

**Plan de Empresa**

Álvaro Martínez Iglesias

Javier Prieto Abad

**Curso 2020/21**

Videojuego “Shinobi Quest”



**Grado Superior**

**Desarrollo de aplicaciones multiplataforma**

Contenido

[**1** **Agradecimientos** 4](#_Toc74571298)

[**2** **Resumen** 4](#_Toc74571299)

[**3** **Abstract** 4](#_Toc74571300)

[**4** **Introducción** 4](#_Toc74571301)

[**5** **Alcance funcional del sistema** 5](#_Toc74571302)

[**6** **Diseño técnico** 5](#_Toc74571303)

[6.1 Diagrama de clases 5](#_Toc74571304)

[6.2 Esquema de bases de datos 8](#_Toc74571305)

[6.3 Diseño de la interfaz de usuario 9](#_Toc74571306)

[**7** **Tecnología utilizada** 14](#_Toc74571307)

[**8** **Código documentado** 16](#_Toc74571308)

[**9** **Presupuesto** 18](#_Toc74571309)

[**10** **Planificación** 19](#_Toc74571310)

[**11** **Manual de instalación** 21](#_Toc74571311)

[**12** **Manual de usuario** 21](#_Toc74571312)

[**13** **Análisis de la competencia y de los clientes** 21](#_Toc74571313)

[**14** **Conclusiones** 21](#_Toc74571314)

[**15** **Bibliografía** 22](#_Toc74571315)

# **Agradecimientos**

Agradecer sobre todo a mi hermano que me ayudó a hacer algunos “sprites” para que no me comiera tanto tiempo la realización de estos. Además de a Óscar, Adrían y Mario, que me dieron ideas y probaron mi proyecto varias veces con el fin de sacarme fallos o ideas.

# **Resumen**

Consiste en un videojuego 2D “Roguelike” (RPG de acción) con un sistema de generación aleatoria y un sistema de combate por turnos. Me he inspirado en grandes juegos como el “The Binding of Isaac”, “Dungeons and Dragons” y “Final Fantasy”.

La dinámica principal del juego se centra en la exploración de los distintos pisos, con el objetivo de completarlo, derrotando al jefe final de cada piso y pasando al siguiente, encontrando distintos objetos que te ayudarán en el camino.

# **Abstract**

It consists of a 2D video game “Roguelike” (action RPG) with a random generation system and a turn-based combat system. I have been inspired by great games like "The Binding of Isaac", "Dungeons and Dragons" and "Final Fantasy".

The main dynamic of the game focuses on exploring the different floors, with the aim of completing it, defeating the final boss of each floor and moving on to the next, finding different objects that will help you along the way.

# **Introducción**

Este proyecto se trata de una muestra de lo que tengo en mente, no es para nada definitivo, estoy enseñando una idea, una idea de en lo que se puede convertir, pero ahora mismo todavía le falta mucho para que sea un proyecto que pueda enseñar bien orgulloso. Tengo la intención de continuar desarrollando esta idea que ha surgido realizando este trabajo de fin de grado, debido a que he disfrutado creando mi primer juego propio. Por lo que planeo continuar desarrollándolo y añadiéndole un montón de ideas, tales como mejorar el sistema de combate, añadir un sistema de mini mapa, la implementación de un inventario junto con todo un sistema de objetos consumibles soltados por los monstruos al morir o recolectados en distintos sitios. Además de añadir un montón de contenido nuevo como mapas, zonas, objetos, enemigos, sonidos, personajes jugables, etc.

Por lo tanto aquí presento el primer boceto hecho en estos tres meses, y deseando poder presentar algún día algo mucho más grande.

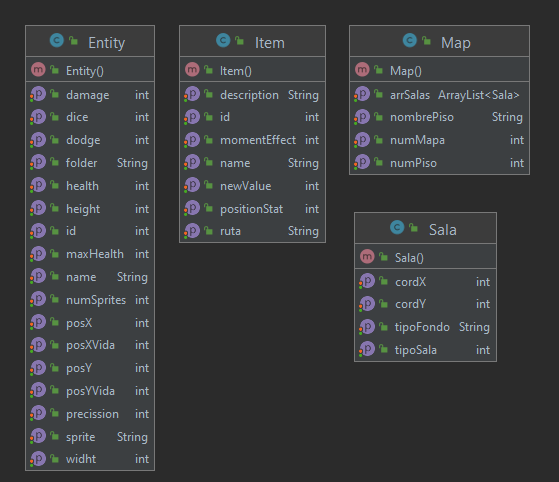
# **Alcance funcional del sistema**

La única función que tiene mi programa es la de jugar al juego, dentro del juego ya explorarás los pisos, pelearás contra enemigos y encontrarás recompensas hasta derrotar al jefe final.

