Ciclo de vida das Activities

Lembrando que conseguimos fazer com que um novo aluno apareça em nossa lista, mas ele só é adicionado quando saímos da aplicação. Vamos mudar isso!

Para fazermos essa mudança temos que entender um pouco sobre o ciclo de vida das *activities*. Para isso olharemos o seguinte desenho que está no site do *Android*. Você pode visualizar a imagem usando o atalho "d.android.com" . No site fazemos uma busca, vamos na lupa e digitamos *activity*. Vai abrir uma página com os resultados. Queremos o guia de referência, então, clicamos no *Reference* no menu do canto esquerdo.

Selecionamos o *Activity| Android Developers* que aparece logo no topo e vai abrir uma página onde logo no início teremos o ciclo de vida das *activities* através de uma imagem.



***Para entender o Ciclo de vida das Activities***

A *activity*, como vemos na imagem, evoca o método onCreate, que chama o onStartpara iniciar e o onResume, que é para a *activity* continuar e só aí ela começa a executar.

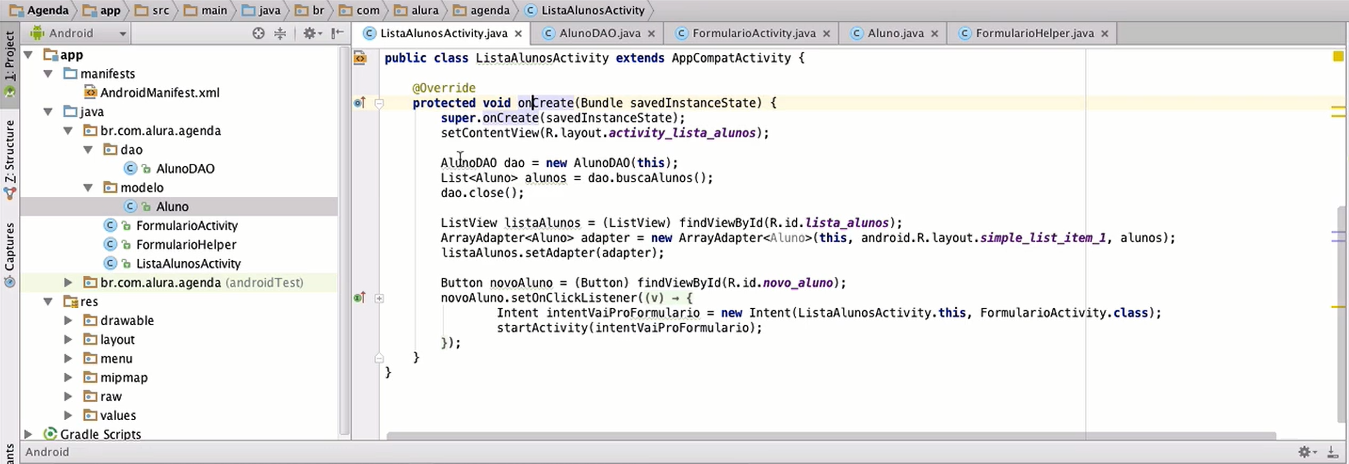
Dependendo do seu celular, se algum outro método entrar na frente da nossa *Activity* ele vai chamar o método onPause, isso ocorrerá no caso dos "*pop up*" que não tapam completamente nossa *activity*, ou seja, ele vai pausar temporariamente nossa *activity* e depois que fecharmos esse "*pop up*" ele vai chamar o onResume para retornar a atividade.

Um outro caso é o das chamadas telefônicas, e a aplicação que nos avisa do recebimento da chamada eletrônica toma a tela inteira, nesse caso o *Android* vai chamar o onPause. Como a aplicação não está mais vísivel ele vai chamar o onStop, isto é, parar nossa *activity*. Esse processo pode fazer com que se o *Android* precisar de memória, e nesse caso, ele pode matar nossa aplicação, isto é, vai seguir o caminho que a imagem mostra de "App process killed" e o processo vai iniciar todo de novo. Isso ocorre apenas quando o *Android* precisa da memória, caso contrário ele retorna a *activity* através do onRestart e reiniciamos a aplicação a partir do onStart.

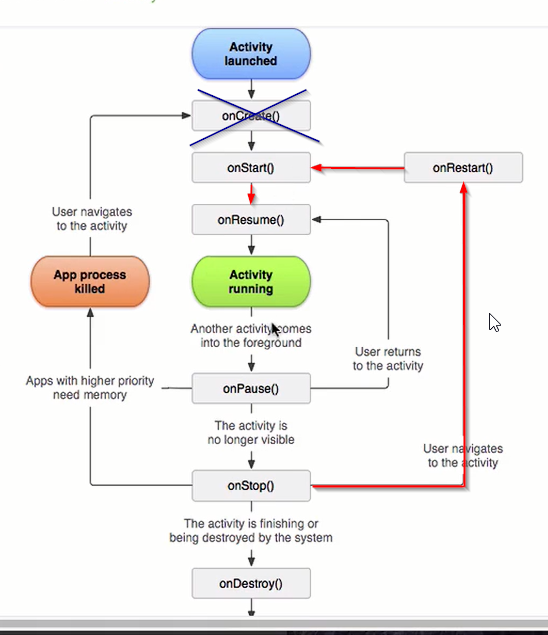
Agora que compreendemos um pouco a respeito da vida da *activity* vamos ver como isso se aplica ao que estamos fazendo!

