

Implementação PlanetApp

QXD0276 - Desenvolvimento de Software para Dispositivos Móveis

Universidade Federal do Ceará - *Campus* Quixadá

Prof. Francisco Victor da Silva Pinheiro
victorpinheiro@ufc.br

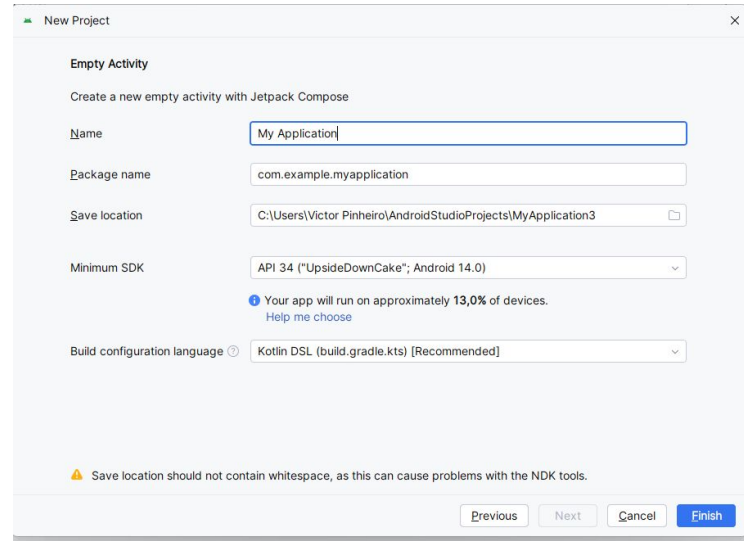


Agenda

- Passo 1: Criar o Projeto no Android Studio
- Passo 2: Configuração do Gradle
- Passo 3: Estrutura do Projeto
- Passo 4: Criar o Modelo
- Passo 5: Criar os Componentes
 - PlanetListItem
 - BottomNavigationBar
 - TopAppBarWithMenu
- Passo 6: Criar Telas
- Passo 7: Configurar Navegação
- Passo 8: Configurar MainActivity
- Passo 9: Adicionar um Ícone Personalizado ao Aplicativo
- Execução do PlanetApp

Passo 1: Criar o Projeto no Android Studio

- Abra o Android Studio.
- Crie um novo projeto:
 - Selecione a opção **Empty Compose Activity**.
 - Nomeie o projeto como **PlanetApp**.
 - Configure o pacote como, por exemplo, ***com.example.planetapp***.
 - Certifique-se de que a linguagem está definida como Kotlin e que a opção Use Jetpack Compose está marcada.
- Configure o mínimo SDK suportado (recomendo API 21 ou superior).
- Link para imagens dos planetas
- https://drive.google.com/drive/folders/1Jhm-amMpqKiB1V_LCi8nr2XGpMAcsqcv?usp=sharing



