

PROJETO SERVIÇO DE ORDEM

V.1.0

Estácio-Teresina, PI

2022

ALUNOS

Marcos Vinicius Carvalho	201908299681
Nathan Castelo Branco	201909106399
Clarice Ellen Leite dos Anjos	201908027169
Pablo Wendell Lino da Silva	201908354411
Ryquelme Cabral dos Santos	201903161657

1.INTRODUÇÃO

Este documento apresenta um sistema de serviço que permite o controle ordens de um estabelecimento, assim como o cadastro e gerenciamento de clientes e funcionários através dele. O serviço solicita que o administrador entre com usuário e senha para que ele possa navegar pelo menu de funcionalidades disponíveis pelo software. Dentro dessas funcionalidades estão inclusas criar, ler, atualizar e deletar dados, classificando assim o sistema dentro do acrônimo CRUD (Create, Read, Update and Delete).

2.IMPLEMENTAÇÃO

O sistema trata-se de um sistema de gerenciamento de ordens e apresenta as seguintes características:

- Gerenciamento de dados utilizando um banco de dados;
- Usuários acessam os dados disponíveis no sistema através de acesso protegido por usuário e senha;
- Notificação da equipe de novas solicitações de serviços;
- Gerar comprovante de conclusão além de realizar consultas e solicitar feedback dos clientes;

O sistema foi implementado em JAVA com banco de dados relacional.

3.ARQUITETURA DE SOFTWARE

Levando em consideração os requisitos para o sistema proposto, foram considerados o incorporamento dos seguintes requisitos e táticas:

Funcionais

- Cadastrar novos clientes.
- Cadastrar funcionários.
- Acesso ao sistema protegidos por usuário e senha.
- Registrar a demanda de serviços e seus tipos (Inserir todos os dados necessários e gerar a ordem de serviço).
- Notificar a equipe de novas solicitações de serviços.
- Consultar o andamento da solicitação.
- Gerar um comprovante de conclusão.
- Solicitar feedback dos clientes.

Não funcionais

- Cadastro dos clientes e funcionários feito com seus dados (nome completo, data de nascimento, endereço, Email, CPF e número de telefone).
- O CRUD é responsável por criar, ler, consultar, atualizar e apagar os dados.

- O software fara o registro das demandas de serviços enviando toda informação para o bando de dados.
- O software vai requisitar os dados registrados no Banco de dados e irá enviar a notificação para os funcionários.
- O cliente fará a solicitação do atendimento, para saber em qual estágio seu produto está (sendo esses estágios: em andamento, finalizando ou pedido cancelado).
- O software enviará um e-mail de pesquisa de satisfação do cliente ao finalizar o serviço.

De Negócios

- O computador do cliente deverá ser rigorosamente diagnosticado a fim de resolver os problemas em sua totalidade.
- Evitar qualquer dano ao equipamento do cliente, caso for comprovado, será descontado do salário
- Garantia de 3 meses para serviços mal realizados.
- O pagamento será feito mediante pix, cartão de crédito ou débito (todas as bandeiras as bandeiras com exceção de credshop) ou em espécie.

