Jon Fernández Huerta

Salesianos Nuestra señora del pilar  C. de María Auxiliadora, 57, 50009 Zaragoza

TFG aPLicación de minijuegos en react

Índice

[Introducción e ideas del proyecto 2](#_Toc138199683)

[Caso de uso 2](#_Toc138199684)

[Tecnologías 2](#_Toc138199685)

[Funcionalidades del Aplicativo 2](#_Toc138199686)

[Manual de usuario 3](#_Toc138199687)

[Problemas surgidos durante el transcurso del proyecto 5](#_Toc138199688)

[Errores conocidos 6](#_Toc138199689)

[Futuras implementaciones 6](#_Toc138199690)

[Conclusiones 6](#_Toc138199691)

# Introducción e ideas del proyecto

## Caso de uso

El objetivo de nuestro proyecto consiste en la construcción de una aplicación para dispositivos móviles Android que permita jugar a diversos juegos. Por una cuestión de tiempo, actualmente solo ha sido implementado el Wordle, consistente en adivinar palabras palabras con un número limitado de intentos, pero por cómo ha sido diseñada la aplicación se pueden añadir otros, como por ejemplo Buscaminas o Snake sin ningún tipo de problema.

Al margen de los propios juegos, la aplicación ofrece la posibilidad de registrar usuarios y que estos configuren los aspectos necesarios de su cuenta (como cambio de contraseña, de nombre o de avatar). Los usuarios registrados tendrán la posibilidad de registrar sus puntuaciones a medida que jueguen y compararlas con las de los demás.

## Tecnologías

En el proyecto se han usado las distintas tecnologías que nombro a continuación:

Como base para la aplicación, un proyecto de Node.js al cual se le han añadido los framework de React, React Native, Native Base.

Como base de los mini juegos, SpringBoot.

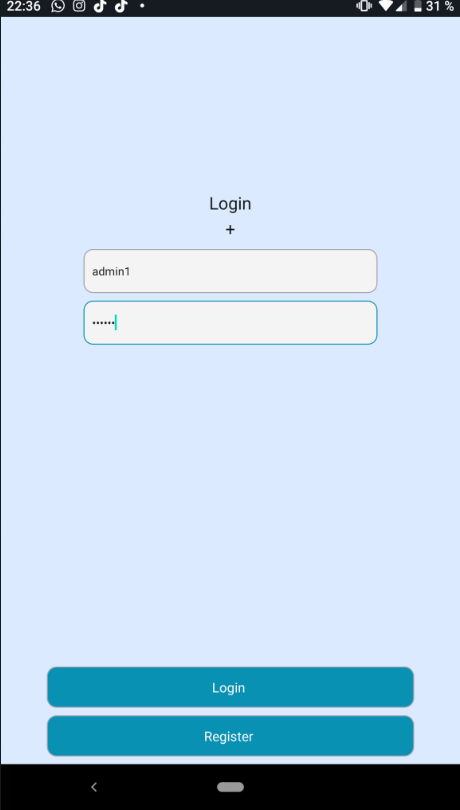
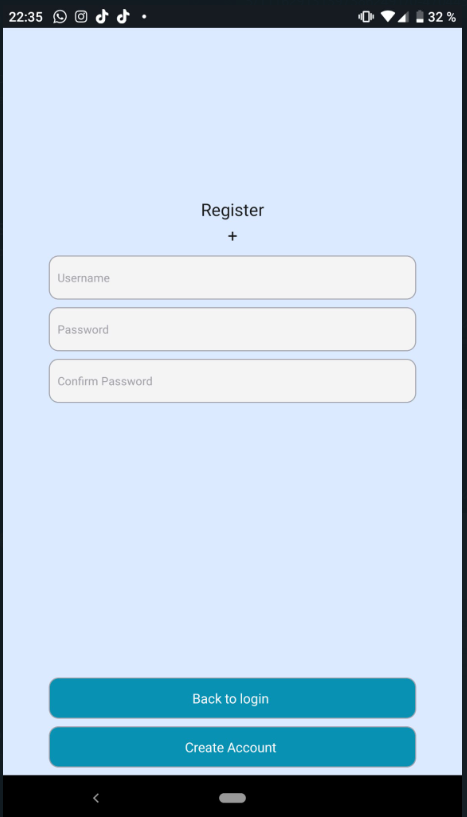
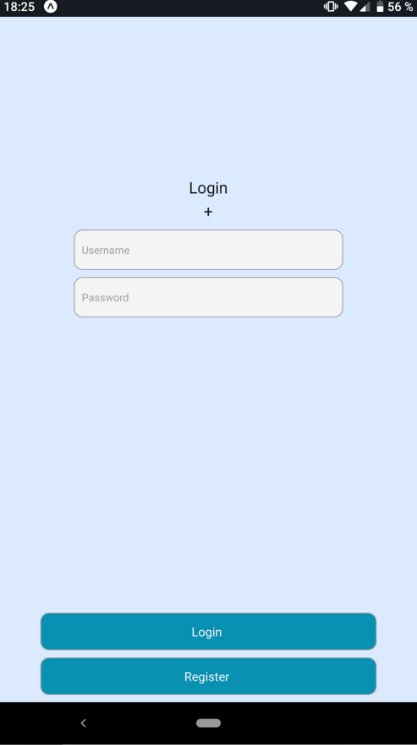
Para hacer la aplicación de móvil, Expo sobre React Native.

