**UNAERP**

**UNIVERSIDADE DE RIBEIRÃO PRETO**

**Raul Felipe de Melo - 816587**

**TÓPICOS AVANÇADOS EM TECNOLOGIA:**

**Projeto de Gerenciamento de Vendas**

**Ribeirão Preto**

**2016**

Sumário

[1. Objetivo da Aplicação 3](#_Toc454187840)

[2. Funcionalidades 3](#_Toc454187841)

[3. Telas Da aplicação 4](#_Toc454187842)

[4. Fonte da aplicação 6](#_Toc454187843)

[4.1. Adapter 6](#_Toc454187844)

[4.2. Atividades 7](#_Toc454187845)

[4.3. Connection 14](#_Toc454187846)

[4.4. Dao 15](#_Toc454187847)

[4.5. Forms 26](#_Toc454187848)

[4.6. Fragment 32](#_Toc454187849)

[4.7. Helper 44](#_Toc454187850)

[4.8. Interfaces 50](#_Toc454187851)

[4.9. Model 50](#_Toc454187852)

[4.10. Tasks 58](#_Toc454187853)

[4.11. Util 59](#_Toc454187854)

# Objetivo da Aplicação

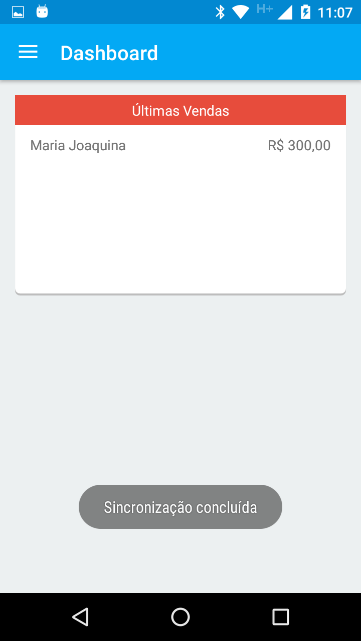
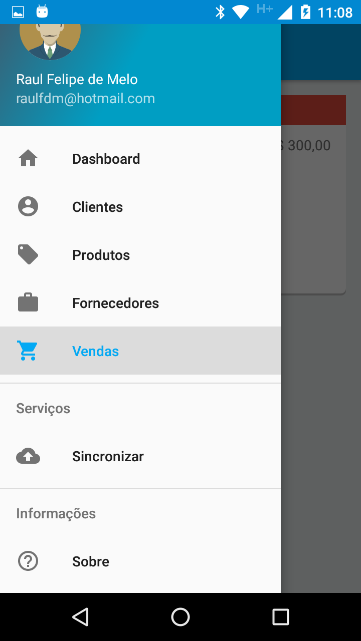
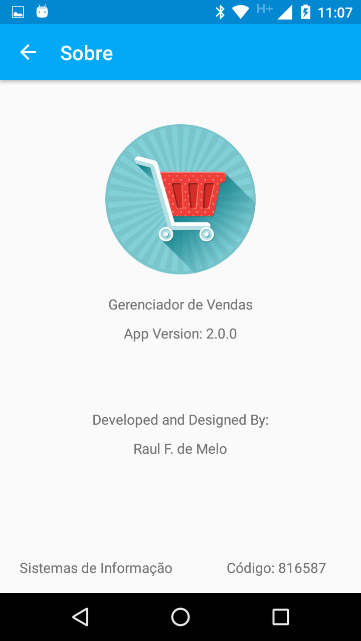
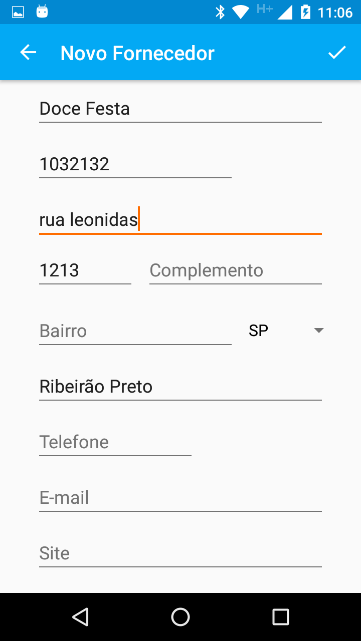
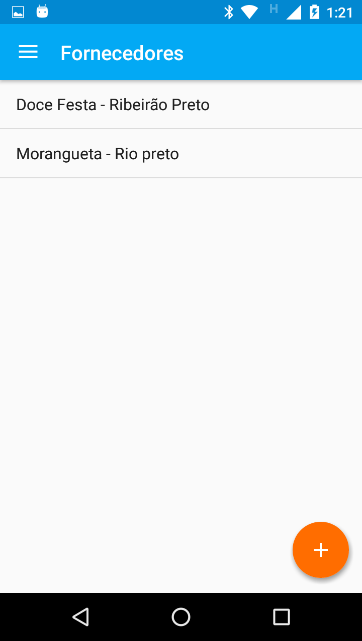
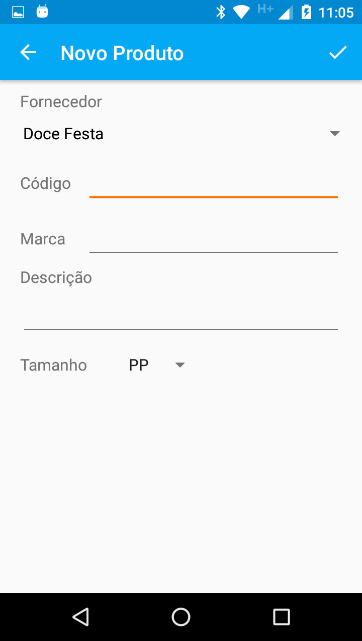
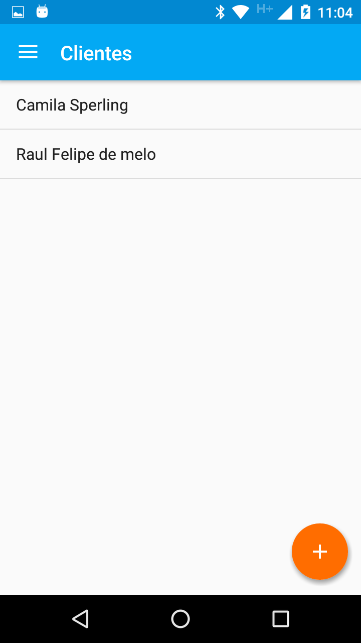
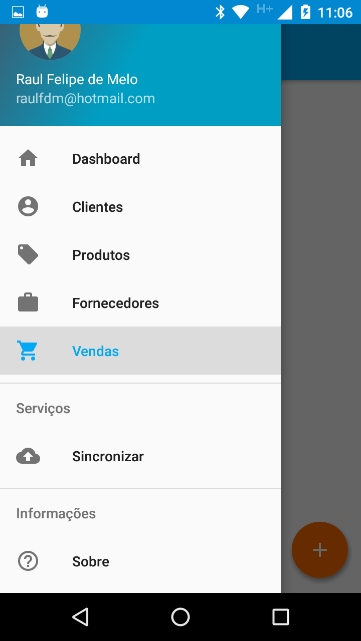
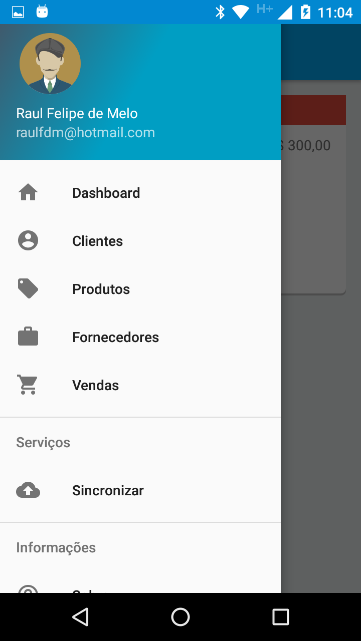
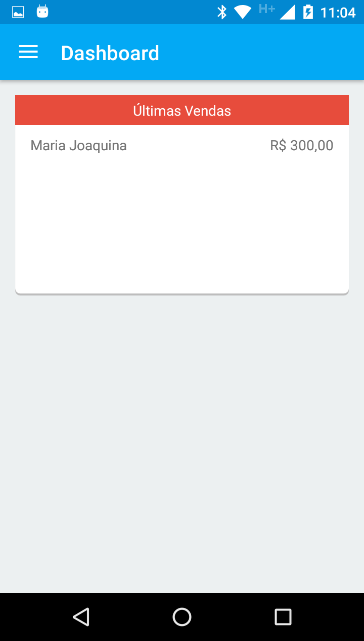
O objetivo desta aplicação, é fazer o controle de vendas de roupas, indo desde a compra com o fornecedor até a venda final.

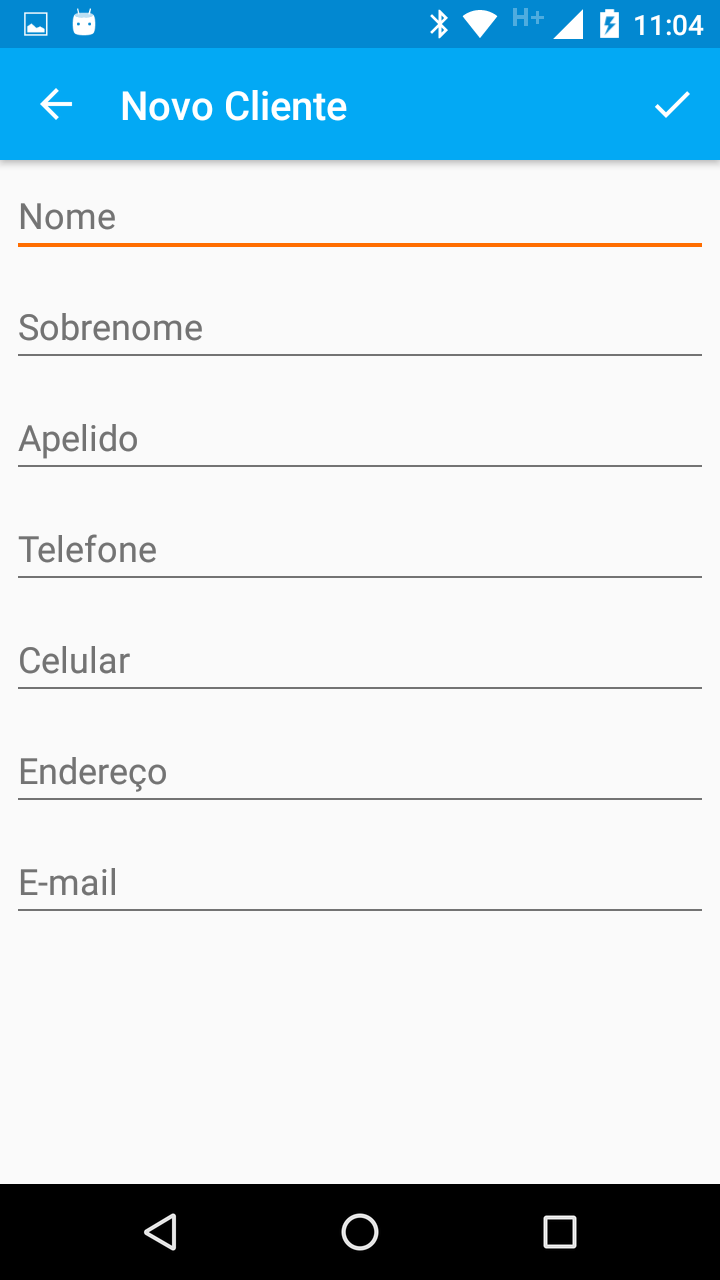
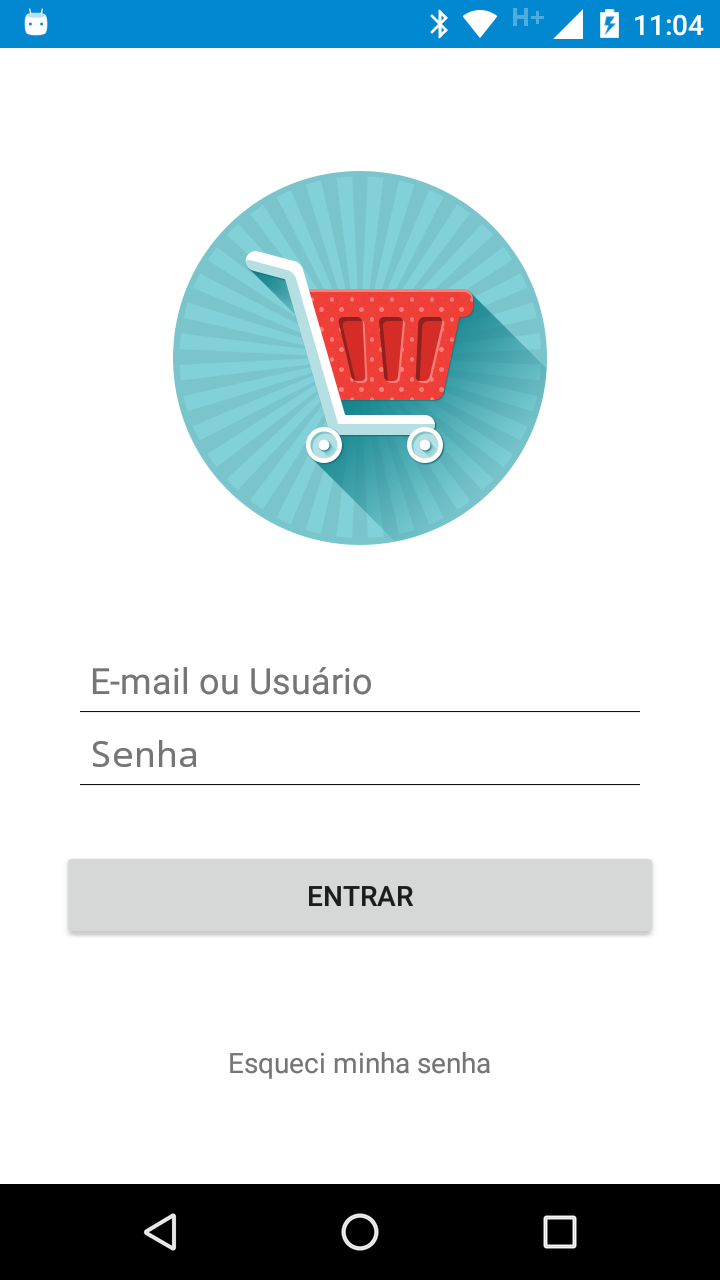
# Funcionalidades

Atualmente é possível:

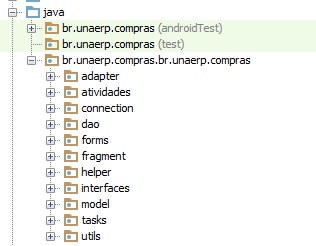
* Cliente:
  + Cadastrar;
  + Alterar;
  + Remover;
  + Enviar E-mail.
* Fornecedor
  + Cadastrar;
  + Alterar;
  + Remover;
  + Enviar E-mail.
* Produtos
  + Cadastrar;
  + Alterar;
  + Remover.
* Tela Principal
  + Constatar as últimas vendas;
* Fazer backup dos clientes na nuvem;

