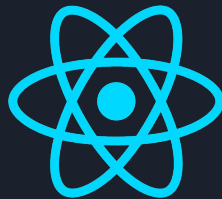


React Native: Navegación en la app



i2T

¿Para qué sirve la navegación entre pantallas?

La navegación es un pilar fundamental en la experiencia de usuario (UX) de cualquier aplicación móvil. Para entender cómo funciona la navegación podemos imaginar que la app es una casa y las pantallas son las habitaciones. La navegación serían los pasillos, puertas y escaleras que permiten a los usuarios moverse de un lugar a otro de forma **intuitiva y lógica**.

Una buena navegación permite:

- **Estructurar la información:** Organiza el contenido de tu aplicación en diferentes pantallas, haciendo que sea más fácil de entender y consumir.
- **Guiar al usuario:** Lleva al usuario a través de las diferentes funcionalidades de la aplicación de manera fluida.
- **Mejorar la experiencia de usuario:** Una navegación predecible y coherente hace que la aplicación se sienta familiar y fácil de usar, lo que aumenta la satisfacción del usuario.
- **Crear flujos de tareas:** Permite a los usuarios completar acciones complejas que requieren múltiples pasos, como un proceso de compra o un formulario de registro.



Diferencias Clave: React Native vs. React Web

Aunque React Native y React (para la web) comparten la misma filosofía de componentes, la forma en que manejan la navegación es fundamentalmente diferente. Esto se debe a que operan en plataformas distintas (móviles vs. navegadores web).

	React (Web)	React Native (Móvil)
Entorno	Se ejecuta en un navegador web.	Se ejecuta como una aplicación nativa en un dispositivo (iOS o Android).
Mecanismo	Utiliza las URLs y el historial del navegador. La librería más común es React Router .	No hay URLs. Simula una "pila" de pantallas, similar a cómo funcionan las apps nativas.



Comenzar a utilizar Navigation

Hay 3 tipos de navegación dentro de react native creados por la comunidad

- Stack Navigator (Navegador de Pila)
- Tab Navigator (Navegador de Pestañas)
- Drawer Navigator (Navegador de Cajón Lateral)

Los más utilizados son el stack y tab navigator.

El Stack Navigator es ideal para flujos en los que el usuario avanza y retrocede entre pantallas, como en un proceso de registro o al hacer clic en un elemento de una lista para ver sus detalles. Cada nueva pantalla se "apila" encima de la anterior.

El Tab Navigator es perfecto para mostrar diferentes secciones principales de una aplicación que tienen una jerarquía similar, como "Inicio", "Perfil" y "Configuración". Generalmente se muestra como una barra de pestañas en la parte inferior de la pantalla.



Uso de stack navigator

Necesitaremos instalar el módulo de node para poder utilizarla.

@react-navigation/stack - @react-navigation/native-stack - @react-navigation/elements

para poder utilizar esta librería necesitamos crear un archivo AppNavigator o AppNavigation en el cual importamos la librería y creemos un stackNavigator, debe tener una sintaxis de este estilo

```
import { NavigationContainer } from '@react-navigation/native';  
import { createNativeStackNavigator } from '@react-navigation/native-stack';
```

```
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>  
  );  
}
```



Modo de uso del app Navigator

Para usar el navigator necesitamos llamar a la referencia de a que pantalla nos queremos mover por ejemplo

```
import { useNavigation } from '@react-navigation/native';
```

```
function HomeScreen() {  
  const navigation = useNavigation();  
  
  return (  
    <View>  
      <Button  
        title="Ir a Detalles"  
        onPress={() => navigation.navigate('Details')}  
      />  
    </View>  
  );  
}
```



Ejemplo usando props

```
import React from 'react';
import { View, Text, Button, StyleSheet } from 'react-native';


// La prop 'navigation' la inyecta automáticamente React Navigation
function UsersListScreen({ navigation }) {
  return (
    <View style={styles.container}>
      <Text style={styles.title}>Lista de Usuarios</Text>

      {/* Botón para navegar al perfil del Usuario con ID 101 */}
      <Button
        title="Ver perfil de Ana (ID: 101)"
        onPress={() => {
          // Navegamos a la ruta 'UserProfile' y le pasamos el 'userId'
          navigation.navigate('UserProfile', { userId: 101 });
        }}
      />

      {/* Botón para navegar al perfil del Usuario con ID 205 */}
      <Button
        title="Ver perfil de Juan (ID: 205)"
        onPress={() => {
          navigation.navigate('UserProfile', { userId: 205 });
        }}
      />
    </View>
  );
}
```

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    alignItems: 'center',
    justifyContent: 'center',
    gap: 15,
  },
  title: {
    fontSize: 22,
    fontWeight: 'bold',
    marginBottom: 20,
  },
});


export default UsersListScreen;
```



```
import React from 'react';
import { View, Text, StyleSheet } from 'react-native';
// React Navigation también inyecta la prop 'route' function UserProfileScreen({
route })
{ // Obtenemos el userId de los parámetros de la ruta // Usamos el operador de
encadenamiento opcional (?) por seguridad
```

```
const { userId } = route.params;
return (
  <View style={styles.container}>
    <Text style={styles.title}>Perfil de Usuario</Text>
    <Text style={styles.text}>El ID del usuario seleccionado es: </Text>
    <Text style={styles.userIdText}>{userId}</Text>
  </View> ); }
```

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    alignItems: 'center',
    justifyContent: 'center',
  },
  title: { fontSize: 22,
    fontWeight: 'bold',
    marginBottom: 20,
  }, text: {
    fontSize: 18,
  },
  userIdText: {
    fontSize: 24,
    fontWeight: 'bold',
    color: '#007BFF',
    marginTop: 10,
  },
});
export default UserProfileScreen;
```

```
import { createNativeStackNavigator } from '@react-navigation/native-stack';
import UsersListScreen from './UsersListScreen';
import UserProfileScreen from './UserProfileScreen';
```

```
const Stack = createNativeStackNavigator();
```

```
function AppNavigator() {
  return (
    <Stack.Navigator>
      <Stack.Screen
        name="UsersList"
        component={UsersListScreen}
        options={{ title: 'Usuarios' }}
      />
      <Stack.Screen
        name="UserProfile"
        component={UserProfileScreen}
        options={{ title: 'Perfil del Usuario' }}
      />
    </Stack.Navigator>
  );
}
```

Práctica

Reutilizar la pantalla creada en la práctica anterior y crear la pantalla de la derecha.

- Observar y dividir en componentes la pantalla, reutilizar componentes ya creados.
- agregar navegación mediante Stack Navigator. moverse de la pantalla principal a la de calificaciones y posibilidad de volver a la pantalla anterior

Calificaciones

Primer Año

Matemática

Comunicación

Administración

Logica y Estructura de Datos

Ingeniería de Software II

Tecnología de la Información

Inglés Técnico I

Sistemas Operativos

Unidad de Definición Institucional I

Parcial 1 : ☐

Parcial 2 : ☐

Tp 1 : ☐

Tp 2 : ☐

Rec 1 : ☐

Rec 1 : ☐

Rec 1 : ☐

Integrador : ☐

Finales : ☐

Segundo Año

Innovación y Desarrollo Emprendedor

Inglés Técnico II

Unidad de Definición Institucional II

Problemática Socio Contemporánea

Estadística

Volver

Mesas

Cerrar Sesión

IES INSTITUTO DE ESTUDIOS SUPERIORES

Universidad de Concepción del Uruguay

Centro Regional Santa Fe