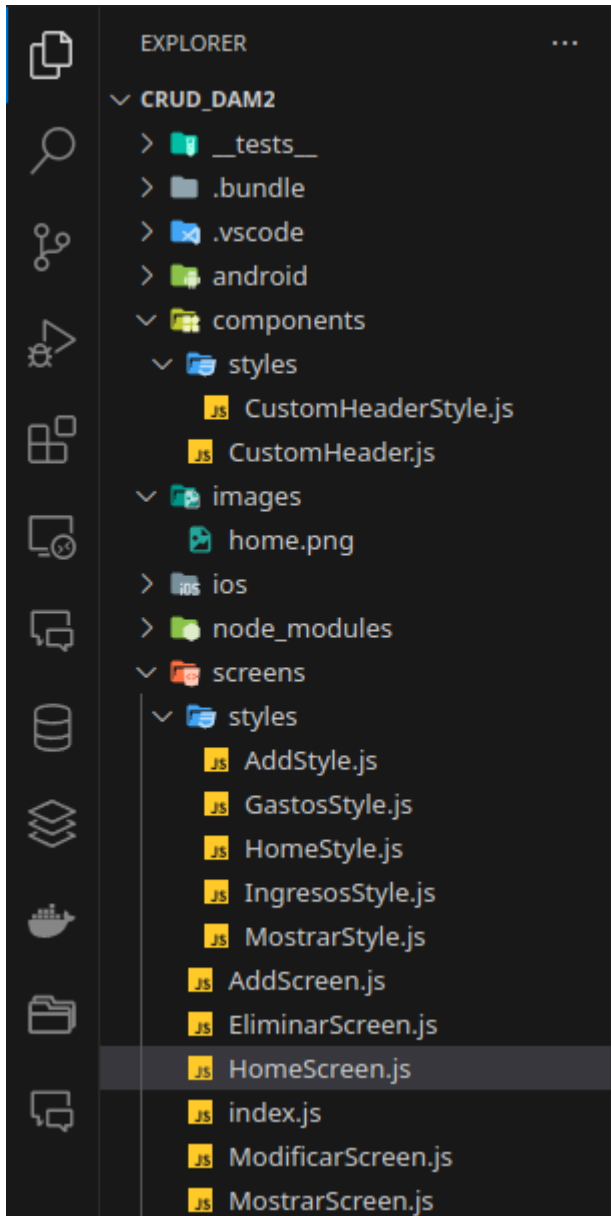


Curso de React Native - Semana 3: Estructuración de una Aplicación

Estructura de Carpetas y Archivos

Organizar los archivos y carpetas de manera eficiente es crucial en cualquier proyecto de React Native. Una estructura típica incluye directorios separados para componentes, pantallas, servicios (APIs), y recursos (imágenes, estilos).



Curso de React Native - Semana 3: Estructuración de una Aplicación

Navegación y Rutas en React Native

La navegación entre pantallas es un aspecto fundamental de las aplicaciones móviles. React Native ofrece varias soluciones para gestionar la navegación, como React Navigation, una de las librerías más populares para este fin.

Aquí tienes un ejemplo de cómo hacerlo:

1. **Instalación de React Navigation:** Abre una terminal en el directorio de tu proyecto React Native y ejecuta los siguientes comandos para instalar React Navigation y sus dependencias:

```
npm install @react-navigation/native
npm install react-native-screens react-native-safe-area-context
npm install @react-navigation/native-stack
```

Para proyectos Expo, también debes instalar `expo install react-native-gesture-handler react-native-reanimated`.

2. **Configuración de un Stack Navigator:** Ahora, configura un Stack Navigator básico con dos pantallas. Crea dos archivos para tus pantallas, por ejemplo, `HomeScreen.js` y `DetailsScreen.js`.

```
javascriptCopy code
// HomeScreen.js
import React from 'react';
import { View, Text, Button } from 'react-native';

function HomeScreen({ navigation }) {
  return (
    <View style={{ flex: 1, alignItems: 'center', justifyContent: 'center' }}>
      <Text>Home Screen</Text>
      <Button
        title="Go to Details"
        onPress={() => navigation.navigate('Details')}
      />
    </View>
  );
}

export default HomeScreen;
javascriptCopy code
// DetailsScreen.js
import React from 'react';
import { View, Text, Button } from 'react-native';

function DetailsScreen({ navigation }) {
  return (
    <View style={{ flex: 1, alignItems: 'center', justifyContent: 'center' }}>
      <Text>Details Screen</Text>
      <Button
        title="Go to Home"
        onPress={() => navigation.navigate('Home')}
      />
    </View>
  );
}
```

Curso de React Native - Semana 3: Estructuración de una Aplicación

```
export default DetailsScreen;
```

3. **Configuración del Navigator en tu App:** En tu archivo principal (por lo general `App.js`), configura el stack navigator para incluir estas dos pantallas:

```
javascriptCopy code
import React from 'react';
import { NavigationContainer } from '@react-navigation/native';
import { createNativeStackNavigator } from '@react-navigation/native-stack';
import HomeScreen from './HomeScreen';
import DetailsScreen from './DetailsScreen';

const Stack = createNativeStackNavigator();

function App() {
  return (
    <NavigationContainer>
      <Stack.Navigator initialRouteName="Home">
        <Stack.Screen name="Home" component={HomeScreen} />
        <Stack.Screen name="Details" component={DetailsScreen} />
      </Stack.Navigator>
    </NavigationContainer>
  );
}

export default App;
```

Este es un ejemplo básico de cómo configurar la navegación entre dos pantallas en React Native utilizando React Navigation. A partir de aquí, puedes expandir y personalizar la navegación según las necesidades de tu aplicación.

RESUMEN

1. Creación de Pantallas:

- Se crearon dos pantallas/componentes (`HomeScreen.js` y `DetailsScreen.js`) que actúan como pantallas individuales en la aplicación. Cada pantalla es un componente de React que devuelve una interfaz de usuario.

2. Configuración de Rutas de Navegación:

- Se utilizó un Stack Navigator (`createNativeStackNavigator`) para configurar las rutas de navegación.
- Dentro del `NavigationContainer`, se definen las rutas de navegación asociando cada pantalla con un nombre de ruta único (`Home` y `Details`).

3. Navegación entre Pantallas:

- En cada pantalla, se utilizó un botón para navegar entre pantallas. Usando la función `navigation.navigate('RouteName')`, se puede navegar a la pantalla deseada.

Curso de React Native - Semana 3: Estructuración de una Aplicación

- Por ejemplo, en `HomeScreen.js`, un botón lleva al usuario a `DetailsScreen.js` al presionar "Go to Details".

Este código proporciona una base sólida para entender cómo se crean pantallas individuales y cómo se configura la navegación básica entre estas pantallas en una aplicación React Native utilizando React Navigation. Puedes personalizar y expandir este código para incluir más pantallas, pasar parámetros entre pantallas, configurar estilos de navegación y más, según las necesidades específicas de tu proyecto.