Desarrollo de una aplicación móvil para el aprendizaje de una lengua extranjera

Sergio Díaz Serena

Resumen– Actualmente el aprendizaje de una lengua extranjera es útil tanto para el ámbito laboral como estudiantil, donde cada vez es más importante tener el conocimiento de más de una lengua. Por ello en el presente trabajo se pretende realizar una aplicación móvil multiplataforma orientada al aprendizaje de una lengua extranjera. La aplicación dará soporte a dos tipos de roles, alumnos y profesores, donde cada usuario alumno tendrá un conjunto de palabras en el idioma a aprender, las cuales serán proporcionadas por parte de un profesor. A partir de un juego donde se irán mostrando un conjunto de palabras, los usuarios alumnos deberán de seleccionar, a partir de otro conjunto de palabras como posibles opciones, la traducción correcta de dicha palabra. El proyecto se realizará a partir de diferentes tecnologías como React Native y Firebase.

Palabras claves– React Native, Firebase, multiplataforma, API, Real-time Database, aprendizaje de idiomas, roles

Abstract– Currently learning a foreign language is useful both for the workplace and for students, where it is increasingly important to have knowledge of more than one language. For this reason, the present work aims to create a multiplatform mobile application oriented to the learning of a foreign language. The application will support two types of roles, students and teachers, where each student user will have a set of words in the language to be learned, which will be provided by a teacher. From a game where a set of words will be shown, the student users must select, from another set of words as possible options, the correct translation of said word. The project will be carried out using different technologies such as React Native and Firebase.

Keywords– React Native, Firebase, cross-plataform, API, Real-time Database, language learning, roles

INTRODUCCIÓN

El diseño de aplicaciones móviles es un mercado en auge, donde se puede observar que el sector lleva creciendo de manera exponencial desde hace bastantes años, por ello muchas empresas optan por este formato para cubrir las demandas del usuario.

La elección del trabajo viene dada por la decisión de aprender a diseñar una aplicación móvil multiplataforma a partir de la tecnología utilizada en el ámbito profesional de dicho sector.

En este documento se resumirá toda la información extraída a lo largo de la realización del proyecto, indicando los resultados de aprendizaje y las conclusiones extraídas.

El documento se distribuirá de la siguiente manera:

- Una primera sección donde se definirá el proyecto realizado y sus objetivos.
- Una segunda sección donde se explicará el estado de arte.
- Una tercera sección donde se expondrá la planificación y metología utilizada.
- Una cuarta sección donde se detallará los requisitos tanto funcionales como no funcionales.
- Una quinta sección donde se describirá y se detallará las tecnologías escogidas.
- Una sexta sección donde se describirá el diseño de la aplicación, detallando su estructura.
- Una séptima sección donde se describirá el desarrollo.
- Una octava sección donde se hablará de la fases de pruebas realizadas.
- Una novena sección donde se explicará las conclusiones extraidas, las posibles líneas de mejoras y ampliaciones.
- Finalmente, el documento acabará con unos agradecimientos, las referencias y apéndices pertinentes.

• E-mail de contacte: sergio.diazs@e-campus.uab.cat
• Menció realitzada: Tecnologies de la Informació
• Treball tutoritzat per: Enric Vergara Carreras (dEIC)
• Curs 2020/21

1 DEFINICICIÓN DEL PROYECTO

Actualmente existen multitudes de aplicaciones referidas a el aprendizaje de una lengua extranjera, no obstante, practicante ninguna de ellas permite que un usuario administrador controle el progreso de un usuario determinado, haciendo que el progreso únicamente sea controlado por la aplicación. En la realización de este proyecto se pretende crear una aplicación multiplataforma donde convivan dos tipos de roles: usuario y profesor.

La aplicación estará enfocada a aquellas organizaciones, como podrían ser centros educativos, en el que un profesor tenga a cargo un conjunto de alumnos y quiera añadir a su enseñanza una herramienta de aprendizaje a distancia. Dada la actual situación que vivimos debido a la COVID-19, esta herramienta puede ser bastante útil a más de un centro docente, ya que fomenta la educación de manera telemática y dinámica.

1.1 Objetivos

Los objetivos generales para alcanzar en el proyecto, tanto de funcionalidad como personales son:

- Conocer y valorar las alternativas dentro de mercado
- Conocer, usar y valorar el framework de React Native, incluyendo su lenguaje Native
- Realizar una aplicación funcional tanto en Android como en iOS.
- Diseñar e implementar una base de datos con la tecnología proporcionada por Firebase
- Implementar una API de traducción para palabras/frases
- Aprender y entender el mundo de desarrollo de aplicaciones móviles.

1.2 Funcionalidad

La finalidad principal de la aplicación será que un estudiante, a través de un juego, vaya repasando un conjunto de palabras específicas para no olvidarlas.

Dicha aplicación estará compuesta por dos roles, el profesor y el alumno. Cada profesor tendrá asignado una cantidad especifica de alumnos, que él mismo dará de alta a través del aplicativo. El profesor podrá asignar manualmente palabras específicas a cada alumno, dichas palabras serán introducidas en el idioma nativo del estudiante y a través de una API se buscará la traducción. De todas formas, el profesor podrá verificar y modificar, en caso de ser necesario, la traducción encontrada por la API.

La finalidad del estudiante será a partir de una palabra en su idioma nativo encontrar su traducción. Como posibles respuestas se mostrarán un conjunto de palabras, entre las cuales se encuentra la traducción correcta. Como punto de partida a la hora de mostrar las posibles respuestas, se utilizarán aleatoriamente palabras del mismo conjunto de palabras que tenga asignado un estudiante.

Tanto estudiante como profesor tendrán acceso a el informe de progreso de cada palabra, mostrándose la ratio aciertos/intentos.

2 ESTADO DEL ARTE

Cuando se quiere desarrollar una aplicación móvil, una de las primeras cosas a realizar es elegir el lenguaje y framework a utilizar. Según la plataforma destino a la que vaya destinada la aplicación existe un lenguaje de programación predominante. En Android se trata de Java y en iOS Swift [1].

Cuando hablamos de aplicaciones cross-plataform son aquellas que han sido diseñadas en un solo lenguaje y que luego pueden ser compiladas y ejecutadas en cualquier plataforma. El lenguaje predominante es sin duda JavaScript, que le acompañan una infinidad de frameworks a utilizar.

Un framework se podría describir como un esqueleto, una estructura previa que se puede aprovechar para el desarrollo de un proyecto. Los frameworks nos ayudan a evitar introducir código repetitivo y a desarrollar mucho más rápido un proyecto [2]. Uno de los frameworks más utilizados actualmente es React Native, desarrollado por la compañía Facebook y utilizado tanto para el desarrollo de aplicaciones móviles como para la creación de páginas web.

