

MEMORIA PROYECTO FIN DE CURSO

ALUMNOS: MIGUEL ÁNGEL SÁNCHEZ DE SABOYA
NACHO RODRÍGUEZ PINTO

Contenido

1.	INTRODUCCIÓN.....	4
2.	JUSTIFICACIÓN DEL PROYECTO	5
3.	OBJETIVO GENERAL Y OBJETIVOS ESPECÍFICOS.....	6
4.	ESTADO DEL ARTE	7
4.1.	DESARROLLO DE LAS PRIMERAS APLICACIONES MÓVILES	7
4.1.1.	ANDROID STUDIO	8
4.1.2.	JAVA	9
4.2.	DESARROLLO DE JUEGOS EN UNITY.	10
4.2.1.	C#	12
4.3.	DESARROLLO DE APLICACIONES WEB	13
4.3.1.	HTML Y CSS	14
4.3.2.	NODE.JS.....	15
4.3.3.	JAVASCRIPT Y JQUERY.....	17
4.4.	OTRAS TECNOLOGÍAS	19
4.4.1.	BASE DE DATOS NO RELACIONALES - MONGODB	19
5.	METODOLOGÍA DE TRABAJO	21
5.1.	METODOLOGÍA DE LEAN.....	21
5.1.1.	CARACTERÍSTICAS DE LA METODOLOGÍA LEAN	21
5.2.	FASES DE EVOLUCIÓN DEL PROYECTO.....	22
5.2.1.	FASE I.....	22
5.2.2.	FASE II.....	22
5.2.3.	FASE III.....	23
5.2.4.	FASE IV.....	23
5.2.5.	FASE V.....	23
5.3.	MARCO TECNOLÓGICO	24
5.4.	HERRAMIENTAS DE GESTIÓN APLICADAS.....	24
5.4.1.	GOOGLE DRIVE	24
5.4.2.	GITHUB.....	25
5.5.	HERRAMIENTAS MODELADO DE SOFTWARE.....	25
5.5.1.	VISUAL PARADIGM [57].....	25
5.6.	HERRAMIENTAS Y TECNOLOGÍAS USUDAS	25
5.6.1.	HTML Y CSS	25
5.6.2.	BOOTSTRAP	26
5.6.3.	NODE.JS.....	26
5.6.4.	MONGODB.....	26

5.6.5.	UNITY Y C#	26
5.6.6.	ANDROID STUDIO Y JAVA	26
5.6.7.	JAVASCRIPT Y JQUERY	27
5.6.8.	PHOTOSHOP CS6.....	27
6.	RESULTADOS.....	28
6.1.	PÁGINA WEB	28
6.1.1.	FUNCIONAMIENTO	28
6.1.2.	DISEÑO	38
6.1.3.	TECNOLOGÍAS USADAS	40
6.2.	ANDROID.....	44
6.2.1.	FUNCIONAMIENTO	44
6.3.	JUEGO EN UNITY.....	53
6.3.1.	MAPA	53
6.3.2.	ENEMIGOS Y TORRETAS	62
6.3.3.	ESCENAS	65
6.4.	DIAGRAMAS DE BASES DE DATOS.	67
6.4.1.	DIAGRAMA UML	67
6.4.2.	DIAGRAMA DE ENTIDAD RELACIÓN.....	70
7.	CONCLUSIONES.....	71
8.	BIBLIOGRAFÍA.....	72
	Bibliografía	72

1. INTRODUCCIÓN

El proyecto que se ha desarrollado está formado por cuatro partes.

La primera parte es una página web, en el que se describe que es lo que se propone nuestra empresa, el plan de futuro y que se puede adquirir dentro de la página. En la página se encuentra un apartado de “registro o inicio de sesión”, para acceder a la cuenta (desde el propio menú, después de iniciar sesión). También se ha desarrollado un sistema de recuperación de contraseña, en caso de que se haya olvidado y se recibe esta misma a través del correo vinculado del usuario. Y habrá un apartado de “Contacta con nosotros”, para que el propio usuario se ponga en contacto con el correo de la empresa.

El segundo apartado es una aplicación de Android, que solo los propios administradores de la empresa gestionen los usuarios. Para poder acceder, se tendrá que loguear el usuario, para acceder a sus datos y a la lista de usuarios registrados. Se podrá visualizar los datos de los usuarios y, además, se podrá deshabilitar o borrar las cuentas de los usuarios.

El tercer apartado es un juego, que se desarrolló con el motor gráfico de “Unity”. El juego esta creado en Unity, pertenece a la categoría de estrategia y es compatible con Windows. Se basa en monstruos que invaden la tierra y se debe ayudar a los humanos, para escapar en un cohete, que se debe llenar de combustible.

En el juego, se debe poner torretas (que se requiere puntos, para poder utilizarlos) para derribar al enemigo, se debe evitar, que el enemigo, destruya el objetivo. Si se gana, saldrá una ventana, indicando de que se ha sobrevivido la ronda, y que se puede acceder al siguiente nivel. En caso de que se pierda, se visualizará una ventana, mostrando de que se ha perdido, y se podrá reiniciar este. Y Esto se guiará, a través de un mapa, y para acceder al juego, se requiere el inicio de sesión (los datos se vincularán a la base de datos)

Y, por último, estará la base de datos, en la cual se puede acceder a todos los datos de las tablas que se hayan desarrollado.

2. JUSTIFICACIÓN DEL PROYECTO

Se ha desarrollado este proyecto, ya que la gente se divierte más jugando a videojuegos, y de que estos tengan mucha innovación, y destaquen sobre los demás.

Los videojuegos están siendo muy influyentes en estos últimos años, y se desea superar tus propios límites, para llegar a tener una puntuación más alta, y llegar a destacar en el videojuego.

El fin que tiene este proyecto, es que las personas se entretengan con los videojuegos que se hayan creado, y de que se enteren de las últimas novedades, que se estén desarrollando.

3. OBJETIVO GENERAL Y OBJETIVOS ESPECÍFICOS

OBJETIVO GENERAL

El objetivo general que se tiene en este proyecto es hacer una gran comunidad, a través del desarrollo de los videojuegos implementados y, además, realizar colaboraciones en el desarrollo de herramientas y de juegos con otras empresas. Los administradores de la empresa sean capaces de ver a los usuarios registrados desde una Apk y, por último, que los usuarios conozcan la información de la empresa, a través de una página web.

OBJETIVOS ESPECÍFICOS

Para poder realizar este objetivo, se requieren varios objetivos específicos.

Se debe de conocer Unity, para la implementación del videojuego, con sus diversas funciones, implementadas en C#.

Para almacenar los datos de todos los usuarios, se requerirá una base de datos no relacional con MongoDB.

Se debe saber Node.js, para la funcionalidad de la página web, además de JavaScript. E implementación de la página con HTML y CSS, con sus librerías correspondientes.

Registrar los datos de los usuarios en una aplicación de Android, desarrollada con Android Studio, con la conexión de datos, a partir de MongoDB y Node.js. Se debe acceder mediante inicio / registro de sesión para administradores. Y se debe de conocer Java, para el desarrollo de la aplicación.

Registro e inicio de sesión en la página web, con el acceso de recuperar la contraseña del usuario. Y un formulario para notificaciones de usuario a empresa.

4. ESTADO DEL ARTE

En este apartado, se van a detallar los lenguajes que se han utilizado, a lo largo del proyecto. Se contarán como empezaron, con sus correspondientes características. Pero, antes de nada, vamos a explicar cómo surgieron las primeras aplicaciones de Android, los primeros juegos y las primeras aplicaciones web. En cada punto, se detallarán los lenguajes utilizados en estas tres casillas.

4.1. DESARROLLO DE LAS PRIMERAS APLICACIONES MÓVILES

Antes de que apareciese los sistemas operativos actuales de móviles de Android e IOS, hubo otro sistema, que fue el inventor del sistema operativo móvil, llamado Palm OS.

Palm OS [1] [2] [3] fue desarrollada en 1996, que se incluía las aplicaciones predeterminadas de un móvil. En este, se incluían la agenda, para poder realizar llamadas, correo electrónico y el bloc de notas.

El inconveniente que se tenía, es que el dispositivo móvil era muy poco frecuente, y no se utilizó mucho en esa época, ya que era novedoso este sistema.

Las aplicaciones tenían una interfaz básica, desarrolladas en C y C++. Las aplicaciones tenían las características más básicas, y no había control de versiones.

En las siguientes ilustraciones, se mostrará el logotipo de Palm OS (Ilustración 1) y como era la interfaz que se tenía, en una de sus aplicaciones, que es el bloc de notas (Ilustración 2).



Ilustración 1: logotipo del Sistema operativo PalmOS, en una de sus versiones más actualizadas.

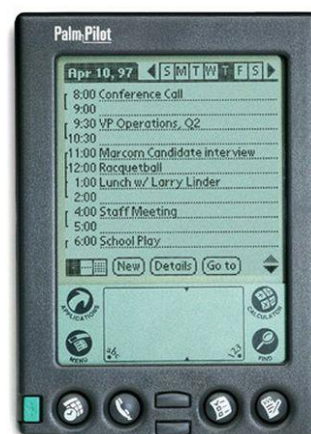


Ilustración 2: interfaz de una de las primeras aplicaciones creadas en este sistema, este es el bloc de notas.

Y más tarde, aparecieron más sistemas operativos, sin embargo, los más importantes son Android e IOS.

Las características que tienen estos sistemas, en cuanto sus aplicaciones son:

- Se lleva a cabo el control de versiones y las actualizaciones de las aplicaciones, con los detalles nuevos, que incluyen la nueva versión de la aplicación.
- La fluidez, a la hora de iniciar la aplicación y los recursos que puedes disponer en ellas.
- Utilización de lenguajes completos. Se pueden apreciar C, C++, Java, Swift, Kotlin etc.
- Implementación de seguridad, para la protección de datos manejados desde la aplicación.
- Y el diseño de la aplicación.

Ahora se va a indicar, la plataforma en la cual se ha desarrollado la aplicación, que esta se corresponde con Android.

4.1.1. ANDROID STUDIO

La aplicación de Android Studio [4] es un entorno de desarrollo, que sirve para el desarrollo de aplicaciones en Android. El programa está desarrollado en IntelliJ IDEA [5] y este compuesto por varios mecanismos:

- Además de poder trabajar con Java, se puede implementar otros lenguajes, como pueden ser C++ [6], Kotlin [7] y NDK [8]. Y también tiene el servicio de Google Cloud, para la facilitar la integración de la aplicación, con los servicios de Google.
- Se pueden guardar los datos creados, a través de la plataforma de GitHub [9] y dispone de compiladores funcionales, con la importación de código.
- Para saber el rendimiento del sistema, disponemos de herramientas desarrolladas en Lint [10]. También dispone de compatibilidad de versiones y de usabilidad.
- Dispone de un emulador rápido, en la que carga las funciones aplicadas. Y tiene un sistema de compilación flexible, que está basado en Gradle [11] .
- Entorno completo y unificado, para todos los dispositivos con sistema Android y tiene diversidad de marcos de trabajo, con herramientas que están en fase de pruebas.

En las siguientes ilustraciones, se va a visualizar el logotipo de Android Studio (Ilustración 3) y como se ve la interfaz del programa (Ilustración 4).



Ilustración 3: logotipo de Android Studio

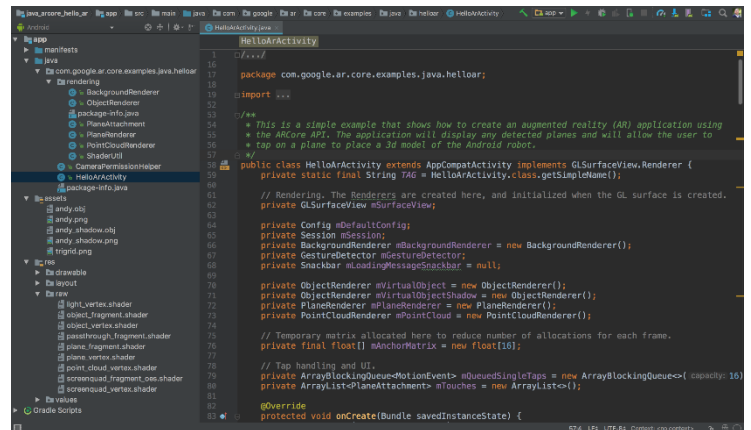


Ilustración 4: interfaz del programa Android Studio

El lenguaje que se aplica en el proyecto de Android es Java [12] y ahora se va a hablar de que trata y que características tiene.

4.1.2. JAVA

Java [13] [14] [15] es un lenguaje de programación orientada a objetos, cuyo diseñador del programa es James Gosling, que fue lanzado en el año 1995.

Es uno de los lenguajes de programación más utilizados en la actualidad, en la cual se puede desarrollar cualquier tipo de aplicación o programa.

No es lo mismo que JavaScript, ya que este se desarrolla en la web, desde un navegador.

A la hora de programar en Java, muchas veces se requiere tener instalado Java en el sistema operativo, ya que muchos de los programas, que se vayan utilizar en Java lo requieren y es necesario instalarlo.

Es multiplataforma, puede soportar cualquier sistema operativo (no es necesario modificar la arquitectura del sistema, ya que este se adapta a ella) y es una plataforma independiente. También este lenguaje notifica si se requiere alguna actualización, y en estas actualizaciones se instalan nuevas herramientas, con la seguridad actualizada.

Y si se necesita programar en la web con este lenguaje, podemos recurrir al XML, con esto, la paginas se ven dinámicas y bien visibles.

Ahora se va a explicar las características que tiene Java:

- Como se ha comentado, es un lenguaje orientado a objetos y es uno de los más utilizados en la actualidad. Y permite diseñar el Software con distintos tipos de datos, ya que estos estarán unidos en la ejecución.
- Es multihilo, ya que puede llevar a cabo varias tareas simultáneas a la vez, en un mismo programa. Mejora el rendimiento y velocidad de ejecución.
- Si no hay ninguna referencia localizada en un objeto, se borra, liberando memoria y con esto, previene la fuga de memoria.
- Distribuye una biblioteca estándar grande y herramientas para que puedan ser distribuidos a distintos programas.
- Java se puede escribir en cualquier tipo de hardware, con lo cual es independiente a la plataforma.
- Y es un lenguaje sencillo, que viene derivado de los lenguajes de C y C++, pero está mejor estructurado, con mucha sencillez de entender.

En la ilustración 5, se mostrará como es el logo de Java.



Ilustración 5: logo de Java

4.2. DESARROLLO DE JUEGOS EN UNITY.

Unity [16] [17] es un motor gráfico, capaz de crear aplicaciones, juegos y experiencias visualizadas en 2D y 3D. La empresa de “Unity Technologies” fue creada en 1988, esta plataforma no tuvo tanto éxito al principio y se iniciaron nuevas herramientas y actualizaciones en este motor gráfico y se consiguió un gran éxito.

Esta plataforma es capaz de desarrollar nuevas experiencias, esto se puede apreciar como en la Realidad Virtual, aplicado para películas animadas y también para los videojuegos. Y también se engloba a la inteligencia artificial.

Además, se puede colaborar con otras personas, desde un mismo proyecto y crear copias de seguridad en la nube. Se puede controlar en que versión se está utilizando el proyecto con la plataforma de Unity y de analizar a los jugadores, en cuanto a la jugabilidad del juego.

También se incorpora una comunidad de usuarios, para el desarrollo de herramientas experimentales, desarrollo de interfaces, análisis del contenido y organización de la aplicación.

Encima la máquina puede desarrollar una gran cantidad de plataformas y es multiplataforma, aunque algunos sistemas están en modo experimental, como se muestra en el sistema operativo de Linux. Y los proyectos se pueden exportar a otras plataformas.

Para poder ejecutar el motor gráfico, se necesita el programa “Unity Hub” para ejecutarlo y elegir la versión correspondiente.

A continuación, se mostrará el logotipo de Unity en la ilustración 6 y la interfaz del programa en la ilustración 7.

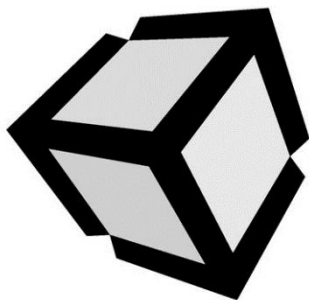


Ilustración 6: logo de Unity.

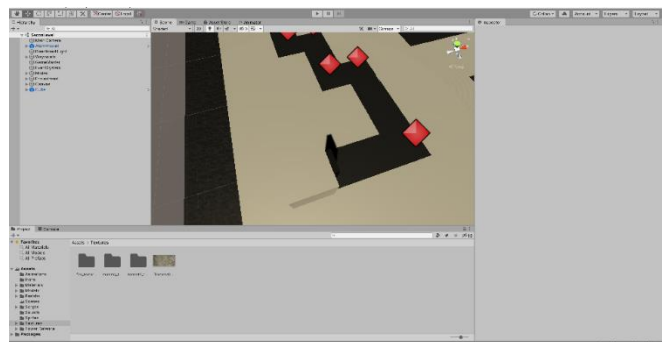


Ilustración 7: interfaz de la aplicación de Unity Hub.

Y ahora se hablará del lenguaje de programación, que se aplica en Unity, y esta es C#.

4.2.1. C#

C# [18] [19] es un lenguaje de programación orientado a objetos, en las que se puede programar aplicaciones, con gran variedad de herramientas y se puede aplicar este lenguaje al .NET Framework, es decir, que se puede programar tanto en web como aplicaciones generales, aplicaciones entorno servidor – cliente, herramientas, componentes, bases de datos etc.

Esta desarrollada por la empresa de “Microsoft” en el año 2000 y dispone de herramientas importantes, como puede ser en la depuración de código y la interfaz de usuario. Se programa a través de las siguientes aplicaciones: Visual Studio Professional, Visual Studio Community y Visual Studio Interprise.

La compilación es sencilla y no hay requisitos a la hora de establecer el orden del código y de declarar los métodos.

Este lenguaje tiene características de los lenguajes de Java, C y C++. Tiene un código más simplificado y permite funciones importantes, como se puede ver estos apartados: encapsulación, polimorfismo y herencia.

Y, además, describe los métodos que se vayan a utilizar en una función, para indicar que es lo que se está realizando en ese momento y que variables o funciones coge.

Ahora se indicará como es el logo de C# (en la ilustración 8) y cuál es el entorno, en la cual se visualiza en Visual Studio (ilustración 9).



