

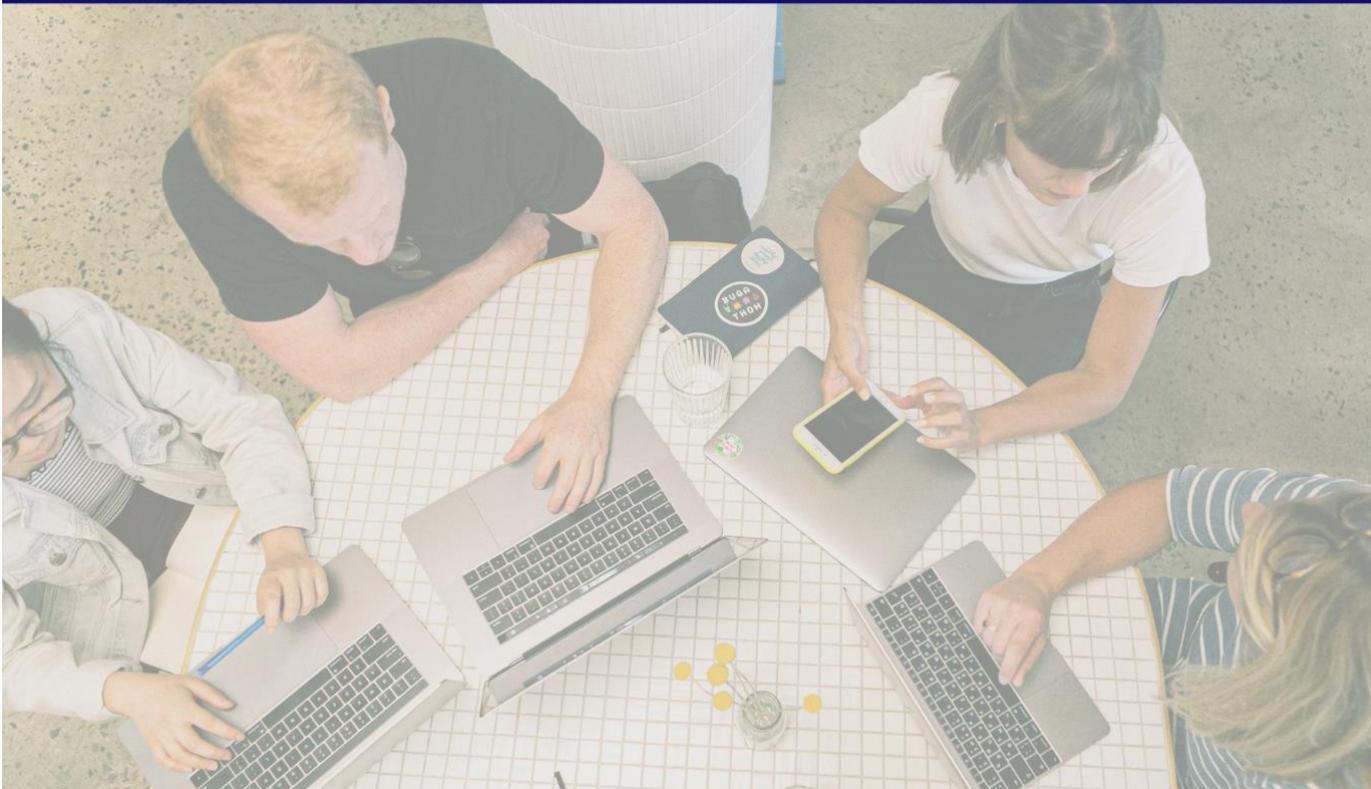


UNifeob

| ESCOLA DE NEGÓCIO

2023

PROJETO DE CONSULTORIA EMPRESARIAL



UNIFEOB
CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO
OCTÁVIO BASTOS
ESCOLA DE NEGÓCIOS
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
GESTÃO DE TECNOLOGIA DA INFORMAÇÃO

PROJETO INTEGRADO
SISTEMA EMPRESARIAL

SÃO JOÃO DA BOA VISTA, SP

ABRIL 2023

UNIFEOB
CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO
OCTÁVIO BASTOS
ESCOLA DE NEGÓCIOS
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
GESTÃO DE TECNOLOGIA DA INFORMAÇÃO

