



UTT

UNIVERSIDAD TECNOLÓGICA DE TIJUANA

GOBIERNO DE BAJA CALIFORNIA

Navegación

PRESENTADO POR:

García Arreola Howard Isaí

GRUPO: 9B

MATERIA:

Desarrollo para Dispositivos inteligentes

PROFESOR:

Ray Brunett Parra Galaviz

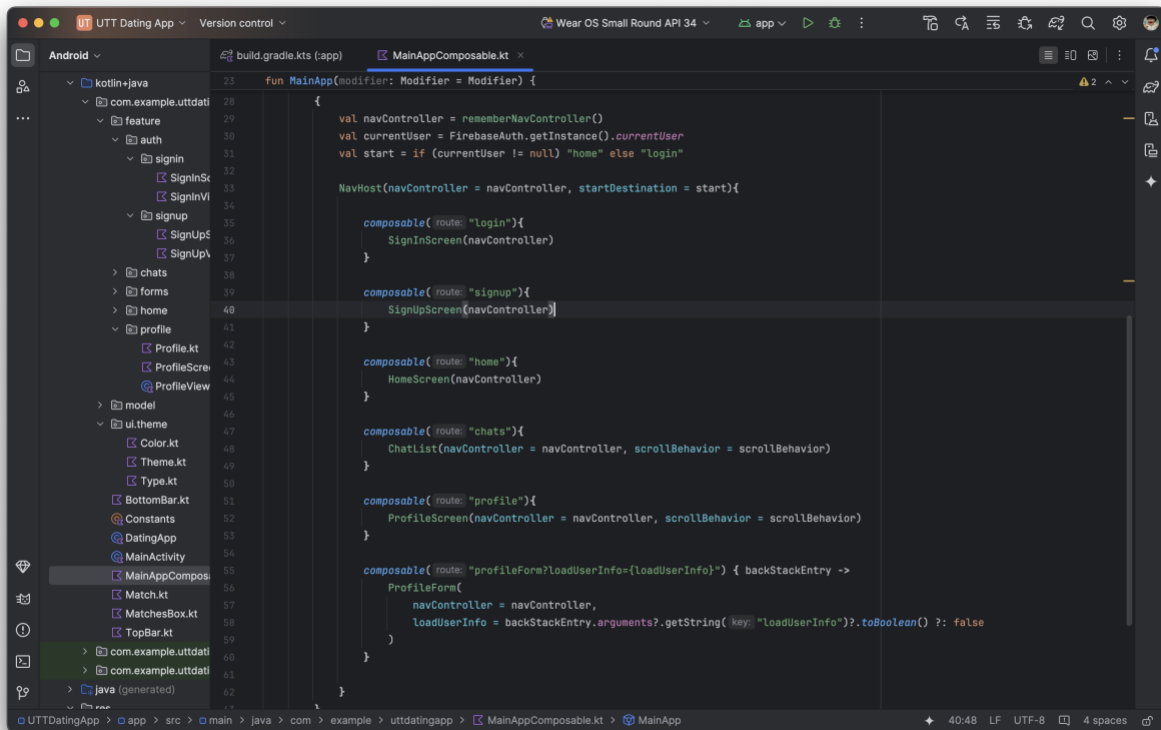
FECHA:

30 de octubre de 2024

Para tener navegación dentro de nuestra aplicación, lo que se puede hacer es utilizar la librería de navigation:

```
implementation(libs.swipe)
implementation(libs.navigation.compose)
implementation(libs.kotlinx.serialization.json)
```

Esta nos dará la posibilidad de navegar dentro de todas nuestras pantallas que realicemos dentro de nuestros proyectos. Para ello, se debe declarar un NavHost, el cual lo usaremos para establecer rutas y navegar a cada una de ellas con el nombre que le demos a cada una:



Debemos de pasar nuestro NavController a cada una de ellas de igual forma para poder modificar nuestro stack de pantallas. Esta es la importancia de Navhost y NavController:

NavHost:

