

# JumpWords: juego Android de validación de datos con técnicas de gamesourcing

Adrián Mengual Valenzuela

**Resumen**— JumpWords consiste en el desarrollo de un juego Android para smartphones donde el usuario validará unos datos de entrada de una manera más rápida, intuitiva e interactiva. Para el diseño se utilizará, tal y como dice el título del proyecto, el modelo de gamesourcing, consiste en realizar un juego más divertido para el usuario, donde tiene un objetivo (pasar y superar nuevos niveles, conseguir puntos, etc.) cada vez que transcriba bien una palabra. JumpWords es un juego destinado al Proyecto Xarxes, que permite estudiar, entender y explicar mejor la evolución social y económica del pasado, transcribiendo/validando fuentes originales para la construcción de redes sociales del pasado. JumpWords tiene el propósito de mejorar la experiencia del usuario, de manera que el tiempo dedicado a jugar sea más divertido y entretenido que juegos similares.

**Paraules clau**— Android, Gamesourcing, usuarios, transcriba, datos de entrada.

**Abstract**— JumpWords is the development of a Android games for smartphones, where the user will validate data in a faster, intuitive and interactive way. For the design will be used, the model of gamesourcing, consists of making a fun game for the user, that has a purpose (pass and overcome new levels, get points, etc.) each time you transcribe well a Word. JumpWords is a game aimed at the Xarxes Project, which allows you to study, understand and explain better the social and economic evolution of the past, transcribing/validating original sources for the construction of social networks of the past. JumpWords has the purpose of improving the user experience, so that the time spent playing is more fun and entertaining than similar games.

**Index Terms**— Android, Gamesourcing, users, transcribe, input data.



## 1 INTRODUCCIÓN

JumpWords: juego Android de validación de datos con técnicas de gamesourcing, es un proyecto basado en un juego para smartphones donde el usuario validará datos de entrada transcribiéndolos de una manera más dinámica y rápida. Para el diseño se utilizará el modelo de Gamesourcing [3], basado en el paradigma de crowdsourcing, es decir, dividir el trabajo en diferentes tareas distribuidas entre diferentes usuarios. Gamesourcing se centra en mejorar la experiencia del usuario haciendo el juego mucho más divertido, interactivo y que atraiga su atención para seguir jugando.

JumpWords irá destinado al Proyecto Xarxes [4]. Xarxes es un proyecto que permite estudiar no solo el comportamiento demográfico, sino que también entender y explicar mejor la evolución social y económica del pasado. Utilizando la visión por computadores como tecnología, se incorporará mecanismos que permitan adaptar contenido de fuentes originales, asociando la información semántica.

- E-mail de contacto: [adrian.mengual4@gmail.es](mailto:adrian.mengual4@gmail.es)
- Mención realizada: Ingeniería del Software.
- Trabajo tutorizado por: Alicia Fornes
- Curso 2018/19

Esto quiere decir que JumpWords se implementará principalmente para la transcripción/validación de estas fuentes originales para la construcción de redes sociales del pasado. La participación de los usuarios será muy importante ya que ellos serán los que validen estos datos jugando a JumpWords.

### 1.1 Objetivo

A continuación, se definirán los objetivos principales del proyecto y, más adelante, se dividirán en diferentes subobjetivos.

El objetivo principal del proyecto es desarrollar un juego Android para transcribir/validar datos de reconocimiento automático de textos manuscritos. Esto quiere decir, que el usuario transcribirá/validará estos datos de entrada.

El juego tiene que permitir a los usuarios transcribir/validar las palabras que irán apareciendo en la pantalla para seguir jugando.

Principalmente este trabajo se centrará en el objetivo principal de transcribir/validar los datos de entrada y mejorar la experiencia del usuario.

Para llevar a cabo el objetivo principal del proyecto se requiere dividirlo en diferentes subobjetivos:

### 1.1.1 Estudiar soluciones existentes

Estudiar los diferentes juegos que hay actualmente y tienen el mismo objetivo que JumpWords. Para examinar dichos juegos se observará los pros y contras que tiene cada uno y así adquirir nuevas ideas para diseñar JumpWords. Sobretudo mejorar estos contras para implementar un juego con muchas más jugabilidad y más llamativo, pero siempre teniendo el mismo objetivo, transcribir/validar estos datos de entrada.

### 1.1.2 Diseñar juego Android basado en los paradigmas de crowdsourcing y gamificación

El diseño de JumpWords está basado en los paradigmas de crowdsourcing y gamesourcing. A continuación, se explican estos dos paradigmas:

- Crowdsourcing es una herramienta para dividir el trabajo en pequeñas tareas distribuidas entre los diferentes usuarios. En JumpWords se verá reflejado cuando varios usuarios jueguen y cada uno de ellos valide una serie de datos de entrada diferentes, así el trabajo de validar una cantidad masiva de datos estará distribuido entre todos los usuarios.
- Gamsourcing es una técnica de aprendizaje que hace que la experiencia del usuario sea más divertida e interactiva. Esto se conseguirá en JumpWords añadiendo sistemas de puntos por cada validación correcta, aumentar el nivel del juego, etc.

### 1.3.2 Implementar el juego

Una vez examinados los diferentes juegos similares que hay actualmente y los paradigmas que utilizaremos para diseñar el juego, ya podemos implementarlo.

Para implementarlo de forma correcta primero se definirá toda la documentación para saber las diferentes modificaciones que se tienen que realizar cuando sea necesario. Este documento, tiene que guardar todos los requisitos, tanto funcionales como los no funcionales.

