



# Flutter Avançado - Aula 7

Exercícios 19 a ?



- Resumo Google IO 2023: Novidades no Flutter 3.10, Firebase, Dart 3 e Material Design!

[Ver mais](#)





A ideia dos flavors se resume a possibilidade de criação de vários apps em um só. Por exemplo:

- Criação de app com as opções pagas e gratuitas
- Apps para diferentes propósitos, com o mesmo código fonte
- Criar versões de testes, produção e desenvolvimento no mesmo app

A vantagem é a utilização do mesmo código fonte para múltiplos propósitos.





O primeiro passo é criar um objeto que armazenará as informações específicas de cada flavor.

```
enum AppType { desenv, test, production }

class AppConfig {
    String title;
    AppType appType;
    String folderImg;

    AppConfig({ required this.title, required this.appType, required this.folderImg });

    String get typeString {
        if (this.appType == AppType.desenv)
            return "Desenvolvimento";

        if (this.appType == AppType.test)
            return "Testes";

        if (this.appType == AppType.production)
            return "Produção";

        return "";
    }
}
```





O segundo passo é alterar o arquivo main.dart, para que não seja mais o arquivo inicial.

```
void mainCommon(AppConfig config) {  
  runApp(Provider<AppConfig>(  
    create: (context) => config,  
    child: MyApp(),  
  ));  
}
```





O terceiro passo é criar os arquivos main específicos para cada flavor

```
import 'package:flavors/config/appConfig.dart';
import 'package:flavors/main_common.dart';

void main() {
  AppConfig config = AppConfig(
    title: "Desenvolvimento",
    appType: AppType.desenv,
    folderImg:
      "image.png");

  mainCommon(config);
}
```





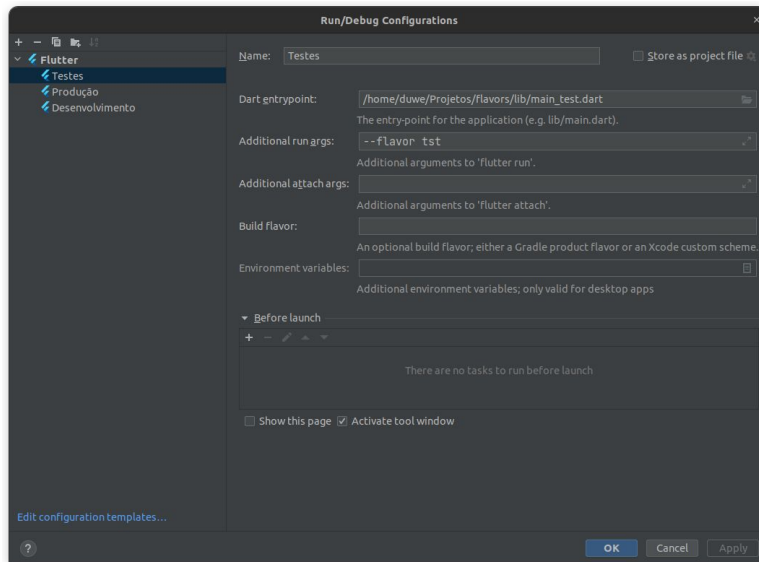
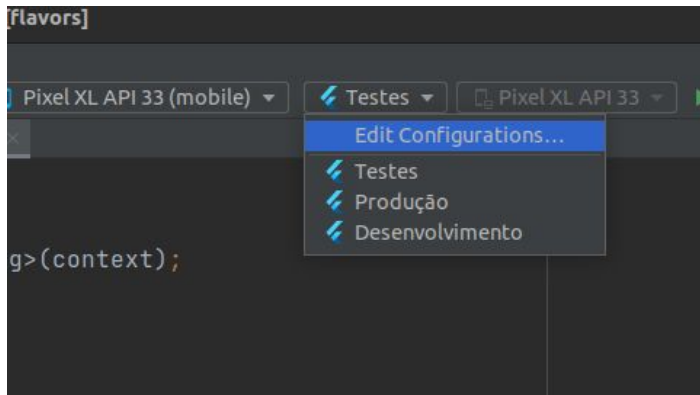
O quarto passo é adaptar o app para ler as configurações específicas de cada app.

```
AppConfig config = Provider.of<AppConfig>(context);

return Scaffold(
  appBar: AppBar(
    title: Text(config.title),
  ),
  body: ListView(
    children: [
      Container(
        height: 500,
        child: Image(
          image: NetworkImage(config.folderImg),
          fit: BoxFit.cover,
        ),
      ),
      ListTile(
        title: Text("Bem vindo!"),
        subtitle: Text(config.typeString),
      )
    ],
  ),
);
```



O quinto passo é alterar as configurações de build, no Android Studio.







O sexto passo é alterar o arquivo `android/app/build.gradle`, adicionando as configurações de build.

```
flavorDimensions "flavors"

productFlavors {
    prod {
        dimension "flavors"
        applicationIdSuffix ".prod"
        resValue "string", "app_name", "App Prod"
        versionNameSuffix ".prod"
    }

    desenv {
        dimension "flavors"
        applicationIdSuffix ".dev"
        resValue "string", "app_name", "App Dev"
        versionNameSuffix ".dev"
    }

    tst {
        dimension "flavors"
        applicationIdSuffix ".test"
        resValue "string", "app_name", "App Test"
        versionNameSuffix ".test"
    }
}
```





Após criar a rotina de cadastro de usuário, vamos criar a rotina de login.

Caso o usuário seja bem sucedido, o login será efetuado.

Utilizar uma store MobX.

A UI mockup of a login form. The form is a white card with rounded corners and a subtle shadow, centered on a light gray background. At the top of the card, the word "Login" is centered in a dark gray font. Below the title, there are two input fields. The first field is labeled "Usuário" in a small, light gray font to its left, and the second field is labeled "Senha" in a similar font to its left. Both fields have thin gray borders and are empty. In the bottom right corner of the overall image, outside the form card, there is a blue circular button with a white right-pointing arrow and a small square icon, suggesting a 'next' or 'submit' action.



### Criar três flavors: desenvolvimento, produção e testes

- Para desenvolvimento, não solicitar login e exibir a informação de debug
- Para testes e desenvolvimento, exibir no menu um cadastro de clientes, que ainda está em desenvolvimento
- Configurar os flavors no Android Studio