

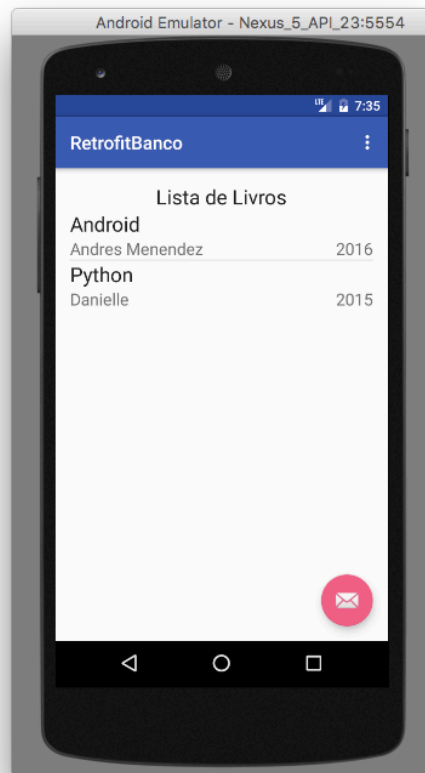
PROGRAMAÇÃO ANDROID

App Retrofit Completo

Universidade Federal de Sergipe
Departamento de Sistemas de Informação
Prof. Andrés Menéndez
ammenendez@gmail.com

Aplicativo com Retrofit

- Vamos fazer um app com as mesmas funcionalidades que foram feitas tanto com SQLite quando com Realm



Interface Retrofit

- Antes de começar vamos criar a interface Retrofit
- Por enquanto definimos somente o método getLivros

```
public interface ILivrosREST {
```

```
    @GET("Livros/webresources/br.ufs.tep.livros")
```

```
    Call<List<Livro>> getLivros();
```

```
    public static final Retrofit retrofit = new Retrofit.Builder()
```

```
        .baseUrl("http://10.0.2.2:8080/")
```

```
        .addConverterFactory(GsonConverterFactory.create())
```

```
        .build();
```

```
}
```

Interface Retrofit

- Um detalhe importante: normalmente usamos localhost quando estamos querendo acessar nossa máquina
- Contudo, no Android é diferente
- No Android temos que usar 10.0.2.2 como IP da nossa máquina
- Veja que baseUrl ficou assim:
 - `baseUrl("http://10.0.2.2:8080/")`

AVISO

- Não se esqueça de
- Dar permissão de internet do arquivo de manifesto
- Incluir as linhas do Retrofit no Gradle para inserir as dependências no projeto

Aplicativo com Retrofit

- Inicialmente temos que fazer a **activity principal** que vai mostrar a lista de Livros que estão cadastradas no postgres
- Para isso faremos um ListView customizado, sendo que teremos que criar:
 - O arquivo de layout
 - O ArrayAdapter
- Além disso teremos que programar a activity para ir buscar os dados

Arquivo de layout

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">

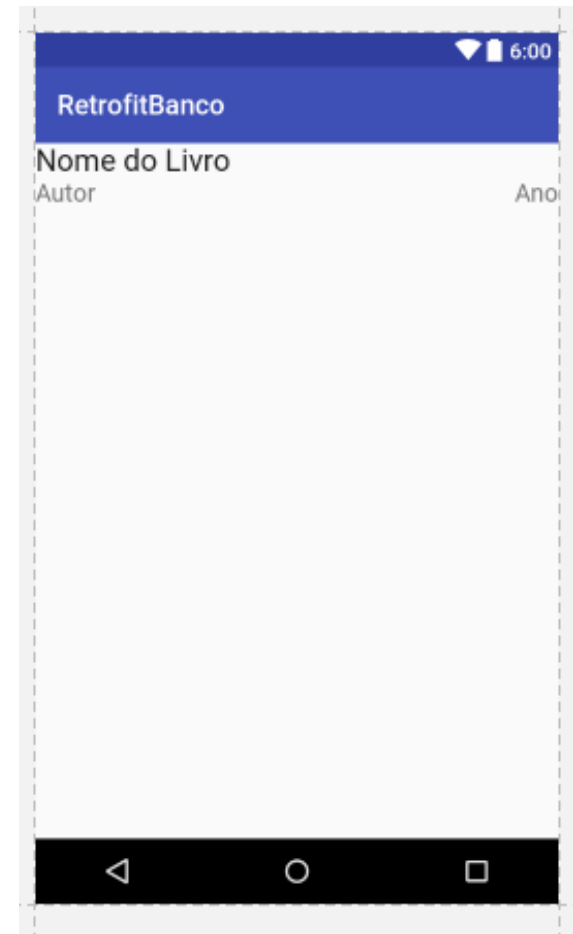
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Nome do Livro"
        android:id="@+id/txtNome" />

