01

Um novo projeto

Transcrição

Para desenvolver um aplicativo para **Android** no ***Android Studio*** é necessário criar um novo projeto. Então, clica-se em "Start a new Android Studio project".

Devemos nomear a aplicação, por exemplo, queremos criar uma agenda de alunos então chamaremos simplesmente de Agenda. No campo domínio da empresa inserimos a convenção do **Java**, a alura.com.br. E, logo abaixo, é mostrado onde o projeto é gerado. Após inserir as informações clica-se em "Next".

Em seguida, será pedido qual a versão mínima que a aplicação vai rodar no **Android**. É sugerido que seja o *Android 4.0* e seu alcance é de 94% do mercado.

A seguir será pedido o que se quer colocar no projeto. Deve ser selecionada a opção "Empty Activity". Após esse procedimento, aparecerá um campo pedindo o nome da *Activity* e troca-se o *Main* pelo que se deseja criar, no caso, uma lista de alunos, assim: "ListaAlunosActivity". Os outros valores não precisam ser preenchidos nesse momento. Clica-se em "Finish". Demora um pouco, mas o projeto é gerado. Então, seleciona-se "Open".

> Na versão anterior do Android Studio, o usuário costumava selecionar a opção "Blank Activity". Com o lançamento da nova versão, a recomendação é que seja utilizada a opção "Empty Activity".

Após o término da renderização devemos rodar a aplicação para visualizá-la. Para executar uma aplicação vamos no Menu e selecionamos: "Run > Run app". Clicando nessa opção ele termina de fazer o *build* do projeto, o que tarda um pouco. Após aguardar um momento será aberta uma nova janela e nela escolhemos a opção de onde desejamos rodar a aplicação. No caso, no emulador Alura. Clicamos em "OK". Na tela aparece uma imagem simulando uma tela de celular onde, inclusive, podemos aumentar volume, som e etc.

> ***Dica:*** Uma vez iniciado o emulador é melhor deixá-lo aberto e minimizado, pois, uma vez iniciado, ele não tarda tanto .

Finalizado o *build* é preciso destravar o celular e isso se faz arrastando o cadeado para cima. Aparecerá a tela da aplicação onde estará escrito: Agenda e *Hello world!*. Mas, esse texto será modificado!

Após testar a aplicação e vê-la rodando voltamos para o **Android Studio** e verificamos o que foi gerado. Fechamos os arquivos já abertos e ficamos apenas com o projeto. No lado esquerdo da tela está o *app* que ele criou e abrindo esse projeto verificamos três pastas: "java", "res" e "manifests". Vamos em "Java > br.alura.agenda > ListaAlunosActivity" e clicando duas vezes na `ListaAlunosActivity` aparecerá a Classe da aplicação e os códigos já gerados.

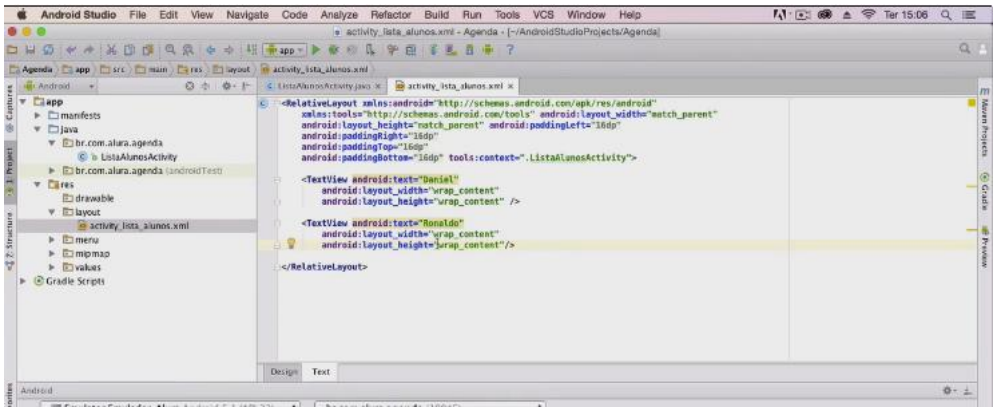
> Na versão anterior do Android Studio são geradas juntamente com a função onCreate, as funções onCreateOptionsMenu() e onOptionsItemSelected() que no exercício, são deletadas. Porém, na versão mais nova do Android Studio essas funções já não são geradas.

O primeiro passo quando deseja-se desenvolver uma aplicação para Desktop, Windows, Mac e outros é ter uma janela para poder mostrar ao usuário, ou seja, um *container*. No **Android** não é diferente, é necessário uma tela, a qual chamamos de *Activity*. É importante lembrar que toda vez que falamos em *Activity* nos referimos a tela da aplicação.

