

# Programação para dispositivos móveis

## Aula 2 - Activity e Intent

### INTRODUÇÃO

---



Esta aula visa apresentar a classe Activity, a pilha de execução de uma aplicação Android, os estados de uma Activity e seu ciclo de vida. Também apresenta a Classe Intent e Intent Filters, bem como desenvolvimento de um pequeno aplicativo.

### OBJETIVOS

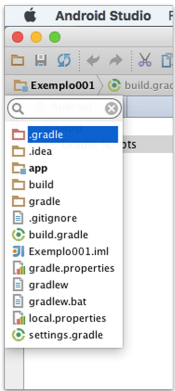
---



Identificar os principais conceitos referentes à Activity Android, bem como aplicá-la no desenvolvimento de uma aplicação simples;

Identificar os principais conceitos referentes à Intent e Intent Filters e seu uso no desenvolvimento de uma aplicação simples.

# ESTRUTURA DE UM PROJETO ANDROID



Fonte da Imagem:

O Android Studio pode abrir um projeto de cada vez.

Conforme é demonstrado na figura, cada projeto pode possuir um ou mais módulos.

Observe, na tabela, os arquivos do diretório raiz do projeto:

Pasta	Descrição
app	Módulo app do projeto. Este módulo é padrão.
build.gradle	Arquivo de configuração do Gradle. Vale para todo o projeto, incluindo todos os módulos.
gradle.properties	Arquivo de propriedades para customizar o build do Gradle.
gradlew.bat	Script que executa o build do Gradle para compilar o projeto.
local.properties	Arquivo com as configurações locais do projeto.
settings.gradle	Arquivo de configuração do gradle que indica quais módulos devem ser compilados.

Veja, também, os arquivos do módulo app:

Pasta	Descrição
build	Local onde ficam os arquivos compilados do módulo.
R.java	A classe R que é gerada automaticamente ao se compilar o projeto. Esta permite que a aplicação acesse qualquer recurso.
libs	Onde devem ser inseridos os arquivos .jars a serem compilados com o projeto.
src/main/java	Pasta que contém os arquivos Java.
src/main/res	Pasta que contém os recursos da aplicação.
res/drawable	Pasta que contém as imagens da aplicação.
res/mipmap	Pasta com o ícone da aplicação.
res/layout	Pasta que contém os arquivos XML de layouts para construir as telas da aplicação.
res/menu	Pasta que contém os arquivos XML que criam os menus da aplicação, que são os botões com ações na action bar.
res/values	Pasta que contém os arquivos XML utilizados para a internacionalização, configuração de temas e outras configurações.

Agora, vamos conhecer detalhes sobre alguns tipos de arquivo:

## Arquivo AndroidManifest.xml

Considerado um dos principais arquivos de sua aplicação. É nele que são descritas informações essenciais à execução de seu projeto como, por exemplo:

- Nome do pacote utilizado;
- Nome das Activities;
- Permissões que o aplicativo possui;
- Versão mínima da API Android.

## Arquivo activity\_main.xml

Esse arquivo, por default, possui este nome. Porém, podemos escolher um nome mais adequado.

Nesse arquivo são definidas as configurações para criação do layout da tela.

## Arquivo strings.xml

Esse arquivo centraliza as mensagens de seu aplicativo. Facilita muito, inclusive, a internacionalização do aplicativo.

## Classe R

Esse arquivo possui as referências para acessar os recursos de seu projeto.

É gerada automaticamente pelo compilador. É recomendável que essa classe não seja alterada manualmente.

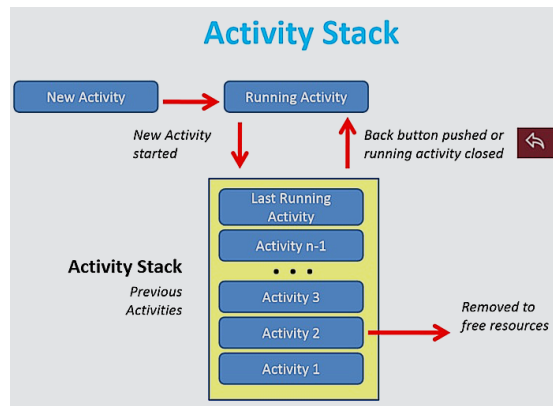
## ACTIVITY

Muito similar ao JFrame do J2SE, a Activity é responsável por construir uma tela em Android, bem como tratar os eventos gerados por ela. Toda aplicação Android deve implementar ao menos um Activity, podendo chamar outras Activities.

