

FUNDAÇÃO INSTITUTO DE EDUCAÇÃO DE BARUERI

Instituto Técnico de Barueri “Brasílio Flores de Azevedo”

Tecnologias Emergentes

Tutorial Android para Conexão com Banco de Dados usando Ponte ABD
versão 3.0

Desenvolvido pelo Prof. Alessandro Tacini

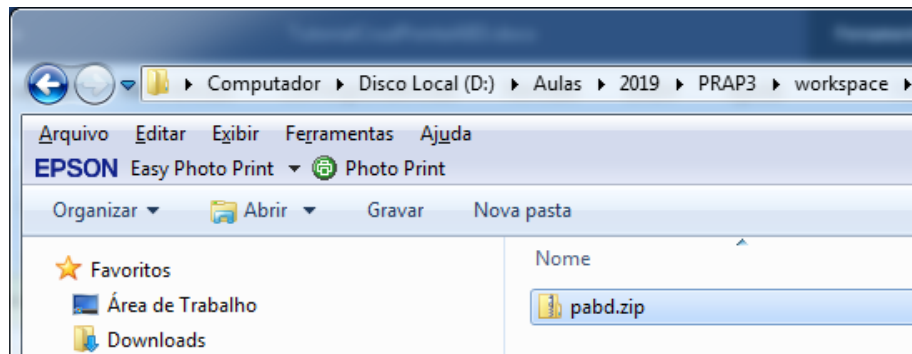
Prof. Adriano Domingues

2019

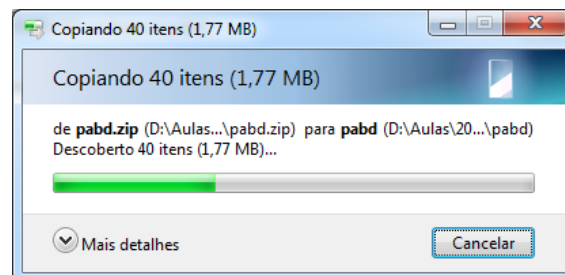
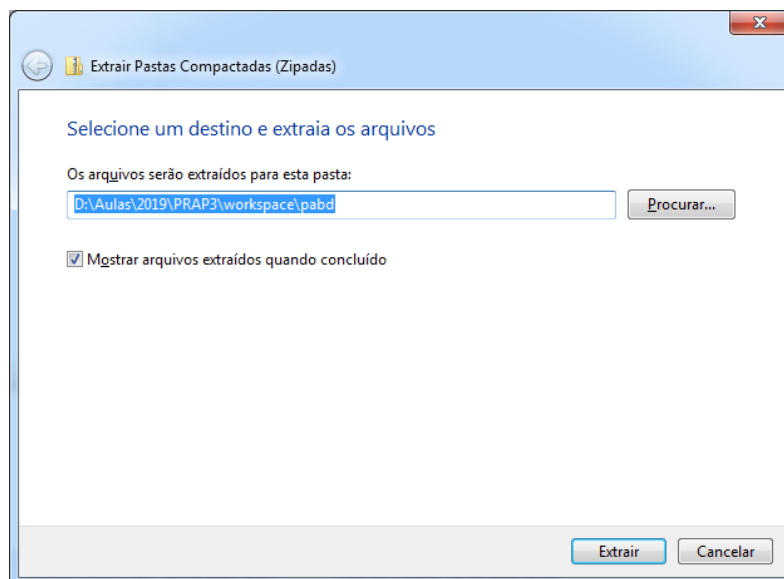
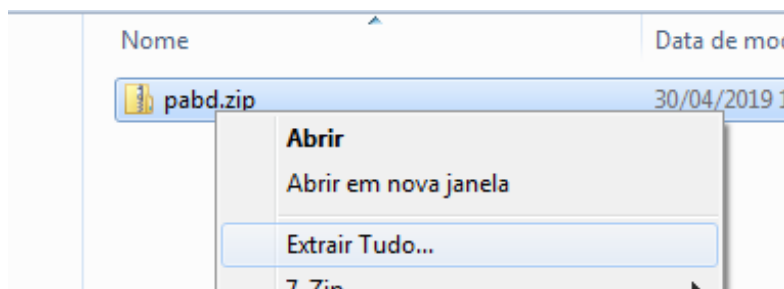
Tutorial Criação CRUD Usando PonteABD

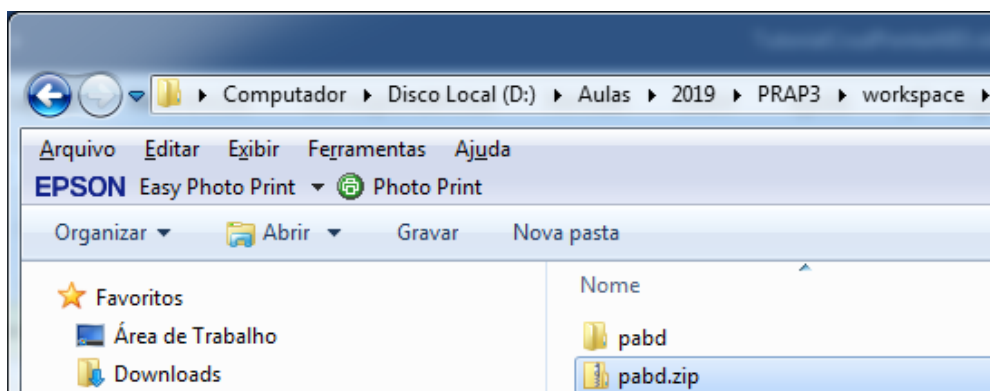
1. Instalar Ponte ABD

1.1. Copiar arquivo ponteabd.zip dentro do seu workspace do Eclipse

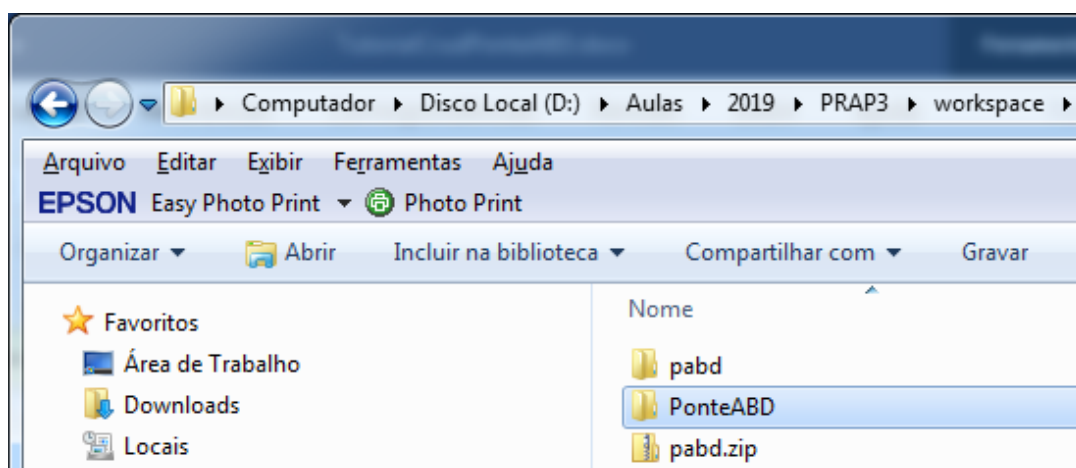
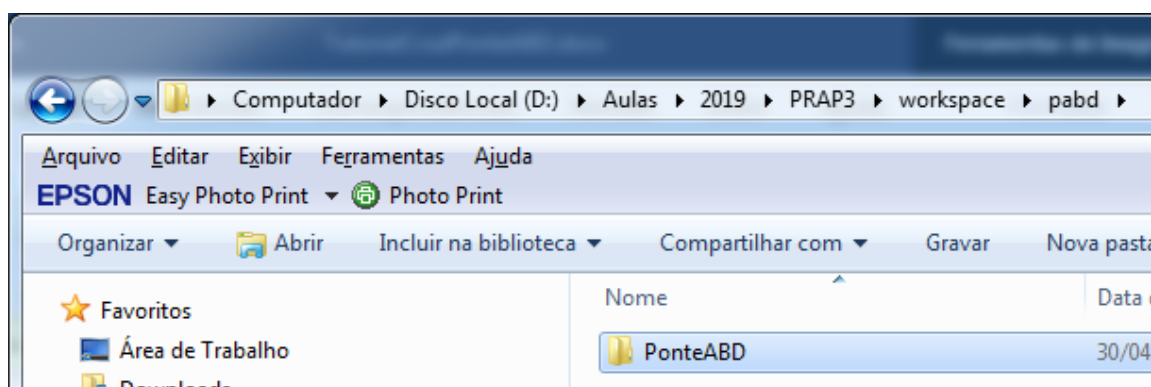


