos móviles, 36 billones en el mercado de los juegos de PC, 7 billones en consola, cerca de un billón en los deportes electrónicos, 2,7 billones en realidad virtual y 4,4 billones para contenido en plataformas de video. Se estima que para 2019 el mercado del videojuego alcance en torno a los 120 billones de dólares a nivel mundial.<sup>2</sup>

En España, en 2016, los videojuegos generaron 1163 millones de euros en beneficios, suponiendo un 7,4% más que el año anterior y constituyéndose ya como la primera opción de ocio audiovisual (el cine obtuvo 601 millones de euros) entre un público cada vez más diverso, alejándose del estereotipo de varón adolescente.

En España, de 15 millones de jugadores el 56% son hombre y el 44% son mujeres, siendo cerca del 50% mayores de 25 años. La venta física sigue superando a la venta online, aunque esta última ya alcanza los 382 millones de euros.

Lemon Jellyfish se centra en el desarrollo de títulos para mayores de 7 años, para un mercado tanto nacional como, sin obviar la mayor complejidad, internacional. Con internacional nos referimos principalmente a Europa y Estados Unidos, donde las diferencias culturales, al ser menores, facilitan el desarrollo de productos que puedan tener buena acogida en mercados similares.

<sup>2</sup> Informe del estado de la industria en 2016 por SuperData

<https://www.superdataresearch.com/market-data/market-brief-year-in-review/>

## Análisis del entorno en el que se desarrolla el proyecto

Como se ha indicado anteriormente, el mercado del videojuego se encuentra en rápida expansión, constituyendo una tendencia global y con importantes líneas de crecimiento que todavía no son desarrolladas, como el uso de videojuegos para la educación o la salud.

Los videojuegos son cada vez más accesibles a edades más tempranas. Además, los antiguos jugadores han crecido y gran parte de ellos no ha abandonado sus hábitos, lo que está produciendo que el público consumidor de videojuegos medio envejezca. Se cubrirá un espectro de edad aún mayor, siendo predecible que en un futuro la incidencia sobre la tercera edad también sea un hecho habitual.

También, conforme los desarrollos tecnológicos tienen un precio más accesible, es de esperar que en un futuro los videojuegos se expandan a países en vías de desarrollo en los que actualmente no están presentes.

Como factor de crecimiento, cabe destacar también la oferta mundial cada vez mayor de títulos universitarios y cursos formativos de desarrollo de videojuegos. Son principalmente ofertados por instituciones privadas, por lo que pese a la demanda de formación de un volumen cada vez mayor de interesados, no siempre es cubierta debido a motivos económicos.

También cabe destacar que el sector del videojuego comience a ocupar un hueco en la agenda política. En España, el año pasado por primera vez, el Congreso de los Diputados aprobó por unanimidad dos iniciativas legislativas en las que se apostaba por el desarrollo y crecimiento de la industria del videojuego en España, apostando por

equiparar las bonificaciones en el sector a las ya existentes para la industria cinematográfica y espectáculos artísticos.

El mercado del videojuego todavía no ha alcanzado su madurez. Es un mercado en el que la competencia cada vez es mayor. Sin embargo, el entorno es ahora más propicio que nunca a que títulos y productos desarrollados por equipos reducidos y con recursos limitados puedan ser acogidos con gran éxito por el público y la industria. Los desarrollos son más ágiles, los medios son más accesibles y un público cada vez más maduro consume propuestas cada vez más diversas, si centrarse tanto en valores asociados a una mayor inversión económica como la potencia gráfica o las grandes campañas de marketing. Es por ello que en los últimos años ha proliferado la industria de los videojuegos indie, la cual actualmente se encuentra en una encrucijada, debido a que el consumidor exige ahora, a pesar de los limitados recursos, propuestas de calidad, dejando fuera del mercado a títulos *amateur* que eran desarrollados sin motivación empresarial. Es por ello que la innovación juega un papel fundamental en proyectos independientes.

Finalmente, la alta incidencia del videojuego en dispositivos móviles, así como el desarrollo de la realidad virtual y aumentada posibilitan nuevos mercados que serán desarrollados en los años venideros.

