

# Programaci n Multimedia y de Dispositivos

## M viles

### Examen 1  Evaluaci n

### PROCESO DE DESARROLLO

Realizar una aplicaci n utilizando React Native y Expo implementando para ello patrones de navegaci n y realizando peticiones HTTP a dos API REST p blicas.

Las API REST a utilizar son las siguientes:

Rick and Morty API, cuya URL es la siguiente:

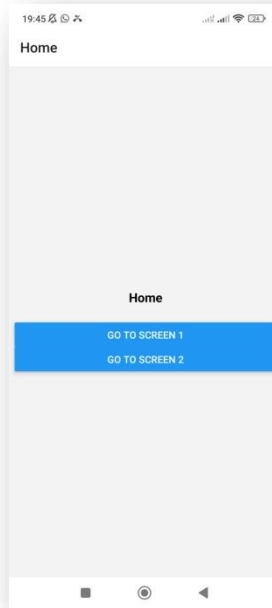
[The Rick and Morty API](#)

Ball Don't Lie API, cuya URL es la siguiente:

[BALLDONTLIE](#)

#### **Navegaci n (2 puntos):**

**Pantalla Home:** contendr  un componente Text con la inscripci n "Home" y debajo dos componentes Button con el texto "Go to screen 1" y "Go to screen 2" que permitir  acceder a los siguientes grupo de pantallas mediante un patr n de navegaci n Stack.



**Pantalla Screen1:** implementada en un patrón de navegación Stack.

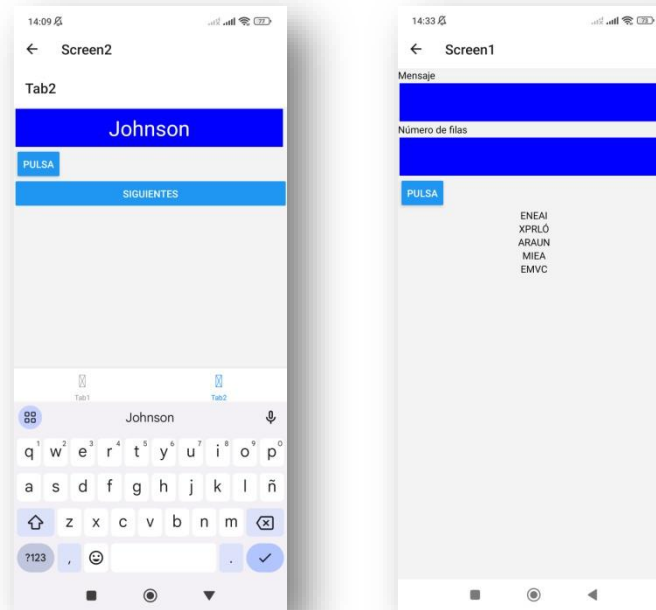
**Pantalla Screen2:** implementadas en un patrón de navegación Tab con dos pestañas Tab1 y Tab2.

El contenido de estas tres pantallas-Screen1, Tab1 y Tab2- se detalla a continuación:

**Screen1 (2,5 puntos):**

La pantalla Screen1 contendrá dos componente TextInput y debajo un componente Button con la inscripción 'Pulsa'. El el primer TextInput se indicará un mensaje de texto y en el segundo un número de filas. Se implementará la lógica necesaria para que cuando se pulse sobre el componente Button se borre el contenido introducido en los componentes TextInput y se realice una conversión del mensaje introducido de manera que se agrupe en las filas indicadas, empezando de arriba abajo. Si por ejemplo se introduce 'Examen primera evaluación' y el usuario indica 5 columnas, el mensaje quedará de la siguiente manera:

E N E A I  
X P R L Ó  
A R A U N  
M I E A  
E M V C



### Tab 1 (3 puntos):

Contendrá cinco componentes Button con la inscripción 'Ricky y Morty', 'Anterior' y 'Siguiente', 'Anteriores' y 'Siguientes', ubicados alineados verticalmente.