1.2. Descompacte o arquivo

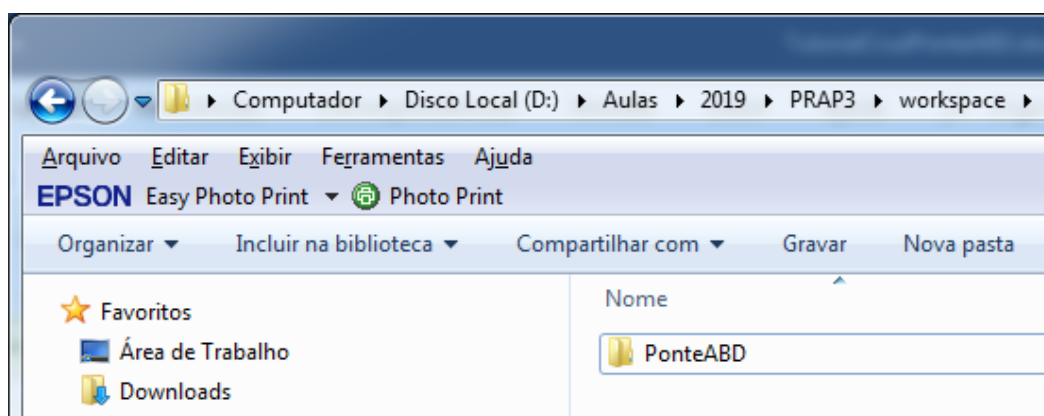


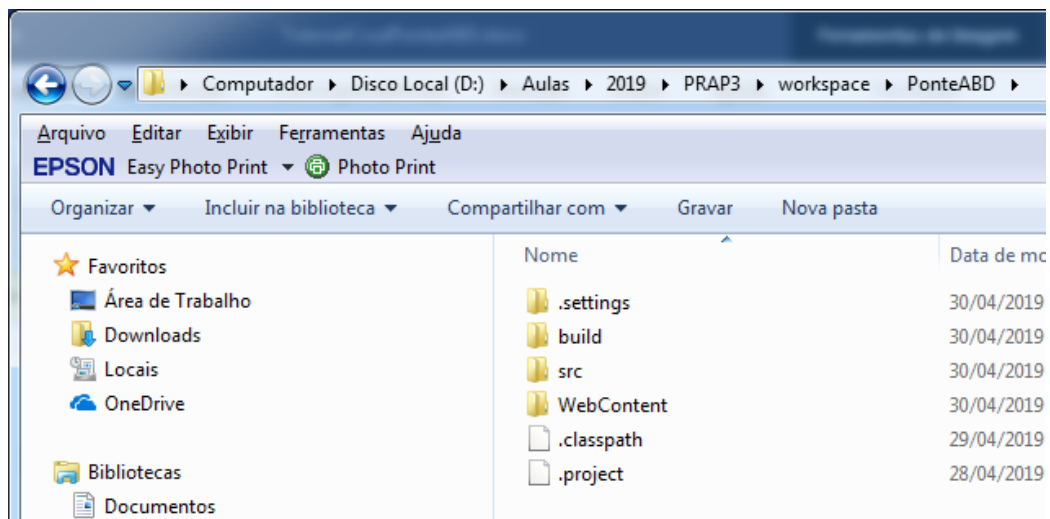


1.3. Mova o diretório PonteABD para dentro de workspace

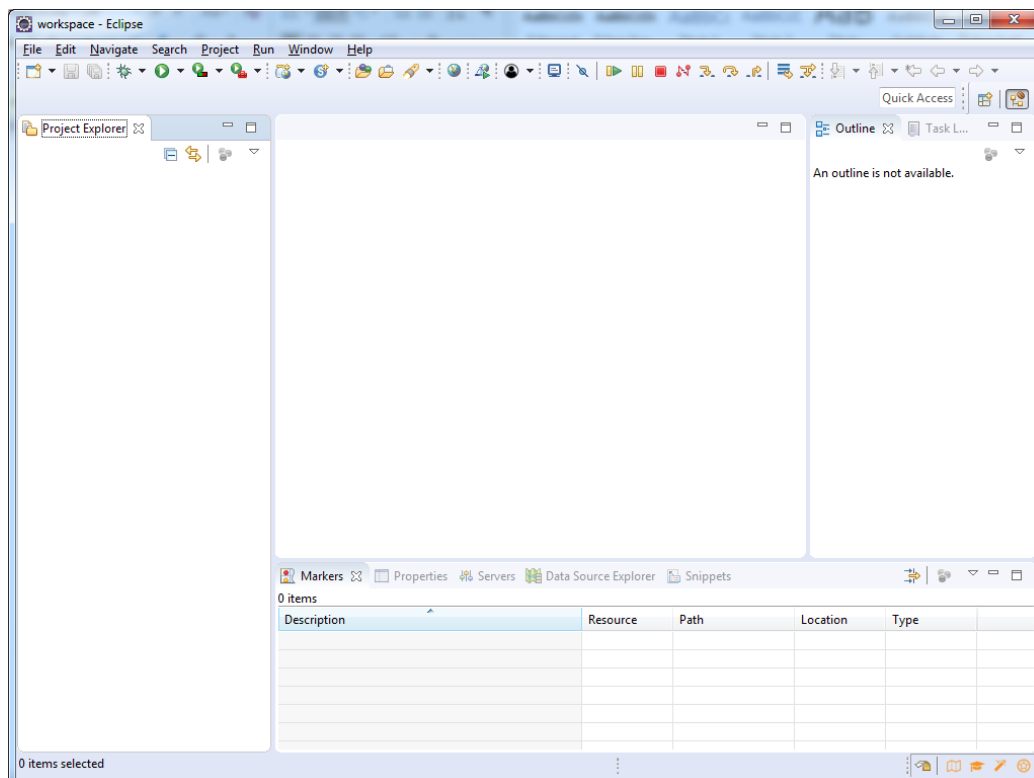
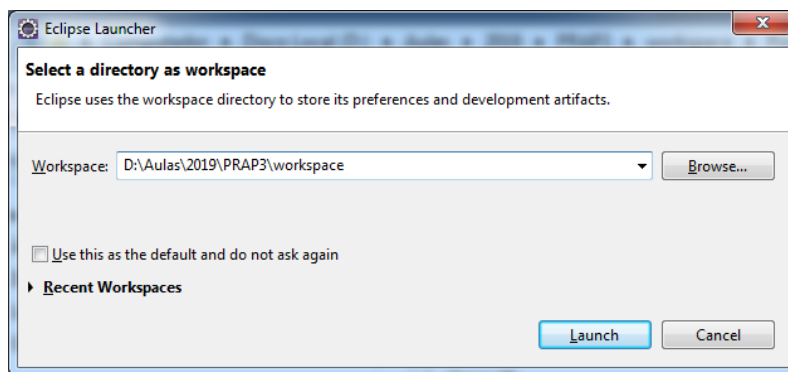


1.4. Apague o diretório e o arquivo zip deixando apenas o diretório PonteABD

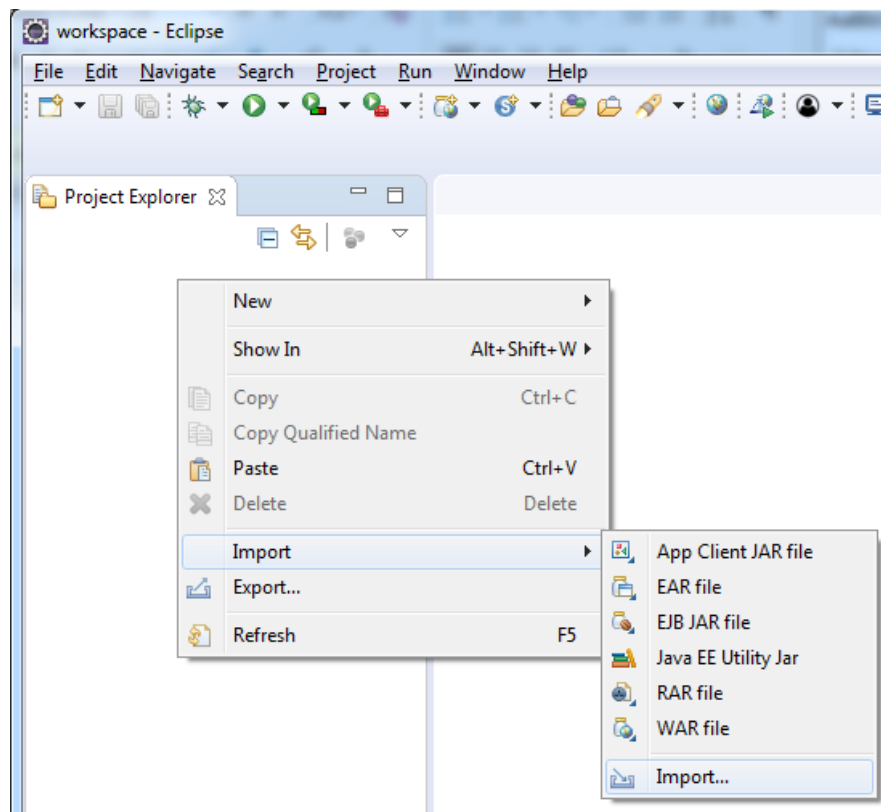




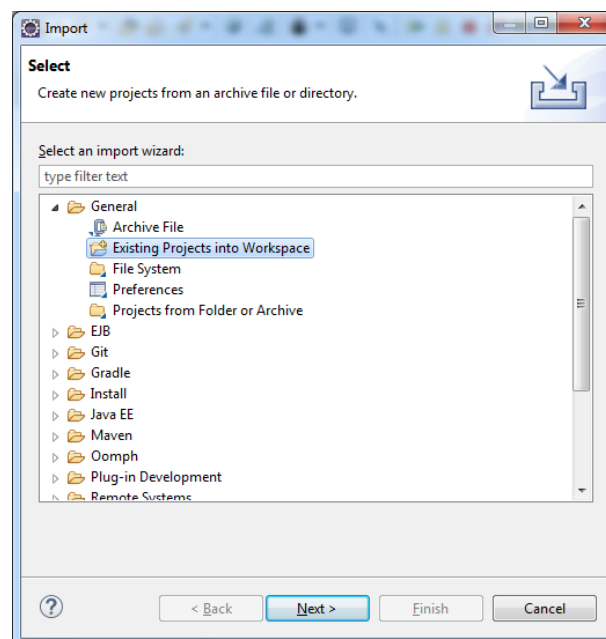
1.5. Abra o Eclipse apontando para este workspace



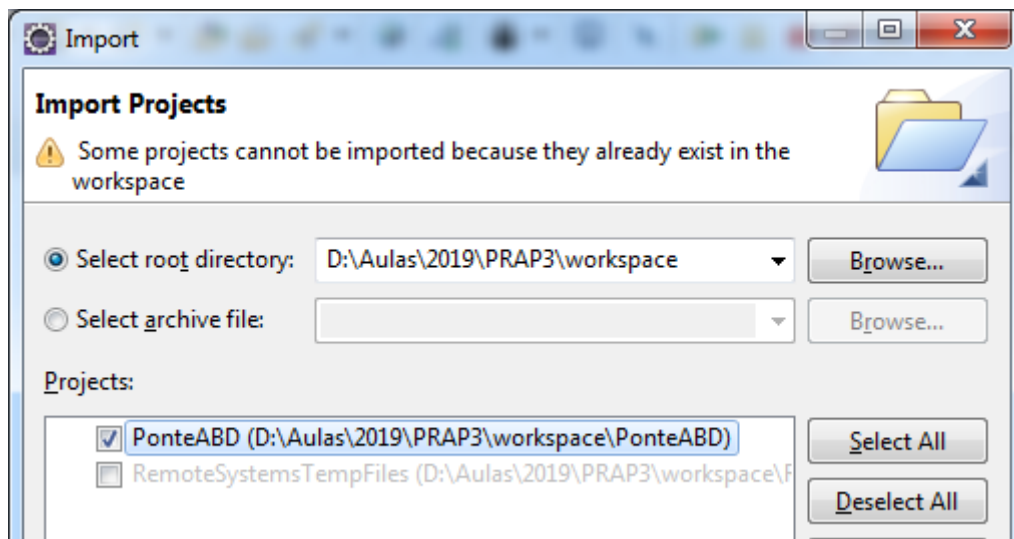
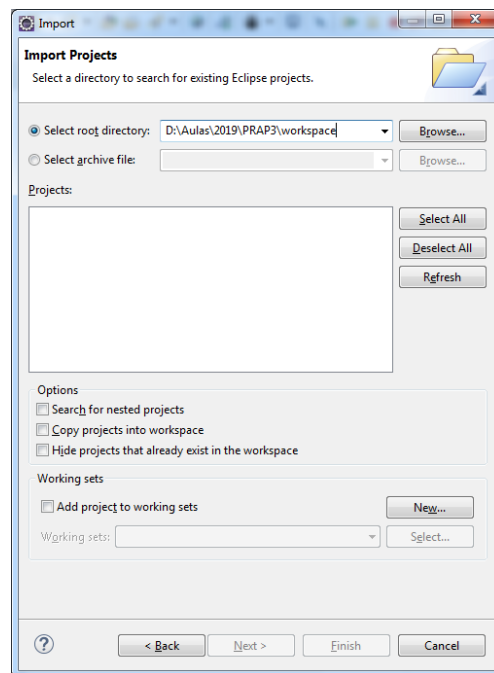
1.6. Na perspectiva “Java EE”, clique com o botão direito do mouse na janela Project Explorer >> Import >> Import...



