

# Creando un clon de Spotify con React Native

David Márquez Mínguez

July 9, 2023

## Contents

<b>1</b>	<b>Introducción</b>	<b>2</b>
1.1	Contexto . . . . .	2
1.2	Motivación . . . . .	2
1.3	Objetivos y limitaciones . . . . .	2
<b>2</b>	<b>Configuración del entorno de desarrollo</b>	<b>3</b>
<b>3</b>	<b>Comienzo del desarrollo</b>	<b>4</b>
3.1	Construcción del sistema de navegación . . . . .	4
3.1.1	Creación de componentes dinámicos . . . . .	4
3.2	Construcción de la primera pantalla . . . . .	5
<b>4</b>	<b>Evaluation</b>	<b>6</b>
<b>5</b>	<b>Conclusions and Future Work</b>	<b>7</b>

# 1 Introducción

## 1.1 Contexto

Este documento alberga todos los problemas que se me han planteado a la hora de construir un clon de Spotify utilizando react-native. Además de ello, se incluirá en el mismo los pasos realizados y las decisiones tomadas para cada uno de los desarrollos.

## 1.2 Motivación

La motivación principal para empezar el proyecto es aprender y mejorar mis habilidades como programador en el desarrollo de aplicaciones.

## 1.3 Objetivos y limitaciones

Puesto que aún no soy un experto en este campo, seguro que surgirán errores en el análisis y desarrollo de las soluciones propuestas. El objetivo de este documento no solo se trata de aprender, sino de dejar evidenciado cada una de las decisiones tomadas y mejorar al respecto.

## 2 Configuración del entorno de desarrollo

Antes de comenzar el proyecto, el primer paso es la configuración del entorno. Para ello accedemos a la página oficial de react-native y seguimos los pasos [4]. En mi caso, puesto que ya he trabajado con alguna aplicación móvil pequeña, no será necesaria mucha modificación.

Algo que si he tenido en cuenta y es importante comentar, es la eliminación del paquete *@react-native-community/cli*. La documentación aconseja su eliminación, ya que pueden surgir problemas inesperados, para ello ejecuto:

```
react-native-cli @react-native-community/cli
```

Antes e iniciar el proyecto, algo que es importante comentar es que React Native tiene una interfaz de línea de comandos incorporada, esto significa que se puede generar un nuevo proyecto utilizando simplemente:

```
npx react-native@latest init src
```

Una vez iniciado el proyecto y con todos los paquetes necesarios instalados, precedemos a comprobar que la ejecución funciona correctamente. Para ello nos creamos un dispositivo móvil virtual utilizando Android Studio. Una vez realizados todos los pasos ejecutamos los comandos:

```
npm start  
npm run android
```

El primero inicia Metro, un paquete de JavaScript que se incluye con React Native. Este devuelve un solo archivo JavaScript que incluye todo el código con sus dependencias. El segundo instala e inicia la aplicación en este caso en Android.

## 3 Comienzo del desarrollo

Una vez instalados todos los requerimientos previos y creado nuestro proyecto, comenzamos con el desarrollo.

### 3.1 Construcción del sistema de navegación

La primera funcionalidad a construir es el sistema de navegación entre pantallas. La aplicación va a tener un contenedor inferior donde se mostrarán los iconos de *'home'*, *'search'* y *'your library'* y podremos ir cambiando de pantalla según se vayan pulsando.

Para ello el primer paso es la instalación del paquete *@react-navigation/native* [1]. Ejecutamos el siguiente código y esperamos a que termine la instalación.

```
npm install @react-navigation/native
```

La propia construcción del sistema de navegación no tiene mucho misterio en sí, solo tenemos que meter las pantallas de nuestra aplicación en el propio contenedor de navegación. Se muestra un código de ejemplo a continuación.

```
<NavigationContainer>
  <Tab.Navigator initialRouteName='Home'>
    <Tab.Screen
      name="Home"
      component={HomeScreen}/>
  </Tab.Navigator>
</NavigationContainer>
```

En nuestro caso, puesto que tendremos tres pantallas principales, tendremos que crear otros dos componentes del tipo *Tab.Navigator*, lo que supondrá la problemática de componentes dinámicos que se comentará más adelante.