Passo 2: Configuração do Gradle

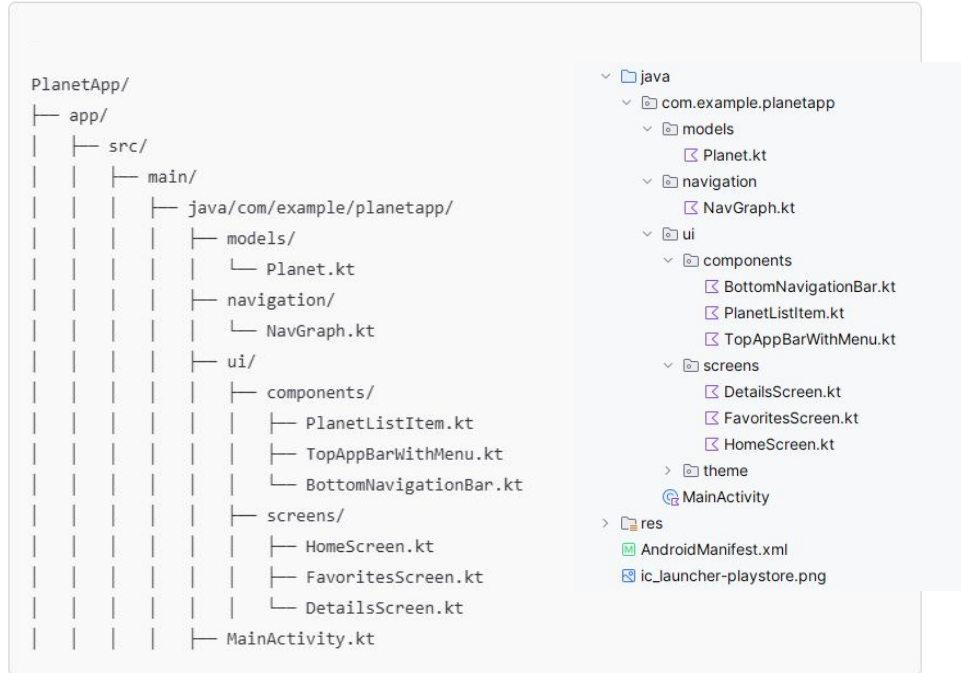
- No arquivo build.gradle do módulo, verifique se as dependências do Compose estão atualizadas:

```
implementation ( "androidx.compose.ui:ui:1.5.0" )
implementation ( "androidx.compose.material3:material3:1.1.1" )
implementation ( "androidx.navigation:navigation-compose:2.5.3" )
implementation ( "androidx.lifecycle:lifecycle-runtime-compose:2.6.1" )
implementation ( "androidx.activity:activity-compose:1.7.2" )
```

- Sincronize o projeto para baixar as dependências.

Passo 3: Estrutura do Projeto

- Organize o projeto para separação lógica entre telas, componentes, modelos e navegação:



Passo 4: Criar o Modelo

- Crie um arquivo Planet.kt na pasta models/:

```
package com.example.planetapp.models

import com.example.planetapp.R

data class Planet(
    val id: Int,
    val name: String,
    val type: String,
    val galaxy: String,
    val distanceFromSun: String,
    val diameter: String,
    val characteristics: String,
    val imageRes: Int,
    var isFavorite: Boolean = false
)
```

Passo 4: Criar o Modelo

- Adicione uma lista mock de planetas:

```
val planetList = listOf(
    Planet (
        id = 1,
        name = "Earth",
        type = "Terrestrial",
        galaxy = "Milky Way",
        distanceFromSun = "149.6 million km",
        diameter = "12,742 km",
        characteristics = "Supports life, has water and atmosphere." ,
        imageRes = R.drawable.terra
    ),
)
```

Passo 5: Criar os Componentes - PlanetListItem Parte 1

- Arquivo:
PlanetListItem.kt na
pasta components/:

```
package com.example.planetapp.ui.components

import androidx.compose.foundation.Image
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.shape. CircleShape
import androidx.compose.material.icons. Icons
import androidx.compose.material.icons.filled. Favorite
import androidx.compose.material.icons.filled. FavoriteBorder
import androidx.compose.material3.*
import androidx.compose.runtime. Composable
import androidx.compose.ui. Alignment
import androidx.compose.ui. Modifier
import androidx.compose.ui.draw.clip
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.text.style. TextAlign
import androidx.compose.ui.unit.dp
import com.example.planetapp.models. Planet
```