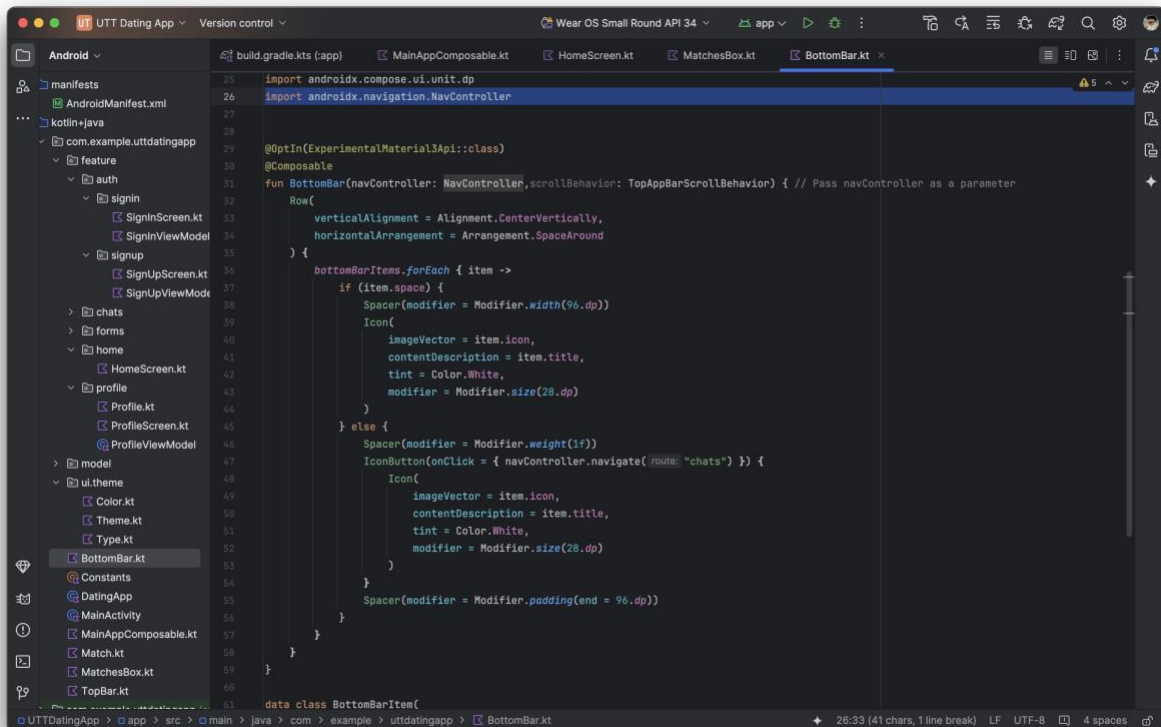
- **Contenedor de pantallas:** Actúa como un contenedor principal para todas las pantallas de la aplicación.

- **Organización:** Organiza la estructura jerárquica de las pantallas, definiendo la relación padre-hijo entre ellas.
- **Navegación eficiente:** Facilita la navegación entre pantallas, simplificando el proceso de gestión de la pila de navegación.

NavController:

- **Controlador de navegación:** Es el encargado de controlar el flujo de navegación entre las pantallas definidas en el NavHost.
- **Pila de navegación:** Maneja la pila de navegación, permitiendo al usuario retroceder a pantallas anteriores.
- **Argumentos y deep links:** Permite pasar datos entre pantallas y navegar directamente a una pantalla específica utilizando deep links.
- **Acciones de navegación:** Permite definir acciones personalizadas para realizar tareas adicionales durante la navegación.

Una forma de utilizar esta navegación es el botones, nos mismos los podemos situar dentro de un BottomBar, estemismo lo podemos utilizar con un Scaffold:



De esta manera combinamos la funcionalidad de un BottomBar y de la navegación, navegando en nuestras pantallas y mostrando el contenido necesario.

un **Scaffold** es un componente predefinido que proporciona una estructura básica para una interfaz de usuario de una aplicación móvil. Actúa como un andamio o esqueleto, proporcionando elementos comunes como:

- **AppBar:** La barra superior que puede contener el título de la pantalla, botones de navegación, y otros elementos de acción.
- **BottomBar:** La barra inferior que puede contener botones de navegación o acciones relacionadas con la aplicación.
- **FloatingActionButton:** Un botón flotante que se coloca sobre el contenido principal, generalmente usado para acciones primarias.
- **Snackbar:** Un mensaje temporal que aparece en la parte inferior de la pantalla, indicando al usuario alguna acción o error.
- **Drawer:** Un menú lateral que se despliega desde el borde de la pantalla.