Al pulsarse sobre el componente Button con la inscripción 'Ricky y Morty' se realizará una petición GET a la API REST de Rick and Morty:

#### [The Rick and Morty API](#)

Se implementará la lógica necesaria para que se muestre de inicio el primer personaje de la serie, mostrando por pantalla los siguientes datos obtenidos de la respuesta a la petición GET:

-Imagen, nombre, especie y estatus del personaje.

Al pulsar sobre los componentes Button con la inscripción 'Anterior' y 'Siguiente' se podrá navegar por los primeros 20 personajes de la serie, de modo que si se llega al último de esos primeros 20 personajes y se pulsa siguiente, se pase a mostrar el primero. Si se está en el primer personaje y se pulsa anterior, se mostrará el último de esos 20 personajes. Si se pulsa sobre los componentes Button con las inscripciones 'Anteriores' y 'Siguientes', se mostrarán los 20 anteriores o siguientes personajes, de manera que se permita completar un carrusel circular y se pueda pasar de los últimos personajes de la serie a los primeros y viceversa.

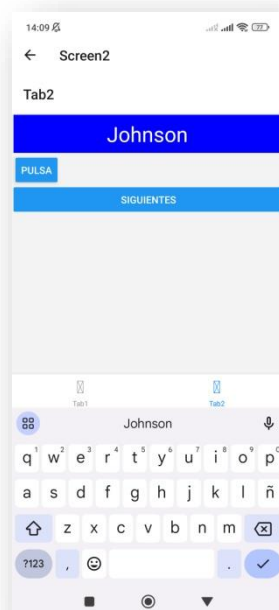


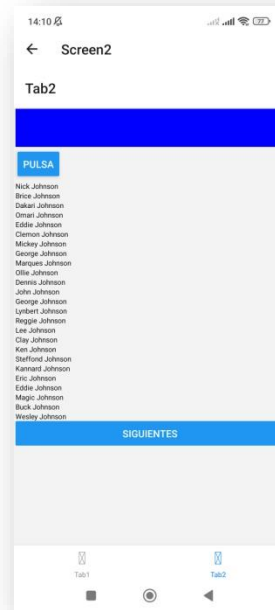
### Tab 2 (2,5 puntos):

Contendrá implementados un componente TextInput y debajo un componente Button con la inscripción “Pulsa”. Bajo este componente Button se ubicará otro con la inscripción ‘Siguietes’. Se implementará la lógica necesaria para que al pulsarse sobre el componente Button con la inscripción ‘Pulsa’ se realice una petición GET a la API REST Ball Don’t Lie con la palabra introducida en el componente TextInput. Se mostrarán entre el componente Button ‘Pulsa’ y el componente Button ‘Siguietes’ los 25 primeros jugadores obtenidos de la respuesta. Si se pulsa sobre el componente Button ‘Siguietes’, se mostrarán los siguientes 25 jugadores. En caso de llegar a los últimos jugadores, si se pulsa sobre ‘Siguietes’, se mostrarán los 25 primeros.

Nota: en caso de haber un número menor a 25, se mostrarán sólo esos jugadores.

Nota: No es necesario validar que el usuario introduce un jugador válido.





Se implementarán los eventos que sean necesarios y se utilizarán Hooks para guardar la información obtenida de la API REST. Se podrán instalar cuantas librerías sean necesarias para la implementación de la navegación y las peticiones HTTP de la aplicación. Los estilos serán libres.

## EVALUACIÓN

---

Esta prueba supondrá el 40% de la nota de la primera evaluación.

## OBSERVACIONES

---

Se entregarán en una carpeta comprimida todos los archivos generados. La entrega se realizará en Florida Oberta. Los archivos a entregar deberán ceñirse a lo que se pide explícitamente en cada enunciado del examen. Las funcionalidades no solicitadas y la utilización de estructuras como useEffect o useContext supondrán una penalización la nota.

Se permitirá la consulta de las siguientes páginas:

[W3Schools Online Web Tutorials](#)

[DevDocs API Documentation](#)

[React Native · Learn once, write anywhere](#)

[Expo](#)

**La copia de contenido de compañeros conllevará suspender la convocatoria.**