



UNIVERSIDADE REGIONAL DE BLUMENAU
CURSO DE SISTEMAS DE INFORMAÇÃO - BACHARELADO

PROTÓTIPO DE FORÇA DE VENDAS PARA DISPOSITIVOS MÓVEIS BASEADOS NA PLATAFORMA ANDROID

FÁBIO DE OLIVEIRA

PROF.FRANCISCO ADELL PÉRICAS,ORIENTADOR

ROTEIRO

Introdução e Objetivos

Fundamentação teórica

Sistema atual

Desenvolvimento e Especificação do sistema

Técnicas e ferramentas utilizadas

Operacionalidade

Resultados e Discussão

Conclusão

Extensão

Demonstração do sistema



INTRODUÇÃO

Informação na área de vendas

Android

Dispositivos móveis em vendas

Empresa Sold Out

OBJETIVOS

O objetivo geral do trabalho é facilitar as operações de vendas, controle de estoque e demonstração de produtos da marca Sold Out

Objetivos específicos:

- desenvolver um aplicativo na plataforma Android que possibilite a realização de cadastros, vendas e controle de estoque de produtos
- disponibilizar um *web service* para integração de informações entre o dispositivo móvel e o servidor



FORÇA DE VENDAS

Conceito

Automação de força de vendas

Sistemas de força de vendas



DISPOSITIVO MÓVEL - ANDROID

Características dos dispositivos móveis

Smartphone

Conceito de Android

Android no mercado

WEB SERVICES

Conceito

Utilização de *web service*

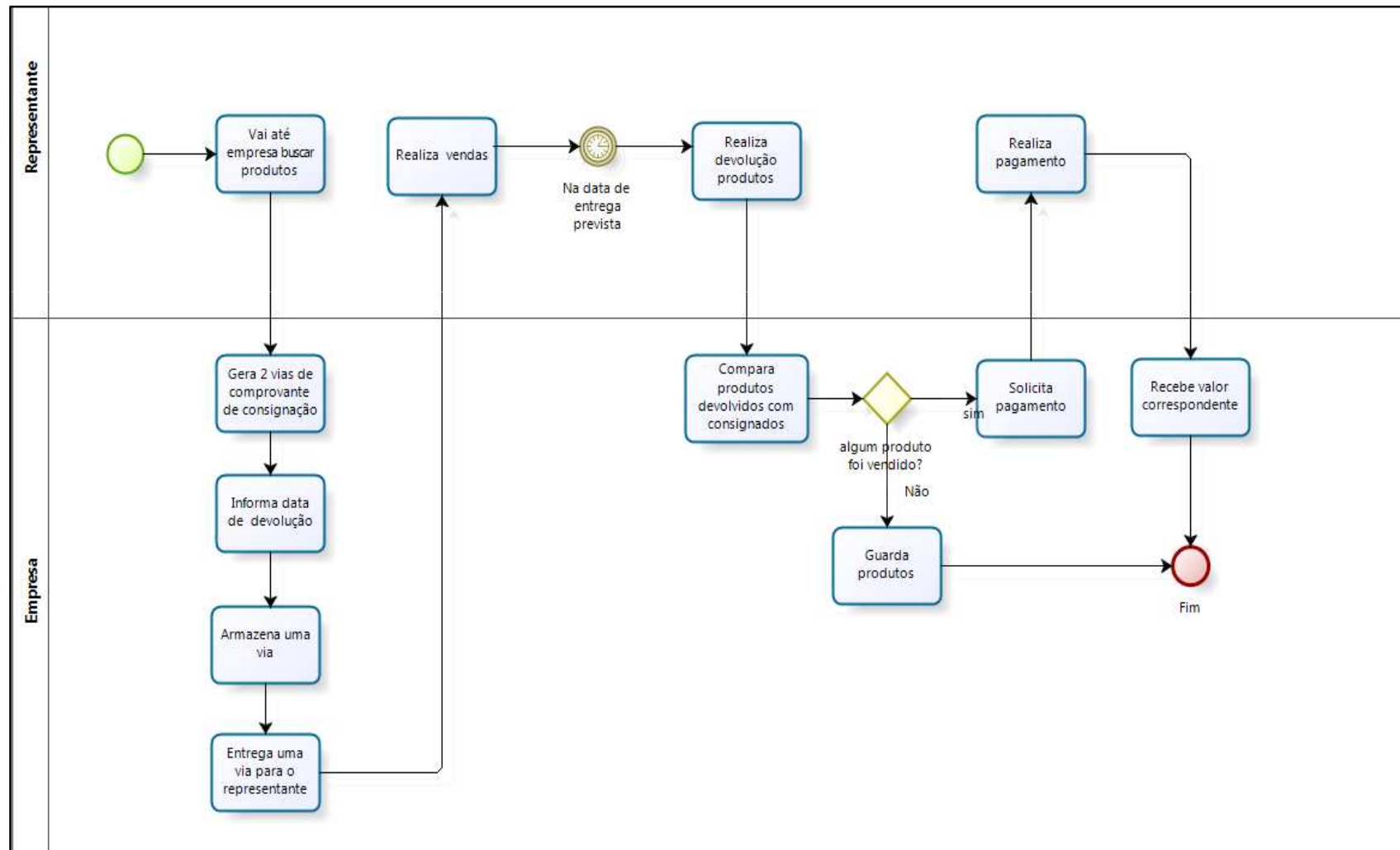
Funcionamento:



SISTEMA ATUAL



Processo de consignação



TRABALHOS CORRELATOS

TCC de Tiago Machado - baseado em PALM OS:

- Aplicação para controle de vendas genérico
- Desenvolvido em *Visual Basic*

A screenshot of a Palm OS application interface titled 'Cadastro Clientes Ativo'. The interface is designed for a handheld device and features a list of client information fields. The fields are: 'Nome: PAULO CESAR LIMA', 'End: R. 7 SETEMBRO 573', 'Rota: 001 - SEGUNDA' (with a dropdown arrow), 'Cidade: BLUMENAU', 'Bairro: CENTRO', 'Telefone: 04733263366', 'Contato: PAULO', and 'CPF/CNPJ: 04783375823'. Below the fields are three buttons: 'Gravar', '+ Inf.', and 'Voltar'. At the bottom of the screen is a standard Palm OS navigation bar with icons for home, back, search, and a keyboard icon.

TRABALHOS CORRELATOS

AppVenda\$, desenvolvido pela empresa Comunicar TI:

- Aplicativo para Android
- Simples e intuitivo com o objetivo de controlar vendas diretas

The screenshot shows the 'Nova Venda' (New Sale) screen of the AppVenda\$ application. At the top, there is a blue header bar with 'Voltar' (Back) on the left, 'Nova Venda' in the center, and 'Menu' on the right. Below the header, the text 'Cliente: Cliente XYZ' is displayed. A section titled 'Lista de Produtos' (Product List) contains a list of items with their quantities and prices. At the bottom of the screen, there is a white box containing the total amount 'R\$2.222,61' and an 'Avançar' (Next) button.

Quantidade	Produto	Preço
4	Batom	R\$42,32
10	Creme	R\$149,90
2	Perfume	R\$30,40
1	Notebook	R\$1.999,99

LEVANTAMENTO INFORMAÇÕES



Problemas no processo atual:

- risco com representantes
- avarias nos produtos
- venda incerta
- falta de controle de entrada e saída
- falta de controle de estoque
- preços indefinidos