### 1.3.3 Testear el juego

El testeo del proyecto irá en primer lugar relacionado con el desarrollo del juego en la parte de transcribir los datos de entrada, una vez acabado irá destinado a los usuarios. Dichos usuarios probaran el juego para dar feedback sobre su experiencia que se tendrá en cuenta para modificar algunos aspectos relacionados con la jugabilidad.

## 2 ESTADO DE JUEGOS SIMILARES

En este apartado se explicará el estado actual de los juegos relacionados con el proyecto a realizar, ya que hay juegos similares, pero no acaban de llamar la atención del usuario.

Nos vamos a centrar en el estado de dos juegos proporcionados por la tutora del proyecto, Alicia Fornes, y otros juegos más populares.

- Difference Game for Validation of Clusters [3]: tiene como entrada una serie de imágenes y el usuario tiene que dar como respuesta si pertenecen al mismo clúster. Un ejemplo de este primer juego lo podemos ver en la *ilustración 1*.



Ilustración 1: Difference Game for Validation of Clusters

- Match Game for Validation of Transcriptions [3]: parecido al juego anterior, pero este tiene como entrada una serie de imágenes, todas diferentes, y varias respuestas de la transcripción de esas imágenes. El usuario deberá relacionar cada imagen con su transcripción. Un ejemplo de este segundo juego lo podemos ver en la *ilustración 2*.



Ilustración 2: Match Game for Validation of Transcriptions

- DigiTalkoot<sup>[6]</sup>: es un juego de Facebook con el objetivo de validar las palabras que le van llegando. Para validarlas el usuario tiene que responder si esa palabra transcrita coincide con la palabra de la imagen. Podemos ver un ejemplo en la *ilustración 3*.

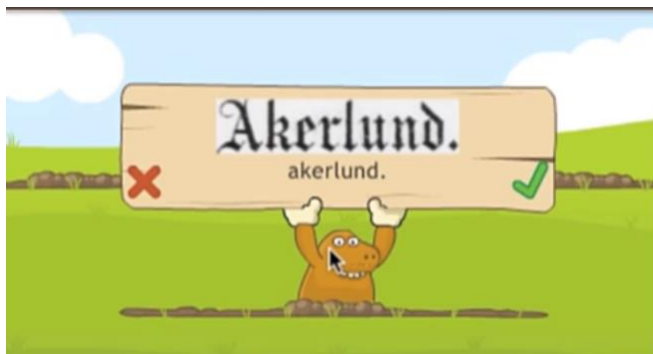


Ilustración 3: DigiTalkoot

- Palabras Cruz: es un juego para Android que consiste en escribir palabras en un tablero con las letras que te proporcionan, según en el nivel que estés. Cada vez que completas un tablero, pasas a un nivel superior donde aumenta la dificultad. Podemos ver un ejemplo en la *ilustración 4*.



Ilustración 4: Palabras Cruz

Estos juegos, menos Palabras Cruz, tienen como principal finalidad validar dichas imágenes que van apareciendo, pero resulta bastante aburrido, ya que la experiencia del usuario es monótona. Esto quiere decir que estos juegos no llaman suficientemente la atención del usuario y solo jugaría un rato y lo dejarían. El objetivo principal de JumpWords es diseñar un juego que atraiga más al usuario incorporando nuevas funcionalidades.

### 3 ANÁLISIS Y DISEÑO

#### 3.1 Solución propuesta

La solución propuesta para este proyecto es realizar un juego similar a los explicados anteriormente y que los tenemos como referencia, pero añadiendo muchas más jugabilidad con el principal objetivo de seguir validando los datos de entrada y aumentando la experiencia del usuario incorporando nuevas funcionalidades como un sistema de puntos, pasar a nuevos niveles, que este el juego recreado en un mapa, etc.

El diseño del proyecto se puede dividir en dos partes, la parte más enfocada a las validaciones de las imágenes que van apareciendo transcribiéndolas y por otra parte la de la jugabilidad y experiencia del usuario.

#### 3.2 Metodología

La metodología que se utilizará para desarrollar el proyecto será el modelo incremental, ya que este modelo tiene como objetivo un crecimiento progresivo de las funcionalidades. Es decir, el proyecto va evolucionando con cada una de las entregas.

Esta metodología, nos permitirá el desarrollo iterativo que facilitará la incorporación de nuevas funcionalidades en cada iteración que realicemos del proyecto o juego.

#### 3.3 Análisis de requisitos

A continuación, se explican los requisitos del juego, tanto los funcionales como los no funcionales.

##### 3.3.1 Requisitos funcionales

- El sistema debe tener un menú principal.
- El sistema debe tener un menú de ayuda.
- El usuario debe registrarse.
- El usuario debe poder elegir la dificultad.
- El usuario tiene que poder saltar cuando toque la pantalla.
- El usuario tiene que ir cogiendo palabras.
- Si el usuario coge palabras correctas, el sistema irá sumando puntos.
- Si el usuario coge una palabra incorrecta, el sistema restará una vida.
- Si el usuario se queda sin vidas, el sistema finalizará la partida.
- El usuario puede parar la partida en cualquier momento.
- El sistema debe mostrar un ranking tras finalizar una partida.
- El usuario debe ir aumentando de nivel.