Quando estamos na nossa tela de celular e nela aparece a Lista de Alunos, já foram executados o onCreate, o onStart e o onResume. Quando passamos para o formulário estamos em uma *activity* que toma a tela inteira, assim, o *Android* chama o onStop. Quando apertarmos o botão de *check mark* a *activity* vai seguir o caminho da *Activity shut down*, como demonstra a imagem. Só que quando acaba a *activity* do fomulário voltamos para a lista de alunos que estará no onStop que seguirá o caminho onRestart e seguir no onStart normalmente.

Vamos voltar um pouquinho no *Android Studio* e ver onde é feito o carregamento da nossa lista. Vamos na ListaAlunosActivity.java e temos a seguinte tela:



O carregamento está sendo feito no método onCreate e aí temos o problema, pois o ciclo da *activity* depois do onStop dirige-se para o onRestarte o onResume e não passa pelo onCreate. Aí é que temos o problema! É por isso que não conseguimos mostrar o aluno que acrescentamos na lista.



Para resolver isso podemos acrescentar um onResume , pois todos os caminhos passam por ele. Se colocássemos um método onStart teríamos o problemas de que, caso houvesse uma pausa, não voltaríamos para o onStart e de novo nossa lista não carregaria.

Para fazer isso, vamos na ListaAlunosActivity.java e escrevemos abaixo do onCreate um onResume, damos um "Enter". Vai ficar da seguinte maneira:

@Override   
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_lista\_alunos);  
  
 AlunoDAO dao = new AlunoDAO(this);   
 List<Aluno> alunos = dao.buscaAlunos(); dao.close();  
  
 ListView listaAlunos = (ListView) findViewById(R.id.lista\_alunos);  
 ArrayAdapter<Aluno> adapter = new ArrayAdapter<Aluno>(this, android.R.layout.simple\_list\_item\_1, alunos);  
 listaAlunos.setAdapter(adapter);   
  
 Button novoAluno = (Button) findViewById(R.id.novo\_aluno);  
 novoAluno.setOnClickListener((v) -> {   
 Intent intentVaiProFormuIario = new Intent(ListaAlunosActivity.this, FormularioActivity.class); startActivity(intentVaiProFormulario);  
 });  
 }  
  
 @Override   
 protected void onResume() {  
 super.onResume();  
 }  
}

Lembre-se de chamar o super para manter o comportamento da AppCompatActivity!

O código que faz o carregamento vai do AlunoDAO até o fim do ListView. Para não precisar jogar esse código que faz o carregamento da lista de um lado para outro, vamos extrair ele para um método que faça o carregamento efetivamente. Para isso, selecionamos com o mouse o seguinte:

AlunoDAO dao = new AlunoDAO(this);   
 List<Aluno> alunos = dao.buscaAlunos(); dao.close();  
  
 ListView listaAlunos = (ListView) findViewById(R.id.lista\_alunos);  
 ArrayAdapter<Aluno> adapter = new ArrayAdapter<ALuno>(this, android.R.layout.simple\_list\_item\_1, alunos);  
 listaAlunos.setAdapter(adapter);

E com o botão direito "Refactor> Extract> Method". Vamos clicar no *Method* e vai abrir uma janela pedindo para dar um nome a esse método. Iremos chamar de "carregalista" e vai ser um método *private*. É só dar um "ok". teremos o seguinte:

@Override   
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_lista\_alunos);  
  
 carregaLista();   
  
 Button novoAluno = (Button) findViewById(R.id.novo\_aluno);  
 novoAluno.setOnClickListener((v) -> {   
 Intent intentVaiProFormuIario = new Intent(ListaAlunosActivity.this, FormularioActivity.class); startActivity(intentVaiProFormulario);  
 });  
 }  
  
 private void carregaLista() {  
 AlunoDAO dao = new AlunoDAO(this);   
 List<Aluno> alunos = dao.buscaAlunos(); dao.close();  
  
 ListView listaAlunos = (ListView) findViewById(R.id.lista\_alunos);  
 ArrayAdapter<Aluno> adapter = new ArrayAdapter<ALuno>(this, android.R.layout.simple\_list\_item\_1, alunos);  
 listaAlunos.setAdapter(adapter);  
 }  
  
 @Override   
 protected void onResume() {  
 super.onResume();  
 }  
}

Mas, repare que o carregaLista está no Oncreate, e não queremos carregá-lo a partir daí, se não teremos o mesmo erro de antes. Vamos selecionar o carregaLista(); e dar um "Comand +X" ou "Ctrl+X" e no onResume, depois do super, damos um "Command+V" ou "Ctrl+V" e teremos:

@Override   
 protected void onResume() {  
 super.onResume();  
 carregaLista();   
 }

Agora sim, carregamos a lista a partir do onResume. Vamos ver se deu certo? Vamos rodar o emulador e quando adicionarmos mais uma pessoa, por exemplo, preencheremos o formulário com o "Felipe". Agora, salvando esse formulário, por fim, teremos Felipe em nossa lista!