Otra tecnología fundamental cuando se desarrolla una aplicación móvil, en la cual se tiene que hacer una cierta gestión de datos, son las bases de datos. Una base de datos se puede definir como una colección de información organizada de forma que un programa puede seleccionar los datos que necesite para realizar cualquier tarea. A la hora de realizar proyectos normalmente se suelen utilizar base de datos relacionales escritas en lenguaje SQL (Structured Query Language), ya que se considera el lenguaje estándar [3], no obstante, existen otro tipo de tecnologías ajenas a las tradiciones, como por ejemplo Firebase, una plataforma en la nube desarrollada por Google. Firebase nos aporta unas ciertas ventajas en comparación a una base de datos tradicional, la más destacada es el Real-Time Database¸es decir, utiliza el procesamiento en tiempo real para manejar los datos cuyo estado cambia constantemente. Cabe destacar que Firebase es una base de datos del tipo NoSQL (not only SQL) que almacena los datos en la nube en formato JSON.

3 METODOLOGÍA Y PLANIFICACIÓN

Durante el proceso de creación de la aplicación, se seguirá la metodología típica de un ciclo de vida de desarrollo de un software, descrito en la figura 1. Por lo tanto, se realizarán diferentes pruebas durante el proceso de desarrollo para comprobar su correcto funcionamiento, y en caso de ser necesario volver a configurar algunas secciones o módulos.

Lo que concierne a la planificación del trabajo se ha realizado a partir de un conjunto de tareas divididas en diversas semanas, para así obtener un mayor rendimiento y control sobre ellas. En el anexo A1 se puede visualizar una tabla con las fechas, tareas y subtareas realizadas a lo largo del proyecto.

Fig. 1: Ciclo de vida de desarrollo de un Software [4]

4 Requisitos

4.1 Requisitos Funcionales

- **Registro de un usuario profesor enviando la información a la base de datos de Firebase.** Cuando un usuario se desea registrar con el rol de profesor, deberá de rellenar un formulario con la información pertinente, como el correo electrónico y la contraseña.
- **Login de usuario.** A partir de un formulario de acceso donde se pedirá el correo electrónico y la contraseña, un usuario accederá a su pantalla de inicio correspondiente (en función de su rol).
- **El sistema controlará el acceso y lo permitirá solamente a usuarios registrados.** La aplicación no deberá de dejar acceder al sistema si un usuario no está registrado. El control lo realizará Firebase.
- **Restablecer contraseña.** En caso de que un usuario no recuerde su contraseña, a partir de la pantalla de inicio tendrá una opción disponible para dicho uso. Al seleccionarla deberá de introducir su correo electrónico y automáticamente le llegará un correo a su bandeja de entrada donde podrá restablecer la contraseña.
- **Acceso y modificación de los datos de perfil.** A partir de una pantalla, similar a la de registro de usuario, un usuario podrá modificar todos los datos relacionados con su usuario a partir de un formulario, incluyendo una imagen de perfil.
- **Registro de un usuario alumno enviando la información a la base de datos de Firebase.** La creación de un usuario alumno únicamente podrá ser gestionada por un usuario profesor, del mismo modo, cuando un usuario profesor dé de alta un usuario alumno, le será asignado automáticamente a él.
- **Añadir palabras a un usuario alumno determinado.** Un usuario profesor encargado de un alumno particular podrá añadirle palabras/frases a partir de un formulario donde se pedirá una palabra y la aplicación le dará una traducción estimada, dando la opción de que el profesor cambie la traducción en caso de ser errónea.
- **Visualizar los usuarios alumnos y las palabras asignadas.** Un usuario profesor, deberá de visualizar todos los alumnos que tiene asignados (que ha dado de alta él mismo) como sus correspondientes palabras/frases asignadas.
- **Visualizar las palabras asignadas de un usuario alumno.** Un usuario alumno, en caso de que lo desee podrá visualizar todas las palabras que tiene asignadas, donde podrá observar la ratio de aciertos de cada una de ellas.
- **Los usuarios alumnos deberán de tener acceso al juego.** Un usuario alumno deberá de tener acceso al juego donde, a partir de sus palabras asignadas, deberá de ir acertando su traducción correcta.

4.2 Requisitos No-Funcionales

- **El sistema debe de implementar Firebase.** La base de datos de la aplicación debe de estar implementada con el servicio de Firebase.
- **API de traducción de palabras/frases.** La aplicación debe de traducir una pablara o una frase a través de una API en un tiempo inferior a 5 segundos.
- **Mensajes informativos.** El sistema debe de proporcionar mensajes de éxito o error informativos orientados al usuario final.
- **Multiplataforma.** El sistema debe de ser compatible tanto con Android como con iOS, por lo que la aplicación debe de estar desarrollada con un framework que sea multiplataforma.
- **Interfaz gráfica.** La aplicación debe de poseer un diseño *Responisve* y poseer interfaces gráficas bien definidas.

5 Tecnologías Escogidas

5.1 Back-End - Firebase

Para la realización del back-end del proyecto se ha utilizado Firebase [5] como servidor. Firebase ofrece un conjunto de herramientas dividida en diferentes módulos: desarrollo, crecimiento y monetización (análisis). Para la realización del proyecto nos centraremos en el grupo de desarrollo, también llamado Develop en Firebase.

Dentro del grupo de desarrollo, Firebase proporciona una herramienta para gestionar una base de datos en tiempo real alojada en la nube. Como se ha mencionado en el Estado de Arte, la base de datos es NoSQL y almacena los datos en formato JSON y funciona de tal manera que se sincroniza en tiempo real con cada cliente conectado, por lo que, si un usuario realiza una modificación, se notifica a la base de datos y simultáneamente es actualizada la información para el resto de los usuarios.

Cabe destacar que, si un usuario realiza cambios y no dispone de conexión a Internet, la plataforma usa una caché local en el dispositivo donde guarda estos cambios. Una vez que vuelve a tener conexión, automáticamente se sincronizan los datos locales.

A parte del servicio de Real-Time Database, en el proyecto se utilizará el servicio de autentificación proporcionado por el mismo Firebase. Permite tanto el registro propiamente dicho (mediante email y contraseña) como el acceso

a través de otros proveedores utilizando los perfiles correspondientes (por ejemplo, de Facebook o Google). Al aportar este servicio, Firebase nos facilita la creación de sistemas de autentificación, sin necesidad de que nos preocupemos en desarrollarlos, asegurándonos el acceso rápido y seguro.