Ilustración 8: logo del lenguaje de programación C#.

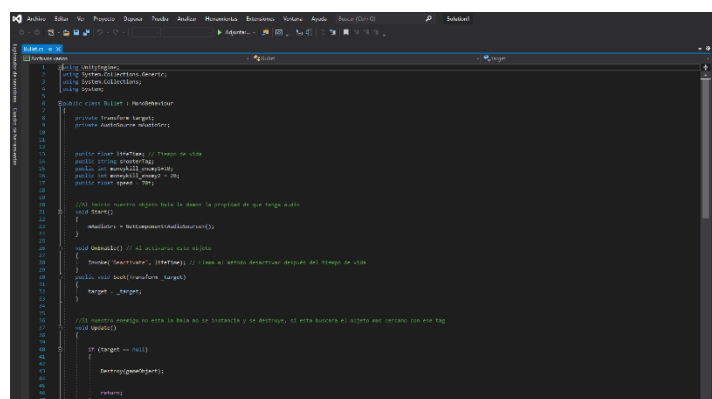


Ilustración 9: interfaz de Visual Studio Professional, donde se puede trabajar con C#.

4.3. DESARROLLO DE APLICACIONES WEB

El desarrollo de las primeras páginas web [20] [21] [22] aparece en la década de los noventa. Estas primeras páginas tenían un desarrollo bastante sencillo y esto se puede apreciar tanto en la estructura del código aplicado, como en el diseño.

Anteriormente solo se podía acceder a pocas páginas y a unas funciones determinadas. Las bases de datos tenían menos seguridad y solo podían acceder una cantidad determinada de usuarios.

Para que la web fuese lo mejor optimizada posible y de que no hubiese ningún tipo de error, se podría recurrir a la página de W3C.org [23], para ver los fallos que se pueden obtener en el desarrollo de la página.

Actualmente se puede ver más desarrollado las páginas web, ya que se pueden aplicar nuevas herramientas, para que su funcionalidad sea mejor, el diseño y la estructura son más desarrolladas.

La seguridad de los datos es más segura y se puede traficar con un mayor número de datos. Y esto se aprecia en la inserción, eliminación y actualización de datos, que se almacenen dentro del servidor.

Hay más navegadores, que disponen de herramientas para la realización de pruebas para aplicación web. Los ejemplos más destacados son Google Chrome, Mozilla Firefox, Opera, Safari entre otros.

Y se puede visualizar en cualquier dispositivo, con lo cual es multiplataforma y se puede visualizar en cualquier medio, independientemente del sistema y del dispositivo que se esté utilizando (también este aparatado está relacionado con la adaptación de pantalla, en cualquier dispositivo - responsive).

Los lenguajes que se han utilizado para programar la página web son las siguientes: HTML y CSS, Node.js, JavaScript y JQuery.

4.3.1. HTML Y CSS

HTML

El HTML [24] [25] es un lenguaje de marcas, que implementa la estructura y el contenido que se tiene en la página web. Se pueden incluir desde textos, hasta enlaces, imágenes y videos.

Al principio este lenguaje era muy básico, y a lo largo de los años se han ido añadiendo nuevas actualizaciones y es fácil de entender e implementar.

Si se requiere más información, en cuanto la implementación del lenguaje, se puede recurrir a W3CSchools, para ver las funciones que se pueden aplicar, tanto en HTML, como en CSS, JavaScript y JQuery entre otros (se indican estos, ya que se requieren en la página web del proyecto).

CSS

Y por último está el CSS [26] [27] [28], que es la hoja de estilos que se aplica en las páginas web. Es importante y necesario para la página web, ya que se define el diseño que va a tener la página.

Esto se enlaza en una plantilla o en una página desarrollada en HTML. Con ello se puede indicar múltiples funciones, para definir la página.

Además, se ha añadido un kit de herramientas, que esta englobado tanto en HTML, como en CSS y JS, y este se llama Bootstrap [29], que será aplicado en el desarrollo de modales y en algunas transiciones y responsive.

A continuación, se mostrará los logotipos de HTML y CSS (ilustración 10).



Ilustración 10: logotipos de HTML (Mano izquierda) y CSS (Mano derecha).

4.3.2. NODE.JS

Node.js [30] es un entorno desarrollado en JavaScript, que fue lanzado en el año 2013 por Ryan Dahl. Este entorno está orientado a eventos asíncronos y está aplicado para crear aplicaciones, que tengan una conexión estable, ya que se enfoca más al servidor que al cliente.

Está relacionado por varios sistemas, como Twisted [31], implementado en Python y Event Machine [32] en Ruby.

Ahora se van a indicar las características [33] [34] [35] , entre las más importantes están:

- Tiene mucha capacidad, a la hora de aguantar múltiples conexiones que se estén realizando a la vez, en un servidor. Y soporta múltiples hilos y es gratuito.
- Se puede gestionar una gran cantidad de peticiones y conexiones, y esto se hace de manera rápida y eficaz.
- Este lenguaje se emplea mucho, en el ámbito de la web.
- Como se ha indicado en la introducción, está basado en eventos asíncronos y orientado a eventos.
- Soporta el gestor de paquetes NPM (Node Package Manager), en la cual contiene una enorme cantidad de librerías, que son de código abierto y puede colaborar la comunidad.
- Tiene escalabilidad y estabilidad.
- Es fácil de emplear, ya que se puede copiar la misma base para otros proyectos.

IMPORTANTE. Además de emplearlo en la página web, también se utiliza en la aplicación de Android, para recoger los datos almacenados en la base de datos.

En las siguientes ilustraciones, se muestra el logotipo de Node.js (Ilustración 10) y como se realiza la conexión a la base de datos, que en este caso se conecta a Node.js, con un paquete NPM [36] llamado NPM Daemon [37]. (Ilustración 11).



Ilustración 11: logotipo de Node.js.

```

Microsoft Windows [Versión 10.0.18362.778]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Wlraire69\Downloads\ProyectoLodeMongo - LampGames>npm start

> nodejs-notes-app@1.0.0 start C:\Users\Wlraire69\Downloads\ProyectoLodeMongo - LampGames
> nodemon src/app.js

[nodemon] 1.19.9
[nodemon] to restart at any time, enter 'rs'
[nodemon] watching: *.*
[nodemon] starting 'node src/app.js'
(node:18192) Warning: Accessing non-existent property 'count' of module exports inside circular dependency
(Use 'node --trace-warnings ...' to show where the warning was created)
(node:18192) Warning: Accessing non-existent property 'findOne' of module exports inside circular dependency
(node:18192) Warning: Accessing non-existent property 'remove' of module exports inside circular dependency
(node:18192) Warning: Accessing non-existent property 'updateOne' of module exports inside circular dependency
Server on port 3000
(node:18192) DeprecationWarning: collection.ensureIndex is deprecated. Use createIndexes instead.
db connected

```

Ilustración 12: ejecución de la conexión a Node.js en CMD.

Ahora se va a hablar de las librerías NPM que se han empleado en el proyecto.

EXPRESS [38]

El paquete Express hace que el desarrollo de las aplicaciones web sean más rápidas y además, aporta múltiples opciones, como es el caso de gestionar las sesiones iniciadas o el enrutamiento que va a tener la aplicación.

PASSPORT [39]

El paquete Passport se encarga de solucionar las necesidades que requiere la aplicación web, en cuanto la autenticación de la aplicación.

EXPRESS.ROUTER [40]

Paquete NPM que está relacionado con el NPM Express. Consiste en enrutar las solicitudes HTTP y también se utilizar para indicar si hay alguna coincidencia de patrones en el enrutamiento.

BYCRYPT [41]

El paquete Bcrypt encripta las contraseñas, lo que hace que sea más seguro las contraseñas de los usuarios.

ASYNC [42]

Paquete NPM que proporciona funciones potentes y directas para trabajar en entornos asíncronos, que se trabajaría en JavaScript. Se puede utilizar directamente en el navegador y es compatible con Webpack y Rollup.

CRYPTO [43]

Paquete similar que Bcrypt. También encripta muchos caracteres, y se le puede asignar fácilmente.

NODEMAILER [44]

Paquete que se utiliza para poder enviar mensajes a correos electrónicos. Se puede emplear en cualquier tipo de correo y es un servicio alternativo para poder solucionar problemas con el Gmail.

En la ilustración número 13, se va a visualizar el logo de NPM.



Ilustración 13: Logotipo de NPM

4.3.3. JAVASCRIPT Y JQUERY

JAVASCRIPT

JavaScript [45] [46] [47] es un lenguaje de programación que consiste en hacer actividades complejas, que se aplican en una página u aplicación web. Se puede decir que es la tercera capa de una página, ya que las primeras son el HTML y el CSS.

Fue creado por Brendan Eich en 1995.

Puede crear un inédito tema, como puede ser el control de archivos o el movimiento de objetos.

Los puntos más fuertes [48] que puede tener este lenguaje son:

- Una de las mejores maneras para la realización de las validaciones de datos.
- Es multiplataforma, se puede trabajar independientemente del sistema operativo que disponga el usuario.
- Es un lenguaje eficiente, bastante sencillo de implementar y se puede ejecutar varias funciones a la vez. Aunque se debe introducir bastante código para lograr su funcionamiento.
- Es un lenguaje FullStack, es decir, se ofrece un sistema de componentes que ofrecen suficientes soluciones, ya que es robusto y funcional.
- Es veleidoso. Se puede producir páginas dinámicas más fácilmente y se logra muchos impactos visuales.

En la posterior ilustración (número 14), se indicará el logo de JavaScript.



Ilustración 14: logo del lenguaje de JavaScript.

JQUERY

Ahora se va a pasar a JQuery [49]. Se trata de una biblioteca de JavaScript, se reduce el código y es rápida, de pequeño tamaño, con muchas funciones.

Esta creado por John Resig y se lanzó en 2006

Maneja muchos eventos, funciona en una amplitud de navegadores y permite el desplazamiento fluido en los apartados HTML.

Es más fácil de utilizar que JavaScript y se pueden ver todas las funciones que tiene, a través de la API oficial de JQuery.

Después se mostrará una imagen (ilustración 15), con el logo de JQuery.



Ilustración 15: Logotipo JQuery.

4.4. OTRAS TECNOLOGÍAS

4.4.1. BASE DE DATOS NO RELACIONALES - MONGODB

Las bases de datos no relacionales [50] [51] [52] son capaces almacenar los datos sin la necesidad de estar relacionados entre sí. Se almacenan en estructuras que son fijas y el punto más importante que tiene este tipo de base de datos, es que no se requiere insertar código SQL, a la hora de insertar, eliminar y modificar los datos, se puede hacer automáticamente, con lo cual hay versatilidad.

Los datos se almacenan en más de una máquina de sistema, con lo cual hay más escalabilidad horizontal y tolera los fallos que se pueden producir en una base de datos.

No se necesitan tantos requerimientos. Hay flexibilidad y adaptación en la base de datos y ampliación de la capacidad, dependiendo de las necesidades del usuario.

Pero una de las consecuencias que puede tener es la limitación de realizar cualquier operación, a la hora de relacionar los datos.

Hay muchas bases de datos no relacionales, pero una de las más importantes es MongoDB.

MONGODB [53]

MongoDB es una base de datos no relacional, que está orientado a documentos. Y es una de las bases de datos más importantes en la actualidad.

Se lanzo esta BBDD en el 2009 y está desarrollado en C++, Python, JavaScript y Go. Y se da aviso a las últimas actualizaciones que dispone el programa, con las características nuevas que se han incluido en la BBDD.

No se requieren el uso de filas y columnas, que se pueden apreciar en las bases de datos relacionales y este, almacena multitud de colecciones.

Tiene escalabilidad y flexibilidad, con un módulo de consultas bastante avanzado.

Se puede seleccionar la versión en la que se vaya utilizar con su lenguaje correspondiente, es decir, puedes adaptar esta base de datos a cualquier lenguaje, con la versión correspondiente.

Se puede almacenar datos almacenados en JSON y se puede asignar a los objetos en el código de su aplicación.

Base de datos distribuida en su núcleo, con la consulta de datos a tiempo real, en las cuales se pueden acceder y analizar. Y es de uso gratuito

Se puede instalar el programa de MongoDB, independientemente del sistema operativo en el que este, con lo cual es multiplataforma. Y se puede conectar los datos, a través de Mongo Atlas.

En la ilustración 16, se ve como es el logo de la plataforma y en la ilustración 17 se indica la interfaz que tiene, a través de la aplicación web, que es el MongoDB Atlas.



Ilustración 16: logotipo de la base de datos de MongoDB.

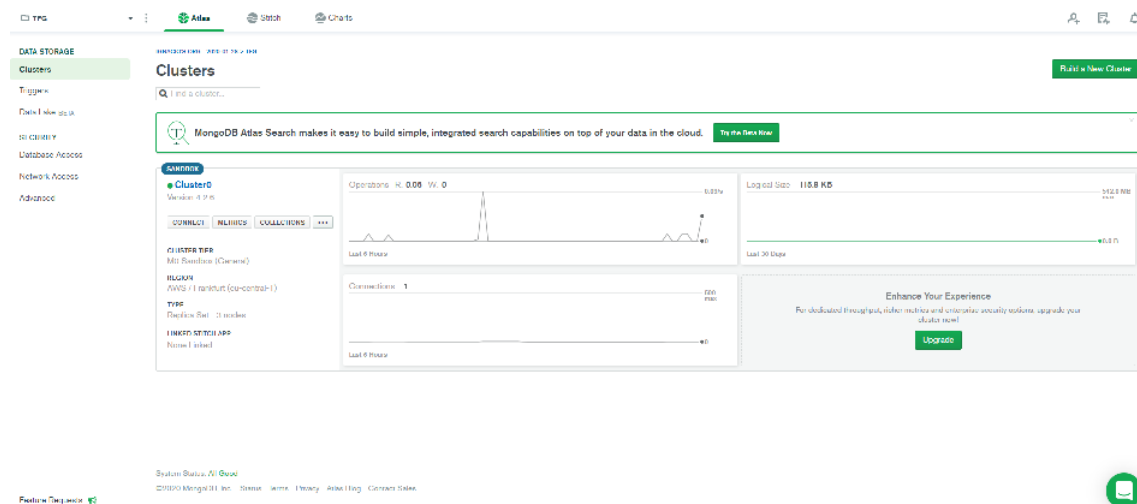


Ilustración 17: visualización de la BBDD de MongoDB Atlas, abierto en una página web.

5. METODOLOGÍA DE TRABAJO

En este apartado se va a detallar que metodología se ha aplicado al proyecto, describiendo en que consiste y como se lleva a cabo.

5.1. METODOLOGÍA DE LEAN

La metodología de Lean [54] [55] [56] consiste en gestionar los proyectos, para que estos sean eficientes a la hora de ejecutar el proyecto.

Esta metodología fue creada por Lean Manufacturing en los años 50, y se realizó en una de las compañías más importantes del automovilismo, que es Toyota.

Antes de nada, se tiene que identificar los valores que vayan a utilizar en cada caso, y luego se da prioridad para optimizar el resultado y eliminar las cosas que sean innecesarias o que no se estén utilizando en ese momento.

Este método se utiliza para que el cliente este satisfecho, utilizando el proyecto desarrollado y escuchar las críticas que se tienen, para irlo mejorando y de que se haya hecho una estructuración correcta.

Y según Lean, es importante la participación de todas las personas integradas en el proyecto y de que haya mejoras y sostenimiento. Las cosas que no generen ningún beneficio al cliente, serán eliminadas, y esto evitará de que haya consumos que sean innecesarios, dentro del proyecto que se ha realizado.

Ahora se van a describir las características que tiene esta metodología.

5.1.1. CARACTERÍSTICAS DE LA METODOLOGÍA LEAN

Las características son:

- Asignación de apartados. Se tienen que distribuir los puntos con todas las personas involucradas. Esto sirve para identificar los factores importantes y de menor valor del proyecto.
- Actualización. Una vez finalizado el proyecto, puede haber algún problema, y esto se debe solucionar, añadiendo o reutilizando las funciones que hayan sido o no creadas.

- Identificar apartados. Se debe analizar el valor que se vaya a realizar, para luego pasar al siguiente paso.
- Distribución del trabajo. Evitar los problemas que puedan surgir en el proyecto, analizando consecuentemente las funciones aplicadas, evitando también las cosas que sean innecesarias y ocupen mucho espacio.

5.2. FASES DE EVOLUCIÓN DEL PROYECTO

Aquí se va a comentar que fases se han aplicado a lo largo del desarrollo del proyecto.

En cada fase, se va a explicar que es lo que se ha realizado a cabo y la finalidad obtenida de la misma.

5.2.1. FASE I

En la primera fase, se analiza cómo está el mercado, en cuanto a la idea que se va a desarrollar, para llevarse a cabo, ya que, si no hubiese mercado, no se podrá llevar a cabo y tiene que convencer a los clientes.

Después de haber analizado esta situación, se tendrá que investigar los tipos de gustos que tienen los clientes y adaptarlos al proyecto, con los requisitos de los clientes y algo innovador, para captar la atención de los usuarios que lo vayan a utilizar.

5.2.2. FASE II

En la segunda fase se tendrá que decidir las tecnologías que se utilizarán, para llevar el proyecto a cabo. Y para tener una organización, a la hora de utilizar el proyecto que se lleve a desarrollar, se tendrá que crear los diagramas UML y entidad / relación.

Se hace esto para simular, como se comportaría el proyecto y que procesos se llevarían a cabo, desde el punto del cliente y del administrador de datos.

Para ello, se tendrá una base de datos, para guardar todos los datos que se utilicen (en el caso de este proyecto, se ha usado una base de datos no relacional) y la administración requerida.

5.2.3. FASE III

En la tercera fase, se tendrá que hacer un croquis, de cómo se va a visualizar la página web, el juego y la aplicación de Android.

Se buscan los colores que se vayan a usar y como se va a distribuir el proyecto. Con estos pasos se puede ahorrar tiempo en el desarrollo del sistema y que beneficio se obtendrá con la distribución que se haya propuesto.

Y planificar, quién es el que se tiene que encargar de cada apartado del proyecto.

Cuando se tenga todos estos pasos, se podrá pasar a la fase IV.

5.2.4. FASE IV

Se comienza con el desarrollo del proyecto. Se tiene que tener en cuenta las cosas que se están realizando, y si pueden repercutir o no al desarrollo del programa.

