

GUI N DE ACTIVIDAD

Programaci n Multimedia y de Dispositivos M viles

ACTIVIDAD EVALUABLE:

OBJETIVOS

- Realizar peticiones GET en React Native.
- Implementar Navegaci n en React Native.
- Trabajar en grupo.

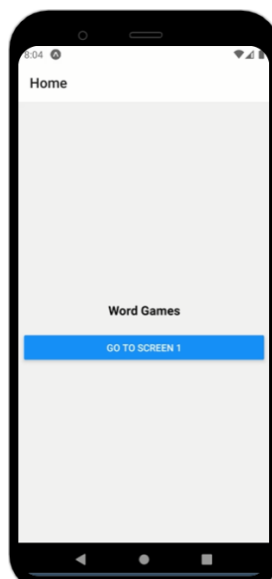
TEMPORALIZACI N

Aproximadamente 4-5 horas

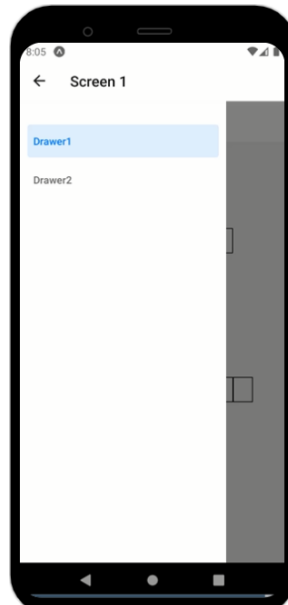
PROCESO DE DESARROLLO

Para la presente actividad debes implementar primeramente la siguiente **navegaci n**:

- a. **Stack**: permitir  pasar mediante un bot n de una pantalla inicial Home a otra pantalla distinta, por ejemplo Screen1.



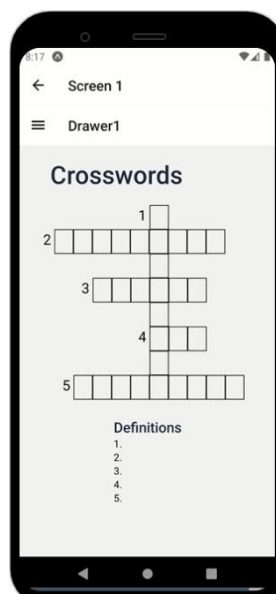
- b. **Drawer:** implementada en la pantalla Screen1, se compondrá a su vez en dos pantallas distintas: Drawer1 y Drawer2.



Nota: para ver con mayor detalle la navegación, se recomienda ver el video publicado en Florida Oberta.

A partir de los dos archivos proporcionados, `crosswords.js` y `quizz.js`, se deberá implementar para cada una de las dos pantallas de la navegación Drawer, las siguientes funcionalidades:

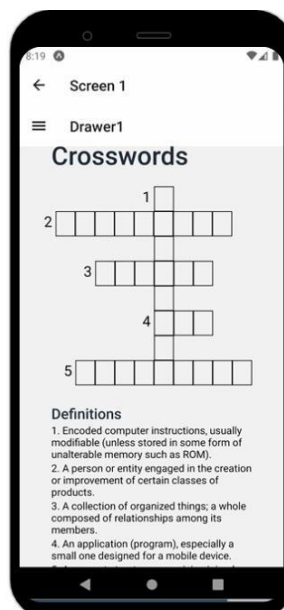
Drawer1: Se implementará el crucigrama contenido en el archivo "crosswords.js".



Dicho crucigrama se compone de las siguientes palabras, en lengua inglesa:

- 1 – Software (vertical).
- 2 – Developer (horizontal).
- 3 – System (horizontal).
- 4 – App (horizontal).
- 5 – Framework (horizontal).

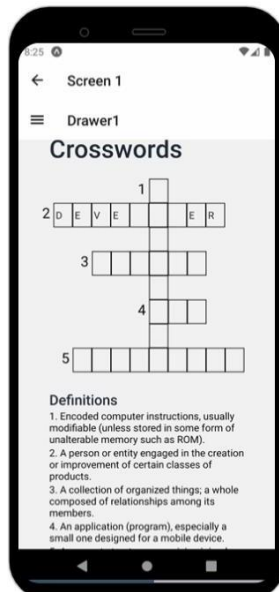
Se implementará la lógica necesaria para que al pulsar sobre el título **Crosswords** se muestren en los componentes Text ubicados bajo la inscripción Definitions las definiciones de cada una de las cinco palabras.