O Android é responsável por gerenciar o ciclo de vida dos Activities. Para tanto, faz uso do conceito de pilha, chamada de "Activity Stacks" (pilha de atividades). Toda Activity ao ser executada é inserida no topo dessa pilha. A Activity anterior é parada e move-se para baixo da pilha.

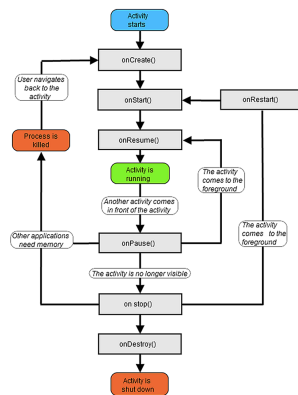
O Android pode até mesmo encerrar Activities, se precisar de recursos. Neste caso, ele verifica a pilha de atividade para determinar a prioridade das atividades e quais podem ser fechadas.

Veja na imagem o exemplo de uma pilha de atividades:



Fonte: //4.bp.blogspot.com/-HSwGf-pS91g/UZx5IRzamSI/AAAAAAAAAEs/y6wK1CnLUw0/s1600/ACTIVITYSTA.png

## CICLO DE VIDA ACTIVITY



Observamos, na imagem, os principais métodos do ciclo de vida Activity. São eles:

## onCreate()

É a primeira função executada quando a Activity é criada. Tem por responsabilidade carregar os layouts XML, inicializar os objetos, variáveis e outras operações de inicialização. É importante lembrar que é executada somente uma vez.

## onStart()

x

É executado antes da Activity ficar visível na tela do dispositivo, podendo ser chamado após os métodos **onCreate()** ou **onRestart()**.

## onResume()

x

Representa o estado que a Activity está executando. É chamado logo após o evento onStart.

## onRestart()

x

É chamado imediatamente após ao método onStart(), quando uma Activity que estava parada volta ao foco.

## onPause()

x

É chamado sempre que a tela da Activity fechar. Isto ocorre quando uma outra Activity (da sua aplicação ou não) ganhar o foco.

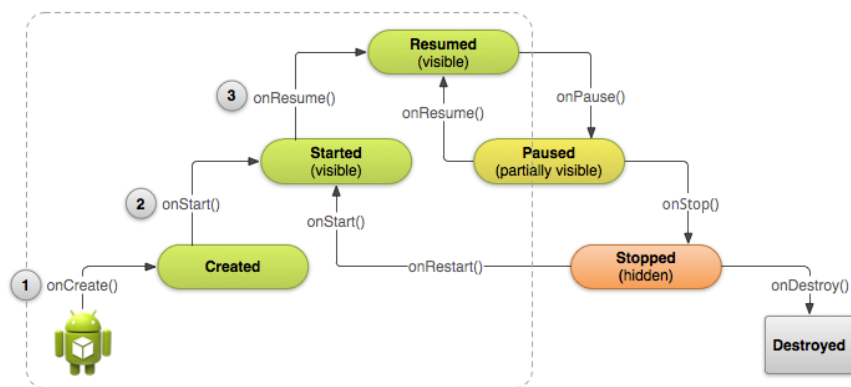
## onStop()

É chamado após o método **onPause()**, quando a Activity não está mais visível e está sendo encerrada.

## onDestroy()

É chamado antes da Activity ser destruída e logo após será liberada a memória.

Você pode observar no diagrama referente ao ciclo de vida da Activity os 3 níveis:



Entire lifetime	Visible lifetime	Foreground lifetime
Tempo de vida completo	Tempo de vida visível	Tempo de vida no topo da pilha
Ocorre desde a primeira chamada ao método <b>onCreate()</b> até a chamada do método <b>onDestroy()</b> , os quais são executados apenas uma única vez durante o ciclo de vida da Activity.	Ocorre entre uma chamada do método <b>onStart()</b> e a chamada correspondente do método <b>onStop()</b> . A Activity pode estar no topo da pilha (visível para o usuário) ou em segundo plano.	Ocorre entre uma chamada do método <b>onResume()</b> e a chamada correspondente do método <b>onPause()</b> . A Activity está no topo da pilha e interagindo com o usuário.

## Exemplo

, Antes de seguir em frente, clique [aqui \(galeria/aula2/docs/a02\\_t05.pdf\)](#) e veja um exemplo.

## APLICANDO INTENT NA PRÁTICA

Embora você ainda não conheça todos os componentes e a estrutura do Android Studio, vamos implementar um pequeno exemplo demonstrando que é muito fácil desenvolver uma aplicação com o Android Studio.