## Análisis DAFO

Expuesto todo lo anterior, el análisis DAFO de Lemon Jellyfish lo podríamos definir cómo:

### Debilidades

En primer lugar, ha de ser mencionado la falta de experiencia práctica en el sector y la necesidad de formación. Relacionado con ello también encontramos la actual ausencia de recursos humanos, lo cual es habitual en la industria del videojuego española, con una marcada ausencia de personal cualificado a pesar de la alta tasa de empleabilidad.<sup>3</sup>

Además, debido al estado del proyecto, encontramos el factor de la escasez de recursos económicos.

### Amenazas

La industria del videojuego en España no se encuentra desarrollada; a pesar de constituir el cuarto mercado europeo y el octavo mundial, sólo el 2% de los beneficios retorna al país. La producción propia escasea, de los trabajadores en la industria del videojuego en España, sólo el 10% se dedica al desarrollo, dedicándose el porcentaje restante a la comercialización de productos extranjeros. En 2016 se registraron 480 compañías de desarrollo de videojuegos, las cuales en su mayoría se dedican a desarrollos de bajo coste como pequeños videojuegos para móviles. En conclusión, el crecimiento y la consolidación de los estudios de desarrollo españoles constituye un gran reto a afrontar.<sup>3</sup>

Además, a pesar de que los grandes desarrollos españoles suelen ser respetados en el extranjero, la ausencia de canales de distribución nacionales hace que la venta de los títulos no sea sencilla, teniendo que ser comercializados mayormente por distribuidoras extranjeras.

Finalmente, teniendo en cuenta el mercado internacional, cabe destacar la competencia en el sector, siendo necesario destacar respecto a propuestas de terceros.

### Fortalezas

Lemon Jellyfish nace con una mentalidad madura respecto al medio de comunicación que constituye el videojuego. No es infravalorado como un producto de entretenimiento simple y, a pesar de no contar con formación académica, se ha realizado un profundo estudio del sector y del proceso de desarrollo del videojuego, aspecto que en la mayoría de ocasiones es ignorado en estudios independientes noveles.

Además, dada la situación geográfica del estudio, se tiene mayor acceso a profesionales del sector que en otras regiones de España.

### Oportunidades

Como ha sido mencionado con anterioridad, la industria del videojuego está en pleno crecimiento, siendo España uno de los mayores mercados. Si bien las empresas de desarrollo españolas no son muchas, esto también supone una mayor facilidad a la hora de acceder a inversiones privadas españolas debido a la menor competencia.

Finalmente, la tendencia a una mayor accesibilidad en formación, herramientas de desarrollo y canales de distribución hace que el coste del desarrollo de un videojuego decrezca respecto a etapas anteriores.

<sup>3</sup>Libro blanco del desarrollo español de videojuegos 2016

<http://www.dev.org.es/images/stories/docs/game%20development%20in%20spain%202016.pdf>

## Información interna del proyecto

El proyecto nace por la vocación a trabajar en la industria, además de por la creencia de los videojuegos como herramienta comunicativa capaz de mejorar el mundo y conectar a las personas.

Lemon Jellyfish tiene como objetivo la creación de títulos de calidad sin grandes derroches económicos durante el desarrollo, optimizando recursos, apostando por la transparencia y agilizando los desarrollos mediante el feedback constante del cliente final. Se utiliza la experiencia jugable como medio para aportar valor a la sociedad.

Además, también nos motiva el reconocimiento de la Región de Murcia y de España como generadores de contenido audiovisual de calidad para un público internacional, rechazando los clichés y estereotipos típicos de los productos españoles.

Todo ello también se verá reflejado en la ética empresarial, teniendo siempre presentes la responsabilidad social y la sostenibilidad en todos sus ámbitos.

## Objetivos del Plan de Marketing

En esta primera fase del proyecto se establecen los siguientes objetivos:

- Generación de reconocimiento en el ámbito nacional. Dado que el desarrollo del producto final se prolongará previsiblemente entre un año o dos en el tiempo, a corto plazo es necesario generar interés por el producto para atraer inversiones y asegurar la viabilidad de la propuesta.
- Lanzamiento de una campaña de Crowdfunding en el plazo de un año.

