

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 patrones de navegaci n.
- Parametrizar componentes.
- Utilizar los Hooks useEffect y useContext.

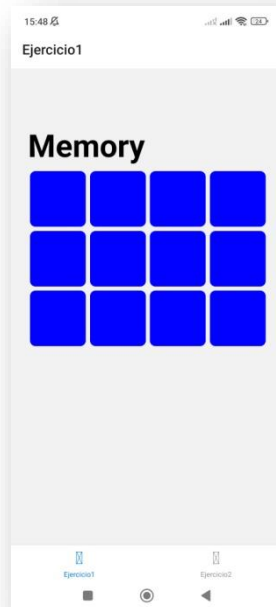
PROCESO DE DESARROLLO

Para la presente actividad debes implementar un patr n de navegaci n Tab de dos pesta as (0.5 puntos).

El contenido de cada pesta a se detalla a continuaci n:

Pesta a 1 (4.5 puntos): En la primera pesta a implementar s una variante del popular juego *memory* a partir de im genes de los personajes de la serie Rick & Morty, obtenidos de la API REST p blica [The Rick & Morty API](#). El juego comprender  una  nica pantalla y constar  de tres niveles, cuya din mica se explica a continuaci n:

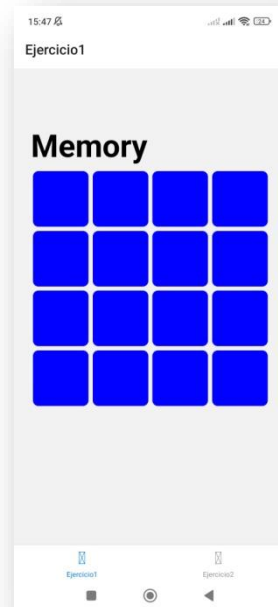
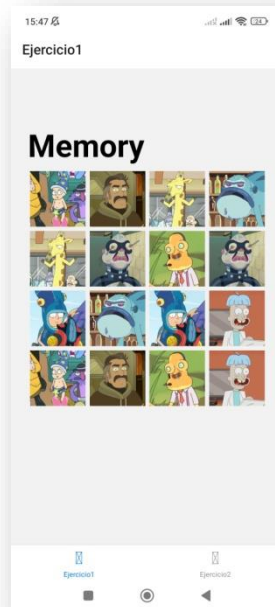
Nivel 1: Al acceder a la aplicaci n aparecer   nicamente la palabra Memory. Al pulsar sobre ella se cargar  el tablero de juego, que para este primer nivel se compondr  de 3 filas y 4 columnas:



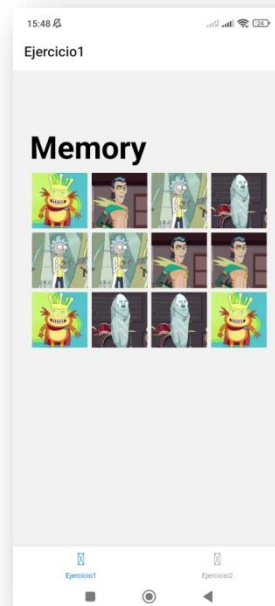
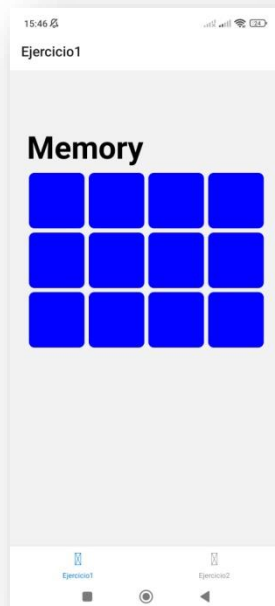
Transcurrido un segundo, se mostrarán distribuidas de forma aleatoria 6 imágenes obtenidas al azar de entre la totalidad de los personajes de The Rick & Morty API, de modo que como el tablero de juego se compone de 12 cuadrículas, cada imagen aparecerá dos veces sobre el tablero de juego. Transcurrido otro segundo, se ocultarán las imágenes y se volverá a mostrar el tablero original. El usuario procederá entonces a identificar pares de imágenes, de modo que si pulsa sobre una de las cuadrículas y, a continuación, pulsa sobre otra que contiene la misma imagen, se mostrarán ambas sobre el tablero de juego. Si en cambio las cuadrículas señaladas no contienen la misma imagen, el usuario habrá cometido un fallo. Se permitirán tres fallos acumulativos para la totalidad de juego. Si el usuario los comete, se terminará el juego y se volverá a mostrar únicamente la inscripción Memory inicial.