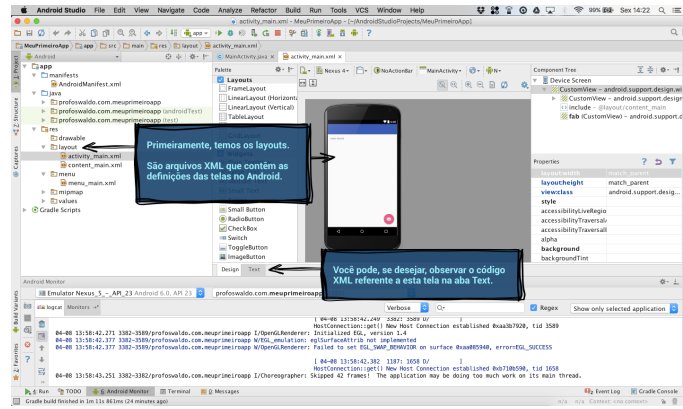
Execute o Android Studio e assista ao vídeo:



**Antes vamos fazer uma análise mais detalhada de nosso app, lembrando que ainda discutiremos cada componente com mais profundidade nas próximas aulas.**

Veja a tela abaixo:





Vinculado a essa tela, temos uma Activity, no nosso caso MainActivity.

Trata-se de um arquivo Java. Nele, entre outras atividades, são definidos os tratamentos dos eventos gerados em nosso dispositivo.

Veja a seguir:

Viu como foi fácil?

## CLASSE INTENT

Como estudamos, cada Activity corresponde a uma tela de nossa aplicação no Android. Muito embora seja muito mais do que simplesmente uma tela, não é mesmo?

# E se precisássemos trabalhar com mais de uma tela?

Devido à limitação de tamanho de nossos dispositivos, é muito comum distribuímos componentes por várias telas visando facilitar o uso da aplicação.

## O que fazer então?

Entre os vários componentes fundamentais, na programação de aplicativos Android (como Activities, Services e Broadcast Receivers), a **Intent** possibilita realizar a ligação, em tempo de execução, de componentes separados (por exemplo, chamar Activities diferentes).

Em outras palavras, podemos dizer que uma Intent nada mais é do que a intenção da aplicação em realizar uma determinada tarefa.

Trata-se de uma intenção, ou seja, não necessariamente será executada, pois depende da permissão do dispositivo.

## Mas isso como é possível?

A Intent envia ao sistema operacional o equivalente a uma mensagem (broadcast). Este receberá a chamada e, dependendo do conteúdo, tomará as **providências necessárias (glossário)**.

Uma Intent é basicamente um conjunto de dados que possui informações de interesse para os componentes que a recebem e também para o Android.

Deve conter:

## Nome do componente

Representa o nome do componente que tratará a Intent. Deve ser o nome completo da classe alvo do componente. Este nome deve ser uma combinação do pacote do componente e o pacote definido no manifesto.

## Ação

É o que você deseja executar propriamente dito. Consiste em uma String com o nome da ação a ser executada. No caso de um broadcast, a ação que aconteceu e que está sendo reportada.

## Dados

URI e MIME type dos dados adicionados à Intent. Diferentes ações são paradas com diferentes tipos de dados.  
Exemplo: Ao enviar foto para que outro aplicativo trate, precisamos definir a URI e o MIME type.

## Categoria

String com informações adicionais sobre o tipo do componente que deve tratar a Intent. Qualquer número de categorias pode ser adicionado em uma Intent.

## Extras

Pares chave-valor para informações adicionais que devem ser entregues para o componente que tratará a Intent.

## Flags

Funcionam como instruções para o sistema Android, de como lançar uma Activity ou como tratá-la depois de lançada. Todas as flags são definidas na classe Intent.

Podemos definir ações específicas de nossa aplicação, biblioteca ou ações pré-definidas conforme a tabela abaixo:

Constante	Componente alvo	Ação
ACTION_CALL	activity	Inicia uma chamada de telefone.
ACTION_EDIT	activity	Mostra dados para o usuário editar.
ACTION_MAIN	activity	Inicia a atividade que está marcada como inicial em modo vazio e com nenhum retorno.
ACTION_SYNC	activity	Sincroniza dados no servidor a partir do dispositivo móvel.
ACTION_BATTERY_LOW	broadcast receiver	Mostra um aviso que a bateria está baixa.
ACTION_HEADSET_PLUG	broadcast receiver	Mostra que um fone de ouvido foi plugado no dispositivo ou desplugado.
ACTION_SCREEN_ON	broadcast receiver	A tela foi ligada.
ACTION_TIMEZONE_CHANGED	broadcast receiver	As configurações da zona de tempo foram mudadas.