# Telas Da aplicação





# Fonte da aplicação



## Adapter

* + 1. **package** br.unaerp.compras.br.unaerp.compras.adapter;  
         
       **import** org.json.JSONException;  
       **import** org.json.JSONStringer;  
         
       **import** java.util.List;  
         
       **import** br.unaerp.compras.br.unaerp.compras.model.ClienteModel;  
         
       */\*\*  
        \* Created by raul on 6/6/2016.  
        \*/***public class** ClienteAdapter {  
        **public** String converteJson(List<ClienteModel> clientes) {  
        JSONStringer js = **new** JSONStringer();  
         
        **try** {  
        js.array();  
         
        **for** (ClienteModel cliente : clientes) {  
        js.object();  
        js.key(**"\_id"**).value(cliente.getId());  
        js.key(**"nome"**).value(cliente.getNome());  
        js.key(**"sobrenome"**).value(cliente.getSobrenome());  
        js.key(**"apelido"**).value(cliente.getApelido());  
        js.key(**"telefone"**).value(cliente.getTelefone());  
        js.key(**"celular"**).value(cliente.getCelular());  
        js.key(**"endereco"**).value(cliente.getEndereco());  
        js.key(**"email"**).value(cliente.getEmail());  
        js.endObject();  
        }  
         
        js.endArray();  
        **return** js.toString();  
        } **catch** (JSONException e) {  
        e.printStackTrace();  
        }  
         
         
        **return null**;  
        }  
       }
    2. **package** br.unaerp.compras.br.unaerp.compras.adapter;  
         
       **import** org.json.JSONException;  
       **import** org.json.JSONStringer;  
         
       **import** br.unaerp.compras.br.unaerp.compras.model.LoginModel;  
         
       */\*\*  
        \* Created by raul on 12/6/2016.  
        \*/***public class** LoginAdapter {  
         
        **public** String converteJson(LoginModel usuario) {  
        JSONStringer js = **new** JSONStringer();  
         
        **try** {  
        js.object();  
        js.key(**"usuario"**).value(usuario.getUsuario());  
        js.key(**"senha"**).value(usuario.getSenha());  
        js.endObject();  
         
        } **catch** (JSONException e) {  
        e.printStackTrace();  
        }  
         
        **return** js.toString();  
        }  
       }

## Atividades

* + 1. **package** br.unaerp.compras.br.unaerp.compras.atividades;  
         
       **import** android.os.Bundle;  
       **import** android.support.design.widget.FloatingActionButton;  
       **import** android.support.design.widget.Snackbar;  
       **import** android.support.v7.app.AppCompatActivity;  
       **import** android.support.v7.widget.Toolbar;  
       **import** android.view.Menu;  
       **import** android.view.MenuInflater;  
       **import** android.view.MenuItem;  
       **import** android.view.View;  
         
       **import** br.unaerp.compras.R;  
         
       **public class** AboutActivity **extends** AppCompatActivity {  
         
        @Override  
        **protected void** onCreate(Bundle savedInstanceState) {  
        **super**.onCreate(savedInstanceState);  
        setContentView(R.layout.activity\_about);  
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);  
        setSupportActionBar(toolbar);  
         
        }  
         
        @Override  
        **public boolean** onCreateOptionsMenu(Menu menu) {  
        getSupportActionBar().setDisplayHomeAsUpEnabled(**true**);  
        getSupportActionBar().setDisplayShowHomeEnabled(**true**);  
        **return super**.onCreateOptionsMenu(menu);  
        }  
         
        @Override  
        **public boolean** onOptionsItemSelected(MenuItem item) {  
        **switch** (item.getItemId()) {  
        **case** android.R.id.home:  
        finish();  
        **break**;  
        }  
        **return super**.onOptionsItemSelected(item);  
        }  
         
         
       }
    2. **package** br.unaerp.compras.br.unaerp.compras.atividades;  
         
       **import** android.app.Activity;  
       **import** android.content.Intent;  
       **import** android.net.Uri;  
       **import** android.os.Bundle;  
       **import** android.view.View;  
       **import** android.view.Window;  
       **import** android.widget.Button;  
       **import** android.widget.TextView;  
       **import** android.widget.Toast;  
         
       **import** br.unaerp.compras.R;  
       **import** br.unaerp.compras.br.unaerp.compras.dao.LoginDAO;  
       **import** br.unaerp.compras.br.unaerp.compras.helper.LoginHelper;  
       **import** br.unaerp.compras.br.unaerp.compras.model.LoginModel;  
         
       **public class** LoginActivity **extends** Activity {  
         
        **private** LoginHelper helperLogin;  
         
        @Override  
        **protected void** onCreate(Bundle savedInstanceState) {  
        **super**.onCreate(savedInstanceState);  
        **this**.requestWindowFeature(Window.FEATURE\_NO\_TITLE);  
        setContentView(R.layout.activity\_login);  
        helperLogin = **new** LoginHelper(**this**);  
         
        Button logar = (Button) findViewById(R.id.login\_btn\_entrar);  
        TextView recuperarSenha = (TextView) findViewById(R.id.recuperarSenha);  
         
        logar.setOnClickListener(**new** View.OnClickListener() {  
        @Override  
        **public void** onClick(View v) {  
        String formValidado = helperLogin.validaFormLogin();  
         
        **if**(formValidado != **null**){  
        Toast.makeText(LoginActivity.**this**, **"Por favor, preencha o "**+formValidado, Toast.LENGTH\_SHORT).show();  
        }**else** {  
        LoginDAO dao = **new** LoginDAO(LoginActivity.**this**,MainActivity.versaoBD);  
        LoginModel retorno = dao.validaAcesso(helperLogin.getUsuario());  
        **if**(retorno.getNomeCompleto().length()>0){  
        Intent abreSistema = **new** Intent(LoginActivity.**this**, MainActivity.**class**);  
        startActivity(abreSistema);  
        finish();  
        }**else**{  
        Toast.makeText(LoginActivity.**this**, **"Usuário ou Senha Inválidos"**, Toast.LENGTH\_SHORT).show();  
        }  
        }  
        }  
        });  
         
        recuperarSenha.setOnClickListener(**new** View.OnClickListener() {  
        @Override  
        **public void** onClick(View v) {  
        Intent intentRecuperar = **new** Intent(Intent.ACTION\_VIEW);  
        intentRecuperar.setData(Uri.parse(**"http://google.com"**));  
        startActivity(intentRecuperar);  
        }  
        });  
        }  
          
       }
    3. **package** br.unaerp.compras.br.unaerp.compras.atividades;  
         
       **import** android.content.Intent;  
       **import** android.os.Bundle;  
       **import** android.os.Handler;  
       **import** android.support.v4.app.FragmentManager;  
       **import** android.support.design.widget.NavigationView;  
       **import** android.support.v4.view.GravityCompat;  
       **import** android.support.v4.widget.DrawerLayout;  
       **import** android.support.v7.app.ActionBarDrawerToggle;  
       **import** android.support.v7.app.AppCompatActivity;  
       **import** android.support.v7.widget.Toolbar;  
       **import** android.util.Log;  
       **import** android.view.MenuItem;  
       **import** android.widget.Toast;  
         
       **import** java.util.List;  
         
       **import** br.unaerp.compras.R;  
       **import** br.unaerp.compras.br.unaerp.compras.adapter.ClienteAdapter;  
       **import** br.unaerp.compras.br.unaerp.compras.connection.WebClient;  
       **import** br.unaerp.compras.br.unaerp.compras.dao.ClienteDAO;  
       **import** br.unaerp.compras.br.unaerp.compras.dao.FornecedorDAO;  
       **import** br.unaerp.compras.br.unaerp.compras.dao.LoginDAO;  
       **import** br.unaerp.compras.br.unaerp.compras.dao.ProdutoDAO;  
       **import** br.unaerp.compras.br.unaerp.compras.dao.UtilsDAO;  
       **import** br.unaerp.compras.br.unaerp.compras.fragment.ClienteFragment;  
       **import** br.unaerp.compras.br.unaerp.compras.fragment.DashboardFragment;  
       **import** br.unaerp.compras.br.unaerp.compras.fragment.FornecedorFragment;  
       **import** br.unaerp.compras.br.unaerp.compras.fragment.ProdutoFragment;  
       **import** br.unaerp.compras.br.unaerp.compras.fragment.VendaFragment;  
       **import** br.unaerp.compras.br.unaerp.compras.interfaces.OnFragmentInteractionListener;  
       **import** br.unaerp.compras.br.unaerp.compras.model.ClienteModel;  
       **import** br.unaerp.compras.br.unaerp.compras.tasks.BackupTask;  
         
       **public class** MainActivity **extends** AppCompatActivity  
        **implements** NavigationView.OnNavigationItemSelectedListener, OnFragmentInteractionListener {  
         
        **public static int** versaoBD = 12;  
        **boolean** duploClick = **false**;  
        FragmentManager manager = getSupportFragmentManager();  
         
        @Override  
        **protected void** onCreate(Bundle savedInstanceState) {  
        **super**.onCreate(savedInstanceState);  
        setContentView(R.layout.activity\_main\_drawer);  
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);  
        setSupportActionBar(toolbar);  
         
        DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer\_layout);  
        ActionBarDrawerToggle toggle = **new** ActionBarDrawerToggle(  
        **this**, drawer, toolbar, R.string.navigation\_drawer\_open, R.string.navigation\_drawer\_close);  
        drawer.addDrawerListener(toggle);  
        toggle.syncState();  
         
        NavigationView navigationView = (NavigationView) findViewById(R.id.nav\_view);  
        navigationView.setNavigationItemSelectedListener(**this**);  
         
         
        *//carregaBancoDeDados();* abrirDashboard();  
         
        }  
         
        **private void** carregaBancoDeDados() {  
        FornecedorDAO fdao = **new** FornecedorDAO(**this**, versaoBD);  
        ClienteDAO cdao = **new** ClienteDAO(**this**, versaoBD);  
        LoginDAO ldao = **new** LoginDAO(**this**, versaoBD);  
        UtilsDAO udao = **new** UtilsDAO(**this**, versaoBD);  
        ProdutoDAO pdao = **new** ProdutoDAO(**this**, versaoBD);  
         
         
        fdao.startaBD(**this**.versaoBD);  
        cdao.startaBD(**this**.versaoBD);  
        ldao.startaBD(**this**.versaoBD);  
        udao.startaBD(**this**.versaoBD);  
        pdao.startaBD(**this**.versaoBD);  
        }  
         
        @Override  
        **public void** onBackPressed() {  
         
        **if** (duploClick) {  
        **super**.onBackPressed();  
        **return**;  
        }  
        duploClick = **true**;  
        Toast.makeText(**this**, **"Clique novamente para sair.."**, Toast.LENGTH\_SHORT).show();  
         
        **new** Handler().postDelayed(**new** Runnable() {  
        @Override  
        **public void** run() {  
        duploClick = **false**;  
        }  
        }, 2000);  
         
        }  
         
       */\* @Override  
        public boolean onCreateOptionsMenu(Menu menu) {  
        // Inflate the menu; this adds items to the action bar if it is present.  
        getMenuInflater().inflate(R.menu.main\_drawer, menu);  
        return super.onCreateOptionsMenu(menu);  
        }\*/* @SuppressWarnings(**"StatementWithEmptyBody"**)  
        @Override  
        **public boolean** onNavigationItemSelected(MenuItem item) {  
        **int** id = item.getItemId();  
        **if** (id == R.id.nav\_dashboard) {  
        abrirDashboard();  
        } **else if** (id == R.id.nav\_clientes) {  
        abrirCliente();  
        } **else if** (id == R.id.nav\_produtos) {  
        abrirProduto();  
        } **else if** (id == R.id.nav\_fornecedor) {  
        abrirFornecedor();  
        } **else if** (id == R.id.nav\_vendas) {  
        abrirVendas();  
        } **else if** (id == R.id.nav\_sinc) {  
        metodoParaEnviarParaServidor();  
        } **else if** (id == R.id.nav\_sobre) {  
        abrirSobre();  
        }  
         
        DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer\_layout);  
        drawer.closeDrawer(GravityCompat.START);  
        **return true**;  
        }  
         
        */\*ABETURA DE TELAS\*/* **private void** abrirSobre() {  
        Intent i = **new** Intent(MainActivity.**this**, AboutActivity.**class**);  
        startActivity(i);  
        }  
         
        **private void** abrirVendas() {  
        VendaFragment vendaFrag = VendaFragment.newInstance();  
        **this**.manager.beginTransaction().replace(  
        R.id.linearlayout\_for\_fragment,  
        vendaFrag,  
        vendaFrag.getTag()  
        ).commit();  
        }  
         
        **private void** abrirFornecedor() {  
        FornecedorFragment fornecedorFrag = FornecedorFragment.newInstance();  
        **this**.manager.beginTransaction().replace(  
        R.id.linearlayout\_for\_fragment,  
        fornecedorFrag,  
        fornecedorFrag.getTag()  
        ).commit();  
        }  
         
        **private void** abrirProduto() {  
        ProdutoFragment produtoFrag = ProdutoFragment.newInstance();  
        **this**.manager.beginTransaction().replace(  
        R.id.linearlayout\_for\_fragment,  
        produtoFrag,  
        produtoFrag.getTag()  
        ).commit();  
        }  
         
        **private void** abrirCliente() {  
        ClienteFragment clienteFrag = ClienteFragment.newInstance();  
        manager.beginTransaction().replace(  
        R.id.linearlayout\_for\_fragment,  
        clienteFrag,  
        clienteFrag.getTag()  
        ).commit();  
        }  
         
        **private void** abrirDashboard() {  
        DashboardFragment dashFrag = DashboardFragment.newInstance();  
        **this**.manager.beginTransaction().replace(  
        R.id.linearlayout\_for\_fragment,  
        dashFrag,  
        dashFrag.getTag()  
        ).commit();  
        }  
         
        **private void** metodoParaEnviarParaServidor() {  
        **new** BackupTask(**this**).execute();  
        }  
         
         
        @Override  
        **protected void** onResume() {  
        **this**.setTitle(**""**);  
        **super**.onResume();  
        }  
         
        @Override  
        **public void** onFragmentInteraction(String title) {  
        getSupportActionBar().setTitle(title);  
         
        }  
       }

## Connection

* + 1. **package** br.unaerp.compras.br.unaerp.compras.connection;  
         
       **import** android.util.Log;  
         
       **import** java.io.IOException;  
       **import** java.io.PrintStream;  
       **import** java.net.HttpURLConnection;  
       **import** java.net.MalformedURLException;  
       **import** java.net.URL;  
       **import** java.net.URLConnection;  
       **import** java.util.Scanner;  
       **import** java.util.logging.Handler;  
       **import** java.util.logging.LogRecord;  
         
       **import** br.unaerp.compras.br.unaerp.compras.model.LoginModel;  
         
       */\*\*  
        \* Created by raul on 7/6/2016.  
        \*/***public class** WebClient {  
         