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceMedium"
            android:text="Autor"
            android:id="@+id/txtAutor"
            android:layout_weight="4" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceMedium"
            android:text="Ano"
            android:id="@+id/txtAno"
            android:layout_weight="1"
            android:textAlignment="textEnd" />

    </LinearLayout>
</LinearLayout>
```



ArrayAdapter

```
public class LivroAdapter extends ArrayAdapter<Livro> {  
    private final Context context;  
    private final List<Livro> elementos;  
  
    public LivroAdapter(Context context, List<Livro> elementos) {  
        super(context, R.layout.linha, elementos);  
        this.context = context;  
        this.elementos = elementos;  
    }  
}
```


ArrayAdapter

@Override

```
public View getView(int position, View convertView, ViewGroup parent) {
```

```
    LayoutInflater inflater = (LayoutInflater) context  
        .getSystemService(Context.LAYOUT_INFLATER_SERVICE);
```

```
    View rowView = inflater.inflate(R.layout.linha, parent, false);
```

```
    TextView titulo = (TextView) rowView.findViewById(R.id.txtNome);
```

```
    TextView ano = (TextView) rowView.findViewById(R.id.txtAno);
```

```
    TextView autor = (TextView) rowView.findViewById(R.id.txtAutor);
```

```
    titulo.setText(elementos.get(position).getNome());
```

```
    autor.setText(elementos.get(position).getAutor());
```

```
    ano.setText(Integer.toString(elementos.get(position).getAno()));
```

```
    return rowView;
```

```
}
```

```
}
```

Layout da Activity

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
app:layout_behavior="@string/appbar_scrolling_view_behavior"
tools:context="br.ufs.tep.retrofitbanco.MainActivity"
tools:showIn="@layout/activity_main">
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="Lista de Livros"
    android:id="@+id/textView"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true" />
<ListView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/lvLivros"
    android:layout_below="@+id/textView"
    android:layout_alignParentStart="true" />
</RelativeLayout>
```



Programação da activity

- Vejamos agora a programação da activity
- dialog é um atributo da classe do tipo ProgressDialog

@Override

```
protected void onStart() {  
    super.onStart();  
    final ListView lista = (ListView) findViewById(R.id.lvLivros);  
    ILivrosREST iLivrosREST = ILivrosREST.retrofit  
                                                .create(ILivrosREST.class);  
    dialog = new ProgressDialog(MainActivity.this);  
    dialog.setMessage("Carregando...");  
    dialog.setCancelable(false);  
    dialog.show();  
    final Call<List<Livro>> call = iLivrosREST.getLivros();  
}
```

Programação da activity

```
call.enqueue(new Callback<List<Livro>>() {  
    @Override  
    public void onResponse(Call<List<Livro>> call,  
                           Response<List<Livro>> response) {  
        if (dialog.isShowing())  
            dialog.dismiss();  
        final List<Livro> listaLivros = response.body();  
        if (listaLivros != null) {  
            LivroAdapter adapter = new LivroAdapter(getBaseContext(), listaLivros);  
            lista.setAdapter(adapter);  
        }  
    }  
})
```

Programação da activity

@Override

public void onFailure(Call<List<Livro>> call, Throwable t) {

if (**dialog**.isShowing())

dialog.dismiss();

 Toast.makeText(getBaseContext(), "**Problema de acesso**",

 Toast.**LENGTH_SHORT**).show();

