Universidade Federal de Sergipe
Centro de Ciências Exatas e Tecnologia
DEPARTAMENTO DE COMPUTAÇÃO - DCOMP
CIÊNCIA DA COMPUTAÇÃO

Documentos Desenvolvimento de Software

Prof. NOME DO PROFESSOR

São Cristóvão, Sergipe
MÊS – ANO

Componentes:

NOME DO ALUNO A – MATRICULA
NOME DO ALUNO B – MATRICULA
NOME DO ALUNO C – MATRICULA
NOME DO ALUNO D – MATRICULA
NOME DO ALUNO E – MATRICULA

Conteúdo

1 Levantamento de Requisitos **4**
- 1.1 Propósito do Documento . . . 4
- 1.2 Escopo do Produto . . . 4
- 1.3 Definições e Abreviações . . . 4
  - 1.3.1 Definições . . . 4
  - 1.3.2 Abreviações . . . 4
- 1.4 Referências . . . 4
- 1.5 Visão Geral do Restante do Documento . . . 4
- 1.6 Descrição Geral . . . 4
  - 1.6.1 Perspectiva do Produto . . . 4
  - 1.6.2 Funções do Produto . . . 4
  - 1.6.3 Características do Usuário . . . 4
  - 1.6.4 Restrições Gerais . . . 5
  - 1.6.5 Suposições e Dependências . . . 5
- 1.7 Requisitos específicos . . . 5
  - 1.7.1 Requisitos Funcionais . . . 5
  - 1.7.2 Requisitos Não Funcionais . . . 5

2 Plano de Projeto **6**
- 2.1 Motivação . . . 6

3 Casos de Uso **7**

4 Codificação Java/Android **8**
- 4.1 Modelos . . . 8
  - 4.1.1 Modelo Base . . . 8
  - 4.1.2 Endereço . . . 8
- 4.2 Controles . . . 9
  - 4.2.1 Controle de Atendimento . . . 9
  - 4.2.2 Controle de Compra . . . 9
- 4.3 Fronteiras . . . 9
  - 4.3.1 Menu Principal . . . 9
  - 4.3.2 Menu Fornecedores . . . 10

5 Diagramas **11**
- 5.1 Diagrama de Classes . . . 11
- 5.2 Diagrama de Casos de Uso . . . 12

1 Levantamento de Requisitos

1.1 Propósito do Documento

Escrever Propósito do documento

1.2 Escopo do Produto

Escrever escopo do produto

1.3 Definições e Abreviações

1.3.1 Definições

Estudante: pessoa sem dinheiro.

1.3.2 Abreviações

RF: Requisito Funcional.

RNF: Requisito Não Funcional.

1.4 Referências

SOMMERVILLE, I. Engenharia de Software. Pearson/Prentice Hall.

1.5 Visão Geral do Restante do Documento

Escrever visão geral.

1.6 Descrição Geral

1.6.1 Perspectiva do Produto

...

1.6.2 Funções do Produto

...

1.6.3 Características do Usuário

...

User 1: Faz isso.

User 2: Faz aquilo.

1.6.4 Restrições Gerais

...

1.6.5 Suposições e Dependências

...

1.7 Requisitos específicos

1.7.1 Requisitos Funcionais

RF1 Inclusão de fornecedores. (Pr.: 3)

O sistema deve efetuar o cadastro dos fornecedores.

RF2 Alteração de fornecedores. (Pr.: 2)

O sistema deve efetuar a alteração dos dados cadastrais de fornecedores.

RF3 Exclusão de fornecedores. (Pr.: 1)

O sistema deve efetuar a exclusão de fornecedores.

1.7.2 Requisitos Não Funcionais

RNF1 (Pr.: 1): O sistema deve retornar as consultas em, no máximo, 6 segundos, em 90% dos casos.

RNF2 (Pr.: 1): O sistema deve retornar as consultas em, no máximo, 6 segundos, em 90% dos casos.