Diagrama

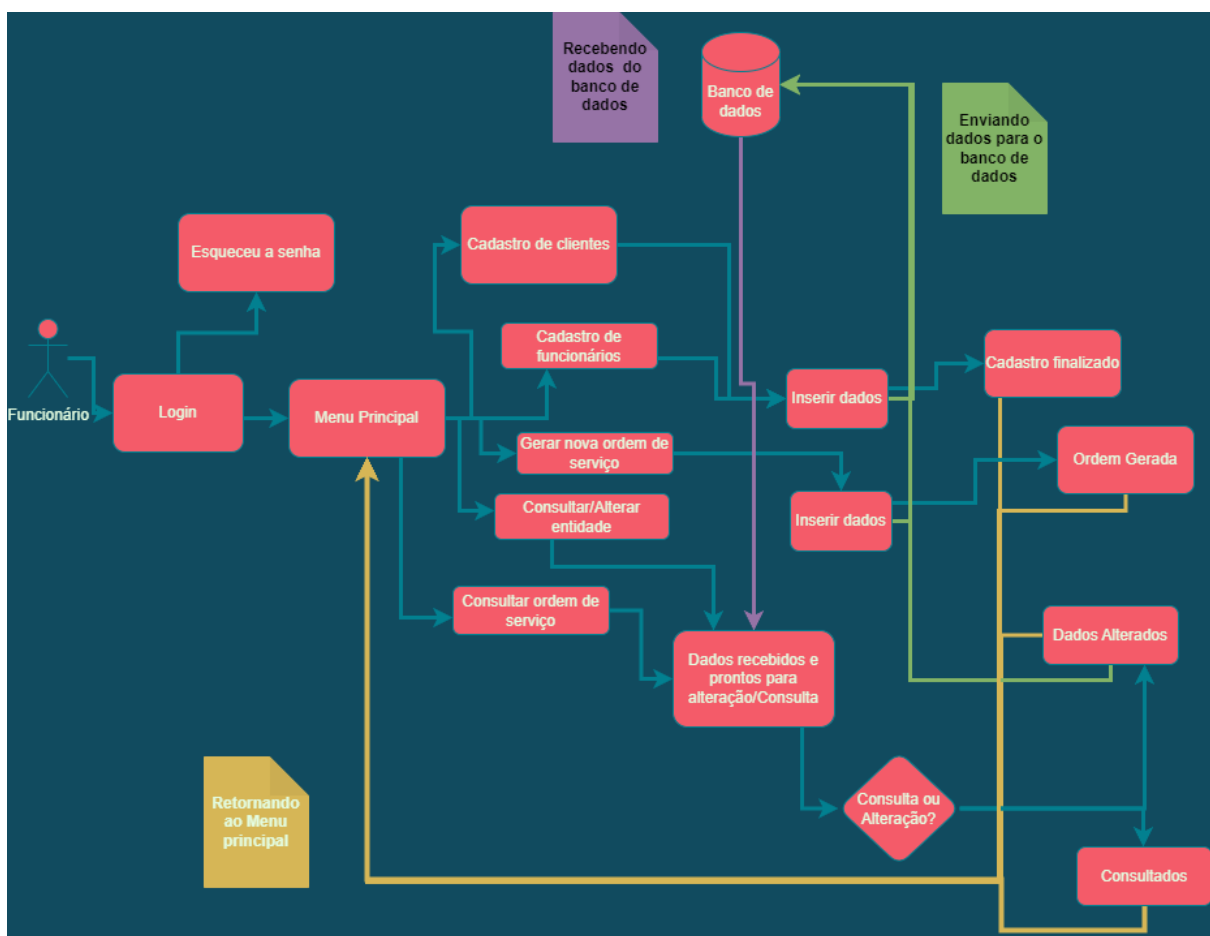


Figura 1 – Diagrama do funcionamento do serviço.

Ordem das telas

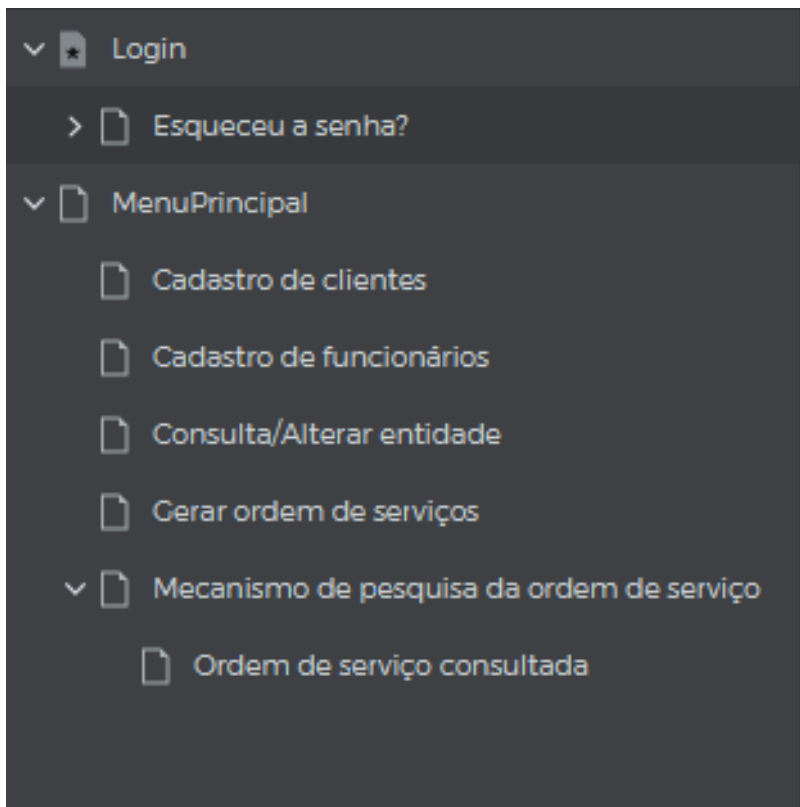
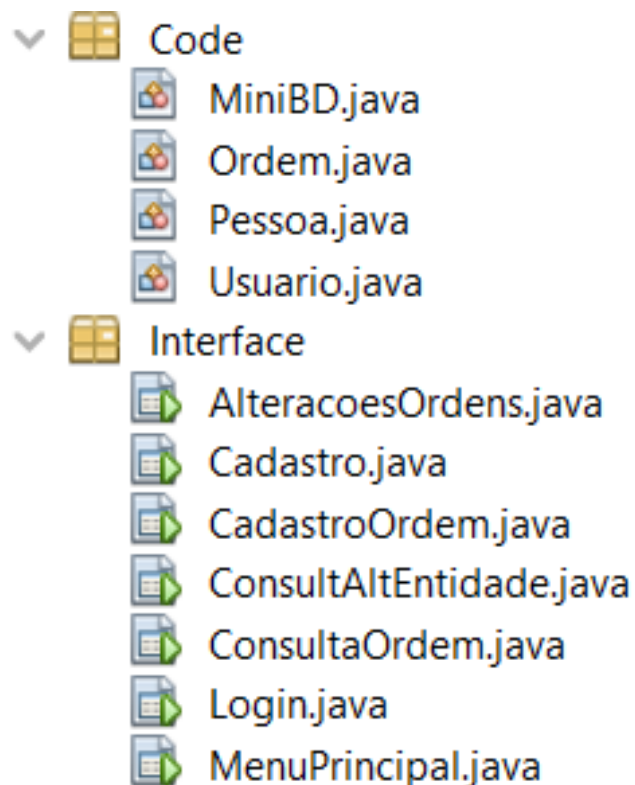