## Estrategia

### Estrategia corporativa

Lemon Jellyfish apuesta por la innovación en videojuegos con un ciclo de desarrollo medio de dos años, mientras que tienen cabida proyectos de ciclo de desarrollo corto para apoyar económicamente de forma paralela a los proyectos principales.

Así, en cuanto a la estrategia corporativa y teniendo en cuenta los objetivos de la empresa, se apostará por una estrategia de diferenciación, poniendo especial énfasis en desarrollar títulos con alto componente creativo e innovador. También se pretende dotar del valor añadido de la participación activa del público durante el desarrollo, fomentando el feedback y la relación con los clientes.

En cuanto a la estrategia de cartera, se optará por el desarrollo de nuevos títulos con una periodicidad de en torno a dos años.

### Estrategias de marketing

Lemon Jellyfish se centra en el desarrollo de títulos para todos los públicos (con calificación PEGI +7). Mientras que no encontramos una segmentación geográfica o demográfica marcada, se tienen en cuenta que los canales de compra de los jugadores más jóvenes no son los mismos que los de un público más adulto, ya que estos últimos utilizan más el medio de compra digital y se pueden permitir la compra por sí mismos.

Dadas estas diferencias, en una primera fase nos centraremos en el sector demográfico de mayores de 16 años, ya que es más sencillo llegar a ellos.

## Plan de acción de marketing

Así pues, de acuerdo a todo lo anteriormente citado podemos concretar la propuesta en función de los siguientes puntos:

### Producto

Los productos ofrecidos son el eje central de Lemon Jellyfish. Con una propuesta de valor muy marcada. Se tiene muy presente la filosofía de la empresa, apostando por títulos diferenciados en el que el componente innovador y de responsabilidad social esté muy marcado.

En un mundo conectado en el que se tiende a que la accesibilidad a los productos audiovisuales sea cada vez mayor, y siendo los videojuegos un medio de comunicación que puede alcanzar un gran impacto, los desarrolladores son responsables de generar contenido de acuerdo a las necesidades éticas del mundo actual.

Actualmente gran parte del mercado es cubierto por propuestas de muy marcado corte comercial y que siguen esquemas propios del inicio de la industria, cuando los videojuegos eran percibidos como un “mundillo” tabú reservado para jóvenes varones con pocas habilidades sociales. Este hecho es totalmente falso, lo que se demuestra en el perceptible agotamiento de la fórmula en favor de propuestas innovadoras y maduras, tanto en lo narrativo como en lo jugable.

Pese a trabajar en un entorno con gran competencia, a la larga raramente se descarta una propuesta completamente en favor de otra, ya que como se trata de productos con una duración limitada antes de agotarse y resultar abandonados, el producto puede seguir siendo atractivo para los jugadores de manera complementaria con otros títulos.

Así, estamos convencidos de que, si se crea un producto de calidad, este tendrá buena acogida en el mercado y se minimizarán los riesgos que pudiera generar la alta competencia.

### Precio

En cuanto a la política de precios, se optará por seguir una fijación de precios de acuerdo al mercado. El precio final típico de cara al cliente de un producto de las características propuestas es de 35€. Dicho precio puede sufrir una ligera rebaja durante las semanas del lanzamiento e incluso gozar de mayor descuento por la venta del producto no finalizado antes del este, tendencia que está gozando de gran aceptación tanto por los jugadores como por los desarrolladores en los últimos años. Los clientes se convierten en inversores a la vez que se les permite acceder al videojuego antes de su salida al mercado y actuar como beta testers, enviando feedback a la empresa.

Además, las participaciones en campañas de ofertas en medios de distribución online pueden servir para revitalizar las ventas del título una vez agotado el impacto inicial,

por lo que son aconsejables en torno al año después de la salida al mercado, siempre teniendo en cuenta el volumen de ventas actual.

