

GUI N DE ACTIVIDAD

Programaci n Multimedia y de Dispositivos M viles

ACTIVIDAD EVALUABLE:

OBJETIVOS

- Realizar peticiones GET en React Native.
- Implementar el Operador Spread para trabajar con arrays.
- Implementar Navegaci n en React Native.
- Reproducir archivos de audio.
- Trabajar con el Hooks useEffect.

PROCESO DE DESARROLLO

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

- a. **Tab**: navegaci n de cuatro pantallas a las que se podr  acceder mediante cuatro pesta as ubicadas en el borde inferior de la pantalla (*1 punto*).

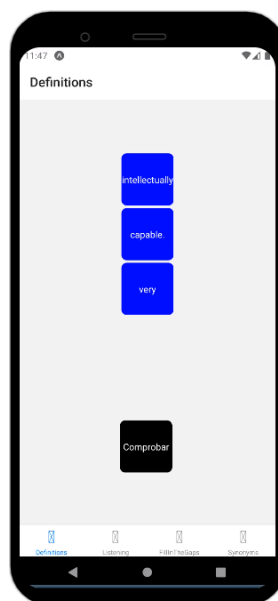
Nota: para ver con mayor detalle la navegaci n, se recomienda ver el video publicado en Florida Oberta.

A partir de los dos archivos proporcionados, Definitions.js, Listening.js, FillInTheGaps.js y Synonyms.js, se deber  implementar para cada una de las cuatro pantallas de la navegaci n Tab, las siguientes funcionalidades:

Definitions: a partir del archivo "Definitions.js" y del archivo "definitions.json" (*2 puntos*).

En esta pantalla se implementará un pequeño juego de dos niveles, cuya dinámica se explica a continuación:

Al acceder a esta pestaña se cargarán automáticamente las palabras correspondientes al primer nivel del archivo 'definitions.json' proporcionado, – cuatro palabras en total-. Se obtendrá al azar una de dichas palabras y con ella se realizará una petición GET a la API pública [Free Dictionary API](#). De la petición GET se obtendrá una definición de la palabra y la localización del archivo de audio con la pronunciación de dicha palabra. Dicho archivo se reproducirá para que el usuario pueda escuchar de qué palabra se trata y mientras suena la palabra, en pantalla aparecerán las palabras que forman la definición desordenadas, a razón de una palabra por cuadrícula, tal y como se observa a continuación:



Se implementará la lógica necesaria para que el usuario pueda ordenar la palabra pulsando sobre las cuadrículas. Cuando crea que esté ordenada, podrá comprobarlo al pulsar sobre el botón comprobar. Si la definición de la palabra está correctamente ordenada, se pasará al siguiente nivel, donde se trabajará sobre las palabras del archivo "definitions.json" correspondientes al segundo nivel.

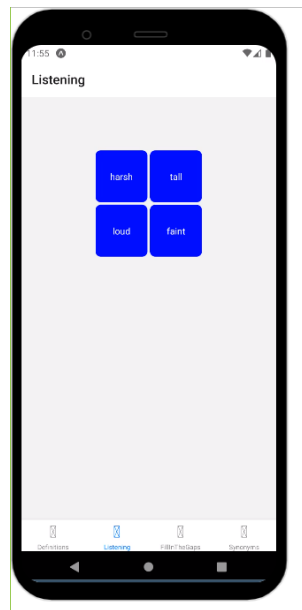
Si el usuario vuelve a ordenar correctamente la definición de la palabra, se habrá terminado el juego y aparecerá un botón 'Try again', con el que podrá volver a jugar.

Si en cambio comete a lo largo del juego más de dos fallos, también terminará el juego y aparecerá el botón 'Try again', con el que podrá volver a jugar.

Listening: a partir del archivo “Listening.js” y del archivo “listening.json” (3 puntos).

En esta pantalla se implementará un pequeño juego de dos niveles, cuya dinámica se explica a continuación:

Al acceder a esta pestaña se cargarán automáticamente las palabras correspondientes al primer nivel del archivo ‘listening.json’ proporcionado, – seis palabras en total-. Se obtendrán al azar cuatro de dichas palabras y con ellas se completará el tablero de juego. Haciendo uso de las funciones predefinidas en JavaScript setTimeout y setInterval, se implementará la lógica necesaria para las palabras se vayan intercalando aleatoriamente en las posiciones del tablero cada 0,1 segundos. Cuando transcurran 3 segundos desde que se ha iniciado la carga del tablero, se detendrán y las palabras se quedarán fijas en la posición en la que estén. En paralelo y durante estos tres segundos, se habrá realizado una petición GET a la API pública [Free Dictionary API](#) con una de las cuatro palabras que se muestran en el tablero, elegida al azar. De la petición GET se habrá obtenido la localización del archivo de audio y se habrá reproducido la pronunciación de dicha palabra. De este modo, el usuario podrá escuchar de qué palabra se trata y, al terminar los tres segundos elegir que palabra ha sonado:



Se implementará la lógica necesaria para que el usuario pueda identificar la palabra pulsando sobre las cuadrículas. Si elige correctamente la palabra, se accederá al siguiente nivel, donde se trabajará sobre un tablero de 3 x 3 cuadrículas. Las palabras se obtendrán del archivo “listening.json” correspondientes al segundo nivel.