Passo 5: Criar os Componentes - PlanetListItem Parte 2

```
@Composable
fun PlanetListItem(
    planet: Planet,
    onPlanetSelected: (Planet) -> Unit,
    onFavoriteToggle: (Planet) -> Unit
) {
    Card(
        modifier = Modifier
            .fillMaxWidth()
            .padding(8.dp),
        elevation = CardDefaults.cardElevation(4.dp),
        colors = CardDefaults.cardColors(containerColor =
MaterialTheme.colorScheme.surface)
    ) {
        Column(
            modifier = Modifier
                .fillMaxWidth()
                .padding(16.dp)
        ) {
```

```
// Imagem e título
Row(verticalAlignment = Alignment.CenterVertically) {
    Image(
        painter = painterResource(id = planet.imageRes),
        contentDescription = "${planet.name} Image",
        modifier = Modifier
            .size(80.dp)
            .clip(CircleShape)
    )
    Spacer(modifier = Modifier.width(16.dp))
    Column(modifier = Modifier.weight(1f)) {
        Text(
            text = planet.name,
            style = MaterialTheme.typography.titleLarge,
            color = MaterialTheme.colorScheme.primary
        )
        Text(
            text = "Galaxy: ${planet.galaxy}",
            style = MaterialTheme.typography.bodySmall,
            color = MaterialTheme.colorScheme.onSurfaceVariant
        )
    }
}
```

Passo 5: Criar os Componentes - PlanetListItem Parte 3

```

IconButton(onClick = { onFavoriteToggle(planet) }) {
    Icon(
        imageVector = if (planet.isFavorite) Icons.Default.Favorite else
        Icons.Default.FavoriteBorder,
        contentDescription = "Toggle Favorite",
        tint = if (planet.isFavorite) MaterialTheme.colorScheme.primary else
        MaterialTheme.colorScheme.onSurfaceVariant
    )
}

Spacer(modifier = Modifier.height(16.dp))

// Informações adicionais
Column(modifier = Modifier.padding(horizontal = 8.dp)) {
    Text(
        text = "Type: ${planet.type}",
        style = MaterialTheme.typography.bodyMedium,
        color = MaterialTheme.colorScheme.onSurface
    )
    Text(
        text = "Distance from Sun: ${planet.distanceFromSun}",
        style = MaterialTheme.typography.bodyMedium,
        color = MaterialTheme.colorScheme.onSurface
    )
    Text(
        text = "Diameter: ${planet.diameter}",
        style = MaterialTheme.typography.bodyMedium,
        color = MaterialTheme.colorScheme.onSurface
    )
}

```

```

Spacer(modifier = Modifier.height(8.dp))

Text(
    text = planet.characteristics,
    style = MaterialTheme.typography.bodySmall,
    color = MaterialTheme.colorScheme.onSurfaceVariant,
    textAlign = TextAlign.Justify
)

Spacer(modifier = Modifier.height(16.dp))

// Botão "Ver mais"
Button(
    onClick = { onPlanetSelected(planet) },
    modifier = Modifier.align(Alignment.End),
    colors = ButtonDefaults.buttonColors(containerColor =
    MaterialTheme.colorScheme.primary)
) {
    Text(text = "Ver mais sobre ${planet.name}")
}
}
}

```

Passo 5: Criar os Componentes - BottomNavigationBar

- Arquivo:
BottomNavigation
Bar.kt na pasta
components/:

```
@Composable
fun BottomNavigationBar(navController: NavController) {
    val screens = listOf(
        BottomBarScreenHome,
        BottomBarScreenFavorites
    )

    val currentDestination = navController.currentBackStackEntryAsStat().value?.destination

    NavigationBar {
        screens.forEach { screen ->
            NavigationBarItem(
                selected = currentDestination?.hierarchy?.any { it.route == screen.route } == true,
                onClick = {
                    navController.navigate(screen.route) {
                        popUpTo(navController.graph.startDestinationId) { saveState = true }
                        launchSingleTop = true
                        restoreState = true
                    }
                },
                icon = screen.icon,
                label = { Text(screen.label) },
                colors = NavigationBarItemDefaultColors(
                    selectedIconColor = MaterialTheme.colorScheme.primary,
                    unselectedIconColor = MaterialTheme.colorScheme.onSurfaceVariant
                )
            )
        }
    }
}
```