Otro servicio que será necesario para el proyecto es Cloud Storage, que es el sistema de almacenamiento de Firebase, donde los desarrolladores pueden guardar los ficheros de sus aplicaciones. Este almacenamiento también sirve para tratar los archivos de los usuarios, como imágenes o videos, por ejemplo, el avatar que hayan seleccionado como imagen, que será el uso principal por el cual se utilizará este servicio.

5.2 Front-End - React Native

Para la realización del front-end del proyecto se necesitará un framework compatible con desarrollo multiplataforma. Se consideraron 3 posibles frameworks: Flutter, Ionic o React Native [6].

Realizando una pequeña búsqueda relacionada con las otras tres opciones posibles se obtuvo que Flutter se basa en el lenguaje de programación Dart (desarrollado por Google), Ionic se basa en lenguaje web (HTML, CSS y JS) y React Native en lenguaje Javascript.

Se acabo eligiendo React Native debido a que durante el curso académico 2019/2020 en la asignatura de Sistemas y Tecnologías Web se hizo un trabajo sobre React Native, realizando una primera prueba de contacto a partir de un pequeño tutorial, y dado los requisitos de la aplicación, realizarla con dicho framework era más que suficiente.

Además de utilizar React Native se optó por implementar Expo [7], un conjunto de herramientas y librerías que facilita la programación en React Native. Su finalidad es crear un puente entre el ordenador y el dispositivo móvil que estemos utilizando para ejecutar la aplicación. Expo nos proporciona un conjunto de ventajas respecto a la instalación manual del entorno de React Native, como, por ejemplo, no hace falta tener Android Studio y/o XCode, ya que puedes correr el proyecto a partir de un enlace especial que proporciona el cliente Expo o a partir de un código QR. Otra ventaja es que Expo instala los componentes necesarios para realizar un proyecto React Native sin que el usuario se tenga que preocupar de instalar manualmente el conjunto de librerías necesarias para el correcto funcionamiento (como node, openjdk, etc). Como contrapartida, limita el SDK (Kit de Desarrollo de Software) haciendo que algunos módulos nativos o librerías no sean compatibles, no obstante, dado los requisitos de la aplicación Expo será más que suficiente para su desarrollo.

6 Diseño

Cuando se empieza a diseñar una aplicación, una de las primeras cosas a realizar es un prototipado o mockup de la aplicación. Su objetivo es mostrar la parte visual del proyecto sin necesidad de escribir código, de manera que en caso de no ser validado el prototipado, se vaya modificando hasta que este esté validado por parte del cliente. En el anexo A2 se puede observar capturas del mockup realizado.

6.1 Estructura de la aplicación

La aplicación se ha diseñado con un total de 12 pantallas, las cuales no todas serán accesibles, ya que variará en función del tipo de usuario.

La primera pantalla que nos encontramos al abrir la aplicación es la de iniciar sesión, donde un usuario podrá realizar 3 acciones: entrar, registrarse o solicitar un cambio de contraseña. En caso de que el usuario ingresado sea un profesor, se le mostrará la pantalla de inicio correspondiente donde podrá realizar una serie de acciones: cerrar sesión, acceder a la pantalla de registrar un usuario alumno, acceder a la pantalla para visualizar y modificar su información de perfil, cambiar el avatar de su perfil y visualizar los usuarios que tiene a su cargo. En la figura 2 se puede observar un diagrama de flujo del comportamiento del módulo de inicio de sesión.

Respecto a la pantalla de registrar un usuario alumno, es similar a la pantalla de registrar que puede ser ingresada desde la pantalla principal de la aplicación.

En caso de seleccionar la visualización de todos los usuarios que tiene a su cargo se mostrará una lista con todos ellos, dando la opción de ir a la pantalla de crear un usuario y, en caso de que seleccione un usuario alumno concreto, se le redirigirá a la pantalla especifica del usuario, donde se mostrará una lista con las palabras que tiene asignado el usuario estudiante con información relevante como los fallos, aciertos y su ratio de aciertos. Dentro de la pantalla del usuario seleccionado podrá acceder a la pantalla de añadir una palabra a dicho estudiante.

Por otra parte, si el usuario ingresado es un alumno se le redirigirá a la pantalla principal correspondiente, donde aparte de cerrar sesión, modificar su avatar y acceder a la pantalla de visualización y modificación de su información de perfil, podrá acceder a la pantalla juego en la cual podrá seleccionar la palabra que crea correcta respecto a la palabra propuesta. Otra acción que podrá realizar es visualizar todas las palabras que tiene asignadas, del mismo modo que pasaba con un usuario profesor, no obstante, el usuario estudiante tendrá restringidas algunas acciones como añadir una palabra.

7 Desarrollo

Una vez se tiene en mente el diseño y la tecnología que se utilizará, tanto para el back-end como para el front-end, se procederá a su respectivo desarrollo. En esta sección se explicará cómo se ha implementado la tecnología y el diseño en el proyecto.

7.1 Desarrollo del Back-End

Añadir Firebase a un proyecto es bastante sencillo, ya que únicamente se deberá de instalar la librería de Firebase a través de un instalador de paquetes JavaScript (por ejemplo, Yarn). Una vez se tenga creado el proyecto dentro del portal de Firebase, deberemos de sincronizar el proyecto Firebase

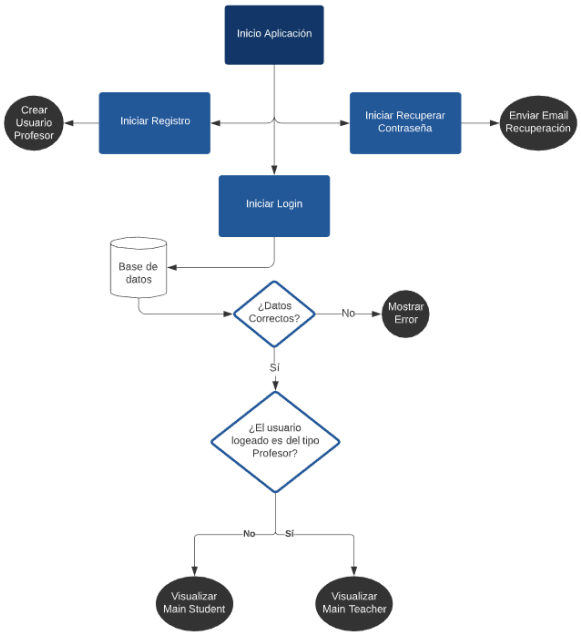


Fig. 2: Diagrama de Flujo Inicio Sesión

con nuestra aplicación. Cuando se configure desde su portal se deberá de tener en cuenta que la aplicación que se está diseñando está realizada en React Native, lo cual implica que se deberá de seleccionar la opción de desarrollo web, ya que React Native se basa en JavaScript.