        **public** String post(String json) {  
        *//String uri = "http://10.0.2.2/sv/adiciona-cliente.php";* String uri = **"http://192.168.0.2:80/sv/adiciona-cliente.php"**;  
        **try** {  
        URL url = **new** URL(uri);  
        HttpURLConnection con = (HttpURLConnection) url.openConnection();  
        con.setConnectTimeout(2000);  
        con.setRequestProperty(**"Content-type"**,**"application/json"**);  
        con.setRequestProperty(**"Accept"**,**"\*/\*"**);  
         
        con.setDoOutput(**true**); *//faz o POST* PrintStream output = **new** PrintStream(con.getOutputStream());  
        output.println(json); *//insere no corpo* con.connect();  
         
        Scanner input = **new** Scanner(con.getInputStream());  
        String resposta = input.next();  
        **return** resposta;  
        } **catch** (MalformedURLException e) {  
        e.printStackTrace();  
        } **catch** (IOException e) {  
        e.printStackTrace();  
        }  
        **return null**;  
        }  
         
         
       }

## Dao

* + 1. **package** br.unaerp.compras.br.unaerp.compras.dao;  
         
       **import** android.content.ContentValues;  
       **import** android.content.Context;  
       **import** android.database.Cursor;  
       **import** android.database.sqlite.SQLiteDatabase;  
       **import** android.database.sqlite.SQLiteOpenHelper;  
         
       **import** java.util.ArrayList;  
       **import** java.util.List;  
         
       **import** br.unaerp.compras.br.unaerp.compras.atividades.MainActivity;  
       **import** br.unaerp.compras.br.unaerp.compras.model.ClienteModel;  
         
       */\*\*  
        \* Created by Raul on 10/04/2016.  
        \*/***public class** ClienteDAO **extends** SQLiteOpenHelper {  
         
        **public** ClienteDAO(Context context, **int** version) {  
        **super**(context, **"GestaoVendas"**, **null**, version);  
        }  
         
        @Override  
        **public void** onCreate(SQLiteDatabase db) {  
        String sql = **"create table clientes ("** +  
        **"id integer primary key autoincrement not null,"** +  
        **"nome text not null,"** +  
        **"sobrenome text not null,"** +  
        **"apelido text,"** +  
        **"telefone text,"** +  
        **"celular text,"** +  
        **"endereco text not null,"** +  
        **"email text);"**;  
         
        db.execSQL(sql);  
         
        }  
         
        @Override  
        **public void** onUpgrade(SQLiteDatabase db, **int** oldVersion, **int** newVersion) {  
        String sql = **"drop table if exists clientes;"**;  
        db.execSQL(sql);  
        onCreate(db);  
        }  
         
        **public void** insertCliente(ClienteModel cliente) {  
        SQLiteDatabase db = getWritableDatabase();*//escrever no db* ContentValues dados = getContentCliente(cliente);  
        db.insert(**"Clientes"**, **null**, dados);  
         
        }  
         
        **public** List<ClienteModel> selectCliente() {  
        String sql = **"select \* from clientes order by nome;"**;  
        SQLiteDatabase db = getReadableDatabase();  
        Cursor c\_Cliente = db.rawQuery(sql, **null**);  
        List<ClienteModel> clientes = **new** ArrayList<ClienteModel>();  
         
        **while** (c\_Cliente.moveToNext()) {  
        ClienteModel cliente = **new** ClienteModel();  
        cliente.setId(c\_Cliente.getLong(c\_Cliente.getColumnIndex(**"id"**)));  
        cliente.setNome(c\_Cliente.getString(c\_Cliente.getColumnIndex(**"nome"**)));  
        cliente.setSobrenome(c\_Cliente.getString(c\_Cliente.getColumnIndex(**"sobrenome"**)));  
        cliente.setApelido(c\_Cliente.getString(c\_Cliente.getColumnIndex(**"apelido"**)));  
        cliente.setTelefone(c\_Cliente.getString(c\_Cliente.getColumnIndex(**"telefone"**)));  
        cliente.setCelular(c\_Cliente.getString(c\_Cliente.getColumnIndex(**"celular"**)));  
        cliente.setEndereco(c\_Cliente.getString(c\_Cliente.getColumnIndex(**"endereco"**)));  
        cliente.setEmail(c\_Cliente.getString(c\_Cliente.getColumnIndex(**"email"**)));  
         
        clientes.add(cliente);  
        }  
        c\_Cliente.close();  
         
        **return** clientes;  
        }  
         
        **public void** deleteCliente(ClienteModel cliente) {  
        SQLiteDatabase db = getWritableDatabase();  
        String[] params = {cliente.getId().toString()};  
        db.delete(**"clientes"**, **"id = ?"**, params);  
        }  
         
        **public void** updateCliente(ClienteModel cliente) {  
        SQLiteDatabase db = getWritableDatabase();  
        ContentValues dados = getContentCliente(cliente);  
        String[] params = {cliente.getId().toString()};  
        db.update(**"clientes"**, dados, **"id = ?"**, params);  
        }  
         
        **private** ContentValues getContentCliente(ClienteModel cliente) {  
        ContentValues dados = **new** ContentValues();  
        dados.put(**"nome"**, cliente.getNome());  
        dados.put(**"sobrenome"**, cliente.getSobrenome());  
        dados.put(**"apelido"**, cliente.getApelido());  
        dados.put(**"telefone"**, cliente.getTelefone());  
        dados.put(**"celular"**, cliente.getCelular());  
        dados.put(**"endereco"**, cliente.getEndereco());  
        dados.put(**"email"**, cliente.getEmail());  
        **return** dados;  
        }  
         
        **public void** startaBD(**int** versaoBD) {  
        SQLiteDatabase db = getWritableDatabase();  
        **this**.onUpgrade(db,db.getVersion(),versaoBD);  
        }  
       }
    2. **package** br.unaerp.compras.br.unaerp.compras.dao;  
         
       **import** android.content.ContentValues;  
       **import** android.content.Context;  
       **import** android.database.Cursor;  
       **import** android.database.sqlite.SQLiteDatabase;  
       **import** android.database.sqlite.SQLiteOpenHelper;  
         
       **import** java.util.ArrayList;  
       **import** java.util.List;  
         
       **import** br.unaerp.compras.br.unaerp.compras.atividades.MainActivity;  
       **import** br.unaerp.compras.br.unaerp.compras.model.FornecedorModel;  
         
       */\*\*  
        \* Created by Raul on 10/04/2016.  
        \*/***public class** FornecedorDAO **extends** SQLiteOpenHelper {  
         
        **public** FornecedorDAO(Context context, **int** version) {  
        **super**(context, **"GestaoVendas"**, **null**, version);  
        }  
         
        @Override  
        **public void** onCreate(SQLiteDatabase db) {  
        String sql = **"create table fornecedor ("** +  
        **"id integer primary key autoincrement not null,"** +  
        **"nomefantasia text not null,"** +  
        **"cnpj text not null,"** +  
        **"endereco text not null,"**+  
        **"numero text not null,"**+  
        **"complemento text,"**+  
        **"bairro text,"**+  
        **"cep text,"**+  
        **"cidade text not null,"**+  
        **"estado text not null,"**+  
        **"telefone text,"**+  
        **"site text,"**+  
        **"email text);"**;  
        db.execSQL(sql);  
        sql = **"insert into fornecedor (nomefantasia, cnpj, endereco,numero,cidade,estado) values ('Doce Festa', '1032132', 'ruasuhashusa','1213','Ribeirão Preto','SP');"**;  
        db.execSQL(sql);  
         
        }  
         
        @Override  
        **public void** onUpgrade(SQLiteDatabase db, **int** oldVersion, **int** newVersion) {  
        String sql = **"drop table if exists fornecedor;"**;  
        db.execSQL(sql);  
        onCreate(db);  
        }  
         
        **public void** insertFornecedor(FornecedorModel fornecedor) {  
        SQLiteDatabase db = getWritableDatabase();*//escrever no db* ContentValues dados = getContentFornecedor(fornecedor);  
        db.insert(**"Fornecedor"**, **null**, dados);  
         
        }  
         
        **public** List<FornecedorModel> selectFornecedor() {  
        String sql = **"select \* from fornecedor order by nomefantasia;"**;  
        SQLiteDatabase db = getReadableDatabase();  
        Cursor c\_Fornecedor = db.rawQuery(sql, **null**);  
        List<FornecedorModel> fornecedores = **new** ArrayList<FornecedorModel>();  
         
        **while** (c\_Fornecedor.moveToNext()) {  
        FornecedorModel fornecedor = **new** FornecedorModel();  
        fornecedor.setId(c\_Fornecedor.getLong(c\_Fornecedor.getColumnIndex(**"id"**)));  
        fornecedor.setNomeFantasia(c\_Fornecedor.getString(c\_Fornecedor.getColumnIndex(**"nomefantasia"**)));  
        fornecedor.setCnpj(c\_Fornecedor.getString(c\_Fornecedor.getColumnIndex(**"cnpj"**)));  
        fornecedor.setEndereco(c\_Fornecedor.getString(c\_Fornecedor.getColumnIndex(**"endereco"**)));  
        fornecedor.setNumero(c\_Fornecedor.getString(c\_Fornecedor.getColumnIndex(**"numero"**)));  
        fornecedor.setComplemento(c\_Fornecedor.getString(c\_Fornecedor.getColumnIndex(**"complemento"**)));  
        fornecedor.setBairro(c\_Fornecedor.getString(c\_Fornecedor.getColumnIndex(**"bairro"**)));  
        fornecedor.setCep(c\_Fornecedor.getString(c\_Fornecedor.getColumnIndex(**"cep"**)));  
        fornecedor.setCidade(c\_Fornecedor.getString(c\_Fornecedor.getColumnIndex(**"cidade"**)));  
        fornecedor.setEstado(c\_Fornecedor.getString(c\_Fornecedor.getColumnIndex(**"estado"**)));  
        fornecedor.setTelefone(c\_Fornecedor.getString(c\_Fornecedor.getColumnIndex(**"telefone"**)));  
        fornecedor.setSite(c\_Fornecedor.getString(c\_Fornecedor.getColumnIndex(**"site"**)));  
        fornecedor.setEmail(c\_Fornecedor.getString(c\_Fornecedor.getColumnIndex(**"email"**)));  
         
        fornecedores.add(fornecedor);  
        }  
        c\_Fornecedor.close();  
         
        **return** fornecedores;  
        }  
         
        **public void** deleteFornecedor(FornecedorModel fornecedor) {  
        SQLiteDatabase db = getWritableDatabase();  
         
        String[] params = {fornecedor.getId().toString()};  
        db.delete(**"fornecedor"**, **"id = ?"**, params);  
        }  
         
        **public void** updateFornecedor(FornecedorModel fornecedor) {  
        SQLiteDatabase db = getWritableDatabase();  
        ContentValues dados = getContentFornecedor(fornecedor);  
        String[] params = {fornecedor.getId().toString()};  
        db.update(**"fornecedor"**, dados, **"id = ?"**, params);  
        }  
         
        **private** ContentValues getContentFornecedor(FornecedorModel fornecedor) {  
        ContentValues dados = **new** ContentValues();  
        dados.put(**"nomefantasia"**, fornecedor.getNomeFantasia());  
        dados.put(**"cnpj"**,fornecedor.getCnpj());  
        dados.put(**"endereco"**,fornecedor.getEndereco());  
        dados.put(**"numero"**,fornecedor.getNumero());  
        dados.put(**"complemento"**,fornecedor.getComplemento());  
        dados.put(**"bairro"**,fornecedor.getBairro());  
        dados.put(**"cep"**,fornecedor.getCep());  
        dados.put(**"cidade"**,fornecedor.getCidade());  
        dados.put(**"estado"**,fornecedor.getEstado());  
        dados.put(**"telefone"**,fornecedor.getTelefone());  
        dados.put(**"site"**,fornecedor.getSite());  
        dados.put(**"email"**,fornecedor.getEmail());  
         
        **return** dados;  
        }  
         
        **public void** startaBD(**int** newVersion){  
        SQLiteDatabase db = getWritableDatabase();  
        **this**.onUpgrade(db,db.getVersion(),newVersion);  
        }  
       }
    3. **package** br.unaerp.compras.br.unaerp.compras.dao;  
         
       **import** android.content.Context;  
       **import** android.database.Cursor;  
       **import** android.database.sqlite.SQLiteDatabase;  
       **import** android.database.sqlite.SQLiteOpenHelper;  
         
       **import** java.util.ArrayList;  
       **import** java.util.List;  
         
       **import** br.unaerp.compras.br.unaerp.compras.atividades.LoginActivity;  
       **import** br.unaerp.compras.br.unaerp.compras.model.ClienteModel;  
       **import** br.unaerp.compras.br.unaerp.compras.model.LoginModel;  
         
       */\*\*  
        \* Created by Raul on 13/04/2016.  
        \*/***public class** LoginDAO **extends** SQLiteOpenHelper {  
        **public** LoginDAO(Context context, **int** version) {  
        **super**(context, **"GestaoVendas"**, **null**, version);  
        }  
         
         
         
        @Override  
        **public void** onCreate(SQLiteDatabase db) {  
        *//cria tabela de login* String sql = **"create table usuario ("** +  
        **" id integer primary key autoincrement not null,"** +  
        **"nomecompleto text not null,"**+  
        **"sexo char not null,"**+  
        **"email text not null,"**+  
        **" usuario text not null,"** +  
        **" senha text not null,"** +  
        **"data\_cadastro date not null);"**;  
        db.execSQL(sql);  
        *//cria usuario default* sql = **"insert into usuario (nomecompleto,sexo,email,usuario, senha, data\_cadastro) values ('Raul Felipe de Melo','M','raulfdm@hotmail.com','a','a',date('now'));"**;  
        db.execSQL(sql);  
        }  
         
        @Override  
        **public void** onUpgrade(SQLiteDatabase db, **int** oldVersion, **int** newVersion) {  
        String sql = **"drop table if exists usuario;"**;  
        db.execSQL(sql);  
        onCreate(db);  
        }  
         
        **public void** insertUsuario() {  
        *//Pendente* }  
         
        **public** LoginModel validaAcesso(LoginModel usuario) {  
         
        SQLiteDatabase db = getReadableDatabase();  
        String sql = **"select nomecompleto, email from usuario where usuario = '"**+usuario.getUsuario() +**"' and senha = '"**+usuario.getSenha()+**"';"**;  
        */\*String[] colunas = {"usuario", "senha"};  
        String[] login = {usuario.getUsuario(), usuario.getSenha()};  
        Cursor c\_login = db.query("usuario", colunas, "usuario = ? and senha = ?", login, null, null, null);\*/  
         
        /\*if (c\_login.getCount() > 0) {  
        validado = true;  
        }\*/* Cursor c\_login = db.rawQuery(sql, **null**);  
        **while** (c\_login.moveToNext()) {  
        usuario.setNomeCompleto(c\_login.getString(c\_login.getColumnIndex(**"nomecompleto"**)));  
        usuario.setEmail(c\_login.getString(c\_login.getColumnIndex(**"email"**)));  
        }  
         
        **return** usuario;  
        }  
         
        **public void** startaBD(**int** versaoBD) {  
        SQLiteDatabase db = getWritableDatabase();  
        **this**.onUpgrade(db,db.getVersion(),versaoBD);  
        }  
       }
    4. **package** br.unaerp.compras.br.unaerp.compras.dao;  
         
       **import** android.content.ContentValues;  
       **import** android.content.Context;  
       **import** android.database.Cursor;  
       **import** android.database.sqlite.SQLiteDatabase;  
       **import** android.database.sqlite.SQLiteOpenHelper;  
         