### 3.3.2 Requisitos no funcionales

- El sistema debe actualizar el sistema de puntos y vidas en menos de 1 segundo.
- El personaje principal realizará un salto acorde con la dimensión de los obstáculos y donde se encuentre la palabra.
- El sistema únicamente funcionará si hay conexión a internet
- El menú principal deberá ser intuitivo para el usuario.
- El sistema debe ser multi-idioma.
- La velocidad de la partida debe ser baja para que haya una mejor experiencia para el usuario.
- El sistema debe funcionar en dispositivos Android.
- El sistema debe funcionar en dispositivos con la versión Android 8.0.0.
- El sistema deberá estar conectado a un servidor.

- `GetImagesURL`: aquí descargamos las imágenes a partir de la URL descargada anteriormente en la clase `GetImages`. Estas imágenes se descargan automáticamente en los assets del proyecto, es decir, el directorio donde encontramos todas las imágenes que se muestran en el juego. Para mostrar mejor las imágenes de las palabras descargadas, se descargan en diferentes carpetas, una para cada clúster, y dentro están las palabras que pertenece a ese clúster. En la siguiente imagen se puede ver como se guardan las imágenes en el proyecto.

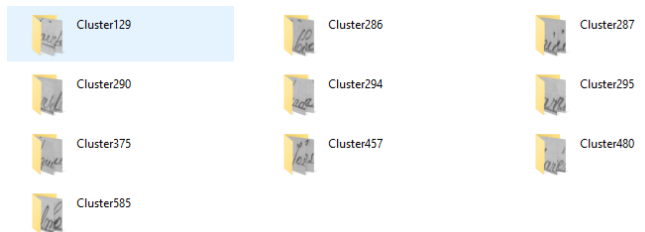


Ilustración 6: los directorio donde se guardan las imágenes está estructurado por clústeres

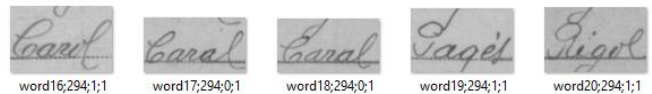


Ilustración 7: imágenes del clúster 294

## 3.4 Estructura del juego

### 3.4.1 Diagrama de clases

En la siguiente imagen se puede ver el diagrama de clases del sistema, describe una vista estática, los atributos y el comportamiento.

Las clases más relevantes en la implementación del juego son las siguientes:

- `GetImages`: esta clase es la que establece la conexión con el servidor y descarga <sup>[8]</sup> en un archivo `.txt` todas las imágenes con su URL para poder descargarlas posteriormente.

```
images/SantaColomaCervello/1932/1932_0031_Column_4_Line_2
_Word2.png;294;1;1<br>
images/SantaColomaCervello/1932/1932_0018_Column_4_Line_26
_Word3.png;294;1;1<br>
images/SantaColomaCervello/1932/1932_0012_Column_4_Line_26
_Word3.png;294;0;1<br>
images/SantaColomaCervello/1932/1932_0005_Column_4_Line_15
_Word3.png;294;1;1<br>
images/SantaColomaCervello/1932/1932_0012_Column_4_Line_24
_Word2.png;294;0;1<br>
```

Ilustración 5: TXT donde se descargan las imágenes

Como se puede observar en la imagen anterior, cada imagen tiene 4 parámetros divididos por un punto y coma:

- o <URL de la imagen>
- o <Clúster al que pertenece la imagen>
- o <Bit que indica si la palabra es diferente o no>: 1 significa que la palabra es diferente y 0 que no lo es.
- o <Bit para indicar si la palabra es de control o no>: 1 significa que la palabra es de control y 0 que no lo es.

- **GameScreen:** es la clase más importante, donde se ejecutan todas las funcionalidades del juego. Crea y muestra todos los actores que participan, implementa las colisiones del personaje principal con las palabras, muestra las imágenes descargadas, etc.