 }

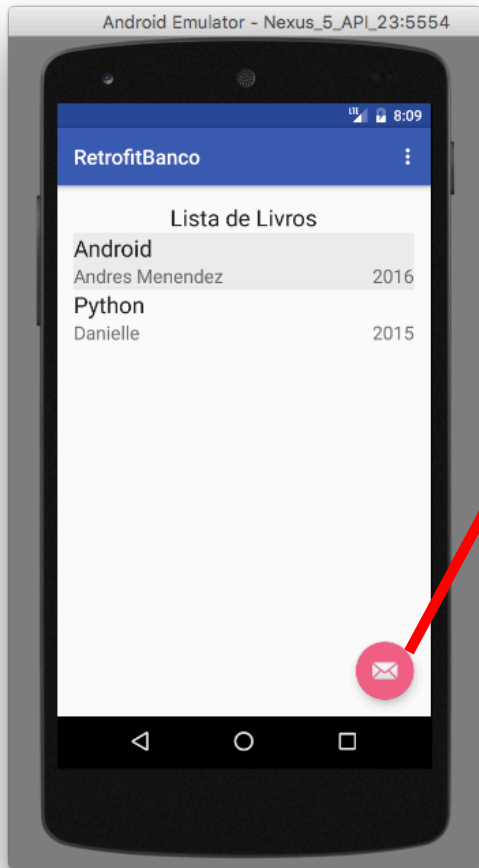
});

}

Adicionar um Livro

- Para adicionar um livro teremos que
 - Disparar a activity de AddLivro a partir da activity principal
 - Criar o layout
 - Programar a activity

Invocar a activity



```
FloatingActionButton fab = (FloatingActionButton)
findViewById(R.id.fab);
fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(MainActivity.this, AddLivroActivity.class);
        startActivity(intent);
    }
});
```

Layout da activity

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/activity_add_livro"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
tools:context="br.ufs.tep.retrofitbanco.AddLivroActivity">
```

```
<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/edNome"
    android:layout_alignParentTop="true"
    android:layout_alignParentStart="true"
    android:layout_alignParentEnd="true"
    android:hint="Nome do livro" />
```

```
<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/edAutor"
    android:layout_below="@+id/edNome"
    android:layout_alignParentStart="true"
    android:layout_alignParentEnd="true"
    android:hint="Autor do livro" />
```



Layout da activity

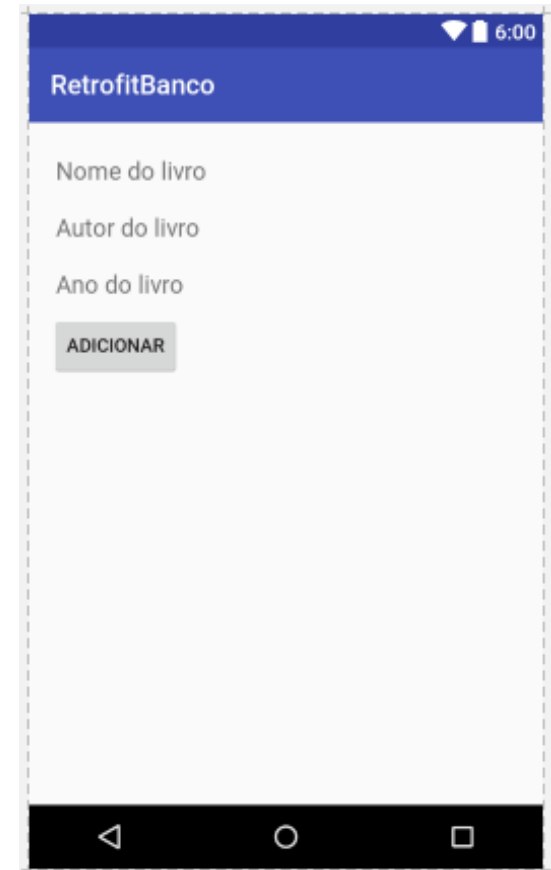
<EditText

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:inputType="number"  
    android:ems="10"  
    android:id="@+id/edAno"  
    android:layout_below="@+id/edAutor"  
    android:layout_alignParentStart="true"  
    android:hint="Ano do livro" />
```

<Button

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Adicionar"  
    android:id="@+id/btnAddLivro"  
    android:layout_below="@+id/edAno"  
    android:layout_alignParentStart="true" />
```

</RelativeLayout>



Programação da activity

- No método onCreate