Una vez configurado, Firebase nos proporcionará un SDK con toda la configuración pertinente que deberá de añadirse al proyecto. Se ha decidido que toda la información relativa a Firebase se pondrá dentro de un fichero llamado con el mismo nombre, dentro de una carpeta denominada *services*. Dentro de ese fichero, a parte del SDK personal, se creará una clase llamada Firebase donde se alojarán todas las funciones necesarias.

Como se puede ver en la figura 3, dentro del constructor se inicializará los módulos que serán utilizados en el proyecto: Autentificación, Real-Time Database y Cloud Storage.

```
const firebaseConfig = {
    apiKey: "**********************************",
    authDomain: "tfg1-reactapp.firebaseapp.com",
    databaseURL: "https://tfg1-reactapp.firebaseio.com",
    projectId: "tfg1-reactapp",
    storageBucket: "tfg1-reactapp.appspot.com",
    messagingSenderId: "817098494534",
    appId: "1:817098494534:web:86cae5e70171fbfe157047",
    measurementId: "G-0W32V8ST8B"
};

class Firebase {
    constructor() {
        firebase.initializeApp(firebaseConfig);
        this.auth = firebase.auth()
        this.database = firebase.database();
        this.storage= firebase.storage();

    }
}
```

Fig. 3: SDK y constructor de la clase Firebase

7.1.1 Estructura de la Base de Datos

Dado que el formato de la base de datos de Firebase son del tipo JSON y estas se pueden representar a partir de una estructura en forma de árbol, en la figura 4 se puede observar la representación de la base de datos de la aplicación, mientras en la figura 5 se puede observar un fragmento real. Siguiendo una estructura lógica y natural, se ha decidido hacer una división por el tipo de usuario. En la parte del estudiante se guardará todo lo relacionado con él, incluyendo el conjunto de palabras que tiene asignado. Por parte de un usuario profesor, además de la información relevante, se guardarán los identificadores de los estudiantes que tiene a su cargo.

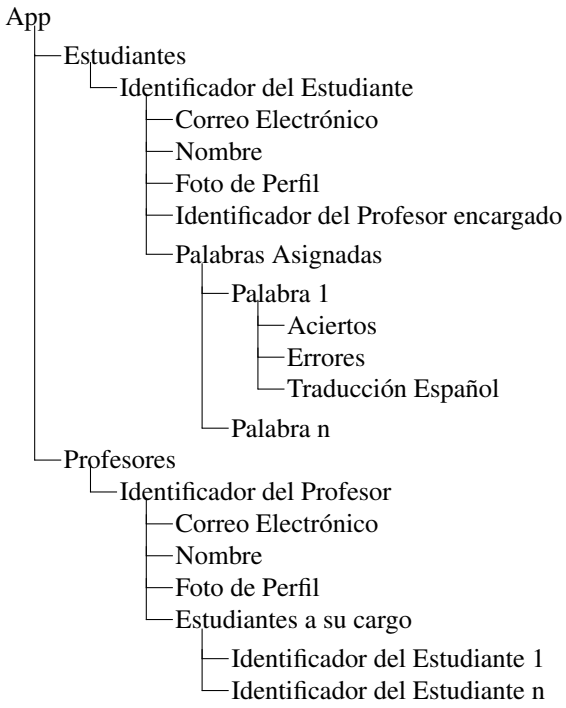


Fig. 4: Representación de la base de datos en forma de árbol

Fig. 5: Fragmento real de la Base de Datos de Firebase

```javascript
var image = React.createElement('img', {
  src: 'react-icon.png',
  className: 'icon-image'
});
var container = React.createElement('div', {
  className: 'icon-container'
}, image);
var icon = React.createElement('Icon', {
className: 'avatarContainer'
}, container);
ReactDOM.render(icon,
  document.getElementById('app')
);
```

(a) JavaScript

```jsx
var Icon = (
  <div className='icon-container'>
    <img
      src='react-icon.png'
      className='icon-image'
    />
  </div>)
ReactDOM.render(Icon, document.getElementById('app'))
```

(b) JSX

Fig. 6: Ejemplo entre las diferentes sintaxis.

7.2 Desarrollo Front-End

7.2.1 React Native

Como se ha mencionado en el apartado de tecnologías escogidas, la tecnología para el front-end ha sido React Native junto con Expo. A la hora de programar con esta herramienta se deberá de elegir que tipo de sintaxis utilizamos para programar, ya que React Native al estar basado en React JS, nos proporciona diferentes sintaxis. La escogida ha sido JSX [8] que consiste en una extensión a la sintaxis de JavaScript.

Aunque no es un requisito utilizarlo se recomienda por la gran capacidad que obtiene. ReactJS acepta el hecho de que la lógica de renderizado está intrínsecamente unida a la lógica de la interfaz de usuario: como se manejan los eventos, como cambia el estado con el tiempo y como se preparan los datos para su visualización.

De la forma en que se estructura parece que se está mezclando código HTML y JavaScript, no obstante, eso no ocurre. Cuando se utiliza JSX se debe de tener en cuenta algunas consideraciones, por ejemplo, algunas palabras reservadas en JavaScript y JSX obligan a nombrar algunos atributos de otra manera, como es el caso del atributo *class*, que en JSX se debe de escribir como *className*.

La ventaja principal que nos otorga JSX es la inserción de datos por parte del usuario, debido a que por defecto el Document Object Model (DOM) [9] implementado por React ignora cualquier valor insertado en JSX antes de renderizarlo. De este modo, se asegura de que nunca se pueda insertar nada que no esté explícitamente escrito en la aplicación. Todo es convertido en un string antes de ser renderizado. Esto ayuda a prevenir vulnerabilidades cross-site-scripting (XSS), inyecciones de código, entre otros.

En la figura 6 se puede observar la diferencia principal entre la sintaxis JavaScript normal y la sintaxis JSX.

7.2.2 API de Traducción

Como se ha mencionado en el apartado de funcionalidad, la aplicación deberá de disponer de un servicio de traducción con posibilidad de traducir palabras e incluso frases. Este servicio será posible a través de una Application Programming Interface (API). A la hora de elegirla se realizó una búsqueda de API's disponibles en el mercado. Se ha encontrado que hay una gran variedad, no obstante, las más utilizadas son dos: la API Translate de Google [10] y la API Translate de Microsoft [11]. Dado que la API de Google no es gratuita (solo los 90 días de acceso que ofrece Google Cloud), se decidió utilizar la API de Microsoft, que tiene una versión gratuita, suficiente para el uso concreto de la aplicación que se está diseñando.