       **import** java.util.ArrayList;  
       **import** java.util.List;  
         
       **import** br.unaerp.compras.br.unaerp.compras.model.ProdutoModel;  
         
       */\*\*  
        \* Created by raul on 17/6/2016.  
        \*/***public class** ProdutoDAO **extends** SQLiteOpenHelper{  
         
        **public** ProdutoDAO(Context context, **int** version) {  
        **super**(context, **"GestaoVendas"**, **null**, version);  
        }  
         
        @Override  
        **public void** onCreate(SQLiteDatabase db) {  
        String sql = **"create table produto ("** +  
        **"id integer primary key autoincrement not null,"** +  
        **"codigo text not null,"** +  
        **"marca text not null,"** +  
        **"descr text not null,"** +  
        **"fornecedor text not null,"** +  
        **"tamanho text not null);"**;  
         
        db.execSQL(sql);  
         
        }  
         
        @Override  
        **public void** onUpgrade(SQLiteDatabase db, **int** oldVersion, **int** newVersion) {  
        String sql = **"drop table if exists produto;"**;  
        db.execSQL(sql);  
        onCreate(db);  
        }  
        **public void** insertProduto(ProdutoModel produto) {  
        SQLiteDatabase db = getWritableDatabase();*//escrever no db* ContentValues dados = getContentProduto(produto);  
        db.insert(**"produto"**, **null**, dados);  
         
        }  
         
        **public** List<ProdutoModel> selectProdutos() {  
        String sql = **"select \* from produto order by codigo;"**;  
        SQLiteDatabase db = getReadableDatabase();  
        Cursor c\_produto = db.rawQuery(sql, **null**);  
        List<ProdutoModel> produtos = **new** ArrayList<ProdutoModel>();  
         
        **while** (c\_produto.moveToNext()) {  
        ProdutoModel produto = **new** ProdutoModel();  
        produto.setId(c\_produto.getLong(c\_produto.getColumnIndex(**"id"**)));  
        produto.setFornecedor(c\_produto.getString(c\_produto.getColumnIndex(**"fornecedor"**)));  
        produto.setCodigo(c\_produto.getString(c\_produto.getColumnIndex(**"codigo"**)));  
        produto.setMarca(c\_produto.getString(c\_produto.getColumnIndex(**"marca"**)));  
        produto.setDescricao(c\_produto.getString(c\_produto.getColumnIndex(**"descr"**)));  
        produto.setTamanho(c\_produto.getString(c\_produto.getColumnIndex(**"tamanho"**)));  
         
        produtos.add(produto);  
        }  
        c\_produto.close();  
         
        **return** produtos;  
        }  
         
        **public void** updateProduto(ProdutoModel produto) {  
        SQLiteDatabase db = getWritableDatabase();  
        ContentValues dados = getContentProduto(produto);  
        String[] params = {produto.getId().toString()};  
        db.update(**"produto"**, dados, **"id = ?"**, params);  
        }  
         
        **private** ContentValues getContentProduto(ProdutoModel produto) {  
        ContentValues dados = **new** ContentValues();  
        dados.put(**"codigo"**, produto.getCodigo());  
        dados.put(**"marca"**, produto.getMarca());  
        dados.put(**"descr"**, produto.getDescricao());  
        dados.put(**"fornecedor"**, produto.getFornecedor());  
        dados.put(**"tamanho"**, produto.getTamanho());  
         
        **return** dados;  
        }  
         
        **public void** startaBD(**int** versaoBD) {  
        SQLiteDatabase db = getWritableDatabase();  
        **this**.onUpgrade(db,db.getVersion(),versaoBD);  
        }  
         
        **public** List<String> selectFornecedores(){  
        String sql = **"select nomefantasia from fornecedor order by nomefantasia;"**;  
        SQLiteDatabase db = getReadableDatabase();  
        Cursor c\_fornecedor = db.rawQuery(sql,**null**);  
        List<String> fornecedores = **new** ArrayList<>();  
         
        **while**(c\_fornecedor.moveToNext()){  
        String estado = c\_fornecedor.getString(c\_fornecedor.getColumnIndex(**"nomefantasia"**));  
        fornecedores.add(estado);  
        }  
        c\_fornecedor.close();  
         
        **return** fornecedores;  
        }  
         
        **public boolean** deleteProduto(ProdutoModel produto) {  
        **try** {  
        SQLiteDatabase db = getWritableDatabase();  
        String[] params = {produto.getId().toString()};  
        db.delete(**"produto"**, **"id = ?"**, params);  
        **return true**;  
        }**catch** (ClassCastException e){  
        **return false**;  
        }  
         
        }  
       }
    5. **package** br.unaerp.compras.br.unaerp.compras.dao;  
         
       **import** android.content.Context;  
       **import** android.database.Cursor;  
       **import** android.database.sqlite.SQLiteDatabase;  
       **import** android.database.sqlite.SQLiteOpenHelper;  
         
       **import** java.util.ArrayList;  
       **import** java.util.List;  
         
       */\*\*  
        \* Created by raul on 15/6/2016.  
        \*/***public class** UtilsDAO **extends** SQLiteOpenHelper {  
        **public** UtilsDAO(Context context, **int** version) {  
        **super**(context, **"GestaoVendas"**, **null**, version);  
        }  
         
         
        @Override  
        **public void** onCreate(SQLiteDatabase db) {  
        String sql = **"create table estados("**+  
        **"id integer primary key autoincrement not null,"**+  
        **"estado text not null,"**+  
        **"abreviacao text not null);"**;  
         
        db.execSQL(sql);  
        ArrayList<String> estados = **new** ArrayList();  
        estados.add(**"insert into estados (estado, abreviacao) values ('Acre','AC');"**);  
        estados.add(**"insert into estados (estado, abreviacao) values ('Alagoas','AL');"**);  
        estados.add(**"insert into estados (estado, abreviacao) values ('Amapá','AP');"**);  
        estados.add(**"insert into estados (estado, abreviacao) values ('Amazonas','AM');"**);  
        estados.add(**"insert into estados (estado, abreviacao) values ('Bahia','BA');"**);  
        estados.add(**"insert into estados (estado, abreviacao) values ('Ceará','CE');"**);  
        estados.add(**"insert into estados (estado, abreviacao) values ('Distrito Federal','DF');"**);  
        estados.add(**"insert into estados (estado, abreviacao) values ('Espírito Santo','ES');"**);  
        estados.add(**"insert into estados (estado, abreviacao) values ('Goiás','GO');"**);  
        estados.add(**"insert into estados (estado, abreviacao) values ('Maranhão','MA');"**);  
        estados.add(**"insert into estados (estado, abreviacao) values ('Mato Grosso','MT');"**);  
        estados.add(**"insert into estados (estado, abreviacao) values ('Mato Grosso do Sul','MS');"**);  
        estados.add(**"insert into estados (estado, abreviacao) values ('Minas Gerais','MG');"**);  
        estados.add(**"insert into estados (estado, abreviacao) values ('Pará','PA');"**);  
        estados.add(**"insert into estados (estado, abreviacao) values ('Paraíba','PB');"**);  
        estados.add(**"insert into estados (estado, abreviacao) values ('Paraná','PR');"**);  
        estados.add(**"insert into estados (estado, abreviacao) values ('Pernambuco','PE');"**);  
        estados.add(**"insert into estados (estado, abreviacao) values ('Piauí','PI');"**);  
        estados.add(**"insert into estados (estado, abreviacao) values ('Rio de Janeiro','RJ');"**);  
        estados.add(**"insert into estados (estado, abreviacao) values ('Rio Grande do Norte','RN');"**);  
        estados.add(**"insert into estados (estado, abreviacao) values ('Rio Grande do Sul','RS');"**);  
        estados.add(**"insert into estados (estado, abreviacao) values ('Rondônia','RO');"**);  
        estados.add(**"insert into estados (estado, abreviacao) values ('Roraima','RR');"**);  
        estados.add(**"insert into estados (estado, abreviacao) values ('Santa Catarina','SC');"**);  
        estados.add(**"insert into estados (estado, abreviacao) values ('São Paulo','SP');"**);  
        estados.add(**"insert into estados (estado, abreviacao) values ('Sergipe','SE');"**);  
        estados.add(**"insert into estados (estado, abreviacao) values ('Tocantins','TO');"**);  
         
         
        **for** (String estado: estados) {  
        sql = estado;  
        db.execSQL(sql);  
        }  
         
        }  
         
        @Override  
        **public void** onUpgrade(SQLiteDatabase db, **int** oldVersion, **int** newVersion) {  
        String sql = **"drop table if exists estados;"**;  
        db.execSQL(sql);  
        onCreate(db);  
        }  
         
        **public** List<String> selectEstado(){  
        String sql = **"select \* from estados order by abreviacao;"**;  
        SQLiteDatabase db = getReadableDatabase();  
        Cursor c\_estado = db.rawQuery(sql,**null**);  
        List<String> estados = **new** ArrayList<>();  
         
        **while**(c\_estado.moveToNext()){  
        String estado = c\_estado.getString(c\_estado.getColumnIndex(**"abreviacao"**));  
        estados.add(estado);  
        }  
        c\_estado.close();  
         
        **return** estados;  
        }  
         
        **public void** startaBD(**int** versaoBD) {  
        SQLiteDatabase db = getWritableDatabase();  
        **this**.onUpgrade(db,db.getVersion(),versaoBD);  
        }  
       }

## Forms

* + 1. **package** br.unaerp.compras.br.unaerp.compras.forms;  
         
         
       **import** android.annotation.TargetApi;  
       **import** android.content.Intent;  
       **import** android.os.Build;  
       **import** android.os.Bundle;  
       **import** android.support.v7.app.AppCompatActivity;  
       **import** android.support.v7.widget.Toolbar;  
       **import** android.telephony.PhoneNumberFormattingTextWatcher;  
       **import** android.view.Menu;  
       **import** android.view.MenuInflater;  
       **import** android.view.MenuItem;  
       **import** android.widget.EditText;  
       **import** android.widget.Toast;  
         
       **import** br.unaerp.compras.R;  
       **import** br.unaerp.compras.br.unaerp.compras.atividades.MainActivity;  
       **import** br.unaerp.compras.br.unaerp.compras.dao.ClienteDAO;  
       **import** br.unaerp.compras.br.unaerp.compras.helper.FormularioClienteHelper;  
       **import** br.unaerp.compras.br.unaerp.compras.model.ClienteModel;  
         
       **public class** FormularioCliente **extends** AppCompatActivity {  
         
        **private** FormularioClienteHelper **helperCliente**;  
        **private** EditText **campoTelefone**;  
        **private** EditText **campoCelular**;  
         
        @TargetApi(Build.VERSION\_CODES.***LOLLIPOP***)  
        @Override  
        **protected void** onCreate(Bundle savedInstanceState) {  
        **super**.onCreate(savedInstanceState);  
        setContentView(R.layout.***activity\_formulario\_cliente***);  
         
        Toolbar toolbar = (Toolbar) findViewById(R.id.***toolbar***);  
        setSupportActionBar(toolbar);  
         
        **helperCliente** = **new** FormularioClienteHelper(**this**);  
        */\*Formatar Campos\*/* **campoTelefone** = (EditText) findViewById(R.id.***form\_cliente\_telefone***);  
        **campoTelefone**.addTextChangedListener(**new** PhoneNumberFormattingTextWatcher(**"BR"**));  
        **campoCelular** = (EditText) findViewById(R.id.***form\_cliente\_celular***);  
        **campoCelular**.addTextChangedListener(**new** PhoneNumberFormattingTextWatcher(**"BR"**));  
         
        */\*Caso seja Edição\*/* Intent editaCliente = getIntent();  
        ClienteModel cliente = (ClienteModel) editaCliente.getSerializableExtra(**"cliente"**);  
        **if** (cliente != **null**) {  
        **helperCliente**.preencheFormulario(cliente);  
        }  
         
        }  
         
         
        @Override  
        **public boolean** onCreateOptionsMenu(Menu menu) {  
        getSupportActionBar().setDisplayHomeAsUpEnabled(**true**);  
        getSupportActionBar().setDisplayShowHomeEnabled(**true**);  
        MenuInflater menuInflater = getMenuInflater();  
        menuInflater.inflate(R.menu.***form\_cliente\_menu***, menu);  
        **return super**.onCreateOptionsMenu(menu);  
        }  
         
         
        @Override  
        **public boolean** onOptionsItemSelected(MenuItem item) {  
        **switch** (item.getItemId()) {  
        **case** R.id.***form\_salvar***:  
         
        String formValidado = **helperCliente**.validaFormulario();  
         
        **if**(formValidado != **null**){  
        Toast.makeText(FormularioCliente.**this**, **"Preencha o campo "**+formValidado, Toast.***LENGTH\_SHORT***).show();  
        }**else**{  
        ClienteModel cliente = **helperCliente**.getCliente();  
        ClienteDAO dao = **new** ClienteDAO(**this**, MainActivity.*versaoBD*);  
        **if** (cliente.getId() != **null**) {  
        dao.updateCliente(cliente);  
        Toast.makeText(**this**, String.*valueOf*(**"Cliente "** + cliente.getNome() + **" Alterado com Sucesso!"**), Toast.***LENGTH\_SHORT***).show();  
        } **else** {  
        dao.insertCliente(cliente);  
        Toast.makeText(**this**, String.*valueOf*(**"Cliente "** + cliente.getNome() + **" Salvo com Sucesso!"**), Toast.***LENGTH\_SHORT***).show();  
        }  
        dao.close();  
        finish();  
        }  
         
        **break**;  
        **case** android.R.id.***home***:  
        finish();  
        **break**;  
        }  
         
        **return super**.onOptionsItemSelected(item);  
        }  
         
       }

**package** br.unaerp.compras.br.unaerp.compras.forms;  
  
**import** android.annotation.TargetApi;  
**import** android.content.Intent;  
**import** android.os.Build;  
**import** android.os.Bundle;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.support.v7.widget.Toolbar;  
**import** android.view.Menu;  
**import** android.view.MenuInflater;  
**import** android.view.MenuItem;  
**import** android.widget.ArrayAdapter;  
**import** android.widget.EditText;  
**import** android.widget.Spinner;  
**import** android.widget.Toast;  
  
**import** java.util.List;  
  
**import** br.unaerp.compras.R;  
**import** br.unaerp.compras.br.unaerp.compras.atividades.MainActivity;  
**import** br.unaerp.compras.br.unaerp.compras.dao.FornecedorDAO;  
**import** br.unaerp.compras.br.unaerp.compras.dao.UtilsDAO;  
**import** br.unaerp.compras.br.unaerp.compras.helper.FormularioFornecedorHelper;  
**import** br.unaerp.compras.br.unaerp.compras.model.FornecedorModel;  
**import** br.unaerp.compras.br.unaerp.compras.utils.Mask;  
  
**public class** FormularioFornecedor **extends** AppCompatActivity {  
  
  
 **private** FormularioFornecedorHelper **helperFornecedor**;  
 *// private EditText campoTelefone;  
 //private EditText campoCnpj;* **private** Spinner **spinnerEstados**;  
  