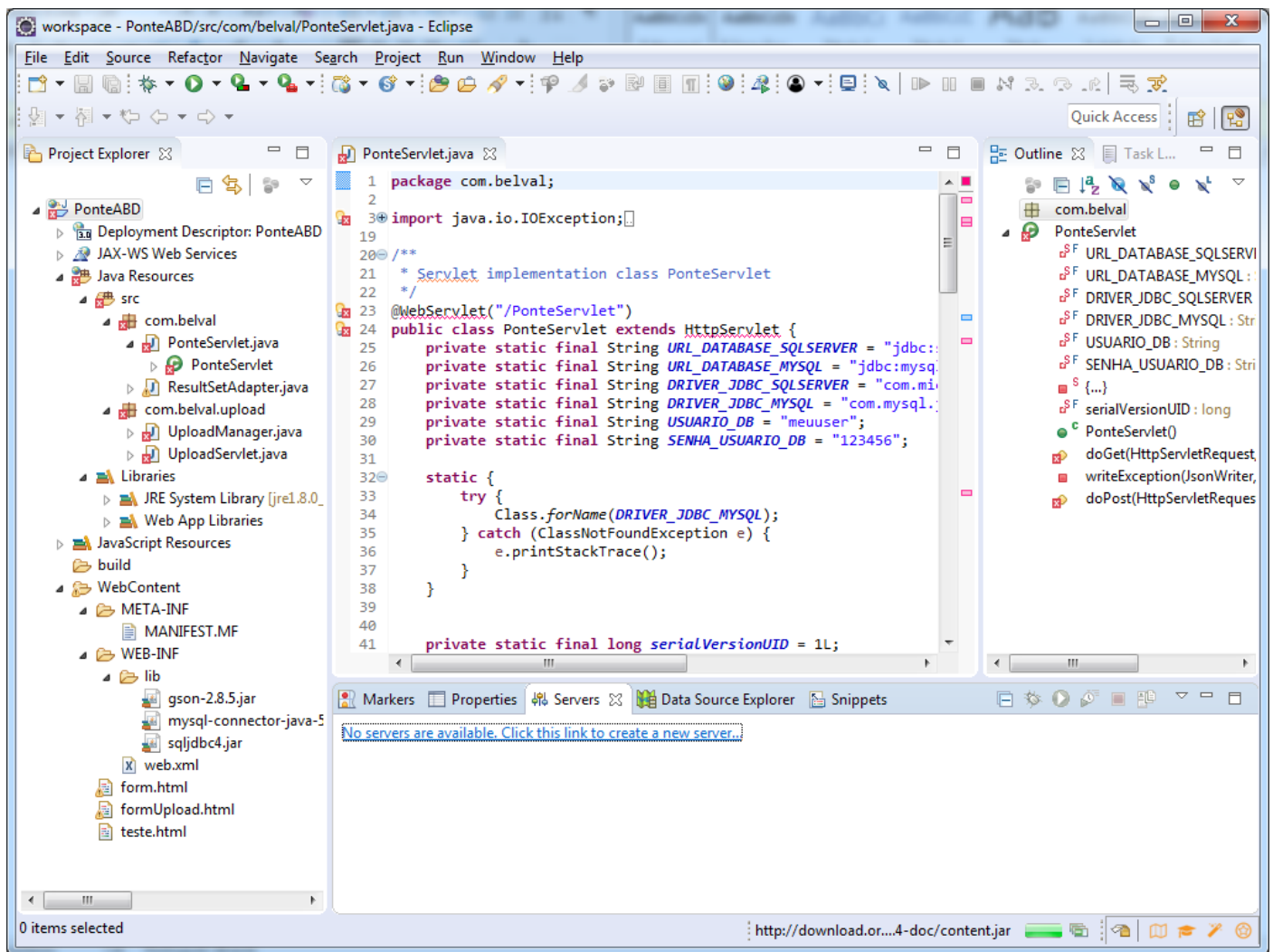
1.7. Escolha a opção General >> Existing Project sinto Workspace e depois clique em Next.



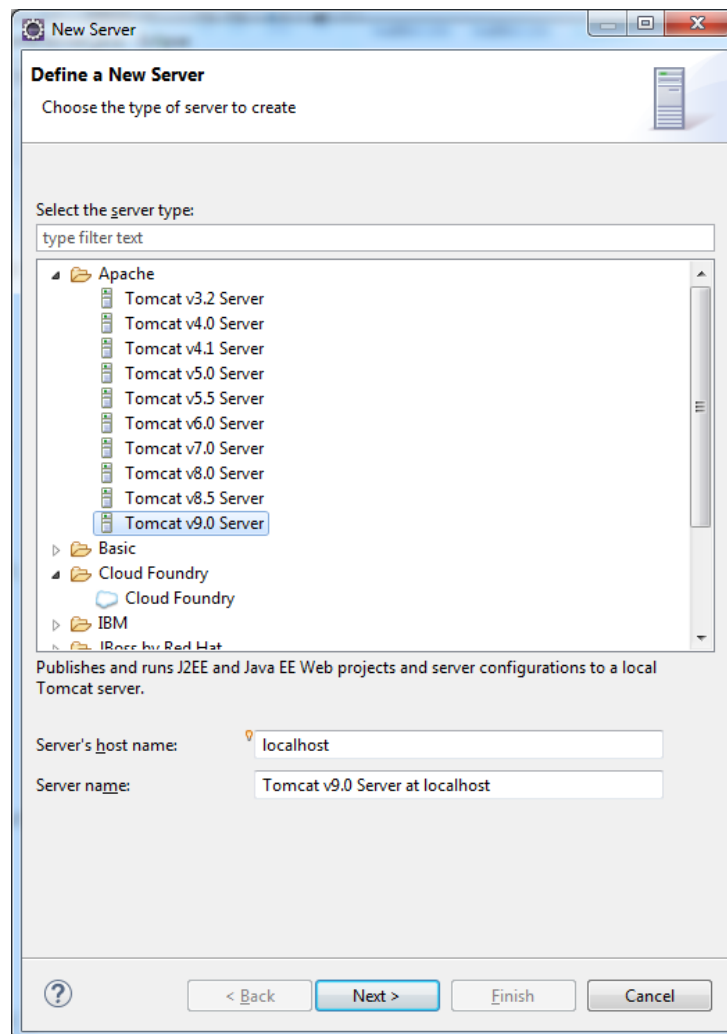
1.8. Cole o caminho do workspace em “Select root directory” e dê um “Tab” para carregar.



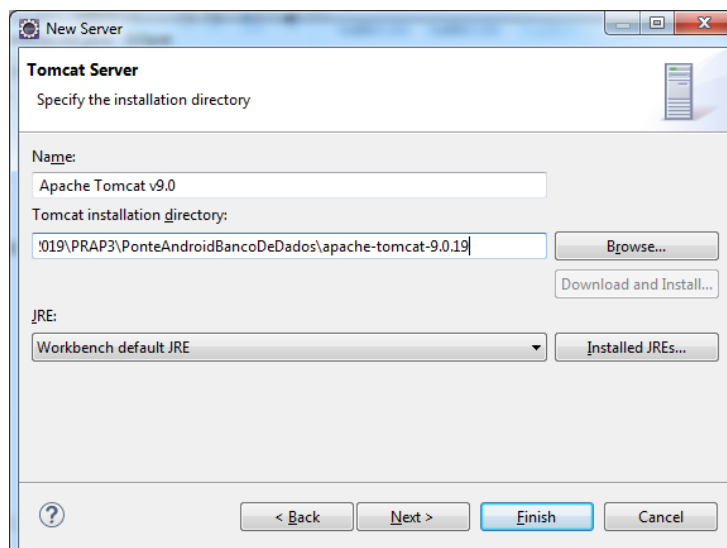
1.9. Selecione PonteABD e clique em Finish



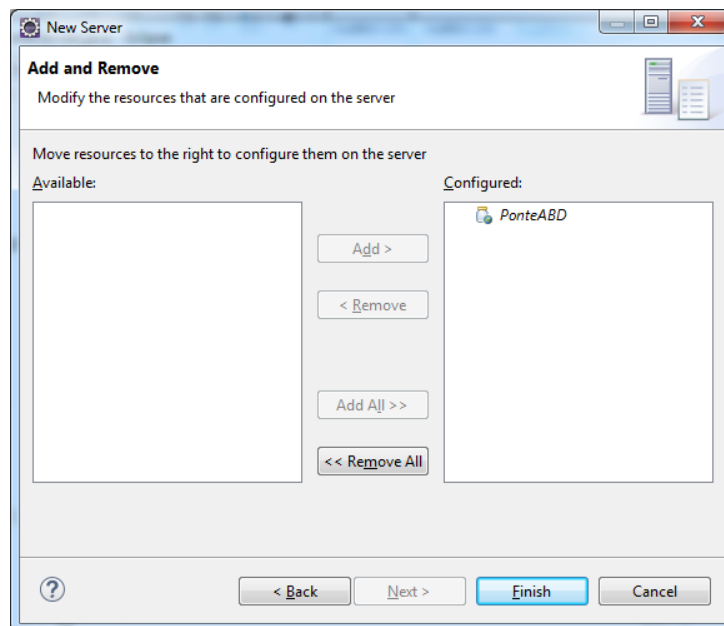
2.2. Selecione a opção Apache >> Apache Tomcat v9.0 Server e clique em Next