Los beneficios del precio de venta final no repercuten directamente en la empresa, ya que las partes que intermedian entre la empresa y el cliente adquieren un porcentaje de cada venta.

## Distribución

En una primera instancia se optará completamente por la distribución digital. Afortunadamente, en los últimos años en el sector de los videojuegos se han facilitado mucho los procesos para lanzar un título al mercado, con portales como Steam o Humble Bundle que suponen un importante escaparate para los jugadores.

Como segundo paso, la distribución física es el siguiente logro a conseguir, aunque este es más complicado debido a que en España los canales de distribución son escasos. A pesar de ello, existen distribuidoras, tanto españolas como europeas, que trabajan con títulos del estilo, por lo que conseguir un acuerdo de distribución es difícil pero posible.

## Promoción y publicidad

La publicidad del videojuego es necesaria desde el momento en el que se dispone de un prototipo o contenido que sirva para ilustrar el aspecto y dinámica final del título (es importante puntualizar la palabra “final”, ya que comenzar la labor promocional con contenidos descuidados a medio hacer puede tener efectos contraproducentes).

Debido a que cada vez es más sencillo desarrollar videojuegos y más propuestas llegan al mercado, destacar se ha convertido en un imperativo que no puede ser dejado de lado. Es por ello que de cara a las labores básicas a realizar dentro del estudio encontramos:

- **Elaboración de un portal web:** Del estudio y/o del título desarrollado. Es necesario que sea atractivo visualmente y que los conceptos y filosofía del videojuego/empresa sean reflejados de manera clara. Es importante la actualización regular.
- **Redes sociales:** La gestión de las redes sociales es un área que no puede ser evitada en el mundo actual. Además de un escaparate publicitario, también servirán de vías de comunicación para el feedback de los clientes.
- **Blog de desarrollo:** Un blog en el que se vaya informando de los avances del proyecto es muy beneficioso por varios aspectos. Por una parte, aumenta la confianza que los clientes tienen en el proyecto, ya que son informados de los avances realizados de manera regular. Además, se genera marketing de contenidos, lo cual repercute en un mejor posicionamiento y reconocimiento de la empresa.

- Videos: Los *trailers* son muy importantes en la dinámica actual de consumo de videojuegos. El canal de vídeo es un medio decisivo de compra para el cliente, por lo que es un área que requiere especial atención.

Además de todo ello, también se apuesta por presencia en foros online del sector y en ferias de videojuegos, de cara a la promoción y la búsqueda de socios e inversiones. En España destacamos la celebración anual de la Barcelona Games World o el Fun & Serious Game Festival, punto de encuentro entre jugadores y la industria del videojuego nacional. Adicionalmente, se estima un gasto proporcional de en torno al 10% del proyecto en términos de marketing y publicidad.

## Personal, procesos y entorno físico

Dado que, al menos en la primera fase del proyecto, los recursos humanos de Lemon Jellyfish serán reducidos, la gestión del personal y la comunicación con los clientes no será una barrera dificultosa. Todos los miembros del equipo deben compartir la visión del estudio y la ética empresarial, lo que se debe ver reflejado en el producto final y las labores de marketing.

Respecto al entorno físico, aunque nuestros canales de distribución y nuestra presencia en el mercado serán online, en estadios más avanzados del proyecto se dispondrá de una sede física que sirva de entorno de trabajo adaptado a las necesidades del equipo y que además sirva para generar mayor confianza en los consumidores.

## Plan de operaciones

El desarrollo de videojuegos es un proceso que puede llegar a ser muy costoso. Es por ello que una de las prioridades de Lemon Jellyfish será minimizar los costes necesarios para producir productos de la calidad adecuada. Al ser un producto de software, los costes consistirán en su mayoría de horas de trabajo dedicadas a los proyectos, por lo que una metodología ágil, flexible y optimizada es necesaria.