Si el usuario identifica los seis pares de imágenes sin equivocarse tres veces, se pasará al siguiente nivel.

Nivel 2. Este nivel contendrá un tablero de 4 x 4 cuadrículas y 8 pares de imágenes obtenidas de The Rick & Morty API. La dinámica de este nivel será la misma que la del nivel anterior. Si el usuario identifica los 8 pares de personajes, se pasará al último nivel.



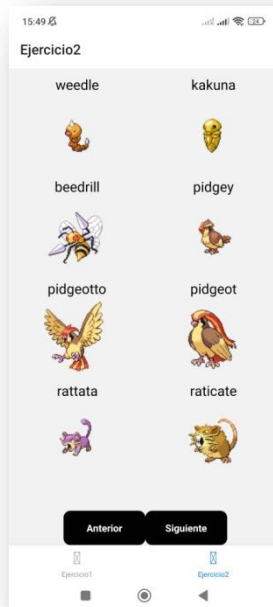
Nivel 3. Este nivel contendrá un tablero de 3 x 4 cuadrículas y 4 tríos de imágenes obtenidas de The Rick & Morty API. La dinámica de este nivel variará con respecto a la del nivel anterior en que el usuario debe identificar cada vez tres imágenes en lugar de dos. Si el usuario identifica los 4 tríos de personajes, se habrá terminado el juego.



Pestaña 2 (5 puntos): En la segunda pestaña implementarás una paginación de los personajes de Pokémon, obtenidos de la API REST pública [PokeAPI](https://pokeapi.co/). De la API se deberán obtener los siguientes datos: nombre del pokémon y las imágenes 'front_default', 'back_default', 'back_shiny' y 'front_shiny' obtenidas de los 'sprites' de cada Pokémon. Deberás implementar la lógica necesaria para que al cargarse la pestaña se muestre por pantalla el nombre y una imagen de los veinte primeros Pokémon, obtenidos de la primera página de la paginación que ofrece la propia API. De cada pokémon se deberá mostrar el nombre y la foto 'front_default'.



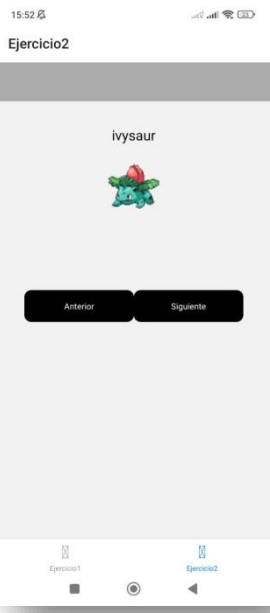
A partir de los botones 'Anterior' y 'Siguiente' se podrá navegar por la paginación proporcionada por la propia API, completando un carrousel que permitirá pasar de la primera página a la última y viceversa.



Si el usuario pulsa sobre la imagen de cualquiera de los pokémon de la página, se abrirá una pantalla modal que no cubrirá la totalidad de la pantalla, donde se mostrará el nombre y la imagen del 'front_default' del pokémon seleccionado. Para mostrar el Pokémon seleccionado, se empleará el componente Card proporcionado en el archivo Card.js, que será instanciado en la pantalla modal y se le pasará la información correspondiente utilizando para ello 'props'.

A partir de los botones 'Anterior' y 'Siguiete' del componente Card se podrá navegar entre las cuatro imágenes obtenidas de cada pokémon –imágenes 'front_default', 'back_default', 'back_shiny' y 'front_shiny'–, completando un carousel que permitirá pasar de la primera imagen a la última y viceversa.

Si se pulsa sobre el nombre del pokémon, se volverá a la pantalla de la paginación.



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 la utilización de recursos no vistos durante el curso penalizarán el valor de la nota.**