## **Utilizando menus**

Apenas lembrando que no último vídeo conseguimos introduzir no botão "+" a função de ir até o formulário e no botão "Salvar" a de voltar para a lista. Também conseguimos apertar o botão de voltar da tela do celular e fazer com que ele retorne ao *Launcher*.

Agora, vamos alterar o botão de "Salvar" por um com um com um design mais bacana. Para que ele combine com o nosso botão "+" com padrão *google*. Lembrando como ele está no momento:



Vamos introduzir o botão de salvar junto a nossa barra de "Formulário". Desse modo, se o formulário for muito grande e a barra tiver que rolar para ver tudo será mais fácil se o botão ficar em cima.

Vamos para a aba FormularioActivity.java. Essa é nossa tela:



A barra, entretanto, é controlada pelo próprio *Android*. Quando extendemos a activity com o AppCompatActivity estamos dizendo que queremos uma aplicação que tem uma *action bar*. Então, precisamos comunicar ao *Android* que queremos introduzir na barra um botão.

Podemos fazer isso introduzindo um método na activity. Para isso, damos alguns "enter"s após o método e escrevemos onCreateOptionsMenu ele vai sugerir um método. Dando um enter, automaticamente ele vai preencher o *override* desse método.

Esse método é no *Android* cria a barra do Formulário e define quais os itens do menu que constarão nela. Os botões que queremos inserir fazem parte do menu da aplicação. A diferença é que algumas opções serão mostradas dentro da barra e as outras como menu.

Ficaremos com:

@Override   
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);   
 setContentView(R.layout.activity\_formulario);   
  
 Button botaoSaLvar = (Button) findViewById(R.id.formulario\_salvar);   
 botaoSalvar.setOnCIickListener(new View.0nCIickListener() {   
 @Override   
 public void onClick(View v) {  
 Toast.makeText(FormularioActivity.this. "Aluno salvo!", Toast.LENGTH\_SHORT).show();  
 finish();  
 }  
 });  
 }  
  
  
 @Override   
 public boolean onCreateOptionsMenu(Menu menu) {   
 return super.onCreateOptionsMenu(menu);   
  
 }

Inicialmente, não temos nada dentro do método e para criar um menu vamos na pasta "res". Clicando duas vezes nela vemos que já existem dois menus criados: "menu\_formulário.xml" e o "menu\_lista\_alunos.xml". Eles foram criados automaticamente pelo *Android Studio*, pois, toda vez que criamos uma activitycriam-se esse menus.

Para ver isso, basta ir em menu, "File > New File > Blanky Activity". Quando abrir a janela pedindo os dados da nova *Activity* repare que existe um campo referente ao menu, o *Menu Resources Name*. Esse campo diz respeito ao menu que será criado. automaticamente, junto com a activity.

Se abrirmos o "menu\_formulário.xml" veremos que já aparece um menu mais ou menos pronto. Nele já contêm uma *tag* menu que diz, justamente, que isso representa um menu. Nós temos a seguinte tela:

<menu 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"   
 tools:context="br.com.alura.agenda.FormularioActivity">   
 <item android:id="@+id/action\_settings"  
 android:title="Settings"  
 android:orderInCategory="100" app:showAsAction="never" />   
  
</menu>

Dentro do menu temos um item, o Settings. Bom, esse item não será necessário e vamos simplesmente apagá-lo.

Vamos começar a descrever os itens que queremos no menu. Damos um enter e inserimos a *tag* item, introduzindo também um id que chamaremos de menu\_formulário\_ok. Já faremos isso pois iremos manipulá-lo mais adiante. Vamos inserir o android:id="@+id/menu\_formulario\_ok".