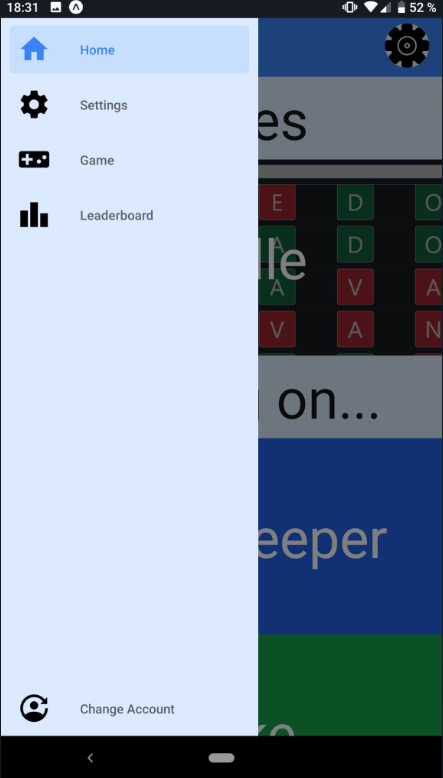
Para gestionar tanto los usuarios como las tablas de puntuación, MongoDB.

# Funcionalidades del Aplicativo

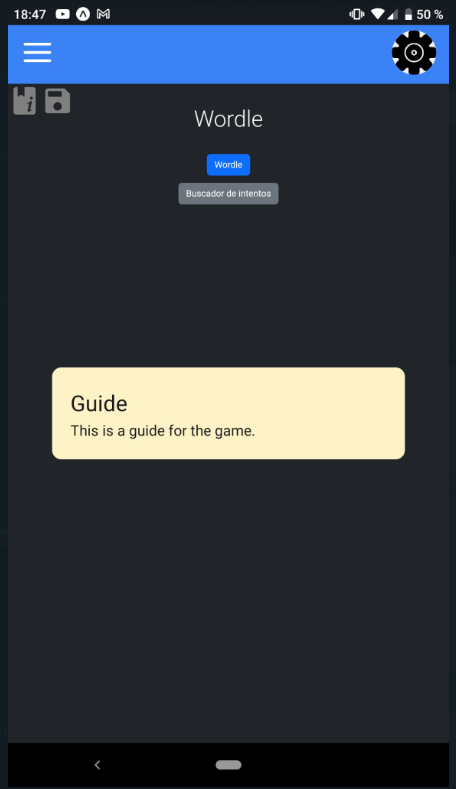
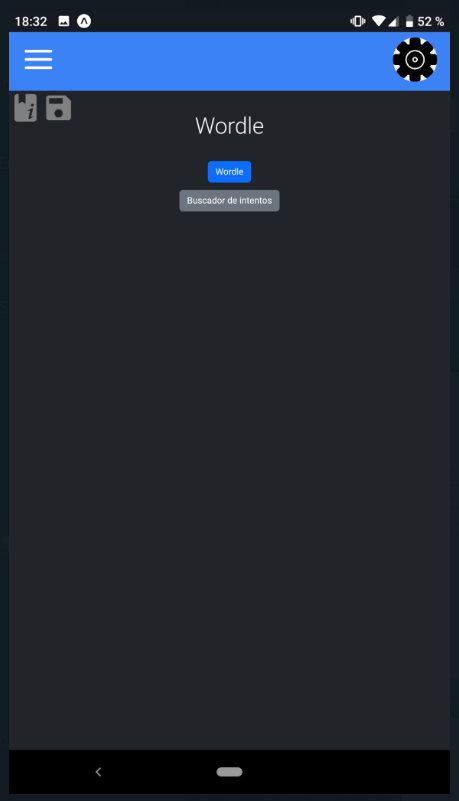
* Drawer interactivo para ver los distintos menús:
  + Home
  + Settings
  + Game
  + Leaderboard
  + Change Account
* Interfaz completamente táctil.
* Posibilidad de cambiar los datos del usuario (nombre, avatar y contraseña).
* Funcionalidad total del primer de los juegos de la aplicación.
* Actualización parcial de las puntuaciones de los usuarios.
* Creación de cuentas.
* La cuenta no expira entre sesiones de juego.
* Opción de limpieza de caché
* Customización de la apariencia de la aplicación
* Enlace a soporte de la aplicación (WIP)