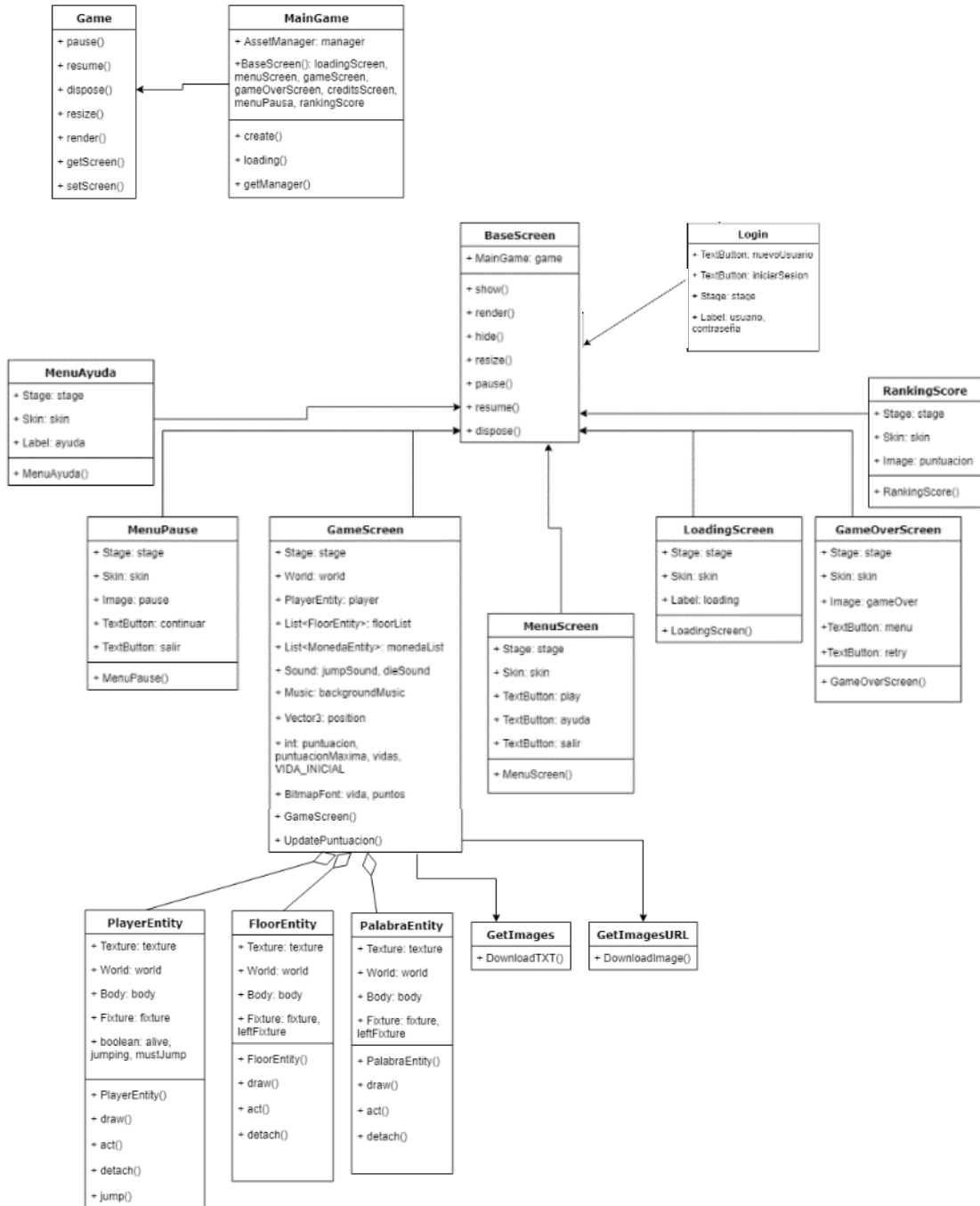


Ilustración 8: Diagrama de clases

### 3.4.2 Diagrama de casos de uso

A continuación, se puede ver el diagrama de casos de uso del juego, donde está representada la interacción del usuario con el sistema.

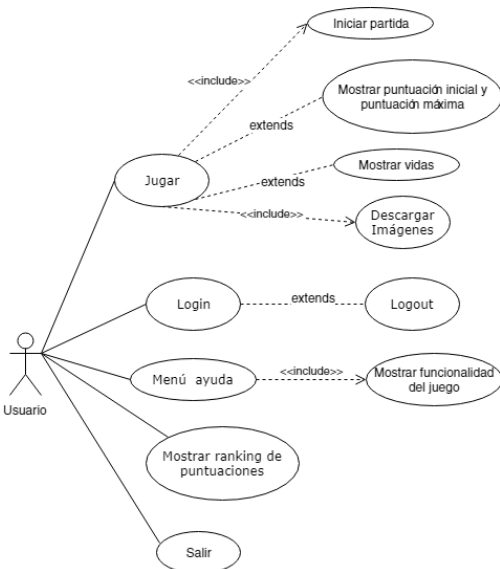


Ilustración 9: Diagrama de casos de uso

### 3.4.3 Diagrama de secuencia

En este apartado se puede observar el diagrama de secuencia del juego representado en la ilustración 11, que tiene el objetivo de describir el comportamiento dinámico del sistema.