## Productos

En el caso del primer título, “Mapu and the lost light”, nos encontramos ante un plataformas en tres dimensiones realizado con el motor Unreal Engine. La elección del motor gráfico es un aspecto fundamental a la hora de plantear un proyecto. El desarrollo de un motor gráfico propio supondría una enorme carga de trabajo y tiempo que, para un estudio de reciente creación, es difícilmente asumible. Afortunadamente, en los últimos años se han hecho accesibles motores de terceros como Unity o Unreal Engine que facilitan enormemente la tarea de los desarrolladores, ya que disponen de una potente herramienta que les permite centrarse en el diseño y la implementación del videojuego sin tener que dedicar tiempo a la programación del entorno de desarrollo.

Se ha escogido Unreal sobre Unity debido a que, a pesar de presentar una mayor dificultad y mayor porcentaje de beneficios perdido a mayor número de ventas, es un motor mucho más sólido y optimizado que Unity, en especial en consolas, a las cuales se planea llevar el proyecto en una futura fase.

Otro software que se utilizará para el desarrollo del proyecto es la suite de Adobe Creative Cloud para diseño gráfico, vídeo... Blender y ZBrush, para modelado 3D, Substance Designer y Substance Painter para la creación de materiales y texturas.

Mediante estas herramientas, se pretende crear un producto de atractivo acabado gráfico y jugable, sin utilizar técnicas fotorrealistas si no optando por una estética estilizada.

## Procesos

Se ha señalado ya la necesidad de una metodología sólida para sacar el máximo partido al tiempo disponible, ya que un ciclo de desarrollo demasiado largo puede tener nefastas consecuencias en el proyecto. Es por ello que se hará uso de la metodología Lean Manufacturing, evitando desperdicios y optimizando el tiempo de trabajo.

Pese a ser originalmente creada para procesos industriales, sus principios pueden ser aplicados en el desarrollo de software. Pongamos como ejemplo el desarrollo de un nivel del videojuego. El proceso podría ser el siguiente: Primero se diseña el nivel en términos jugables, se guioniza, luego se genera el arte 2D e ilustraciones, las cuales servirán al equipo de modelado 3D a diseñar los recursos necesarios para, más tarde, ensamblar el nivel previamente diseñado. Luego se realizarían la programación necesaria para que el nivel cumpla su cometido. Después, se haría el diseño de sonido y, cuando todos estos elementos están en concordancia, se pasaría a la fase de testeo del nivel.

Es fácilmente reconocible la dependencia de las labores unas etapas respecto a las labores de fases anteriores. Si una etapa inicial se alarga en el tiempo puede producir retrasos que se vean agravados aún más en fases posteriores e incluso periodos en los que miembros del equipo del desarrollo no pueden llevar a cabo sus tareas porque tareas anteriores no han sido completadas. Limitando las tareas de los departamentos en tiempos idénticos se puede optimizar la cadena de producción e incluso detectar fácilmente en que parte se producen cuellos de botella y son necesarias ampliar los recursos. Además, al producirse la fase de testeo en unidades funcionales más pequeñas, podemos optimizar esfuerzos al obtener de manera más ágil el feedback del cliente, detectando así cuando este está suficientemente satisfecho con cierto aspecto y ya no merece la pena centrar más recursos en el mismo, pudiendo estos ser aprovechados en otras áreas con mayor necesidad.

Como se ha indicado anteriormente, se planifican ciclos de desarrollo de dos años. Lemon Jellysifh es vulnerable en el primero de esos desarrollos, ya que los ingresos que

aseguren la viabilidad del proyecto serán limitados. Para solventar este aspecto, se abre la puerta a la oferta de soluciones de gamificación para terceros.

## Plan de recursos humanos

Gran parte de los recursos necesarios para llevar a cabo el proyecto lo constituyen el equipo humano que se hará cargo del desarrollo de los títulos. Especialmente en el sector de los videojuegos en España, es complicado encontrar profesionales cualificados, siendo habitual encontrar perfiles de Ingeniería informática y ramas artísticas o audiovisuales.