# Manual de usuario

Al iniciar la aplicación se ofrece la posibilidad de iniciar sesión con una cuenta ya existente o crear una nueva.   


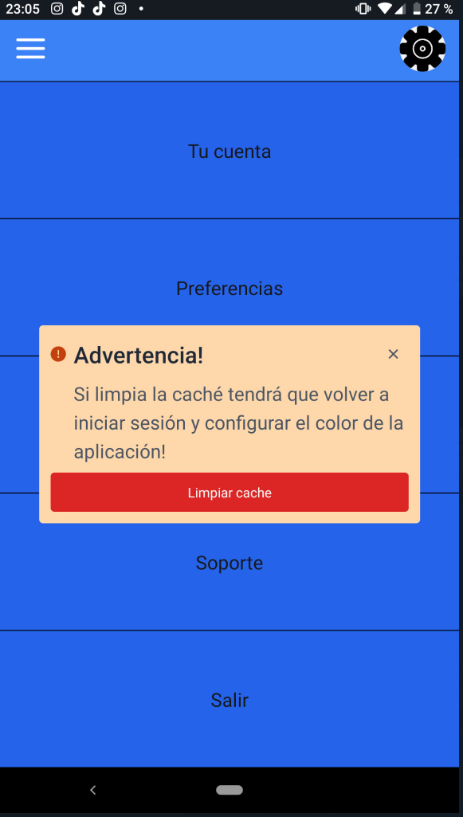
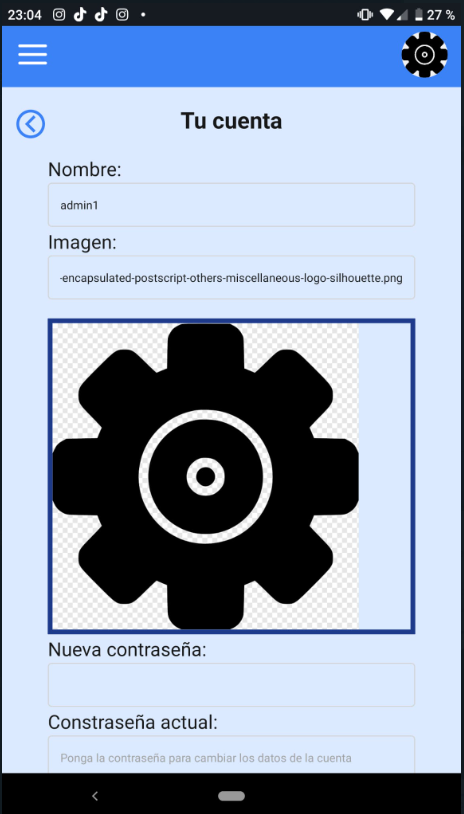
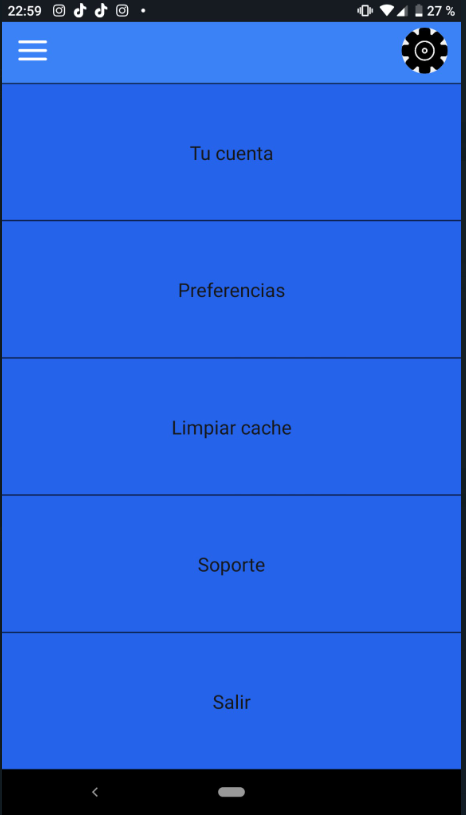
Una vez iniciada la sesión se presenta al usuario con el menú Home de la aplicación.  