Ao criar uma nova classe *Java*, faremos o seguinte:

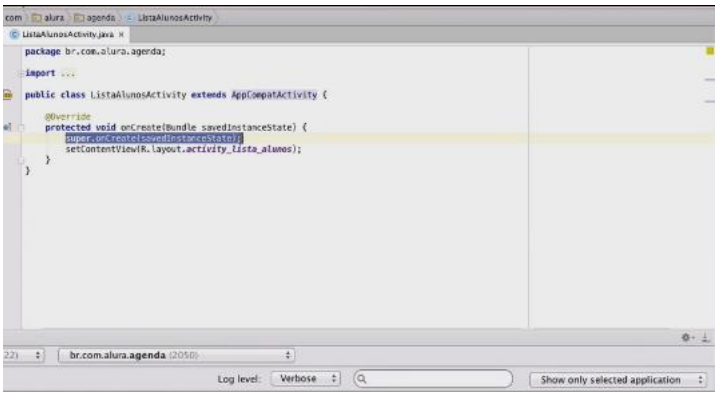
```
public class ListaAlunosActivity extends AppCompactActivity {

//....
}
```

Assim, nós teríamos apenas uma classe normal, mas como queremos dizer que nossa classe representa uma *Activity*, utilizamos uma herança que é feita através do comando `extends` .



Esse `AppCompatActivity` é uma classe que já existe no **Android** e usando ela estendemos o seu comportamento para o **Android**. No código podemos verificar que existem três métodos, mas o importante é apenas o método `onCreate` . Os outros dois podem ser apagados e nada será perdido ao fazer isso. Primeiro, `onCreate` é um método que está sendo sobrescrito da classe `AppCompactActivity` e o **Android** chama ele quando cria a tela do celular.

A primeira coisa que se faz na `onCreate` é chamar a `onCreate` que já existia na classe super.



Se existe algum comportamento específico ele será recuperado ao se chamar o super. Caso contrário, não chamando o super, estaremos sobrescrevendo todo o comportamento e jogando fora o que já existe. Por isso, é importante lembrar de recuperá-lo antes de inserir o comportamento da nossa tela. O que resta na Activity é apenas uma linha:

```
setContentView(R.layout.activity_lista_alunos);
```

O `setContentView` é de onde vem o conteúdo da tela e isso separa dois itens: a parte visual e o comportamento. Na nossa *Activity* coloca-se o comportamento, ou seja, como queremos que o usuário interaja nela. A parte visual, o que está sendo exibida e disposta na tela, localiza-se em um arquivo diferente, o *Layout*.

O `R.layout.activity_lista_alunos` é um atalho no **Android** para acessar um arquivo em determinada pasta.

Observando a estrutura do projeto vemos três pastas: "manifests", "java" e "res". Ao abrirmos a pasta res vemos que nela existem várias outras pastas e dentre elas a *Layout*. Abrindo a pasta *Layout* vemos um arquivo chamado `activity_lista_alunos`. Perceba que é o mesmo nome que está escrito no código!

A classe R, gerada automaticamente pelo **Android**, serve para fazer referência a qualquer recurso que está na pasta "res". O res vem de *resources*, isto é, recursos.

Na subpasta *Layout* queremos acessar o arquivo "activity_lista_alunos.xml". Ao clicar duas vezes nesse arquivo iremos visualizar uma perspectiva mais visual. Mas, vamos deixar isso de lado por um momento!

Clicando na aba "Text" - situada no lado esquerdo - abre-se o conteúdo do arquivo `.xml`. Nela está descrito o conteúdo da tela, isto é, o seu *Layout*. Vemos um arquivo `.xml` e algumas tags. Por exemplo, a tela abre em cima com `<RelativeLayout>` e fecha com `</RelativeLayout>`. Dentro do `RelativeLayout` tem um texto, por enquanto vamos deixá-lo de lado e prestar atenção no `TextView`. Quando falamos em View no **Android** é uma referência ao que o usuário pode ver e as coisas com as quais ele interage. Por exemplo: um texto, um botão, uma imagem, etc. Tudo isso são as *views*, se você quiser colocar algum componente visual na tela para mostrar um texto é necessário utilizar um `<TextView ...>`.

Após fazer isso é necessário especificar algumas propriedades, por enquanto vamos nos atentar ao atributo `text`. Nele está descrito qual o texto que desejamos que seja exibido na tela, no caso já estava escrita a expressão *"Hello World!"*. Como queremos mostrar o nome dos alunos no aplicativo apagamos esse texto e acrescentamos o texto desejado, por exemplo, o nome de um primeiro aluno: "Daniel":

```
<TextView android:text="Daniel" android:layout_width="wrap_content" android:layout_height="wrap_
```

Feita essa modificação podemos rodar novamente a aplicação. Para isso temos que voltar ao Menu e "Run > Run app". Como a aplicação já foi rodada, existe uma opção mais rápida que é clicar apenas no botão verde de *play*. Selecionando esse botão o *building* é realizado e é mostrada novamente a janela "Choose Device". Como o emulador já foi rodado podemos selecionar a opção do `EmuladorAlura Android 5.1` e clicamos no "OK". Aguarda-se um pouco e ao subir vemos a modificação no `TextView`, aparece o nome digitado: "Daniel".