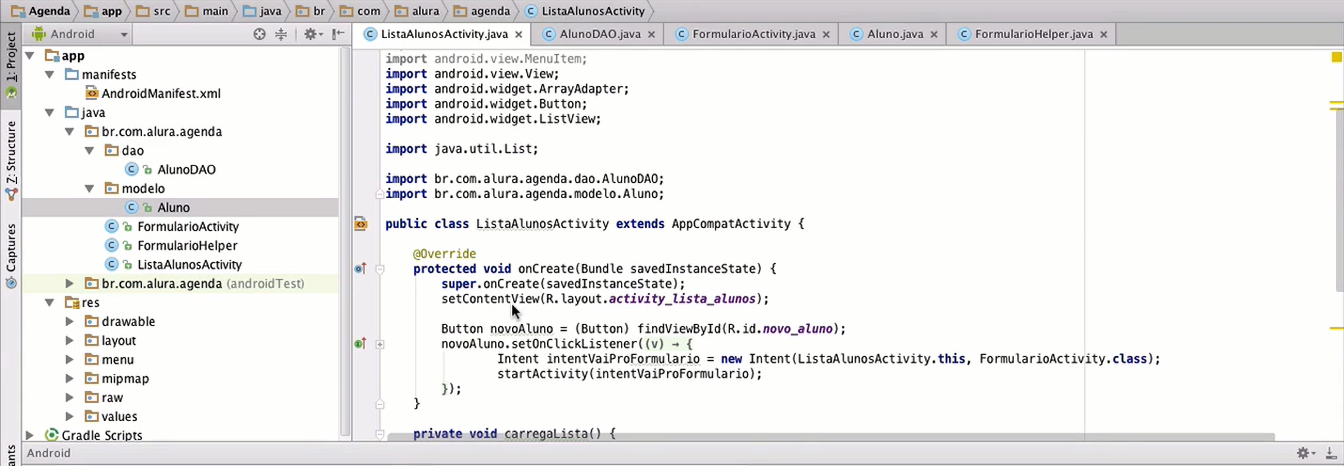
Removendo alunos

Nossa aplicação agora já consegue armazenar os alunos. Suponha que nós precisamos deletar algum aluno. Como faríamos para apagar esse nome da nossa lista?

O que vamos fazer agora é criar um menu de contexto para cada nome, isto é, ao darmos um *click* longo em cima do nome do aluno que queremos excluir, abrirá uma janelinha, isto é, um menu de contexto, e nela poderemos escolher opções, dentre elas, a de "Deletar" um aluno.

Vamos voltar na aba *ListaAlunosActivity.java* e dar uma olhada em como está nossa tela:



Agora, precisamos dizer qual o componente da nossa tela que vai possuir um menu de contexto. Vamos digitar abaixo de onCreate , um registerForContextMenu e acrescentamos entre os parênteses listaAlunos . A listaAlunos ficará em vermelho, pois ela está sem referência, o findViewById da lista de alunos está no carregaLista. Para solucionar isso copiaremos a linha ListView listaAlunos = (ListView) findViewById (R.id.lista\_alunos) através do "Command+X" e daremos um "Command+V" acima da linha Button novoAluno.

Para que a listaAlunos que acabamos de mover continue funcionando em ambos os lados, temos que transformá-la em um atributo. Fazemos um "Alt+Enter" em cima dele e selecionamos "Create field". Agora o listaAlunos virou um atributo que podemos acessar tanto no onCreate quanto no carregaLista.