```
final EditText nome = (EditText) findViewById(R.id.edNome);  
final EditText autor = (EditText) findViewById(R.id.edAutor);  
final EditText ano = (EditText) findViewById(R.id.edAno);  
Button adicionar = (Button) findViewById(R.id.btnAddLivro);  
adicionar.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        dialog = new ProgressDialog(AddLivroActivity.this);  
        dialog.setMessage("Carregando...");  
        dialog.setCancelable(false);  
        dialog.show();  
    }  
});
```

Programação da activity

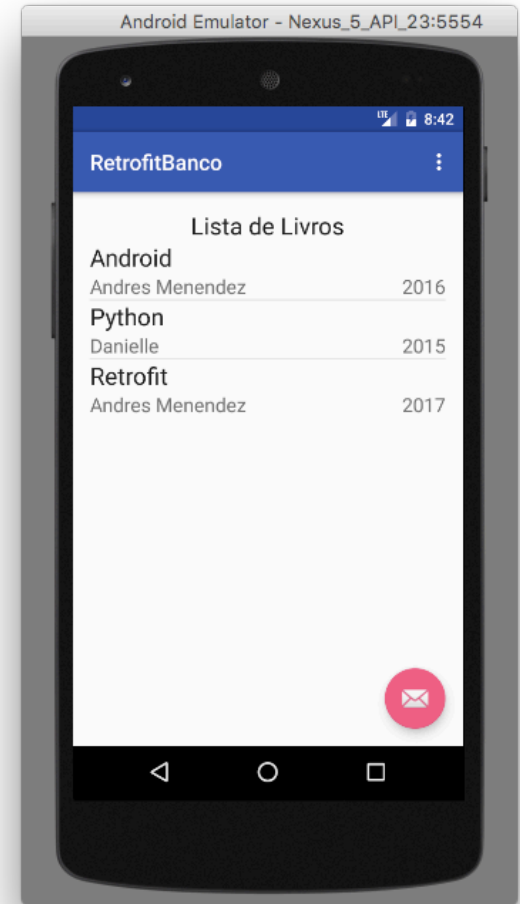
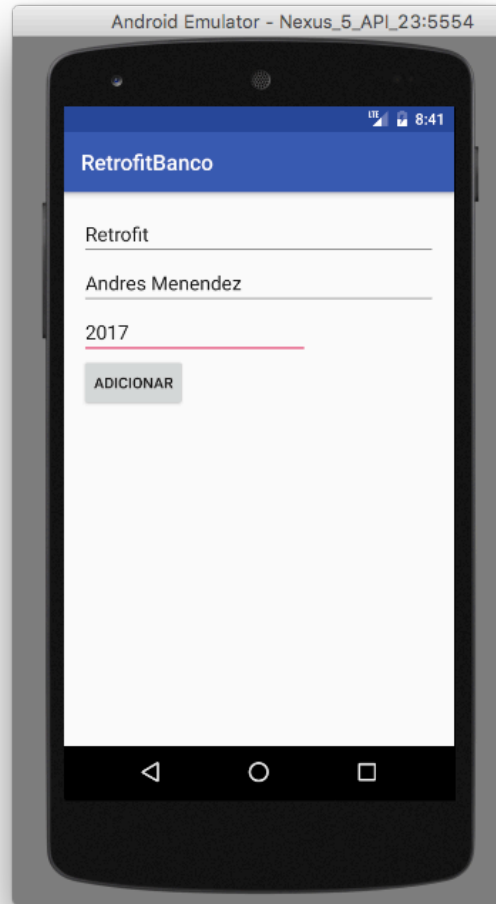
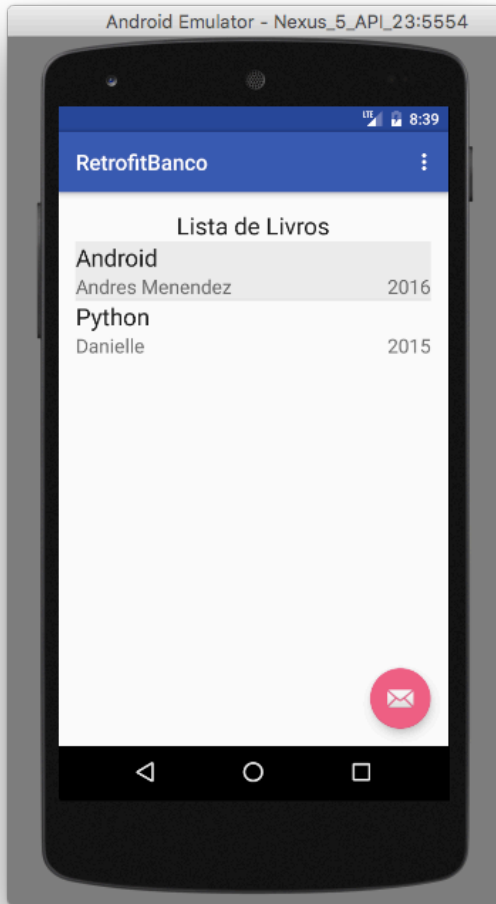
```
Livro livro = new Livro();
livro.setNome(nome.getText().toString());
livro.setAutor(autor.getText().toString());
livro.setAno(Integer.parseInt(ano.getText().toString()));
ILivrosREST iLivrosREST = ILivrosREST.retrofit.create(ILivrosREST.class);
final Call<Void> call = iLivrosREST.inserirPessoa(livro);
call.enqueue(new Callback<Void>() {
    @Override
    public void onResponse(Call<Void> call, Response<Void> response) {
        if (dialog.isShowing())
            dialog.dismiss();
        Toast.makeText(getApplicationContext(), "Livro inserido com sucesso",
            Toast.LENGTH_SHORT).show();
    }
}
```

Programação da activity

@Override