De manera básica, podríamos dividir el equipo de desarrollo en las siguientes áreas:

- Equipo de diseño: Los encargados de definir las mecánicas jugables, diseñar los elementos del videojuego o guion... Además, sirven de puente entre los equipos artísticos y la tecnológicos, conociendo los límites realizables en ambos realizables en ambas partes.
- Equipo de arte: Son los que elaboran el arte 2D y 3D del proyecto, uno de los tres pilares del videojuego.
- Equipo de programación: De ellos depende la parte tecnológica del título. Dan forma al sentido al videojuego mediante la programación de las mecánicas jugables especificadas.
- Equipo de administración: Aquellos que velan por la viabilidad del proyecto en términos económicos y administrativos.

Aunque se podría extender esta lista con muchos más cargos, como técnicos de sonido, marketing, producción, animador... sería irreal definir esos puestos en los primeros pasos del estudio. Así pues, se establece la necesidad mínima de cuatro personas para cubrir cada una de las áreas en el estadio inicial del proyecto.

De estas cuatro personas dependerá la creación de una primera versión del videojuego, en el plazo de un año desde el inicio del proyecto, que sirva para demostrar el potencial del mismo y atraer inversiones y, con ellas, nuevos recursos humanos al equipo.

Cabe puntualizar la elaboración de bandas sonoras para videojuegos. Dicha labor suele ser externalizada y suponer un coste de entre un 5 y un 15% del total del proyecto, lo cual también se tendrá en cuenta.

## Estructura legal de la empresa

A pesar de que en una primera etapa se podría comenzar el proyecto sin una estructura empresarial fija e incluso como autónomo, gran parte de las subvenciones disponibles para el sector requieren el establecimiento de una sociedad, por lo que se optará por este último escenario.

## Plan de Inversiones y ubicación

Uno de los aspectos positivos del proyecto es que, en la primera etapa, no requiere recursos materiales de manera imperativa. Al tratarse de un proyecto de software, gran parte de las tareas iniciales se pueden realizar en equipos informáticos de potencia media de los miembros del equipo promotor.

Es a partir del primer año, cuando el proyecto toma forma y se comienzan a ver resultados, el potencial de la idea y a generar interés en los clientes, cuando es necesario atraer inversiones, la adquisición de material y determinar las bases administrativas necesarias para comenzar la actividad empresarial.

Llegados a ese punto, si será necesario contar con un espacio físico de trabajo, contar con equipos informáticos que faciliten el desarrollo y la disposición de licencias del software a utilizar.

En cuanto al espacio físico, en Cartagena existen varios viveros de empresas de nueva creación que ofrecen oficinas a precios muy asequibles para las primeras etapas del proyecto. Estas se encuentran en el centro de la ciudad o en el polígono Industrial Cabezo Beaza. También destacar la cercanía del Parque tecnológico de Fuente Álamo, el cual también dispone de un vivero de empresas.

Pese a que las grandes compañías de las Tecnologías de la Información suelen estar ubicadas en grandes ciudades como Madrid o Barcelona, se está experimentando una tendencia a contratar productos tecnológicos a profesionales del sector en la Región de Murcia, debido a su alta cualificación y a la competitividad de los precios.

Es por ello que la creación de empresas dedicadas al sector de los videojuegos es un paso adelante en la dinámica actual, ya que los precios son más bajos y al haber menos competencia en el sector hay más posibilidades de acceder a capital de inversión privado. Pese a ello, en contra encontramos que los grandes eventos y foros de la industria del videojuego de España se realizan en el norte del país, por lo que será indispensable la movilidad nacional para tareas de promoción y búsqueda de socios.

Finalmente, en referencia a las licencias de software, la mayoría de ellas se basan en modelo por suscripción, por lo que es un gasto en el tiempo que deberá ser contemplado.

## Plan económico – financiero

Dada la naturaleza del proyecto, los beneficios derivados de la actividad desarrollada sólo comenzarán a obtenerse por la venta de títulos una vez acabado su desarrollo. Es por ello fundamental encontrar vías alternativas de financiación para que este pueda ser llevado a cabo. En términos económicos, el objetivo principal del primer título a desarrollar es la recuperación durante el primer año de mercado de la inversión utilizada más un margen que posibilite la continuación de la actividad empresarial durante, al menos, un año más.