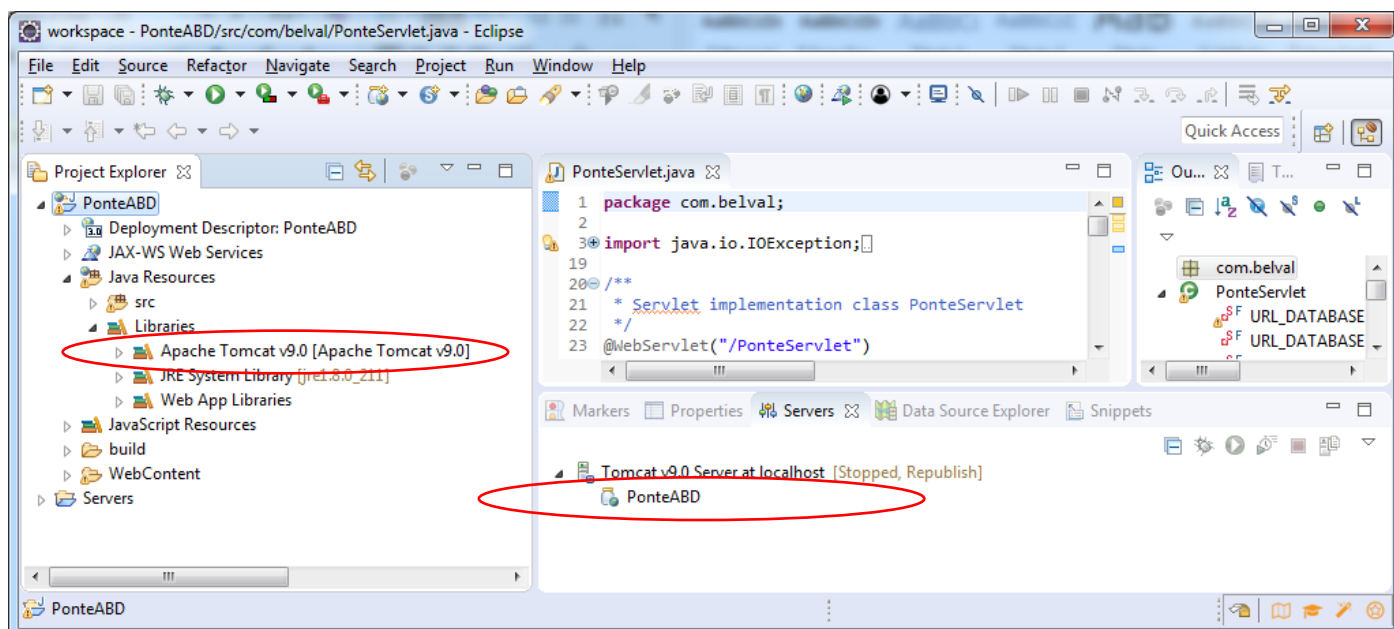
2.3. Copie o caminho de instalação do Tomcat9 e clique em Next.



2.4. Adicione o projeto PonteABD para rodar neste servidor e clique em Finish.

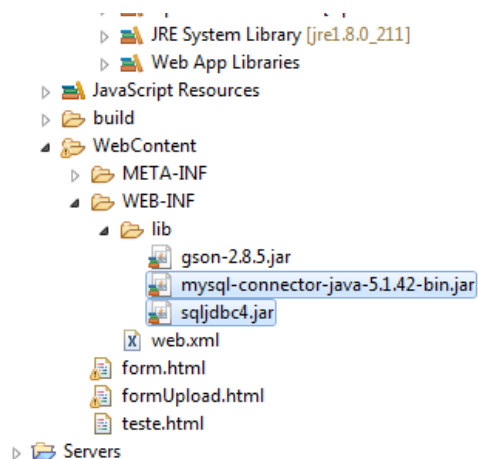


2.5. Agora nosso projeto compila pois agora possui a referência para a biblioteca de Servlet's e pode ser executado.

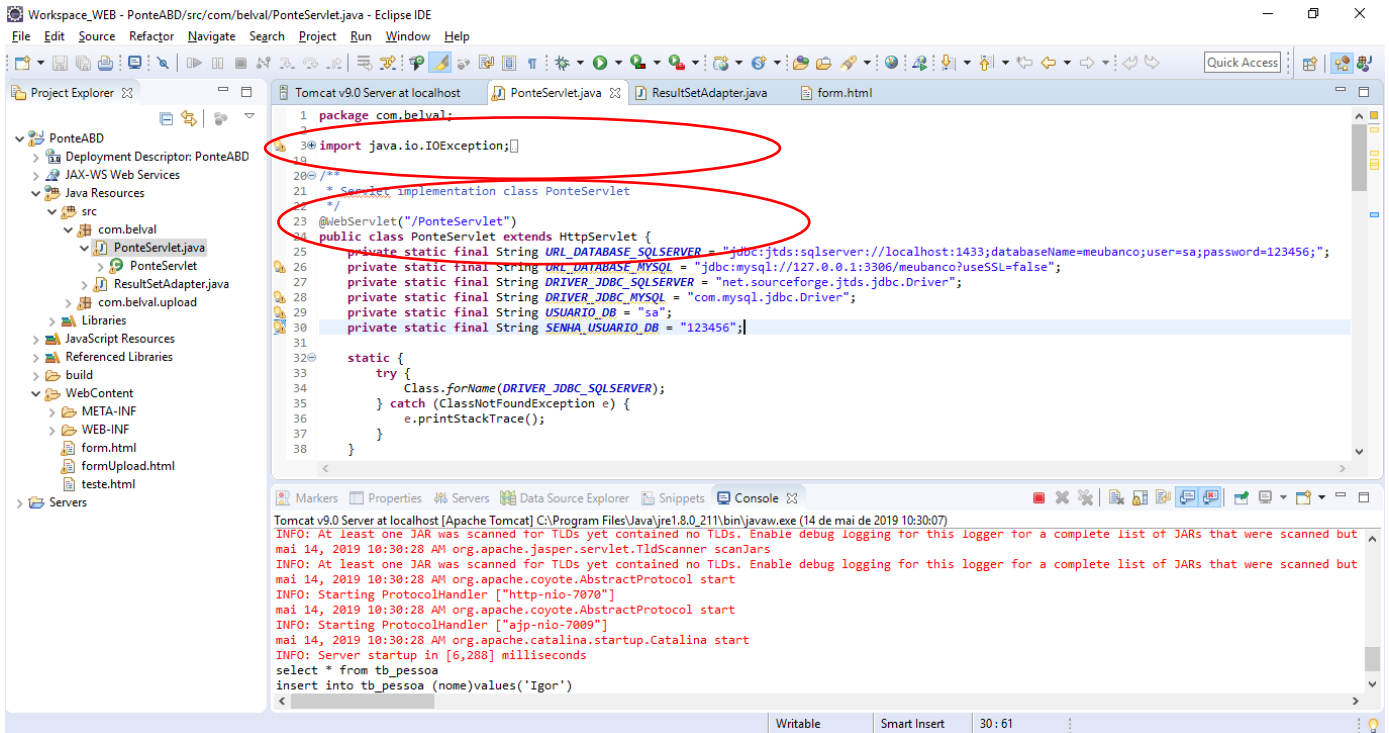


3. Configure a conexão do seu banco de dados

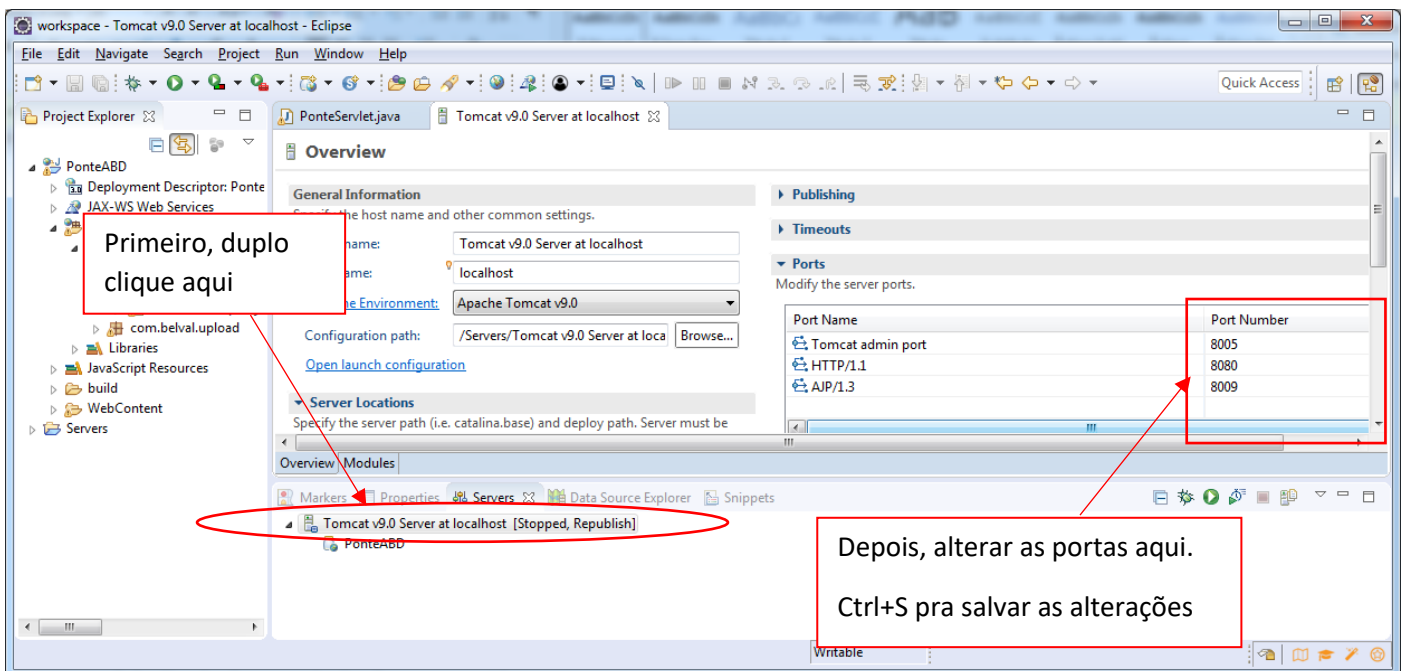
3.1. O projeto já vem com drivers para acesso ao MySQL e para o SQL Server. Caso seja necessário acessar um banco de dados de outro tipo é necessário copiar o driver na pasta lib.