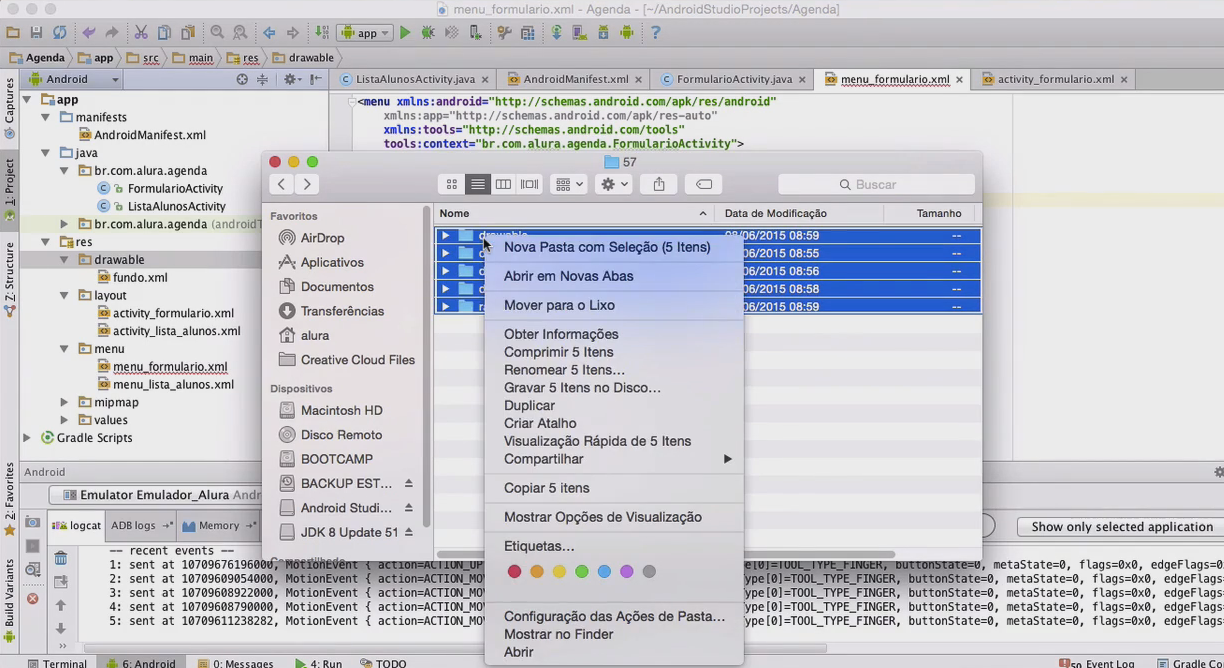
Também iremos introduzir um texto, para indicar o que significa essa opção, vamos colocar que o título dele será "ok". Damos um enter e acrescentamos 'android:title="Ok"'.

Outra coisa que acrescentaremos é um ícone. Usaremos, para tanto, o icon,assim, o botão aparecerá em forma de desenho na nossa barra. Para fazer isso utilizaremos, na próxima linha, icone também um recurso desenhável, o drawable, ficando android:icon="@drawable" . Teremos:

<menu 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"   
 tools:context="br.com.alura.agenda.FormularioActivity">   
 <item android:id="@+id/action\_settings"  
 android:title="Settings"  
 android:orderInCategory="100"   
 app:showAsAction="never" />   
  
 <item android:id="@+id/menu\_formulario\_ok"  
 android:title="Ok"  
 android:icon="@drawable" />  
  
</menu>

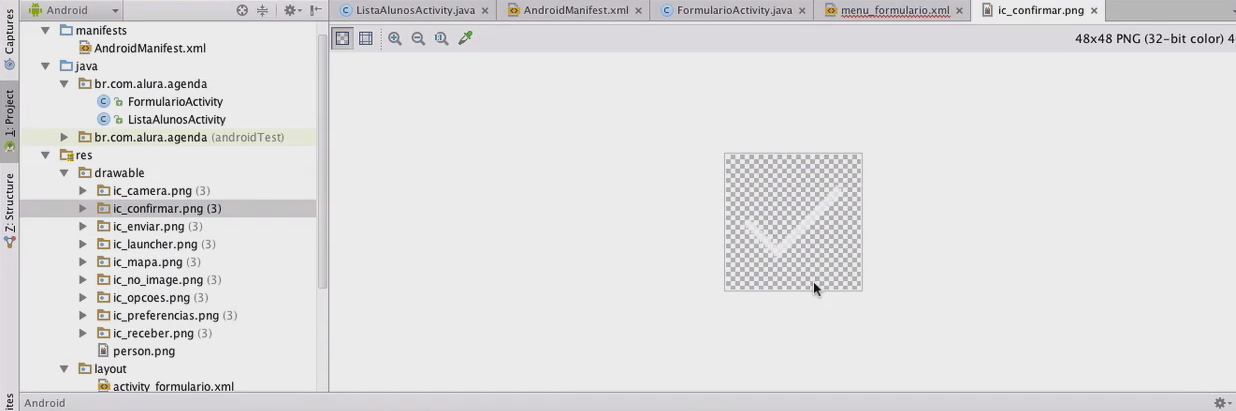
Para completar o icon temos que utilizar um recurso, por enquanto, se olharmos na pasta "drawable" temos apenas o arquivo fundo.xml.