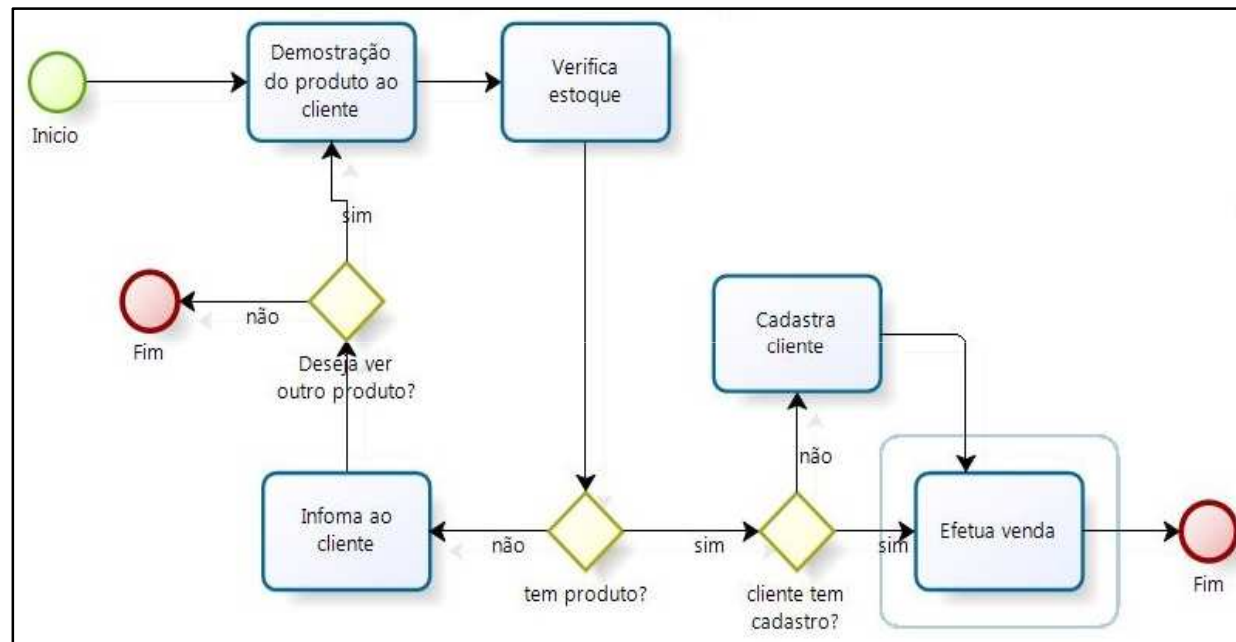
Proposta de solução:



LEVANTAMENTO INFORMAÇÕES



Novo processo de vendas





FERRAMENTAS UTILIZADAS

Eclipse para implementação do código fonte

PostgreSQL para armazenamento dos dados

GlassFish 4 como servidor de aplicação

Enterprise Architect para modelagem de diagramas

REQUISITOS FUNCIONAIS



Requisitos Funcionais	Caso de Uso
RF01: O sistema deverá permitir ao usuário cadastrar produtos.	UC01
RF02: O sistema deverá permitir ao usuário cadastrar clientes.	UC02
RF03: O sistema deverá permitir ao usuário controlar estoque.	UC03
RF04: O sistema deverá possibilitar ao usuário realizar a venda de produtos.	UC05
RF05: O sistema deverá permitir a integração de dados entre aplicativo e servidor.	UC06
RF06: O sistema deverá permitir ao representante realizar a demonstração de produtos.	UC04
RF07: O sistema deverá permitir ao representante consultar o estoque.	UC07

REQUISITOS NÃO FUNCIONAIS



Requisitos Não Funcionais
RNF01: O sistema deverá utilizar ambiente de desenvolvimento Eclipse com <i>plugin</i> ADT.
RNF02: O sistema deverá ser compatível com a plataforma Android.
RNF03: O sistema deverá utilizar banco de dados PostgreSQL.
RNF04: O sistema deverá utilizar como servidor de aplicação Glassfish 4.
RNF05: O sistema necessita de conexão com internet para seu funcionamento.