2 Plano de Projeto

Descrever Plano de Projeto

2.1 Motivação

Motivação para o projeto:

- Motivação 1
- Motivação 2
- Motivação 3

3 Casos de Uso

Nome: ESCREVER NOME.
Descrição: ESCREVER DESCRIÇÃO.
Identificador: ESCREVER IDENTIFICADOR.
Importância: ESCREVER IMPORTÂNCIA.
Ator Primário: ESCREVER ATOR PRIMÁRIO.
Fluxo Principal:

Sistema	Gerente	Funcionário
	1 - Ação	
2 - Ação		
	3 - Ação	
		4 - Ação
	5 - Ação	
6 - Ação		

Nome: ESCREVER NOME.
Descrição: ESCREVER DESCRIÇÃO.
Identificador: ESCREVER IDENTIFICADOR.
Importância: ESCREVER IMPORTÂNCIA.
Ator Primário: ESCREVER ATOR PRIMÁRIO.
Pré-condições: ESCREVER PRÉ-CONDIÇÕES.
Fluxo Principal:

Sistema	Gerente	Funcionário
	1 - Ação	
2 - Ação		
	3 - Ação	
		4 - Ação
	5 - Ação	
6 - Ação		

4 Codificação Java/Android

4.1 Modelos

4.1.1 Modelo Base

```java
package modelos;

import android.content.Context;

import java.util.ArrayList;

public class Model<T> {
    protected Integer id;
    protected transient Context context;

    public Model(Context context){
        this.context = context;
    }

    public void setContext(Context context){
        this.context = context;
    }

    public void save(){
        save(null);
    }

    public void save(final RequestCallback<Model> callback){
        new ServerRequest(context).sendRequest(getControllerName(), ServerRequest.Action.SAVE, new Gson().toJson(this),
      new RequestCallback<JSONObject>() {
            @Override
            public void execute(JSONObject json) throws Exception {
                setId(json.getInt("id"));
                if (callback != null){
                    callback.execute(Model.this);
                }

                super.execute(json);
            }
        });
    }
}
```

Código 1: Model.java

4.1.2 Endereço

```java
package modelos;

import android.content.Context;

public class Endereco extends Model<Endereco> {
    private String logradouro;
    private String rua;
    private int numero;
    private String bairro;
    private String cidade;
    private String estado;
    private String cep;

    public Endereco(Context context){
        super(context);
    }

    public Endereco(Context context, String logradouro, String rua, int numero, String bairro, String cidade, String
     estado, String cep) {
        super(context);
        this.logradouro = logradouro;
        this.rua = rua;
        this.numero = numero;
        this.bairro = bairro;
        this.cidade = cidade;
        this.cep = cep;
        this.estado = estado;
    }
}
```

Código 2: Endereco.java

4.2 Controles

4.2.1 Controle de Atendimento

```java
package controles;

import android.content.Context;

import java.util.ArrayList;

import modelos.Comanda;
import modelos.Mesa;

public class ControleDeAtendimento {
    private Context context;
    private ControleDeImpressao controleDeImpressao;

    private ArrayList<Mesa> mesas = new ArrayList<Mesa>();

    public ControleDeAtendimento(Context context){
        this.context = context;
        this.controleDeImpressao = new ControleDeImpressao(context);
        int numeroDeMesas = 20;
        for(int i = 1; i <= numeroDeMesas; i++) {
            mesas.add(new Mesa(context, i));
        }
    }
}
```

Código 3: ControleDeAtendimento.java

4.2.2 Controle de Compra

```java
package controles;

import android.content.Context;

import com.google.gson.reflect.TypeToken;

import java.util.ArrayList;

import modelos.Compra;

public class ControleDeCompra {
    private ArrayList<Compra> compras = new ArrayList<>();
    private Context context;
    private static Compra selecionada;

    public ControleDeCompra(Context context){
        this.context = context;
    }

    public static void salvarCompra(Compra compra) {
        compra.save();
    }

    public static void excluirCompra(Compra compra) {
        compra.delete();
    }
}
```