Não tem problemas, vamos no arquivo e copiamos todas as pastas pois nelas estão contidas todas as imagens que utilizaremos ao longo do curso.



Selecionamos com o mouse todas as pastas, clicamos com o botão direito e selecionamos "Copiar 5 itens" e levamos para a pasta do nosso projeto. Fazemos o seguinte caminho, "AndroidStudioProjects > Agenda > app > src > main > res ". Chegando no "res" é só pedir para colar os itens, clicando com o botão direito e selecionando "Colar 5 itens". Se abrir uma janela perguntando se desejamos substituir é só escolher o "Substituir" . Agora temos novos recursos na aplicação.

Assim, voltando para o *Android Studio* se abrirmos a pasta "res", veremos que ela está cheia de novos itens. O que utilizaremos é o item de confirmar, clicando duas vezes ele irá aparecer,



O nome desse item é ic\_confirmar.png.

Vamos retornar na nossa aba menu\_formulario.xml e completamos o nosso icon com o nome do item que vamos adicionar. Ficando android:icon="@drawable/ic\_confirmar". Lembre-se de fechar com o />. Teremos:

<menu 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"   
 tools:context="br.com.alura.agenda.FormularioActivity">   
 <item android:id="@+id/action\_settings"  
 android:title="Settings"  
 android:orderInCategory="100"   
 app:showAsAction="never" />   
  
 <item android:id="@+id/menu\_formulario\_ok"  
 android:title="Ok"  
 android:icon="@drawable/ic\_confirmar"/>  
  
</menu>

Agora que criamos o .xml precisamos utilizar ele no nosso código. Se voltarmos no FormularioActivity.java vemos o método que o *Android* criou para chamar a barrinha:

@Override   
 public boolean onCreateOptionsMenu(Menu menu) {   
 return super.onCreateOptionsMenu(menu);   
  
 }

Nesse método vamos pedir que o *Android* pegue o menu\_formulario.xml e transforme em View. Isto é, transformar em coisas com as quais o usuário possa interagir.

Para pegar um .xml qualquer em um conjunto de Views a gente usa um conceito do *Android* chamado 'inflater'

Para isso, ficamos no FormulárioActivity.java e no método menu, damos um enter e na próxima linha, acrescentamos um conceito especialista em fazer menus, que é ogetMenuInflater.

Esse método getMenuInflater está vindo da AppCompatActivity e serve para devolver uma instância de um menu inflater. Então, acrescentaremos antes do que já escrevemos, MenuInflater inflater e um "=" a getMenuInflater. Teremos, MenuInflater inflater = getMenuInflater. Se ficar em vermelho lembre de importar dando "Alt+Enter".

Temos que lembrar que o inflater tem que inflar algo, no caso ele é especialista em inflar uma .xml. Temos que informar para ele qual a xmlque ele vai inflar.

Então, digitamos a classe R, um ".", e como queremos inflar um menu, vamos buscar ele dentre as opções apresentadas, escolhemos "menu", digitamos "." de novo e vamos escolher o "menu\_formulario", pois é justamente isso que queremos inflar. Teremos, inflater(R.menu.menuformulário).

Precisamos dizer, ainda, onde ele vai colocar essas informações, no caso, no menu da *action bar* que está vazio. Ele é aquele "menu" da *action bar* que está vazio.



Completaremos com mais um "." e menu, para indicar que queremos que todos os itens estejam nele. Completando, ficaremos com, inflater(R.menu.menuformulário.menu).

Não podemos esquecer de chamar o método inflate. Ele será adicionado depois do `inflater', damos um "." digitamos 'inflate'.

