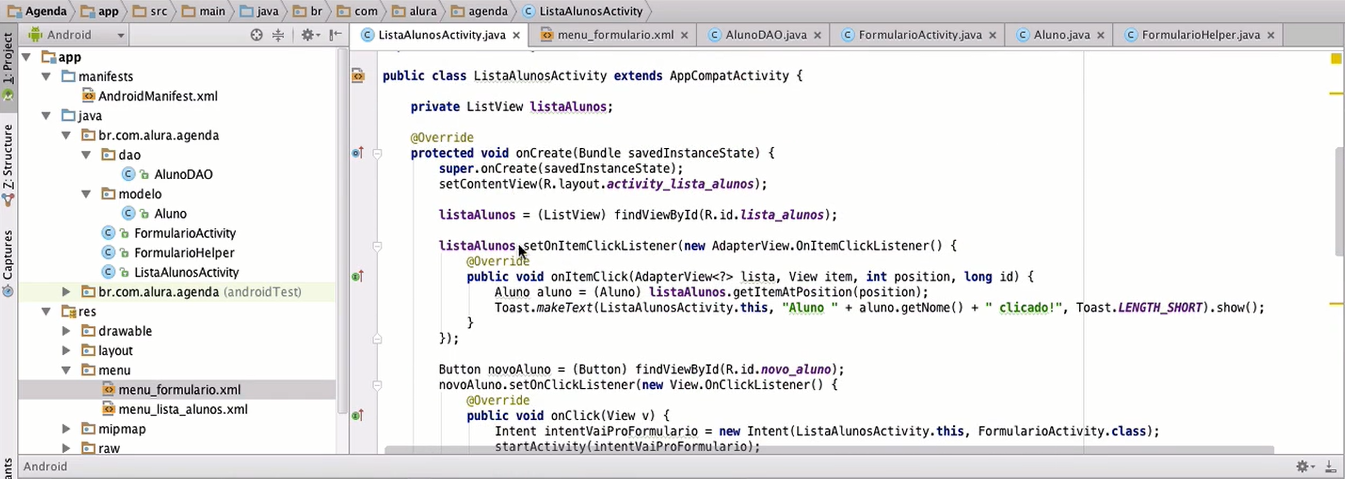
Passando dados entre Activities

Bom, já conseguimos estabelecer uma interação quando o aluno é clicado. O que queremos é que depois de clicarmos no nome do aluno, que a nossa aplicação nos leve de volta ao formulário para que possamos editar.

Para fazer isso vamos no comportamento da nossa classe, no *ListaAlunosActivity.java*.

Nossa tela estará da seguinte maneira:



Lembrando que inserimos um setOnClickListener e também imprimimos através do *Toast* uma mensagem de que o aluno foi clicado. Vamos apagar a linha do *Toast*, Toast.makeText(ListaAlunosActivity.this, "Aluno " + aluno.getNome() + " clicado!, Toast.LENGHT\_SHORT).show()", pois ela não será mais necessária.

O que queremos é, ao clicar no aluno, ir para a tela do formulário. Para isso usaremos uma intent onde declararemos ao *Android* que queremos mudar de tela. Escreveremos uma intentna última linha da classe setOnItemClickListener. Lembrando que a intent pede duas informações, um contexto, que será ListaAlunosActivity.this e a classe do formulário, FormularioActivity.class. Ficaremos com, Intent intentVAiProFormulario = new Intent(ListaAlunosActivity.this, FormularioActivity.class). Para abrir a tela de fato completaremos com um startActivity. Teremos `startActivity(intentVaiProFormulario).

Nosso código estará da seguinte maneira:

@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 onItemCIick(AdapterView<?> lista, View item, int position, long id) {  
 Aluno aluno = (Aluno) listaAlunos.getItemAtPosition(position);  
 Intent intentVaiProFormulario = new Intent(ListaAlunosActivity.this, FormularioActivity.class); startActivity(intentVaiProFormulario);  
 }  
 });  
...//   
 }

Vamos ver o que acontece no emulador?



O formulário abriu mas, falta recuperar os dados do aluno que clicamos nesse formulário.

Vamos voltar na *ListaAlunosActivity.java*, nós já recuperamos o aluno em um momento anterior, o Aluno aluno é isso. Agora, precisamos transmitir ele para o *FormularioActivity.java*.

A intent é usada para demostrar a nossa intenção. Ao criarmos a intent o *Android* abre ela e interpreta que a *ListaAlunosActivity* pede para mudar de lugar, isto é, que deseja ser transposta para ir ao *FormularioActivity*. Feita essa leitura o *Android fecha a intent e instância o* FormularioActivity*. Se conseguirmos colocar algo na intent, como ela vai para o* Android*, e depois para o* FormularioActivity\* conseguiremos recuperar o que penduramos.