 @TargetApi(Build.VERSION\_CODES.***LOLLIPOP***)  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_formulario\_fornecedor***);  
  
 Toolbar toolbar = (Toolbar) findViewById(R.id.***toolbar***);  
 setSupportActionBar(toolbar);  
  
 */\*Formatar Campos\*//\*  
 campoTelefone = (EditText) findViewById(R.id.form\_fornecedor\_telefone);  
 campoTelefone.addTextChangedListener(Mask.insert(Mask.TELEFONE\_MASK,campoTelefone));  
 campoCnpj = (EditText) findViewById(R.id.form\_fornecedor\_cnpj);  
 campoCnpj.addTextChangedListener(Mask.insert(Mask.CNPJ\_MASK,campoCnpj));\*/* **helperFornecedor** = **new** FormularioFornecedorHelper(**this**);  
 ArrayAdapter<String> adapter = carregarEstados();  
  
 Intent editaFornecedor = getIntent();  
 FornecedorModel fornecedor = (FornecedorModel) editaFornecedor.getSerializableExtra(**"fornecedor"**);  
 **if** (fornecedor != **null**) {  
 **helperFornecedor**.preencheFormulario(fornecedor,adapter);  
 }  
 }  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 getSupportActionBar().setDisplayHomeAsUpEnabled(**true**);  
 getSupportActionBar().setDisplayShowHomeEnabled(**true**);  
 MenuInflater menuInflater = getMenuInflater();  
 menuInflater.inflate(R.menu.***form\_cliente\_menu***, menu);  
 **return super**.onCreateOptionsMenu(menu);  
 }  
  
 **private** ArrayAdapter<String> carregarEstados() {  
 UtilsDAO dao = **new** UtilsDAO(**this**, MainActivity.*versaoBD*);  
 List<String> estados = dao.selectEstado();  
 **spinnerEstados** = (Spinner) findViewById(R.id.***form\_fornecedor\_estados***);  
 ArrayAdapter<String> adapter = **new** ArrayAdapter<String>(**this**, android.R.layout.***simple\_spinner\_dropdown\_item***, estados);  
 **spinnerEstados**.setAdapter(adapter);  
  
 **return** adapter;  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **switch** (item.getItemId()) {  
 **case** R.id.***form\_salvar***:  
  
 String formValidado = **helperFornecedor**.validaFormulario();  
  
 **if**(formValidado != **null**){  
 Toast.makeText(FormularioFornecedor.**this**, **"Preencha o campo "**+formValidado, Toast.***LENGTH\_SHORT***).show();  
 }**else**{  
 FornecedorModel fornecedor = **helperFornecedor**.getFornecedor();  
 FornecedorDAO dao = **new** FornecedorDAO(**this**, MainActivity.*versaoBD*);  
 **if** (fornecedor.getId() != **null**) {  
 dao.updateFornecedor(fornecedor);  
 Toast.makeText(**this**, String.*valueOf*(**"Fornecedor "** + fornecedor.getNomeFantasia() + **" alterado com Sucesso!"**), Toast.***LENGTH\_SHORT***).show();  
 } **else** {  
 dao.insertFornecedor(fornecedor);  
 Toast.makeText(**this**, String.*valueOf*(**"Fornecedor "** + fornecedor.getNomeFantasia() + **" salvo com Sucesso!"**), Toast.***LENGTH\_SHORT***).show();  
 }  
 dao.close();  
 finish();  
 }  
 **break**;  
 **case** android.R.id.***home***:  
 finish();  
 **break**;  
 }  
  
 **return super**.onOptionsItemSelected(item);  
 }  
}

* + 1. **package** br.unaerp.compras.br.unaerp.compras.forms;  
         
       **import** android.content.Intent;  
       **import** android.os.Bundle;  
       **import** android.support.v7.app.AppCompatActivity;  
       **import** android.support.v7.widget.Toolbar;  
       **import** android.view.Menu;  
       **import** android.view.MenuInflater;  
       **import** android.view.MenuItem;  
       **import** android.widget.ArrayAdapter;  
       **import** android.widget.Spinner;  
       **import** android.widget.Toast;  
         
       **import** java.util.List;  
         
       **import** br.unaerp.compras.R;  
       **import** br.unaerp.compras.br.unaerp.compras.atividades.MainActivity;  
       **import** br.unaerp.compras.br.unaerp.compras.dao.ProdutoDAO;  
       **import** br.unaerp.compras.br.unaerp.compras.helper.ProdutoHelper;  
       **import** br.unaerp.compras.br.unaerp.compras.model.ProdutoModel;  
         
       **public class** FormularioProduto **extends** AppCompatActivity {  
         
        **private** ProdutoHelper **helperProduto**;  
        **private** Spinner **spinnerFornecedor**;  
        **private** Spinner **spinnerTamanho**;  
         
        @Override  
        **protected void** onCreate(Bundle savedInstanceState) {  
        **super**.onCreate(savedInstanceState);  
        setContentView(R.layout.***activity\_formulario\_produto***);  
         
        Toolbar toolbar = (Toolbar) findViewById(R.id.***toolbar***);  
        setSupportActionBar(toolbar);  
         
        **helperProduto** = **new** ProdutoHelper(**this**);  
         
         
        ArrayAdapter<CharSequence> adaptarTamanho = carregaTamanhos();  
        ArrayAdapter<String> adapterFornecedor = carregarFornecedor();  
         
        Intent editaProduto = getIntent();  
        ProdutoModel produto = (ProdutoModel) editaProduto.getSerializableExtra(**"produto"**);  
        **if** (produto != **null**) {  
        **this**.**helperProduto**.preencheFormulario(produto, adaptarTamanho, adapterFornecedor);  
        }  
        }  
         
        @Override  
        **public boolean** onCreateOptionsMenu(Menu menu) {  
        getSupportActionBar().setDisplayHomeAsUpEnabled(**true**);  
        getSupportActionBar().setDisplayShowHomeEnabled(**true**);  
        MenuInflater menuInflater = getMenuInflater();  
        menuInflater.inflate(R.menu.***form\_cliente\_menu***, menu);  
        **return super**.onCreateOptionsMenu(menu);  
        }  
         
        @Override  
        **public boolean** onOptionsItemSelected(MenuItem item) {  
        **switch** (item.getItemId()) {  
        **case** R.id.***form\_salvar***:  
        String formValidado = **helperProduto**.validaFormulario();  
        **if** (formValidado != **"sucesso"**) {  
        Toast.makeText(FormularioProduto.**this**, **"Preencha o campo "** + formValidado, Toast.***LENGTH\_SHORT***).show();  
        } **else** {  
        ProdutoModel produto = **helperProduto**.getProduto();  
        ProdutoDAO dao = **new** ProdutoDAO(**this**, MainActivity.*versaoBD*);  
        **if** (produto.getId() != **null**) {  
        dao.updateProduto(produto);  
        Toast.makeText(**this**, String.*valueOf*(**"Produto "** + produto.getDescricao() + **" Alterado com Sucesso!"**), Toast.***LENGTH\_SHORT***).show();  
        } **else** {  
        dao.insertProduto(produto);  
        Toast.makeText(**this**, String.*valueOf*(**"Produto "** + produto.getDescricao() + **" Salvo com Sucesso!"**), Toast.***LENGTH\_SHORT***).show();  
        }  
        dao.close();  
        finish();  
        }  
         
        **break**;  
        **case** android.R.id.***home***:  
        finish();  
        **break**;  
        }  
         
        **return super**.onOptionsItemSelected(item);  
        }  
         
        **private** ArrayAdapter<String> carregarFornecedor() {  
        ProdutoDAO dao = **new** ProdutoDAO(**this**, MainActivity.*versaoBD*);  
        List<String> fornecedores = dao.selectFornecedores();  
        **spinnerFornecedor** = (Spinner) findViewById(R.id.***form\_produto\_fornecedor***);  
        ArrayAdapter<String> adapter = **new** ArrayAdapter<String>(**this**, android.R.layout.***simple\_spinner\_dropdown\_item***, fornecedores);  
        **spinnerFornecedor**.setAdapter(adapter);  
        **return** adapter;  
        }  
         
        **private** ArrayAdapter<CharSequence> carregaTamanhos() {  
        **spinnerTamanho** = (Spinner) findViewById(R.id.***form\_produto\_tamanho***);  
         
        ArrayAdapter<CharSequence> adapter = ArrayAdapter.*createFromResource*(**this**,  
        R.array.***produto\_tamanho***, android.R.layout.***simple\_spinner\_item***);  
         
        adapter.setDropDownViewResource(android.R.layout.***simple\_spinner\_dropdown\_item***);  
         
        **spinnerTamanho**.setAdapter(adapter);  
        **return** adapter;  
        }  
       }

## Fragment

* + 1. **package** br.unaerp.compras.br.unaerp.compras.fragment;  
         
         
       **import** android.content.Intent;  
       **import** android.net.Uri;  
       **import** android.os.Bundle;  
       **import** android.support.v4.app.Fragment;  
       **import** android.view.ContextMenu;  
       **import** android.view.LayoutInflater;  
       **import** android.view.MenuItem;  
       **import** android.view.View;  
       **import** android.view.ViewGroup;  
       **import** android.widget.AdapterView;  
       **import** android.widget.ArrayAdapter;  
       **import** android.widget.ListView;  
         
       **import** com.github.clans.fab.FloatingActionButton;  
         
       **import** java.util.List;  
         
       **import** br.unaerp.compras.R;  
       **import** br.unaerp.compras.br.unaerp.compras.atividades.MainActivity;  
       **import** br.unaerp.compras.br.unaerp.compras.dao.ClienteDAO;  
       **import** br.unaerp.compras.br.unaerp.compras.forms.FormularioCliente;  
       **import** br.unaerp.compras.br.unaerp.compras.interfaces.OnFragmentInteractionListener;  
       **import** br.unaerp.compras.br.unaerp.compras.model.ClienteModel;  
         
       **public class** ClienteFragment **extends** Fragment {  
         
        **private** ListView **listaClientes**;  
        **private** View **viewPrincipal**;  
        **private** OnFragmentInteractionListener **mListener**;  
         
         
        **public** ClienteFragment() {  
        *// Required empty public constructor* }  
         
        **public static** ClienteFragment newInstance() {  
        ClienteFragment fragment = **new** ClienteFragment();  
        Bundle args = **new** Bundle();  
        fragment.setArguments(args);  
        **return** fragment;  
        }  
         
        @Override  
        **public void** onCreate(Bundle savedInstanceState) {  
        **try** {  
        **mListener** = (OnFragmentInteractionListener) **super**.getActivity();  
        } **catch** (ClassCastException e) {  
        **throw new** ClassCastException(**super**.getActivity().toString()  
        + **" must implement OnFragmentInteractionListener"**);  
        }  
        **super**.onCreate(savedInstanceState);  
        }  
         
         
         
        @Override  
        **public** View onCreateView(LayoutInflater inflater, ViewGroup container,  
        Bundle savedInstanceState) {  
         
        **if** (**mListener** != **null**) {  
        **mListener**.onFragmentInteraction(getString(R.string.***title\_fragment\_clientes***));  
        }  
         
        **this**.**viewPrincipal** = inflater.inflate(R.layout.***fragment\_cliente***,container,**false**);  
        carregaListaClientes();  
         
        com.github.clans.fab.FloatingActionButton adicionaCliente = (FloatingActionButton) **viewPrincipal**.findViewById(R.id.***ativ\_cliente\_add***);  
        adicionaCliente.setOnClickListener(**new** View.OnClickListener() {  
        @Override  
        **public void** onClick(View v) {  
        Intent i = **new** Intent(ClienteFragment.**this**.getActivity(), FormularioCliente.**class**);  
        startActivity(i);  
        }  
        });  
         
        **return viewPrincipal**;  
        }  
         
        @Override  
        **public void** onResume() {  
        carregaListaClientes();  
        registerForContextMenu(**listaClientes**);  
         
        **super**.onResume();  
        }  
         
        **private void** carregaListaClientes() {  
        ClienteDAO dao = **new** ClienteDAO(**this**.getActivity(),MainActivity.*versaoBD*);  
        List<ClienteModel> clientes = dao.selectCliente();  
        **listaClientes** = (ListView) **viewPrincipal**.findViewById(R.id.***view\_cliente\_lista***);  
        ArrayAdapter<ClienteModel> adapter = **new** ArrayAdapter<ClienteModel>(**this**.getActivity(), android.R.layout.***simple\_list\_item\_1***, clientes);  
        **listaClientes**.setAdapter(adapter);  
         
        **listaClientes**.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  
        @Override  
        **public void** onItemClick(AdapterView<?> lista, View item, **int** position, **long** id) {  
        ClienteModel cliente = (ClienteModel) **listaClientes**.getItemAtPosition(position);  
        Intent editaCliente = **new** Intent(ClienteFragment.**this**.getActivity(), FormularioCliente.**class**);  
        editaCliente.putExtra(**"cliente"**, cliente);  
        startActivity(editaCliente);  
        }  
        });  
        }  
         
        @Override  
        **public void** onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo) {  
        */\*Instancia o objeto cliente\*/* AdapterView.AdapterContextMenuInfo info = (AdapterView.AdapterContextMenuInfo) menuInfo;  
        **final** ClienteModel cliente = (ClienteModel) **listaClientes**.getItemAtPosition(info.**position**);  
         
        */\*LISTA DE MENUS\*/* MenuItem mandaEmail = menu.add(**"Enviar e-mail"**); *//Cria a opção no menu* MenuItem deletar = menu.add(**"Deletar"**);  
         
        */\*Envia E-mail através do e-mail cadastrado\*/* String emailCliente = **"mailto:"** + cliente.getEmail();  
        Intent intentEmail = **new** Intent(Intent.***ACTION\_SENDTO***); *//ativa a ação de enviar e-mail* intentEmail.setType(**"message/rfc822"**);  
        intentEmail.putExtra(Intent.***EXTRA\_EMAIL***, emailCliente);  
        intentEmail.putExtra(Intent.***EXTRA\_SUBJECT***, **""**);  
        intentEmail.setData(Uri.*parse*(emailCliente));  
        mandaEmail.setIntent(intentEmail);  
         
        */\*Deletar contato\*/* deletar.setOnMenuItemClickListener(**new** MenuItem.OnMenuItemClickListener() {  
        @Override  
        **public boolean** onMenuItemClick(MenuItem item) {  
        ClienteDAO dao = **new** ClienteDAO(ClienteFragment.**this**.getActivity(),MainActivity.*versaoBD*);  
        dao.deleteCliente(cliente);  
        dao.close();  
        carregaListaClientes();  
        **return false**;  
        }  
        });  
         
        }  
         
         
       }
    2. **package** br.unaerp.compras.br.unaerp.compras.fragment;  
         
         
       **import** android.os.Bundle;  
       **import** android.support.v4.app.Fragment;  
       **import** android.view.LayoutInflater;  
       **import** android.view.View;  
       **import** android.view.ViewGroup;  
       **import** android.widget.ListView;  
         
       **import** br.unaerp.compras.R;  
       **import** br.unaerp.compras.br.unaerp.compras.interfaces.OnFragmentInteractionListener;  
         
       **public class** DashboardFragment **extends** Fragment {  
        **private** ListView **listaFornecedores**;  
        **private** View **viewPrincipal**;  
        **private** OnFragmentInteractionListener **mListener**;  
         