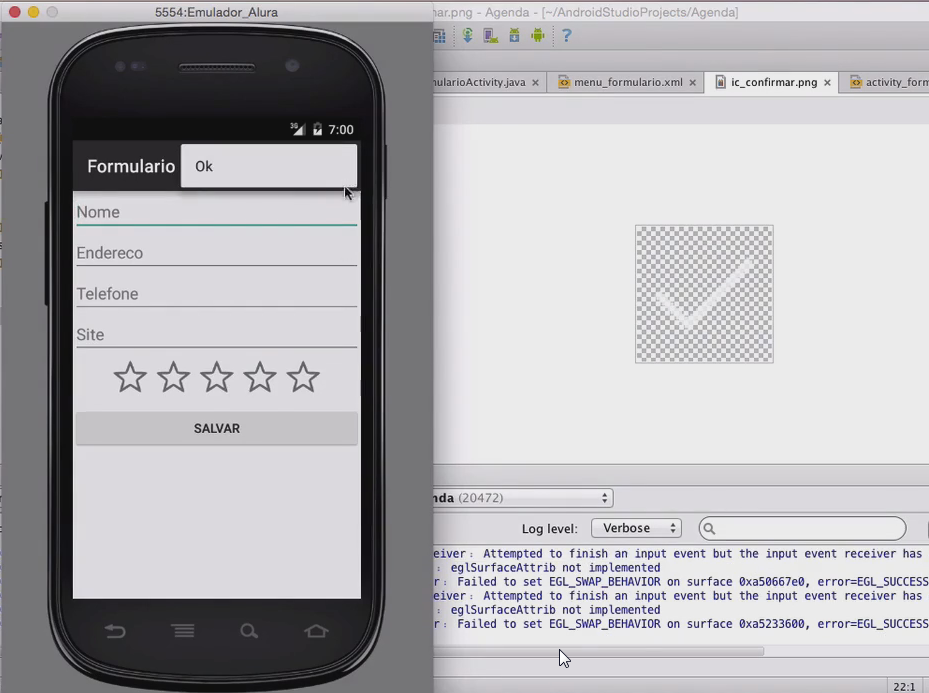
Ficaremos com:

@Override   
 public boolean onCreateOptionsMenu(Menu menu) {   
  
 MenuInflater inflater = getMenuInflater();  
 inflater.inflate(R.menu.menu\_formulario, menu);  
  
 return super.onCreateOptionsMenu(menu);   
  
  
 }

Vamos ver o que acontece se fizermos só isso:



Nossa barra não está mais vazia! Ela está com esses três pontinhos no canto direito. Se clicarmos em cima das três bolinhas o que aparece é um "ok" escrito. Nosso ícone do menu veio parar aqui e de maneira textual.



Mas, nós tínhamos pedido para ele colocar o item de *check mark*, aquele que aparece na imagem acima. Lembra?!

Para mostrar o item do menu, não em forma de texto, mas em forma de ícone vamos voltar na aba 'menu\_formulario.xml'. Vamos usar a propriedade showAsaction Quando digitarmos ele vai dar algumas opções de escolha, vamos usar o always, pois queremos que nosso ícone seja sempre mostrado na *action bar*. Acrescentaremos, android:showAsaction="always".

O showAsaction precisa vir de um outro *name space*. Aqui temos dois *name space*, um 'android' e um 'app'. Usaremos o app. Apagamos o android e digitaremos app no lugar e ":". Teremos app:showAsaction="always".

Ficaremos com:

<menu 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"   
 tools:context="br.com.alura.agenda.FormularioActivity">   
 <item android:id="@+id/action\_settings"  
 android:title="Settings"  
 android:orderInCategory="100"   
 app:showAsAction="never" />   
  
 <item android:id="@+id/menu\_formulario\_ok"  
 android:title="Ok"  
 android:icon="@drawable/ic\_confirmar"  
 app:showAsaction="always"/>  
  
</menu>

Com isso estamos dizendo que esse item do menu deve ser mostrado com uma das ações da *action bar*.

Vamos rodar para ver como fica.



Agora temos o botão de confirmar, que inclusive já tem uma animação quando clicamos nele!

Falta colocar o comportamento do botão e excluir o botão de "Salvar" que está embaixo do formulário.

Vamos na aba activity\_formulario.xml para tirar o botão antigo. Selecionamos com o mouse o seguinte e depois apagamos:

<Button android:layout\_width="match\_parent"   
 android:layout\_height="wrap\_content"   
 android:text="Salvar"   
 android:id="@+id/formulario\_salvar"/>

Pronto! Se voltarmos agora no FormularioActivity.java não encontraremos mais o R.id.formulario\_salvar.