Para ello hay que tener una estructuración y un orden, a la hora de desarrollar el modelo o las funciones que se integran dentro de él.

Se debe de llevar una serie de pruebas, para el correcto funcionamiento, cuando se haya completado esa función, se continua con el siguiente, sin embargo, si pasa lo contrario, se deja comentado para arreglar y se continua con otra cosa.

También se debe de asegurar el entorno adecuado para trabajar, si se produce algún problema, se tiene que indagar el error buscando las herramientas y los manuales necesarios, para su completo desarrollo.

5.2.5. FASE V

Cuando se haya terminado todas estas fases, se debe tener un punto de inflexión, sobre lo desarrollado en el proyecto. Se pondrán las cosas en común, que han sido desarrollados y que cosas se deben añadir.

En el caso de añadir algún factor, se añadirá a los diagramas desarrollados y a la estructuración del proyecto.

Si todo está correcto, se dará un visto bueno a la aplicación desarrollada.

5.3. MARCO TECNOLÓGICO

En el marco tecnológico se van a indicar las tecnologías que se han aplicado en el proyecto. Se ve reflejado en la ilustración 18.

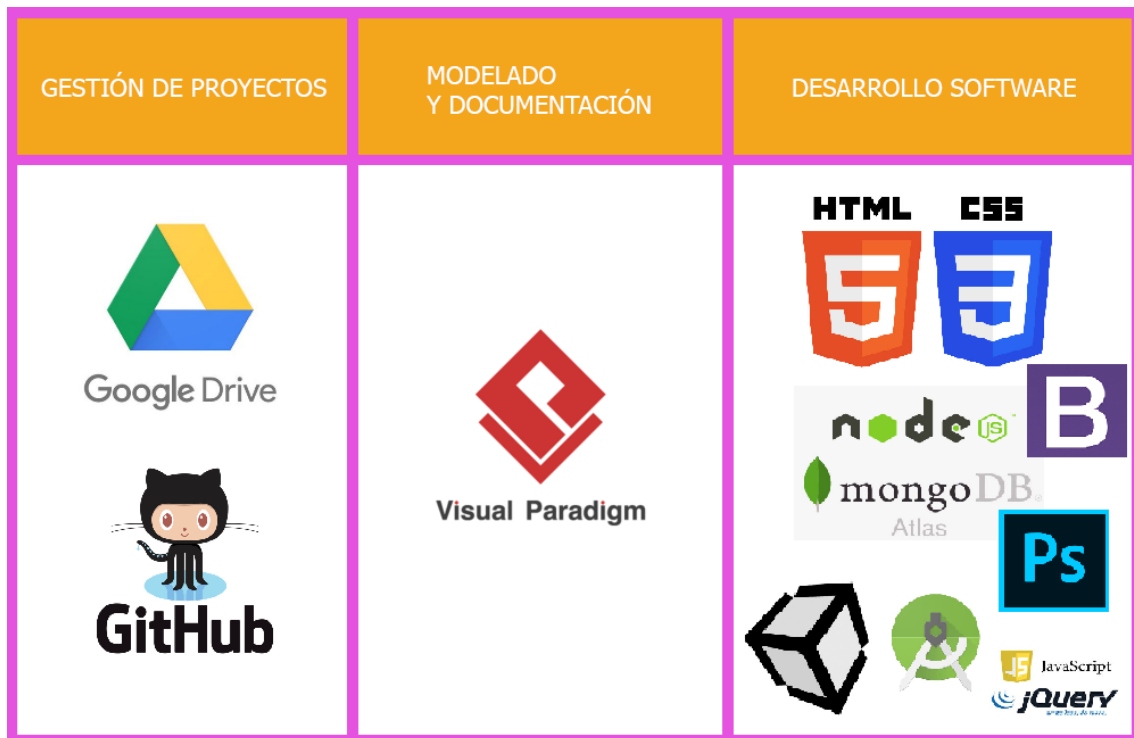


Ilustración 18: Tabla con las tecnologías que se están utilizando.

5.4. HERRAMIENTAS DE GESTIÓN APLICADAS.

5.4.1. GOOGLE DRIVE

Google Drive es un medio para almacenar todas las versiones que se hayan ido trabajando en el proyecto.

Esto se ha realizado como caso de emergencia, en caso de que la siguiente opción no estuviese disponible en ese momento, se recurriría a esta alternativa.

Pero, aunque esté disponible esa alternativa, por seguridad se guarda en Google Drive.

5.4.2. GITHUB

GitHub es un sistema de gestión de proyectos, en la cual se puede hacer un control de versiones sobre estos mismo. Se podría decir que es una red social para los desarrolladores.

Es un repositorio Online bastante importante y estos repositorios se pueden poner públicos (todos los desarrolladores que accedan a GitHub pueden verlo) o privados (los participantes que estén en ese repositorio solo pueden acceder a verlo).

Se puede conectar en cualquier entorno y recibir las últimas actualizaciones que se hayan realizado, con el nombre del usuario que haya modificado y subido los cambios a la plataforma.

5.5. HERRAMIENTAS MODELADO DE SOFTWARE

5.5.1. VISUAL PARADIGM [57]

Visual Paradigm se trata de una herramienta CASE, que se encarga de proporcionar un conjunto de herramientas para el desarrollo de programas. Y esto influye en el diseño, análisis y en la planificación de tareas.

Hay varios tipos de diagramas, pero una de las más aplicadas son los diagramas UML, que están relacionados con las bases de datos, pero hay otros tipos como pueden ser los diagramas de clases o diagramas de caso de uso, entre otros.

5.6. HERRAMIENTAS Y TECNOLOGÍAS USUDAS

5.6.1. HTML Y CSS

Se ha aplicado estos lenguajes para el desarrollo del proyecto de desarrollo de aplicaciones multiplataforma. Toda la información se puede ver en el punto 4.3.1, en la página 14.

5.6.2. BOOTSTRAP

Se ha aplicado este kit de herramientas para el desarrollo del proyecto de desarrollo de aplicaciones multiplataforma. La información de este apartado, se ubica en el punto 4.3.1, en la página 14.

5.6.3. NODE.JS

Se ha aplicado este lenguaje para el desarrollo del proyecto de desarrollo de aplicaciones multiplataforma. La información está en el punto 4.3.2, en las páginas 15, 16 y 17.

5.6.4. MONGODB

Esta base de datos se utiliza para el desarrollo del proyecto de desarrollo de aplicaciones multiplataforma. La información se encuentra en el punto 4.4.1, páginas 19, 20 y 21.

5.6.5. UNITY Y C#

El motor gráfico Unity y el lenguaje de C# son aplicados en el proyecto de desarrollo de aplicaciones multiplataforma. La información está en los puntos 4.2 y 4.2.1, en las páginas 10, 11 y 12.

5.6.6. ANDROID STUDIO Y JAVA

La plataforma de Android Studio y el lenguaje de Java han sido usados en el proyecto de desarrollo de aplicaciones multiplataforma. La información se puede buscar en los puntos 4.1.1 y 4.1.2, en las páginas 8 y 9.

5.6.7. JAVASCRIPT Y JQUERY

JavaScript y JQuery son lenguajes que se han aplicado en el proyecto de desarrollo de aplicaciones multiplataforma. La información se encuentra en el punto 4.3.3, en las páginas 18 y 19.

5.6.8. PHOTOSHOP CS6

Photoshop es un programa, que se encarga de la edición de imágenes. Esta aplicación es desarrollada por Adobe System Incorporated, que soporta los sistemas operativos de Windows y MacOS.

Se utiliza esta versión, ya que aplica nuevas funciones de color y recortes, permite aplicar la herramienta de desenfoque, selección de capas y máscaras y herramientas mejoras de autocorrección.

Este programa se utiliza para la creación del logotipo de la empresa y de la modificación de imágenes (tanto de forma, como de color), que se muestren en la página web.

6. RESULTADOS

6.1. PÁGINA WEB

En este apartado vamos a describir todos los puntos que tiene la página web y que tecnologías se han aplicado.

6.1.1. FUNCIONAMIENTO

Primero se va a hablar de lo que se encuentra al principio de la página.

Se encuentra el menú situado a la derecha de la página, y el logotipo se ubica a la izquierda. Si se pulsa sobre él, te re direccionará al principio de la página. Si pulsamos sobre algún apartado del menú, se nos ubicará sobre ese punto indicado, ya que se ubica todo en una misma página. Los apartados que tiene el menú, son los siguientes (que estos se deben modificar en la página web, ya que en este momento no están puestos): Inicio, Acerca de nosotros, Servicios, suscripciones, equipo, clientes, contacta con nosotros e Inicio y Registro de sesión.

Y se cumple el responsive de la página, el menú se oculta en un icono, cuando se pulsa sobre él, se nos desplegará a la izquierda de la pantalla.

En la ilustración 19, se muestra la vista principal de la página web desde un ordenador. Y a continuación, se ve la ilustración 20, en la que se muestra la vista de la página, en un dispositivo móvil con el menú cerrado, sin embargo, en la ilustración 21, se visualiza el menú desde el dispositivo móvil.



Ilustración 19: Inicio de la página web, en vista de ordenador.

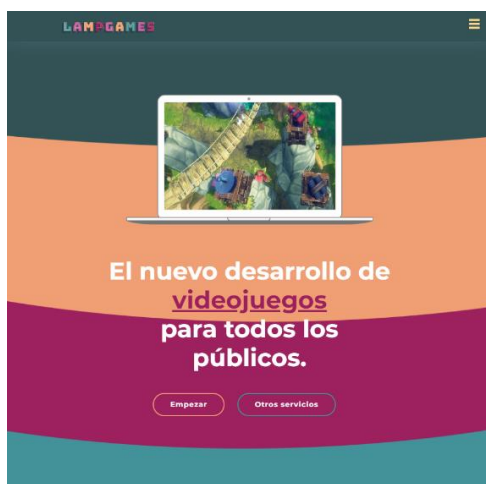


Ilustración 20: Inicio de la página web, vista móvil con el menú cerrado.

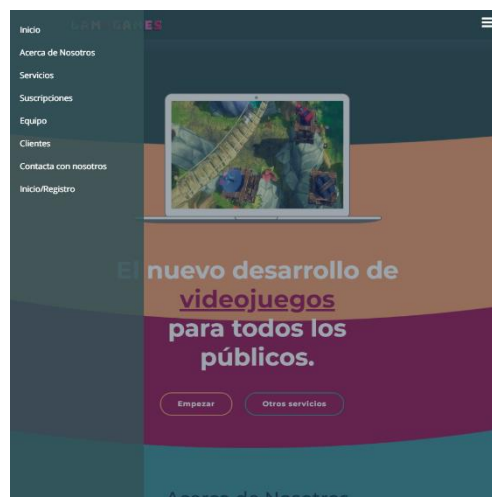


Ilustración 21: Inicio de la página web, vista móvil con el menú en ejecución.

A continuación, tenemos el apartado de “Acerca de nosotros”, que aquí se hablará de que es lo que hace la empresa, lo que proponemos y planes de futuro a llevar a cabo.

En la ilustración 22, se indica como es la primera parte del apartado de “Acerca de nosotros” y en la ilustración 23, se plasma la segunda parte.



Ilustración 22: Primer apartado de "Acerca de Nosotros".



Ilustración 23: Segundo apartado de "Acerca de Nosotros".

Después, se ve el siguiente apartado, que son los servicios que se ofrecen a los usuarios.

En la ilustración número 24, se va a reflejar el apartado de "Servicios", con las funciones que se proponen en la empresa.

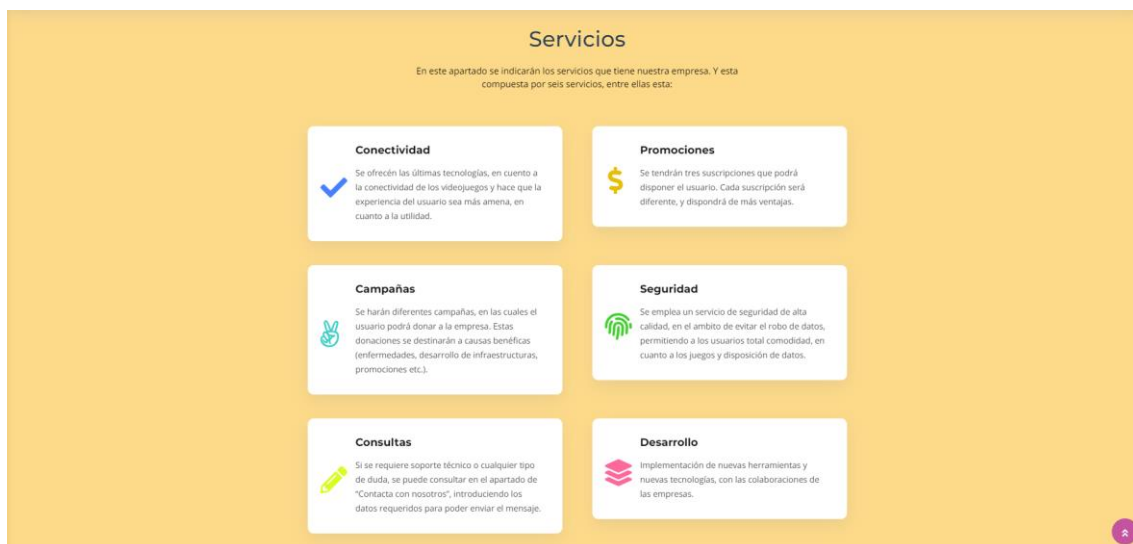


Ilustración 24: Vista del apartado de "Servicios".

Más adelante, se encuentra las suscripciones que tienen para los clientes, va a estar constatado por tres suscripciones: la primera es la gratuita, la segunda es la suscripción “Pro” y por último se tiene la suscripción “Ultra”.

En la ilustración 25, se puede apreciar la vista que tendría el apartado de “Suscripciones”.

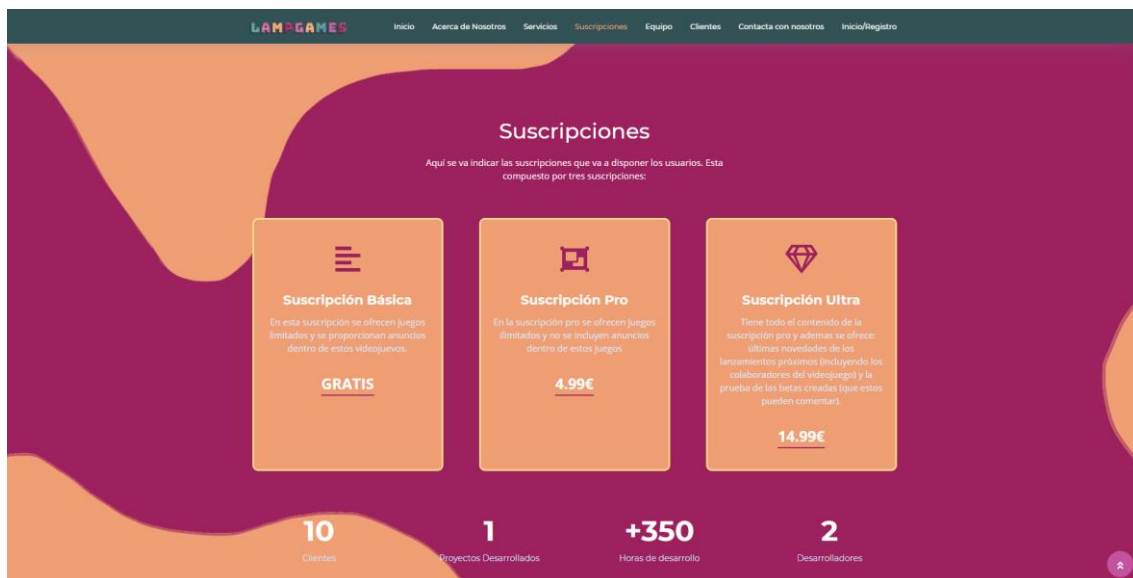


Ilustración 25: Vista del apartado de "Suscripciones".

Más tarde está el apartado del equipo, que son los pilares importantes de la empresa, y en la cual se puede acceder al perfil de las redes sociales, que se tengan vinculadas (en este caso se han cogido cuatro imágenes, procedentes del banco de imágenes, para mostrar cómo se visualizaría los perfiles originales).

En la ilustración 26, se mostraría como es este apartado en cuestión.



Ilustración 26: Vista del apartado de "Equipo".

Además, se cuenta con otro apartado, que será el de clientes. En este se podrá visualizar las empresas para hacer colaboraciones de proyectos, ya que la sistemática de juego que se quiere ofrecer, está relacionada con estas compañías.

En la ilustración 27, se indica como es la vista de apartado de clientes.



Ilustración 27: Vista del apartado de clientes.

Y en el último apartado, nos encontramos el apartado de “Contacta con nosotros”. Se podrá enviar las dudas que el usuario. Será obligatorio poner el nombre y el correo del usuario que se ponga en contacto con la empresa, y también se requiere el asunto y el mensaje del solicitante. Después de haber cumplido todos estos requerimientos, se podrá enviar el mensaje al correo oficial de la empresa, y este lo recibirá en su bandeja de entrada.

También se visualiza el mapa, con la dirección de ubicación de empresa, con su número de teléfono correspondiente y el correo oficial de la empresa

A continuación, se mostrará la ilustración número 28, en la que se podrá apreciar la vista de contacto.

Ilustración 28: Vista del apartado de "Contacta con nosotros".

Y se debe contar también con un pie de página, para ver los términos de privacidad, datos de la empresa y re direccionamiento de vistas

En la ilustración 29, se indica como es el pie de página y en la ilustración 30 es un botón que lo que hará es redireccionarse al principio de la página web.



Ilustración 29: Pie de página de la web.



Ilustración 30: vista del botón de re dirección al apartado inicial de la página.

Ahora, se explicará el apartado de inicio y registro de sesión. Si se pulsa sobre “Inicio”, se nos mostrará un modal. El modal que es visible por defecto, es el de inicio de sesión, se tendrá que ingresar el correo de la cuenta vinculada y la contraseña.

En el caso de no tener cuenta, seleccionados el apartado de registrar sesión (que está dentro del propio modal). Y se tendrá que ingresar una serie de datos para registrarse a la cuenta.

En la ilustración 31, se ve la vista del modal de inicio de sesión y en la ilustración 32 se contempla la vista de registro de sesión.