        **public** DashboardFragment() {  
        *// Required empty public constructor* }  
         
        **public static** DashboardFragment newInstance() {  
        DashboardFragment fragment = **new** DashboardFragment();  
        Bundle args = **new** Bundle();  
        fragment.setArguments(args);  
        **return** fragment;  
        }  
         
        @Override  
        **public void** onCreate(Bundle savedInstanceState) {  
        **try** {  
        **mListener** = (OnFragmentInteractionListener) **super**.getActivity();  
        } **catch** (ClassCastException e) {  
        **throw new** ClassCastException(**super**.getActivity().toString()  
        + **" must implement OnFragmentInteractionListener"**);  
        }  
        **super**.onCreate(savedInstanceState);  
        }  
         
        @Override  
        **public** View onCreateView(LayoutInflater inflater, ViewGroup container,  
        Bundle savedInstanceState) {  
        **if** (**mListener** != **null**) {  
        **mListener**.onFragmentInteraction(getString(R.string.***title\_fragment\_dashboard***));  
        }  
         
        **this**.**viewPrincipal** = inflater.inflate(R.layout.***fragment\_dashboard***, container, **false**);  
         
        **return viewPrincipal**;  
        }  
         
       }
    3. **package** br.unaerp.compras.br.unaerp.compras.fragment;  
         
         
       **import** android.content.Intent;  
       **import** android.net.Uri;  
       **import** android.os.Bundle;  
       **import** android.support.v4.app.Fragment;  
       **import** android.view.ContextMenu;  
       **import** android.view.LayoutInflater;  
       **import** android.view.MenuItem;  
       **import** android.view.View;  
       **import** android.view.ViewGroup;  
       **import** android.widget.AdapterView;  
       **import** android.widget.ArrayAdapter;  
       **import** android.widget.ListView;  
       **import** android.widget.Toast;  
         
       **import** com.github.clans.fab.FloatingActionButton;  
         
       **import** java.util.List;  
         
       **import** br.unaerp.compras.R;  
       **import** br.unaerp.compras.br.unaerp.compras.atividades.MainActivity;  
       **import** br.unaerp.compras.br.unaerp.compras.dao.ClienteDAO;  
       **import** br.unaerp.compras.br.unaerp.compras.dao.FornecedorDAO;  
       **import** br.unaerp.compras.br.unaerp.compras.forms.FormularioFornecedor;  
       **import** br.unaerp.compras.br.unaerp.compras.interfaces.OnFragmentInteractionListener;  
       **import** br.unaerp.compras.br.unaerp.compras.model.ClienteModel;  
       **import** br.unaerp.compras.br.unaerp.compras.model.FornecedorModel;  
         
         
       **public class** FornecedorFragment **extends** Fragment {  
         
        **private** ListView **listaFornecedores**;  
        **private** View viewPrincipal;  
        **private** OnFragmentInteractionListener **mListener**;  
         
        **public** FornecedorFragment() {  
        }  
         
         
        **public static** FornecedorFragment newInstance() {  
        FornecedorFragment fragment = **new** FornecedorFragment();  
        Bundle args = **new** Bundle();  
        fragment.setArguments(args);  
        **return** fragment;  
        }  
         
        @Override  
        **public void** onCreate(Bundle savedInstanceState) {  
        **try** {  
        **mListener** = (OnFragmentInteractionListener) **super**.getActivity();  
        } **catch** (ClassCastException e) {  
        **throw new** ClassCastException(**super**.getActivity().toString()  
        + **" must implement OnFragmentInteractionListener"**);  
        }  
        **super**.onCreate(savedInstanceState);  
         
        }  
         
        @Override  
        **public** View onCreateView(LayoutInflater inflater, ViewGroup container,  
        Bundle savedInstanceState) {  
        **if** (**mListener** != **null**) {  
        **mListener**.onFragmentInteraction(getString(R.string.***title\_fragment\_fornecedores***));  
        }  
         
        **this**.viewPrincipal = inflater.inflate(R.layout.***fragment\_fornecedor***, container, **false**);  
        carregaListaFornecedores();  
         
        com.github.clans.fab.FloatingActionButton adicionaFornecedor = (FloatingActionButton) viewPrincipal.findViewById(R.id.***fornecedor\_add***);  
        adicionaFornecedor.setOnClickListener(**new** View.OnClickListener() {  
        @Override  
        **public void** onClick(View v) {  
        Intent i = **new** Intent(FornecedorFragment.**this**.getActivity(), FormularioFornecedor.**class**);  
        startActivity(i);  
        }  
        });  
         
         
        **return** viewPrincipal;  
        }  
         
        @Override  
        **public void** onResume() {  
        carregaListaFornecedores();  
        registerForContextMenu(**listaFornecedores**);  
         
        **super**.onResume();  
        }  
         
        **private void** carregaListaFornecedores() {  
        FornecedorDAO dao = **new** FornecedorDAO(**this**.getActivity(), MainActivity.*versaoBD*);  
        List<FornecedorModel> fornecedores = dao.selectFornecedor();  
        **listaFornecedores** = (ListView) viewPrincipal.findViewById(R.id.***view\_fornecedor\_lista***);  
        ArrayAdapter<FornecedorModel> adapter = **new** ArrayAdapter<FornecedorModel>(**this**.getActivity(), android.R.layout.***simple\_list\_item\_1***, fornecedores);  
        **listaFornecedores**.setAdapter(adapter);  
         
        **listaFornecedores**.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  
        @Override  
        **public void** onItemClick(AdapterView<?> lista, View view, **int** position, **long** id) {  
        FornecedorModel fornecedor = (FornecedorModel) **listaFornecedores**.getItemAtPosition(position);  
        Intent editaFornecedor = **new** Intent(FornecedorFragment.**this**.getActivity(), FormularioFornecedor.**class**);  
        editaFornecedor.putExtra(**"fornecedor"**, fornecedor);  
        startActivity(editaFornecedor);  
        }  
        });  
        }  
         
        @Override  
        **public void** onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo) {  
        AdapterView.AdapterContextMenuInfo info = (AdapterView.AdapterContextMenuInfo) menuInfo;  
        **final** FornecedorModel fornecedor = (FornecedorModel) **listaFornecedores**.getItemAtPosition(info.**position**);  
         
        */\*LISTA DE MENUS\*/* MenuItem mandaEmail = menu.add(**"Enviar e-mail"**);  
        MenuItem deletar = menu.add(**"Deletar"**);  
         
        */\*Envia E-mail através do e-mail cadastrado\*/* String emailFornecedor = **"mailto:"** + fornecedor.getEmail();  
        Intent intentEmail = **new** Intent(Intent.***ACTION\_SENDTO***); *//ativa a ação de enviar e-mail* intentEmail.setType(**"message/rfc822"**);  
        intentEmail.putExtra(Intent.***EXTRA\_EMAIL***, emailFornecedor);  
        intentEmail.putExtra(Intent.***EXTRA\_SUBJECT***, **""**);  
        intentEmail.setData(Uri.*parse*(emailFornecedor));  
        mandaEmail.setIntent(intentEmail);  
         
         
        */\*Deletar contato\*/* deletar.setOnMenuItemClickListener(**new** MenuItem.OnMenuItemClickListener() {  
        @Override  
        **public boolean** onMenuItemClick(MenuItem item) {  
        FornecedorDAO dao = **new** FornecedorDAO(FornecedorFragment.**this**.getActivity(), MainActivity.*versaoBD*);  
        dao.deleteFornecedor(fornecedor);  
        dao.close();  
        carregaListaFornecedores();  
        **return false**;  
        }  
        });  
        }  
         
       }
    4. **package** br.unaerp.compras.br.unaerp.compras.fragment;  
         
         
       **import** android.content.Intent;  
       **import** android.os.Bundle;  
       **import** android.support.v4.app.Fragment;  
       **import** android.view.ContextMenu;  
       **import** android.view.LayoutInflater;  
       **import** android.view.MenuItem;  
       **import** android.view.View;  
       **import** android.view.ViewGroup;  
       **import** android.widget.AdapterView;  
       **import** android.widget.ArrayAdapter;  
       **import** android.widget.ListView;  
       **import** android.widget.Toast;  
         
       **import** com.github.clans.fab.FloatingActionButton;  
         
       **import** java.util.List;  
         
       **import** br.unaerp.compras.R;  
       **import** br.unaerp.compras.br.unaerp.compras.atividades.MainActivity;  
       **import** br.unaerp.compras.br.unaerp.compras.dao.ProdutoDAO;  
       **import** br.unaerp.compras.br.unaerp.compras.forms.FormularioProduto;  
       **import** br.unaerp.compras.br.unaerp.compras.interfaces.OnFragmentInteractionListener;  
       **import** br.unaerp.compras.br.unaerp.compras.model.ProdutoModel;  
         
         
       **public class** ProdutoFragment **extends** Fragment {  
         
        **private** ListView **listaProdutos**;  
        **private** View **viewPrincipal**;  
        **private** OnFragmentInteractionListener **mListener**;  
         
        **public** ProdutoFragment() {  
        *// Required empty public constructor* }  
         
        **public static** ProdutoFragment newInstance() {  
        ProdutoFragment fragment = **new** ProdutoFragment();  
        Bundle args = **new** Bundle();  
        fragment.setArguments(args);  
        **return** fragment;  
        }  
         
        @Override  
        **public void** onCreate(Bundle savedInstanceState) {  
        **try** {  
        **mListener** = (OnFragmentInteractionListener) **super**.getActivity();  
        } **catch** (ClassCastException e) {  
        **throw new** ClassCastException(**super**.getActivity().toString()  
        + **" must implement OnFragmentInteractionListener"**);  
        }  
        **super**.onCreate(savedInstanceState);  
         
        }  
         
        @Override  
        **public** View onCreateView(LayoutInflater inflater, ViewGroup container,  
        Bundle savedInstanceState) {  
         
        **if** (**mListener** != **null**) {  
        **mListener**.onFragmentInteraction(getString(R.string.***title\_fragment\_produtos***));  
        }  
         
        **this**.**viewPrincipal** = inflater.inflate(R.layout.***fragment\_produto***, container, **false**);  
        carregaListaDeProdutos();  
         
        com.github.clans.fab.FloatingActionButton adicionaFornecedor = (FloatingActionButton) **viewPrincipal**.findViewById(R.id.***produto\_add***);  
        adicionaFornecedor.setOnClickListener(**new** View.OnClickListener() {  
        @Override  
        **public void** onClick(View v) {  
        Intent i = **new** Intent(ProdutoFragment.**this**.getActivity(), FormularioProduto.**class**);  
        startActivity(i);  
        }  
        });  
         
         
        **return viewPrincipal**;  
        }  
         
        **private void** carregaListaDeProdutos() {  
        ProdutoDAO dao = **new** ProdutoDAO(**this**.getActivity(), MainActivity.*versaoBD*);  
        List<ProdutoModel> produtos = dao.selectProdutos();  
        **listaProdutos** = (ListView) **viewPrincipal**.findViewById(R.id.***view\_produto\_lista***);  
        ArrayAdapter<ProdutoModel> adapter = **new** ArrayAdapter<ProdutoModel>(**this**.getActivity(), android.R.layout.***simple\_list\_item\_1***, produtos);  
        **listaProdutos**.setAdapter(adapter);  
         
        **listaProdutos**.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  
        @Override  
        **public void** onItemClick(AdapterView<?> lista, View item, **int** position, **long** id) {  
        ProdutoModel produto = (ProdutoModel) **listaProdutos**.getItemAtPosition(position);  
        Intent editaProduto = **new** Intent(ProdutoFragment.**this**.getActivity(), FormularioProduto.**class**);  
        editaProduto.putExtra(**"produto"**, produto);  
        startActivity(editaProduto);  
        }  
        });  
        }  
         
        @Override  
        **public void** onResume() {  
        carregaListaDeProdutos();  
        registerForContextMenu(**listaProdutos**);  
         
        **super**.onResume();  
        }  
         
        @Override  
        **public void** onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo) {  
        */\*Instancia o objeto cliente\*/* AdapterView.AdapterContextMenuInfo info = (AdapterView.AdapterContextMenuInfo) menuInfo;  
        **final** ProdutoModel produto = (ProdutoModel) **listaProdutos**.getItemAtPosition(info.**position**);  
         
        */\*LISTA DE MENUS\*/* MenuItem deletar = menu.add(**"Deletar"**);  
         
        */\*Deletar produto\*/* deletar.setOnMenuItemClickListener(**new** MenuItem.OnMenuItemClickListener() {  
        @Override  
        **public boolean** onMenuItemClick(MenuItem item) {  
        ProdutoDAO dao = **new** ProdutoDAO(ProdutoFragment.**this**.getActivity(),MainActivity.*versaoBD*);  
        String nomeProduto = produto.getDescricao();  
        **if**(dao.deleteProduto(produto)){  
        Toast.makeText(ProdutoFragment.**this**.getActivity(), **"Produto "**+nomeProduto + **" removido com sucesso"**, Toast.***LENGTH\_SHORT***).show();  
        }**else**{  
        Toast.makeText(ProdutoFragment.**this**.getActivity(), **"Erro ao excluir o produto"**, Toast.***LENGTH\_SHORT***).show();  
        }  
        dao.close();  
        carregaListaDeProdutos();  
        **return false**;  
        }  
        });  
         
        }  
         
       }
    5. **package** br.unaerp.compras.br.unaerp.compras.fragment;  
         
       **import** android.content.Intent;  
       **import** android.os.Bundle;  
       **import** android.support.v4.app.Fragment;  
       **import** android.view.LayoutInflater;  
       **import** android.view.View;  
       **import** android.view.ViewGroup;  
         
       **import** com.github.clans.fab.FloatingActionButton;  
         
       **import** br.unaerp.compras.R;  
       **import** br.unaerp.compras.br.unaerp.compras.forms.FormularioFornecedor;  
       **import** br.unaerp.compras.br.unaerp.compras.interfaces.OnFragmentInteractionListener;  
         
         
       **public class** VendaFragment **extends** Fragment {  
         
         
        **private** View **viewPrincipal**;  
        **private** OnFragmentInteractionListener **mListener**;  
         
        **public** VendaFragment() {  
        *// Required empty public constructor* }  
         
         
        **public static** VendaFragment newInstance() {  
        VendaFragment fragment = **new** VendaFragment();  
        Bundle args = **new** Bundle();  
        fragment.setArguments(args);  
        **return** fragment;  
        }  
         
        @Override  
        **public void** onCreate(Bundle savedInstanceState) {  
        **try** {  
        **mListener** = (OnFragmentInteractionListener) **super**.getActivity();  
        } **catch** (ClassCastException e) {  
        **throw new** ClassCastException(**super**.getActivity().toString()  
        + **" must implement OnFragmentInteractionListener"**);  
        }  
        **super**.onCreate(savedInstanceState);  
         
        }  
         
        @Override  
        **public** View onCreateView(LayoutInflater inflater, ViewGroup container,  
        Bundle savedInstanceState) {  
        **if** (**mListener** != **null**) {  
        **mListener**.onFragmentInteraction(getString(R.string.***title\_fragment\_vendas***));  
        }  
         
