

# TIPOS DE NAVEGACION

En una aplicación Expo con React Native, la navegación se puede implementar de diferentes formas utilizando componentes de la librería react-navigation. Los dos más comunes son **Stack Navigator** y **Drawer Navigator**, pero también existen otras opciones como **Tab Navigator** y **Bottom Tabs**.

## 1. Stack Navigator

- **Función:** El **Stack Navigator** funciona de forma similar a la navegación de páginas en un navegador web, donde cada pantalla nueva se apila sobre la anterior, y el usuario puede navegar hacia atrás en la pila (stack) de pantallas.
- **Usos comunes:** Es ideal para flujos de aplicaciones donde las pantallas siguen un orden lineal, como pasos de un formulario o detalles de un producto.
- **Navegación:** Se navega usando un sistema de push y pop, lo que significa que puedes agregar nuevas pantallas encima de la actual y retroceder.
- **Características principales:**
  - Tiene animaciones de transición entre pantallas.
  - Proporciona una experiencia de navegación común en la mayoría de las aplicaciones móviles.
  - Es fácil de implementar cuando hay un flujo secuencial claro entre pantallas.

## Ejemplo de uso:

```
import { createStackNavigator } from '@react-navigation/stack';

const Stack = createStackNavigator();

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

## 2. Drawer Navigator

- **Función:** El **Drawer Navigator** permite que las pantallas se naveguen a través de un menú lateral (desplegable desde la izquierda o derecha), lo que le da una estructura más libre a la navegación.
- **Usos comunes:** Se utiliza cuando quieres que los usuarios tengan acceso rápido a diferentes secciones de la aplicación, como configuración, perfil, o secciones menos críticas.
- **Navegación:** En lugar de apilar pantallas, se accede a cada pantalla de manera independiente a través del menú desplegable.
- **Características principales:**
  - Proporciona un acceso fácil a varias secciones de la aplicación desde cualquier parte.
  - El menú puede ser personalizado para incluir imágenes, nombres de usuarios, o enlaces adicionales.
  - Funciona bien para aplicaciones con múltiples secciones que no dependen de un flujo lineal.

## Ejemplo de uso:

```
import { createDrawerNavigator } from '@react-navigation/drawer';

const Drawer = createDrawerNavigator();

function MyDrawer() {
  return (
    <Drawer.Navigator>
      <Drawer.Screen name="Home" component={HomeScreen} />
      <Drawer.Screen name="Settings" component={SettingsScreen} />
    </Drawer.Navigator>
  );
}
```

### 3. Tab Navigator (Bottom Tabs)

- **Función:** El **Tab Navigator** coloca una barra de pestañas en la parte inferior o superior de la pantalla para permitir la navegación entre diferentes pantallas.
- **Usos comunes:** Es ideal cuando las pantallas principales de la aplicación deben estar accesibles todo el tiempo, como una aplicación de redes sociales donde tienes secciones de "Feed", "Perfil", "Mensajes", etc.
- **Navegación:** Cada pestaña representa una pantalla independiente, pero se puede combinar con un Stack Navigator para profundizar en flujos.
- **Características principales:**
  - Proporciona una navegación rápida entre las pantallas principales.
  - Muy común en aplicaciones con secciones diferenciadas que requieren acceso rápido.
  - Puede incluir íconos y etiquetas para cada pestaña.

## Ejemplo de uso:

```
import { createBottomTabNavigator } from '@react-navigation/bottom-tabs';

const Tab = createBottomTabNavigator();

function MyTabs() {
  return (
    <Tab.Navigator>
      <Tab.Screen name="Home" component={HomeScreen} />
      <Tab.Screen name="Profile" component={ProfileScreen} />
    </Tab.Navigator>
  );
}
```

## 4. Material Top Tabs Navigator

- **Función:** Este componente es similar al **Bottom Tabs**, pero coloca las pestañas en la parte superior, como en algunas aplicaciones de mensajería o redes sociales.
- **Usos comunes:** Se utiliza cuando las secciones de la aplicación deben estar disponibles en la parte superior de la pantalla. Es útil para contenido categorizado, como "Chats", "Estados", "Llamadas" en aplicaciones de mensajería.
- **Navegación:** Igual que el **Bottom Tabs**, pero con pestañas arriba.
- **Características principales:**
  - Facilita la navegación entre diferentes secciones de una aplicación.
  - Puede incluir deslizamiento para cambiar entre pestañas.

## Ejemplo de uso:

```
import { createMaterialTopTabNavigator } from '@react-navigation/material-top-tabs';

const Tab = createMaterialTopTabNavigator();

function MyTopTabs() {
  return (
    <Tab.Navigator>
      <Tab.Screen name="Feed" component={FeedScreen} />
      <Tab.Screen name="Messages" component={MessagesScreen} />
    </Tab.Navigator>
  );
}
```

## Comparativa de los navegadores

Navegador	Flujo Secuencial	Acceso Rápido a Secciones	Ideal para	Personalización
Stack Navigator	✓	✗	Flujos lineales (formulario, detalles)	Moderada (transiciones)
Drawer Navigator	✗	✓	Aplicaciones con muchas secciones (configuración, perfil)	Alta (menú personalizado)
Bottom Tabs	✗	✓	Aplicaciones con secciones principales accesibles todo el tiempo	Alta (íconos y etiquetas)
Top Tabs	✗	✓	Aplicaciones con categorías o contenido similar que cambia frecuentemente	Moderada (deslizamiento)

## Conclusión

- **Stack Navigator** es ideal para flujos secuenciales o cuando tienes pantallas que dependen unas de otras.
- **Drawer Navigator** se usa mejor cuando tienes muchas secciones a las que quieres acceder desde cualquier pantalla de la aplicación.

- **Bottom Tabs** y **Top Tabs** son útiles cuando las pantallas principales de la aplicación deben ser fácilmente accesibles y no siguen un flujo lineal.

En muchos casos, se puede combinar varios de estos navegadores para crear una estructura más compleja en tu aplicación. Por ejemplo, puedes tener un **Tab Navigator** como navegación principal y luego usar **Stack Navigators** dentro de cada pestaña para manejar el flujo interno de pantallas.