Código 4: ControleDeCompra.java

4.3 Fronteiras

4.3.1 Menu Principal

```java
package fronteiras;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;

import controles.ControleDeFornecedor;
import controles.ControleDeImpressao;
```

```
12 import modelos.Fornecedor;
13
14 public class MenuPrincipal extends AppCompatActivity {
15
16     @Override
17     protected void onCreate(Bundle savedInstanceState) {
18         super.onCreate(savedInstanceState);
19         setContentView(R.layout.activity_menu_principal);
20
21         findViewById(R.id.menu_fornecedores).setOnClickListener(new View.OnClickListener() {
22             @Override
23             public void onClick(View v) {
24                 open(MenuFornecedores.class);
25             }
26         });
27     }
28 }
```

Código 5: MenuPrincipal.java

4.3.2 Menu Fornecedores

```
1  package fronteiras;
2
3  import android.content.Intent;
4  import android.support.v7.app.AppCompatActivity;
5  import android.os.Bundle;
6  import android.view.View;
7
8  public class MenuFornecedores extends AppCompatActivity {
9
10     @Override
11     protected void onCreate(Bundle savedInstanceState) {
12         super.onCreate(savedInstanceState);
13         setContentView(R.layout.activity_menu_fornecedores);
14
15
16         findViewById(R.id.menu_cadastrar_fornecedor).setOnClickListener(new View.OnClickListener() {
17             @Override
18             public void onClick(View v) {
19                 open(CadastroFornecedor.class);
20             }
21         });
22
23         findViewById(R.id.menu_buscar_fornecedor).setOnClickListener(new View.OnClickListener() {
24             @Override
25             public void onClick(View v) {
26                 open(BuscaFornecedor.class);
27             }
28         });
29     }
30
31     public void open(Class activity){
32         startActivity(new Intent(this, activity));
33     }
34 }
```