DIAGRAMA DE CASOS DE USO

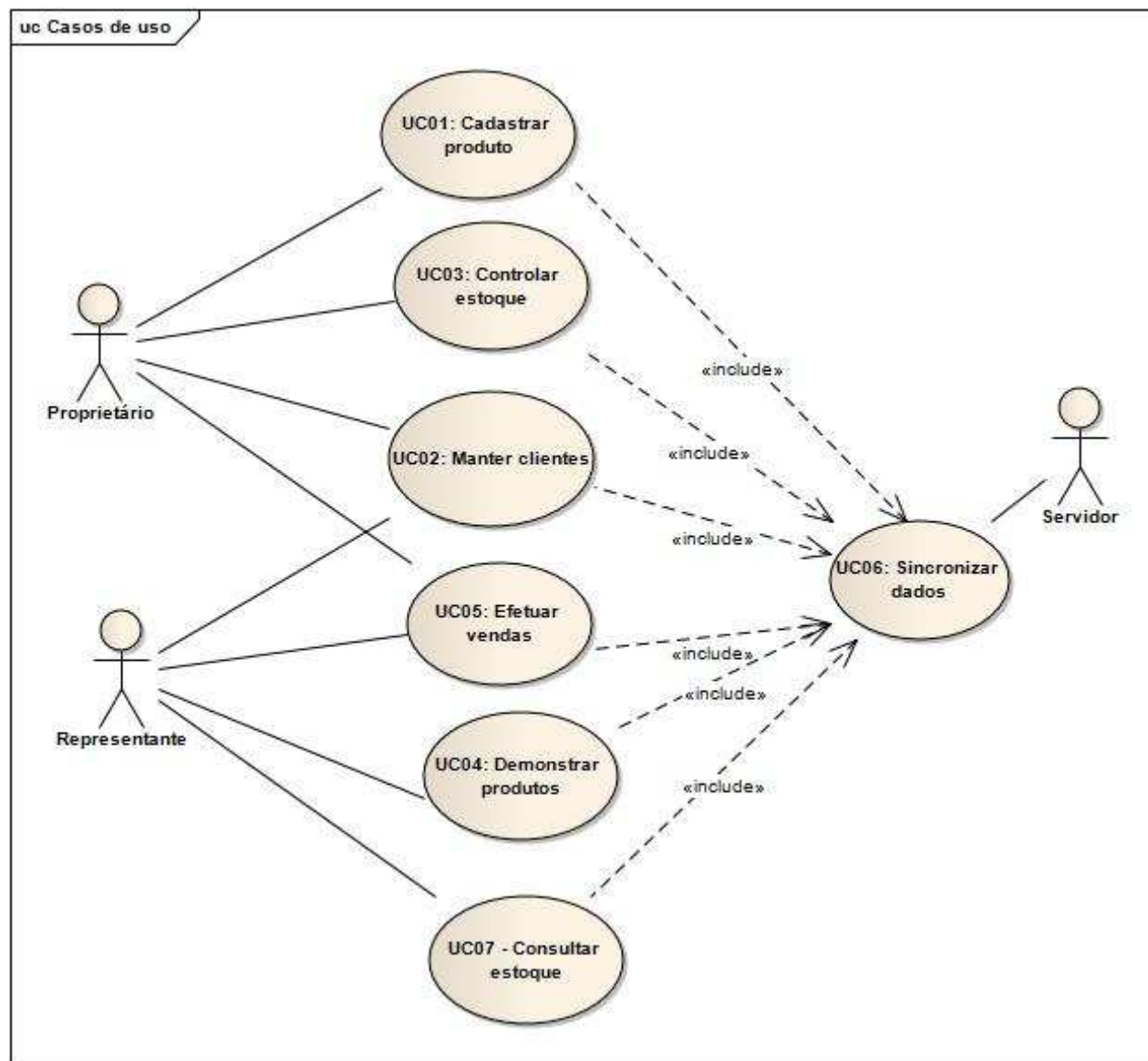
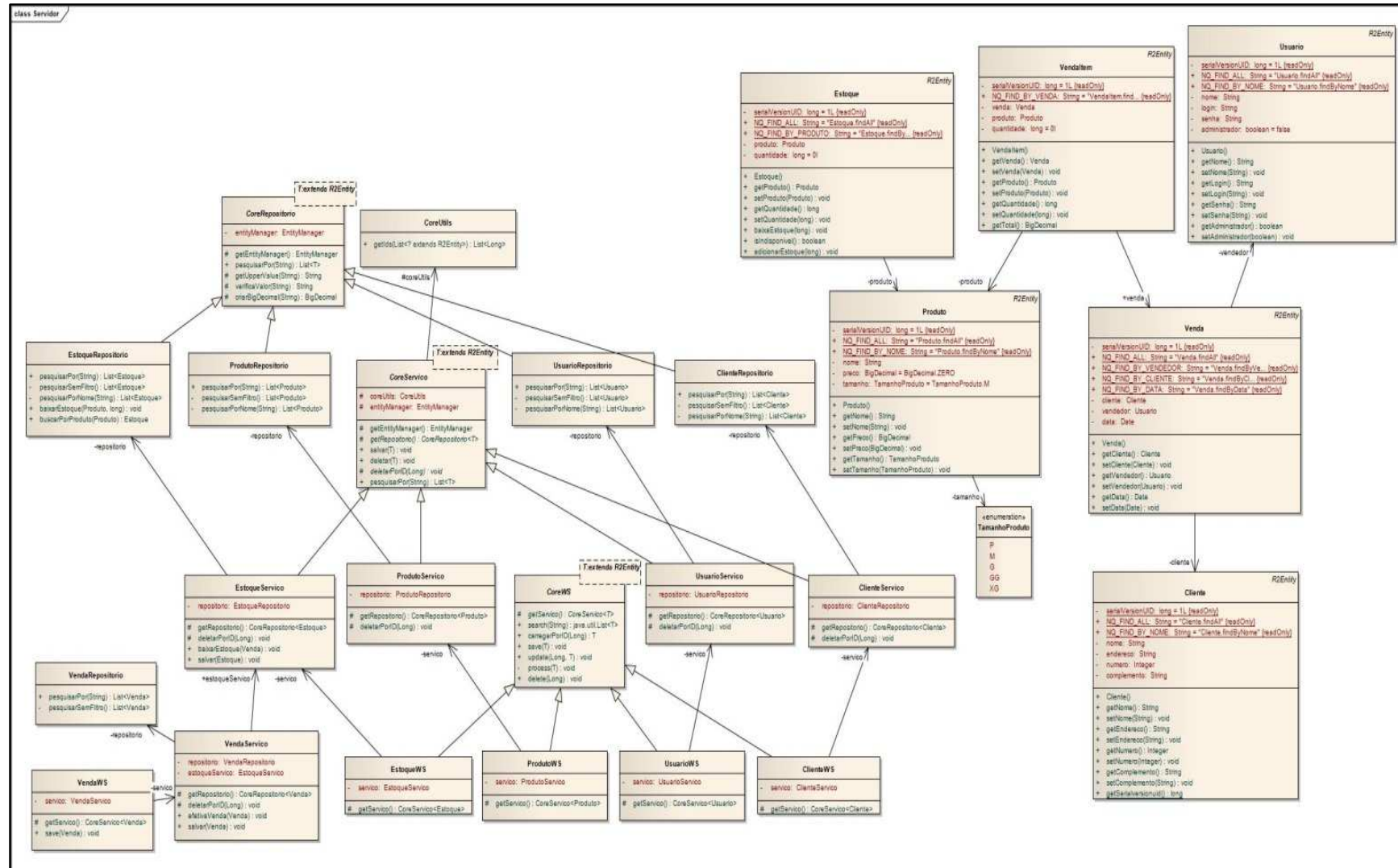