Ficaremos com:

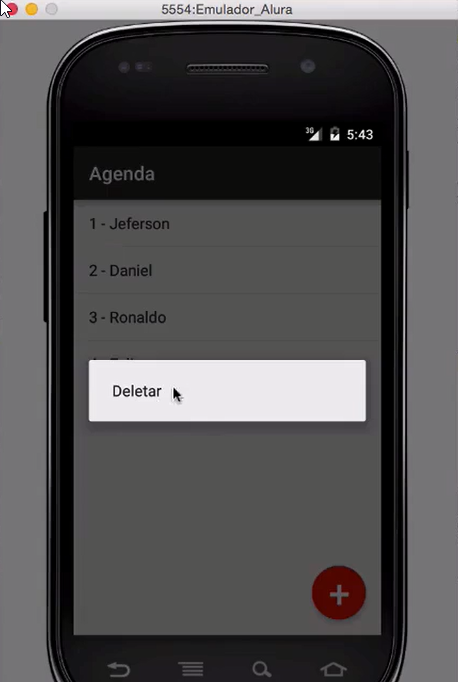
public class ListaAlunosActivity extends AppCompatActivity {  
  
 private ListView listaAlunos;  
  
 @Override   
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_lista\_alunos);   
  
 listaAlunos = (ListView) findViewById(R.id.lista\_alunos);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

Lembra que já utilizamos um método para criar um menu antes? O onOptionsMenu. Agora, utilizaremos algo similar, o onCreateContextMenu, que acrescentaremos abaixo do método onResume. Ao escrevermos é só dar um "Enter" e ele já vai sobrescrever. Podemos apagar a linha do super, pois ela não será necessária. Dentro desse método vamos popular o Menu.

Para fazer isso poderíamos fazer o mesmo que fizemos no FormularioActivity.java, que foi criar uma .xml e fazer um *inflate*. Como já fizemos esse caminho antes, vamos fazer um diferente agora, vamos fazer "a mão". Para adicionar um menu digitaremos menu.add e entre o parênteses o item que queremos acrescentar "Deletar". Ficaremos com o seguinte abaixo do onResume:

@Override  
 protected void onResume() {  
 super.onResume();  
 carregaLista();   
 }  
  
 @Override  
 public void onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo) {  
 menu.add("Deletar");   
 }

Vamos rodar no emulador para ver o que acontece!



Agora que temos uma opção no nosso menu, vamos introduzir um comportamento no "Deletar" para que ele avise que o aluno foi deletado.

Para fazer isso, poderíamos seguir o caminho que fizemos na *FormularioActivity.java*, por exemplo, colocar um comportamento similar ao onOptionsItemSelected, o onContextItemSelected, mas se fizermos isso teremos que acrescentar um switch, para filtrar, isto é, indicar o caminho a ser feito. Quando precisarmos inserir um Id é aqui que teremos um problema, pois o item a mão não tem um id, só possui um quando criado na .xml.

Vamos seguir por outro caminho, vamos utilizar a ideia do Listener, assim, quando o botão é clicado o *Android* nós avisará.

seguimos no onCreateContextMenu. Se queremos manipular um item do menu vamos ter que adicionar uma referência desse item. Vamos colocar essa referência na frente do menu.add. Ficaremos com MenuItem deletar = menu.add("Deletar"). Agora, acrescentaremos na linha de baixo o deletar.setOnMenuItemClicklistener que significa que estamos interessados em escutar o evento de *click* no item do menu. Ele vai pedir, ainda, um "OnMenuItemClickListener" e poderemos preencher com um new onMenu. Dando um "Enter", o *Android* vai criar uma classe anônima, dentro da qual poderemos fazer o que quisermos.

Estamos com o seguinte:

@Override  
 public void onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo) {  
 MenuItem deletar = menu.add("Deletar");  
 deletar.setOnMenuItemClickListener(new MenuItem.OnMenuItemClickListener() {  
 @Override  
 public boolean onMenuItemClick(MenuItem item) {  
  
 return false;  
 }  
 });  
 }

Lembrando que o MenuItemse refere ao "Deletar" e não ao aluno número 5.

Vamos observar o onCreateContextMenu. Ele traz para a gente, um ContextMenuInfo que contêm a informação que necessitamos. É ele quem vai dizer qual o item da lista foi clicado para gerar um ContextMenu. Para isso precisaremos informar que o ContextMenuInfo é um *adapter*.