### Plan de inversión inicial

A pesar de que el proyecto, debido a que la actividad comercial no comienza hasta la promoción o el lanzamiento del producto, puede sufragarse con aportaciones personales, para asegurar un adecuado empresarial se estima la inversión necesaria para cubrir la actividad durante dos años de la siguiente manera:

Concepto	Importe
<b>ACTIVO NO CORRIENTE</b>	
<b>Inmovilizado Intangible:</b>	
Gastos de Investigación y Desarrollo	107.520,00 €
- Sueldo de 850 € por dos años para 4 empleados	
Aplicaciones informáticas	
- Creative Cloud Anual - 840€	
- Substance Suite - 270€	
- Otros Software y plugins - 400€	1.510,00 €
Registro de marca, dominios y servidores	500,00 €
Subcontratación Banda Sonora	13.000,00 €
Marketing Online	10.000,00 €
<b>Inmovilizado Material:</b>	
Instalaciones	2.880,00 €
- Alquiler por dos años de oficina en vivero	
Equipamiento informático	8.000,00 €
- Adquisición de workstations	
- Periféricos y accesorios de diseño	
Gastos de desplazamientos	700,00 €
Adquisición de material publicitario	1.000,00 €
<b>TOTAL INVERSIONES</b>	<b>145.110,00 €</b>

## Plan de financiación

Uno de los objetivos es que el proyecto se financie en gran medida mediante una campaña de crowdfunding a realizar tras el primer año de desarrollo. Los gastos generados hasta ese momento, en términos de licencias, equipos o gastos asociados a promoción y publicitar la campaña, sean cubiertos por los miembros del equipo promotor junto a una pequeña ronda de inversiones de familiares y conocidos. También se planea solicitar subvenciones regionales, nacionales o europeas para conseguir más capital, pero su acceso, especialmente en las de ámbito nacional, es dificultoso, ya que los requisitos suelen ser difíciles de alcanzar y las subvenciones son limitadas, a pesar de que en los últimos años está mejorando el acceso y la cuantía de las mismas.

Concepto	Importe	%
<b>PATRIMONIO NETO:</b>		
Capital y otras aportaciones de socios	8.000,00 €	0,055130591
Subvenciones, donaciones y legados recibidos	20.000,00 €	0,137826476
Campaña de Crowdfunding	70.000,00 €	0,482392668
Deudas a corto plazo con entidades de crédito	50.000,00 €	0,344566191
<b>TOTAL FINANCIACIÓN</b>	<b>148.000,00 €</b>	<b>1,019915926</b>

## Previsión de ventas

Basándonos únicamente en el modelo de ventas de producto, ignorando el servicio de soluciones de gamificación citado, ya que este apartado no ha sido desarrollado durante el proyecto, y teniendo en cuenta el coste del producto (pasando de 35€ a 28€, dado que el mayor volumen de ventas se genera durante el lanzamiento, donde el precio es reducido) menos los porcentajes que se llevarían intermediarios, si consideramos una estimación de ventas inferior a las ventas medias de un título de similares características esperadas en *Steam* (21000 ventas) durante el primer año:

Concepto	Año 1 (Tras lanzamiento)
<b>Mapu and the Lost Light</b>	
Unidades	15000
Precio venta/unidad	28
Gastos intermediarios (Distribuidor, motor gráfico...) -40% Aprox	168.000,00 €
<b>TOTAL VENTAS</b>	<b>420.000,00 €</b>
<b>BENEFICIOS TOTALES</b>	<b>252.000,00 €</b>

## Gastos de explotación

Los gastos de explotación del primer año serán los siguientes, asumiendo que la actividad comercial de la empresa comienza desde el punto de partida del proyecto, lo cual es improbable debido a la naturaleza del desarrollo. A pesar de ello, se considera:

Concepto	Año 1
<b>Otros gastos de explotación:</b>	
Transportes	700,00 €
Publicidad y relaciones públicas	6.000,00 €
Alquiler oficina	1.440,00 €
Software	500,00 €
Dominios, registros y servidores	670,00 €
Otras contingencias	1.500,00 €
<b>Gastos de personal contratado:</b>	
Sueldos y Salarios	40.800,00 €
Seguridad Social	12.960,00 €
<b>TOTAL PRIMER AÑO</b>	<b>64.570,00 €</b>

## Cuenta de pérdidas y ganancias provisional

Atendiendo a lo anteriormente expuesto, son previstos los siguientes puntos:

- a) El resto de la inversión necesaria no gastada por los gastos de explotación del primer año es gastado en el segundo.
- b) El primer producto sale a la venta durante el segundo año
- c) Tras ello, durante el segundo tercer las ventas decrecen un 70%. Comienza el desarrollo del segundo título, para el cual se dobla el personal contratado, así como se triplican los gastos de explotación.
- d) Durante el cuarto año se aumentan las ventas del nuevo título hasta un 1,5 respecto al primero. Los gastos de explotación también aumentan, duplicando los del año anterior
- e) En el quinto año comenzaría el desarrollo de un nuevo producto, con el mismo personal y presupuesto que el anterior. Las ventas del año anterior también disminuyen un 70%

Con todo ello, obtenemos:

Concepto	Año 1	Año 2	Año 3	Año 4	Año 5
Ventas	0,00 €	252.000,00 €	75.600,00 €	378.000,00 €	113.400,00 €
Gastos de personal contratado	53.760,00 €	53.760,00 €	107.520,00 €	107.520,00 €	107.520,00 €
Otros gastos de explotación	10.810,00 €	26.780,00 €	32.430,00 €	64.860,00 €	32.430,00 €
<b>CASH-FLOW</b>	<b>-64.570,00 €</b>	<b>171.460,00 €</b>	<b>-64.350,00 €</b>	<b>205.620,00 €</b>	<b>-26.550,00 €</b>
<b>TOTAL ACUMULADO</b>	<b>-64.570,00 €</b>	<b>106.890,00 €</b>	<b>42.540,00 €</b>	<b>248.160,00 €</b>	<b>221.610,00 €</b>

## Calendario de ejecución y conclusiones

Una vez expuestos todos los aspectos anteriores, está previsto el siguiente hilo de actuación:

1. **Septiembre de 2017**. Primer prototipo jugable y búsqueda de socios
2. **Febrero de 2018**. Establecimiento del equipo de trabajo. Primera búsqueda de inversiones.
3. **Septiembre de 2018**. Comienzo de promoción del título. Por vía online y mediante asistencia a ferias.
4. **Enero de 2019**. Campaña de Crowdfunding. Segunda búsqueda de inversiones privadas.
5. **Septiembre 2019**. Lanzamiento del primer producto.

### Conclusiones

Lemon Jellyfish nace en una época en la que el desarrollo de videojuegos es más accesible que nunca, pero a su vez presenta nuevos retos. En primer lugar, encontramos la alta competencia, lo que unido al encarecimiento del precio de desarrollo de los títulos hace que el desarrollo de productos conlleve cierto riesgo económico, aspecto que muchos desarrolladores dejan de lado al iniciar su andadura en la industria, motivados por pasión y afición sin mentalidad empresarial clara.

Hace 30 años el coste de producción de un videojuego de primera línea, los denominados triple A, era de unos 50,000\$. A día de hoy ya supera a los costes de producir una película de Hollywood, alcanzando cifras muy elevadas: No es raro encontrar desarrollos con presupuestos de 40 o 60 millones de dólares, a los que, sumados los enormes gastos en marketing hace que el coste de un desarrollo ascienda a cifras astronómicas.

Pero en los últimos tiempos, con la aparición de pequeños estudios que, al no poder competir en potencia lo hacen en creatividad, minimizan los costes de las producciones reduciendo así los riesgos y, en algunos casos, superando los beneficios de producciones mayores.

Es un camino complicado, con cada vez más viajeros y en un entorno muy cambiante. A pesar de todo ello, Lemon Jellyfish es capaz en recorrerlo con éxito.