Para poder acceder la API Translate de Microsoft se deberá de acceder mediante el portal Azure, y una vez sea validado el recurso se deberá de extraer una KEY proporcionada a través del portal que nos otorgará acceso a dicho servicio. La comunicación de la API se hace a través de REST, una interfaz basada en el protocolo HTTP que proporcionará los datos en formato JSON. En la figura 7 se puede observar un ejemplo del valor retornado de la traducción *How are you?* al español.

```json
[
  {
    "translations": [
      {
        "text": "¿Cómo estás?",
        "to": "es",
      },
    ],
  },
]
```

Fig. 7: Respuesta obtenida por la API Translate de Microsoft.

La implementación de la API al proyecto se incluirá dentro de la carpeta services del proyecto, juntamente con él archivo de Firebase. Una vez configurado, siguiendo los pasos mencionados en la documentación que proporciona Microsoft Translate [12], ya estará listo para recibir palabras y frases. En el anexo A3 se puede observar el código implementado de la API.

7.2.3 Algoritmo de ordenación de las palabras

Como se ha mencionado en la funcionalidad, cuando un usuario estudiante entra en la sección de jugar tendrá disponibles una a una su conjunto de palabras donde deberá de escoger la palabra correcta. Para que el orden no sea siempre el mismo se pensó una manera de visualizar el orden de las palabras a partir de un algoritmo. La opción más sencilla, utilizada durante el desarrollo de la aplicación para verificar su funcionamiento, es un orden aleatorio, es decir, cada vez que el usuario entre en la sección de jugar le saldrán las palabras disponibles aleatoriamente sin tener en cuenta si es una palabra nueva o su ratio de fallos. No obstante, para el diseño final se decidió que el orden fuese: primero las palabras nuevas agregadas y segundo el resto de las palabras ordenadas de mayor a menor ratio de errores.

Una vez decidido el orden había que buscar un algoritmo de ordenación eficiente, ya que pensando hacia una vista futura en la que un estudiante tenga asignadas bastantes palabras, no tardase más de la cuenta en obtener la ordenación. A la hora de buscar el algoritmo se decidió utilizar el algoritmo QuickSort o el algoritmo Merge Sort, ya que son los algoritmos de ordenación más rápidos conocidos y aseguran un tiempo de ejecución promedio de O(n log (n)) y en el peor de los casos de $O(n^2)$. QuickSort es más eficiente para datos pequeños, en cambio para MergeSort funciona mejor con datos grandes.

La estrategia de los algoritmos consiste en la técnica divide y vencerás, por la que en cada recursión el problema se divide en partes más pequeñas y se resuelve cada parte por separado utilizando la misma técnica. Una vez finalizada la ejecución de cada subproblema se unen para obtener el resultado final.

Para implementar el algoritmo se ha utilizado el método *sort()* disponible en JavaScript, ya que en función del tamaño de los datos utilizará un algoritmo u otro, y además, en caso de que se tenga un número inferior a 7 elementos utilizará el algoritmo Insertion Sort [13], por lo que nos asegura que siempre se esté utilizando un algoritmo eficiente, en función del tamaño de los datos.

Para adaptar el algoritmo al formato de datos de nuestro diseño se han realizado unos pequeños cambios para obtener los resultados esperados. Tal y como se puede observar en la figura 8 se aplica el algoritmo a partir de la ratio de errores obtenido en una palabra concreta y se va ordenando de menor a mayor comparándose entre otro conjunto. Dado que en las palabras nuevas su ratio corresponderá a una división con un resultado NaN, debido a que tendrá un total de 0 intentos (0/0), se le asignará una ratio mayor a 1 (100%) para que se sitúen al final de la lista ordenada. Finalmente, quedará dar la vuelta al resultado obtenido y se tendrá el conjunto de palabras ordenadas a partir del requisito expuesto.

```
//Words Order: 1st new words after highest to lowest by percentage of error
const orderWords = (originalObjects)=>{
  return (Object.keys(originalObjects).sort(function(a,b){
    let first = originalObjects[a].miss/(originalObjects[a].miss+originalObjects[a].hit)
    let second = originalObjects[b].miss/(originalObjects[b].miss+originalObjects[b].hit)
    isNaN(first) ? first=1.1: "";
    isNaN(second) ? second=1.1: "";
    return (first-second)
  }).reverse())
}
```

Fig. 8: Algoritmo de ordenación de las palabras.

7.3 Comunicación entre Front-End y Back-End

La comunicación entre el Front-End y el Back-End se hace a partir de protocolos HTTPS. Por ejemplo, cuando un usuario inicia sesión en la aplicación, los datos son enviados al SDK Authentication de Firebase y en el caso de que los datos introducidos sean correctos recibirá un token único que se guardará en el dispositivo, este token a parte de otros usos el principal será que en caso de que el usuario salga de la aplicación no tenga que volver a iniciar sesión ya que se verificará a partir del token.

En caso de que se requiera acceder a alguna función de Firebase, como por ejemplo acceder a la base de datos, la petición se hará a partir de un POST, donde en la cabecera siempre enviará el token asociado. El encargado de procesar las peticiones recibidas por el front-end le corresponde a las denominadas Cloud Functions, que permiten guardar código JavaScript en la nube. Posteriormente, las Cloud Functions se encargarán de realizar las peticiones a los módulos correspondientes. En la figura 9 se puede observar un esquema sencillo de la comunicación entre ambas capas.

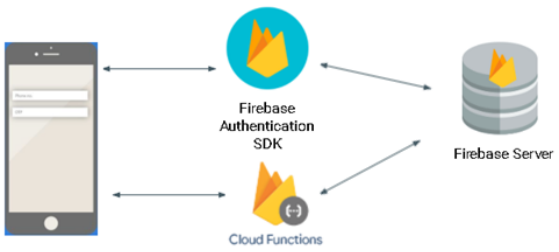


Fig. 9: Comunicación entre Front-End y Back-End.

8 Fase de Pruebas - Tests

Un punto importante a la hora de desarrollar una aplicación es realizar sus respectivas pruebas para ir validando los diferentes módulos. En este apartado se expondrán las pruebas realizadas tanto en el back-end como en el front-end.

8.1 Pruebas Unitarias

Las pruebas unitarias consisten en comprobar el correcto funcionamiento de diferentes módulos, partes del código, aisladas del resto. El objetivo de ellas será detectar errores o posibles problemas y comprobar que la lógica del código funciona de manera correcta.