Figura 2 – Esquema de Menu do serviço.

3.1.CLASSES

Inclui onze classes sendo sete delas classes gráficas da interface do programa:



3.1.1.CODE

MiniBD

Esta classe chama o Baco de Dados, permitindo gerenciar os usuários e contendo listas com todas as entidades cadastradas, sendo a Entidade tipo 1 os clientes e a Entidade tipo 2 os Funcionários.

```
public class MiniBD {  
  
    //Uma lista com todas as entidades cadastradas  
    public ArrayList <Pessoa> entidades = new ArrayList<>();  
  
    //Uma lista somente com usuários (Entidade tipo 2)  
    public ArrayList <Usuario> usuarios = new ArrayList<>();  
  
    //Uma lista vazia foi adicionada, então  
    ArrayList<Ordem> ord = new ArrayList<>();  
}
```

Figura 4 – Exemplo de ArrayList contido no Banco de Dados.

Ordem

Esta classe contém variáveis criadas para gerenciar o conteúdo dos serviços realizados bem como detalhes dos clientes e dos pedidos como na figura a seguir:

```
//Detalhes do cliente  
public String nome, email, telefone, cpf;  
  
//Detalhes do pedido  
public String motivo, detalhes, id;  
public String funcionarios; //Aqui estão os funcionários responsáveis pelo serviço  
public String dataIni, dataFim, dataEmit, dataDaOp;  
public double maoDeObra, pecas, orcamentoFinal;  
  
//Serviços realizados / para serem preenchidos no dia da conclusão da operação  
public String servicosR;  
public double orcamentoImp;
```

No registro do serviço é possível inserir dados diversos assim como a pesquisa de dados existentes de clientes cadastrados.

```

this.id = id;
this.nome = nome;
this.email = email;
this.telefone = telefone;
this.cpf = cpf;
this.motivo = motivo;
this.detalhes = detalhes;
this.funcionarios = funcionarios;
this.dataIni = dataIni;
this.dataFim = dataFim;
this.dataEmit = dataEmit;
this.maoDeObra = maoDeObra;
this.pecas = pecas;
this.orcamentoFinal = orcamentoFinal;
this.servicosR = servicosR;
this.dataDaOp = dataDaOp;
this.orcamentoImp = orcamentoImp;

```

Figura 6 – Exemplos de Strings do código.