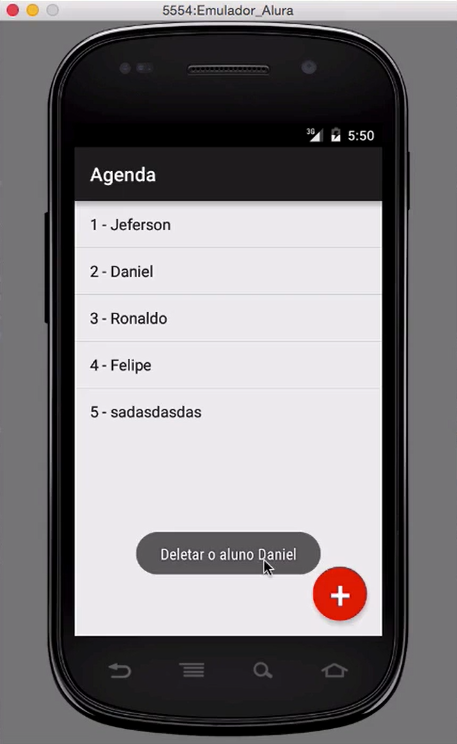
Na linha de baixo vamos digitar um *adapter* que chamaremos de info. Teremos AdapterView.AdapterContextMenuInfo info = menuInfo. Vamos falar para o *Android que queremos nessa referência algo mais específico*. Vamos fazer um *Cast*. Para isso basta dar um "Alt+Enter" em cima do menuInfo. Caso não ocorra essa sugestão podemos fazer "a mão" completando o *Adapter*. Teremos, AdapterView.AdapterContextMenuInfo info = (AdapterView.AdapterContextMenuInfo) menuInfo. Como o parâmetro da classe anônima não conseguimos declarar, vamos apenas dizer no onCreate que é um final.

Dentro do info teremos um position, que nos informa qual posição da lista que acabamos de clicar. Bom, se ele nos informa a posição basta informar para o *Android* que queremos que ele devolva o aluno que está nessa determinada posição. Para fazer isso basta acrescentarmos na linha de baixo do Adapter um método chamado getItemAtPosition que pedirá a posição que queremos recuperar, que pediremos por sua vez ao info que devolverá um Aluno e, ainda, falaremos que ele é igual a todo o item que escrevermos. Por fim, faremos um *Cast* nele através do "Alt+Enter" e ficaremos com Aluno aluno = (Aluno) listaAlunos.getItemAtPosition(info.position).

Para mostrar um comportamento após escolhermos "Deletar", podemos fazer um Toast.makeText, importar o Toast e passar o contexto, ListaAlunosActivity.this e a mensagem, "Deletar o aluno" + aluno.getNome, acrescentando a duração da nossa mensagem Toast.LEGHT\_SHORT. Ficaremos com, Toast.makeText(ListaAlunosActivity.this, "Deletar o aluno" + aluno.getNome(), Toast.LEGTH\_SHORT) e no final evocaríamos um método *show*, digitando um .e show. Ficaríamos com:

@Override  
 public void onCreateContextMenu(ContextMenu menu, View v, final ContextMenu.ContextMenuInfo menuInfo) {  
 MenuItem deletar = menu.add("Deletar");  
 deletar.setOnMenuItemClickListener(new MenuItem.OnMenuItemClickListener() {  
 @Override  
 public boolean onMenuItemClick(MenuItem item) {  
 AdapterView.AdapterContextMenuInfo info = (AdapterView.AdapterContextMenuInfo) menuInfo;  
 Aluno aluno = (Aluno) listaAlunos.getItemAtPosition(info.position);  
 Toast.makeText(ListaAlunosActivity.this, "Deletar o aluno" + aluno.getNome(), Toast.LENGTH\_SHORT).show();  
 return false;  
 }  
 });  
 }

Vamos rodar no emulador para ver como ficou?



Agora só falta deletar de fato o aluno. Para isso usaremos o dao. Vamos instanciar o AlunoDAO na linha de baixo do getItemAtPosition. Escreveremos, AlunoDAO dao = new AlunoDAO(ListaAlunosActivity.this). Falaremos para ele, na linha seguinte, deletar o aluno que selecionarmos, para isso, digitaremos dao.deleta(aluno).