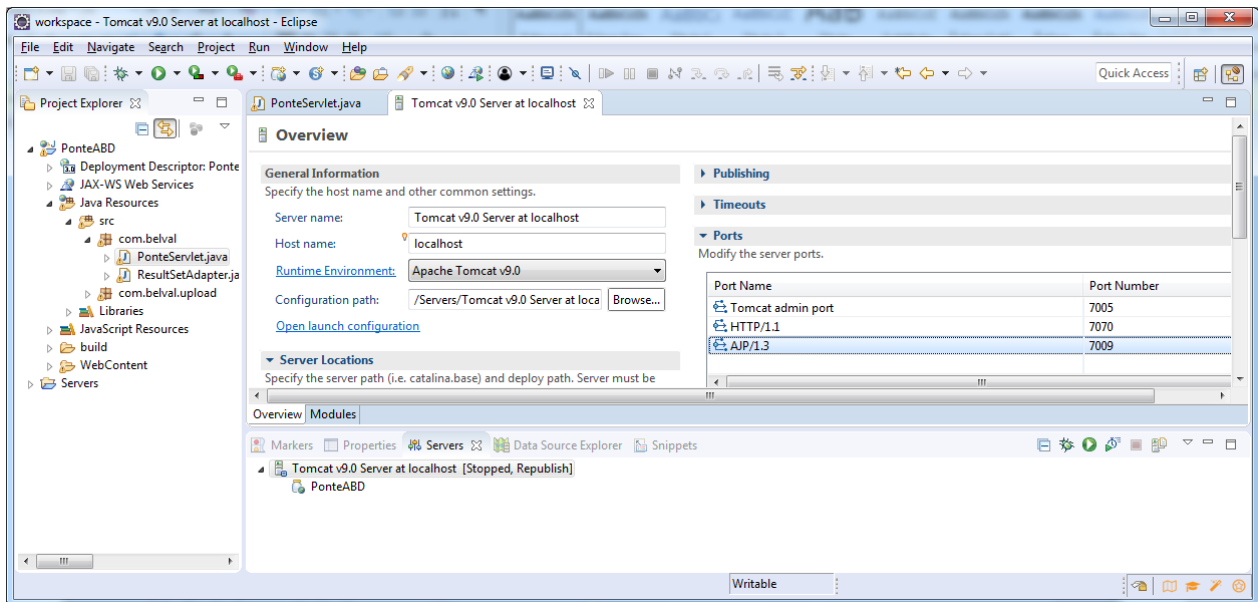


3.2. Abra a classe PonteServlet

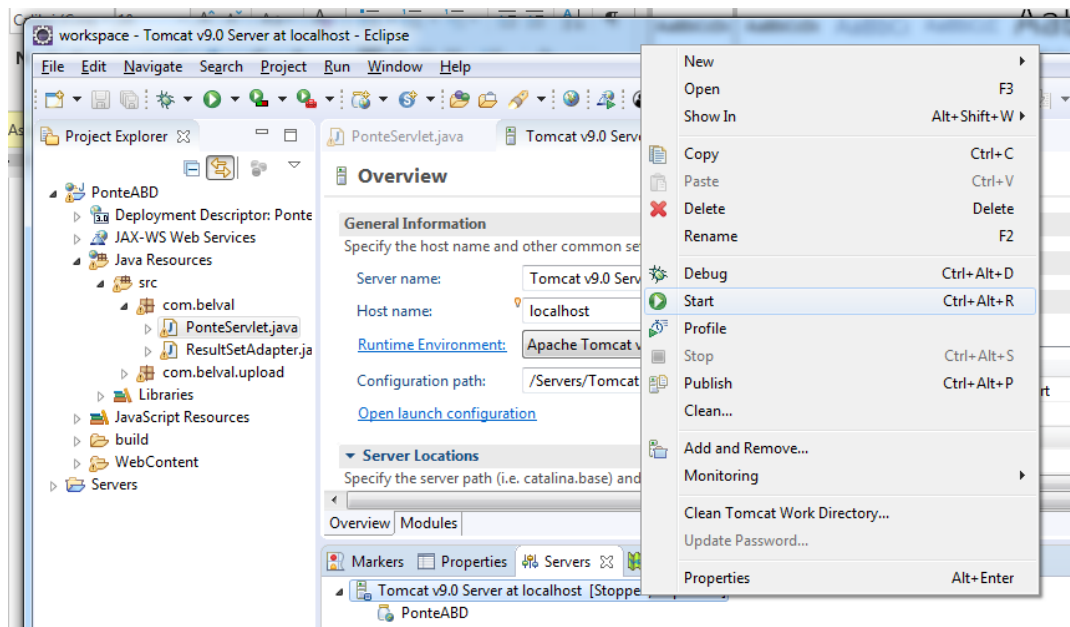


3.3. Na aba Servers, duplo clique no servidor e altere as portas para não conflitar com outros serviços

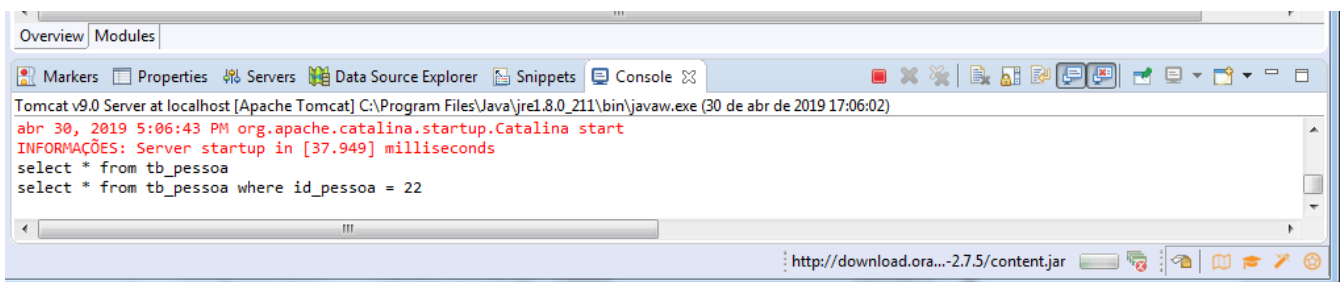




3.4. Clique com o botão direito do mouse no servidor e depois em Start.



3.5. Com o isso o servidor está no ar e poderá executar e responder às consultas do banco de dados.



4. Projeto do Banco de dados de teste

4.1. Crie o banco de dados meubanco com o script abaixo (SQLServer)

```
create database meubanco
use meubanco

CREATE TABLE tb_pessoa (
    id_pessoa int NOT NULL IDENTITY,
    nome varchar(255) NOT NULL,
    sobre_nome varchar(255),
    PRIMARY KEY (id_pessoa)
);

insert into tb_pessoa(nome, sobre_nome)
values('Arthur', 'Conan Doyle');
insert into tb_pessoa(nome, sobre_nome)
values('Clark', 'Kent');
insert into tb_pessoa(nome, sobre_nome)
values('Bruce', 'Wayne');

select * from tb_pessoa

COMMIT
```

5. Criar projeto Android para CRUD

5.1. AndroidManifest.xml(observe as permissões de acesso à internet e à rede)

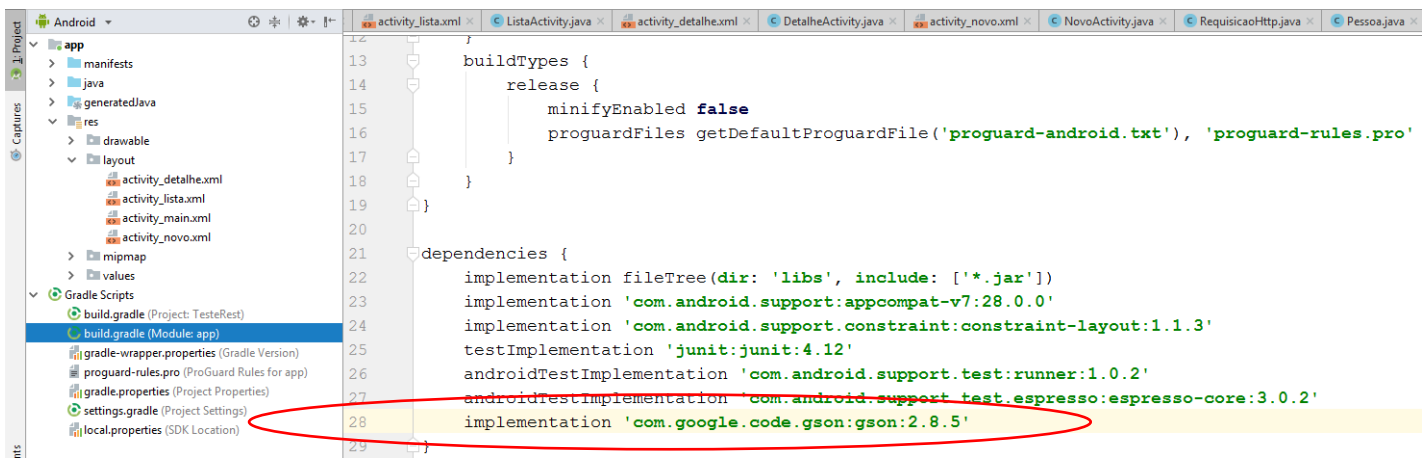
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.belval.testertest">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

    <application
        android:allowBackup="true"
        android:usesCleartextTraffic="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".ListaActivity" />
        <activity android:name=".DetalheActivity" />
        <activity android:name=".NovoActivity"></activity>
    </application>
```

5.2. No arquivo build.gradle é necessário incluir a dependência da biblioteca Gson que faz a serialização/desserialização



5.3. MainActivity.java

```
package com.belval.testertest;

import android.app.AlertDialog;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

import com.google.gson.FieldNamingStrategy;
import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;

import java.lang.reflect.Type;
import java.util.ArrayList;
import java.util.Map;

public class MainActivity extends AppCompatActivity {

    private AlertDialog progressDialog;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button btnListar = findViewById(R.id.btnListar);
        btnListar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent irParaLista = new Intent(MainActivity.this, ListaActivity.class);
                startActivity(irParaLista);
            }
        });