Log In Sign Up

INICIA LA SESIÓN PARA ACCEDER A TUS DATOS

Correo electrónico

Contraseña

[Se me ha olvidado la contraseña](#)

Iniciar Sesión

Ilustración 31: modal de Inicio de Sesión.

Log In Sign Up

REGÍSTRATE A NUESTRA PÁGINA WEB

Nombre

Apellidos

Email

Usuario

Contraseña

País

Calle

Fecha Nacimiento
dd/mm/aaaa

Teléfono

Imagen de perfil

Seleccionar archivo Ningún archiv...seleccionado

Registrarse

Ilustración 32: modal de Registro de Sesión

En el caso de que haya salido correctamente (tanto el inicio, como registro de sesión) se nos mostrará una alerta, indicando que todo ha salido correctamente.

En la ilustración 33, se divisa la imagen de alerta, que en este caso es cuando se registra alguien en la página web.

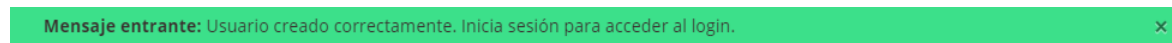


Ilustración 33: alerta indicando que se ha creado correctamente el usuario.

Si pasa lo contrario, y se ha introducido algo que está incorrecto o que ya está registrado actualmente el correo indicado, se nos mostrará una alerta, indicando el error cometido.

En la ilustración 34, se indica la imagen de alerta, que en este caso es cuando se introduce mal la contraseña, o el correo del usuario.



Ilustración 34: alerta indicando que se ha equivocado en alguna de las propiedades indicadas en la imagen.

Cuando se haya logueado correctamente la cuenta, nos aparecerá en el menú un nuevo apartado. En él se da la bienvenida al usuario, en la cual se podrá disponer de dos opciones, en ver el perfil o en cerrar la sesión.

En la ilustración 35, se puede apreciar las opciones que tiene un usuario, al haber iniciado la sesión. Cuando se haya logueado, desaparece la opción de “Inicio”, que se apreciaba en la imagen.

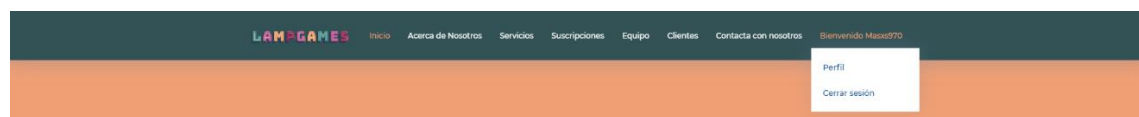
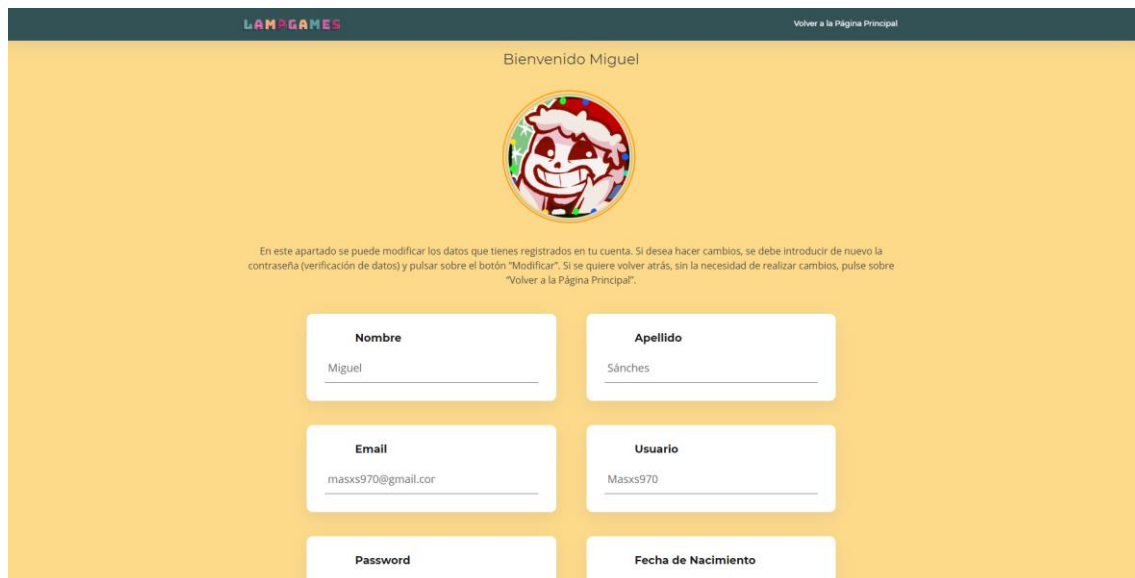


Ilustración 35: apartado del menú, cuando un usuario inicia la sesión en la página web.

Cuando se selecciona el apartado de perfil, se nos mostrará todos los datos del perfil que se haya logueado actualmente. Y se puede modificar los datos y actualizarlos, seleccionando el botón de modificar.

Y para la verificación de datos, se tendrá que introducir de nuevo la contraseña para modificación. Por último, si queremos volver a atrás, sin realizar ninguna modificación, se debe pulsar sobre “Volver a la Página principal”.

En la ilustración 36, se puede ver la vista de los datos del perfil logueado.



The screenshot shows a user profile page for 'LAMPGAMES'. At the top, there is a dark header with the logo 'LAMPGAMES' on the left and a link 'Volver a la Página Principal' on the right. Below the header, the text 'Bienvenido Miguel' is centered. Underneath is a circular profile picture of a cartoon character with a red hat and a wide smile. Below the profile picture, there is a paragraph of text: 'En este apartado se puede modificar los datos que tienes registrados en tu cuenta. Si desea hacer cambios, se debe introducir de nuevo la contraseña (verificación de datos) y pulsar sobre el botón "Modificar". Si se quiere volver atrás, sin la necesidad de realizar cambios, pulse sobre "Volver a la Página Principal".' Below this text are six input fields arranged in three rows and two columns. The first row contains 'Nombre' (Miguel) and 'Apellido' (Sánchez). The second row contains 'Email' (maskx970@gmail.com) and 'Usuario' (Maskx970). The third row contains 'Password' and 'Fecha de Nacimiento'.

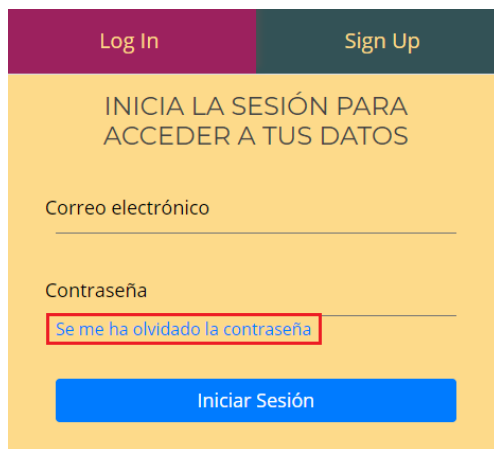
Ilustración 36: vista del perfil de la cuenta logueada (Se sigue realizando este proceso).

Y se finaliza la página con la recuperación de contraseña. Para poder recuperar, se debe ir al apartado de inicio, y habrá un apartado debajo de la contraseña, que pondrá “se te ha olvidado la contraseña”.

Cuando se pulsa sobre este enlace, se generará otro modal, indicando de que, si se requiere recuperar la contraseña, se debe poner el correo que esté vinculada.

En caso de que se haya puesto mal el correo, saltará una alerta, indicando de que ese correo no está vinculado con ninguna cuenta. En caso contrario, se nos mostrará una alerta, indicando de que debemos ir al correo, para recuperar la contraseña.

En la ilustración 37, se indicará dónde está situado el enlace. Y en el enlace 38, el modal de recuperación de contraseña, introduciendo el correo vinculado.



Log In Sign Up

INICIA LA SESIÓN PARA ACCEDER A TUS DATOS

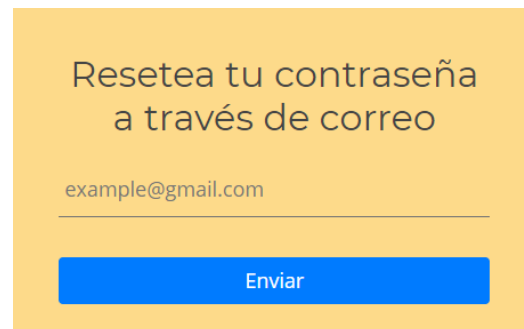
Correo electrónico

Contraseña

[Se me ha olvidado la contraseña](#)

Iniciar Sesión

Ilustración 37: modal de login, resaltando donde está la opción de recuperar contraseña.



Resetea tu contraseña a través de correo

example@gmail.com

Enviar

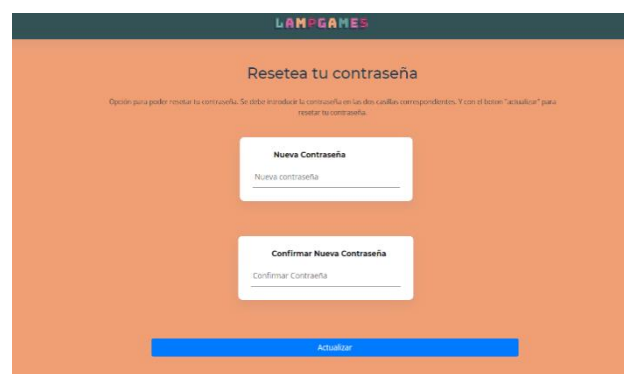
Ilustración 38: modal de recuperación de contraseña.

Cuando se haya ido al correo y se haya pulsado sobre el enlace, podremos recuperar la contraseña. Y cuando se haya confirmado la contraseña, se da al botón actualizar, y se indicará que la contraseña se ha modificado correctamente.

En caso de que se haya introducido mal la contraseña, pulsemos sobre el enlace de modificación de nuevo (al haber modificado la contraseña) o se haya pasado un tiempo determinado, nos indicará que no se puede volver a cambiar la contraseña, desde el enlace solicitado.

Cuando se haya modificado la contraseña, se nos enviará un correo, indicando que la contraseña se ha cambiado correctamente.

A continuación, se muestra la interfaz de cambio de contraseña en la ilustración 20.



LAMP GAMES

Resetea tu contraseña

Opción para poder resetear tu contraseña. Se debe introducir la contraseña en los dos campos correspondientes. Y con el botón "Actualizar" para resetear la contraseña.

Nueva Contraseña

Nueva contraseña

Confirmar Nueva Contraseña

Confirmar Contraseña

Actualizar

Ilustración 39: vista de cambio de contraseña.

6.1.2. DISEÑO

En la ilustración 40 se van a mostrar los colores que se han utilizado en la página web.



Ilustración 40: imagen de los colores que se han utilizado en la página web.

Se han seleccionado estos colores, ya que llaman la atención del usuario. Se utilizan colores fríos y cálidos, para activar los diferentes sentidos, dependiendo del color frío o cálido, se consigue que el jugador este más tranquilo o impaciente o nervioso, por conseguir ganar en el juego.

Dependiendo del lugar donde ocurra el juego, se pondrá un color u otro para dar más o menos amplitud al paisaje.

En el mundo de los videojuegos, los colores de diferente tono cromático, se utilizan sobre todo para enfrentamientos, un equipo contra el otro.

También se han incorporado iconos de la página de Font Awesome, se cogen a través del script que se va a mostrar a continuación, en la ilustración 41. Y en la ilustración 42, podemos ver el logotipo de “Font Awesome”.

```
<link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.8.2/css/all.css"
integrity="sha384-
oS3vJWv+0UjzBfQzYUhtDYW+Pj2yciDJxpsK10YPAYjqT085Qq/1cq5FLXAZQ7Ay"
crossorigin="anonymous">
```

Ilustración 41: script para utilizar la herramienta de "Font Awesome".



Ilustración 42: logotipo de la página de Font Awesome.

Y las imágenes que se han cogido, pertenecen al banco de imágenes, y también este retocado por Photoshop, aplicando los colores que se muestran en la página.

En la siguiente imagen, visualizada en la ilustración 43, se puede apreciar una imagen, que ha sido modificada por Photoshop y que pertenece al banco de imágenes.



Ilustración 43: imagen modificada en Photoshop, utilizada en la página web

6.1.3. TECNOLOGÍAS USADAS

Para la creación de la página web, se ha requerido la utilización de los distintos lenguajes: HTML5 (Para la estructuración de la página web), CSS (Diseño de la página web), JavaScript (funciones de movimiento de la página), NodeJS (Para la funcionalidad de la página web, en cuanto los procesos) y MongoDB (Base de datos).

Ahora se pondrán dos ejemplos de código, para ver qué es lo que se está realizando en esa función en cuestión.

Para el envío de “modificación de contraseña de usuario”, se ha realizado una función de NodeJS, que lo que hace es recopilar los datos que se están indicando dentro del input, después analiza si ese correo existe en la base de datos, y cuando lo encuentra, se envía el mensaje creado. Se ha requerido la librería de “NodeMailer”, para poder realizar esta función.

Ahora en la ilustración 44, se indicará el código empleado en esa función.

```
router.post('/forgot', function(req, res, next) {
  async.waterfall([
    function(done) {
      crypto.randomBytes(5, function(err, buf) {
        var token = buf.toString('hex');
        done(err, token);
      });
    },
    function(token, done) {
      Usuario.findOne({ email: req.body.email }, function(err, user) {
        if (!user) {
          req.flash('SingupMessage', 'El correo no se encuentra registrado actualmente.');
```

```
          return res.redirect('/');
        }

        user.resetPasswordToken = token;
        user.resetPasswordExpires = Date.now() + 3600000; // 1 hour

        req.flash('Recuperacion', 'Se ha mandado un correo a su email , verifique que le haya llegado correctamente.');
```

```
        res.redirect('/');
        user.save(function(err) {
          done(err, token, user);
        });
      });
    },
  ],
```



```

function(token, user) {
  var mailOptions = {
    to: user.email,
    from: req.body.email,
    subject: 'Reseteo contraseña Lampgames',
    text: 'Se ha enviado este correo, ya que usted quiere resetear la contraseña de su cuenta de Lampgames.\n\n' +
      'Porfavor, si quiere cambiar la contraseña, se le tiene que dar click al siguiente enlace:\n\n' +
      'http://' + req.headers.host + '/reset/' + token + '\n\n' +
      'Si no quiere cambiar la contraseña, ignore este mensaje.\n\n' +
      'LampGames.'
  };
  transporter.sendMail(mailOptions, function(error, info){
    if (error) {
      console.log(error);
    } else {
      console.log('Email enviado: ' + info.response);
    }
  });
}
], function(err) {
  if (err) return next(err);
  res.redirect('/');
});
});

```

Ilustración 44: código de envío de correo electrónico, para resetear la contraseña con el enlace generado.

Y otro ejemplo de código sería el tiempo limitado de token, en el reseteo de contraseña. Se ha utilizado esta función en caso de que el usuario ignore el correo enviado a su email. Si este vuelve, después del tiempo puesto en el código, saltará un error, indicando de que el periodo de tiempo ha terminado.

En la ilustración 45, se muestra el código, de la función explicada previamente,

```

router.get('/reset/:token', function(req, res) {
  Usuario.findOne({ resetPasswordToken: req.params.token, resetPasswordExpires: { $gt: Date.now() } }, function(err, user) {
    if (!user) {
      req.flash('token_error', 'El reseteo de contraseña es invalido, y a que ha expirado.');
```

```

    res.render('reset', {token: req.params.token});
  });
});

```

Ilustración 45: código de la función, que indica que se ha pasado el tiempo de reseteo.

Y se han aplicado otras librerías, como es caso de Wow.js, que lo que hace, son las transiciones de imágenes, texto y cuadros que hayamos indicado, en cada una de ellas.

Más tarde, se indica el código del sector “Servicios” en la ilustración 47, ya que se aplica Bootstrap dentro de este sector, y hace animaciones que se han indicado de la librería Wow.js

En la ilustración 46, se muestra el sector de servicios.