Passo 5: Criar os Componentes - TopAppBarWithMenu

```
@ExperimentalMaterial3Api
@Composable
fun TopAppBarWithMenu(
    onSettingsClick: () -> Unit,
    onHelpClick: () -> Unit
) {
    var expanded by remember { mutableStateOf(false) }

    TopAppBar(
        title = {
            Row(verticalAlignment = Alignment.Top) {
                Image(
                    painter = painterResource(id = R.drawable.icon),
                    contentDescription = "App Logo",
                    modifier = Modifier
                        .size(30.dp)
                        .padding(end = 8.dp)
                )
                Text(
                    text = "PlanetApp",
                    style = MaterialTheme.typography.titleLarge,
                    maxLines = 1,
                    overflow = TextOverflow.Ellipsis
                )
            }
        },
    ),
```

```
actions = {
    IconButton (onClick = { expanded = true }) {
        Icon (
            imageVector = Icons.Default.MoreVert,
            contentDescription = "Menu"
        )
    }
    DropdownMenu (
        expanded = expanded,
        onDismissRequest = { expanded = false }
    ) {
        DropdownMenuItem (
            text = { Text("Configurações") },
            onClick = {
                expanded = false
                onSettingsClick ()
            }
        )
        DropdownMenuItem (
            text = { Text("Ajuda") },
            onClick = {
                expanded = false
                onHelpClick ()
            }
        )
    }
},
colors = TopAppBarDefaults.topAppBarColors (
    containerColor = MaterialTheme.colorScheme.primary,
    titleContentColor = MaterialTheme.colorScheme.onPrimary,
    actionIconContentColor = MaterialTheme.colorScheme.onPrimary
)
}
```

Passo 6: Criar Telas - HomeScreen

```
@ExperimentalMaterial3Api
@Composable
fun HomeScreen (
    onPlanetSelected : (Planet) -> Unit,
    onSettingsClick : () -> Unit,
    onHelpClick : () -> Unit
) {
    var searchQuery by remember { mutableStateOf("") }
    val filteredPlanets = remember (searchQuery) {
        planetList.filter { it.name.contains(searchQuery, ignoreCase = true) }
    }

    val recentSearches = remember { mutableStateListOf<Planet>() }

    Scaffold (
        topBar = {
            TopAppBarWithMenu (
                onSettingsClick = onSettingsClick,
                onHelpClick = onHelpClick
            )
        }
    ) { innerPadding ->
        Column(modifier = Modifier.padding(innerPadding)) {
            // Barra de pesquisa
            TextField (
                value = searchQuery,
                onValueChange = { searchQuery = it },
                label = { Text("Pesquisar") },
                modifier = Modifier
                    .fillMaxWidth()
                    .padding(8.dp)
            )
        }
    }
}
```

```
// LazyRow para buscas recentes
LazyRow (
    modifier = Modifier.padding(vertical = 8.dp, horizontal = 8.dp),
    horizontalArrangement = Arrangement.spacedBy(8.dp)
) {
    items(recentSearches) { planet ->
        Button(onClick = { onPlanetSelected (planet) }) {
            Text (planet.name)
        }
    }
}

// LazyColumn para lista de planetas filtrados
LazyColumn (
    verticalArrangement = Arrangement.spacedBy(8.dp),
    modifier = Modifier.padding(horizontal = 8.dp)
) {
    items(filteredPlanets) { planet ->
        PlanetListItem (
            planet = planet,
            onPlanetSelected = { selectedPlanet ->
                if (!recentSearches.contains(selectedPlanet)) {
                    recentSearches.add(0, selectedPlanet)
                }
                onPlanetSelected (selectedPlanet)
            },
            onFavoriteToggle = { favoritePlanet ->
                favoritePlanet.isFavorite = !favoritePlanet.isFavorite
            }
        )
    }
}
}
```