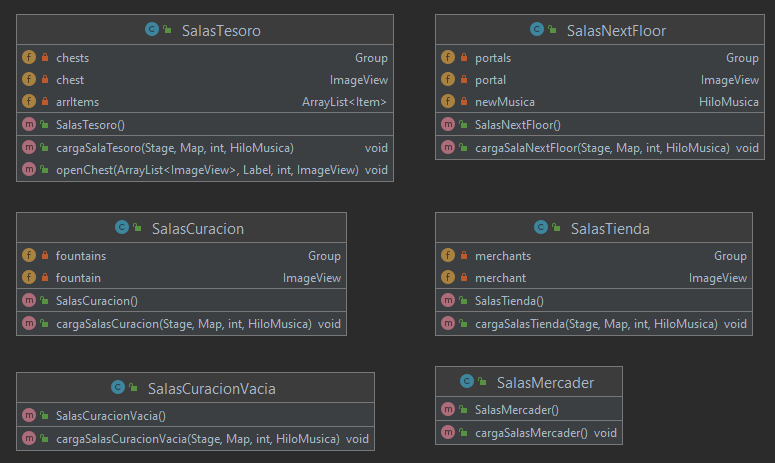
# **Diseño técnico**

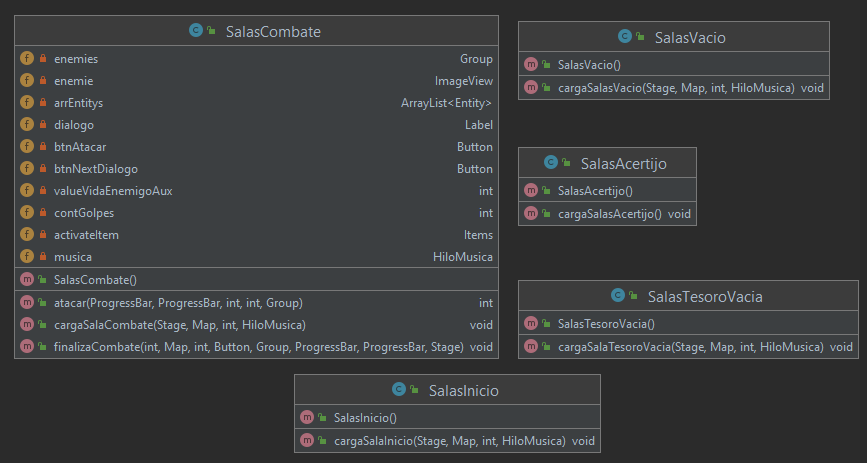
## Diagrama de clases

Clases modelo:

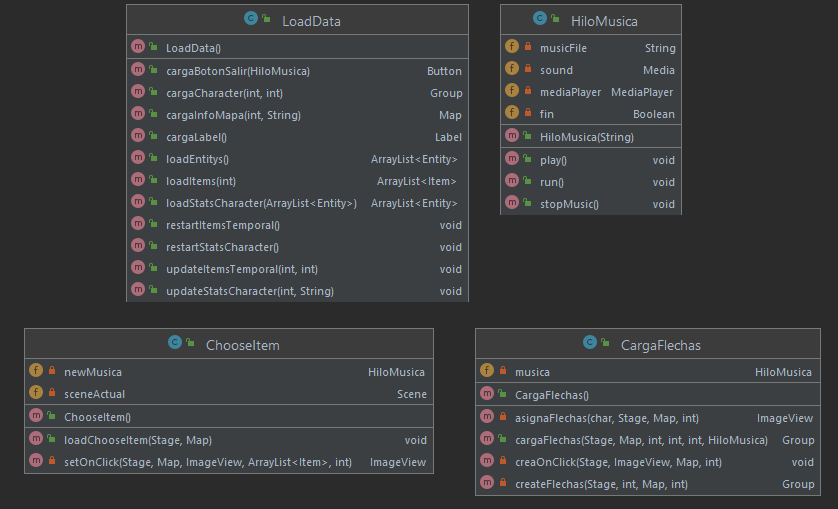


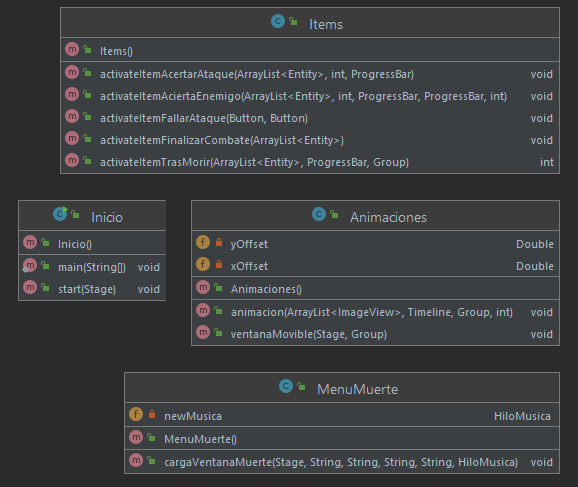
Clases vista:



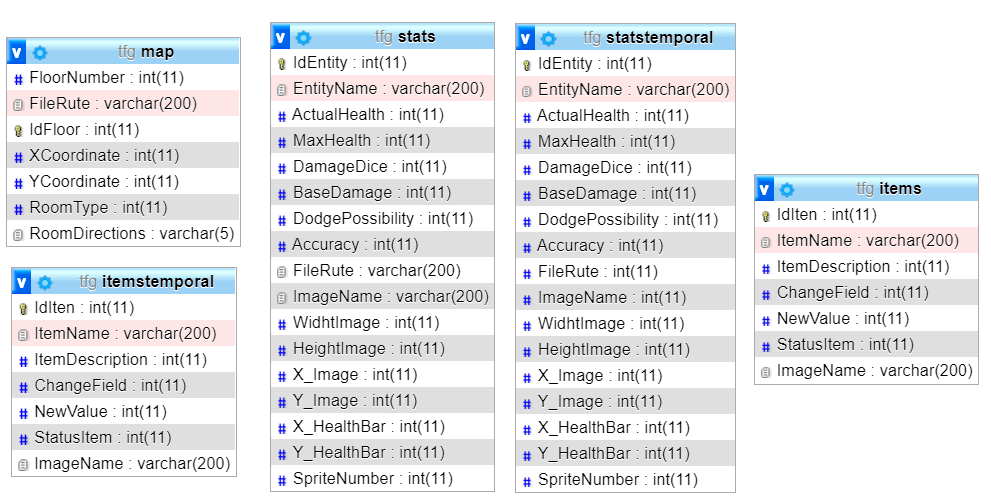


Clases controlador:



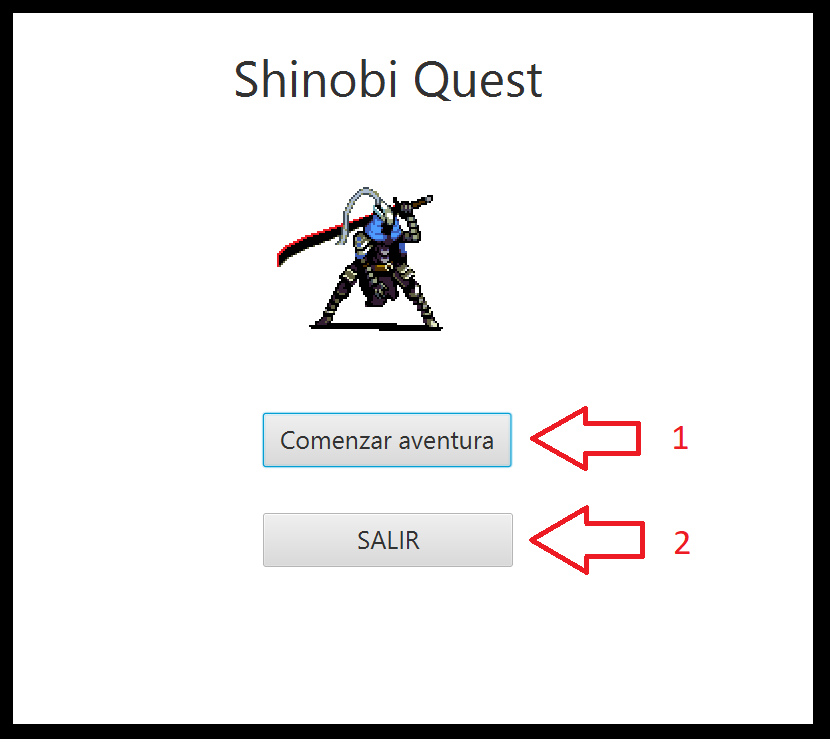


## Esquema de bases de datos

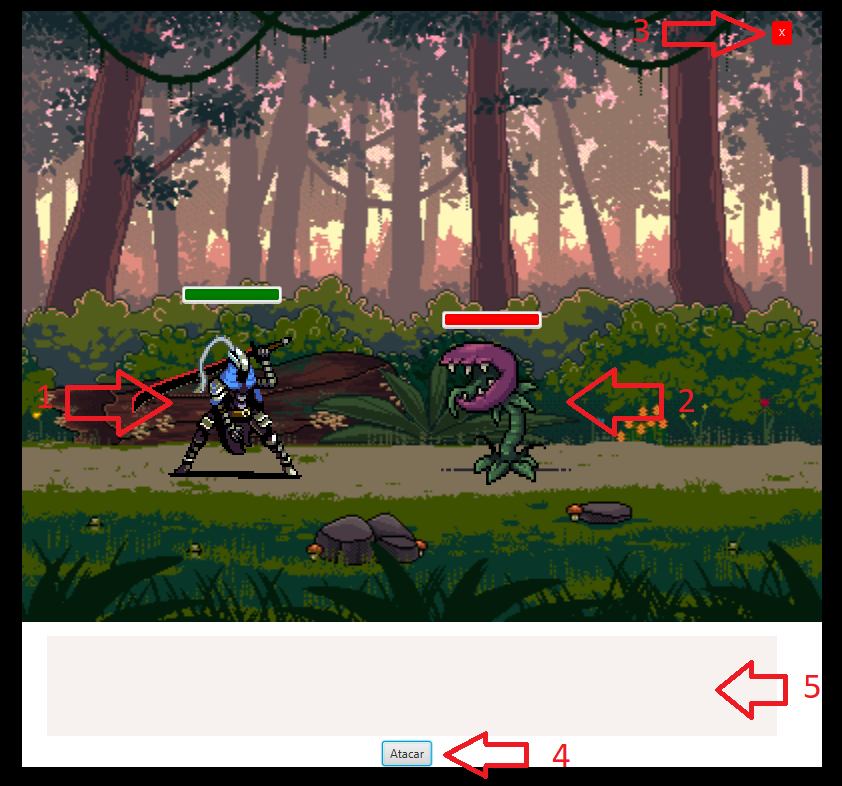


Mi sistema de almacenamiento de la información lo realizo a través de ficheros CSV, pero he creado una BBDD para que se entienda bien la información que estoy guardando. Como se puede ver utilizo 5 ficheros, dos de ellos son temporales y son iguales a su fichero correspondiente, pero es con los que trabaja el juego y va sobrescribiendo la información mediante el transcurso del juego, dejando los otros intactos con la información de inicio. Por último, en el fichero llamado “Map”, los últimos 4 campos que vemos se repiten tantas veces salas haya en dicho piso.

## Diseño de la interfaz de usuario



La primera interfaz que ve el usuario es una interfaz sencilla, recreando el menú inicial del juego, esta cuenta con la opción de comenzar el juego (1), y la opción de salirse del mismo (2) junto con el título del juego y el personaje.



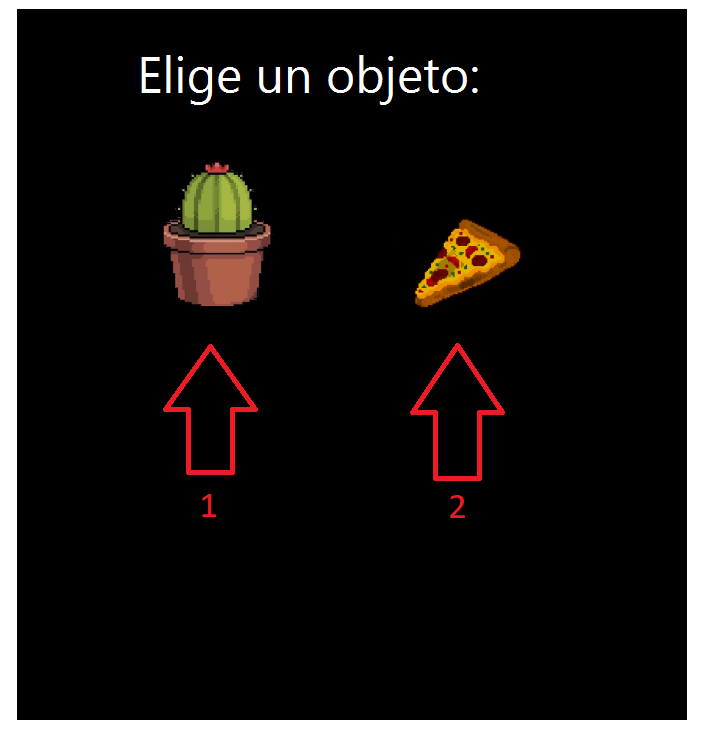
La segunda interfaz muestra el sistema de combate del juego, mostrando a tú personaje junto con su barra de vida (1), el enemigo junto con su barra de vida (2), la opción de salirte del juego (3), que te preguntará si estás seguro que quieres salir, el botón de acción, ya sea atacar, continuar, etc. (4) y el panel de diálogo, dónde va apareciendo la información de lo que va sucediendo.



La tercera interfaz es la más común, es la interfaz dónde te irás moviendo por el mapa, mostrando las flechas de direcciones, el resto de la interfaz es igual a la anterior.



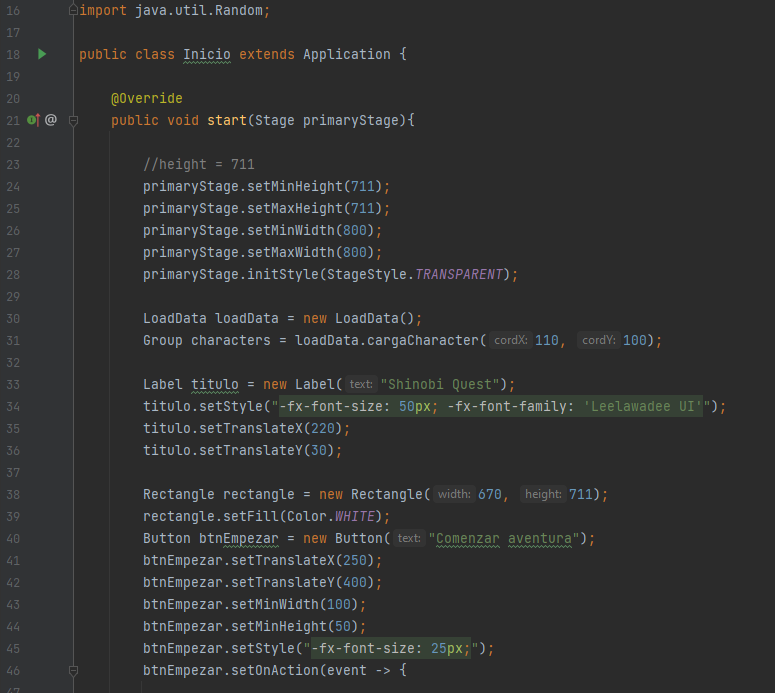
Esta es la interfaz del final del juego, ya sea de victoria o de derrota, lo único que cambian son los textos y la imagen. Como podemos ver consta de in título junto a una descripción y una imagen (1), luego el botón de volverá empezar el juego o salirte.