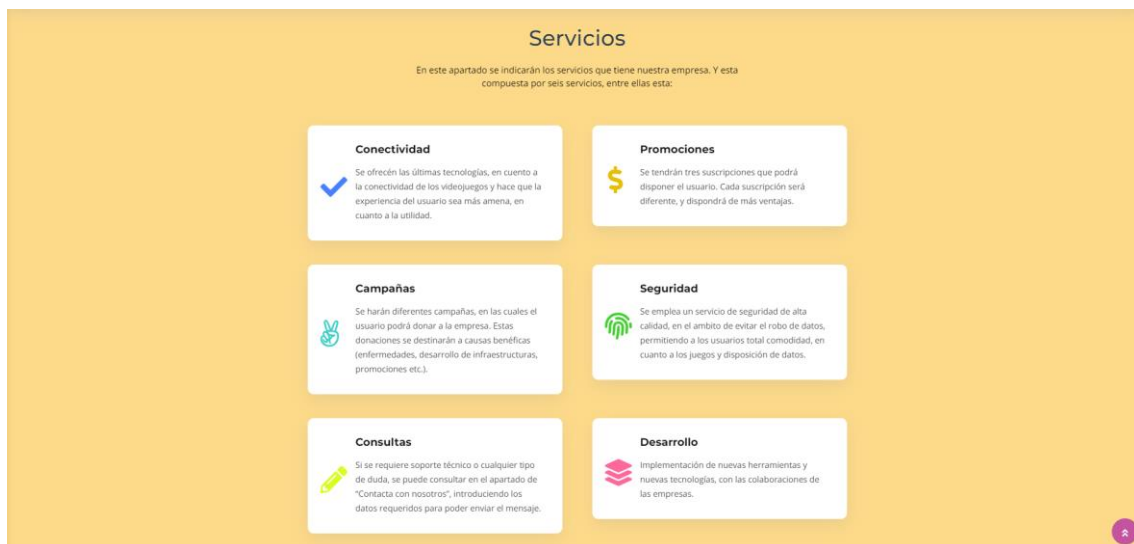


Ilustración 46: sector de servicios.

```

<section id="services" class="section-bg">
  <div class="container">

    <header class="section-header">
      <h3>Services</h3>
      <p>Laudem latine persequeris id sed, ex fabulas delectus quo. N
o vel partiendo abhorreant vituperatoribus.</p>
    </header>

    <div class="row">

      <div class="col-md-6 col-lg-5 offset-lg-1 wow bounceInUp" data-
wow-duration="1.4s">

```

```

        <div class="box">
            <div class="icon"><i class="fa fa-address-
book" style="color: #ff689b;"></i></div>
            <h4 class="title"><a href="">Lorem Ipsum</a></h4>
            <p class="description">Voluptatum deleniti atque corrupti q
uos dolores et quas molestias excepturi sint
                occaecati cupiditate non provident</p>
            </div>
        </div>
        <div class="col-md-6 col-lg-5 wow bounceInUp" data-wow-
duration="1.4s">
            <div class="box">
                <div class="icon"><i class="fab fa-
atlassian" style="color: #e9bf06;"></i></div>
                <h4 class="title"><a href="">Dolor Sitema</a></h4>
                <p class="description">Minim veniam, quis nostrud exercitat
ion ullamco laboris nisi ut aliquip ex ea
                    commodo consequat tarad limino ata</p>
                </div>
            </div>

            <div class="col-md-6 col-lg-5 offset-lg-1 wow bounceInUp" data-
wow-delay="0.1s" data-wow-duration="1.4s">
                <div class="box">
                    <div class="icon"><i class="fas fa-align-
left" style="color: #3fcdc7;"></i></div>
                    <h4 class="title"><a href="">Sed ut perspiciatis</a></h4>
                    <p class="description">Duis aute irure dolor in reprehender
it in voluptate velit esse cillum dolore eu
                        fugiat nulla pariatur</p>
                    </div>
                </div>
                <div class="col-md-6 col-lg-5 wow bounceInUp" data-wow-
delay="0.1s" data-wow-duration="1.4s">
                    <div class="box">
                        <div class="icon"><i class="fas fa-
book" style="color:#41cf2e;"></i></div>
                        <h4 class="title"><a href="">Magni Dolores</a></h4>
                        <p class="description">Excepteur sint occaecat cupidatat no
n proident, sunt in culpa qui officia deserunt
                            mollit anim id est laborum</p>
                        </div>
                    </div>

                    <div class="col-md-6 col-lg-5 offset-lg-1 wow bounceInUp" data-
wow-delay="0.2s" data-wow-duration="1.4s">
                        <div class="box">
                            <div class="icon"><i class="fab fa-
buffer" style="color: #d6ff22;"></i></div>

```

```

        <h4 class="title"><a href="">Nemo Enim</a></h4>
        <p class="description">At vero eos et accusamus et iusto od
io dignissimos ducimus qui blanditiis
        praesentium voluptatum deleniti atque</p>
    </div>
</div>
<div class="col-md-6 col-lg-5 wow bounceInUp" data-wow-
delay="0.2s" data-wow-duration="1.4s">
    <div class="box">
        <div class="icon"><i class="far fa-calendar-
check" style="color: #4680ff;"></i></div>
        <h4 class="title"><a href="">Eiusmod Tempor</a></h4>
        <p class="description">Et harum quidem rerum facilis est et
expedita distinctio. Nam libero tempore, cum
        soluta nobis est eligendi</p>
    </div>
</div>

</div>
</section>

```

Ilustración 47: código del sector indicado en la ilustración 46

6.2. ANDROID

En este apartado vamos a hablar del funcionamiento de la aplicación, que se ha desarrollado con Android Studio.

6.2.1. FUNCIONAMIENTO

La estructura de la aplicación, está formada por un Layout, donde se encuentra la parte visible de la aplicación.

En la ilustración 48, se puede visualizar el Layout de la aplicación de Android.

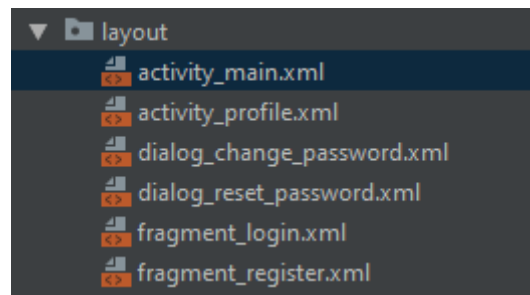


Ilustración 48: partes que tiene el Layout de la Aplicación.

Ahora se hablará de los distintos puntos que tiene la aplicación:

Primero está el Activity_Main, en este apartado se encuentran los distintos fragmentos, para poder iniciar la aplicación. Estos son el “Fragment_Login” y el “Fragment_register”.

En las ilustraciones 49 y 50, se muestran el fondo de arranque de la aplicación, y el código que lo ejecuta.



Ilustración 49: fondo de la Aplicación

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.TFG.MainActivity">

    <FrameLayout
        android:id="@+id/fragmentFrame"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>
</RelativeLayout>
```

Ilustración 50: código de ejecución de arranque.

A continuación, tendremos el “Activity_profile”, en este aparatado se ubicarán los datos del usuario, que se haya logueado dentro de la aplicación.

Y los campos que se van a mostrar del Administrador, tanto en el campo de email, como en la fecha de registro en un “RelativeLayout”.

A continuación, se mostrará la vista de datos del administrador (ilustración 51), y su código de ejecución (ilustración 52).

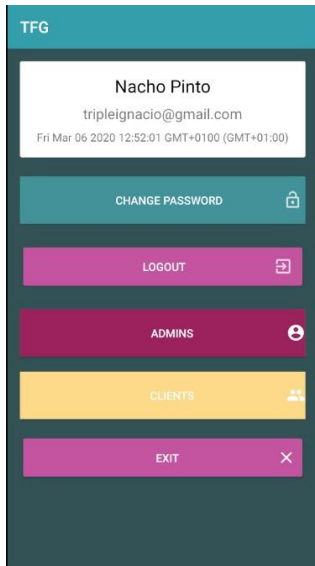


Ilustración 51: vista de los datos del Administrador, con las funciones que puede disponer.

```
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="16dp">

    <TextView
        android:id="@+id/tv_name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:textColor="@android:color/black"
        android:textAppearance="?android:attr/textAppearanceLarge"/>

    <TextView
        android:id="@+id/tv_email"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/tv_name"
        android:layout_marginTop="8dp"
        android:layout_centerHorizontal="true"
        android:textAppearance="?android:attr/textAppearanceMedium"/>

    <TextView
        android:id="@+id/tv_date"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/tv_email"
        android:layout_marginTop="8dp"
        android:layout_centerHorizontal="true"/>
</RelativeLayout>
```

Ilustración 52: código de ejecución de datos de la vista de administrador.

En el tercer apartado, se encuentra el apartado de “Dialog_change_password”, que este será el diálogo que aparecerá al pulsar en el botón de Change Password del Activity_profile, que se utilizará para modificar la contraseña del administrador que inició la sesión.

A continuación, se mostrará la vista de modificación de contraseña (ilustración 53), con su código correspondiente (ilustración 54).

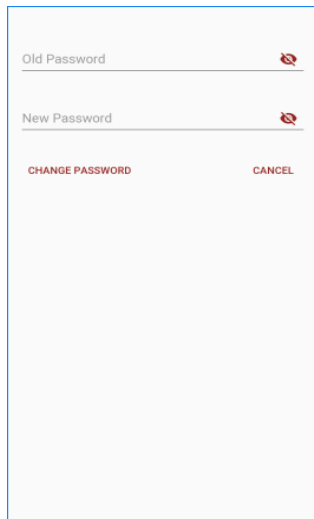


Ilustración 53: vista de modificación de contraseña.

```
<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/ti_old_password"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="24dp"
    app:passwordToggleEnabled="true"
    app:passwordToggleTint="@color/colorAccent">

    <EditText
        android:id="@+id/et_old_password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/old_password"
        android:inputType="textPassword"
        android:textColor="@color/colorAccent"/>

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/ti_new_password"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/ti_old_password"
    android:layout_marginTop="24dp"
    app:passwordToggleEnabled="true"
    app:passwordToggleTint="@color/colorAccent">

    <EditText
        android:id="@+id/et_new_password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/new_password"
        android:inputType="textPassword"
        android:textColor="@color/colorAccent"/>

</com.google.android.material.textfield.TextInputLayout>
```

Ilustración 54: código que hace la funcionalidad de modificación de contraseña.

Avanzamos con el apartado de “Dialog_reset_password”. Este se encargará de ofrecer la opción de recuperar la contraseña, en caso de que no se pueda acceder a la sesión

Primero se debe introducir el correo del usuario. Y una vez ingresado el correo, este se verificará si está en la base de datos, en caso de que se encuentre, se creará una ventana, en la cual se podrá introducir una nueva contraseña, y esta debe ser verificada otra vez, para que se pueda cambiar automáticamente.

A continuación, se mostrará la vista recuperación de contraseña (ilustración 54), con su código correspondiente (ilustración 55). Y después, con los campos de contraseña y nueva contraseña (ilustración 56) y su código (ilustración 57).

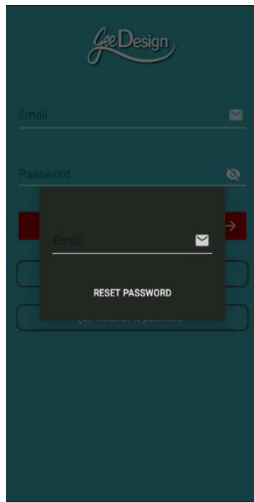


Ilustración 54: vista de petición de correo, para recuperar contraseña.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    android:background="@color/colorDarkGrey">

    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/ti_email"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="24dp">

        <EditText
            android:id="@+id/et_email"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Email"
            android:drawableRight="@drawable/ic_email"
            android:inputType="textEmailAddress"
            android:textColor="@color/colorAccent"/>

    </com.google.android.material.textfield.TextInputLayout>

    <Button
        android:id="@+id/btn_reset_password"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/ti_email"
        android:layout_centerHorizontal="true"
        android:text="Reset Password"
        android:layout_marginTop="24dp"
        style="@style/Widget.AppCompat.Button.Borderless.Colored"/>

</RelativeLayout>
```

Ilustración 55: código de petición de código, para recuperar la contraseña.

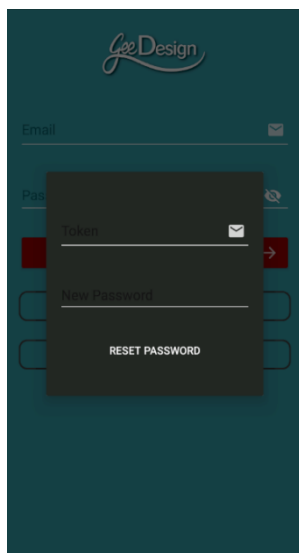


Ilustración 56: vista de recuperación de contraseña, con sus campos correspondientes.

```
<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/ti_token"
    android:visibility="gone"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/ti_email"
    android:layout_marginTop="24dp">

    <EditText
        android:id="@+id/et_token"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Token"
        android:drawableRight="@drawable/ic_email"
        android:inputType="textEmailAddress"
        android:textColor="@color/colorAccent"/>

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/ti_password"
    android:visibility="gone"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/ti_token"
    android:layout_marginTop="24dp"
    app:passwordToggleTint="@android:color/white">

    <EditText
        android:id="@+id/et_password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="New Password"
        android:inputType="textPassword"
        android:textColor="@color/colorAccent"/>

</com.google.android.material.textfield.TextInputLayout>
```

Ilustración 57: código de la vista de recuperación de contraseña.

Más tarde se llega al “Fragment_Login”, en esta vista se percibe los campos para poder iniciar la sesión y acceder a los datos. Además, se muestra dos botones, uno es para el registro, y el segundo para la recuperación de la contraseña.

La progress bar se hará visible una vez que se pulse en el botón login, en caso de que no se tenga usuario, se pulsará en registrar nuevo usuario y te llevaría a el Fragment_register.

Y ahora se mostrará dos imágenes, uno con la vista del login (ilustración 58) y otro con su código correspondiente (ilustración 59).

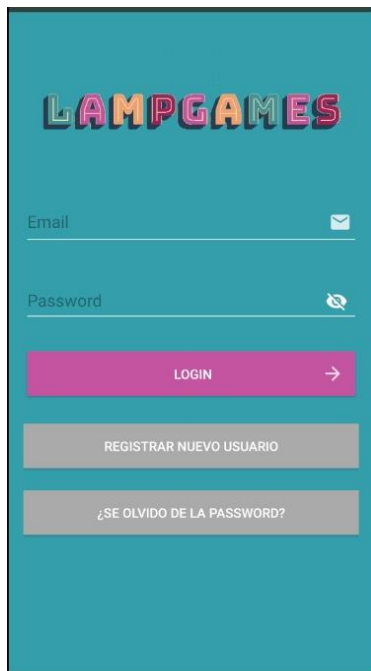


Ilustración 58: vista del login de la aplicación.

Ilustración 59: código de la vista del login de la aplicación.

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <RelativeLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_margin="16dp">

        <ImageView
            android:id="@+id/img_logo"
            android:layout_width="wrap_content"
            android:layout_height="80dp"
            android:layout_centerHorizontal="true"
            android:src="@drawable/cabecera"/>

        <com.google.android.material.textfield.TextInputLayout
            android:id="@+id/ti_name"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_below="@+id/img_logo"
            android:layout_marginTop="24dp">

            <EditText
                android:id="@+id/et_name"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:drawableRight="@drawable/ic_person"
                android:hint="Name"
                android:inputType="text" />

        </com.google.android.material.textfield.TextInputLayout>

    </RelativeLayout>

</LinearLayout>
```

También está el apartado de registro, que este se llama “Fragment_register”, en este se pedirán tres campos de formulario, habrá un botón de registro y otro botón para volver al inicio de sesión, en caso de que se tenga ya una cuenta ya registrada.

A continuación, se muestra la vista de registro (ilustración 60) y su código (ilustración 61).

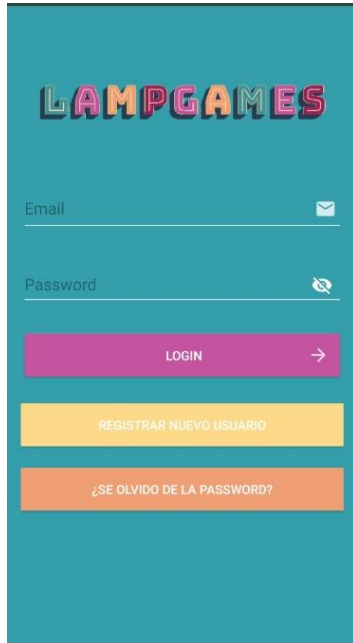


Ilustración 60: vista de registro de sesión de la aplicación.

Ilustración 61: código de la vista de registro de sesión

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <RelativeLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_margin="16dp">

        <ImageView
            android:id="@+id/img_logo"
            android:layout_width="wrap_content"
            android:layout_height="80dp"
            android:layout_centerHorizontal="true"
            android:src="@drawable/cabecera"/>

        <com.google.android.material.textfield.TextInputLayout
            android:id="@+id/ti_name"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_below="@+id/img_logo"
            android:layout_marginTop="24dp">

            <EditText
                android:id="@+id/et_name"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:drawableRight="@drawable/ic_person"
                android:hint="Name"
                android:inputType="text" />

        </com.google.android.material.textfield.TextInputLayout>

    </RelativeLayout>

</LinearLayout>
```

Luego esta el apartado de perfiles, en las cuales se van a visualizar los usuarios que están registrados en la página web.

Se hace un arraylist, para coger todos los datos y se plasman en un listview. Y a través del compositeSubscription hace que los valores que se muestran del perfil tengan sentido, y de que se puedan consultar a través de un contenedor.

A continuación, se va a mostrar el código (ilustración 62) con la vista correspondiente (ilustración 63).

```
androidManifest.xml Users.java Datos.java DatosC.java activity_profile.xml Client.java ProfileActivity.java
public class Clientes extends AppCompatActivity {
    public static final String TAG = Users.class.getSimpleName();

    private CompositeSubscription mSubscriptions;
    ListView listView;
    ArrayList<DatosC> lista;

    //Inflamos la vista accedemos a los datos de nuestra JSON de modo mediante la BASE_URL y los añadimos a la vista

    @Override
    @Override
    public void onCreate(Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.userlist, container, attachToRoot: false);
        mSubscriptions = new CompositeSubscription();
        listView = (ListView) view.findViewById(R.id.listView);
        Retrofit retrofit = new Retrofit.Builder()
            .baseUrl(Constants.BASE_URL)
            .addConverterFactory(GsonConverterFactory.create())
            .build();

        RetrofitInterface retrofitInterface = retrofit.create(RetrofitInterface.class);

        Call<List<Client>> call = retrofitInterface.getProfileClientes();

        call.enqueue(new Callback<List<Client>>() {

            //Si la respuesta al request los datos es positiva entonces creamos una lista con los datos y cada dato

            @Override
            public void onResponse(Call<List<Client>> call, Response<List<Client>> response) {
                List<Client> client = response.body();

                String[] clienteNames = new String[client.size()];
                String[] clienteNick = new String[client.size()];
                String[] clienteEmail = new String[client.size()];
            }
        });
    }
}
```

Ilustración 62: código de la función explicada previamente.

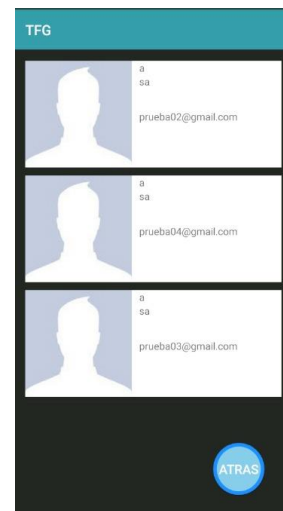


Ilustración 63: vista de los usuarios, repartidos en contenedores, con sus datos correspondientes.

Y por último esta la vista, para confirmar la eliminación de uno de los clientes o usuarios, en las que se haya presionado.

Para ello habrá un adaptador, en la cual se reflejará la vista, con el texto y los botones de borrar o no borrar.