Passo 6: Criar Telas - DetailsScreen Parte 1

```
@ExperimentalMaterial3Api
@Composable
fun DetailsScreen(planet: Planet) {
    Scaffold(
        topBar = {
            TopAppBar(
                title = {
                    Text(
                        text = planet.name,
                        style = MaterialTheme.typography.titleLarge
                    )
                }
            )
        }
    ) { paddingValues ->
        Column(
            modifier = Modifier
                .fillMaxSize()
                .padding(paddingValues)
                .padding(16.dp)
        ) {
            // Imagem destacada
            Box(
                modifier = Modifier
                    .fillMaxWidth()
                    .height(240.dp),
                contentAlignment = Alignment.Center
            ) {
                Image(
                    painter = painterResource(id = planet.imageRes),
                    contentDescription = "Imagem de ${planet.name}",
                    modifier = Modifier
                        .size(200.dp)
                        .clip(CircleShape)
                )
            }
        }
    }
}
```

```
Spacer(modifier = Modifier.height(16.dp))

// Informações Gerais
Card(
    modifier = Modifier.fillMaxWidth(),
    shape = RoundedCornerShape(16.dp),
    colors = CardDefaults.cardColors(containerColor = MaterialTheme.colorScheme.surface)
) {
    Column(modifier = Modifier.padding(16.dp)) {
        Text(
            text = "Informações Gerais",
            style = MaterialTheme.typography.titleMedium.copy(fontWeight = FontWeight.Bold),
            color = MaterialTheme.colorScheme.primary
        )
        Spacer(modifier = Modifier.height(8.dp))
        Text(
            text = "Tipo: ${planet.type}",
            style = MaterialTheme.typography.bodyLarge
        )
        Text(
            text = "Galáxia: ${planet.galaxy}",
            style = MaterialTheme.typography.bodyLarge
        )
        Text(
            text = "Distância do Sol: ${planet.distanceFromSun}",
            style = MaterialTheme.typography.bodyLarge
        )
        Text(
            text = "Diâmetro: ${planet.diameter}",
            style = MaterialTheme.typography.bodyLarge
        )
    }
}
```

Passo 6: Criar Telas - DetailsScreen Parte 2

```
Spacer(modifier = Modifier.height(16.dp))

// Características
Card(
    modifier = Modifier.fillMaxWidth(),
    shape = RoundedCornerShape(16.dp),
    colors = CardDefaults.cardColors(containerColor = MaterialTheme.colorScheme.surfaceVariant)
) {
    Column(modifier = Modifier.padding(16.dp)) {
        Text(
            text = "Características",
            style = MaterialTheme.typography.titleMedium.copy(fontWeight = FontWeight.Bold),
            color = MaterialTheme.colorScheme.primary
        )
        Spacer(modifier = Modifier.height(8.dp))
        Text(
            text = planet.characteristics,
            style = MaterialTheme.typography.bodyMedium,
            lineHeight = 20.sp
        )
    }
}
}
```

Passo 6: Criar Telas - FavoritesScreen

```
@ExperimentalMaterial3Api
@Composable
fun FavoritesScreen(
    onPlanetSelected: (Planet) -> Unit,
    onFavoriteToggle: (Planet) -> Unit
) {
    Scaffold(
        topBar = {
            TopAppBar(
                title = {
                    Text(
                        text = "Favoritos",
                        style = MaterialTheme.typography.titleLarge
                    )
                }
            )
        }
    ) { innerPadding ->
        val favoritePlanets = planetList.filter { it.isFavorite }

        if (favoritePlanets.isEmpty()) {
            // Exibe o texto padrão quando não há favoritos
            Box(
                modifier = Modifier
                    .fillMaxSize()
                    .padding(innerPadding),
                contentAlignment = Alignment.Center
            ) {
```

```
Text(
            text = "Você ainda não adicionou favoritos.",
            style = MaterialTheme.typography.titleMedium,
            textAlign = TextAlign.Center,
            color = MaterialTheme.colorScheme.onSurfaceVariant
        )
    } else {
        // Exibe a lista de favoritos
        LazyColumn(
            verticalArrangement = Arrangement.spacedBy(8.dp),
            modifier = Modifier
                .padding(innerPadding)
                .padding(horizontal = 8.dp)
        ) {
            items(favoritePlanets) { planet ->
                PlanetListItem(
                    planet = planet,
                    onPlanetSelected = { onPlanetSelected(it) },
                    onFavoriteToggle = { onFavoriteToggle(it) }
                )
            }
        }
    }
}
```