        Button btnNovo = findViewById(R.id.btnNovo);
        btnNovo.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent irParaTelaCriacao = new Intent(MainActivity.this, NovoActivity.class);
                startActivity(irParaTelaCriacao);
            }
        });
    }
}
```

5.4. activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout 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"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/btnListar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginBottom="8dp"
        android:text="LISTAR"
        android:textSize="25dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/txtIdPessoa"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="EXEMPLO CRUD"
        android:textAlignment="center"
        android:textSize="40dp"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/btnNovo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginBottom="8dp"
        android:text="NOVO"
        android:textSize="25dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```

5.5. ListaActivity.java

```
package com.belval.testertest;

import android.app.ProgressDialog;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;

import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;

import java.lang.reflect.Type;
import java.util.ArrayList;

public class ListaActivity extends AppCompatActivity {
    private ProgressDialog progressDialog;
    private ListView listView;
    private ArrayList<Pessoa> listaPessoas;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_lista);

        listView = findViewById(R.id.pessoaList);

        listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
                mostrarDetalhe(position);
            }
        });

    }

    @Override
    protected void onResume() {
        super.onResume();
        buscarPessoas();
    }

    private void mostrarDetalhe(int position) {
        Pessoa pessoa = listaPessoas.get(position);
        Intent irParaDetalhe = new Intent(ListaActivity.this, DetalheActivity.class);
        Bundle extras = new Bundle();
        extras.putInt("id", pessoa.getId());
        irParaDetalhe.putExtras(extras);
        startActivity(irParaDetalhe);
    }

    private void buscarPessoas() {
        new RequisicaoHttp(ListaActivity.this,
            new RequisicaoHttp.TratadorRetornoChamada() {
                @Override
                public void trataRetornoChamada(String dados) {
                    trataRetornoChamadaListar(dados);
                }
            },
            progressDialog)
            .addFormParam("comandoSql", "select * from tb_pessoa")
            .execute("http://10.0.2.2:7070/PonteABD/PonteServlet");
    }

    public void trataRetornoChamadaListar(String dados) {
        Gson gson = new Gson();
        Type tipoListaPessoas = new TypeToken<ArrayList<Pessoa>>().getType();
        listaPessoas = gson.fromJson(dados, tipoListaPessoas);

        ArrayAdapter<Pessoa> adapter =
            new ArrayAdapter<Pessoa>(
                ListaActivity.this,
                android.R.layout.simple_list_item_1,
                listaPessoas);

        listView.setAdapter(adapter);
    }
}
```

5.6. activity_lista.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout 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"
    tools:context=".ListaActivity">

    <ListView
        android:id="@+id/pessoaList"
        android:layout_width="368dp"
        android:layout_height="495dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginBottom="8dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```

5.7. DetalheActivity.java – Parte 1, package e imports

```
package com.belval.testertest;

import android.app.ProgressDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;

import java.lang.reflect.Type;
import java.util.ArrayList;
```


5.8. DetalheActivity.java – Parte 2

```

public class DetalheActivity extends AppCompatActivity {
    private ProgressDialog progressDialog;
    private Pessoa pessoa;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detalhe);
        //Dispara a chamada asincrona que recupera os dados
        // e que serão recuperados no metodo preencheCampos()
        int idPessoa = getIntent().getExtras().getInt("id");
        buscarPessoa(idPessoa);
    }
    private void buscarPessoa(int idPessoa) {
        new RequisicaoHttp(DetalheActivity.this,
            new RequisicaoHttp.TratadorRetornoChamada() {
                @Override
                public void trataRetornoChamada(String dados) {
                    preencheCampos(dados);
                }
            },
            progressDialog)
            .addFormParam("comandoSql", "select * from tb_pessoa where id_pessoa = " + idPessoa)
            .execute("http://10.0.2.2:7070/PonteABD/PonteServlet");
    }
    private void preencheCampos(String dados) {
        Gson gson = new Gson();
        Type tipoListaPessoas = new TypeToken<ArrayList<Pessoa>>().getType();
        ArrayList<Pessoa> list = gson.fromJson(dados, tipoListaPessoas);
        if (list.size() > 0) {
            this.pessoa = list.get(0);
            TextView txtIdPessoa = findViewById(R.id.txtIdPessoa);
            txtIdPessoa.setText("" + pessoa.getId());
            TextView txtNome = findViewById(R.id.editNome);
            txtNome.setText(pessoa.getNome());
            Button btnExcluir = findViewById(R.id.btnExcluir);
            btnExcluir.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    excluir(pessoa.getId());
                }
            });
            Button btnSalvar = findViewById(R.id.btnSalvar);
            btnSalvar.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    EditText editNome = v.getRootView().findViewById(R.id.editNome);
                    pessoa.setNome(editNome.getText().toString());
                    salvar(pessoa);
                }
            });
        } else {
            this.pessoa = null;
        }
    }
    private void excluir(int idPessoa) {
        new RequisicaoHttp(DetalheActivity.this,
            new RequisicaoHttp.TratadorRetornoChamada() {
                @Override
                public void trataRetornoChamada(String dados) {
                    if (dados.contains("comandoExecutado") &&
                        dados.contains("true")) {
                        Toast.makeText(DetalheActivity.this, "Exclusão concluída", Toast.LENGTH_LONG);
                        DetalheActivity.this.finish();
                    }
                }
            },
            progressDialog)
            .addFormParam("comandoSql", "delete from tb_pessoa where id_pessoa = " + idPessoa)
            .execute("http://10.0.2.2:7070/PonteABD/PonteServlet");
    }
    private void salvar(Pessoa p) {
        //Continua...
    }
}

```

5.9. DetalheActivity.java – Parte 3, método salvar(Pessoa p)

```
...
private void salvar(Pessoa p) {
    String comandoSql =
        String.format(
            "update tb_pessoa " +
            "set nome = '%s' " +
            "where id_pessoa = %d ",
            p.getNome(),
            p.getId());
    new RequisicaoHttp(DetalheActivity.this,
        new RequisicaoHttp.TratadorRetornoChamada() {
            @Override
            public void trataRetornoChamada(String dados) {
                if (dados.contains("comandoExecutado") &&
                    dados.contains("true")) {
                    Toast.makeText(DetalheActivity.this,
                        "Atualização concluída", Toast.LENGTH_LONG);
                    DetalheActivity.this.finish();
                }
            }
        },
        progressDialog)
        .addFormParam("comandoSql", comandoSql)
        .execute("http://10.0.2.2:7070/PonteABD/PonteServlet");
}
```

ABAIXO TEMOS A EXPLICAÇÃO DA UTILIZAÇÃO DO CELULAR E DO EMULADOR, O CÓDIGO ACIMA ESTA VISANDO O USO DO EMULADOR, PARA O CELULAR DEVE SER UTILIZADO O IP DO SERVIDOR 192.168.0.1 OU ALGO PARECIDO COM ISSO.

Observe que aqui usamos o ip 192.168.0.11 pois executamos o app em um celular e não no emulador e como esse era o endereço da máquina que rodava a Ponte ABD na rede local foi o endereço acessível. Caso o app rode na mesma máquina que a Ponte ABD a partir do emulador o endereço deve ser 10.0.2.2 conforme as informações em <https://developer.android.com/studio/run/emulator-networking>.

Android Studio interface showing the 'Network address space' documentation page. The page explains that each emulator instance runs behind a virtual router/firewall service that isolates it from the development machine network interfaces and settings. It lists various network addresses and their descriptions, with 10.0.2.2 highlighted as the special alias to the host loopback interface.

Network Address	Description
10.0.2.1	Router/gateway address
10.0.2.2	Special alias to your host loopback interface (i.e., 127.0.0.1 on your development machine)
10.0.2.3	First DNS server
10.0.2.4 / 10.0.2.5 / 10.0.2.6	Optional second, third and fourth DNS server (if any)
10.0.2.15	The emulated device network/ethernet interface
127.0.0.1	The emulated device loopback interface

5.10. activity_detalhe.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout 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"
    tools:context=".DetalheActivity">
    <TextView
        android:id="@+id/txtLabelIdPessoa"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="8dp"
        android:layout_marginTop="8dp"
        android:text="ID_PESSOA:"
        android:textSize="25dp"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <TextView
        android:id="@+id/txtIdPessoa"
        android:layout_width="73dp"
        android:layout_height="32dp"
        android:layout_marginStart="40dp"
        android:layout_marginTop="8dp"
        android:text="----"
        android:textSize="25dp"
        app:layout_constraintStart_toEndOf="@+id/txtLabelIdPessoa"
        app:layout_constraintTop_toTopOf="parent" />
    <TextView
        android:id="@+id/txtLabelNome"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:text="NOME:"
        android:textSize="25dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/txtLabelIdPessoa" />
    <EditText
        android:id="@+id/editNome"
        android:layout_width="173dp"
        android:layout_height="44dp"
        android:layout_marginStart="96dp"
        android:layout_marginTop="8dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:text="----"
        app:layout_constraintStart_toEndOf="@+id/txtLabelNome"
        app:layout_constraintTop_toBottomOf="@+id/txtIdPessoa" />
    <Button
        android:id="@+id/btnExcluir"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginBottom="8dp"
        android:text="Excluir"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
    <Button
        android:id="@+id/btnSalvar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginBottom="8dp"
        android:text="Salvar"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />
</android.support.constraint.ConstraintLayout>
```

5.11. NovoActivity.java

```
package com.belval.testertest;

import android.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class NovoActivity extends AppCompatActivity {
    private ProgressDialog progressDialog;
    private Pessoa pessoa;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_novo);

        Button btnCriar = findViewById(R.id.btnCriar);
        btnCriar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                pessoa = new Pessoa();
                //Aqui recupera os campos da tela e preenche o bean pessoa
                EditText editNome = v.getRootView().findViewById(R.id.editNome);
                pessoa.setNome(editNome.getText().toString());
                criar(pessoa);
            }
        });
    }
    private void criar(Pessoa p) {
        String comandoSql =
            String.format(
                "insert into tb_pessoa (nome)" + // o id sera gerado pelo banco
                "values('%s') ",
                p.getNome());
        new RequisicaoHttp(NovoActivity.this,
            new RequisicaoHttp.TratadorRetornoChamada() {
                @Override
                public void trataRetornoChamada(String dados) {
                    if (dados.contains("comandoExecutado") &&
                        dados.contains("true")) {
                        Toast.makeText(NovoActivity.this, "Inclusão concluída", Toast.LENGTH_LONG);
                        NovoActivity.this.finish();
                    }
                }
            },
            progressDialog)
            .addFormParam("comandoSql", comandoSql)
            .execute("http://10.0.2.2:7070/PonteABD/PonteServlet");
    }
}
```