Código 6: MenuFornecedores.java

5 Diagramas

5.1 Diagrama de Classes

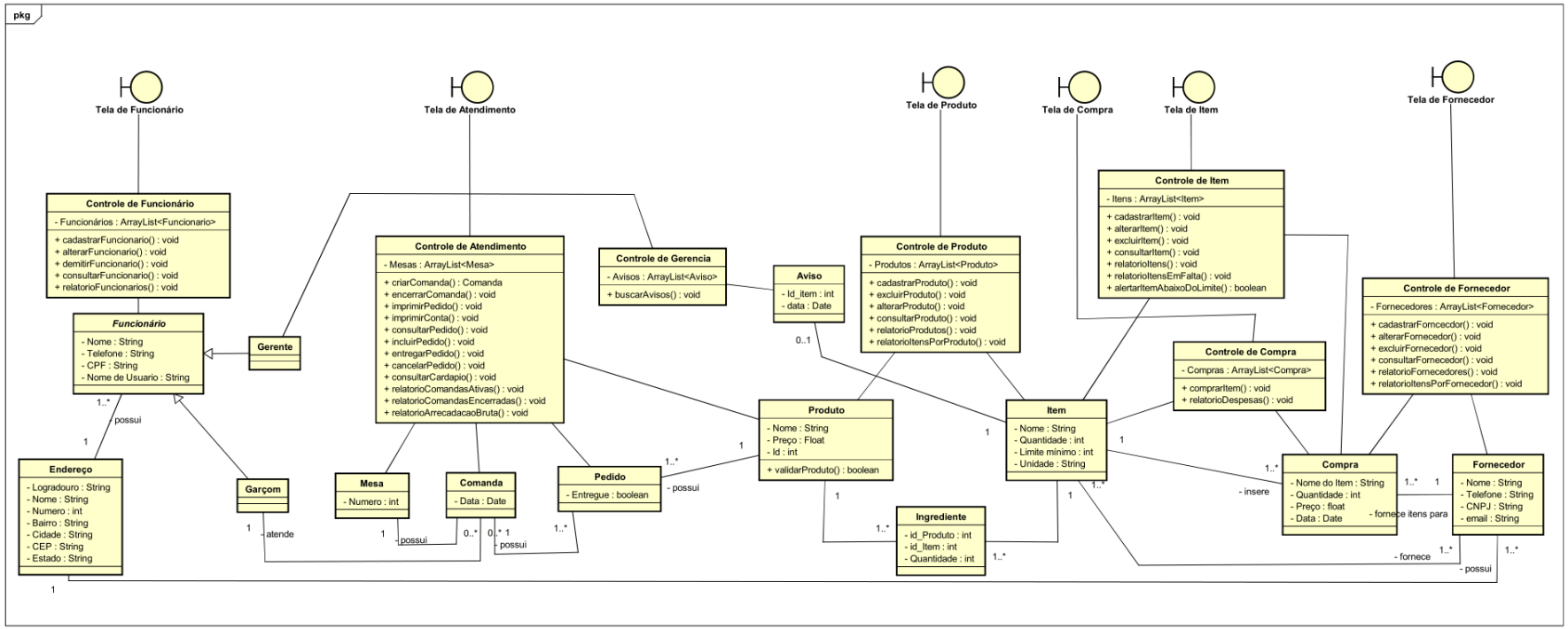

Figura 1: Diagrama de Classes

5.2 Diagrama de Casos de Uso

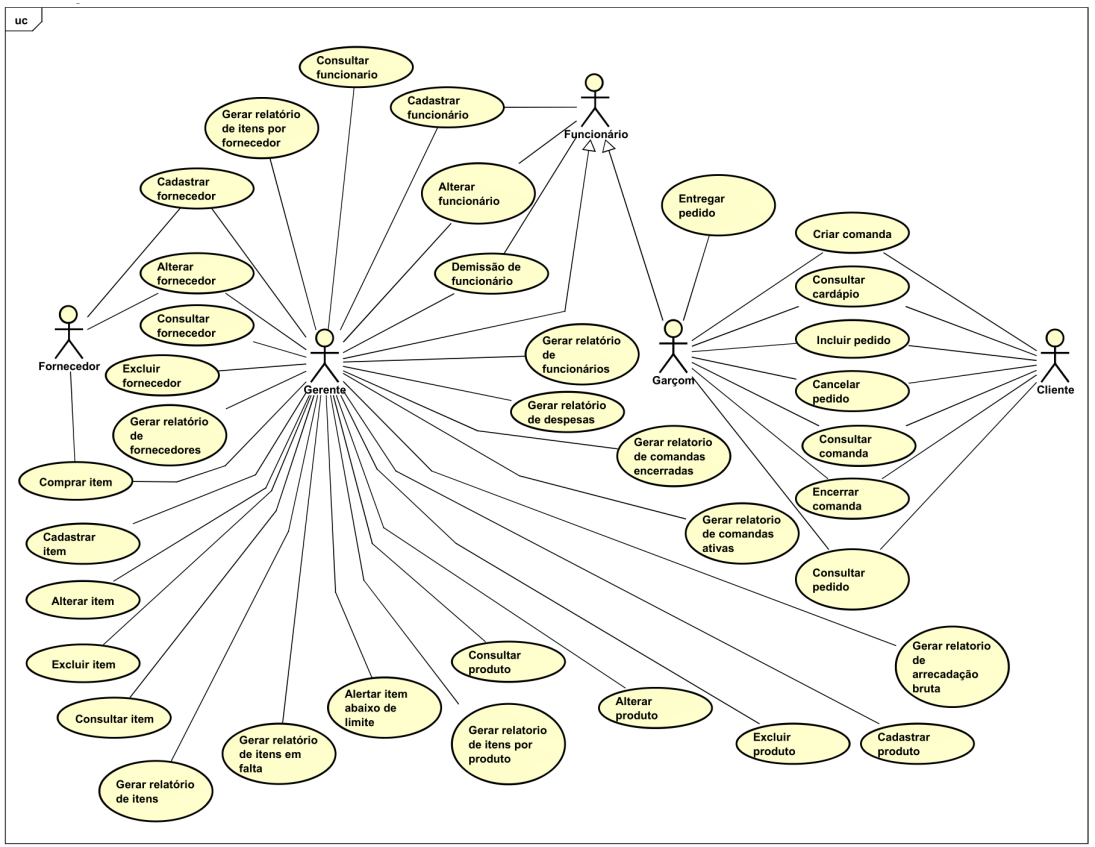

Figura 2: Diagrama de Casos de Uso