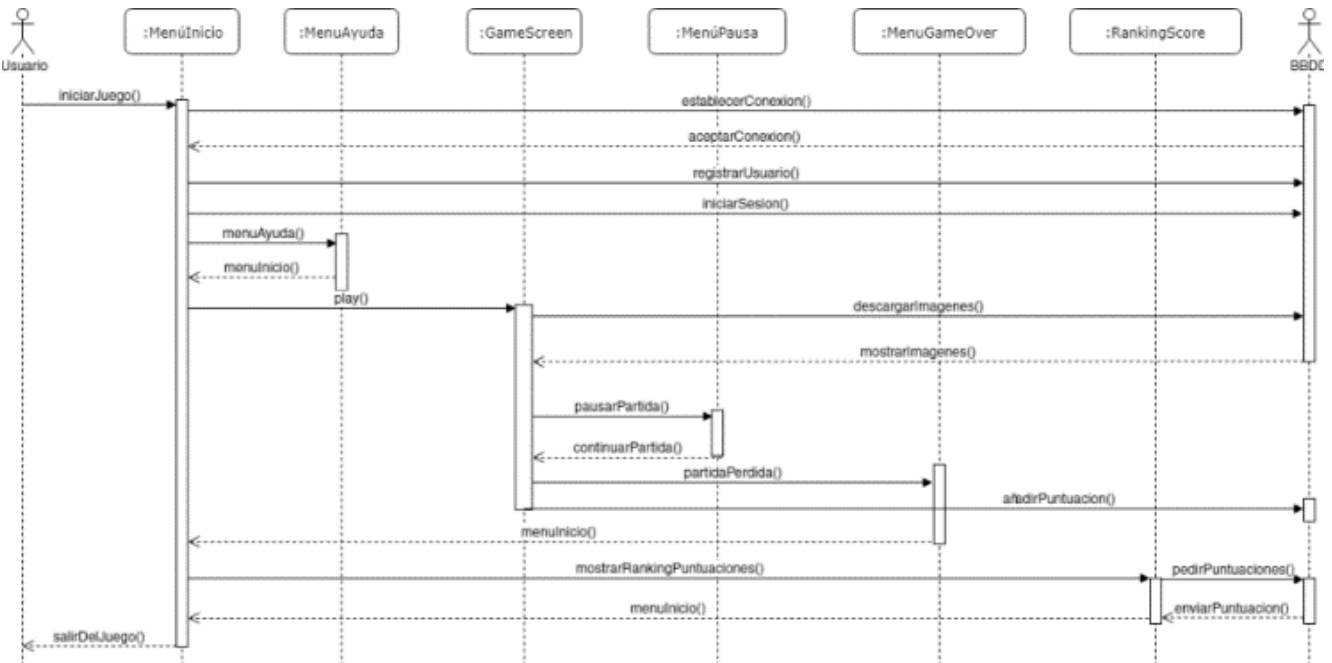


Ilustración 10: Diagrama de secuencia

### 3.4.4 Conexión con el servidor

Una vez se ha realizado la jugabilidad del juego, llega el momento de establecer la conexión con el servidor, de donde hay que descargar las imágenes que irán apareciendo durante la partida.

Para realizar la conexión, principalmente se han utilizado dos clases ya mencionadas en el diagrama de clases anterior, `GetImages` y `GetImagesURL`.

`GetImages` se encarga de realizar la primera conexión con el servidor. Para conectarse con el servidor utiliza la función llamada `download`, que realiza la conexión y descarga el contenido de la URL en un archivo `.txt`.

Una vez descargado todo el contenido en el archivo `.txt`, la clase `GetImagesURL` se encarga de abrir el archivo y parsearlo para obtener únicamente el primer parámetro, la url para descargar cada imagen.

`images/SantaColomaCervello/1932/1932_0029_Column_4_Line_9_Word3.png;`

Ilustración 11: URL de una de las imágenes descargadas

Una vez tiene todas las url mediante un bucle, descarga cada imagen y las guarda en la carpeta `assets`.

Cuando ya tenemos todas las imágenes descargadas, la clase `MainGame` se encarga de cargarlas en el juego antes de ser llamadas por la clase `GameScreen`, donde se ejecuta toda la funcionalidad del juego.

## 4 DESARROLLO

A lo largo de este apartado se realizará un seguimiento del desarrollo que se ha realizado.

El desarrollo del proyecto está dividido en 4 fases:

- Entorno utilizado: el entorno utilizado es Android Studio, junto con las librerías y funcionalidades que ofrece LibGDX [1].
- Diseño: estructura y jugabilidad que tendrá el juego.
- Implementación: implementar las diferentes funcionalidades que tendrá el juego.
- Testing.

### 4.1 Entorno utilizado

El desarrollo del juego se ha realizado en Android Studio, que permite realizar aplicaciones para dispositivos Android.

Sobre el desarrollo del juego, más concretamente en el tema de las funcionalidades y la jugabilidad, se ha utilizado LibGDX [2].

LibGDX es un framework para el desarrollo de juegos multiplataforma escritos en el lenguaje de programación Java con una mezcla de C/C++ para dar soporte y rendimiento a tareas relacionadas con la física y procesamiento de audio para dispositivos actuales. Ofrece diferentes módulos para utilizar gráficos, audio y entradas del usuario de una manera más sencilla.

Las ventajas más importantes que nos ofrece LibGDX [5] son las siguientes:

- Soporta 2D y 3D.
- Tiene mucha documentación, tutoriales y ejemplos de código.
- Antes de probar la aplicación en el dispositivo Android se puede probar en el escritorio.
- Maneja Audio, inputs del usuario y físicas.

Por otra parte, como he dicho antes LibGDX [7] utiliza JAVA como lenguaje de programación. JAVA se ha utilizado para implementar diferentes funcionalidades, entre ellas:

- Establecer la conexión con el servidor para descargar las imágenes que se mostrarán en el juego.
- Permitir a un usuario iniciar sesión o registrarse.
- Enviar los resultados adquiridos.

## 4.2 Diseño del juego

### 4.2.1 Explicación del juego

JumpWords es en un juego para dispositivos Android que consiste en validar las diferentes palabras que van apareciendo, tanto si son correctas o incorrectas.

El juego está basado en el famoso juego Jump Don't Die, donde el personaje principal tiene que sobrevivir mientras avanza por el mapa, para sobrevivir tiene que saltar obstáculos.

En este caso el personaje deberá ir validando las diferentes palabras que le van apareciendo por el mapa. Al usuario le irán apareciendo diferentes clústeres con sus respectivas palabras.

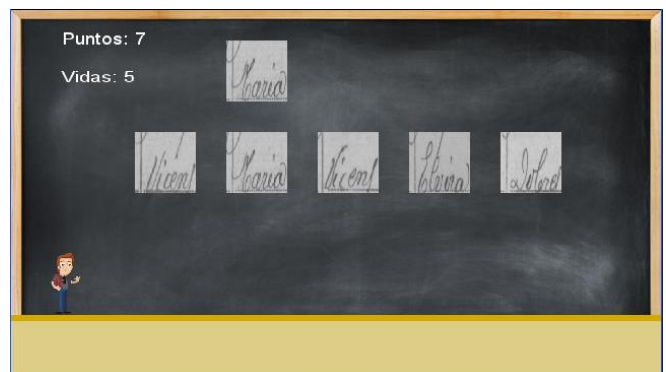


Ilustración 12: inicio de partida donde el personaje principal tiene 5 vidas y los puntos que va adquiriendo

En los clústeres que se muestran, hay una palabra de control y varias palabras que están dentro de ese clúster. Estas palabras pueden ser correcta, es decir, que son iguales a la palabra de control; o incorrectas, que son diferentes a la palabra de control. El usuario deberá escoger la transcripción correcta de la palabra de control, de esta manera irá sumando puntos. Si el usuario coge una palabra incorrecta, una transcripción errónea, irá restando vidas.

