

Removendo alunos

Transcrição

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

O que faremos é criar um menu de contexto para cada nome, isto é, ao dar um *click* longo sobre um nome, abre-se um menu de contexto. Nele podemos escolher dentre as opções e uma delas será "Deletar" aluno.

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

Agora, precisamos dizer qual componente 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, pois 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 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, FormularioActivi
        });

        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_listitem_
        listaAlunos.setAdapter(adapter);
    }

    @Override
    protected void onResume() {
```

```
        super.onResume();
        carregaLista();
    }
}
```

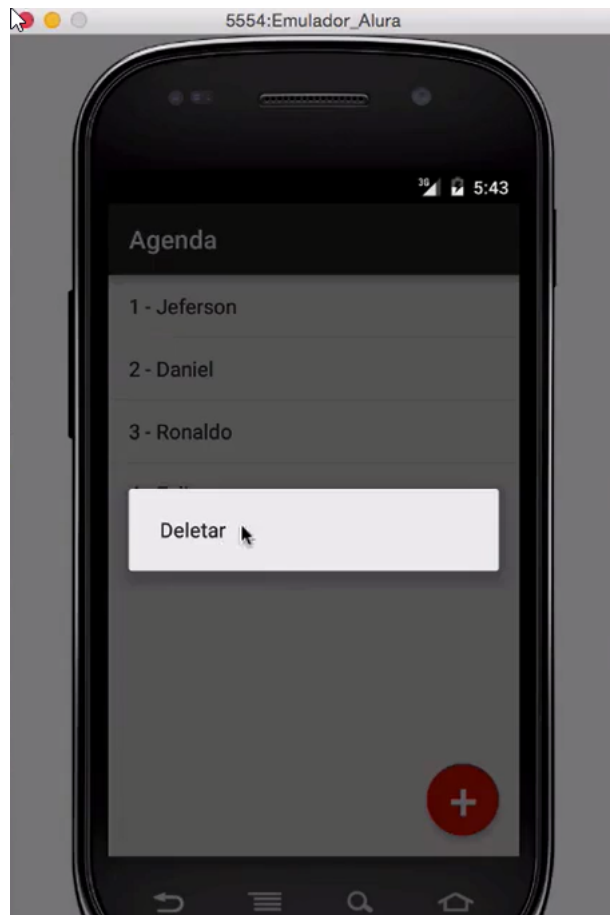
Lembra que já utilizamos um método para criar um menu antes? O `onOptionsItemSelected` ! Utilizaremos algo similar, o `onCreateContextMenu` , que acrescentaremos abaixo do método `onResume` . Ao escrevermos ele é só dar um "Enter" que já vai sobrescrever. Podemos apagar a linha do `super` , pois ela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

Para fazer isso repetiremos o que fizemos no `FormularioActivity.java` , que foi criar uma `.xml` e fazer um *inflate*. Como já fizemos esse caminho antes, vamos fazer um diferente, "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, vamos introduzir um comportamento no "Deletar" para que ele avise que o aluno foi deletado.

Para fazer isso, podemos seguir o caminho que fizemos na *FormularioActivity.java*, por exemplo, colocar um comportamento similar ao `onOptionsItemSelected`, o `onContextItemSelected`. Mas, ao fazer isso teremos que acrescentar um `switch` para filtrar, isto é, indicar o caminho a ser feito. Porém, inserir um `id` é um problema, pois o item feito a mão não tem `id`, ele só possui um quando é criado na `.xml`.

Vamos seguir por outro caminho, vamos utilizar a ideia do `Listener`, assim, quando o botão é clicado o *Android* nos 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 `new OnMenu`. Dando um "Enter", o *Android* cria uma classe anônima, dentro da qual podemos 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 `MenuItem` refere-se ao "Deletar" e não ao aluno número 5.

Vamos observar o `onCreateContextMenu`. Ele traz um `ContextMenuInfo` que contém a informação que precisamos. É ele quem vai dizer qual 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 não conseguimos declarar o parâmetro da classe anônima, vamos apenas dizer no `onCreate` que é um `final`.

Dentro da `info` teremos um `position`, que nos informa qual posição da lista acabou de ser clicada. Basta informar ao *Android* que queremos que ele devolva o aluno que está nessa posição. Para fazer isso acrescentamos na linha de baixo do *Adapter* um método chamado `getItemAtPosition` que pedirá a posição a ser recuperada. Pediremos ao `info` que devolva um `Aluno` e, por fim, faremos um *Cast* nele através do "Alt+Enter" e ficaremos com `Aluno aluno = (Aluno) listaAlunos.getItemAtPosition(info.position)`.

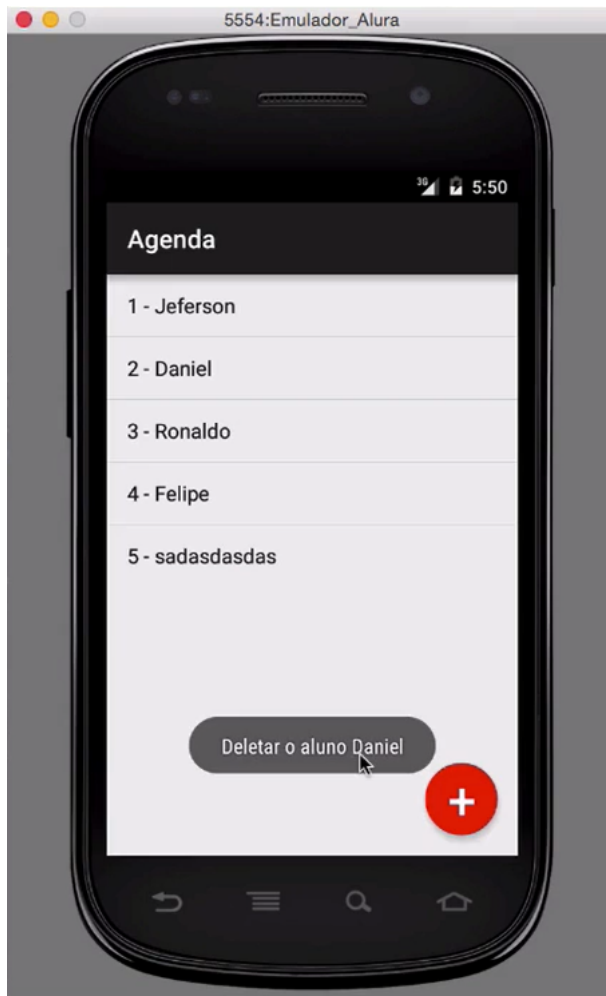
Para mostrar um comportamento após escolher "Deletar", vamos fazer um `Toast.makeText`. Importamos o `Toast` e passamos o contexto, `ListaAlunosActivity.this`. Adicionamos a mensagem, "Deletar o aluno" + `aluno.getNome()`, e acrescentamos sua duração, `Toast.LENGTH_SHORT`. Ficaremos com:

```
Toast.makeText(ListaAlunosActivity.this, "Deletar o aluno" + aluno.getNome(), Toast.LENGTH_SHORT)
```

No final evocamos um método *show*, digitando um `.` e *show*. Ficaremos com:

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

Vamos rodar no emulador para ver como ficou?



Ainda falta deletar o aluno. Para isso usaremos o `dao`. Vamos instanciar o `AlunoDAO` na linha de baixo do `getItemAtPosition`. Escreveremos, `AlunoDAO dao = new AlunoDAO(ListaAlunosActivity.this)`. Na linha seguinte falaremos que o aluno selecionado deve ser deletado, 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
    MenuItem deletar = menu.add("Deletar");
    deletar.setOnMenuItemClickListener(new MenuItem.OnMenuItemClickListener() {
        @Override
        public boolean onMenuItemClick(MenuItem item) {
            AdapterView.AdapterContextMenuInfo info = (AdapterView.AdapterContextMenuInfo) r
            Aluno aluno = (Aluno) listaAlunos.getItemAtPosition(info.position);

            AlunoDao dao = new AlunoDao(ListaAlunosActivity.this);
            dao.deleta(aluno);
            dao.close();

            return false;
        }
    });
}
```

Falta implementar o "Deleta", então, damos um "Alt+Enter" e "Create method". Iremos para a aba *AlunoDAO* e o novo método aparecerá depois do `while`. Poderíamos fazer a operação a mão, mas vamos pedir para que o `SQLiteOpenHelper` faça isso por nós. Vamos pegar uma referência do banco de dados e digitá-la na linha de baixo do `public void deleta(Aluno aluno)`. Vamos escrever o seguinte: `SQLiteDatabase db = getWritableDatabase()`. O `getWritableDatabase` serve para manipular o banco de dados.

Embaixo disso, chamamos o método `db.delete` que pedirá como parâmetros, a tabela e a cláusula *where*. Preenchemos também com todos os alunos que desejamos remover, cujo `id` seja `getId`. Teremos `db.delete("Aluno", "id = " + aluno.getId())`.

Observem que de novo estamos concatenando. Mas, sempre que quisermos concatenar algum valor no parâmetro, ao em vez de fazer isso, deixaremos um espaço reservado. Vamos marcar o espaço com uma interrogação e informaremos na frente da interrogação o parâmetro, digitaremos `params`. O `params` pede uma *Array* de *String*, então, adicionaremos na linha anterior o que o `params` pede, que são os valores. Acrescentaremos o `Id` do aluno e lembrando que como o `Id` é uma *Long*, ele pede conversão para *String*. Teremos `String[] params = {String.valueOf(aluno.getId())}`. No todo teremos:

```
//...

public void deleta(Aluno aluno) {
    SQLiteDatabase db = getWritableDatabase();

    String [] params = {String.valueOf(aluno.getId())};
    db.delete("Alunos", "id = ?", params);
}
```

Rodando o emulador, vemos que o aluno só é deletado quando saímos da aplicação.