Pessoa

Contém strings de gerenciamento de entidades sendo o tipo 1 de Entidades o cliente, e o tipo 2 o funcionário. Aqui é possível cadastrar, alterar e excluir dados completos das Entidades existentes.

```

public class Pessoa {

    public int id; //não é necessário no preenchimento
    public String nome;
    public String dataNasc;
    public String cpf;
    public String endereco;
    public String email;
    public String telefone;
    public int tipoEntidade; // 1 cliente 2 funcionário
    public ArrayList<Ordem> ordens;

    public Pessoa(String nome, String dataNasc, String cpf, String email, String endereco, String telefone, int tipoEntidade) {
        this.nome = nome;
        this.dataNasc = dataNasc;
        this.cpf = cpf;
        this.email = email;
        this.endereco = endereco;
        this.telefone = telefone;
        this.tipoEntidade = tipoEntidade;
    }
}

```

Figura 7 – Exemplos de Strings contendo tipos de dados das Entidades.

Usuário

O usuário do Sistema, também chamado de Funcionário, é uma entidade do tipo 2 no software, sendo assim, nessa classe é possível gerir apenas esse tipo de pessoa.

Permitindo assim, juntamente com o cadastro dessa entidade, o acesso ao sistema com uma senha também registrada no sistema.

```
public Usuario( String nome, String senha, String dataNasc, String cpf, String endereco,
    super(nome, dataNasc, cpf, endereco, telefone, tipoEntidade);
    this.senha = senha;
}

public Usuario( String nome, String senha, String cpf, int tipoEntidade) {
    super(nome, cpf, tipoEntidade);
    this.senha = senha;
}

public String getSenha() {
    return senha;
}

public void setSenha(String senha) {
    this.senha = senha;
}
```

3.1.2.INTERFACE

Login

É a primeira tela a ser exibida quando o sistema é aberto, ela solicita que o usuário/funcionário entre com o user e a senha cadastrados no sistema que vai buscar os dados no Banco de Dados relacionado, só assim é possível acessar o Menu Principal, assim evitando que qualquer pessoa acesse as ordens e entidades contidas no sistema.



The image shows a login interface with a dark gray background. At the top, the text "Bem-Vindo" is displayed in white. Below it, there are two input fields: the first is labeled "Usuário" and the second is labeled "Senha". Both fields are empty and have a light gray border. Below the "Senha" field, there is a button labeled "Entrar" with a light gray background and a dark gray border. In the bottom right corner, the text "copyright @2022" is visible in a small, light gray font.

MenuPrincipal



Nessa tela são encontrados os botões disponíveis para acessar as funcionalidades do programa, sendo eles:

- Cadastrar Clientes

The image shows the 'Cadastro de Clientes' (Client Registration) form. The title 'Cadastro de Clientes' is centered at the top in a large, white, sans-serif font. The background is a dark gray. Below the title, there are six input fields, each preceded by a label in white text: 'Nome :', 'Data de nascimento :', 'CPF :', 'Email :', 'Endereço :', and 'Telefone :'. Each input field is a dark gray rectangle with a thin white border. At the bottom of the form, there are two buttons with rounded corners and white text: 'Cancelar' on the left and 'Cadastrar' on the right.

- Cadastrar Funcionários

Cadastro de Funcionários

Nome :

Data de nascimento :

CPF :

Email :

Endereço :

Telefone :

- Cadastrar Ordem de Serviço

Data da Emissão : 11 / 11 / 2022

Ordem de Serviço

Dados do Cliente

*Buscar por CPF : Telefone :

Nome : Email :

Dados do Pedido

*Motivo : *Funcionário :

Detalhes :

*Data de Inicio : *Data de Conclusão(Estimada) :

*Mão de Obra : R\$ *Peças : R\$ *Orçamento Final : R\$

Serviços Realizados

Preencher no dia da operação.

Data da realização da Operação :

Orçamento não planejado :

- Consultar/Alterar Clientes e Funcionários

Consultar ou Alterar Cadastros

Consultar clientes ou funcionários por CPF

CPF :

Dados encontrados

Nome :

CPF : Tipo : ...

Data de Nasc : Telefone :

Email :

Endereço :

- Consultar/Alterar Ordem de Serviço

Consultar Ordem de Serviço

CPF do Cliente :

Pedidos encontrados

Identificador

Dados Gerais do pedido

Cliente :

Motivo :

Data de Inicio :

Orçamento Final :

4.CONCLUSÃO

Neste documento, observou-se que o software foi desenvolvido com a finalidade de controlar ordens de serviços assim como as entidades de determinada localidade. Tendo em vista o que foi apresentado neste documento juntamente a testes realizados em cima do código desenvolvido, concluimos que a implementação do sistema CRUD para o gerenciamento de elementos desta organização foi um sucesso, assim, cumprindo o que lhe foi proposto.