

**Jetpack**



# Compose Navigation



**Centro de  
Informática**  
UFPE

# Como navegar entre telas?

Como já explicado, telas feitas com Jetpack Compose são Funções, e por isso, não podem ser chamadas em um Intent padrão.

Para navegar entre telas compose, precisamos definir uma Activity como host, e a partir dela, utilizar a biblioteca [Jetpack Navigation](#).

# Jetpack Navigation

Principais conceitos da navegação com compose:

**NavController:** É o cérebro operacional. Ele é quem controla a navegação. Criado com `rememberNavController()`

**NavHost:** É um Composable de UI que exibe as telas. Ele recebe o `NavController` para saber qual estado de navegação deve refletir.

**NavGraph:** É a definição estrutural, composto por todas as chamadas `composable()` que mapeiam uma rota (string) para uma tela `@Composable` específica.

# Exemplo:

## NavController

“Cérebro”

## NavHost

“Hospedeiro”

## NavGraph

“Mapa”

```
@Composable
fun MyAppNavController() {
    // utilização do rememberNavController para manter uma instância que sobrevive a recomposição
    val navController = rememberNavController()

    NavHost(
        navController = navController,
        startDestination = "home"
    ) {

        composable("home") {
            HomeScreen(
                onNavigateToDetail = {
                    navController.navigate("detail")
                }
            )
        }

        // Destino para a tela de detalhes
        composable("detail") {
            DetailScreen(
                onNavigateBack = {
                    navController.popBackStack()
                }
            )
        }
    }
}
```

Funções de navegação são chamadas a partir do navController

# Passos

1) Adicione as dependências no `build.gradle` (Module :app)

```
dependencies {  
    // ... outras dependências  
  
    // Para Jetpack Navigation Compose  
    implementation("androidx.navigation:navigation-compose:2.7.7")  
}
```

# Passos

## 2) Crie as telas com jetpack Compose

### Tela Home:

```
@Composable
fun HomeScreen(onNavigateToDetail: () -> Unit) {
    Column(
        modifier = Modifier.fillMaxSize(),
        verticalArrangement = Arrangement.Center,
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        Text("Bem-vindo à Tela Inicial!")
        Button(onClick = onNavigateToDetail) {
            Text("Ir para Detalhes")
        }
    }
}
```

### Tela de detalhes:

```
@Composable
fun DetailScreen(onNavigateBack: () -> Unit) {
    Column(
        modifier = Modifier.fillMaxSize(),
        verticalArrangement = Arrangement.Center,
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        Text("Você está na Tela de Detalhes!")
        Button(onClick = onNavigateBack) {
            Text("Voltar")
        }
    }
}
```

# Passos

3) Criação do  
NavController,  
NavHost, e  
NavGraph

NavGraph

```
@Composable
fun MyAppNavHost() {
    // utilização do rememberNavController para manter uma instância que sobrevive a recomposição
    val navController = rememberNavController()

    NavHost(
        navController = navController,
        startDestination = "home"
    ) {

        composable("home") {
            HomeScreen(
                onNavigateToDetail = {
                    navController.navigate("detail")
                }
            )
        }

        // Destino para a tela de detalhes
        composable("detail") {
            DetailScreen(
                onNavigateBack = {
                    navController.popBackStack()
                }
            )
        }
    }
}
```

Config do NavController com remember

Define a tela de inicio

Define a rota "home"

Lambda passada para a tela

Define a rota "detail"

# Passos

4) Inicializar  
NavController no  
MainActivity

```
class MainActivity : ComponentActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContent {  
            Surface(  
                modifier = Modifier.fillMaxSize(),  
                color = MaterialTheme.colorScheme.background  
            ) {  
                MyAppNavController()  
            }  
        }  
    }  
}
```

Chama o NavController dentro do setContent



**Quando usar Intents?**

# Intent

Embora a navegação interna seja feita com o Jetpack Navigation, as Intents ainda são cruciais para interações **fora do aplicativo**.

Exemplo: Abrir o Google

# Intent

Exemplo: Abrir o Google

```
@Composable
fun SimpleIntentScreen() {
    val context = LocalContext.current

    Column(
        modifier = Modifier.fillMaxSize(),
        verticalArrangement = Arrangement.Center,
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        Text("Clique para abrir um site.")
        Button(
            onClick = {
                val intent = Intent(Intent.ACTION_VIEW, "https://www.google.com".toUri())
                // O sistema Android encontrará e iniciará a Activity que pode lidar com esta Intent
                // (navegador padrão do usuário)
                context.startActivity(intent)
            }
        ) {
            Text("Abrir Google")
        }
    }
}
```

Obtém o contexto do Composable

Cria o intent, com a ação "ACTION\_VIEW"

Inicia o intent

# Obrigado!



**Centro de  
Informática**  
UFPE