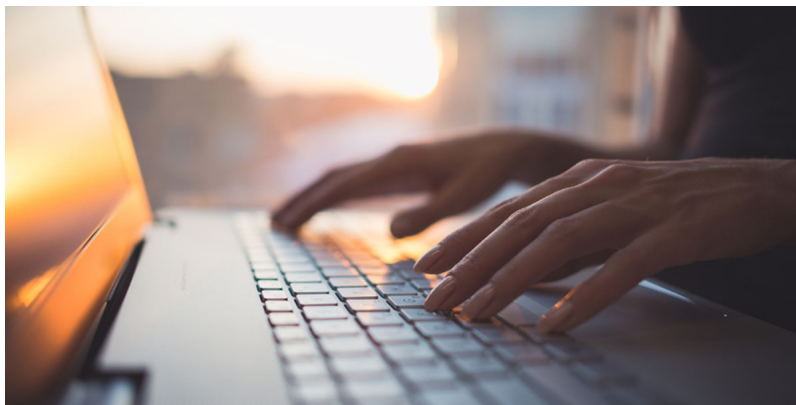


Programação para dispositivos móveis

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.

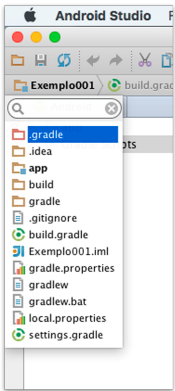
OBJETIVOS



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:

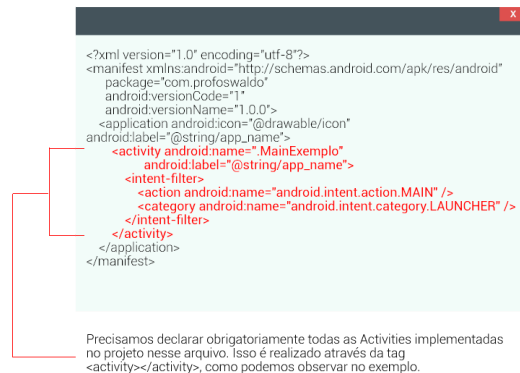
Pasta	Descrição
build	Local onde ficam os arquivos compilados do módulo.
R.java	A classe R que é gerada automaticamente ao se compilar o projeto. Esta permite que a aplicação acesse qualquer recurso.
libs	Onde devem ser inseridos os arquivos .jars a serem compilados com o projeto.
src/main/java	Pasta 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:

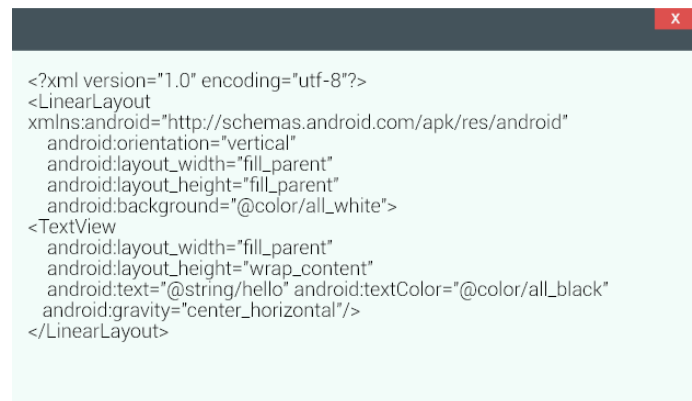
- Nome do pacote utilizado;
- Nome das Activities;
- Permissões que o aplicativo possui;
- Versão mínima da API Android.



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.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="hello">Exemplo</string>
  <string name="app_name">MainActivity</string>
  <color name="branco">#FFFFFF</color>
  <color name="preto">#000000</color>
</resources>
```

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.

```
public final class R {
    public static final class attr { }
    public static final class drawable {
        public static final int icon=0x7f020000;
    }
    public static final class id {
        public static final int b1=0x7f050001;
        public static final int text1=0x7f050000;
    }
    public static final class layout {
        public static final int main=0x7f030000;
    }
    public static final class string {
        public static final int app_name=0x7f040001;
        public static final int app_name1=0x7f040003;

        public static final int hello=0x7f040000;
        public static final int hello1=0x7f040002;
    }
}
```

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: <http://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()

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

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



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:

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

Segue abaixo as principais constantes pré-definidas:

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:



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.



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.