Vamos tentar entender!

Quando selecionamos a opção de "Deletar", observe que a *activity* não sai do primeiro plano, ela fica aparente, no plano de fundo do botão "Deletar". A *activity* continua a rodar e por isso a ação não passa pelo `onResume` do `AlunoDAO.java`. Assim, a lista não é carregada. 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, FormularioAtividade.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();
}

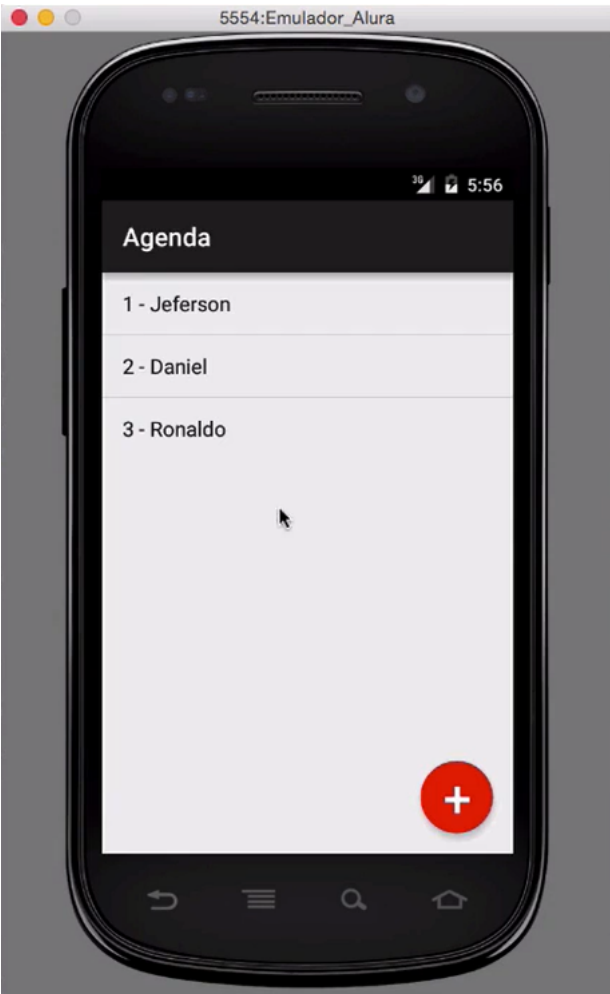
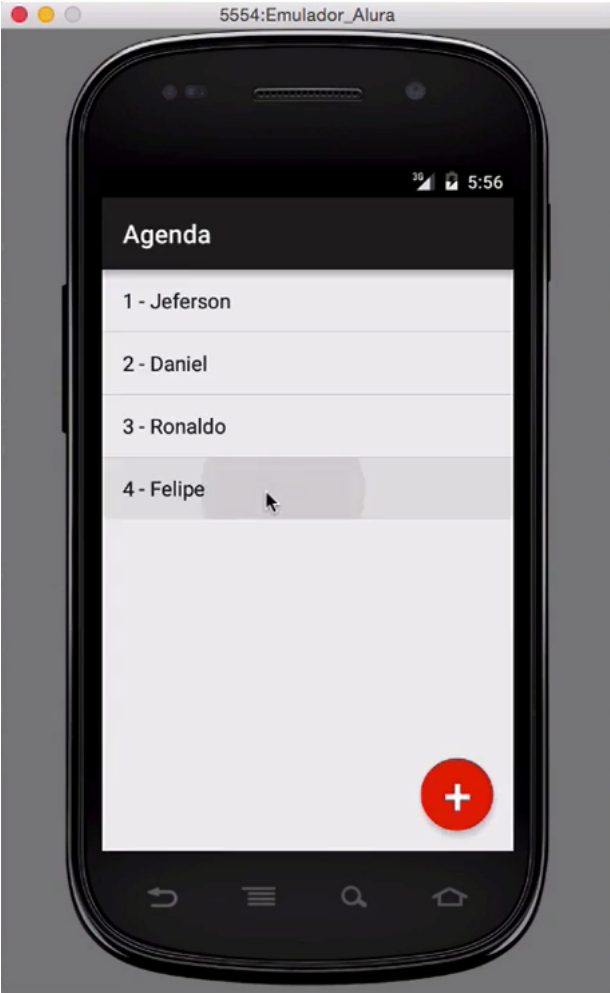
@Override
public void onCreateContextMenu(ContextMenu menu, View v, final ContextMenu.ContextMenuInfo menuInfo) {
    MenuItem deletar = menu.add("Deletar");
    deletar.setOnMenuItemClickListener(new MenuItem.OnMenuItemClickListener() {
        @Override
        public boolean onMenuItemClick(MenuItem 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 o emulador e verificar se as alterações funcionaram! Vamos selecionar na tela o aluno Felipe para que ele seja deletado:



Pronto! Consequimos deletar os alunos!