Não esquecendo que se trata de um banco de dados e que é importante fechá-lo. Utilizaremos o dao.close.

Podemos apagar o Toast porque veremos a operação acontecendo, então, não precisaremos mais de uma mensagem nos avisando que algo vai acontecer.

Ficaremos com:

@Override  
 public void onCreateContextMenu(ContextMenu menu, View v, final ContextMenu.ContextMenuInfo menuInfo) {  
 MenuItem deletar = menu.add("Deletar");  
 deletar.setOnMenuItemClickListener(new MenuItem.OnMenuItemClickListener() {  
 @Override  
 public boolean onMenuItemClick(MenuItem item) {  
 AdapterView.AdapterContextMenuInfo info = (AdapterView.AdapterContextMenuInfo) menuInfo;  
 Aluno aluno = (Aluno) listaAlunos.getItemAtPosition(info.position);  
  
 AlunoDao dao = new AlunoDao(ListaAlunosActivity.this);  
 dao.deleta(aluno);  
 dao.close();  
  
 return false;  
 }  
 });  
 }

Agora, falta implementar o "Deleta", então, damos um "Alt+Enter" e "Create method". Nós iremos para a aba *AlunoDAO* e o novo método vai aparecer lá embaixo depois do while. Poderíamos fazer a operação a mão, mas, novamente, vamos pedir para que o SQLiteOpenHelper faça isso por nós. Vamos pegar uma referência do banco de dados e digitar isso na linha de baixo do public void deleta(Aluno aluno) que acabamos de criar. Vamos digitar o seguinte, SQLiteDatabase db = getWritableDatabase. O getWritableDatabase é porque queremos manipular esse banco de dados.

Embaixo disso que acabamos de introduzir vamos chamar o método db.delete que pedirá alguns parâmetros, a tabela, por isso adicionaremos "Aluno" e também a cláusula *where*, o que queremos remover, no caso quais os alunos. Preenchemos com todos os alunos cujo id seja getId. Teremos db.delete("Aluno", "id = " + aluno.getId()).

Observem que de novo estamos concatenando, mas não vamos fazer isso, sempre que quisermos concatenar algum valor no parâmetro, ao em vez de concatenar vamos deixar um espaço reservado, vamos marcar o espaço com uma interrogação e vamos informar na frente dessa interrogação o parâmetro, digitaremos params. O params pede uma *Array* de *String*, então, adicionaremos na linha anterior os parâmetros que o params pede, que são os valores. Acrescentaremos o Id do nosso aluno. Lembrando que o Id é uma *Long* e pede um toString para ser convertido. TeremosString[] params = {aluno.getId().toString}.

No todo teremos:

//...  
  
 public void deleta(Aluno aluno) {  
 SQLiteDatabase db = getWritableDatabase();  
  
 String [] params = {aluno.getId().toString()};  
 db.delete("Alunos", "id = ?", params);   
 }

Se rodarmos o emulador veremos que o aluno só deleta quando sairmos da aplicação.

Vamos tentar entender!

Quando selecionamos a opção de "Deletar", observe que a nossa *activity* não sai do primeiro plano, ela está aparente, ao fundo do botão de "Deletar". Nossa activity, portanto, continua a rodar e por isso essa ação não vai passar pelo onResume do nosso AlunoDAO.java e, assim, não vai carregar a lista.

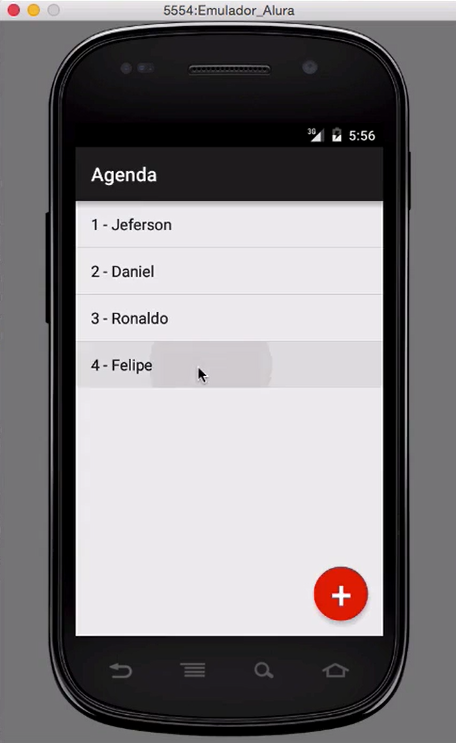
Como não saímos da *Activity* não podemos depender apenas do ciclo de vida da *activity*, teremos que chamar o método carregaLista.