Vamos pendurar o aluno na intent e assim conseguiremos passar ele para o *FormularioActivity*.

Para isso, na linha de baixo de Intent intentVaiProFormulario digitaremos intentVaiProForulario e inserimos um método, putExtra. Dentro dos parênteses vamos colocar o que queremos que a nossa intent leve para o *FormularioActivity*, no caso o aluno e também digitaremos uma identificação, uma espécie de etiqueta, "aluno". Ficaremos com, intentVaiProFormulario.putExtra("aluno", aluno).

Ao todo teremos:

@Override   
public void onItemCIick(AdapterView<?> lista, View item, int position, long id) {  
 Aluno aluno = (Aluno) listaAlunos.getItemAtPosition(position);  
 Intent intentVaiProFormuIario = new Intent(ListaAlunosActivity.this, FormularioActivity.class);  
 intentVaiProFormulario.putExtra("aluno", aluno) startActivity(intentVaiProFormulario);  
}

Perceba que o aluno que inserimos continua com problemas, o *Android* não consegue reconhecé-lo. Teremos que "serializar" esse aluno, isto é, transformá-lo em binário.

Para tranformar nossa classe em algo que pode ser serializado, vamos na aba *Aluno.java*. Na mesma linha da public class, logo no topo da página, acrescentamos que a classe aluno implementa a *serializable*, teremos, public class Aluno implements Serializable. Teremos que importar o Serializable através de um "Alt+Enter". Agora o *java* saberá como converter as String, o Long e o Double em binário. Encontraremos nossa tela da seguinte maneira:

public class Aluno implements Serializable {  
 private Long id;  
 private String nome;  
 private String endereço;  
 private String telefone;  
 private String site;  
 private Double nota;   
}

Vamos voltar no *ListaAlunosActivity*. Repare que agora mais nada atrapalha o aluno de estar pendurado na intent. Bom, agora fizemos a parte de enviar o aluno para o *FormularioActivity.java*. Vamos partir para a parte do quando os dados já estão lá.

Voltamos na aba FormularioActivity.java. No momento em que estamos criando nosso formulário, na onCreate se algo estiver pendurado na intent teremos que recuperar essa intent e popular isso na OnCreate. Para fazer isso vamos recuperar a Intent, digitando Intent intent = getIntent. Vamos importar isso através do "Alt+Enter". Agora, vamos recuperar o que está pendurado na intent, no caso o aluno. Para isso, digitaremos intent.getExtra e temos vários tipos de getExtra, usaremos o Serializable. Ficaremos com, intent.getSerializableExtra e ele vai pedir um nome, vamos usar a identificação, a etiqueta que colocamos anteriormente, digitaremos "aluno". Como o getSerializableExtra nos devolve o aluno, vamos falar que Aluno aluno é = a intent.getSerializableExtra. Damos um "Alt+Enter" para fazer o "Cast" do aluno.

Falta trazer o conteúdo do aluno e jogar na nossa tela do formulário. Trazemos o aluno da intent que está no ListaAlunos.Activity e temos que lembrar de verificar se esse aluno é diferente de nulo e se ele for nulo, teremos que preencher o formulário. Para fazer essa verificação acrescentamos if (aluno !=null).

Quando queríamos pegar os dados do formulário e o aluno que era representado pelos dados utilizamos o Helper, por isso, vamos utilizá-lo, novamente, aqui. Vamos dar um "Ctrl+C" e um "Ctrl+V" na linha helper = new FormularioHelper(this); e deslocá-la para dentro das chaves, abaixo da linha do setContentView.

Abaixo da linha if (aluno !=null) digitamos helper.preencheFormulario e completamos com aluno. Ficaremos com, helper.preencheFormulario(aluno). Vamos dar um "Alt+Enter" em cima do preencheFormulario e escolhemos "Create Method" e o método será criado ao final da página FormularioHelper.java.

Teremos na nossa *FormularioActivity.java* o seguinte:

@Override   
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState); setContentView(R.layout.activity\_formulario);  
  
 helper = new FormularioHelper(this);  
  
 Intent intent = getIntent();  
 Aluno aluno = (Aluno) intent.getSerializableExtra("aluno");   
 if (aluno !=null) {  
 helper.preencheFormulario(aluno);  
 }  
  
 }

Vamos na aba do *FormularioHelper.java*. Falta, agora, preencher os campos, basta para isso, na linha seguinte, acrescentar campoNome.setText(aluno.getNome()) e assim por diante com os outros dados. Lembrando que a nota recebe um setProgress e também um intValue. Teremos:

public void preencheFormulario(Aluno aluno) {  
campoNome.setText(aluno.getNome());  
campoEndereco.setText(aluno.getEndereco());  
campoTelefone.setText(aluno.getTelefone());  
campoSite.setText(aluno.getSite());  
campoNota.setProgress(aluno.getNota().intValue());  
}

Perceba que, os campos já foram todos introduzidos no FormularioHelper, isto é, estão prontos para uso, o que facilita muito as coisas.