        **this**.**viewPrincipal** = inflater.inflate(R.layout.***fragment\_venda***, container, **false**);  
         
        com.github.clans.fab.FloatingActionButton adicionaFornecedor = (FloatingActionButton) **viewPrincipal**.findViewById(R.id.***venda\_add***);  
        adicionaFornecedor.setOnClickListener(**new** View.OnClickListener() {  
        @Override  
        **public void** onClick(View v) {  
        Intent i = **new** Intent(VendaFragment.**this**.getActivity(), FormularioFornecedor.**class**);  
        startActivity(i);  
        }  
        });  
         
         
        **return viewPrincipal**;  
        }  
         
       }

## Helper

* + 1. **package** br.unaerp.compras.br.unaerp.compras.helper;  
         
       **import** android.widget.EditText;  
         
       **import** br.unaerp.compras.R;  
       **import** br.unaerp.compras.br.unaerp.compras.forms.FormularioCliente;  
       **import** br.unaerp.compras.br.unaerp.compras.model.ClienteModel;  
         
       */\*\*  
        \* Created by Raul on 10/04/2016.  
        \*/***public class** FormularioClienteHelper {  
         
        **private final** EditText **campoNome**;  
        **private final** EditText **campoSobrenome**;  
        **private final** EditText **campoTelefone**;  
        **private final** EditText **campoApelido**;  
        **private final** EditText **campoCelular**;  
        **private final** EditText **campoEndereco**;  
        **private final** EditText **campoEmail**;  
         
        **private** ClienteModel **cliente**;  
         
        **public** FormularioClienteHelper(FormularioCliente activity) {  
         
        **campoNome** = (EditText) activity.findViewById(R.id.***form\_cliente\_nome***);  
        **campoSobrenome** = (EditText) activity.findViewById(R.id.***form\_cliente\_sobrenome***);  
        **campoApelido** = (EditText) activity.findViewById(R.id.***form\_cliente\_apelido***);  
        **campoTelefone** = (EditText) activity.findViewById(R.id.***form\_cliente\_telefone***);  
        **campoCelular** = (EditText) activity.findViewById(R.id.***form\_cliente\_celular***);  
        **campoEndereco** = (EditText) activity.findViewById(R.id.***form\_cliente\_endereco***);  
        **campoEmail** = (EditText) activity.findViewById(R.id.***form\_cliente\_email***);  
         
        **cliente** = **new** ClienteModel();  
        }  
         
        **public** ClienteModel getCliente() {  
        **cliente**.setNome(**campoNome**.getText().toString());  
        **cliente**.setSobrenome(**campoSobrenome**.getText().toString());  
        **cliente**.setApelido(**campoApelido**.getText().toString());  
        **cliente**.setTelefone(**campoTelefone**.getText().toString());  
        **cliente**.setCelular(**campoCelular**.getText().toString());  
        **cliente**.setEndereco(**campoEndereco**.getText().toString());  
        **cliente**.setEmail(**campoEmail**.getText().toString());  
         
        **return cliente**;  
        }  
         
        **public void** preencheFormulario(ClienteModel cliente) {  
        **campoNome**.setText(cliente.getNome());  
        **campoSobrenome**.setText(cliente.getSobrenome());  
        **campoApelido**.setText(cliente.getApelido());  
        **campoTelefone**.setText(cliente.getTelefone());  
        **campoCelular**.setText(cliente.getCelular());  
        **campoEndereco**.setText(cliente.getEndereco());  
        **campoEmail**.setText(cliente.getEmail());  
        **this**.**cliente** = cliente;  
        }  
         
         
        **public** String validaFormulario() {  
        **if** (**campoNome**.getText().toString() == **null** || **campoNome**.getText().toString().isEmpty()) {  
        **return "nome"**;  
        } **else if**(**campoSobrenome**.getText().toString() == **null** || **campoSobrenome**.getText().toString().isEmpty()){  
        **return "sobrenome"**;  
        }**else if**(**campoTelefone**.getText().toString() == **null** || **campoTelefone**.getText().toString().isEmpty() &&  
        (**campoCelular**.getText().toString() == **null** || **campoCelular**.getText().toString().isEmpty())){  
        **return "telefone ou celular"**;  
        }  
         
        **return null**;  
        }  
       }
    2. **package** br.unaerp.compras.br.unaerp.compras.helper;  
         
       **import** android.widget.ArrayAdapter;  
       **import** android.widget.EditText;  
       **import** android.widget.Spinner;  
         
       **import** br.unaerp.compras.R;  
       **import** br.unaerp.compras.br.unaerp.compras.forms.FormularioFornecedor;  
       **import** br.unaerp.compras.br.unaerp.compras.model.FornecedorModel;  
         
       */\*\*  
        \* Created by Raul on 10/04/2016.  
        \*/***public class** FormularioFornecedorHelper {  
         
        **private final** EditText **campoNomeFantasia**;  
        **private final** EditText **campoCnpj**;  
        **private final** EditText **campoEndereco**;  
        **private final** EditText **campoNumero**;  
        **private final** EditText **campoComplemento**;  
        **private final** EditText **campoBairro**;  
        **private final** Spinner **campoEstado**;  
        **private final** EditText **campoCidade**;  
        **private final** EditText **campoTelefone**;  
        **private final** EditText **campoEmail**;  
        **private final** EditText **campoSite**;  
         
         
        **private** FornecedorModel **fornecedor**;  
         
        **public** FormularioFornecedorHelper(FormularioFornecedor activity) {  
        **campoNomeFantasia** = (EditText) activity.findViewById(R.id.***form\_fornecedor\_nomefantasia***);  
        **campoCnpj** = (EditText) activity.findViewById(R.id.***form\_fornecedor\_cnpj***);  
        **campoEndereco** = (EditText) activity.findViewById(R.id.***form\_fornecedor\_endereco***);  
        **campoNumero** = (EditText) activity.findViewById(R.id.***form\_fornecedor\_num***);  
        **campoComplemento** = (EditText) activity.findViewById(R.id.***form\_fornecedor\_complemento***);  
        **campoBairro** = (EditText) activity.findViewById(R.id.***form\_fornecedor\_bairro***);  
        **campoEstado** = (Spinner) activity.findViewById(R.id.***form\_fornecedor\_estados***);  
        **campoCidade** = (EditText) activity.findViewById(R.id.***form\_fornecedor\_cidade***);  
        **campoTelefone** = (EditText) activity.findViewById(R.id.***form\_fornecedor\_telefone***);  
        **campoEmail** = (EditText) activity.findViewById(R.id.***form\_fornecedor\_email***);  
        **campoSite** = (EditText) activity.findViewById(R.id.***form\_fornecedor\_site***);  
         
        **fornecedor** = **new** FornecedorModel();  
         
        }  
         
        **public** FornecedorModel getFornecedor() {  
        **fornecedor**.setNomeFantasia(**campoNomeFantasia**.getText().toString());  
        **fornecedor**.setCnpj(**campoCnpj**.getText().toString());  
        **fornecedor**.setEndereco(**campoEndereco**.getText().toString());  
        **fornecedor**.setNumero(**campoNumero**.getText().toString());  
        **fornecedor**.setComplemento(**campoComplemento**.getText().toString());  
        **fornecedor**.setBairro(**campoBairro**.getText().toString());  
        **fornecedor**.setEstado(**campoEstado**.getSelectedItem().toString());  
        **fornecedor**.setCidade(**campoCidade**.getText().toString());  
        **fornecedor**.setTelefone(**campoTelefone**.getText().toString());  
        **fornecedor**.setEmail(**campoEmail**.getText().toString());  
        **fornecedor**.setSite(**campoSite**.getText().toString());  
         
        **return fornecedor**;  
        }  
         
        **public void** preencheFormulario(FornecedorModel fornecedor, ArrayAdapter<String> estados) {  
        **campoNomeFantasia**.setText(fornecedor.getNomeFantasia());  
        **campoCnpj**.setText(fornecedor.getCnpj());  
        **campoEndereco**.setText(fornecedor.getEndereco());  
        **campoNumero**.setText(fornecedor.getNumero());  
        **campoComplemento**.setText(fornecedor.getComplemento());  
        **campoBairro**.setText(fornecedor.getBairro());  
        **campoEstado**.setSelection(estados.getPosition(fornecedor.getEstado()));  
        **campoCidade**.setText(fornecedor.getCidade());  
        **campoTelefone**.setText(fornecedor.getTelefone());  
        **campoEmail**.setText(fornecedor.getEmail());  
        **campoSite**.setText(fornecedor.getSite());  
        **this**.**fornecedor** = fornecedor;  
        }  
         
         
         
        **public** String validaFormulario() {  
        **if** (**campoNomeFantasia**.getText().toString() == **null** || **campoNomeFantasia**.getText().toString().isEmpty()) {  
        **return "Nome Fantasia"**;  
        } **else if** (**campoCnpj**.getText().toString() == **null** || **campoCnpj**.getText().toString().isEmpty()) {  
        **return "CNPJ"**;  
        } **else if** (**campoEndereco**.getText().toString() == **null** || **campoEndereco**.getText().toString().isEmpty()) {  
        **return "endereço"**;  
        } **else if** (**campoNumero**.getText().toString() == **null** || **campoNumero**.getText().toString().isEmpty()) {  
        **return "número"**;  
        } **else if** (**campoCidade**.getText().toString() == **null** || **campoCidade**.getText().toString().isEmpty()) {  
        **return "cidade"**;  
        }  
         
        **return null**;  
        }  
       }
    3. **package** br.unaerp.compras.br.unaerp.compras.helper;  
         
       **import** android.widget.EditText;  
         
       **import** br.unaerp.compras.br.unaerp.compras.atividades.LoginActivity;  
       **import** br.unaerp.compras.R;  
       **import** br.unaerp.compras.br.unaerp.compras.model.LoginModel;  
         
       */\*\*  
        \* Created by Raul on 13/04/2016.  
        \*/***public class** LoginHelper {  
         
        **private final** EditText **campoUsuario**;  
        **private final** EditText **campoSenha**;  
        **private** LoginModel **usuario**;  
         
        **public** LoginHelper(LoginActivity activity){  
        **campoUsuario** = (EditText) activity.findViewById(R.id.***login\_nome***);  
        **campoSenha** = (EditText) activity.findViewById(R.id.***login\_senha***);  
        **usuario** = **new** LoginModel();  
        }  
         
        **public** LoginModel getUsuario(){  
        **usuario**.setUsuario(**campoUsuario**.getText().toString());  
        **usuario**.setSenha(**campoSenha**.getText().toString());  
         
        **return usuario**;  
        }  
         
        **public** String validaFormLogin(){  
        **if**(**campoUsuario**.getText().toString() == **null** || **campoUsuario**.getText().toString().isEmpty()){  
        **return "Usuário ou E-mail"**;  
        }**else if**(**campoSenha**.getText().toString() == **null** || **campoSenha**.getText().toString().isEmpty()){  
        **return "Senha"**;  
        }  
         
        **return null**;  
        }  
       }
    4. **package** br.unaerp.compras.br.unaerp.compras.helper;  
         
       **import** android.widget.ArrayAdapter;  
       **import** android.widget.EditText;  
       **import** android.widget.Spinner;  
         
       **import** br.unaerp.compras.R;  
       **import** br.unaerp.compras.br.unaerp.compras.forms.FormularioProduto;  
       **import** br.unaerp.compras.br.unaerp.compras.model.ProdutoModel;  
         
       */\*\*  
        \* Created by raul on 17/6/2016.  
        \*/***public class** ProdutoHelper {  
         
        **private final** Spinner **campoFornecedor**;  
        **private final** EditText **campoCodigo**;  
        **private final** EditText **campoMarca**;  
        **private final** EditText **campoDescricao**;  
        **private final** Spinner **campoTamanho**;  
         
        **private** ProdutoModel **produto**;  
         
        **public** ProdutoHelper(FormularioProduto activity) {  
         
        **campoFornecedor** = (Spinner) activity.findViewById(R.id.***form\_produto\_fornecedor***);  
        **campoCodigo** = (EditText) activity.findViewById(R.id.***form\_produto\_cod***);  
        **campoMarca** = (EditText) activity.findViewById(R.id.***form\_produto\_marca***);  
        **campoDescricao** = (EditText) activity.findViewById(R.id.***form\_produto\_descr***);  
        **campoTamanho** = (Spinner) activity.findViewById(R.id.***form\_produto\_tamanho***);  
         
        **produto** = **new** ProdutoModel();  
        }  
         
        **public** ProdutoModel getProduto() {  
        **produto**.setFornecedor(**campoFornecedor**.getSelectedItem().toString());  
        **produto**.setCodigo(**campoCodigo**.getText().toString());  
        **produto**.setMarca(**campoMarca**.getText().toString());  
        **produto**.setDescricao(**campoDescricao**.getText().toString());  
        **produto**.setTamanho(**campoTamanho**.getSelectedItem().toString());  
         
        **return produto**;  
        }  
         
        **public void** preencheFormulario(ProdutoModel produto, ArrayAdapter<CharSequence> tamanho, ArrayAdapter<String> fornecedor) {  
        **campoFornecedor**.setSelection(fornecedor.getPosition(produto.getFornecedor()));  
        **campoCodigo**.setText(produto.getCodigo());  
        **campoMarca**.setText(produto.getMarca());  
        **campoDescricao**.setText(produto.getDescricao());  
        **campoTamanho**.setSelection(tamanho.getPosition(produto.getTamanho()));  
        **this**.**produto** = produto;  
        }  
         
        **public** String validaFormulario() {  
        **if** (**campoCodigo**.getText().toString() == **null** || **campoCodigo**.getText().toString().isEmpty()) {  
        **return "Código do produto"**;  
        } **else if** (**campoDescricao**.getText().toString() == **null** || **campoDescricao**.getText().toString().isEmpty()) {  
        **return "descrição do produto"**;  
        } **else if** (**campoMarca**.getText().toString() == **null** || **campoMarca**.getText().toString().isEmpty()) {  
        **return "marca do produto"**;  
        }  
         
        **return "sucesso"**;  
        }  
         
         
       }

## Interfaces

* + 1. **package** br.unaerp.compras.br.unaerp.compras.interfaces;  
         
       */\*\*  
        \* Created by raul on 14/6/2016.  
        \*/***public interface** OnFragmentInteractionListener {  
        **public void** onFragmentInteraction(String title);  
       }

## Model

* + 1. **package** br.unaerp.compras.br.unaerp.compras.model;  
         
       **import** android.support.v4.app.FragmentActivity;  
         
       */\*\*  
        \* Created by raul on 19/6/2016.  
        \*/***public class** BackupModel {  
        **private** FragmentActivity context;  
        **private** ClienteModel cliente;  
        **private** ProdutoModel produto;  
        **private** FornecedorModel fornecedor;  
         
        **public** BackupModel() {  
        }  
         
        **public void** setContext(FragmentActivity context) {  
        **this**.context = context;  
        }  
         
        **public void** setCliente(ClienteModel cliente) {  
        **this**.cliente = cliente;  
        }  
         
        **public void** setProduto(ProdutoModel produto) {  
        **this**.produto = produto;  
        }  
         
        **public void** setFornecedor(FornecedorModel fornecedor) {  
        **this**.fornecedor = fornecedor;  
        }  
         
        **public** FragmentActivity getContext() {  
        **return** context;  
        }  
         
        **public** ClienteModel getCliente() {  
        **return** cliente;  
        }  
         
        **public** ProdutoModel getProduto() {  
        **return** produto;  
        }  
         
        **public** FornecedorModel getFornecedor() {  
        **return** fornecedor;  
        }  
       }
    2. **package** br.unaerp.compras.br.unaerp.compras.model;  
         