Después de seleccionar el juego, se abre una pantalla que tiene la misma estructura sin importar el juego, el juego como parte principal y los iconos de guía y guardar.



El icono de guardar envía una petición a la base de datos donde se guardará la nueva puntuación.

El aspecto del menú de ajustes es el siguiente:



# Problemas surgidos durante el transcurso del proyecto

A la hora de instalar los paquetes de dependencias para utilizar el Drawer con el proyecto de Expo, surgieron errores de compatibilidad debido a que no estaba actualizado a la última versión de React.

El uso de AsyncStorage también causó bastantes errores ya que una vez se ha asignado una variable es necesario borrarla y volverla a asignar para que funcione, y es necesario hacer esperas para poder acceder a estos datos, lo cual ralentiza el uso de la aplicación.

La implementación de variables globales se intentó implementar en un primer momento mediante el AsyncStorage, lo cual resultó ser demasiado complicado, cosa que se solucionó utilizando variables de manera común entre varias pantallas con createContext. En el proyecto final, AsyncStorage solo se usa para las variables que se desea guardar entre sesión y sesión.

Tanto en la base de datos como en Node.JS no se pueden hacer relaciones entre las tablas, por lo que a la hora de actualizar los datos de múltiples tablas era necesario hacerlas manualmente en la API de MongoDB.

Al principio el método usado para declarar los estilos obligaba a repetir mucho código pero gracias a la una librería de estilos NativeBase, que permite aplicar estilos globales, la cantidad de código requerido se ha reducido significativamente.

A la hora de utilizar componentes y aplicarles estilos en las pantallas no se podía utilizar el mismo contenedor de estilos del padre para el hijo y en los componentes había que implementar los estilos manualmente.

# Errores conocidos

Al modificar el perfil en ‘Tu cuenta’ si se vacía el Input de la imagen aparece un error en pantalla porque el elemento no es capaz de leer un string vacío.

Si se abre la pestaña del juego antes de seleccionar uno en la ventana de Home te enlaza al servidor de SpringBoot sin especificar el juego y el servidor da error.

# Futuras implementaciones

* Testeo de funciones de la aplicación.
* Aplicar estilos para la lectura de datos de las leaderBoards.
* Redirección al soporte en Ajustes.
* Añadir las preferencias de customización de color de la aplicación.
* Ajustes de UI en la página de juego y Settings
* Añadir descripción y guía de los juegos.
* Proporcionar feedBack a la hora de guardar.

# Conclusiones

El proyecto ha sido una buena introducción a programar un proyecto de aplicaciones móviles Android con Expo React Native y usar bases de datos como MongoDB Atlas para la API.

Aunque el tamaño y complejidad del proyecto ha sido menor que el requerido para una aplicación de uso real, ha servido para ver muchas de las dificultades que pueden aparecer en un proyecto de este tipo y también para apreciar la cantidad de trabajo que hay que poner para sacar un producto bien acabado.

A la hora de afrontar nuevos proyectos, el haber trabajado con este creo que será de gran ayuda para trabajar de una manera más eficiente y organizada.