

## Entendendo a activity no Android

A activity é o ponto de entrada de um aplicativo. No caso de aplicativos Android, não teremos um método main, então a activity será utilizada para inicializá-lo.

Vamos considerar a seguinte situação. Temos o aplicativo Orgs e dentro dele temos a activity, que terá dois componentes principais que podemos refletir no momento que criamos uma activity. Um deles é a View, uma entidade e referência que representará o layout apresentado para a pessoa usuária. Se você quer editar o layout ou apresentar algo visual, você trabalhará com as Views, que são as visualizações.

Em seguida, temos a parte lógica, onde mexeremos com o código Java ou Kotlin, neste caso Kotlin, para então modificar e conversar com a View. A View é acessível por meio de código-fonte, e conseguimos manipular as informações para apresentar o resultado visual esperado.

Sendo assim, perceba que na activity, trabalharemos muito com esse conceito de utilizar a View para fazer a parte visual, mas para fazer a parte funcional, trabalharemos com a parte lógica que envolve o código-fonte, seja Java ou Kotlin. Sabendo disso, já estamos preparados para criar a primeira activity.

### Criando uma activity

Precisamos criar um arquivo, um código que represente uma activity. No Android, como mencionado anteriormente, trabalhamos bastante com orientação a objetos. Então, uma activity será uma classe.

Então, clicamos com o botão direito no pacote e depois em "Kotlin Class/File". Na janela que abre, nomeamos de MainActivity. Agora temos uma classe que queremos representar por uma activity.

Porém, isso não é suficiente para que o sistema do Android entenda que isso é uma activity. A Activity é criada a partir do sistema do Android. Nós não criamos uma activity, apenas implementamos representando uma classe, quem cria é o próprio Android como sistema operacional.

Considerando então, que ele não consegue entender que isso é uma activity, como faremos? Basicamente, identificamos a partir de uma herança. Trabalhamos com o conceito de herança, que vem de Programação Orientada a Objetos. É a partir de uma herança, de uma extensão da classe activity, que vem do pacote android.app, do sistema operacional do Android. Ao pressionar "Enter", temos a seguinte estrutura:

```
import android.app.Activity  
  
class MainActivity: Activity ()
```

Precisamos deixar o construtor padrão, ou seja, vazio. Agora nossa classe é uma activity do Android. Porém, mesmo tendo essa configuração, **o sistema não reconhecerá ainda como uma activity**. Isso porque **exige uma configuração a mais** relacionada ao pacote chamado de manifest, que dá acesso a esse arquivo chamado de **AndroidManifest.xml**

Nesse arquivo .xml, teremos toda a configuração do aplicativo, desde o nome, ícone que será utilizado, qualquer tipo de configuração de tema, podemos adicionar dentro desse arquivo de android.manifest.

Se você quiser saber mais informações sobre as capacidades desse arquivo, disponibilizamos um material complementar no Para Saber Mais.

Analisaremos a página que fala sobre a visão geral desse arquivo de manifesto do aplicativo Android. Nele, você conseguirá fazer todas as configurações que mencionamos. Além disso, tem um grau de importância muito grande, porque é a partir dele que conseguimos descrever as informações do aplicativo, para ser gerado, instalado em um dispositivo, até mesmo para a Google Play reconhecer o que tem no aplicativo.

São muitas coisas, como, por exemplo, permissões, falar quais são as activities e assim por diante. Não é o momento para falarmos a respeito de todas as possibilidades, por isso recomendamos a leitura da documentação.

Agora, voltamos para o Android Studio. Para registrar nossa activity precisamos fazer uma configuração na tag <application>, que representa nossa aplicação.

Agora, dentro dessa tag, podemos colocar mais configurações. Então, escrevemos:

```
<activity android:name="">,
```

repare que ela já vem com essa propriedade chamada de name, que representa aqui o nome que vamos dar para a nossa activity. Após o sinal de igual = escrevemos .MainActivity.

```
<activity android:name=".MainActivity"/>
```

Nesse momento, conseguimos declarar uma activity. Só isso já é o suficiente para definirmos que temos essa activity no aplicativo. Porém, ainda tem um detalhe. Apenas configurar dessa forma, não indica qual será a activity principal do nosso aplicativo, nem que no momento que executarmos o aplicativo, lançaremos a activity para a pessoa usuária visualizar a tela. Então, precisamos fazer essas configurações.

Fechamos a tag <activity>. Dentro dela, teremos uma tag chamada intent-filter, que seria um filtro de intenção. No momento que queremos fazer essa configuração, que é deixar a nossa activity como principal, e no momento que executar ela ser lançada, precisamos colocar tanto esse action, para falar que ela é a principal, como também temos que colocar o category, que seria a categoria para falar que ela será lançada no momento que executarmos o aplicativo.

```
<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>
```

Toda vez que tivermos que colocar uma activity, que será a principal e precisa ser lançada, temos que colocar o intent-filter. Se ela não tivesse essa característica, fosse executada em algum outro momento, seria preciso colocar da outra maneira que aprendemos, abrindo a tag, definindo o nome e fechando.

**Isso já é o suficiente para executarmos nosso aplicativo.** Porém, ao clicar no botão, na barra de menu superior, identificado pelo ícone de um triângulo verde, repare que não encontra nenhum dispositivo.

Isso acontece, pois no momento que executamos um aplicativo Android, compilamos nosso código, desde o código-fonte, é gerado um arquivo chamado de APK, que seria uma extensão disponível e compatível para um aplicativo Android. Esse APK não pode ser rodado pelo Windows, Linux e Mac, tem que ser rodado no sistema operacional Android.

Então, precisamos de um dispositivo Android para que isso seja possível executar o código. Dependendo da versão do Android Studio que você instalou, pode ser que ele já tenha um dispositivo aqui disponível para você.

Aula Desenvolvimento Móvel – 19/02/2025

Apresentação do slide:

<https://professorjorgelescura.blogspot.com/p/serra-douradadesenvolvimento-movel.html>

Pós aula Desenvolvimento Móvel – 19/02/2025

Criar o projeto “Prototipo” dentro do Android Studio