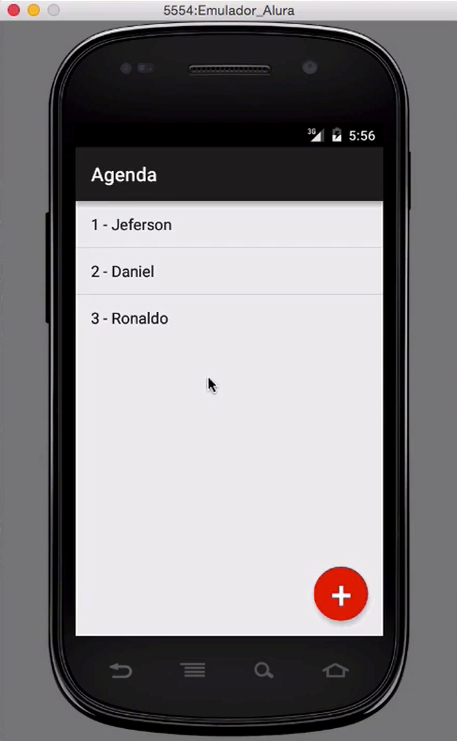
Bom, vamos voltar na *ListaAlunosActivity.java*, no onCreateContextMenu. Na linha seguinte do dao.close vamos carregar a lista de alunos, para isso, introduzimos um carregaLista. Isso serve para atualizar nossa lista.

Ficaremos com:

public class ListaAlunosActivity extends AppCompatActivity {  
  
 private ListView listaAlunos;  
  
 @Override   
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_lista\_alunos);   
  
 listaAlunos = (ListView) findViewById(R.id.lista\_alunos);  
  
 Button novoAluno = (Button) findViewById(R.id.novo\_aluno);   
 novoAluno.setOnClickListener((v) -> {   
 Intent intentVaiProFormulario = new Intent(ListaAlunosActivity.this, FormularioActivity.class); startActivity(intentVaiProFormulario);   
 });  
  
 registerForContextMenu(listaAlunos);  
 }  
  
 private void carregaLista() {  
 AlunoDAO dao = new AlunoDAO(this);  
 List<Aluno> alunos = dao.buscaAlunos();  
 dao.close();  
  
 ArrayAdapter<Aluno> adapter = new ArrayAdapter<Aluno>(this, android.R.layout.simple\_list\_item\_1, alunos);  
 listaAlunos.setAdapter(adapter);   
 }  
  
 @Override  
 protected void onResume() {  
 super.onResume();  
 carregaLista();   
 }  
  
 @Override  
 public void onCreateContextMenu(ContextMenu menu, View v, final ContextMenu.ContextMenuInfo menuInfo) {  
 MenuItem deletar = menu.add("Deletar");  
 deletar.setOnMenuItemClickListener(new MenuItem.OnMenuItemClickListener() {  
 @Override  
 public boolean onMenuItemClick(MenuItem item) {  
 AdapterView.AdapterContextMenuInfo info = (AdapterView.AdapterContextMenuInfo) menuInfo;  
 Aluno aluno = (Aluno) listaAlunos.getItemAtPosition(info.position);  
  
 AlunoDAO dao = new AlunoDAO(ListaAlunosActivity.this);  
 dao.deleta(aluno);  
 dao.close();  
  
 carregaLista();   
 return false;  
 }  
 });  
 }  
}

Vamos rodar de novo e ver se deu certo, vou selecionar na nossa tela o aluno Felipe para deletar.





Pronto! Agora conseguimos deletar os alunos.