A continuación, se va a mostrar dos ilustraciones, en las cuales una se va a mostrar el código empleado, para esta vista (ilustración 64) y luego esta la vista móvil, con las funciones explicadas previamente (ilustración 65).

```

wordDialog.java x Users.java x Datos.java x DatosCl.java x activity_profile.xml x Clients.java x ProfileActivity.java x AdapterClients.java x RetrofitInterface.java x Client.java x

String[] clientNames = new String[client.size()];
String[] clientNick = new String[client.size()];
String[] clientEmails = new String[client.size()];

Lista = new ArrayList<DatosCl>();

for(int i = 0; i<client.size(); i++){
    //clientsImages[i] = client.get(i).getImagen();
    Lista.add(new DatosCl( id= 1,R.drawable.defecto,clientNames[i] = client.get(i).getNombre(),clientNick[i] = client.get(i).getUsuario(),clientEmails[i] = client.get(i).getEmail()));
}

AdapterClients miadaptador = new AdapterClients(getActivity().getApplicationContext(),Lista);

listaDatos.setAdapter(miadaptador);

//Creamos el dialogo de confirmacion de eliminacion si es una respuesta positiva se borra de la lista el cliente seleccionado

listaDatos.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        //Toast.makeText(parent.getContext(), "View clicked: " + client.get(position).getEmail(), Toast.LENGTH_SHORT).show();
        AlertDialog.Builder alertDialog = new AlertDialog.Builder(getActivity(), R.style.AppCompatAlertDialogStyle);
        alertDialog.setMessage("¿Desea eliminar al cliente "+client.get(position).getEmail()+"?");
        .setCancelable(false) // Aquí defines que no se pueda dar atrás (cancelar el dialog) el usuario debiera seleccionar una opcion
        .setPositiveButton("Si", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                Lista.remove(position);
                miadaptador.notifyDataSetChanged();
                //Toast.makeText(getActivity(), "Cliente eliminado con éxito", Toast.LENGTH_SHORT).show();
                Call<com.TFG.model.Response> call = retrofitInterface.deleteProfile(client.get(position).getEmail());
                //Call<com.TFG.model.Response> call = retrofitInterface.deleteProfile(client.get(position).getId());
                call.enqueue(new Callback<com.TFG.model.Response>() {

```

Ilustración 64: Código de la vista de eliminar usuarios.

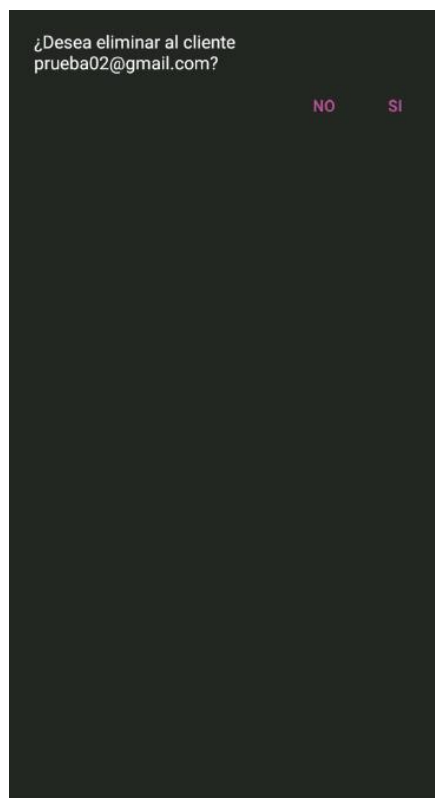


Ilustración 65: Vista para borrar al cliente, que se haya seleccionado.

6.3. JUEGO EN UNITY

Vamos a hablar de los distintos puntos que tiene el juego, desarrollado con la máquina de Unity.

6.3.1. MAPA

INTRODUCCIÓN Y CUBO

Primero se debe crear la escena para visualizar el proyecto que se vaya a desarrollar. En este caso, el proyecto va a estar desarrollado en 3D.

Cuando se tenga la escena creada, creamos un cubo, de las cuales se tienen que dar unas propiedades. Cuando se tengan, se clonan para formar el mapa.

En la ilustración 66, se muestra la base de cómo va a ser el cubo, con sus características correspondientes.

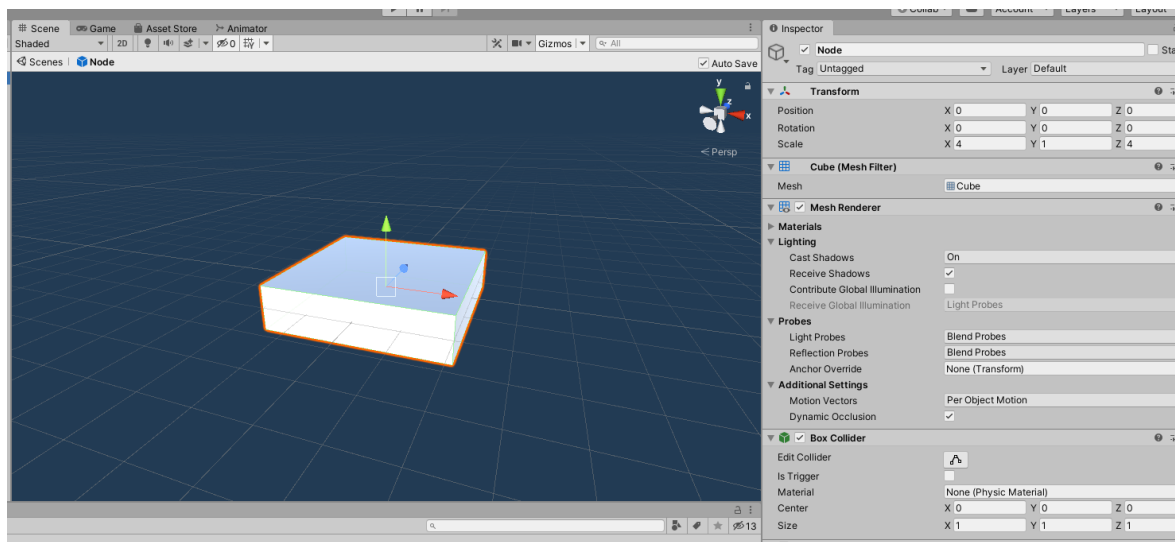


Ilustración 66: vista del cubo, que va a formar el mapa. Las características del cubo se ven a la derecha de la imagen.

WAYPOINTS

También hay otra función, que son los Waypoints. Estos son puntos, por las cuales pasa el enemigo, hasta llegar a su punto final, que será destruir el punto aliado.

Este contiene una función, para que el enemigo se pueda guiar, a través de estos puntos. En las siguientes ilustraciones, se ve que es el Waypoint (ilustración 67), con su código correspondiente (ilustración 64).



Ilustración 67: Imagen del Waypoint.

```
using UnityEngine;

public class Waypoints : MonoBehaviour
{
    public static Transform[] points;

    void Awake()
    {
        points = new Transform[transform.childCount];
        for (int i = 0; i < points.Length; i++)
        {
            points[i] = transform.GetChild(i);
        }
    }
}
```

Ilustración 64: Función de los Waypoints.

MAIN CAMERA

También se usa una herramienta, que viene por defecto en el motor gráfico y se trata del “Main Camera”.

Se encarga de visualizar la vista que envuelve al jugador, dentro del propio juego. Se proporciona una vista de la escena, desde la posición que se le haya indicado.

Todas las escenas que se han creado dentro del proyecto, deberán contener como mínimo una cámara. Se pueden implementar cámaras ilimitadas y por defecto, Unity te crea una cámara por escena.

Este componente se le puede asignar muchas características, entre las más utilizadas esta la distancia en la que se va a visualizar y esto también afecta en la rotación del propio componente, también se indican los elementos que se puedan eliminar y establecer el ángulo de visión, entre las características más destacadas y utilizadas en el proyecto (todas las características que modifiquen la posición, serán influidos por los ejes “x”, “y” y “z” se utilizan los vectores.).

A continuación, se visualizará la imagen de la cámara, en la ilustración 68 y las características generales, que se pueden modificar de la cámara creada en la escena, en la ilustración 69.

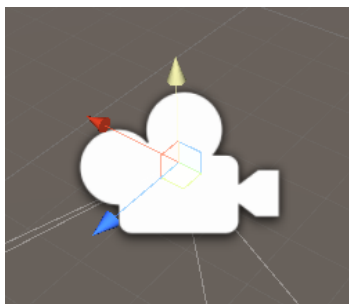


Ilustración 68: imagen de la cámara con los ejes indicados. El eje rojo es “x”, el eje amarillo es “y” y por último el eje azul, que es el “z”.

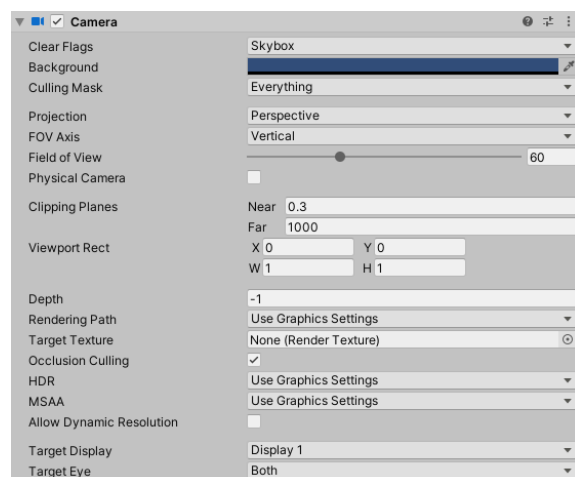


Ilustración 69: ajustes que vienen por defecto a la herramienta.

EVENTSYSTEM

Con esta herramienta, se mandan eventos a objetos que estén dentro de la aplicación, es decir, se puede interactuar a través de periféricos que estén conectados, como se puede apreciar en un ratón o teclado, pulsando sobre el botón de los mismos.

Esta acción se aplica a un objeto del juego (GameObject), que, aunque no se aprecie mucho funcionamiento, está diseñado para el administrador, y este puede poner los eventos dentro del EventSystem.

Las funciones que se aplican a esta herramienta son:

- Poder seleccionar los objetos de juego (GameObject) cuando el juego este en ejecución.
- Controlar inputs y Raycasters (averiguar donde acabo el puntero del ratón).
- Actualizar los inputs (en el caso de que se requiera).

En la ilustración número 70, se mostrarán las opciones que se están aplicando dentro de la herramienta.

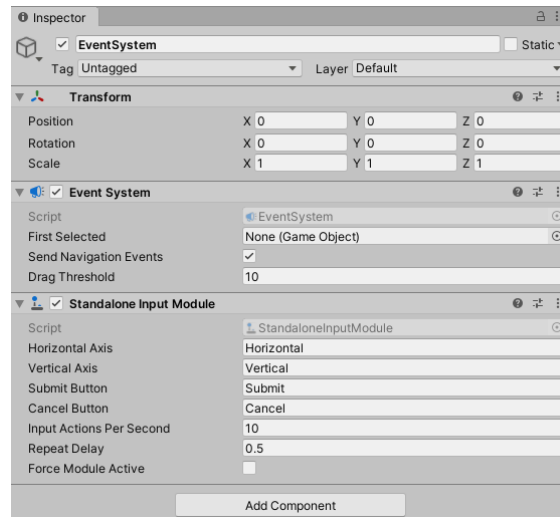


Ilustración 70: opciones que se tienen puestas en la herramienta de "EventSystem".

DIRECTIONAL LIGHT

Herramienta capaz de simular el direccionamiento de la luz del sol. Con solamente poner una, se puede iluminar toda la escena y se crean las sombras visualizadas desde un objeto.

En la ilustración 71 se muestra la herramienta indicada previamente.

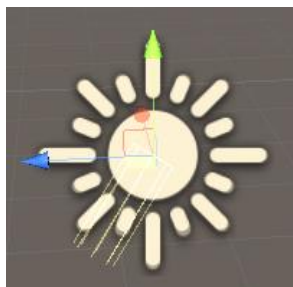


Ilustración 71: herramienta "Directional Light".

GENERADOR DE ENEMIGOS

Es la zona donde se generan los enemigos. En este punto se relacionan dos scripts, de las cuales uno genera los enemigos y el otro realiza la acción de abrir y cerrar la puerta, por donde salen los enemigos.

En la función de generar enemigos, se indican todos los enemigos que se generen, con un tiempo de reaparición de enemigos. También se aprecia la cantidad de enemigos que aparecen, para realizar el intercambio con el otro enemigo asignado.

Además, el tiempo entre un enemigo y otro es al azar, con una cantidad mínima y máxima permitida.

En la siguiente ilustración, se puede ver el código de ejecución (ilustración 72 e ilustración 73)

```
public class InvokeRepeating : MonoBehaviour
{
    public GameObject enemyPrefab1;
    public GameObject enemyPrefab2;
    public GameObject ground;

    public float timer = 0f;
    private float timer2 = 0f;

    void Start()
    {
        timer2 = 0f;
    }

    void Update()
    {
        timer2 += Time.deltaTime;
        DecrementTimer();

        if (timer <= 0)
        {
            SpawnEnemy();
            timer = RandomTimer();
        }

    }

    void DecrementTimer()
    {
        timer -= Time.deltaTime;
    }

    void SpawnEnemy()
    {
        if (timer2 < 10)
        {
            Instantiate(enemyPrefab2, transform.position, Quaternion.Euler(0.0f, -90.0f, 0.0f));
        }
        if (timer2 > 10)
        {
            Instantiate(enemyPrefab1, transform.position, Quaternion.Euler(0.0f, -90.0f, 0.0f));
        }
    }

    float RandomTimer()
    {
        return Random.Range(0.3f, 1f);
    }
}
```

Ilustración 72: primera parte de la función de generar enemigos. Generador de enemigos, con relación al tiempo de ejecución y el tiempo en la que reaparecen los enemigos.

```
float RandomTimer()
{
    return Random.Range(0.3f, 1f);
}

Vector3 RandomSide()
{
    int test = Random.Range(1, 2);
    if (test == 1)
    {
        return new Vector3(ground.transform.lossyScale.x / 2, 0f);
    }
    else
    {
        return new Vector3(-ground.transform.lossyScale.x / 2, 0f);
    }
}
```

Ilustración 73: segunda parte de la función de generar enemigos. Tiempo de reaparición entre un enemigo y otro.

Y por último esta la función de abrir y cerrar la puerta, donde salen los enemigos.

A continuación, se indicará el código (ilustración 74) y como quedaría la plataforma, donde salen todos los enemigos (ilustración 75).

```

public class nvpDoor_scr : MonoBehaviour {

    [SerializeField] private bool _open;
    [SerializeField] private bool _opened;
    [SerializeField] private Animator _doorAnimator;
    public GameObject doorIndicator;
    private float timer;

    void Start () {
        timer = 0.0f;
        var animationStateBehaviours = _doorAnimator.GetBehaviours<nvpDoorStateBehaviour_scr>();
        foreach(var beh in animationStateBehaviours){
            beh.doorStateIndicatorRenderer = doorIndicator.GetComponent<Renderer>();
        }
    }

    void Update () {
        timer += Time.deltaTime;
        if(!_open != _opened){
            _opened = _open;
            _doorAnimator.SetBool("open", _open);
        }
    }

    void OnTriggerEnter(Collider other)
    {
        if(other.tag == "Enemy2" || other.tag == "Enemy")
        {
            _open = true;
        }
    }

    void OnTriggerExit(Collider other){
        if(other.tag == "Enemy2" || other.tag == "Enemy")
        {
            _open = false;
        }
    }
}

```

Ilustración 72: Código de ejecución de la función de abrir y cerrar la puerta.

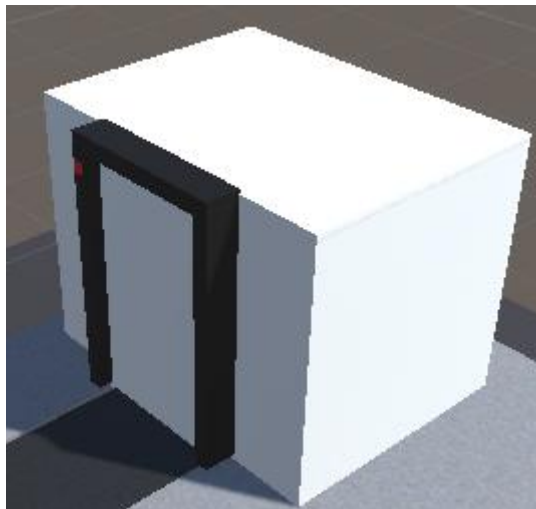


Ilustración 73: Base enemiga.

CANVAS

Es un apartado importante, ya que con esto podemos ver varios datos que se disponen, a la hora de ejecutar el juego.

Se podrá ver dos barras, una de color verde, que corresponde a la vida que tiene el cohete, que en el caso de que colisione un enemigo en él, se reducirá la vida del cohete y se empezaría a rellenar de color rojo (ilustración 76).

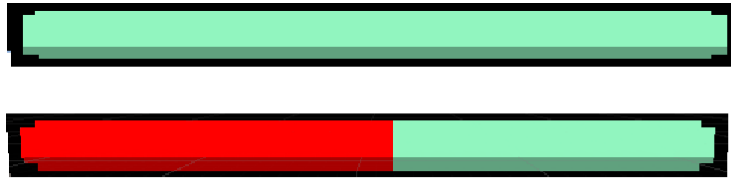


Ilustración 74: Aquí se muestran dos barras, la primera empieza con la vida completa, y en caso de que se choque un enemigo, se podría pasar la segunda barra.

Luego estará otra segunda barra, que es la fase del nivel. Esta barra se relaciona con el tiempo y cada vez que va pasando unos segundos, se ira rellenando la barra. En caso de que se complete, se habrá ganado y pasarás a la siguiente escena, para continuar con el siguiente nivel.

En la ilustración 77, se mostrará la barra vacía (comienzo de nivel) y una casi llena (a punto de terminar el nivel).



Ilustración 75: aquí se muestran dos barras, una que está vacío (empieza la partida) y otra a punto de llenarse (a punto de acabar el nivel).

Después esta la puntuación y el tiempo que tardas en superarte el nivel.

La puntuación se utiliza principalmente para colocar nuevas armas, que requieren unos puntos determinados. Se pueden poner una cierta cantidad de armas y a la hora de terminar el nivel, se resetea el tiempo y la puntuación vuelve a estar como al principio del inicio de nivel.

Se puede elegir entre tres armas, que valdrán una puntuación diferente y se podrán colocar en cualquier punto del mapa, excepto en el camino por donde pasan los enemigos, ni por el plano del mapa (los bordes del mapa.)

A continuación, se mostrará el inicio del juego, con la perspectiva del jugador y con el menú obtenido en la ilustración número 78.

