

Aqui está o passo 1 detalhado com os códigos para configurar o projeto inicial de um aplicativo de barbearia no Android Studio:

## 1. Configuração Inicial

### 1.1 Instalar o Android Studio

Se ainda não tiver o Android Studio instalado:

1. Acesse <https://developer.android.com/studio> e faça o download da versão mais recente do Android Studio.
2. Siga as instruções de instalação para o seu sistema operacional.

### 1.2 Criar um Novo Projeto

1. Abra o Android Studio e clique em "New Project".
2. Selecione "Empty Activity" como o tipo de projeto.
3. Defina o nome do aplicativo como `BarbeariaApp`.
4. Defina o pacote como `com.exemplo.barbearia`.
5. Selecione **Kotlin** como a linguagem de programação.
6. Defina a **API mínima** como a versão 21 (Android 5.0 Lollipop) para garantir compatibilidade com a maioria dos dispositivos.
7. Clique em **Finish** para gerar o projeto.

### 1.3 Estrutura do Projeto Gerado


Quando o projeto for criado, o Android Studio gerará automaticamente uma estrutura de diretórios. Aqui estão os principais arquivos:

- **MainActivity.kt**: O arquivo principal de atividade.
- **activity\_main.xml**: O layout associado à MainActivity.
- **build.gradle (Module: app)**: Arquivo de configuração onde serão adicionadas dependências.

### 1.4 Configuração do `build.gradle`

Vamos configurar o arquivo `build.gradle` para adicionar as dependências necessárias ao projeto. Abra o arquivo `build.gradle (Module: app)` e adicione as seguintes dependências:

kotlin

 Copiar código

```
plugins { id 'com.android.application' id 'org.jetbrains.kotlin.android' } android {
compileSdk 34 defaultConfig { applicationId "com.exemplo.barbearia" minSdk 21
targetSdk 34 versionCode 1 versionName "1.0" testInstrumentationRunner
"androidx.test.runner.AndroidJUnitRunner" } buildTypes { release { minifyEnabled false
proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-
rules.pro' } } compileOptions { sourceCompatibility JavaVersion.VERSION_17
targetCompatibility JavaVersion.VERSION_17 } kotlinOptions { jvmTarget = '17' } }
dependencies { implementation "org.jetbrains.kotlin:kotlin-stdlib:1.8.21"
implementation 'androidx.core:core-ktx:1.9.0' implementation
```

```
'androidx.appcompat:appcompat:1.5.0' implementation
'com.google.android.material:material:1.6.1' implementation
'androidx.constraintlayout:constraintlayout:2.1.4' testImplementation
'junit:junit:4.13.2' androidTestImplementation 'androidx.test.ext:junit:1.1.3'
androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0' }
```


## 1.5 Atualizar o projeto

1. Depois de adicionar essas dependências, clique no botão **Sync Now** no Android Studio para sincronizar as dependências com o projeto.
2. Verifique se não há erros durante a sincronização.

## 1.6 Configuração do arquivo `AndroidManifest.xml`

No arquivo `AndroidManifest.xml`, podemos ajustar a configuração do nome do aplicativo e a activity principal. O código gerado por padrão deve parecer com o seguinte:

xml


 Copiar código

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

## 1.7 Código Inicial da MainActivity (Kotlin)

Agora, vamos ajustar o arquivo `MainActivity.kt`, que será o ponto de entrada do aplicativo.

kotlin

 Copiar código

```
package com.exemplo.barbearia import androidx.appcompat.app.AppCompatActivity import
android.os.Bundle class MainActivity : AppCompatActivity() { override fun
onCreate(savedInstanceState: Bundle?) { super.onCreate(savedInstanceState)
setContentView(R.layout.activity_main) // Exemplo: Configuração inicial da
MainActivity } }
```

## 1.8 Layout Inicial (XML)

No arquivo `res/layout/activity_main.xml`, você pode definir um layout inicial simples, como um título e um botão para prosseguir para a próxima tela. Aqui está um exemplo básico:

xml

 Copiar código

```

<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
android:layout_height="match_parent" tools:context=".MainActivity"> <TextView
android:id="@+id/titleTextView" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:text="Bem-vindo à Barbearia!"
android:textSize="24sp" android:layout_marginTop="100dp"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintEnd_toEndOf="parent"/> <Button android:id="@+id/bookButton"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:text="Agendar Serviço" android:layout_marginTop="50dp"
app:layout_constraintTop_toBottomOf="@id/titleTextView"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintEnd_toEndOf="parent"/>
</androidx.constraintlayout.widget.ConstraintLayout>

```

Neste layout, há um `TextView` centralizado com uma mensagem de boas-vindas e um `Button` abaixo dele para iniciar o agendamento.

## 1.9 Executar o Projeto

1. Conecte um dispositivo físico Android ou utilize um emulador.
2. No Android Studio, clique no ícone **Run** (ou pressione Shift+F10) para compilar e executar o aplicativo no dispositivo ou emulador.

Quando o aplicativo for iniciado, você verá a mensagem "Bem-vindo à Barbearia!" e um botão "Agendar Serviço".

## 1.10 Próximos Passos

Agora que o projeto está configurado, você pode começar a expandir a lógica e a interface do aplicativo, adicionando mais funcionalidades e ajustando o design conforme a necessidade.

Com esses passos, seu projeto básico estará pronto para começar o desenvolvimento das funcionalidades específicas do aplicativo de barbearia.