Antes de iniciar la partida aparecerá un menú principal donde el usuario puede escoger lo siguiente:

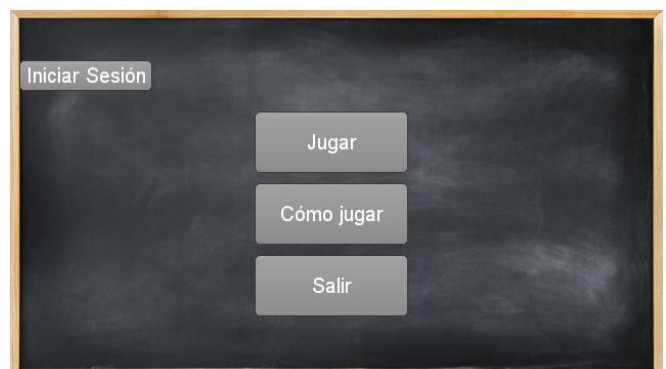


Ilustración 13: Menú principal

- Iniciar Sesión: el usuario puede iniciar sesión o crear un nuevo usuario. Tendrá que introducir un usuario y una contraseña para poder iniciar sesión o crear un nuevo usuario.
- Jugar: si el usuario escoge jugar, empieza la partida. Para jugar no hace falta que el usuario inicie sesión obligatoriamente.
- Ayuda: mostrará un menú donde explica el funcionamiento del juego.

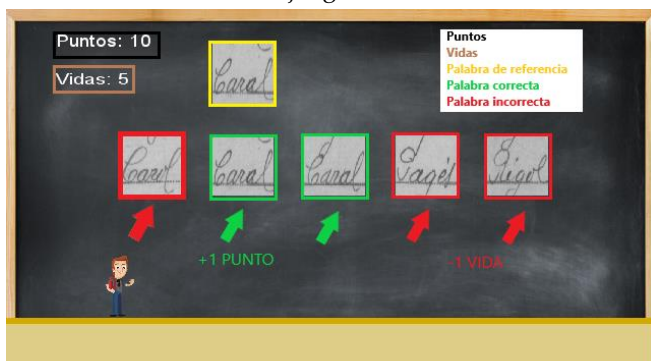


Ilustración 14: Menú ayuda

- Salir: se cerrará la aplicación.

Hay que destacar que la aplicación es multi-idioma y universal, se mostrará en el idioma en el que esté el dispositivo Android, es decir, si el idioma del dispositivo Android está en inglés, la aplicación se mostrará en inglés automáticamente.

Si el usuario empieza la partida, le irán apareciendo los diferentes clústeres, con la palabra de control y las que pertenecen a ese clúster.

Cuando el usuario inicie la partida tendrá 5 vidas y 0 puntos. Cada vez que el usuario coja una palabra correcta de ese clúster, es decir, que la palabra sea igual a la de control, se irá actualizando el marcador de puntos automáticamente. Cada palabra correcta sumará un punto en el marcador.

De otro modo, si el usuario coge una palabra incorrecta, se actualizará automáticamente el marcador de vidas, restándole una.

El usuario puede parar la partida en cualquier momento y volver a reanudarla.

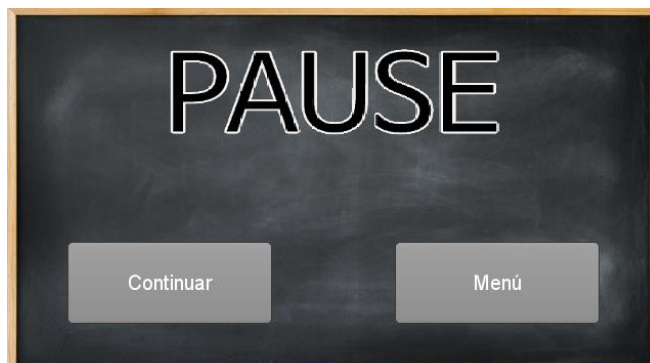


Ilustración 15: Menú pausa

Si el usuario se queda sin vidas, la partida finalizará y se mostrará un nuevo menú en la pantalla Game Over, donde el usuario deberá tocar la pantalla para volver al menú principal.

### 4.3 Implementación

En este apartado se explicarán los puntos más críticos que ha habido a lo largo del desarrollo de la aplicación.

El primer punto crítico ha sido implementar la funcionalidad de eliminar las palabras cuando el usuario la toca. Las primeras implementaciones fueron erróneas, ya que cuando intentaba eliminar una sola palabra, desaparecían todas. Consultando la wiki de LibGDX, entendí como tratar las texturas de las imágenes y como eliminarlas, implementando la funcionalidad correctamente.

Uno de los puntos más críticos ha sido la conexión con el servidor, ya que al principio la conexión no se establecía correctamente. Recibí ayuda técnica por parte de la persona que ha desarrollado la parte del servidor y la conexión se efectuó de forma correcta. Una vez tuve la conexión con el servidor establecida, la función de descargar las imágenes la implementé con facilidad.