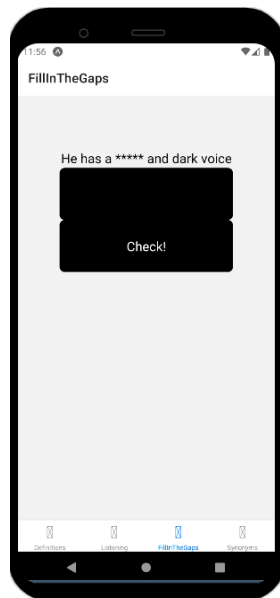
Si el usuario vuelve a identificar correctamente la palabra, se habrá terminado el juego y aparecerá un botón ‘Try again’, con el que podrá volver a jugar.

Si en cambio comete a lo largo del juego más de dos fallos, también terminará el juego y aparecerá el botón ‘Try again’, con el que podrá volver a jugar.

FillInTheGaps: a partir del archivo “Listening.js” y del archivo “fill_in_the_gaps.json” (3 puntos).

En esta pantalla se implementará un pequeño juego de dos niveles, cuya dinámica se explica a continuación:

Al acceder a esta pestaña se cargarán automáticamente los adjetivos correspondientes al primer nivel del archivo ‘fill_in_the_gaps.json’ proporcionado, –seis adjetivos en total-. Se implementará la lógica necesaria para obtener al azar un adjetivo al azar con su frase asociada. Con el adjetivo se realizará una petición GET a la API pública [Free Dictionary API](#) y se obtendrá la localización del archivo de audio y se reproducirá la pronunciación de dicha palabra. En la pantalla aparecerá la frase asociada a esta palabra y bajo ella un componente TextInput, mediante el cual el usuario deberá introducir la palabra que ha escuchado.



Cuando el usuario escriba la palabra, podrá pulsar sobre ‘Check!’ para comprobar si es correcta. Si escribe correctamente la palabra, se accederá al siguiente nivel, donde se trabajará sobre una de las palabras del segundo nivel del archivo “fill_in_the_gaps.json” correspondientes al segundo nivel.

Si el usuario vuelve a escribir correctamente la palabra, se habrá terminado el juego y aparecerá un botón ‘Try again’, con el que podrá volver a jugar.

Si en cambio comete a lo largo del juego más de dos fallos, también terminará el juego y aparecerá el botón ‘Try again’, con el que podrá volver a jugar.

Synonyms: a partir del archivo “Synonyms.js” y del archivo “synonyms.json” (3 puntos).

En esta pantalla se implementará un pequeño juego de dos niveles, cuya dinámica se explica a continuación:

Al acceder a esta pestaña se cargarán automáticamente los adjetivos correspondientes al primer nivel del archivo ‘synonyms.json’ proporcionado, – ocho adjetivos en total-. Se implementará la lógica necesaria para obtener al azar un adjetivo. Con el adjetivo se realizará una petición GET a la API pública [Free Dictionary API](#) y se obtendrá la localización del archivo de audio y se reproducirá la pronunciación de dicha palabra. En el tablero de juego aparecerán en cada cuadrícula un sinónimo de cada una de los adjetivos del primer nivel de juego:



Se implementará la lógica necesaria para que el usuario pueda identificar la palabra pulsando sobre las cuadrículas. Si elige correctamente la palabra, se accederá al siguiente nivel, donde se trabajará sobre los adjetivos del segundo nivel. Las palabras se obtendrán del archivo “synonyms.json”.

Si el usuario vuelve a identificar correctamente la palabra, se habrá terminado el juego y aparecerá un botón ‘Try again’, con el que podrá volver a jugar.

Si en cambio comete a lo largo del juego más de dos fallos, también terminará el juego y aparecerá el botón ‘Try again’, con el que podrá volver a jugar.

Se refactorizará el código JSX proporcionado utilizando funciones de orden superior sobre arrays.

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 o el uso de estructuras no vistas en la asignatura penalizarán el valor de la nota.**