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout 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"
    tools:context=".NovoActivity">

    <TextView
        android:id="@+id/txtLabelIdPessoa"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="8dp"
        android:layout_marginTop="8dp"
        android:text="ID_PESSOA:"
        android:textSize="25dp"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/txtIdPessoa"
        android:layout_width="73dp"
        android:layout_height="32dp"
        android:layout_marginStart="40dp"
        android:layout_marginTop="8dp"
        android:text="---"
        android:textSize="25dp"
        app:layout_constraintStart_toEndOf="@+id/txtLabelIdPessoa"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/txtLabelNome"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:text="NOME:"
        android:textSize="25dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/txtLabelIdPessoa" />

    <EditText
        android:id="@+id/editNome"
        android:layout_width="173dp"
        android:layout_height="44dp"
        android:layout_marginStart="96dp"
        android:layout_marginTop="8dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:text="---"
        app:layout_constraintStart_toEndOf="@+id/txtLabelNome"
        app:layout_constraintTop_toBottomOf="@+id/txtIdPessoa" />

    <Button
        android:id="@+id/btnCriar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginBottom="8dp"
        android:text="Criar"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />

</android.support.constraint.ConstraintLayout>

```

5.13. RequisicaoHttp.java – Parte 1, package e imports

```
package com.belval.testertest;

import android.app.Activity;
import android.app.ProgressDialog;
import android.os.AsyncTask;
import android.util.Log;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.HashMap;
import java.util.Map;
import java.util.Set;
```

5.14. RequisicaoHttp.java – Parte 2, estrutura geral da classe e atributos

```
public class RequisicaoHttp extends AsyncTask<String, Void, String> {
    private Activity activity;
    private TratadorRetornoChamada tratadorRetorno;
    private ProgressDialog progressDialog;
    private Map<String, String> formParans = new HashMap<>();

    public interface TratadorRetornoChamada {
        public void trataRetornoChamada(String dados);
    }

    public RequisicaoHttp(Activity activity, TratadorRetornoChamada tratadorRetorno,
        ProgressDialog progressDialog) {
        this.activity = activity;
        this.tratadorRetorno = (TratadorRetornoChamada) tratadorRetorno;
        this.progressDialog = progressDialog;
    }

    public RequisicaoHttp(Activity activity,
        TratadorRetornoChamada tratadorRetorno,
        ProgressDialog progressDialog) {
        this.activity = activity;
        this.tratadorRetorno = tratadorRetorno;
        this.progressDialog = progressDialog;
    }

    public RequisicaoHttp addFormParam(String chave, String valor) {
        this.formParans.put(chave, valor);
        return this;
    }

    @Override
    protected String doInBackground(String... params) {
        ...
    }

    private String montarUrl(String base) throws UnsupportedEncodingException {
        ...
    }

    @Override
    protected void onPreExecute() {
        ...
    }

    @Override
    protected void onPostExecute(String dados) {
        ...
    }
}
```

5.15. RequisicaoHttp.java – Parte 3, método doInBackground()

```
protected String doInBackground(String... params) {
    // Esta etapa é usada para executar a tarefa em background ou fazer a chamada para o webservice
    HttpURLConnection urlConnection = null;
    BufferedReader reader = null;
    try {
        String urlStr = montarUrl(params[0]);
        URL url = new URL(urlStr);
        urlConnection = (HttpURLConnection) url.openConnection();
        urlConnection.setRequestMethod("POST");
        //urlConnection.connect();
        InputStream inputStream = urlConnection.getInputStream();
        reader = new BufferedReader(new InputStreamReader(inputStream));
        String linha;
        StringBuffer buffer = new StringBuffer();
        while((linha = reader.readLine()) != null) {
            buffer.append(linha);
            buffer.append("\n");
        }
        return buffer.toString();
    } catch (Exception e) {
        e.printStackTrace();
        if (urlConnection != null) {
            urlConnection.disconnect();
        }
        if (reader != null) {
            try {
                reader.close();
            } catch (IOException e1) {
                e1.printStackTrace();
            }
        }
    }
    return null;
}
```

5.16. RequisicaoHttp.java – Parte 4, método montarUrl()

```
private String montarUrl(String base) throws UnsupportedOperationException {
    StringBuilder url = new StringBuilder(base);
    if (formParans.size() > 0) {
        url.append("?");
        Set<Map.Entry<String, String>> entradas = formParans.entrySet();

        for (Map.Entry<String, String> entrada : entradas) {

            url.append(entrada.getKey()).append("=").append(entrada.getValue()).append("&");

        }

    }

    String urlComParametros = url.substring(0, url.length() - 1).toString();
    //urlComParametros = URLEncoder.encode(urlComParametros, "UTF-8");
    return urlComParametros;
}
```

5.17. RequisicaoHttp.java – Parte 4, métodos onPreExecute() e onPostExecute()

```
@Override
protected void onPreExecute() {
    Log.i("AsyncTask", "Exibindo ProgressDialog na tela Thread: "+Thread.currentThread().getName());
    this.progressBar.show((Activity) this.activity, "Por favor Aguarde ...",
        "Baixando Dados ...");
}

@Override
protected void onPostExecute(String dados) {
    // Faça alguma coisa com os dados
    if (this.tratadorRetorno != null) {
        this.tratadorRetorno.trataRetornoChamada(dados);
    }
    this.progressBar.dismiss();
}
}
```

5.18. Classe Pessoa que mapeia a tabela tb_pessoa

```
package com.belval.testertest;

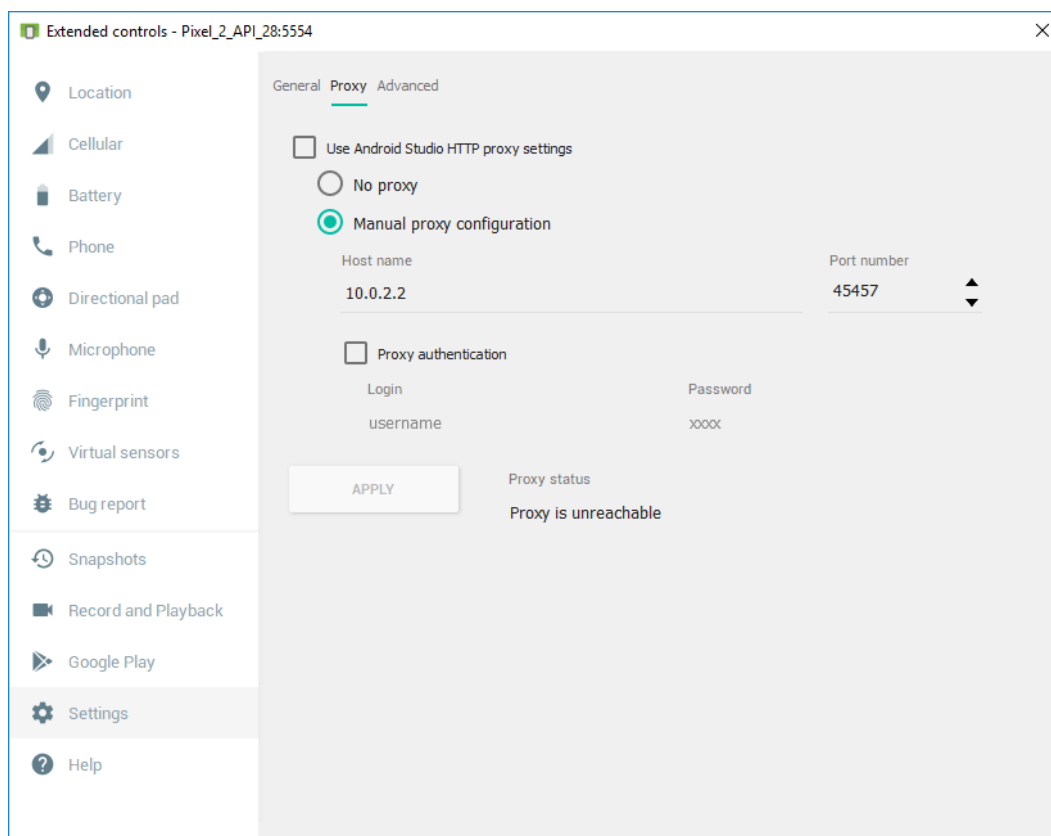
import com.google.gson.annotations.SerializedName;