### 4.4 Testing

Toda aplicación se tiene que testear antes de su puesta en marcha y, por lo tanto, este proyecto también ha recibido una serie de pruebas.

Una vez finalizada la funcionalidad del juego se ha comprobado que todos los menús de la aplicación funcionan correctamente y hacen lo que el usuario desea.

Finalmente, se ha evaluado que las imágenes se eliminan cuando el usuario las toca y que suma o resta puntos cuando toca una palabra correcta o incorrecta.

## 5 PRESENTACIÓN Y DISCUSIÓN DE LOS RESULTADOS

Una vez finalizado el análisis y desarrollo de toda la aplicación, se han determinado unos resultados que se mostrarán a continuación.

Primero de todo, hay que destacar que se ha conseguido el objetivo principal, el cual era desarrollar un juego basado en los paradigmas de crowdsourcing y gamificación.

Se ha conseguido desarrollar un juego interactivo, donde el usuario puede validar todos los clústeres que van apareciendo en la partida. Hasta el momento, se han cumplido la mayoría de los requisitos establecidos al principio del desarrollo de la aplicación.

En un futuro, se seguirá trabajando para acabar de introducir nuevas funcionalidades que quedan por hacer, por ejemplo, que el usuario vaya aumentando niveles en la misma partida o que pueda pausar la partida en cualquier momento y continuar en la misma.

Para finalizar se ha seleccionado una serie de usuarios de diferente edad para que prueben la aplicación y den su opinión al respecto.

La edad de los usuarios se ha dividido en tres rangos:

- 20 años o menos.
- 20 a 40 años.
- 40 años o más.

Los usuarios de 20 años o menos, comparados con los usuarios de 40 años máximo, las opiniones han sido similares. Los aspectos negativos que han encontrado en JumpWords son referentes a la visualización de las palabras, ya que algunas no se entienden bien.

Por lo que respecta a los aspectos positivos, han observado que JumpWords ayuda a aumentar la concentración a la hora de jugar, te anima a seguir jugando una vez has acabado una partida y el menú principal es intuitivo. Unos de los comentarios que me ha impactado ha sido que la aplicación se puede utilizar como práctica para aumentar la concentración en personas que tengan déficit de atención.

Para finalizar, mejorarían la información del menú de ayuda añadiendo una explicación sobre los niveles y cambiar de música según el nivel en el que estés, es decir, si el nivel es más alto, la música debería ir más rápida.

Los usuarios de más de 40 años, tienen una opinión diferente a los anteriores, ya que tienen más dificultades para jugar. Los aspectos negativos que han observado son referentes a la velocidad inicial de la partida y la visualización de las palabras.

En los aspectos positivos, coinciden en la mejora de la concentración al fijarse en las palabras. También han destacado que JumpWords es útil para reforzar la memoria ya

que tienen que memorizar la palabra de referencia para coger las correctas.

Para finalizar, mejorarían la visualización de las imágenes y la velocidad de la partida.

En el anexo se pueden ver dos gráficos sobre las opiniones de los usuarios respecto a JumpWords una vez probado. El primer gráfico<sup>Gráfico1</sup> corresponde a los aspectos positivos, he seleccionado los más comentados por los usuarios. Como se puede ver he calculado el porcentaje de los usuarios que han seleccionado cada aspecto.

El segundo gráfico<sup>Gráfico2</sup> corresponde a los aspectos negativos comentados por los usuarios. Como en el gráfico anterior, se puede ver el porcentaje de usuarios que ha seleccionado cada aspecto negativo, y por lo tanto a mejorar.

Las opiniones de los usuarios se tendrán en cuenta para mejorar JumpWords en un futuro implementando nuevas funcionales y aumentar la jugabilidad.

## 6 CONCLUSIONES

Una vez realizada toda la implementación del juego y testado, se han consultado todos los resultados obtenidos para formular las conclusiones del proyecto.

El objetivo principal era desarrollar un juego Android para transcribir/validar datos de reconocimiento automático de textos manuscritos, basado en los paradigmas de crowdsourcing y gamificación.

Analizando el objetivo principal, se puede decir que el objetivo del proyecto se ha cumplido, aunque en un futuro hay que seguir trabajando para acabar de implementar las funcionalidades que faltan y mejorar la aplicación.

También hay que destacar que se ha hecho un seguimiento correcto de la planificación del proyecto respetando todas las entregas. El modelo incremental ha sido una gran opción para poder desarrollar el proyecto correctamente, incluso con las dificultades que han ido surgiendo en ciertos momentos del desarrollo.

## AGRADECIMIENTOS

Como punto final del proyecto, me gustaría agradecer la visión aportada por Alicia Fornes, tutora del proyecto, para el correcto desarrollo del proyecto.

Y a mi compañero Jialuo Chen, por su ayuda a la hora de resolver dudas para establecer la conexión con el servidor, ya que fue quien lo desarrolló.