Passo 7: Configurar Navegação NavGraph Parte 1

```
// Classe representando os itens da Bottom Bar
sealed class BottomBarScreen(val route: String, val icon:
@Composable () -> Unit, val label: String) {
    object Home : BottomBarScreen(
        route = "home",
        icon = {
            androidx.compose.material3.Icon(Icons.Default.Home,
            contentDescription = "Home") },
        label = "Home"
    )

    object Favorites : BottomBarScreen(
        route = "favorites",
        icon = {
            androidx.compose.material3.Icon(Icons.Default.Favorite,
            contentDescription = "Favorites") },
        label = "Favoritos"
    )
}
```

```
@ExperimentalMaterial3Api
@Composable
fun NavGraph(
    onSettingsClick: () -> Unit,
    onHelpClick: () -> Unit
) {
    val navController = rememberNavController()

    Scaffold(
        bottomBar = {
            BottomNavigationBar(navController = navController)
        }
    ) { innerPadding ->
        NavHost(
            navController = navController,
            startDestination = BottomBarScreen.Home.route,
            modifier = Modifier.padding(innerPadding)
        ) {
            // Tela Home
            composable(BottomBarScreen.Home.route) {
                HomeScreen(
                    onPlanetSelected = { planet ->
                        navController.navigate("details/${planet.name}")
                    },
                    onSettingsClick = onSettingsClick,
                    onHelpClick = onHelpClick
                )
            }
        }
    }
}
```

Passo 7: Configurar Navegação NavGraph Parte 2

```
// Tela de Favoritos
composable(BottomBarScreen.Favorites.route) {
    FavoritesScreen (
        onPlanetSelected = { planet ->
            navController.navigate("details/${planet.name}")
        },
        onFavoriteToggle = { planet ->
            planet.isFavorite =
                !planet.isFavorite
        }
    )
}
```

```
// Tela de Detalhes
composable("details/{planetName}") {
    backStackEntry ->
        val planetName =
            backStackEntry.arguments?.getString("planetName")

        val selectedPlanet =
            planetList.first { it.name == planetName }
            DetailsScreen(selectedPlanet)
        }
    }
}
```