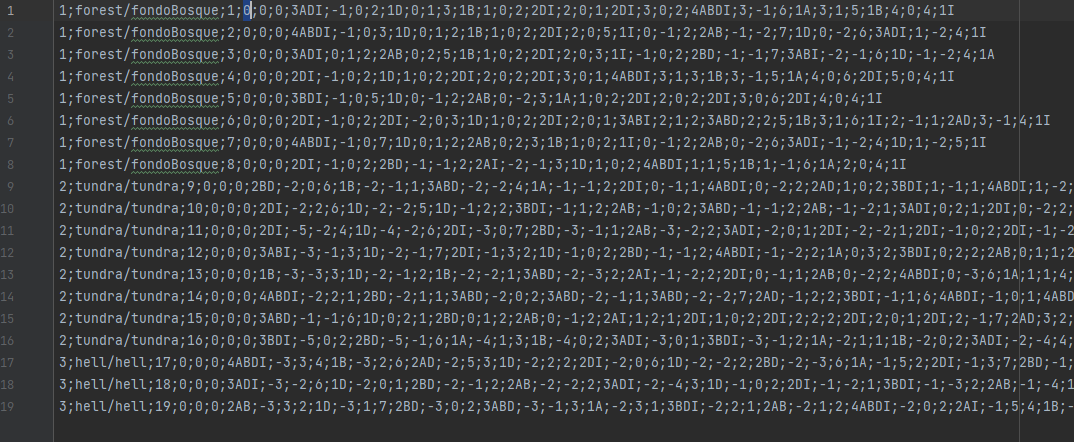
Esta interfaz aparece tras derrotar a cada jefe final de un piso, antes de pasar al siguiente piso nos da la opción de seleccionar un objeto entre los dos generados aleatoriamente.

# **Tecnología utilizada**

Para la creación de la aplicación he usado Java FX, que es una librería de Java, que me facilita la creación de las interfaces y el diseño de estas, es como una versión mejorada de la librería Java Swing, que es la que hemos estudiado en este curso.



Como sistema de almacenamiento, como he dicho antes he utilizado ficheros CSV para guardar la información y acceder a ella luego.



También he usado las clase ”Thread” para hacer funciones paralelas al funcionamiento del programa.



# **Código documentado**

He comentado el código como se tendría que comentar para realizar el JavaDoc, pego un fragmento del código de mi programa para que se vea de qué forma lo he comentado:

*/\*\*  
 \* Función que se encarga de la información del CSV "maps" y pasarlo a un objeto que guarde esas características  
 \** ***@param*** *mapaSeleccionado  
 \** ***@param*** *rutaMapa  
 \** ***@return*** *map  
 \*/*public Map cargaInfoMapa(int mapaSeleccionado, String rutaMapa){  
  
 Map map = new Map();  
 BufferedReader bufferedReader;  
 boolean salir = false;  
  
 try {  
 bufferedReader = new BufferedReader(new FileReader("files/maps.txt"));  
 String line="";  
 String[] fields = null;  
  
 while ((line=bufferedReader.readLine())!=null && !salir) {  
 fields = line.split(";");  
 if (Integer.*parseInt*(fields[2]) == mapaSeleccionado){  
 salir=true;  
 }  
 }  
  
  
 map.setNumPiso(Integer.*parseInt*(fields[0]));  
 map.setNombrePiso(rutaMapa);  
 map.setNumMapa(Integer.*parseInt*(fields[2]));  
 ArrayList<Sala> arrSalas = new ArrayList<>();  
  
 for (int i=0 ; i<(fields.length-2)/4 ; i++){  
  
 Sala sala = new Sala();  
  
 sala.setCordX(Integer.*parseInt*(fields[(i\*4)+3]));  
 sala.setCordY(Integer.*parseInt*(fields[(i\*4)+4]));  
 sala.setTipoSala(Integer.*parseInt*(fields[(i\*4)+5]));  
 sala.setTipoFondo(fields[(i\*4)+6]);  
  
 arrSalas.add(sala);  
 }  
  
 map.setArrSalas(arrSalas);  
  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 return map;  
}  
  
*/\*\*  
 \* Función que crea la animación del personaje principal y le asigna sus características  
 \** ***@param*** *cordX  
 \** ***@param*** *cordY  
 \** ***@return*** *characters (Grupo con todos los Sprites del personaje)  
 \*/*public Group cargaCharacter(int cordX, int cordY){  
  