       **import** java.io.Serializable;  
         
       */\*\*  
        \* Created by Raul on 10/04/2016.  
        \*/***public class** ClienteModel **implements** Serializable {  
         
        **private** Long **id**;  
        **private** String **nome**;  
        **private** String **sobrenome**;  
        **private** String **apelido**;  
        **private** String **telefone**;  
        **private** String **celular**;  
        **private** String **endereco**;  
        **private** String **email**;  
         
        **public** ClienteModel() {  
        }  
         
        **public** Long getId() {  
        **return id**;  
        }  
         
        **public void** setId(Long id) {  
        **this**.**id** = id;  
        }  
         
        **public** String getNome() {  
        **return nome**;  
        }  
         
        **public void** setNome(String nome) {  
        **this**.**nome** = nome;  
        }  
         
        **public** String getSobrenome() {  
        **return sobrenome**;  
        }  
         
        **public void** setSobrenome(String sobrenome) {  
        **this**.**sobrenome** = sobrenome;  
        }  
         
        **public** String getApelido() {  
        **return apelido**;  
        }  
         
        **public void** setApelido(String apelido) {  
        **this**.**apelido** = apelido;  
        }  
         
        **public** String getTelefone() {  
        **return telefone**;  
        }  
         
        **public void** setTelefone(String telefone) {  
        **this**.**telefone** = telefone;  
        }  
         
        **public** String getCelular() {  
        **return celular**;  
        }  
         
        **public void** setCelular(String celular) {  
        **this**.**celular** = celular;  
        }  
         
        **public** String getEndereco() {  
        **return endereco**;  
        }  
         
        **public void** setEndereco(String endereco) {  
        **this**.**endereco** = endereco;  
        }  
         
        **public** String getEmail() {  
        **return email**;  
        }  
         
        **public void** setEmail(String email) {  
        **this**.**email** = email;  
        }  
         
        @Override  
        **public** String toString() {  
        **if** (getApelido() == **null** || getApelido().isEmpty()) {  
        **return** getNome() + **" "** + getSobrenome();  
        } **else** {  
        **return** getNome() + **" "** + getSobrenome() + **" "** + **"("** + getApelido() + **")"**;  
        }  
         
        }  
       }
    3. **package** br.unaerp.compras.br.unaerp.compras.model;  
         
       **import** java.io.Serializable;  
         
       */\*\*  
        \* Created by raul on 15/6/2016.  
        \*/***public class** FornecedorModel **implements** Serializable{  
         
        **private** Long **id**;  
        **private** String **nomeFantasia**;  
        **private** String **cnpj**;  
        **private** String **endereco**;  
        **private** String **numero**;  
        **private** String **complemento**;  
        **private** String **bairro**;  
        **private** String **estado**;  
        **private** String **cidade**;  
        **private** String **cep**;  
        **private** String **telefone**;  
        **private** String **site**;  
        **private** String **email**;  
         
        **public** FornecedorModel() {  
        }  
         
        **public** Long getId() {  
        **return id**;  
        }  
         
        **public void** setId(Long id) {  
        **this**.**id** = id;  
        }  
         
        **public** String getNomeFantasia() {  
        **return nomeFantasia**;  
        }  
         
        **public void** setNomeFantasia(String nomeFantasia) {  
        **this**.**nomeFantasia** = nomeFantasia;  
        }  
         
        **public** String getCnpj() {  
        **return cnpj**;  
        }  
         
        **public void** setCnpj(String cnpj) {  
        **this**.**cnpj** = cnpj;  
        }  
         
        **public** String getNumero() {  
        **return numero**;  
        }  
         
        **public void** setNumero(String numero) {  
        **this**.**numero** = numero;  
        }  
         
        **public** String getBairro() {  
        **return bairro**;  
        }  
         
        **public void** setBairro(String bairro) {  
        bairro = bairro;  
        }  
         
        **public** String getCidade() {  
        **return cidade**;  
        }  
         
        **public void** setCidade(String cidade) {  
        **this**.**cidade** = cidade;  
        }  
         
        **public** String getEstado() {  
        **return estado**;  
        }  
         
        **public void** setEstado(String estado) {  
        **this**.**estado** = estado;  
        }  
         
        **public** String getCep() {  
        **return cep**;  
        }  
         
        **public void** setCep(String cep) {  
        **this**.**cep** = cep;  
        }  
         
        **public** String getTelefone() {  
        **return telefone**;  
        }  
         
        **public void** setTelefone(String telefone) {  
        **this**.**telefone** = telefone;  
        }  
         
        **public** String getSite() {  
        **return site**;  
        }  
         
        **public void** setSite(String site) {  
        **this**.**site** = site;  
        }  
         
        **public** String getEmail() {  
        **return email**;  
        }  
         
        **public void** setEmail(String email) {  
        **this**.**email** = email;  
        }  
         
        **public** String getEndereco() {  
        **return endereco**;  
        }  
         
        **public void** setEndereco(String endereco) {  
        **this**.**endereco** = endereco;  
        }  
         
        **public** String getComplemento() {  
        **return complemento**;  
        }  
         
        **public void** setComplemento(String complemento) {  
        **this**.**complemento** = complemento;  
        }  
         
        @Override  
        **public** String toString() {  
        **return** getNomeFantasia() + **" - "** + getCidade();  
        }  
       }
    4. **package** br.unaerp.compras.br.unaerp.compras.model;  
         
       */\*\*  
        \* Created by Raul on 13/04/2016.  
        \*/***public class** LoginModel {  
         
        **private** String **usuario**;  
        **private** String **senha**;  
        **private** String **email**;  
         
        **public** String getEmail() {  
        **return email**;  
        }  
         
        **public void** setEmail(String email) {  
        **this**.**email** = email;  
        }  
         
        **public** String getNomeCompleto() {  
        **return nomeCompleto**;  
        }  
         
        **public void** setNomeCompleto(String nomeCompleto) {  
        **this**.**nomeCompleto** = nomeCompleto;  
        }  
         
        **private** String **nomeCompleto**;  
         
        **public** LoginModel() {  
        }  
         
        **public** String getUsuario() {  
        **return usuario**;  
        }  
         
        **public void** setUsuario(String usuario) {  
        **this**.**usuario** = usuario;  
        }  
         
        **public** String getSenha() {  
        **return senha**;  
        }  
         
        **public void** setSenha(String senha) {  
        **this**.**senha** = senha;  
        }  
       }
    5. **package** br.unaerp.compras.br.unaerp.compras.model;  
         
       **import** java.io.Serializable;  
         
       */\*\*  
        \* Created by raul on 17/6/2016.  
        \*/***public class** ProdutoModel **implements** Serializable{  
         
        **private** Long **id**;  
        **private** String **fornecedor**;  
        **private** String **codigo**;  
        **private** String **marca**;  
        **private** String **descricao**;  
        **private** String **tamanho**;  
         
        **public** ProdutoModel() {  
        }  
         
        **public** Long getId() {  
        **return id**;  
        }  
         
        **public void** setId(Long id) {  
        **this**.**id** = id;  
        }  
         
        **public** String getDescricao() {  
        **return descricao**;  
        }  
         
        **public void** setDescricao(String descricao) {  
        **this**.**descricao** = descricao;  
        }  
         
        **public** String getTamanho() {  
        **return tamanho**;  
        }  
         
        **public void** setTamanho(String tamanho) {  
        **this**.**tamanho** = tamanho;  
        }  
         
        **public** String getMarca() {  
        **return marca**;  
        }  
         
        **public void** setMarca(String marca) {  
        **this**.**marca** = marca;  
        }  
         
        **public** String getCodigo() {  
        **return codigo**;  
        }  
         
        **public void** setCodigo(String codigo) {  
        **this**.**codigo** = codigo;  
        }  
         
        **public** String getFornecedor() {  
        **return fornecedor**;  
        }  
         
        **public void** setFornecedor(String fornecedor) {  
        **this**.**fornecedor** = fornecedor;  
        }  
         
        @Override  
        **public** String toString() {  
        **return** getDescricao() + **" - "**+getTamanho();  
        }  
       }

## Tasks

* + 1. **package** br.unaerp.compras.br.unaerp.compras.tasks;  
         
       **import** android.app.ProgressDialog;  
       **import** android.content.Context;  
       **import** android.os.AsyncTask;  
       **import** android.support.v4.app.FragmentActivity;  
       **import** android.util.Log;  
       **import** android.widget.Toast;  
         
       **import** java.util.List;  
         
       **import** br.unaerp.compras.br.unaerp.compras.adapter.ClienteAdapter;  
       **import** br.unaerp.compras.br.unaerp.compras.atividades.MainActivity;  
       **import** br.unaerp.compras.br.unaerp.compras.connection.WebClient;  
       **import** br.unaerp.compras.br.unaerp.compras.dao.ClienteDAO;  
       **import** br.unaerp.compras.br.unaerp.compras.model.BackupModel;  
       **import** br.unaerp.compras.br.unaerp.compras.model.ClienteModel;  
       **import** br.unaerp.compras.br.unaerp.compras.model.FornecedorModel;  
       **import** br.unaerp.compras.br.unaerp.compras.model.ProdutoModel;  
         
       */\*\*  
        \* Created by raul on 7/6/2016.  
        \*/***public class** BackupTask **extends** AsyncTask<Object,Object,String> {  
         
        **private** BackupModel backup;  
        **private** Context context;  
        **private** ProgressDialog dialog;  
         
        **public** BackupTask(Context context) {  
        **this**.context = context;  
        }  
         
        @Override  
        **protected** String doInBackground(Object[] params) {  
        **return** enviaCliente();  
        }  
         
        @Override  
        **protected void** onPostExecute(String resposta) {  
        dialog.dismiss();

**if**(resposta == **null**){  
 resposta = **"Erro"**;  
}

**if**(resposta.equals(**"Sucesso"**)){  
 Toast.makeText(context, **"Sincronização concluída"**, Toast.LENGTH\_SHORT).show();  
 }**else**{  
 Toast.makeText(context, **"Erro na sincronização"**, Toast.LENGTH\_SHORT).show();  
 }  
  
  
 }  
  
 **private** String enviaCliente(){  
 ClienteDAO clientedao = **new** ClienteDAO(context, MainActivity.versaoBD);  
 List<ClienteModel> clientes = clientedao.selectCliente();  
 clientedao.close();  
  
 ClienteAdapter clienteAdapter = **new** ClienteAdapter();  
 String json = clienteAdapter.converteJson(clientes);  
  
 WebClient cliente = **new** WebClient();  
 String resposta = cliente.post(json);  
 *//Log.d("CLIENTES JSON:",json);* **return** resposta;  
 }  
  
 @Override  
 **protected void** onPreExecute() {  
 dialog = ProgressDialog.show(context,**"Aguarde.."**,**"Enviando clientes..."**,**true**,**true**);  
 }  
}

## Util

* + 1. **package** br.unaerp.compras.br.unaerp.compras.utils;  
         
       */\*\*  
        \* Created by raul on 12/6/2016.  
        \*/***public class** Encript {  
         
        **public static** String encripta(String valor) {  
        String resultado =**""**;  
        **int** ascii;  
        **char** letra;  
        **for** (**int** i = 0; i < valor.length(); i++) {  
        letra = valor.charAt(i);  
        ascii = (**int**) letra;  
        resultado += String.*valueOf*(ascii);  
        }  
        **return** resultado;  
        }  
       }
    2. **package** br.unaerp.compras.br.unaerp.compras.utils;  
         
       **import** android.text.Editable;  
       **import** android.text.TextWatcher;  
       **import** android.widget.EditText;  
         
       **public abstract class** Mask {  
        **public static** String *CNPJ\_MASK* = **"##.###.###/####-##"**;  
        **public static** String *CPF\_MASK* = **"###.###.###-##"**;  
        **public static** String *TELEFONE\_MASK* = **"(##) ####-####"**;  
        **public static** String *CELULAR\_MASK* = **"(##) #### #####"**;  
        **public static** String *CEP\_MASK* = **"#####-###"**;  
         
        **public static** String unmask(String s) {  
        **return** s.replaceAll(**"[.]"**, **""**).replaceAll(**"[-]"**, **""**)  
        .replaceAll(**"[/]"**, **""**).replaceAll(**"[(]"**, **""**)  
        .replaceAll(**"[)]"**, **""**).replaceAll(**" "**, **""**)  
        .replaceAll(**","**, **""**);  
        }  
         
        **public static boolean** isASign(**char** c) {  
        **if** (c == **'.'** || c == **'-'** || c == **'/'** || c == **'('** || c == **')'** || c == **','** || c == **' '**) {  
        **return true**;  
        } **else** {  
        **return false**;  
        }  
        }  
         
        **public static** String mask(String mask, String text) {  
        **int** i = 0;  
        String mascara = **""**;  
        **for** (**char** m : mask.toCharArray()) {  
        **if** (m != **'#'**) {  
        mascara += m;  
        **continue**;  
        }  
        **try** {  
        mascara += text.charAt(i);  
        } **catch** (Exception e) {  
        **break**;  
        }  
        i++;  
        }  
         
        **return** mascara;  
        }  
         
        **public static** TextWatcher insert(**final** String mask, **final** EditText ediTxt) {  
        **return new** TextWatcher() {  
        **boolean** isUpdating;  
        String old = **""**;  
         
        **public void** onTextChanged(CharSequence s, **int** start, **int** before, **int** count) {  
        String str = Mask.unmask(s.toString());  
        String mascara = **""**;  
        **if** (isUpdating) {  
        old = str;  
        isUpdating = **false**;  
        **return**;  
        }  
         
        **int** index = 0;  
        **for** (**int** i = 0; i < mask.length(); i++) {  
        **char** m = mask.charAt(i);  
        **if** (m != **'#'**) {  
        **if** (index == str.length() && str.length() < old.length()) {  
        **continue**;  
        }  
        mascara += m;  
        **continue**;  
        }  
         
        **try** {  
        mascara += str.charAt(index);  
        } **catch** (Exception e) {  
        **break**;  
        }  
         
        index++;  
        }  
         
        **if** (mascara.length() > 0) {  
        **char** last\_char = mascara.charAt(mascara.length() - 1);  
        **boolean** hadSign = **false**;  
        **while** (isASign(last\_char) && str.length() == old.length()) {  
        mascara = mascara.substring(0, mascara.length() - 1);  
        last\_char = mascara.charAt(mascara.length() - 1);  
        hadSign = **true**;  
        }  
         
        **if** (mascara.length() > 0 && hadSign) {  
        mascara = mascara.substring(0, mascara.length() - 1);  
        }  
        }  
         
        isUpdating = **true**;  
        ediTxt.setText(mascara);  
        ediTxt.setSelection(mascara.length());  
        }  
         
        **public void** beforeTextChanged(CharSequence s, **int** start, **int** count, **int** after) {  
        }  
         
        **public void** afterTextChanged(Editable s) {  
        }  
        };  
        }  
       }