Para la realización de las pruebas unitarias se ha decidido utilizar Jest, un framework que React Native tiene configurado por defecto. Funciona de tal manera que ejecuta las funciones que queramos probar, pasando parámetros determinados de tal manera que sepamos cual debería de ser el resultado esperado. A partir de la ejecución de las pruebas, que compara el valor esperado con el obtenido, se extrae un resumen de las pruebas correctas y erróneas.

Dentro del fichero de test se han realizado 5 pruebas diferentes, considerados los módulos o funciones más críticas. Tres de ellas son pertenecientes al módulo de Firebase (Register, Login y extracción del usuario registrado), las otras dos, una corresponde al módulo de la API de traducción de Microsoft y la última al algoritmo de ordenación de palabras.

En la figura 10 se puede observar que la prueba de los 5 módulos ha sido exitosa, dando como información extra el tiempo que tarda en ejecutarse cada función. Analizando los datos podemos observar que el módulo de registro y el de traducción son los elementos más críticos en cuanto a tiempo. Por otra parte, el algoritmo de ordenación obtenemos un buen resultado de tiempo de ejecución.

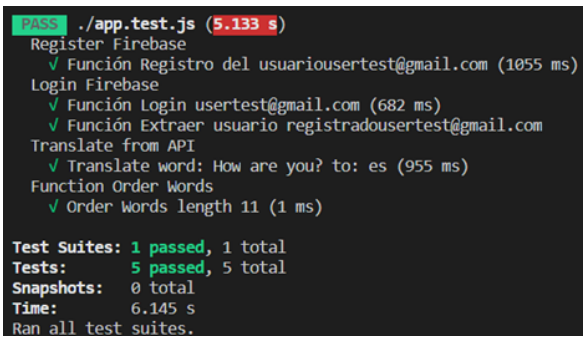


Fig. 10: Salida de la ejecución del fichero Jest.

8.2 Test de Usabilidad

La segunda prueba que se ha realizado para verificar el correcto funcionamiento de la aplicación ha sido un test o prueba de usabilidad. Los tests de usabilidad consisten en saber cómo los posibles usuarios potenciales navegan y utilizan la aplicación, con el objetivo de que, al finalizar la prueba, se obtenga un *feedback*.

Para realizar el test se han seleccionado a un grupo de personas que debían de realizar un conjunto de tareas (registrarse, iniciar sesión, crear un usuario alumno, asignarle palabras, jugar, etc.), con el objetivo de que los usuarios naveguen por toda la aplicación en busca de alguna anomalía. Al finalizar el primer contacto con la aplicación, los mismos usuarios debían de realizar un formulario, donde a partir de unas preguntas deben de expresar sus impresiones. En el anexo A4 se puede observar el formulario con las preguntas realizadas y las respuestas obtenidas.

A partir del formulario se obtuvo un *feedback* bastante adecuado, donde se podía ver los puntos débiles y las mejoras que se debían de hacer en la aplicación. A la gran mayoría le gustaba el diseño de navegación entre pantallas, no obstante, casi todos coincidían de que la interfaz gráfica podía ser mejorada, haciendo énfasis en los colores primarios y secundarios de la aplicación.

En el siguiente apartado, en líneas de mejoras y ampliación, se extraen algunos consejos e ideas que han sido extraídos de la prueba de usabilidad.

9 Conclusiones

Una vez finalizado el proyecto se ha llegado a la conclusión que realizar una aplicación desde cero es un reto, no obstante, se podría decir que el desarrollo ha sido finalizado satisfactoriamente. El objetivo de desarrollar una aplicación multiplataforma funcional con React Native junto con Firebase, que ha permitido implementar una base de datos en tiempo real, ha sido alcanzado.

Desde un punto personal, gracias a la realización del trabajo, se ha proporcionado un aprendizaje exhaustivo de la tecnología empleada para el desarrollo de la aplicación, ya sea React Native como Firebase, dando lugar a profundizar conocimientos relacionados con tecnologías y lenguajes multiplataforma, como JavaScript, los cuales durante el curso académico no se han entrado en tanto detalle.

Sobre la plataforma de Firebase se puede concluir que es muy útil para todo tipo de aplicaciones, ya sean grandes o pequeñas, ya que toda la gestión relacionada con el backend, como la autentificación, base de datos y funciones de almacenamiento lo realiza de una manera sencilla haciendo que el programador no deba de invertir más tiempo del necesario, no obstante, si la aplicación está pensada para un trafico con mayor concurrencia, se deberá de optar por hacer un desembolso económico analizando los diferentes planes que nos permite contratar Firebase [14].

Por otra parte, realizar pruebas de testeo de diferentes secciones de la aplicación, o con los mismos tests de usabilidad, ha servido para encontrar fallos y debilidades más importantes, aportando mejoras pertinentes al desarrollo de esta misma.

Lo que concierne a la evolución del proyecto, debido a diferentes fechas de exámenes, no se ha seguido una evolución lineal de trabajo en el tiempo, no obstante, se ha considerado correcta como ha ido evolucionando el proyecto hasta su fin.

Como en cualquier proyecto hay muchos aspectos que se pueden mejorar o tratarse de otra manera, por eso a continuación se expondrán unas líneas de mejoras y ampliación sobre la aplicación desarrollada.

9.1 Líneas de mejoras

En primer lugar, una línea de mejora podría ser la internacionalización de la aplicación, dando lugar a que esta esté en diferentes idiomas, según la configuración del dispositivo.

Otra mejora podría ser la opción de visualizar el conjunto de palabras en los dos idiomas configurados, es decir, que no solo se muestre la palabra principal en inglés y las opciones en castellano, sino que se vayan combinando.

Durante la fase de pruebas hubiera sido interesante realizar otro tipo de pruebas, como por ejemplo pruebas del diseño UI (interfaz de usuario), que consisten en probar la interfaz gráfica de usuario para así garantizar que se cumplen todas las especificaciones. Otra implementación para añadir podría haber sido agregar el módulo Firebase Analytics, que nos ofrece herramientas para analizar diferentes aspectos como su estabilidad, dando un mayor control sobre la aplicación realizando Crash Reports u otro tipo de información útil.

Una línea de mejora posible podría ser la implementación de subida y lectura de algún fichero, por ejemplo, de formato Excel, con el objetivo de agilizar la introducción de palabras o el registro de usuarios estudiantes.

Respecto a la distribución de las pantallas se podría mejorar dando lugar a menús desplegables para hacer que la navegación sea más intuitiva.