 Group characters;  
 ImageView character;  
  
 ArrayList<ImageView> arrCharacter = new ArrayList<>();  
  
 String ruta;  
 for (int i=1 ; i<=10 ; i++) {  
 ruta = "imgs/shinobi/shinobi" + i + ".png";  
 character = new ImageView(ruta);  
 arrCharacter.add(character);  
 }  
  
 characters = new Group(arrCharacter.get(0));  
 characters.setTranslateX(cordX);  
 characters.setTranslateY(cordY);  
  
 Timeline timelineCharacter = new Timeline();  
 timelineCharacter.setCycleCount(Timeline.*INDEFINITE*);  
  
 new Animaciones().animacion(arrCharacter, timelineCharacter, characters, 10);  
 timelineCharacter.play();  
  
 return characters;  
}  
  
*/\*\*  
 \* Función que crea y le asigna los datos a el cuadro de texto donde aparece el texto emergente  
 \** ***@return*** *dialogo  
 \*/*public Label cargaLabel(){  
  
 Label dialogo = new Label();  
 dialogo.setTranslateX(25);  
 dialogo.setTranslateY(625);  
 dialogo.setMinWidth(730);  
 dialogo.setMaxWidth(730);  
 dialogo.setMinHeight(100);  
 dialogo.setMaxHeight(100);  
 dialogo.setStyle("-fx-background-color: #f7f1f0; -fx-padding: 20px; -fx-font-size: 15px;");  
  
 return dialogo;  
}  
  
*/\*\*  
 \* Función que crea el botón de salir, deteniendo el hilo de la múscia y cerrando el programa  
 \** ***@param*** *musica  
 \** ***@return*** *btnSalir  
 \*/*public Button cargaBotonSalir(HiloMusica musica){  
  
 Button btnSalir = new Button("X");  
 btnSalir.setTranslateX(750);  
 btnSalir.setTranslateY(10);  
 btnSalir.setStyle("-fx-background-color: red; -fx-text-fill: white; -fx-font-size: 10");  
 btnSalir.setOnAction(event -> {  
 Alert alert = new Alert(Alert.AlertType.*CONFIRMATION*);  
 alert.setTitle("Cuidado!");  
 alert.setHeaderText("Estas seguro que deseas cerrar el juego?");  
  
 Optional<ButtonType> result = alert.showAndWait();  
 if (result.get() == ButtonType.*OK*){  
 musica.stopMusic();  
 musica.stop();  
 Platform.*exit*();  
 }  
 });  
  
 return btnSalir;  
}

# **Presupuesto**

Para la creación de este proyecto no he tenido ningún gasto, ya que no he tenido que alquilar ningún servidor ni ningún servicio, pero si contara con que esto es un proyecto de desarrollo de una empresa y tuviera que analizar cuánto costaría la producción de este, saldrían los siguientes números:

Lo primero, contando con que habría que pagar la licencia de cada “asset” (sprites, música, etc.) poniendo como que la media de coste por cada enemigo son 3€ y que el conjunto de objetos me saldría por unos 2€, además de 2€ por audio saldría una cifra de 93€.

Luego pagando 8€/hora a el encargado del desarrollo de la aplicación, habiendo realizado yo aproximadamente unas 70 horas saldría el costo de 560€.

Por lo tanto, nos quedaría una cifra de 653€.

