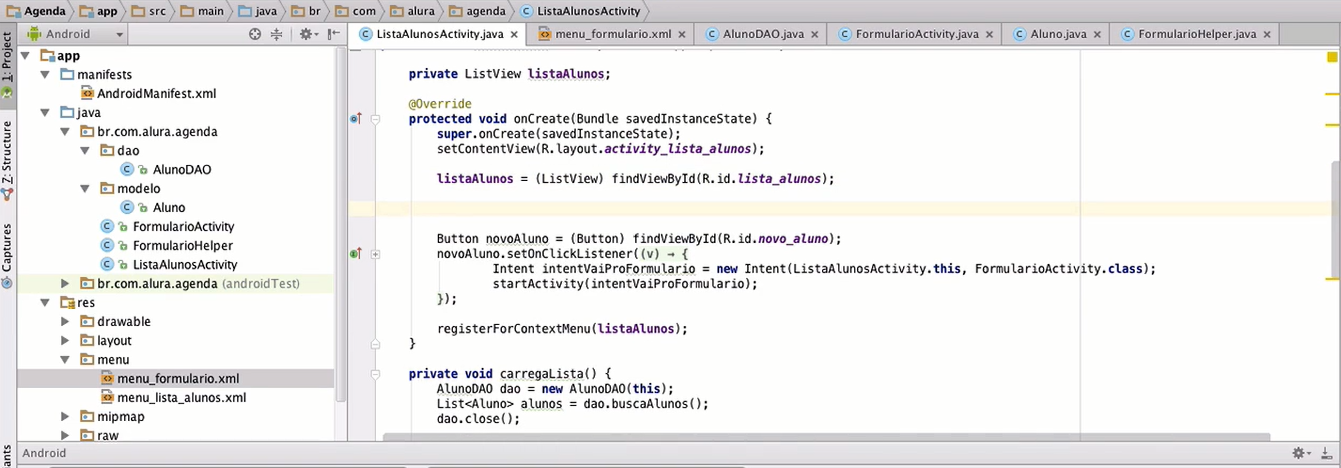
Tratando clique na lista

Agora já conseguimos inserir e remover os alunos. Vamos inserir também algum meio para editar os alunos. Assim, caso acrescentarmos uma informação errada podemos modificá-la quando quisermos.

Poderíamos inserir um novo MenuContext para abrir a opção de "Editar" junto a de "Deletar". Mas, vamos ver algo novo para a nossa aplicação. Vamos iniciar o caminho para clicar no nome do aluno e voltar para o formulário para editarmos.

Se queremos aplicar um comportamento ao clique no nome do aluno vamos voltar na ListaAlunosActivity.java. Vamos na onCreateonde acabamos de recuperar nossa lista.

Encontraremos nossa ListaAlunosActivity.java da seguinte maneira:



Poderíamos inserir um setOnClickListener, como já fizemos anteriormente mas, na verdade o que ele nos devolveria seria uma View que nesse caso é a lista inteira. O que queremos na verdade são os subcomponentes da lista, apenas os nomes dos alunos que clicarmos, por isso não usaremos o setOnClickListener.

Usaremos um Listener diferente. O setOnItemClickListener que manifesta que estamos interessados em um item clicado na lista e não na lista inteira. Teremos que completar com new OnItemClick e ficaremos com listaAlunos.setOnItemClickListener(new OnItemClick) e agora é só dar um "Enter" e criar a classe, ela será anônima.

Teremos o seguinte:

@Override   
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_lista\_alunos);   
  
 listaAlunos = (ListView) findViewById(R.id.lista\_alunos);  
  
 listaAlunos.setOnItemClickListener(new AdapterView.OnItemClickListener() {   
 @Override  
 public void onItemClick(AdapterView<?> parent, View view, int position, long id) {   
 }   
 });  
 }

Agora, iremos implementar o método onItemClick. Aproveitando para explicar algumas coisas. O parent é justamente a lista na qual clicamos, vamos alterar o nome por algo mais claro, como lista, a View é o item que clicamos, então trocaremos por View item, o ´int position´ é a posição do item e o long id é o id do item.

Para poder pegar o aluno basta usar o método que já tínhamos utilizado no onMenuItem, o getItemAtPosition, onde nós só informamos a posição e ele nós diz qual é o aluno.

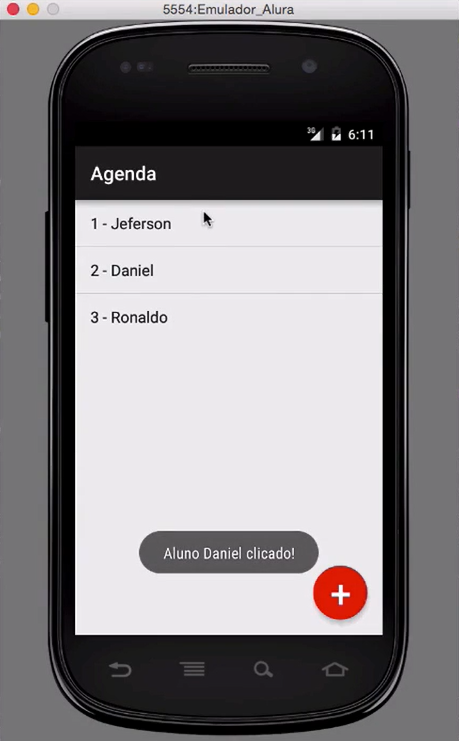
Vamos acrescentar no setOnItemClickListener, na linha de baixo do onItemClickum Aluno aluno = listaAlunos.getItemAtPosition(position) e fazemos um *Cast*.

Para imprimir o nome do aluno podemos acrescentar um Toast seguido do contexto e do nome do aluno clicado, a duração e o show para mostrar. Teremos o Toast.makeText(ListaAlunosActivity.this, "Aluno " + aluno.getNome() + " clicado!, Toast.LENGTH\_SHORT).show()".

Ficaremos com:

@Override  
public void onItemClick(AdapterView<?> parent, View item, int position, long id) {   
 Aluno aluno = (Aluno) listaAlunos.getItemAtPosition(position);  
 Toast.makeText(ListaAlunosActivity.this, "Aluno " + aluno.getNome() + " clicado!", Toast.LENGTH\_SHORT).show();  
}

Vamos rodar a aplicação e ver como ficou!



Agora nosso clique no aluno possui uma interação!

**Observação:** Podemos usar também um outro tipo de *Listener*, o setOnItemLongClick, que está interessado em um clique longo na lista. Se você quiser criar umsetOnItemLongClick basta se guiar nos parâmetros que já fizemos, serão os mesmos. As únicas coisas que mudarão é o public boolean e o return false. Que ao rodar a aplicação veremos que mostrará o Toast do clique longo e o MenuContext.