Finalmente, relacionado con lo anterior, otra línea de mejora podría ser la interfaz, sus colores principales y secundarios, dando lugar a una interfaz más amigable con el usuario y así mejorar la experiencia de usuario.

9.2 Líneas de ampliación

Lo que concierne a la línea de ampliación, la aplicación tiene la capacidad de ampliarse, añadiendo bastantes más funcionalidades.

Una de ellas podría ser la incorporación de mensajería instantánea a partir de un chat localizado en la propia aplicación para mejorar la comunicación entre profesor y alumno, dando lugar a una mayor experiencia de usuario. Esta incorporación se haría a través de la plataforma Firebase utilizando la base de datos en tiempo real, ya que facilitaría su implementación.

Otra línea de ampliación posible podría ser la de añadir más de un idioma para que un usuario alumno practicase, y él mismo pueda seleccionar con el idioma que desea trabajar.

Como ultima línea de ampliación podría ser la adaptación a los navegadores web, ya que al utilizar la tecnología de React, nos da la opción de poder desarrollar el proyecto en un entorno web, pudiéndose ejecutar en cualquier navegador.

AGRADECIMIENTOS

En primer lugar, me gustaría agradecer a mi tutor, Enric Vergara, por toda la ayuda y consejos ofrecidos durante la realización de esta pequeña aplicación.

También, agradecer a toda mi familia por todo el soporte y apoyo ofrecido, no solo durante la realización del trabajo, sino también por el ofrecido durante todos los años de la carrera.

Finalmente, me gustaría agradecer a todas aquellas personas que me han ayudado a mejorar el trabajo, ya sea a partir de las entrevistas realizadas, con sus opiniones y/o ideas.

Gracias de corazón.

REFERENCIAS

[1] S. Siripathi, “Lenguajes de Desarrollo Para Móvil”, Code Envato Tuts+. [En línea]. Disponible en: https://code.tutsplus.com/es/articles/mobile-development-languages--cms-29138. [Accedido: 29-sept-2020]

[2] O. Systems, “¿Qué es un framework y para qué se utiliza?”, Orix. [En línea]. Disponible en: https://www.orix.es/que-es-un-framework-y-para-que-se-utiliza. [Accedido: 29-sept-2020]

[3] “Base de datos SQL: ¿cómo funciona y se gestiona esta base de datos?”, TIC Portal. [En línea]. Disponible en: https://www.ticportal.es/glosario-tic/base-datos-sql. [Accedido: 30-sept-2020]

[4] F. Conislla, “Cómo aplicar seguridad en el ciclo de vida del desarrollo de software», Belatrix. [En línea]. Disponible en: https://www.belatrixsf.com/blog/seguridad-desarrollo-software. [Accedido: 5-oct-2020]

[5] “Firebase”, Google. [En línea]. Disponible en: https://firebase.google.com/docs?hl=es-419. [Accedido: 5-oct-2020]

[6] S. India, “React Native vs. Ionic vs. Flutter”, Codeburst - Medium. [En línea]. Disponible en: https://codeburst.io/react-native-vs-ionic-vs-flutter-comparison-of-top-cross-platform-app-development-tools-71c8011309ac. [Accedido: 5-oct-2020]

[7] Y. Kruhlyk, “Expo vs Vanilla React Native Development: What to Choose”, Apiko. [En línea]. Disponible en: https://apiko.com/blog/expo-vs-vanilla-react-native/. [Accedido: 5-oct-2020]

[8] “Presentando JSX”, React. [En línea]. Disponible en: https://es.reactjs.org/docs/introducing-jsx.html. [Accedido: 8-oct-2020]

[9] M. A. Alvarez, “Qué es el DOM”, Desarrollo Web. [En línea]. Disponible en: https://desarrolloweb.com/articulos/que-es-el-dom.html. [Accedido: 15-oct-2020]

[10] “Cloud Translation”, Google Cloud. [En línea]. Disponible en: https://cloud.google.com/translate?hl=es. [Accedido: 7-nov-2020]

[11] “Translator”, Microsoft Azure. [En línea]. Disponible en: https://azure.microsoft.com/es-es/services/cognitive-services/translator/. [Accedido: 25-nov-2020]

[12] “Inicio rápido: Introducción a Translator”, Azure Cognitive Services. [En línea]. Disponible en: https://docs.microsoft.com/es-es/azure/cognitive-services/translator/quickstart-translator. [Accedido: 25-nov-2020]

[13] “Clase Array”, Open JDK. [En línea]. Disponible en: https://hg.openjdk.java.net/jdk/jdk/file/9c3fe09f69bc/src/java.base/share/classes/java/util/Arrays.java#l1331. [Accedido: 12-dic-2020]

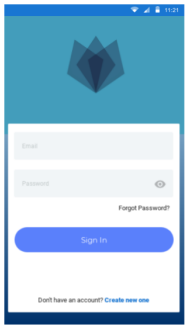
[14] “Firebase Pricing”, Firebase. [En línea]. Disponible en: https://firebase.google.com/pricing. [Accedido: 9-ene-2020]

ANEXOS

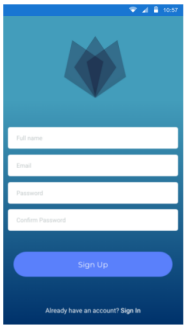
A1. RESUMEN TEMPORAL DEL PROYECTO

20/09/20 – 11/10/20	**Toma de contacto**	Reunión Inicial
		Objetivos
		Investigación y Testeo Frameworks
		Metodología
		Estado de Arte
		Mockup
		Redacción Informe Inicial
11/10/20 – 15/11/20	**Primera Etapa - Desarrollo**	2nda Reunión
		Posibles cambios
		Configuración Entorno
		Desarrollo UI
		Redacción Informe P.I
15/11/20 – 20/12/20	**Segunda Etapa – Desarrollo**	Desarrollo UI
		Investigación API Traducción
		Implementación API Traducción
		Posibles cambios
		Redacción Informe P. II
20/12/20 – 24/01/21	**Etapa Final - Testeo**	Verificación y Testeos Finales
		Análisis de los Tests
		Modificaciones UI/UX
		Redacción Informe Final
24/01/21 – 07/01/21	**Etapa Presentación**	Propuesta de Presentación

