

Aqui está o passo 1 detalhado para configurar o projeto no Android Studio para o aplicativo de loja de venda de açaí.

Passo 1: Configuração do Projeto no Android Studio

1.1 Criando o Projeto no Android Studio

1. Abra o Android Studio:

- Certifique-se de que o Android Studio está instalado e atualizado.
- Abra o Android Studio e clique em **"New Project"**.

2. Escolha o template de atividade:

- Na janela de criação de novo projeto, você verá várias opções de templates.
- Selecione **"Empty Activity"** (Atividade Vazia) para começar do zero.
- Clique em **Next**.

3. Configuração do Projeto:

- **Nome do Aplicativo:** Defina um nome para o seu aplicativo. Exemplo: `AcaiStore`.
- **Nome do pacote:** Escolha o nome do pacote. Um exemplo seria `com.example.acaiStore`.
- **Local do Projeto:** Escolha onde deseja salvar o projeto em seu computador.
- **Linguagem:** Selecione **Kotlin** como a linguagem de programação.
- **Versão mínima do SDK:** Selecione a versão mínima de SDK. Uma boa prática é usar **API Level 21 (Android 5.0 Lollipop)** para garantir que seu aplicativo funcione na maioria dos dispositivos.
- Clique em **Finish** para criar o projeto.

O Android Studio agora irá criar um projeto com a estrutura básica de arquivos.

1.2 Configurando o Gradle para Dependências

O `build.gradle` gerencia as dependências do projeto. Vamos adicionar bibliotecas úteis para o aplicativo de venda de açaí, como gerenciamento de imagens e rede (se precisar conectar com APIs).

1. Abra o arquivo `build.gradle` (Module: app):

- No painel de projeto à esquerda, navegue até `Gradle Scripts > build.gradle (Module: app)`.

2. Adicionar Dependências:

- Certifique-se de que as seguintes dependências estão adicionadas para as funcionalidades essenciais:

groovy

 Copiar código

```
// Adiciona Retrofit para chamadas de API implementation
'com.squareup.retrofit2:retrofit:2.9.0' implementation
'com.squareup.retrofit2:converter-gson:2.9.0' // Adiciona Glide para carregar
imagens implementation 'com.github.bumptech.glide:glide:4.12.0'
```

```
annotationProcessor 'com.github.bumptech.glide:compiler:4.12.0' // Adiciona Room
para persistência de dados local implementation "androidx.room:room-runtime:2.4.3"
kapt "androidx.room:room-compiler:2.4.3"
```

3. Sincronize o projeto:

- Depois de adicionar as dependências, clique em **"Sync Now"** que aparecerá na parte superior da tela para sincronizar o Gradle.

1.3 Organizando o Projeto

A estrutura inicial do projeto deve ser algo como:

perl


 Copiar código

```
AcaiStore/ ├── app/ | ├── manifests/ | | └── AndroidManifest.xml | ├── java/ | | └──
com.example.acaiStore/ | | └── MainActivity.kt | └── res/ | ├── layout/ | | └──
activity_main.xml | ├── drawable/ | ├── mipmap/ | └── values/ | ├── colors.xml | ├──
strings.xml | └── themes.xml └── build.gradle
```

1.4 Configuração do AndroidManifest.xml

O `AndroidManifest.xml` é onde você define as permissões e as atividades principais do aplicativo. Inicialmente, ele conterá algo como:

xml

 Copiar código

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.acaiStore"> <application android:allowBackup="true"
android:icon="@mipmap/ic_launcher" android:label="@string/app_name"
android:roundIcon="@mipmap/ic_launcher_round" android:supportsRtl="true"
android:theme="@style/Theme.AcaiStore"> <activity android:name=".MainActivity"
android:exported="true"> <intent-filter> <action
android:name="android.intent.action.MAIN" /> <category
android:name="android.intent.category.LAUNCHER" /> </intent-filter> </activity>
</application> </manifest>
```

- O `MainActivity` é a atividade principal que será aberta quando o aplicativo for iniciado.
- O ícone (`@mipmap/ic_launcher`) pode ser personalizado mais tarde, de acordo com a identidade visual da loja de açaí.

1.5 Configuração de Temas e Cores

No arquivo `res/values/colors.xml` , você pode definir a paleta de cores do seu aplicativo.

xml

 Copiar código

```
<resources> <color name="purple_200">#BB86FC</color> <color
name="purple_500">#6200EE</color> <color name="purple_700">#3700B3</color> <color
```

```
name="teal_200">#03DAC5</color> <color name="teal_700">#018786</color> <color
name="black">#000000</color> <color name="white">#FFFFFF</color> <!-- Cores
personalizadas --> <color name="primaryColor">#FFC107</color> <!-- Amarelo para a
temática de açaí --> <color name="accentColor">#4CAF50</color> <!-- Verde para
destaque --> </resources>
```

Essas cores podem ser usadas posteriormente nos layouts para criar um visual atrativo para a loja.

1.6 Executando o Aplicativo

Agora que a configuração básica do projeto está feita:

1. **Execute o aplicativo** clicando no botão de "Play" na barra de ferramentas superior.
2. O emulador ou dispositivo conectado exibirá a tela em branco com o nome do aplicativo definido.

Próximos Passos

Agora que o projeto foi configurado e está pronto para rodar, você pode seguir para a próxima etapa: **Criação da Interface de Usuário (Frontend com XML)**, onde iremos detalhar a criação dos layouts e a estrutura da loja visualmente.