# **Planificación**

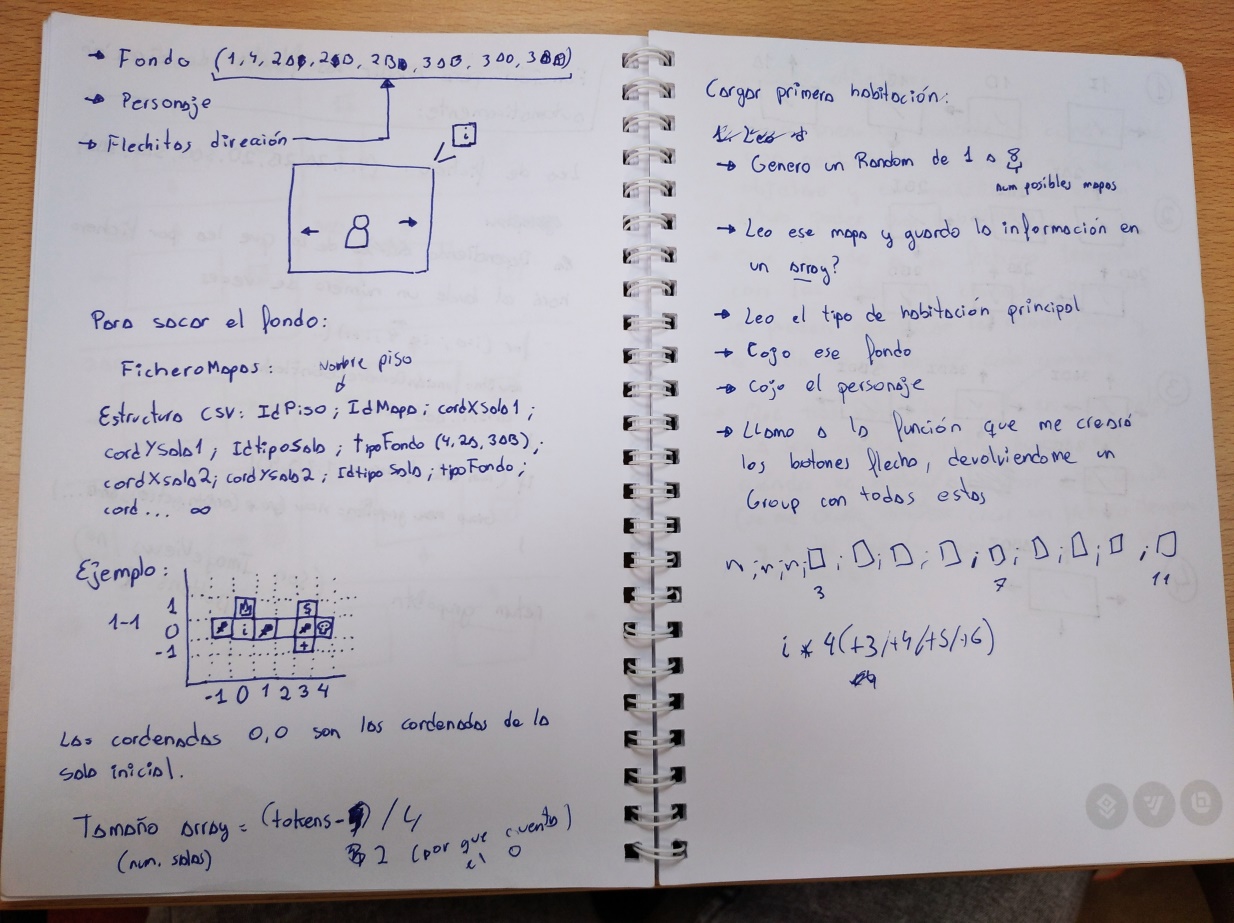
Desde que supe que había que presentar un proyecto final al terminar el curso lo primero que se me ocurrió fue crear un videojuego, que es lo que siempre me ha atraído y de lo que más tenía ganas. Una vez con esto en mente empecé a pensar ideas de qué podría hacer con los conocimientos aprendidos a lo largo del curso, llegando a la conclusión de que lo más acertado sería hacerlo con Java, ya que es el lenguaje que más hemos tocado. Otra opción hubiera sido Unity, lo que hubiera servido para hacer el diseño del juego mucho más profesional, pero puestos a que solo habíamos dado un mes no me veía lo suficientemente preparado para esto, así que decidí quedarme con Java.

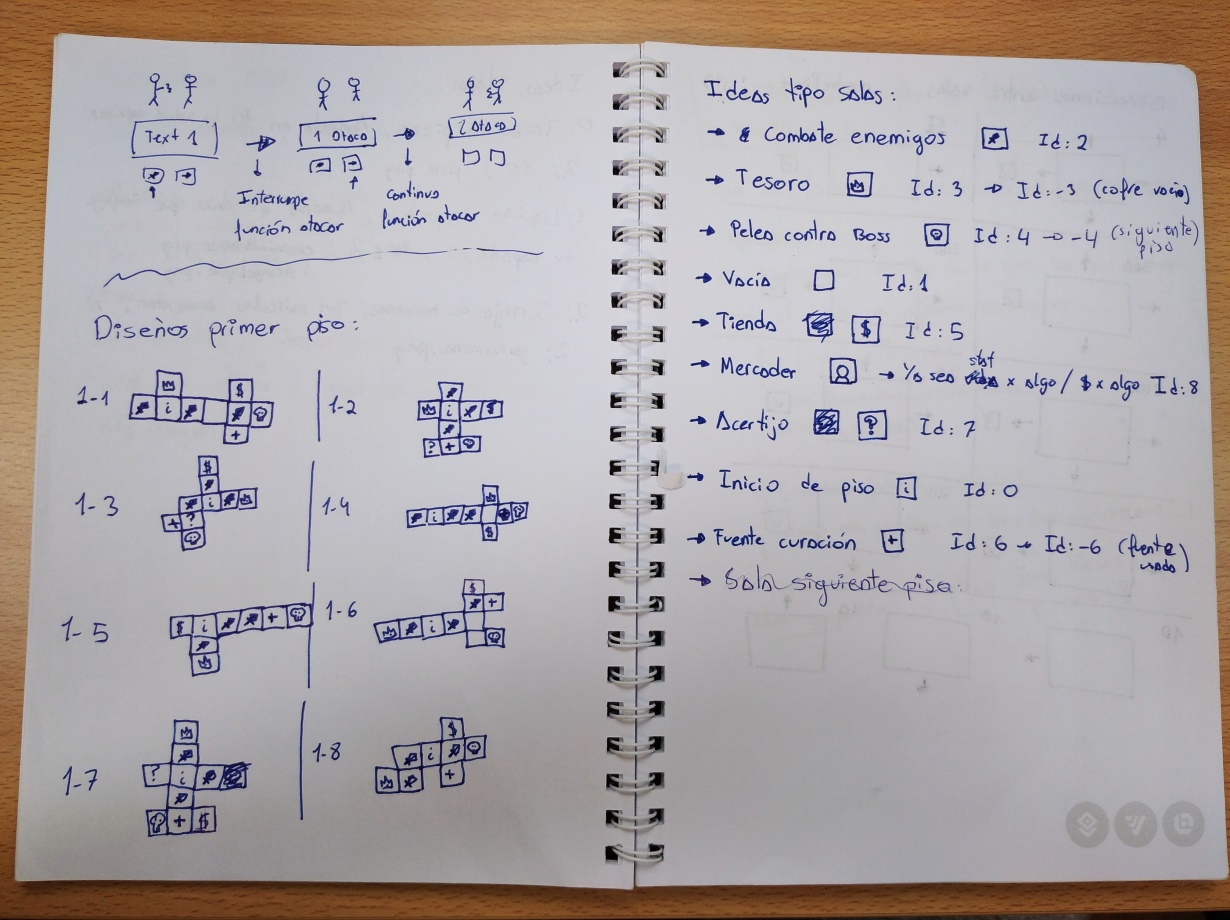
Una vez decidido el lenguaje lo siguiente era pensar la librería con la que realizar el diseño gráfico del juego, debido a que era lo único que se me escapaba. Tras investigar llegué a la conclusión de que Java FX sería lo más óptimo.