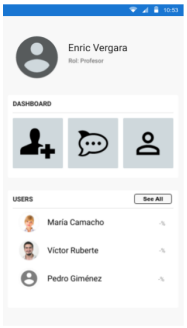
A2. MOCKUP



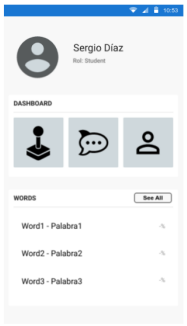
Vista 1: Inicio Sesión



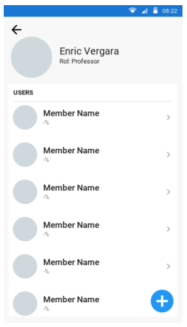
Vista 2: Registro Profesor



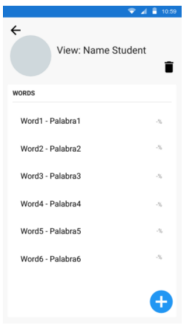
Vista 3: Main Teacher



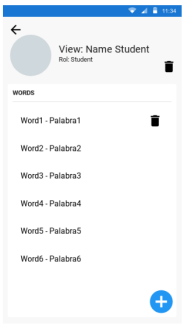
Vista 4: Main Student



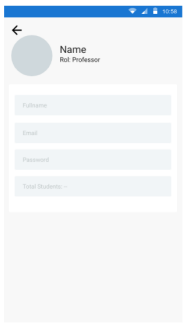
Vista 5: Estudiantes asignados



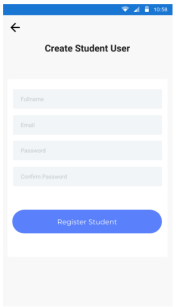
Vista 6: Palabras asignadas Estudiante (vista Prof.)



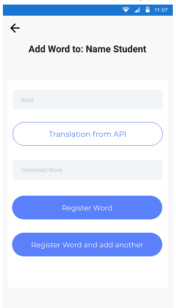
Vista 7: Eliminar palabra a Estudiante



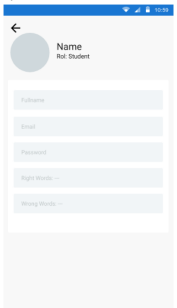
Vista 8: Vista Perfil Profesor



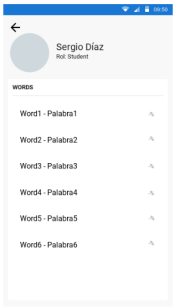
Vista 9: Crear usuario Estudiante



Vista 10: Añadir palabra a Estudiante



Vista 11: Información Perfil Est.



Vista 12: Palabras asignadas del Estudiante (vista Estud.)

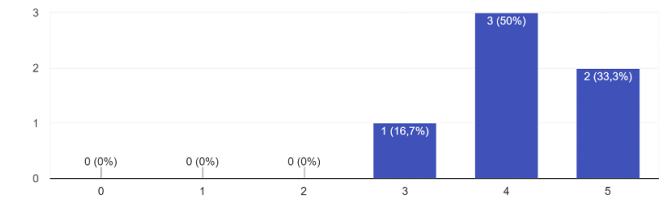
Vista 13: Juego

A3. CÓDIGO API TRANSLATE

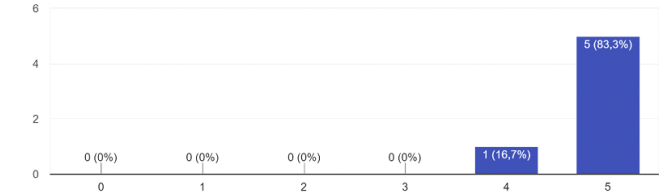
```
const axios = require('axios').default;
var subscriptionKey = "**************************";
var endpoint = "https://api.cognitive.microsofttranslator.com";

var location = "westeurope";

class apiMT{
  translate = (word, from, to)=>{
    return axios({
      baseURL: endpoint,
      url: '/translate',
      method: 'post',
      headers: {
          'Ocp-Apim-Subscription-Key': subscriptionKey,
          'Ocp-Apim-Subscription-Region': location,
          'Content-type': 'application/json',

      },
      params: {
          'api-version': '3.0',
          'from': from,
          'to': to
      },
      data: [{
          'text': word
      }],
      responseType: 'json'
    })
  }
}

const translatorService = new apiMT();
export default translatorService;
```

A4. FORMULARIO DE SATISFACCIÓN - TEST DE USABILIDAD

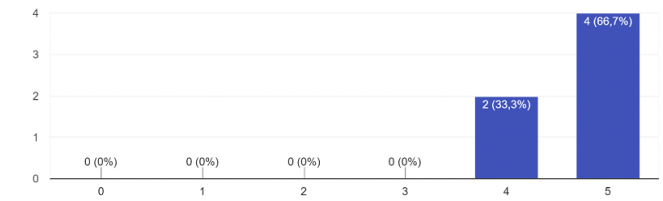
¿Cuál es el grado de satisfacción con el diseño de navegación entre pantallas?
6 respostes



¿Le resulta cómodo el sistema para añadir palabras?
6 respostes



¿Cree oportuno la manera en que se priorizan las palabras cada vez que se inicia una sesión de trabajo?*
6 respostes



¿Ha detectado algún comportamiento erróneo en la aplicación?
4 respostes

No

Cuando he creado el usuario del alumno, en vez de salir Andrea ha salido Andre

Ninguno.

¿Propone alguna mejora?
6 respostes

Los colores no me acaban de convencer a la hora de crear una imagen de marca (un buen ejemplo en el que inspirarse sería Duolingo), crear un "algoritmo" donde se junten las palabras con su gremio (por ejemplo oficios, animales y cuando te salga caballo te salga caballo, potro, vaca,etc.

El color del fondo.

Aunque la aplicación es intuitiva el diseño no me acaba de convencer sobretodo los colores.

La interfaz gráfica y el juego

Los iconos como te he comentado no me convencen, tal vez son muy grandes tal vez el grosor de la línea o tal vez la elección de color.
La navegación entre pantallas hace un efecto como de desplazamiento hacia arriba en el cambio de pantalla (no sé si voluntario) que, personalmente no me gusta.

El juego lo veo un poco repetitivo, añadiría algún otro tipo de juego. La distribución de la aplicación esta bien pero el diseño lo veo un poco soso.

¿Añadiría una nueva funcionalidad?
6 respostes

Un profesor 24 horas, es decir, que a parte de tu tutor hubiese un profesor que pueda estar disponible para responder tus dudas, speaking en directo, y que no sean solo palabras simples, que también incluya frases.

Mas idiomas a parte del ingles.

El chat, añadir pronunciación de palabras

Añadir mas idiomas a la aplicación y mas tipos de juegos

No

Como he dicho en la anterior respuesta añadiría otro tipo de juego, en vez de una palabra salga una imagen o algo del estilo.