DIAGRAMA DE CLASSES - SERVIDOR



TÉCNICAS E FERRAMENTAS UTILIZADAS

Desenvolvimento Servidor:

- JPA - *Java Persistence API*
- Beans Validation
- REST
- CoreWS – web services

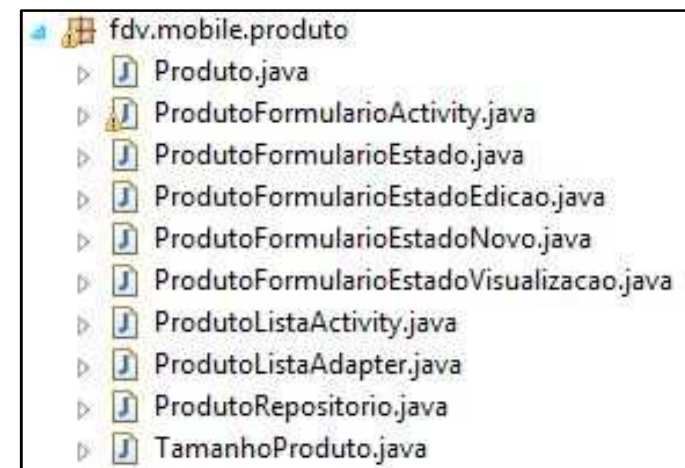
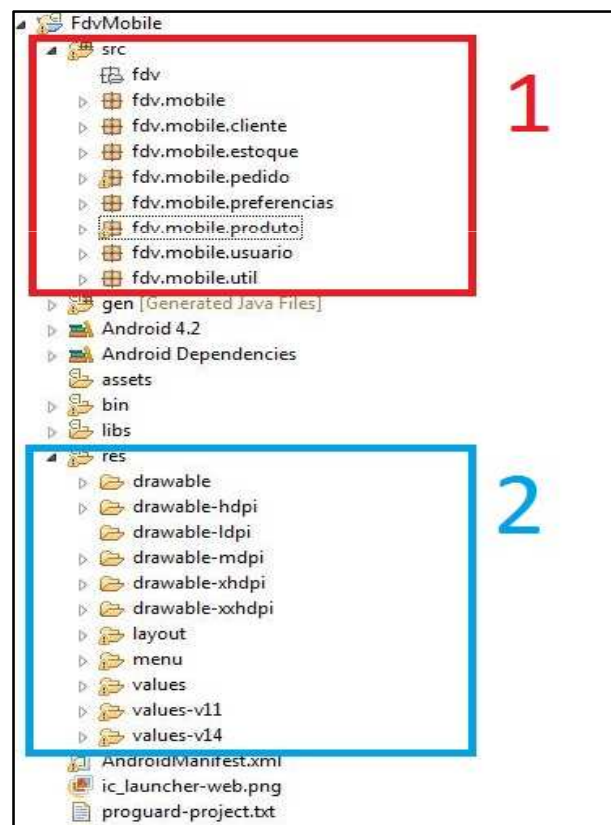
Verbos HTTP:

GET	<ul style="list-style-type: none">• Buscar recursos• Cache
POST	<ul style="list-style-type: none">• Criar um novo recurso
PUT	<ul style="list-style-type: none">• Atualizar um recurso existente
DELETE	<ul style="list-style-type: none">• Remover um recurso

TÉCNICAS E FERRAMENTAS UTILIZADAS

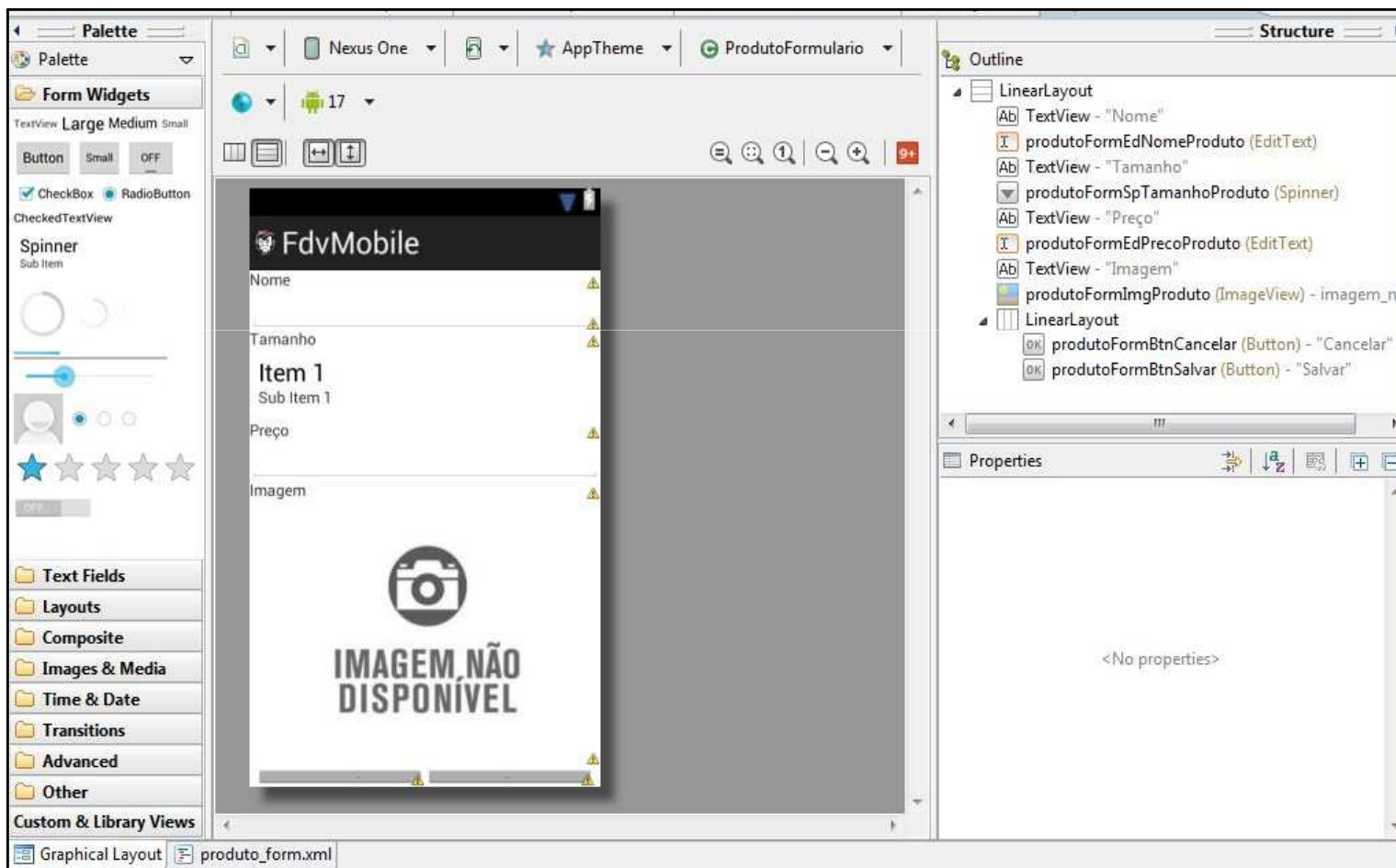
Desenvolvimento Aplicativo:

Estrutura projeto/pacotes



TÉCNICAS E FERRAMENTAS UTILIZADAS

Layout – tela/XML



TÉCNICAS E FERRAMENTAS UTILIZADAS



Chamada web service

```
public void salvar(Produto produto) {
    try {
        if (produto.getId() == null) {
            Rest.post("http://192.168.0.2:8080/teste/api-fdv/produto",
                produto.toJSON());
        } else {
            Rest.put("http://192.168.0.2:8080/teste/api-fdv/produto/" + produto.getId()
                + produto.toJSON());
        }

        produtos.remove(produto);
        produtos.add(produto);
    } catch (Throwable e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```

```
public static String post(String url, String json) throws Throwable {
    HttpParams httpParameters = new BasicHttpParams();

    int timeoutConnection = 6000;
    HttpConnectionParams.setConnectionTimeout(httpParameters,
        timeoutConnection);

    int timeoutSocket = 6500;
    HttpConnectionParams.setSoTimeout(httpParameters, timeoutSocket);
    DefaultHttpClient httpClient = new DefaultHttpClient(httpParameters);
    HttpContext localContext = new BasicHttpContext();
    HttpPost postRequest = new HttpPost(url);

    StringEntity input = new StringEntity(json);
    input.setContentType("application/json");
    postRequest.setEntity(input);

    HttpResponse response = httpClient.execute(postRequest, localContext);

    if (response.getStatusLine().getStatusCode() != 201) {
        throw new RuntimeException("Falha : HTTP cod : "
            + response.getStatusLine().getStatusCode());
    }

    BufferedReader br = new BufferedReader(new InputStreamReader(
        (response.getEntity().getContent())));

    String output;
    StringBuilder resultado = new StringBuilder();
    while ((output = br.readLine()) != null) {
        resultado.append(output);
    }

    httpClient.getConnectionManager().shutdown();
    return resultado.toString();
}
```

OPERACIONALIDADE



3G 23h12

FdvMobile

Nome
Produto04

Tamanho
M

Preço
20.0

Imagem


IMAGEM NÃO DISPONÍVEL

Cancelar Salvar



3G 23h48

Estoque

Produto
Selecione um produto

Q

- Produtos
- Produto01
- Produto02
- Produto03
- Produto04



RESULTADOS E DISCUSSÕES

Este trabalho versus trabalhos correlatos

Atingimento dos objetivos

CONCLUSÕES

A grande vantagem é que o aplicativo permite que seja alterado e aperfeiçoado o processo de vendas

O representante passará utilizar o aplicativo para realizar todo o processo de vendas

O estoque estará sempre atualizado

Escolha da tecnologia: Java, Android



EXTENSÕES

Implementar integração do cadastro de usuário com o sistema de Recursos Humanos (RH), para que o *login* dos representantes seja único para a empresa toda

Expandir sistema para ser usado em outros departamentos

Criar módulo de relatórios e gráficos

Disponibilizar o aplicativo para outros sistemas operacionais, como IOS e Windows Phone

DEMONSTRAÇÃO DO SISTEMA!