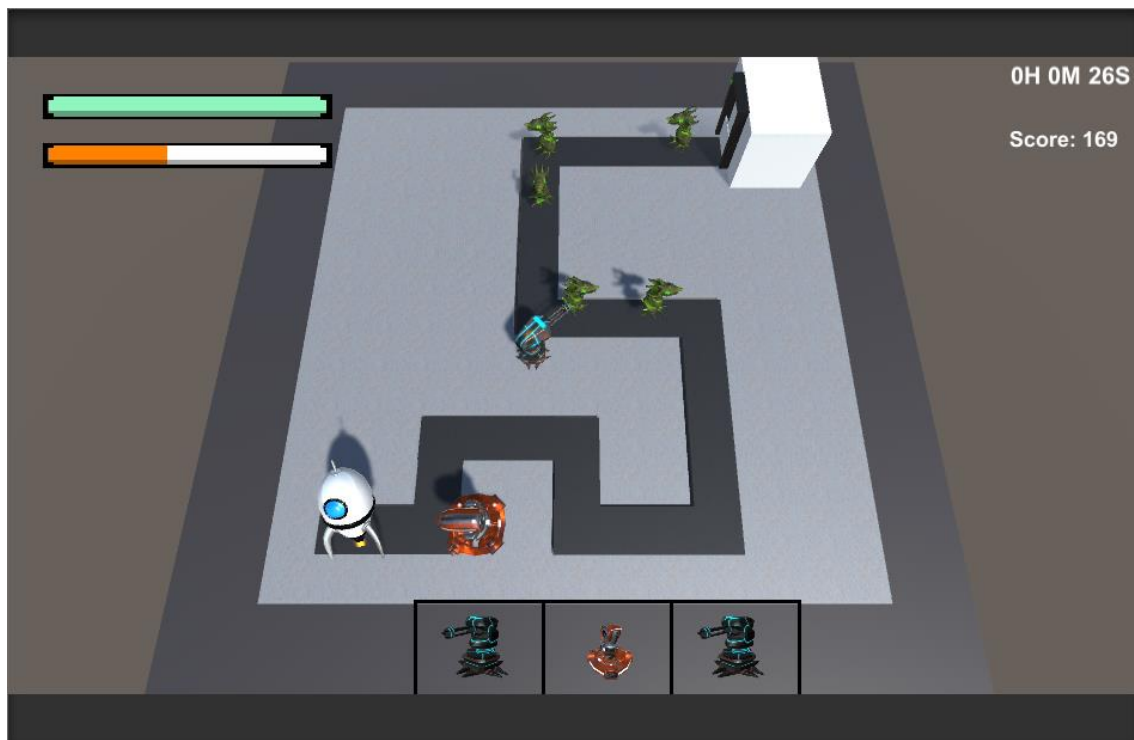


Ilustración 76: Primer nivel del mapa, con el menú, los enemigos y las bases.

Y para poder obtener las armas, canjearlos por los puntos que se tengan y ponerlos en la casilla correspondiente, se necesitará dos funciones, función para interactuar con el objeto (ilustración 79) y la función de comprar (ilustración 80).

```
public class Manager : MonoBehaviour
{
    public static Manager instance;

    void Awake()
    {
        if (instance != null)
        {
            Debug.LogError("More than one manager in scene!");
            return;
        }
        instance = this;
    }

    public GameObject standardTurretPrefab;
    public GameObject anotherTurretPrefab;
    private Prices turretToBuild;

    public bool CanBuild { get { return turretToBuild != null; } }

    public void BuildTurretOn(Node node)
    {
        if (points.Score < turretToBuild.cost)
        {
            Debug.Log("No tienes suficiente dinero");
            return;
        }

        points.Score -= turretToBuild.cost;
        GameObject turret = (GameObject)Instantiate(turretToBuild.prefab, node.GetBuildPosition(), Quaternion.identity);
        node.turret = turret;
        Debug.Log("Torre comprada correctamente");
    }

    public void SelectTurretToBuild (Prices turret)
    {
        turretToBuild = turret;
    }
}
```

Ilustración 79: función para interactuar con la torreta, para ponerla en el mapa y se hace la acción de comprar la torreta.

```
public class shop : MonoBehaviour
{
    public Prices torrent1;
    public Prices torrent2;

    Manager manager;

    void Start()
    {
        manager = Manager.instance;
    }

    public void SelectStandardTorrent()
    {
        Debug.Log("Seleccionada la torreta1");
        manager.SelectTurretToBuild(torrent1);
    }

    public void SelectAnothrTorrent()
    {
        Debug.Log("Seleccionada la torreta2");
        manager.SelectTurretToBuild(torrent2);
    }
}
```

Ilustración 80: función para la selección del mapa

ATOMROCKET

Es el punto final de recorrido de enemigos, se trata de una nave espacial que tiene una vida determinada. Si hay un enemigo que se choca sobre este, perderá vida. En caso de haberle quitado toda la vida, el juego terminaría y saltaría a la siguiente escena.

Este compuesto por dos funciones:

La primera es la función de "Hearth". Es la vida que tiene el cohete y que este está conectado a una de las barras de apartado de "Canvas" (la barra de vida).

Y después esta la función de colisión. Esta función se pone en marcha cuando un enemigo impacta sobre la nave espacial. Dependiendo del enemigo, puede quitar más o menos. Y además se ha incluido un efecto a la nave, en la cual sale fuego en el motor de arranque.

Ahora se va indicar el código, el primero es el de vida (ilustración 81 y 82) y el segundo se trata de la colisión (ilustración 83). Además, se mostrará como es el cohete (ilustración 84)

```
public class Health : MonoBehaviour
{
    public int maxHealth = 100;
    public int curHealth = 0;
    public bool alive=true;
    //public GameObject enemy;
    public healthBar healthBar;

    public int maxfuel = 100;
    public int curfuel = 0;
    public healthBar fuel;
    private float timer;
    public GameObject fuego;

    void Start()
    {
        alive = true;
        curHealth = maxHealth;
        curfuel = maxfuel;
    }

    public void DoDamage()
    {
        TakeDamage(20);
    }

    void Update()
    {
        timer += Time.deltaTime;
        if (Input.GetKeyDown(KeyCode.Space))
        {
            TakeDamage(20);
        }
        if (curHealth<=0)
        {
            SceneManager.LoadScene("GameOver");
        }
        if (curfuel <= 0)
        {
            SceneManager.LoadScene("Survived");
        }
        if (timer>3)
        {
            UpdateFuel(5);
            timer = 0;
        }
        if (curHealth<=60)
        {
            Instantiate(fuego, new Vector3(-36, 13, -40), transform.rotation);
        }
    }
}
```

Ilustración 81: primera parte de la funcionalidad de vida. (Secciones y factores de daño).

```
public void TakeDamage(int amount)
{
    if (!alive)
    {
        return;
    }

    if (curHealth<=0)
    {
        curHealth = 0;
        alive = false;
    }

    curHealth -= amount;
    healthBar.SetHealth(curHealth);
}

public void UpdateFuel(int full)
{
    if (curfuel <= 0)
    {
        curfuel = 0;
    }
    curfuel -= full;
    fuel.SetHealth(curfuel);
}
```

Ilustración 82: segunda parte de la funcionalidad de vida.

```

public class CollisionBola : MonoBehaviour
{
    private void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.tag=="Enemy")
        {
            gameObject.GetComponent<Health>().TakeDamage(20);
            Destroy(GameObject.FindWithTag("Enemy"));
        }
        if (other.gameObject.tag == "Enemy2")
        {
            gameObject.GetComponent<Health>().TakeDamage(50);
            Destroy(GameObject.FindWithTag("Enemy2"));
        }
    }
}

```

Ilustración 83: función de colisión, con dos ejemplos: enemigo1 quita 20 y el enemigo2 quita 50



Ilustración 84: imagen de cómo es la nave espacial, que es el punto que hay que defender.

6.3.2. ENEMIGOS Y TORRETAS

Luego tenemos los apartados de enemigos y torretas [58] [59] [60]. Los enemigos tienen que llegar a la nave espacial. El enemigo se colisiona con la nave espacial, y este le quita vida a la nave. Y con las torretas, se puede derribar al enemigo, canjeando una serie de puntos. Si no se tiene ninguna puntuación, no se conseguirán puntos y siempre en cada partida se empieza con 50 puntos.

Las figuras de los enemigos se almacenan dentro de la carpeta de "Prefabs". Ahí se guardarán las figuras de los enemigos (carpeta "Enemy").

En las siguientes ilustraciones, se puede apreciar una de las figuras de los enemigos (ilustración 85) con los movimientos programados, lo que hace que el enemigo se mueva con realismo (ilustración 86).



Ilustración 85: figura de un enemigo

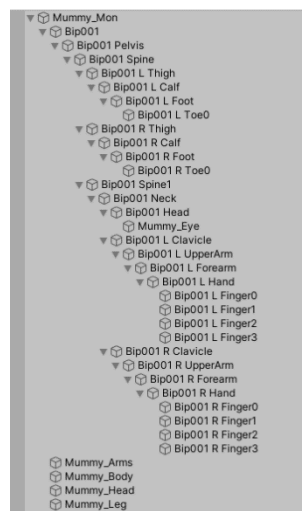


Ilustración 86: los movimientos que puede realizar este enemigo.

Los enemigos siguen los waypoints, que se hayan indicado previamente en el mapa, sin embargo, este parte desde un punto de partida, es decir, se inicializa desde cero.

Después se crearán los vectores, para poder inicializar el punto de inicio del enemigo.

Además, tienen vida y en el caso de que la torreta le quite toda la vida, este será eliminado.

A continuación, se indicará la función que tiene el enemigo, para realizar todas las funciones que se han indicado previamente (ilustración 87).

```
using UnityEngine;
using System.Collections.Generic;
using UnityEngine.SceneManagement;

public class Enemy : MonoBehaviour
{
    public float startSpeed = 10f;
    private Transform target;
    private int wavepointIndex = 0;
    public int damage = 20;

    void Start()
    {
        target = Waypoints.points[0];
    }

    void Update()
    {
        Vector3 dir = target.position - transform.position;

        transform.Translate(dir.normalized * startSpeed * Time.deltaTime, Space.World);
        transform.rotation = Quaternion.LookRotation(dir);
        if (Vector3.Distance(transform.position, target.position) <= 0.4f)
        {
            GetNextWaypoint();
        }
    }

    public void GetNextWaypoint()
    {
        if (wavepointIndex >= Waypoints.points.Length - 1)
        {
            Destroy(gameObject);
            return;
        }
        wavepointIndex++;
        target = Waypoints.points[wavepointIndex];
    }
}
```

Ilustración 87: función que tiene el enemigo.

Y luego se tendrán las torretas. Las torretas también tienen modelos que están almacenados dentro de la carpeta Prefabs, y dentro de estas se pueden aplicar las funciones.

A continuación, se va a mostrar un modelo de torreta (ilustración 88), que tiene animaciones (ilustración 89) y se puede configurar el movimiento que puede ejercer dentro del propio juego.

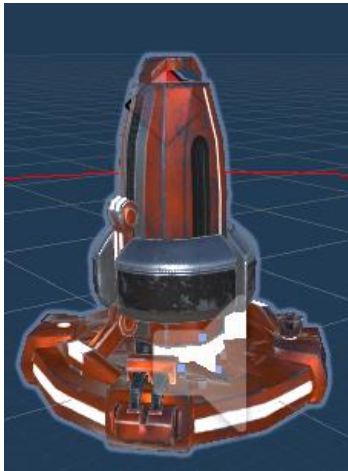


Ilustración 88: una de las torretas que se pueden poner en el juego.

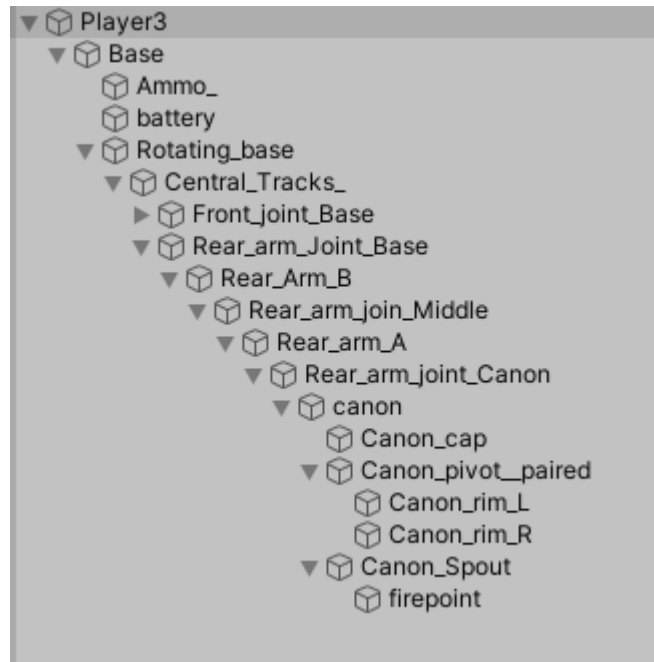


Ilustración 89: Movimientos que tiene la torreta.

En el caso de las torretas, se indica el daño, la velocidad de la bala, la rotación de la torreta y el rango que puede tener la torreta, al detectar al enemigo (se trata de la distancia detectada entre este y el enemigo).

Las torretas tienen dos funciones, una es las funciones que se han explicado previamente de la torreta (ilustración 90 e ilustración 91) y una función, que dependiendo del enemigo que se haya eliminado, dará tantos puntos, como se haya indicado en el script (ilustración 92).

```
public class Torretas : MonoBehaviour
{
    public float range = 15f;
    public float turnSpeed = 10f;
    public GameObject bulletPrefab;
    public Transform Rotate;
    public Transform firepoint;

    private float fireRate = 2f;
    private string enemyTag = "Enemy";
    private string enemyTag2 = "Enemy2";
    private Transform target;

    private Time countHit;
    private AudioSource mAudioSrc;
    private float fireCountdown = 0f;

    void Start()
    {
        InvokeRepeating("UpdateTarget", 0f, 0.5f);
        mAudioSrc = GetComponent();
    }

    void UpdateTarget()
    {
        GameObject[] enemies = GameObject.FindGameObjectsWithTag(enemyTag);
        GameObject[] enemies2 = GameObject.FindGameObjectsWithTag(enemyTag2);

        float shortestDistance = Mathf.Infinity;
        GameObject nearestEnemy = null;
        foreach (GameObject enemy in enemies)
        {
            float distanceToEnemy = Vector3.Distance(transform.position, enemy.transform.position);
            if (distanceToEnemy < shortestDistance)
            {
                shortestDistance = distanceToEnemy;
                nearestEnemy = enemy;
            }
        }
        foreach (GameObject enemy in enemies2)
        {
            float distanceToEnemy = Vector3.Distance(transform.position, enemy.transform.position);
            if (distanceToEnemy < shortestDistance)
            {
                shortestDistance = distanceToEnemy;
                nearestEnemy = enemy;
            }
        }
        if (nearestEnemy != null && shortestDistance <= range)
        {
            target = nearestEnemy.transform;
        }
        else
        {
            target = null;
        }
    }
}
```

Ilustración 90: primera parte que tiene la función de la torreta (posición, distancia etc.).


```

void Update()
{
    if (target == null)
    {
        return;
    }
    else
    {
        Rotate.LookAt(new Vector3(target.position.x, Rotate.position.y, target.position.z));
        if (fireCountdown <= 0f)
        {
            Shoot();
            fireCountdown = 1f / fireRate;
        }
        fireCountdown -= Time.deltaTime;
    }
}

void Shoot()
{
    GameObject bulletGO = (GameObject)Instantiate(bulletPrefab, firePoint.position, firePoint.rotation);
    Bullet bullet = bulletGO.GetComponent<Bullet>();

    if (bullet != null)
    {
        mAudioSrc.Play();
        bullet.Seek(target);
    }
}

void OnDrawGizmosSelected()
{
    Gizmos.color = Color.red;
    Gizmos.DrawWireSphere(transform.position, range);
}

```

Ilustración 91: segunda parte de la funcionalidad de la torreta (audio, bala etc.).

```

public class points : MonoBehaviour
{
    private Text scoreText;
    private float timer;
    public static int Score = 50;

    void Start()
    {
        scoreText = GameObject.FindGameObjectWithTag("Score").GetComponent<Text>();
    }

    void Update()
    {
        scoreText.text = "Score: " + Score.ToString();
        timer += Time.deltaTime;
        if (timer > 5f)
        {
            points.Score += 2;
            timer = 0;
        }
    }
}

```

Ilustración 92: función para la obtención de puntos.

6.3.3. ESCENAS

Ahora se van a describir las escenas adicionales, que se tienen en el juego. Está constituido por siete escenas, de las cuales, tres corresponden a tres niveles desarrollados, tres de fases cuando se completa el nivel correctamente y una de “Game Over”.

Vamos a describir como son las escenas de “nivel superado” y de “Game Over”, ya que uno de los niveles se describe todos los puntos de un nivel de un mapa, en el apartado 6.3.1.

Ahora se va a hablar de estas dos escenas. Para que se cumple el de “nivel superado”, se tendrá que aguantar en un nivel, para que se reproduzca la escena (es importante la nave espacial, ya que determina la escena que pueda pasar en ese momento, la función se puede ver en la ilustración 81).

Al salir esta escena, se indicará que se ha superado el nivel y se tendrá dos opciones, una que es para quitar el juego, y el otro es para continuar con el siguiente nivel.

Ahora se va a mostrar cómo es una de las escenas, cuando se sobrevive a un nivel, en la ilustración 93.

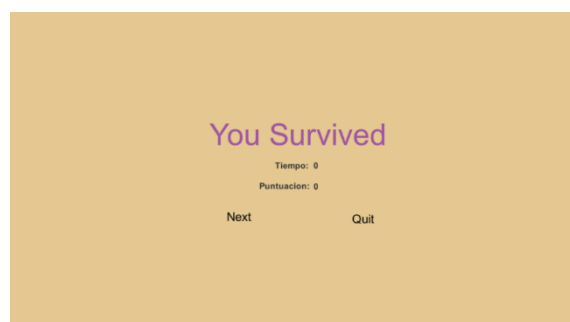


Ilustración 93: pantalla de haber sobrevivido a un nivel.

Y ahora se va a indicar las funciones que se emplean dentro de la ilustración 93, en ella está la función para continuar al siguiente nivel (ilustración 94) y otra para quitar la aplicación (ilustración 95).

```
public class ChangeLevel : MonoBehaviour
{
    public void ChangLevel()
    {
        SceneManager.LoadScene("SeconLevel");
        points.Score = 50;
    }
}
```

Ilustración 94: función para cambiar la escena, se indica la escena que se vaya a cambiar y se vincula con el botón, donde se va a utilizar esta función.

```
public class exit : MonoBehaviour
{
    // Start is called before the first frame update
    public void Exit()
    {
        Application.Quit();
    }
}
```

Ilustración 95: función para salir del juego (está relacionado con un botón).