Tras esto ya quedaba la idea del juego, desde hace tiempo ya se me ocurrió la idea de hacer un juego 2D con “Pixel Art” y que la mecánica fuese de ir explorando pisos, mientras se va descendiendo y llegando hasta el final, peleando contra enemigos y jefes, encontrando objetos por el camino que cambiaran toda la forma de pensar y actuar, causando así que cada partida sea distinta a la anterior.

Una vez establecida la idea me puse a realizar proyectos pequeñitos para ir probando varias cosas que desconocía, como era la primera vez que usaba Java FX hice una serie de pruebas usando esta librería y viendo cómo funcionaba por debajo. Hice pruebas como el sistema de pasar de una ventana a otra o el sistema de las animaciones.

Tras ya tener un poco la idea de cómo funcionaba todo empecé a programar poquito a poco, haciendo primero la sala de combate ya que sería la más importante dentro del juego, e ideando el sistema de combate. Tras eso fui haciendo paso a paso todo lo que se iba ocurriendo, e iba documentando o escribiendo ideas mientras trabajaba en una libreta de mano.





# **Manual de instalación**

Lo primero de todo, debido a que el juego está hecho en Java hace falta instalar la Java ™ Platform SE binary para poder ejecutar el juego, luego simplemente es copiarte la carpeta con los archivos a tu escritorio y ya podrías ejecutarlo.

# **Manual de usuario**

Para jugar lo único que tiene que hacer es abrir el juego y ya está, luego hay dos carpetas que están ocultas junto al juego, que no debe tocar el usuario, son carpetas que contienen información importante que si son editadas pueden causar daños en el juego.

# **Análisis de la competencia y de los clientes**

Hay bastantes juegos que se basan un poco en lo mismo, la exploración de pisos y el descenso de estos hacia nuevos pisos, generando objetos, enemigos, zonas, etc. Todo esto aleatoriamente. Cada juego en sí es distinto, todos comparten la misma mecánica pero tiene detalles específicos, el mío actualmente cuenta con un sistema de combate por turnos, cosa que lo diferencia ya de la mayoría de los demás, también tengo planeado mejorar el sistema de combate añadiendo más incógnitas a la fórmula general del combate, con más ataques y muchas más posibilidades.

Respecto a los clientes este sistema de juego es uno que vende mucho, es un sistema muy querido por la comunidad, lo que si se consigue hacer bien y saber diferenciarlo de los demás, dándole una característica novedosa y divertida funcionará muy bien en la industria.

# **Conclusiones**

Como he dicho en la introducción, este proyecto es una idea de todo lo que tengo en mente, una primera visualización de lo que se puede convertir.

He disfrutado mucho haciendo este proyecto, me lo he pasado como un niño enseñando lo que llevaba a mis compañeros, explicándoles una nueva mecánica que había implementado, y planeo seguir con ese sentimiento, tengo curiosidad por ver dónde puede acabar este proyecto en unos años.

# **Bibliografía**

Librería de Java FX para la realización del proyecto: <https://openjfx.io/>

Sprites: <https://www.pinterest.es/>, <https://www.freepng.es/>, entre otras, de normal buscaba en google lo que quería conseguir y lo sacaba de ahí.

Música: <https://www.youtube.com/>

Solución de errores: <https://es.stackoverflow.com/> o <https://www.youtube.com/>