```
public void onFailure(Call<Void> call, Throwable t) {  
    if (dialog.isShowing())  
        dialog.dismiss();  
    Toast.makeText(getBaseContext(),  
        "Não foi possível fazer a conexão",  
        Toast.LENGTH_SHORT).show();  
}  
});  
}
```

Adicionando um livro



Editando e removendo um livro

- Para editar e remover um livro teremos que
 - Alterar a interface do Retrofit para adicionar os novos métodos
 - Passar como parâmetro o id para a activity EditLivro
 - Criar o layout da activity de edição
 - Programar a activity tanto para editar como para excluir

Interface final do Retrofit


```
public interface ILivrosREST {  
    @POST("Livros/webresources/br.ufs.tep.livros/inserir")  
    Call<Void> insereLivro(@Body Livro livro);  
    @GET("Livros/webresources/br.ufs.tep.livros")  
    Call<List<Livro>> getLivros();  
    @GET("Livros/webresources/br.ufs.tep.livros/buscar/{id}")  
    Call<Livro> getLivroPorId(@Path("id") String id);  
    @PUT("Livros/webresources/br.ufs.tep.livros/editar/{id}")  
    Call<Void> alteraLivro(@Path("id") String id, @Body Livro livro);  
    @DELETE("Livros/webresources/br.ufs.tep.livros/remover/{id}")  
    Call<Void> removeLivro(@Path("id") String id);  
  