Y después se hablará de la pantalla de “Game Over”. Esta escena se ejecutará cuando el enemigo quite toda la vida de la nave espacial. Cuando se llegue a este caso, saltará la ilustración 96.



Ilustración 96: escena de Game Over.

Se tendrá dos funciones, una se reutilizará de la anterior escena, que se trata de cerrar el juego. Y luego estará el de reiniciar, que lo que hará es reiniciar el juego, hasta el nivel 1. Eso quiere decir, que, si se está en el nivel 3 y se pierde, el usuario tendrá que volver a hacer todos los niveles que haya pasado.

Para finalizar este apartado, se va a mostrar la función de reinicio en la ilustración número 97.

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class Restart : MonoBehaviour
{
    public void Restart2()
    {
        SceneManager.LoadScene("FirstLevel");
        points.Score = 50;
    }
    public void Quit()
    {
        Application.Quit();
    }
}

```

Ilustración 97: función de reiniciar, lo que hace es cargar el primer nivel con los puntos básicos, para construir las torretas.

6.4. DIAGRAMAS DE BASES DE DATOS.

6.4.1. DIAGRAMA UML

Aquí se va a plasmar la base de datos, compuesta por tres tablas: clientes, administradores y comentarios.

La tabla que tiene más privilegios es la de administradores, ya que este puede controlar los comentarios y usuarios registrados.

Los usuarios son capaces de mandar comentarios, y estos comentarios lo recibe el correo oficial de la empresa, que estos serán leídos por los administradores.

A continuación, se muestra el diagrama UML de la base de datos en la ilustración 98.

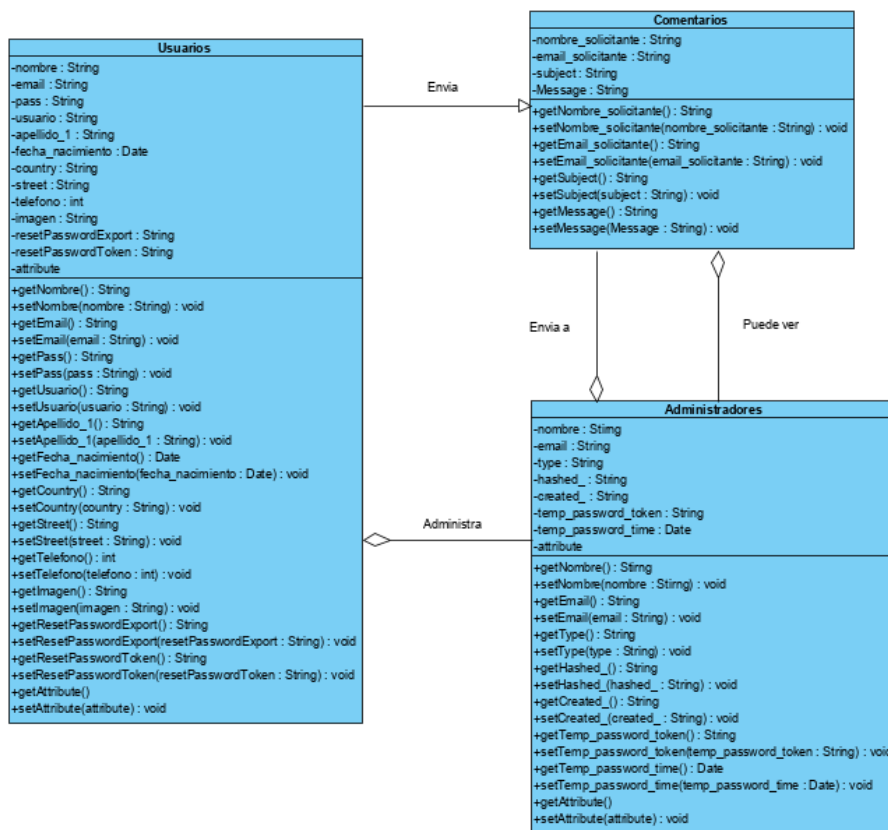


Ilustración 98: Diagrama UML de las bases de datos.

A parte, esta el diagrama de Android, en la que se muestra como es el funcionamiento de todo el programa en ejecución, sabiendo las cosas que se han explicado previamente en los apartados de Android (ilustraciones 99, 100 y 101).

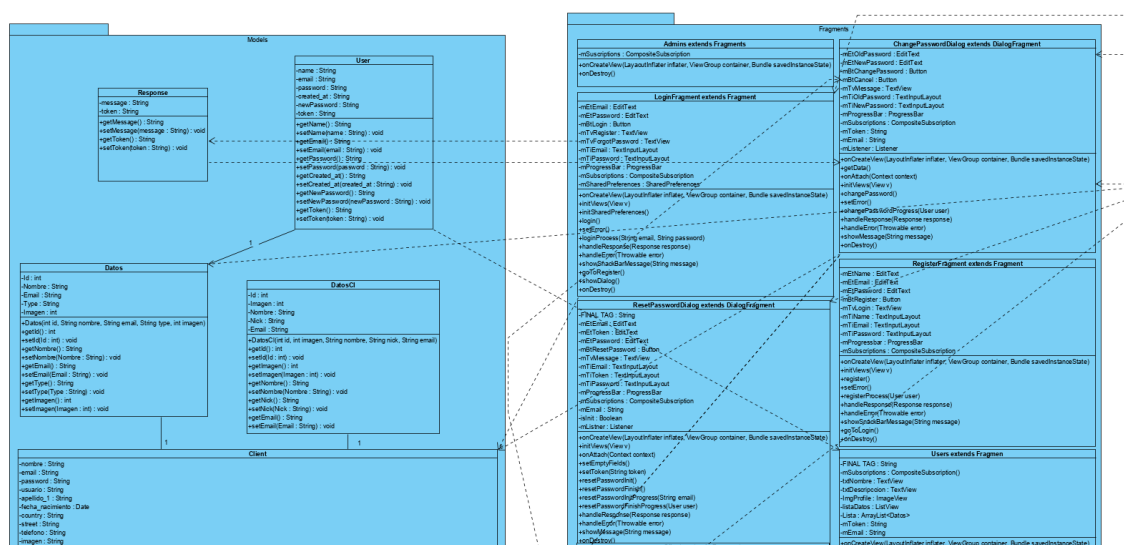


Ilustración 99: 1ª parte de diagrama de Android.

6.4.2. DIAGRAMA DE ENTIDAD RELACIÓN.

En el diagrama de entidad relación, se ven los procesos que tiene cada cosa del proyecto.

Para ello, primero la persona, si quiere iniciar una cuenta como usuario, se debe dirigir a la página web. Este puede iniciar, o registrar la sesión. Si no hay sesión, se debe completar los distintos campos para poder crear la cuenta. En caso contrario, si ya estas registrado, puedes ver los datos de perfil, una vez iniciado la sesión-

Se puede contactar con la empresa, tanto si estas iniciado la sesión, como si no está.

Si en otro caso, la persona quiere acceder al inicio de sesión de administradores. Hay dos tipos, uno es de súper administradores y otro es de administradores normales. Los súper administradores pueden controlar todo, sin embargo, los administradores normales solo pueden controlar los usuarios.

A continuación, se mostrará la imagen del diagrama de entidad relación (ilustración 102).

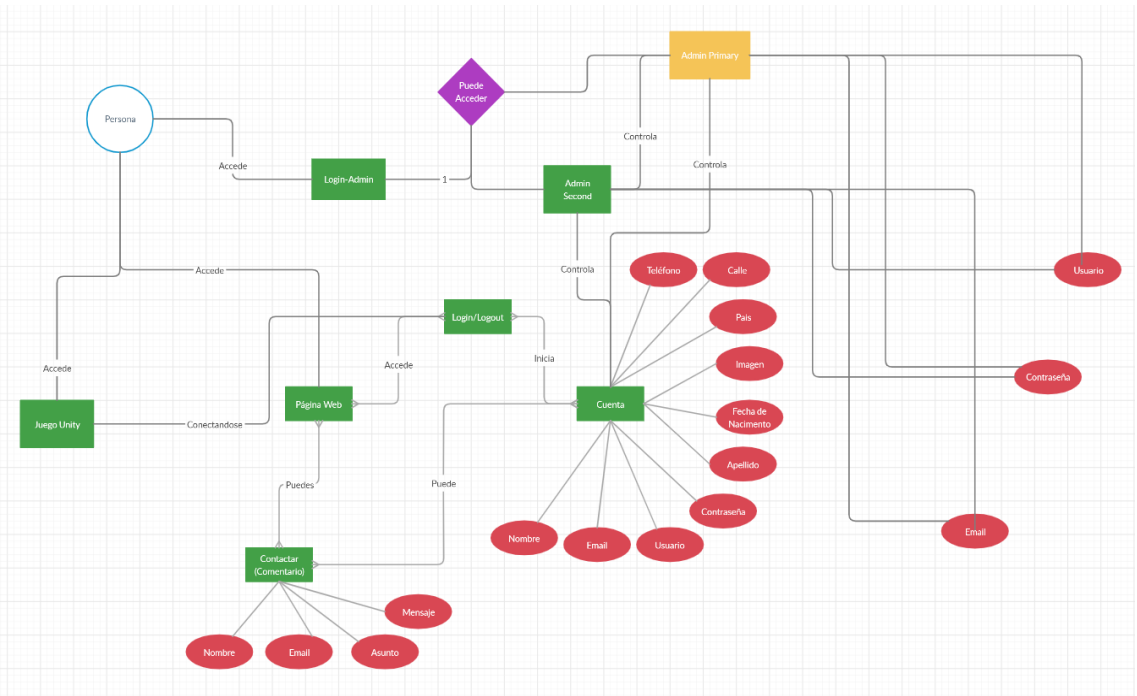


Ilustración 102: Diagrama E/R.

7. CONCLUSIONES

La idea principal que se ha llevado a cabo en este proyecto, es desarrollar un juego, con una misma temática, que, a la hora de pasarse el primer nivel, se fuera complicando cada vez más. Y, además, complementarlo con una página web, indicando las propuestas que se van a desarrollar, con su base de datos correspondiente.

Seguir evolucionando y crear una gran comunidad en el mundo de los videojuegos.

8. BIBLIOGRAFÍA

Bibliografía

- [1] «PalmOS. Historia y características.,» [En línea]. Available: <https://www.ecured.cu/PalmOS>.
- [2] «Palm OS. Características e historia.,» [En línea]. Available: <https://www.xatakamovil.com/xatakamovil/que-fue-de-las-palm>.
- [3] «Más características e historia de PalmOS.,» [En línea]. Available: <https://www.pasionmovil.com/plataformas/palm/la-historia-interna-de-la-muerte-de-palm-y-webos/>.
- [4] «Guía de usuario de Android Studio.,» [En línea]. Available: <https://developer.android.com/studio/intro>.
- [5] «Herramienta IntelliJ IDEA.,» [En línea]. Available: <https://www.jetbrains.com/es-es/idea/>.
- [6] «Agregar C++ en la plataforma de Android Studio.,» [En línea]. Available: <https://developer.android.com/studio/projects/add-native-code?hl=es-419>.
- [7] «Introducción de Kotlin en Android Studio.,» [En línea]. Available: <https://developer.android.com/kotlin>.
- [8] «NDK en Android Studio.,» [En línea]. Available: <https://developer.android.com/ndk>.
- [9] «GitHub.,» [En línea]. Available: <https://github.com/>.
- [10] «Introducción de la nueva herramienta de Lint. Con sus características.,» [En línea]. Available: <http://tools.android.com/tips/lint>.
- [11] «Cosas que se pueden hacer con Gradle en Android Studio.,» [En línea]. Available: https://developer.android.com/studio/build/gradle-tips/?gclid=EAlaIQobChMII-uko9Wz6QIVz9vVCh0CUgMoEAAYASAAEgIJA_D_BwE&gclsrc=aw.ds.
- [12] «Introducción de java en Android Studio.,» [En línea]. Available: <https://developer.android.com/studio/write/create-java-class?hl=es-419>.
- [13] «Que es Java. Página Oficial.,» [En línea]. Available: https://www.java.com/es/download/faq/whatis_java.xml.

- [14 «Que es y características de Java,» [En línea]. Available:
] <https://desarrolloweb.com/articulos/497.php>.
- [15 «Características, historia y principios de Java,» [En línea]. Available:
] <https://openwebinars.net/blog/que-es-java/>.
- [16 «Principios básicos para aprender Unity.,» [En línea]. Available:
] <https://openwebinars.net/blog/que-es-unity/>.
- [17 «Documentación de Unity.,» [En línea].
]
- [18 «Primeros paros para programar en C#,» [En línea]. Available:
] <https://docs.microsoft.com/es-es/dotnet/csharp/programming-guide/>.
- [19 «Pasos que puedes seguir para programar en CSharp,» [En línea]. Available:
] <https://docs.microsoft.com/es-es/dotnet/csharp/>.
- [20 «Las primeras aplicaciones lanzadas por PalmOS,» [En línea]. Available:
] <https://www.appandweb.es/blog/historia-aplicaciones-moviles/>.
- [21 «Como surgieron las primeras páginas web.,» [En línea]. Available:
] <https://www.appandweb.es/blog/historia-aplicaciones-moviles/>.
- [22 «Ventajas y desventajas a analizar las páginas web,» [En línea]. Available:
] <https://blog.bitgare.com/2018/11/05/ventajas-y-desventajas-de-tener-o-no-tener-una-pagina-web-en-tu-empresa/>.
- [23 «W3.org. Analizar los errores que se puede tener en una página.,» [En línea]. Available:
] <https://www.w3.org/>.
- [24 «Principios básicos HTML,» [En línea]. Available:
] <https://desarrolloweb.com/articulos/que-es-html.html>.
- [25 «Que es el HTML, con la explicación de funciones,» [En línea]. Available:
] <https://developer.mozilla.org/es/docs/Web/HTML>.
- [26 «CSS, que es y para que se utiliza.,» [En línea]. Available:
] <https://www.workana.com/i/glosario/css/>.
- [27 «CSS, que es y para que se utiliza,» [En línea]. Available:
] <https://www.hostinger.es/tutoriales/que-es-css/>.
- [28 «CSS, que es y para que se utiliza.,» [En línea]. Available: <https://silicodevalley.com/que-es-el-html-y-el-css/>.
]
- [29 «Bootstrap,» [En línea]. Available: <https://getbootstrap.com/>.
]
- [30 «Node.js. Página oficial - acerca de,» [En línea]. Available: <https://nodejs.org/es/about/>.
]

- [31 «Introducción a Twisted y al desarrollo de servidores,» [En línea]. Available:
] <https://recursospython.com/guias-y-manuales/introduccion-a-twisted/>.
- [32 «EventMachine,» [En línea]. Available:
] <https://rubygems.org/gems/eventmachine/versions/1.2.0.1?locale=es>.
- [33 «Características y conclusiones que tiene Node.js,» [En línea]. Available:
] <https://www.luisllamas.es/que-es-node-js/>.
- [34 «Características y conclusiones que tiene Node.js,» [En línea]. Available:
] <https://ifgeekthen.everis.com/es/que-es-node-js-y-primeros-pasos>.
- [35 «Características y conclusiones que tiene Node.js,» [En línea]. Available:
] <https://apasionados.es/blog/nodejs-4430/>.
- [36 «Página oficial de NPM,» [En línea]. Available: <https://www.npmjs.com/>.
]
- [37 «NPM Daemon,» [En línea]. Available: <https://www.npmjs.com/package/daemon>.
]
- [38 «NPM Express,» [En línea]. Available: <https://www.npmjs.com/package/express>.
]
- [39 «NPM Passport,» [En línea]. Available: <https://www.npmjs.com/package/passport>.
]
- [40 «NPM Express.Router,» [En línea]. Available: <https://www.npmjs.com/package/express-router>.
]
- [41 «NPM Bcrypt,» [En línea]. Available: <https://www.npmjs.com/package/bcrypt>.
]
- [42 «NPM Async,» [En línea]. Available: <https://www.npmjs.com/package/async>.
]
- [43 «NPM Crypto,» [En línea]. Available: <https://www.npmjs.com/package/crypto-js>.
]
- [44 «NPM Nodemailer,» [En línea]. Available: <https://www.npmjs.com/package/nodemailer-smtp-transport>.
]
- [45 «Página oficial JavaScript,» [En línea]. Available: <https://www.javascript.com>.
]
- [46 «Características e información de JavaScript,» [En línea]. Available:
] https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/Qu%C3%A9_es_JavaScript.
- [47 «Características e información de JavaScript,» [En línea]. Available:
] <https://openwebinars.net/blog/que-es-javascript/>.

- [48 «Ventajas y desventajas en JavaScript,» [En línea]. Available:
] <https://www.nextu.com/blog/conoce-las-ventajas-y-desventajas-de-javascript/>.
- [49 «Página oficial JQuery.,» [En línea]. Available: <https://jquery.com/>.
]
- [50 «Información de las bases de datos no relacionales.,» [En línea]. Available:
] <https://www.silicon.es/bases-datos-no-relacionales-nosql-cuando-usarlas-2324948>.
- [51 «Información de las bases de datos no relacionales.,» [En línea]. Available:
] <https://www.acens.com/wp-content/images/2014/02/bbdd-nosql-wp-acens.pdf>.
- [52 «Información de las bases de datos no relacionales.,» [En línea]. Available:
] <https://www.unir.net/ingenieria/revista/noticias/bases-de-datos-nosql/549204809132/>.
- [53 «Página oficial MongoDB.,» [En línea]. Available: <https://www.mongodb.com/es> .
]
- [54 «Metodología Lean: todo lo que se debe saber sobre esta metodología.,» [En línea].
] Available: <https://thedigitalprojectmanager.com/es/metodologias-gestion-proyectos-simplificadas/#waterfall>.
- [55 «Metodología Lean: todo lo que se debe saber sobre esta metodología.,» [En línea].
] Available: <http://kailean.es/el-metodo-lean/>.
- [56 «Metodología Lean: todo lo que se debe saber sobre esta metodología.,» [En línea].
] Available: <https://obsbusiness.school/es/blog-investigacion/direccion-general/lean-management-metodologia-origenes-y-principios>.