Passo 8: Configurar MainActivity

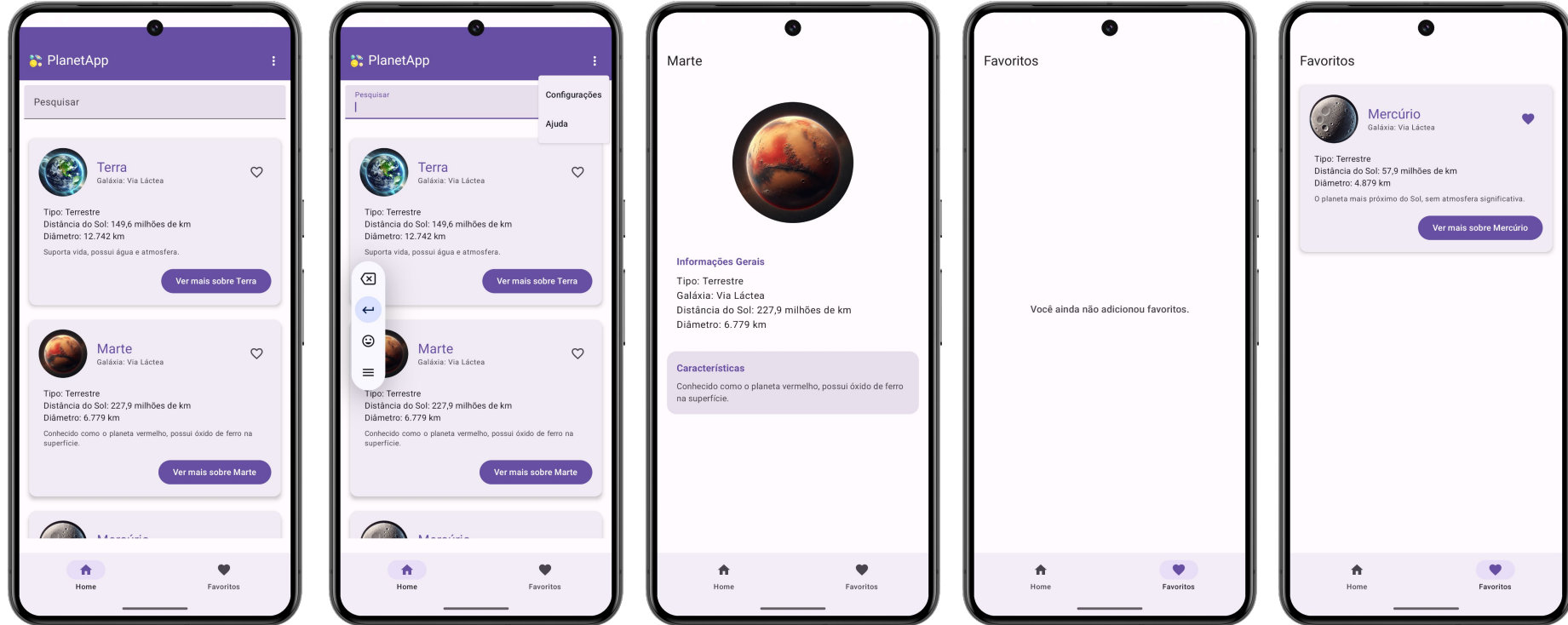
- Arquivo:
MainActivity.kt:

```
@ExperimentalMaterial3Api
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            NavGraph(
                onSettingsClick = {
                    // Ação para Configurações (pode abrir uma nova tela ou exibir
um diálogo)
                },
                onHelpClick = {
                    // Ação para Ajuda (pode abrir uma nova tela ou exibir um
diálogo)
                }
            )
        }
    }
}
```

Passo 9: Adicione um Ícone Personalizado ao Aplicativo

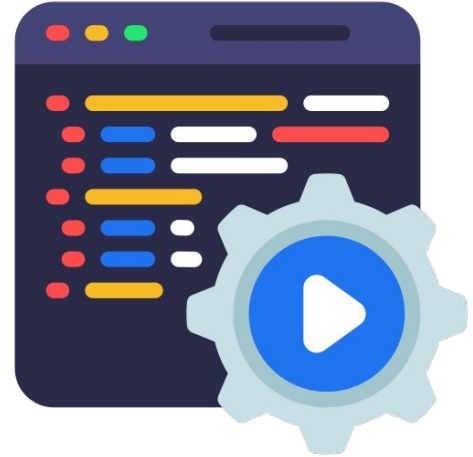
- **Crie ou obtenha o ícone do aplicativo:**
 - O ícone deve ser um arquivo de imagem em formato PNG com dimensões mínimas de 512x512 pixels e fundo transparente.
- **Adicione o ícone ao Android Studio:**
 - No Android Studio, clique com o botão direito na pasta res e selecione:
 - New > Image Asset.
 - Escolha o tipo Launcher Icons.
 - Selecione o arquivo PNG personalizado como Fonte.
 - Ajuste as opções de corte e redimensionamento conforme necessário.
 - Clique em Next e depois em Finish.
- **Verifique o Manifesto:**
 - Certifique-se de que o ícone foi configurado corretamente no arquivo AndroidManifest.xml:

Execução do PlanetApp



Referências

- <https://developer.android.com/>
- <https://developer.android.com/courses/fundamentals-training/>
- <https://flutter.dev/>



Obrigado! Dúvidas?



Universidade Federal do Ceará - *Campus* Quixadá

Prof. Francisco Victor da Silva Pinheiro
victorpinheiro@ufc.br