public class Pessoa {
    @SerializedName("id_pessoa")
    private int id;
    private String nome;
    public Pessoa() {
    }
    public Pessoa(int id, String nome) {
        this.id = id;
        this.nome = nome;
    }
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
    @Override
    public String toString() {
        return "[id=" + id +
            ", nome='" + nome + "\'"]";
    }
}
```

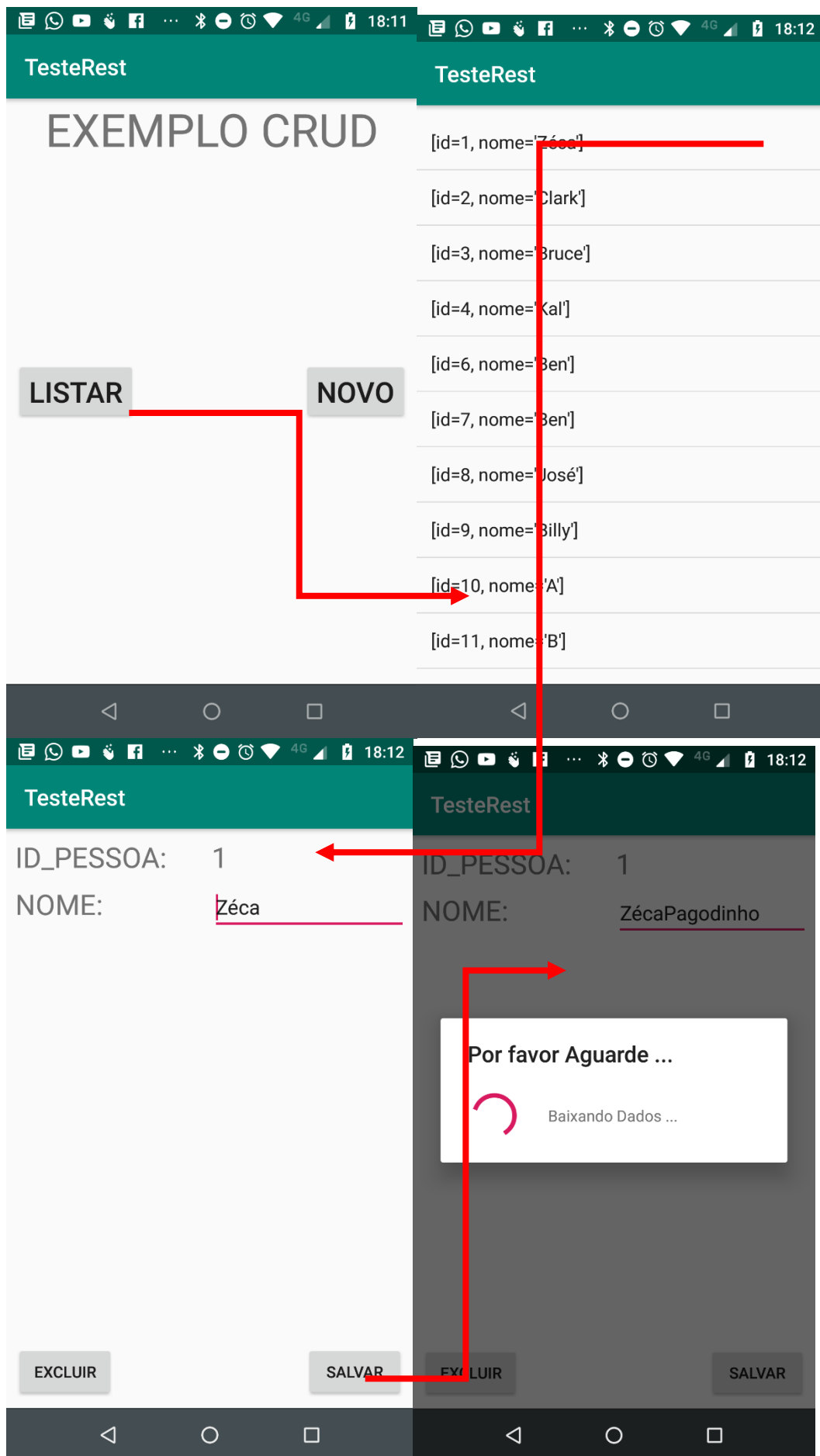
Observe que nesta classe não incluímos o atributo `sobre_nome` que existe no banco. Fizemos assim para mostrar que nem todos os campos da tabela precisam descer para o cliente de forma economizar largura de banda (o nosso select poderia trazer apenas os campos a serem utilizados).

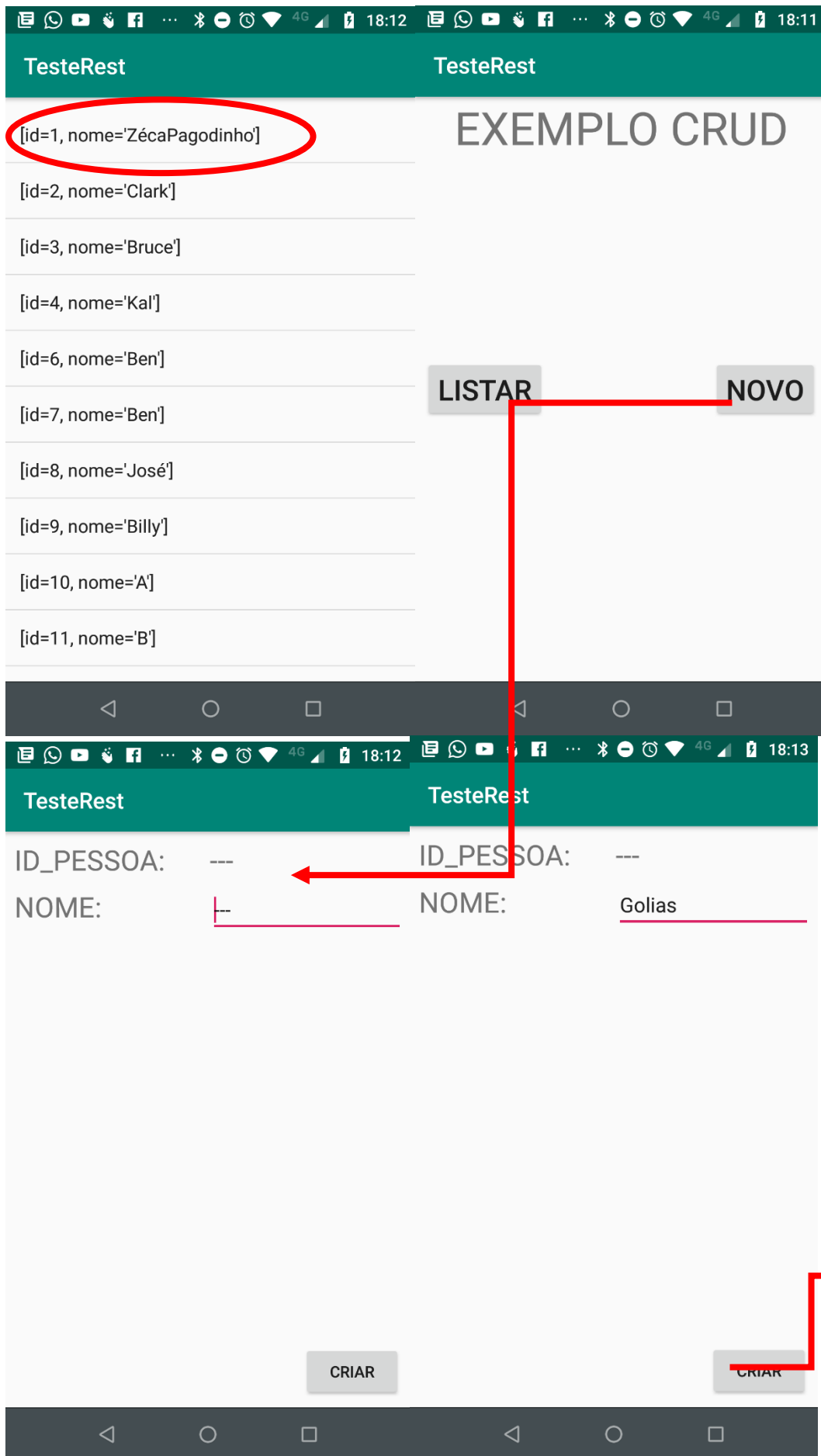
Outra coisa a ser observada é que o nome do atributo(`id`) é diferente do nome do campo da tabela(`id_pessoa`) e pra que o Gson faça a correlação entre eles usamos a annotation `@SerializedName`. Caso eles fossem idênticos, como é o caso de `nome`, essa anotação não seria necessária.

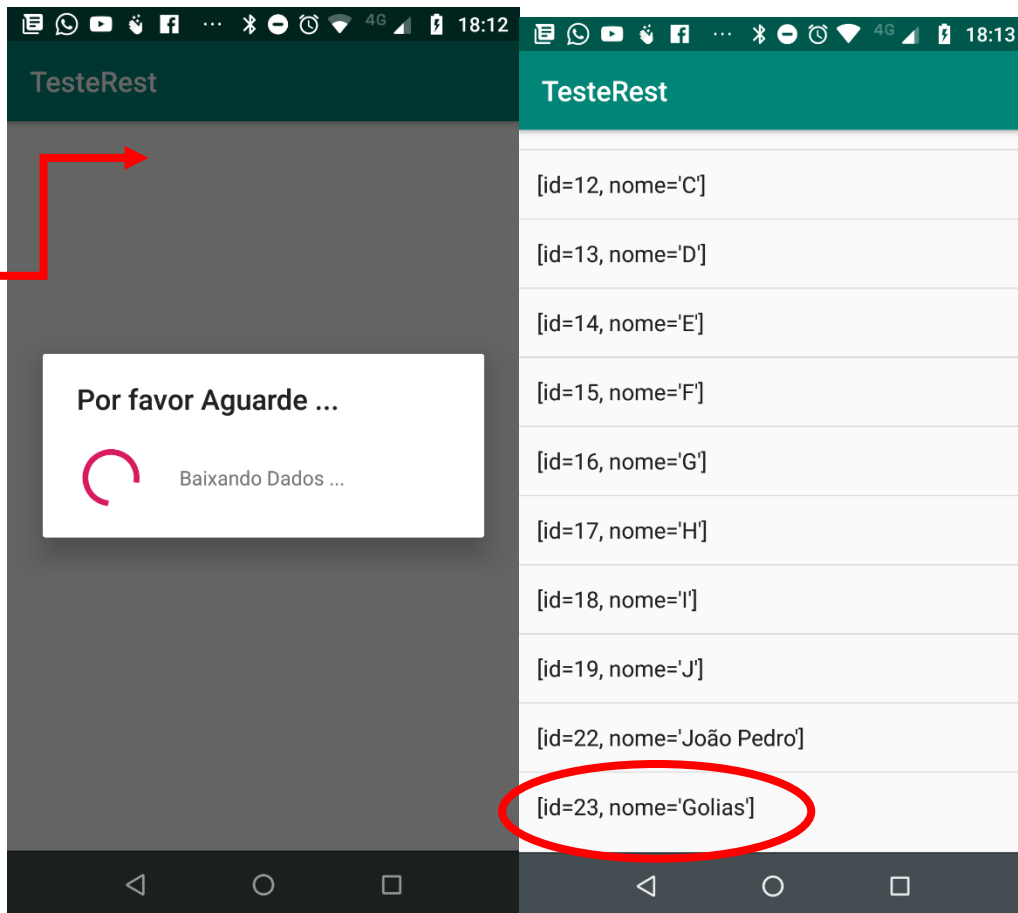
ANTES DE TESTAR O EMULADOR, DEVE-SE ALTERAR A CONFIGURAÇÃO DO PROXY, COMO ABAIXO:



5.19. Abaixo algumas telas do app







REFERÊNCIA BIBLIOGRÁFICA

- [1] <https://stackoverflow.com/questions/6760585/accessing-localhostport-from-android-emulator>. Acessado em 13/05/2019.
- [2] <https://medium.com/@son.rommer/fix-cleartext-traffic-error-in-android-9-pie-2f4e9e2235e6>. Acessado em 13/05/2019.