Só que agora precisamos de uma forma de definir qual o comportamento do clique no menu. Poderíamos definir um Listener, mas existe uma forma mais elegante de fazer isso no *Android*. De novo vamos subscrever um método que já existe, o onOptionsItemSelected e acrescentaremos isso abaixo do seguinte código:

//...  
  
 @Override   
 public boolean onCreateOptionsMenu(Menu menu) {   
  
 MenuInflater inflater = getMenuInflater;  
 inflater.inflate(R.menu.menuformulario, menu);  
  
 return super.onCreateOptionsMenu(menu);   
  
  
 }

Ao digitar o método onOptionsItemSelected o *Android* já vai sinalizar as opções logo abaixo. Escolhemos o método que queremos e é só dar um "Enter". Pronto! Está feito o *override* do método. Teremos:

//...  
  
 @Override   
 public boolean onCreateOptionsMenu(Menu menu) {   
  
 MenuInflater inflater = getMenuInflater;  
 inflater.inflate(R.menu.menuformulario, menu);  
  
 return super.OnCreateOptionMenu(menu);  
  
 }  
  
 @Override   
 public boolean onOptionsItemSelected(MenuItem item) {   
 return super.onOptionsItemSelected(item);   
  
 }

Repare que recebemos nesse método o item que foi selecionado, o MenuItem item. Assim, toda vez que clicarmos no item no menu o *Android* vai evocar esse método na sua activity e vai informar qual foi o item no menu que foi clicado.

Agora, vamos pegar o comportamento do botão salvar, o Toast e selecionar com o mouse até o finish.

@Override   
 public void onClick(View v) {  
 Toast.makeText(FormularioActivity.this. "Aluno salvo!", Toast.LENGTH\_SHORT).show();  
 finish();  
 }

Podemos recortar o 'Toast', através do "Comand+X" ou "Ctrl+X" e colar no Override do onOptionsItemSelected com "Command+V" ou "Ctrl+V".

Temos que tomar cuidado que esse comportamento que acabamos de inserir será extendido para os demais itens do menu. Assim, se tivessemos 50 itens no menu, todos eles seriam englobados nesse comportamento. Como não queremos que isso ocorra, precisamos fazer uma distinção. Para isso, podemos fazer um switch, que acrescentaremos na linha de baixo de public void. Digitaremos switch e entre parênteses colocaremos "item", um ".", e acrescentaremos o Id do item. Teremos, switch (item.getItemId()).

Dependendo do Id que tivermos teremos casos distintos. Na linha de baixo do switchteremos case R.id.menu\_formulario\_ok.

Caso o id seja o menu\_formulário\_ok queremos ter o comportamento que inserimos anteriormente, selecionamos, mais uma vez, o Toast até o finish, damos um "Comand+X" ou "Ctrl+X" e "Comand+V" ou "Ctrl+V" e deslocamento ele para baixo do case que acabamos de introduzir.

Lembre-se de colocar no final o break, seguido de um ;. Caso contrário, ele ficará executando todos os cases que vierem depois dele. Se não quisermos esse comportamento, acrescentamos o breakque ele "tranca" o switch.

@Override   
 public boolean onOptionsItemSelected(MenuItem item) {   
 switch (item.getItemId()) {  
 case R.id.menu\_formulario\_ok:  
 Toast.makeText(FormularioActivity.this, "Aluno salvo!", Toast.LENGTH\_SHORT).show();  
 finish();  
 break;  
  
 }  
 return super.onOptionsItemSelected(item);   
  
 }

Vamos deletar também o botão de salvar, pois ele não existe mais na nossa tela. Deletamos o seguinte:

Button botaoSalvar = (Button) findViewById(R.id.formulario\_salvar);   
 botaoSalvar.setOnCIickListener(new View.0nCIickListener() {   
 @Override   
 public void onClick(View v) {  
  
 }  
 });

E agora quando clicarmos no botão ele vai chamar o método onOptionsItemSelectedpassando o item que foi selecionado. Se esse item que foi selecionado for o menu\_formulario\_ok que é o item que definimos no menu\_formulario.xml, ele mostrará o Toast e termina a activity como queremos.

Vamos salvar e ver como ficou?



Agora, quando quisermos adicionar mais um aluno, vamos preencher o formulário e dar um "Ok", ele vai nos avisar que o aluno foi salvo e vai voltar para lista!