Una vez definidos los componentes de navegación, queda darle estilo y agregar los iconos correspondientes a los mismos. Para los estilos emplearé el mismo color de fondo que usa Spotify, así como el nombre de los botones. Para el tema de los iconos usaré un paquete que no necesita previa instalación, *'react-native-vector-icons'*.

Para el uso de este paquete lo único que se debe hacer es modificar el archivo *./android/app/build.gradle* y añadir en la última línea *apply from: ../../node\_modules/react-native-vector-icons/fonts.gradle* [2].

#### 3.1.1 Creación de componentes dinámicos

Cada uno de los iconos del sistema de navegación es un propio componente. ¿Hay que crear un componente distinto para cada pantalla de la aplicación? ¿Qué ocurre entonces cuando queramos introducir iconos dependiendo del estado de enfoque? La solución más sencilla a este problema es crear componentes dinámicos [3].

Lo primero a definir es una constante que almacenara los componentes disponibles en función de los imports definidos. En este caso *FoundationIcon*, *OcticonsIcon*, *EvilIcons* y *Ionicons* son imports del paquete *react-native-vector-icons*.

```
const iconComponents = {
  foundationIcon: FoundationIcon,
  octiconsIcon: OcticonsIcon,
  evilIcons: EvilIcons,
  ionicons: Ionicons
}
```

Lo segundo es retornar el propio componente en cuestión. Para ello creo una función que dadas unas propiedades definidas por parámetro, cree ese componente a medida. El último paso será llamar al componente definido.

```
function MyIconComponent(props) {  
  const MyComponent = iconComponents[props.component];  
  return <MyComponent name={props.name} style={props.style} />;  
}
```

### 3.2 Construcción de la primera pantalla

a

## 4 Evaluation

We did some experiments ...

## 5 Conclusions and Future Work

From our experiments we can conclude that ...

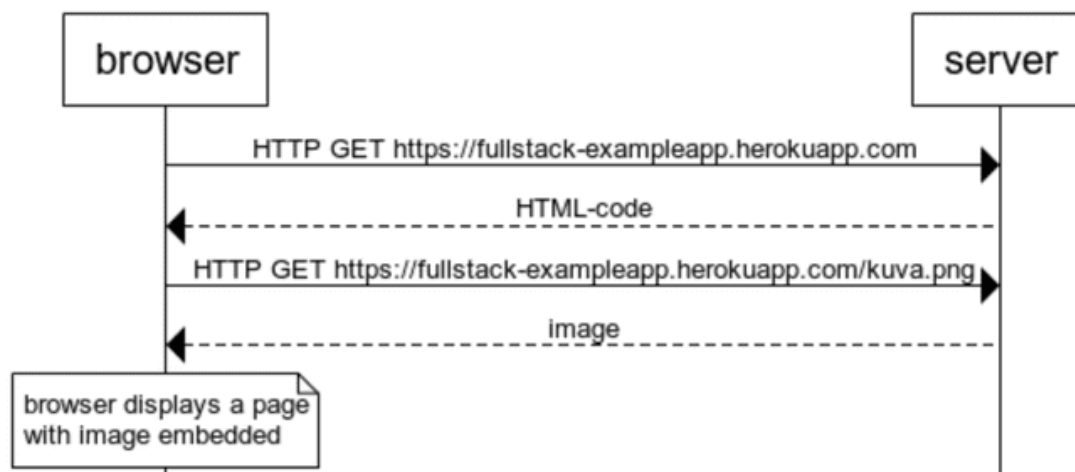


Figure 1: testing

## References

- [1] R. Native. React navigation docs. <https://reactnavigation.org/docs/getting-started/>. [Last access 07/July/2023].
- [2] oblador. React native vector icons. <https://github.com/oblador/react-native-vector-icons#android>. [Last access 07/July/2023].
- [3] React. Jsx in depth. <https://legacy.reactjs.org/docs/jsx-in-depth.html#choosing-the-type-at-runtime>. [Last access 07/July/2023].
- [4] M. O. Source. Setting up the development environment. <https://reactnative.dev/docs/environment-setup?guide=native>. [Last access 23/June/2023].