## 7 BIBLIOGRAFÍA

[1] Santiago Faci. (2016). Desarrollo de videojuegos con libGDX. 2016, de Programación Multimedia y Dispositivos Móviles Sitio web: <http://multimedia.codeandcoke.com/apuntes:libgdx>

[2] Jose Javier Villena. (2016). Programación de Android desde cero. 2016, de UdeMy Sitio web: <https://www.udemy.com/home/my-courses/learning/>

[3] Jialuo Chen, Pau Riba, Alicia Fornes, Joan Mas & Josep Lladós. (2018). Word-Hunter: A Gamesourcing Experience to Validate the Transcription of Historical Manuscripts. 2018, de Computer Vision Center Sitio web: [http://www.cvc.uab.es/people/afornes/publi/conferencias/2018\\_ICFHR\\_JChen.pdf](http://www.cvc.uab.es/people/afornes/publi/conferencias/2018_ICFHR_JChen.pdf)

[4] Alicia Fornés. (2018). XARXES: Tecnologia i innovació ciutadana en la construcció de xarxes socials històriques per a la comprensió del llegat demogràfic. 2018, de Xarxes Sitio web: <http://dag.cvc.uab.es/xarxes/info/>

[5] Javier Osuna Herrera. (2014). Tutorial de libGDX. Parte 5: Colisión con la bola. 2014, de Youtube Sitio web: <https://www.youtube.com/watch?v=1gUgkY3gzz8>

[6] microtaskblog. (2011). How to play Mole Hunt. 2011, de Youtube Sitio web: <https://www.youtube.com/watch?v=G7gXkdSPXWQ>

[7] Anatoly Ivanov. (2018). LibGDX. 2018, de libgdx Sitio web: <https://github.com/libgdx/libgdx/wiki>

[8] NathanSweet. (2014). Download Image. 2014, de Github LibGDX Sitio web: <https://github.com/libgdx/libgdx/blob/master/tests/gdx-tests/src/com/badlogic/gdx/tests/TextureDownloadTest.java>

[9] Victor Cuervo. (2008). Borrar un directorio con Java. 2018, de Lineadecodigo Sitio web: [lineadecodigo.com/java/borrar-un-directorio-con-java/](http://lineadecodigo.com/java/borrar-un-directorio-con-java/)

[10] Víctor Cuervo. (2007). Crear un directorio con Java IO. 2018, de Lineadecodigo Sitio web: [lineadecodigo.com/java/crear-un-directorio-con-java-io/](http://lineadecodigo.com/java/crear-un-directorio-con-java-io/)

[11] rosettacode. (2018). Walk a directory/Recursively. 2018, de rosettacode Sitio web: [rosettacode.ord/wiki/Walk\\_a\\_directory/Recursively#Java](http://rosettacode.ord/wiki/Walk_a_directory/Recursively#Java)

[12] Boann. (2014). Java - sending HTTP parameters via POST method easily. 2019, de stackoverflow Sitio web: <https://stackoverflow.com/questions/4205980/java-sending-http-parameters-via-post-method-easily>

## 8 ANEXO

- Código fuente del proyecto.
- Wiki LibGDX.
- Diagramas utilizados.
- Funciones implementadas.

### Gráficos de los resultados obtenidos

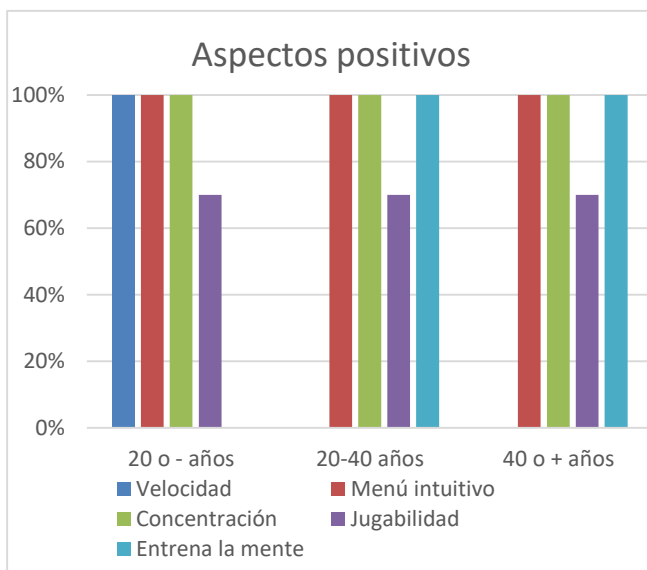


Gráfico 1: aspectos positivos comentados por los usuarios

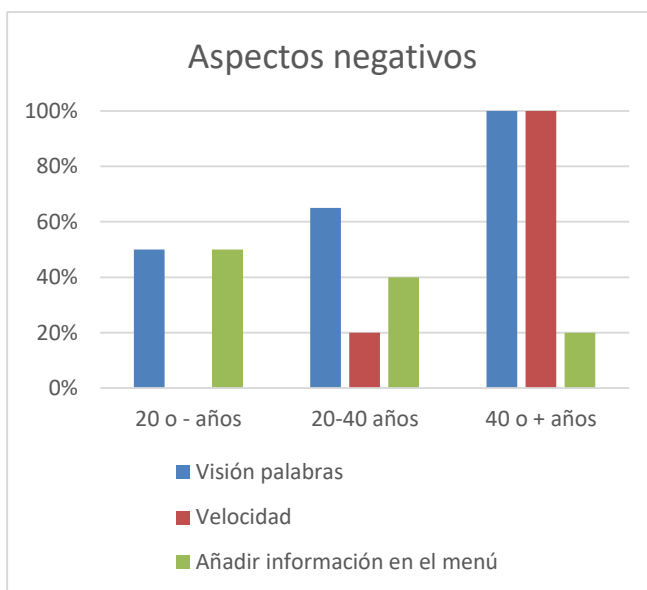


Gráfico 2: aspectos negativos comentados por los usuarios