    public static final Retrofit retrofit = new Retrofit.Builder()  
        .baseUrl("http://10.0.2.2:8080/")  
        .addConverterFactory(GsonConverterFactory.create())  
        .build();  
}
```

Chamando a activity

- Dentro da resposta de retorno da activity principal programamos o onItemClick do ListView

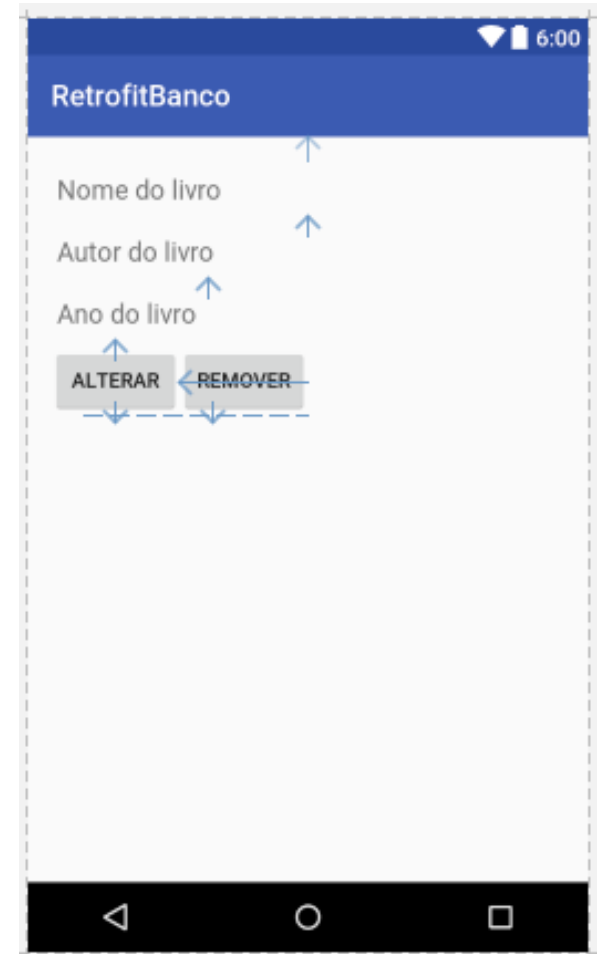
@Override

```
public void onResponse(Call<List<Livro>> call, Response<List<Livro>> response) {  
    if (dialog.isShowing())  
        dialog.dismiss();  
    final List<Livro> listaLivros = response.body();  
    if (listaLivros != null) {  
        LivroAdapter adapter = new LivroAdapter(getBaseContext(), listaLivros);  
        lista.setAdapter(adapter);  
        lista.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
            @Override  
            public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {  
                Intent intent = new Intent(MainActivity.this, EditLivroActivity.class);  
                intent.putExtra("ID", listaLivros.get(i).getId());  
                startActivity(intent);  
            }  
        });  
    }  
}
```



Layout da activity

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_edit_livro"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="br.ufs.tep.retrofitbanco.EditLivroActivity">
    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/edNome"
        android:layout_alignParentTop="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true"
        android:hint="Nome do livro" />
    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/edAutor"
        android:layout_below="@+id/edNome"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true"
        android:hint="Autor do livro" />
```



Layout da activity

<EditText

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:inputType="number"
    android:ems="10"
    android:id="@+id/edAno"
    android:layout_below="@+id/edAutor"
    android:layout_alignParentStart="true"
    android:hint="Ano do livro" />
```

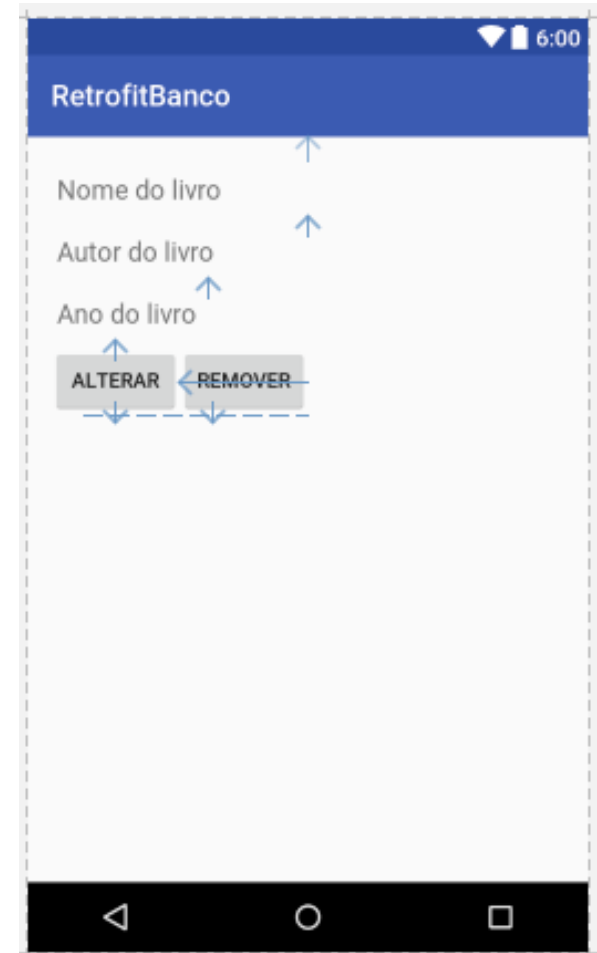
<Button

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Alterar"
    android:id="@+id/btnEditLivro"
    android:layout_below="@+id/edAno"
    android:layout_alignParentStart="true" />
```