Segue abaixo as principais constantes pré-definidas:

Constante	Significado
CATEGORY_BROWSABLE	A atividade alvo pode ser chamada de maneira segura pelo navegador para mostrar dados referenciados por um link (imagem ou e-mail etc.).
CATEGORY_GADGET	A atividade pode ser embarcada dentro de outra atividade que hospede gadgets.
CATEGORY_HOME	A atividade mostra a tela home. A primeira tela que o usuário vê quando o dispositivo é ligado ou quando a tela HOME é pressionada.
CATEGORY_LAUNCHER	A atividade pode ser a atividade inicial ou uma tarefa e é listada no tipo da aplicação launcher.
CATEGORY_PREFERENCE	A atividade alvo é o painel de preferências.

## TIPOS DE INTENTS

Como estudamos, cada Activity corresponde a uma tela de nossa aplicação no Android. Muito embora seja muito mais do que simplesmente uma tela, não é mesmo?



## INTENT FILTER

Basicamente, um Intent Filter informa ao sistema quais Intents um certo componente pode tratar. Um componente pode ter uma ou mais Intent Filters. Veja o exemplo a seguir:

```

<activity android:name="MainActivity">
    <!-- This activity is the main entry, should appear in app launcher -->
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<activity android:name="ShareActivity">
    <!-- This activity handles "SEND" actions with text data -->
    <intent-filter>
        <action android:name="android.intent.action.SEND" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:mimeType="text/plain" />
    </intent-filter>
    <!-- This activity also handles "SEND" and "SEND_MULTIPLE" with media data -->
    <intent-filter>
        <action android:name="android.intent.action.SEND" />
        <action android:name="android.intent.action.SEND_MULTIPLE" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:mimeType="application/vnd.google.panorama360.jpg" />
        <data android:mimeType="image/*" />
        <data android:mimeType="video/*" />
    </intent-filter>
</activity>

```

## Atenção

, Com relação ao tipo implícito, as Intents serão entregues se um dos filtros atender aos critérios da Intent.

Já no caso da explícita, será entregue diretamente ao componente designado, não importando o filtro, pois nem chega a consultá-lo.

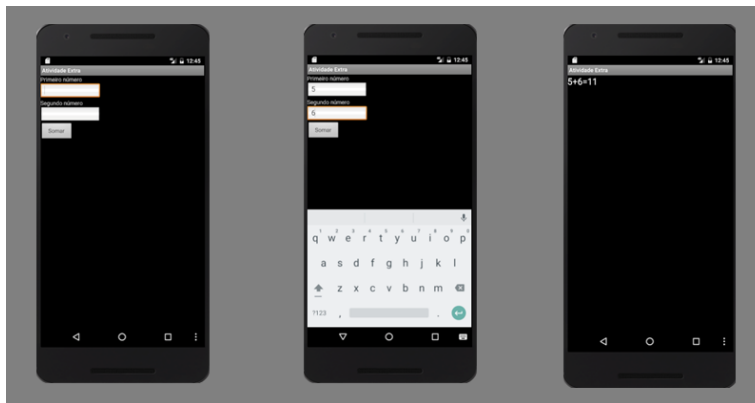
## APLICANDO INTENT NA PRÁTICA

Vamos criar um projeto demonstrando a Intent em ação? Aperte o play e vamos lá!

## ATIVIDADE

Para fixarmos o conteúdo é muito importante praticarmos. Vamos lá!

Observe as telas abaixo:



Desenvolva um aplicativo que, conforme demonstrado nas telas acima, efetue a soma de 2 valores.

Este deverá ser composto de duas Activities. Sendo uma principal, que efetuará a leitura dos dados e a soma, e uma resultado que deverá exibir o resultado desta soma.

Depois de terminar a atividade, veja o gabarito e compare.

## PDF

, Clique [aqui \(galeria/aula2/docs/a02\\_t10.pdf\)](#) para ver o gabarito.

# Glossário

## PROVIDÊNCIAS NECESSÁRIAS

---

Como exemplo, podemos citar:

- Iniciar uma nova Activity;
- Iniciar Bluetooth do aparelho;
- Ligar o GPS (Global Positioning System);
- Efetuar uma ligação telefônica;
- Abrir o programa de envio de SMS (Short Message Service);
- Chamar o navegador Web;
- Enviar mensagens para o Sistema Operacional.