***Atenção para um detalhe!***

Salvamos o nome, o telefone, o site e a nota. Falta salvar também o Id. Se não salvarmos ele em nenhum lugar iremos perder o id, o que nós complicará as coisas, pois, mais adiante precisaremos dele para alterar o banco. Então, o que vamos fazer é salvar o aluno inteiro, criando um atributo inteiro. Digitaremos no topo da página do FormularioHelper.java, após o private final RatingBar campoNota. Um private Aluno aluno e no final do FormularioHelper criaremos, aluno = new Aluno. Agora, podemos apagar no Aluno pegaAluno a linha Aluno aluno = new Aluno, pois ele só precisa estar instanciado uma vez.

Ficaremos no FormularioHelper com:

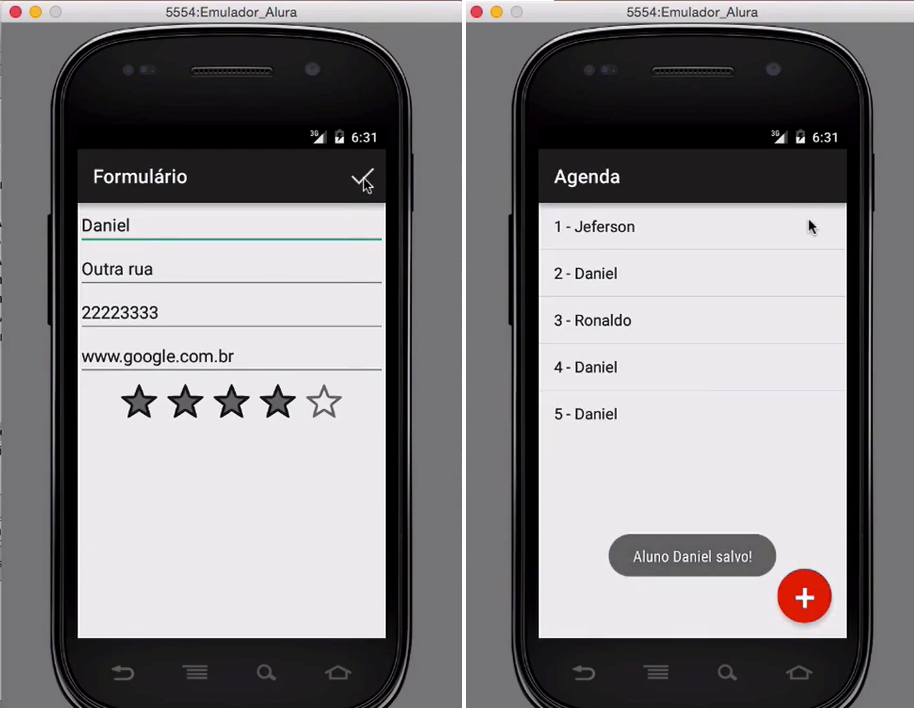
public FormularioHelper(FormularioActivity activity) {  
 campoNome = (EditText) activity.findViewById(R.id.formulario\_nome);  
 campoEndereço = (EditText) activity.findViewById(R.id.formulario\_endereço);  
 campoTelefone = (EditText) activity.findViewById(R.id.formulario\_Telefone);  
 campoNota = (RatingBar) activity.findViewById(R.id.formulario\_nota);  
 aluno = new Aluno();  
}

Agora, no preencheFormulario vamos guardar o aluno inteiro no atributo. Digitaremos ao final dele this.aluno =aluno. Ficaremos com:

public void preencheFormulario(Aluno aluno) {  
 campoNome.setText(aluno.getNome());  
 campoEndereco.setText(aluno.getEndereco());  
 campoTelefone.setText(aluno.getTelefone());  
 campoSite.setText(aluno.getSite());  
 campoNota.setProgress(aluno.getNota().intValue());  
 this.aluno = aluno;  
 }

Como salvamos o aluno no preencheFormulario é daí que vai advir o id.

Vamos ver se isso está funcionando no nosso código! Vamos salvar e rodar o emulador.



Conseguimos trazer para o formulário todas as informações de "Daniel", mas repare que se alterarmos e salvarmos o formulário estaremos apenas acrescentando mais "Daniels" na nossa lista.

Isso resolveremos na sequência!