<Button

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Remover"
    android:id="@+id/btnDeleteLivro"
    android:layout_alignBottom="@+id/btnEditLivro"
    android:layout_toEndOf="@+id/btnEditLivro" />
```

</RelativeLayout>



Programação da activity

- Por ser uma activity com várias funcionalidades a programação ficou relativamente grande
- Vejamos no método onCreate, como fazer a chamada para recuperar o livro do **id** passado como parâmetro

@Override

[illegible]

Programação da activity

```
final ILivrosREST iLivrosREST = ILivrosREST.retrofit
                                     .create(ILivrosREST.class);

final Call<Livro> call = iLivrosREST.getLivroPorId(id);
dialog = new ProgressDialog(EditLivroActivity.this);
dialog.setMessage("Carregando...");
dialog.setCancelable(false);
dialog.show();
call.enqueue(new Callback<Livro>() {
    @Override
    public void onResponse(Call<Livro> call, Response<Livro> response) {
        if (dialog.isShowing())
            dialog.dismiss();
        Livro livro = response.body();
        nome.setText(livro.getNome());
        autor.setText(livro.getAutor());
        ano.setText(String.valueOf(livro.getAno()));
    }
}
```

Programação da activity

@Override

```
public void onFailure(Call<Livro> call, Throwable t) {  
    if (dialog.isShowing())  
        dialog.dismiss();  
    Toast.makeText(getBaseContext(), "Não foi possível fazer a conexão",  
        Toast.LENGTH_SHORT).show();  
}  
});
```

Programação da activity

- Depois de obter o livro temos que fazer a programação do clique do botão alterar

```
Button alterar = (Button) findViewById(R.id.btnEditLivro);
alterar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        dialog = new ProgressDialog(EditLivroActivity.this);
        dialog.setMessage("Carregando...");
        dialog.setCancelable(false);
        dialog.show();
        Livro livro = new Livro();
        livro.setId(Integer.parseInt(id));
        livro.setNome(nome.getText().toString());
        livro.setAutor(autor.getText().toString());
        livro.setAno(Integer.parseInt(ano.getText().toString()));
        Call<Void> call = iLivrosREST.alteraLivro(id, livro);
```

Programação da activity

```
call.enqueue(new Callback<Void>() {  
    @Override  
    public void onResponse(Call<Void> call, Response<Void> response) {  
        if (dialog.isShowing())  
            dialog.dismiss();  
        Toast.makeText(getBaseContext(), "Livro alterado com sucesso",  
            Toast.LENGTH_SHORT).show();  
    }  
    @Override  
    public void onFailure(Call<Void> call, Throwable t) {  
        if (dialog.isShowing())  
            dialog.dismiss();  
        Toast.makeText(getBaseContext(), "Não foi possível fazer a conexão",  
            Toast.LENGTH_SHORT).show();  
    }  
});  
}
```

Programação da activity

- Finalmente precisa fazer a programação para a remoção de um livro

```
Button remover = (Button) findViewById(R.id.btnDeleteLivro);
remover.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        dialog = new ProgressDialog(EditLivroActivity.this);
        dialog.setMessage("Carregando...");
        dialog.setCancelable(false);
        dialog.show();
        Call<Void> call = iLivrosREST.removeLivro(id);
```


Programação da activity

```
call.enqueue(new Callback<Void>() {  
    @Override  
    public void onResponse(Call<Void> call, Response<Void> response) {  
        if (dialog.isShowing())  
            dialog.dismiss();  
        Toast.makeText(getBaseContext(), "Livro removido com sucesso",  
                        Toast.LENGTH_SHORT).show();  
    }  
    @Override  
    public void onFailure(Call<Void> call, Throwable t) {  
        if (dialog.isShowing())  
            dialog.dismiss();  
        Toast.makeText(getBaseContext(), "Não foi possível fazer a conexão",  
                        Toast.LENGTH_SHORT).show();  
    }  
});  
}
```