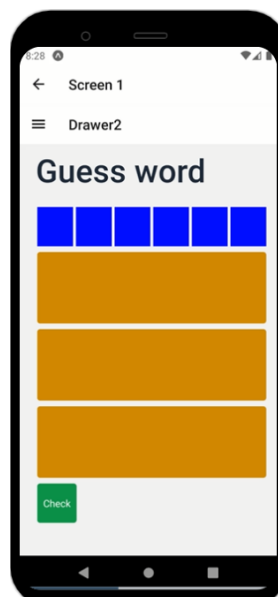
Las definiciones de cada palabra se obtendrán a partir de una petición HTTP GET a la siguiente API pública –se debe guardar únicamente una definición por palabra–:

[Free Dictionary API](#)

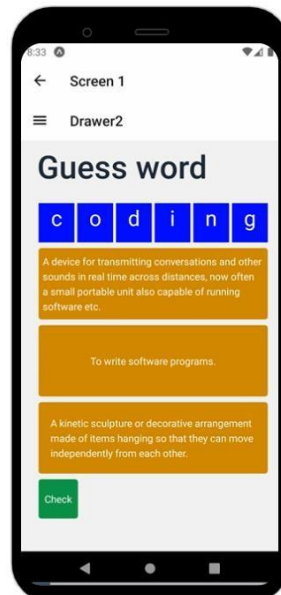
Se debe implementar la siguiente lógica para que un usuario pueda realizar el crucigrama: en cada uno de los componentes TextInput de cada palabra se introducirán las letras de cada palabra. Al pulsar sobre cada uno de los números ubicados en las cuadrículas del crucigrama, se comprobará si la palabra introducida es correcta y en caso de no serlo, se dejarán visibles por pantalla únicamente aquellas letras introducidas que coinciden con las letras de la palabra requerida. Ejemplo: la palabra 2 (horizontal) del crucigrama es ‘developer’. Si un usuario introduce ‘deveeeeeer’, al pulsar sobre el número 2 se mostrará por pantalla lo siguiente:



Drawer2: Se implementará el crucigrama contenido en el archivo “quizz.js”.

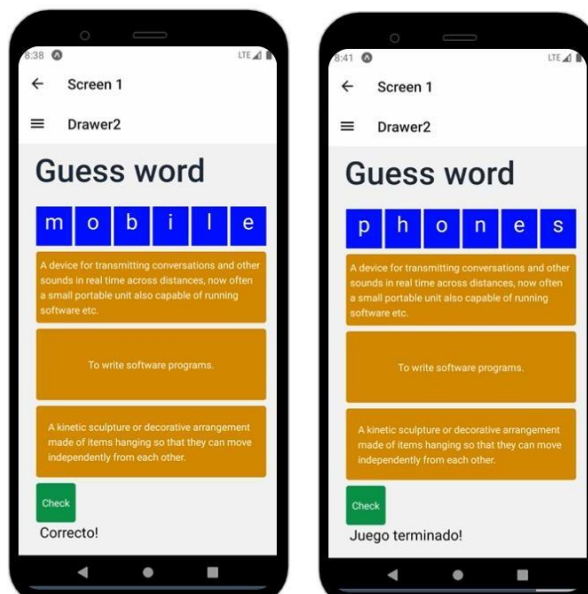


Deberá implementarse la lógica necesaria para que al pulsarse sobre el título “Guess Word” se realice peticiones GET a la misma API pública indicada anteriormente, con las tres palabras siguientes: ‘coding’, ‘mobile’, ‘phones’. En las cuadrículas azules ubicadas bajo el título Guess Word se mostrará la primera de las tres palabras y debajo, en los espacios naranjas habilitados para ello, las definiciones para cada una de las tres palabras. Las definiciones deben aparecer en un orden distinto al de las palabras. Es decir, si en el flujo del juego la primera palabra que se muestra en las cuadrículas es ‘coding’, la primera definición ubicada en la parte superior no puede ser la correspondiente a esta palabra –nota: este orden puede codificarse directamente ‘a mano’, no es necesario implementar ninguna función especial ni recurrir a Math.random para ello-. En cada uno de los tres espacios naranjas se implementará un evento onPress, de modo que permita guardar cual de los tres espacios ha pulsado el usuario.



Nota: se presupone que el usuario presionará una única vez sobre los espacios naranjas.

Flujo del cuestionario: cada vez que el usuario pulse sobre uno de los tres espacios naranjas y posteriormente haga lo mismo sobre el componente Button de color verde con la inscripción Check, se mostrará en el espacio habilitado para ello la inscripción 'correcto' o 'error' según haya o no acertado con su respuesta. Posteriormente se mostrará la siguiente palabra, sobre la que se podrá volver a elegir una de las tres definiciones. El flujo del juego termina cuando se han comprobado las tres palabras, en cuyo caso y tras pulsar sobre Check aparecerá la inscripción "Juego terminado".



Para poder realizarse la actividad, deberán instalarse las librerías correspondientes a la navegación y la librería Native Base. Para ello se introducirán los siguientes comandos en el terminal de Visual Studio sobre un proyecto ya generado con Expo:

Librería Native Base:

```
npm install native-base --force  
expo install react-native-svg@12.1.1  
expo install react-native-safe-area-context@3.3.2
```

En caso de trabajar en Expo Snack, bastará con pulsar a “add dependency” en el terminal donde se muestran los errores.

EVALUACIÓN

Esta actividad se engloba en la parte de actividades evaluables de la asignatura.

OBSERVACIONES

La entrega se realizará en el espacio habilitado para ello en Florida Oberta. Se entregarán comprimidos en una única carpeta todos los archivos generados para el correcto funcionamiento de la aplicación. **Los scripts a entregar deberán ceñirse estrictamente a lo que se pide en el enunciado de la actividad. Las funcionalidades no solicitadas de manera explícita en el enunciado penalizarán el valor de la nota.**