PROJETO INTEGRADO
SISTEMA EMPRESARIAL

MÓDULO DESENVOLVIMENTO DESKTOP

Banco de Dados – Prof. Sidney Gitcoff Telles

Programação Orientada a Objeto – Prof. Sidney Gitcoff Telles

Projeto de Desenvolvimento Desktop – Prof. Sidney Gitcoff Telles

Estudantes:

André Luiz de Melo Filho, RA 1012022101042

João Paulo Godoi de Andrade RA 1012022101264

Oziniel Ewerton Silva, RA 1012022101458

Predro Riquelme Borsone, RA 1012022100206

SÃO JOÃO DA BOA VISTA, SP
ABRIL, 2023

SUMÁRIO

1 INTRODUÇÃO	4
2 DESCRIÇÃO DA EMPRESA	6
3 PROJETO DE CONSULTORIA EMPRESARIAL	7
3.1 BANCO DE DADOS	7
3.1.1	7
3.1.2 TÓPICO 2	7
3.1.3	8
3.2 PROGRAMAÇÃO ORIENTADA A OBJETOS	8
3.2.1	11
3.2.2	13
3.2.3	17
3.3 CONTEÚDO DA FORMAÇÃO PARA A VIDA: ADAPTANDO-SE A MUDANÇAS	9
3.3.1 ADAPTANDO-SE A MUDANÇAS	9
3.3.2 ESTUDANTES NA PRÁTICA	9
4 CONCLUSÃO	11
REFERÊNCIAS	12
ANEXOS	13

1 INTRODUÇÃO

Este projeto tem como objetivo o desenvolvimento de um micro sistema desktop para gerenciamento de uma empresa de pequeno porte, com a possibilidade de cadastramento de serviços, clientes e funcionarios.

O projeto será desenvolvido em Java utilizando o Apache NetBeans IDE 17 para a criação de telas e codificação de backend e MYSQL workbench 8 para o desenvolvimento do banco de dados onde as informações serão armazenadas.

2 DESCRIÇÃO DA EMPRESA

Depois uma pesquisa com pequenos empresários na região de Mogi Guaçu-Sp foi identificado a necessidade de implementação de um sistema para automatização de processos administrativos na empresa "Auto Mecânica SMA" localizada na Rua Rodolpho Innarelli N° 228, Jd. Herminio Bueno - Mogi Guaçu-SP, empresa no ramo de manutenção automobilística há mais de 10 anos atendendo clientes da região.

3 PROJETO DE CONSULTORIA EMPRESARIAL

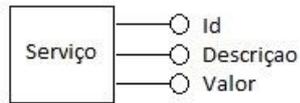
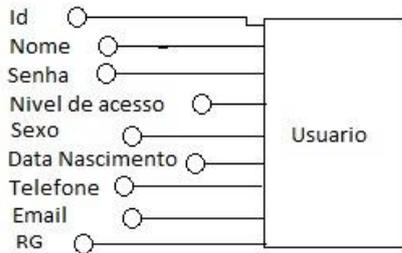
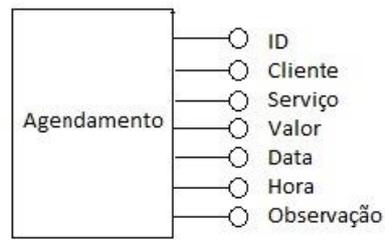
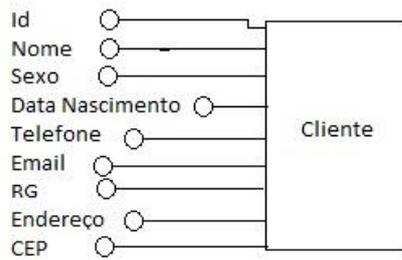
Após uma entrevista com representante da empresa foi identificado a necessidade de implementação de um sistema para uma melhor organização dos dados da empresa como cadastro de clientes, cadastro de funcionários e um agendamento mais automatizado. Para isso será necessário a criação de um micro sistema

3.1 BANCO DE DADOS

Para criação do banco de dados foi identificada a necessidade da criação de quatro tabelas são elas:

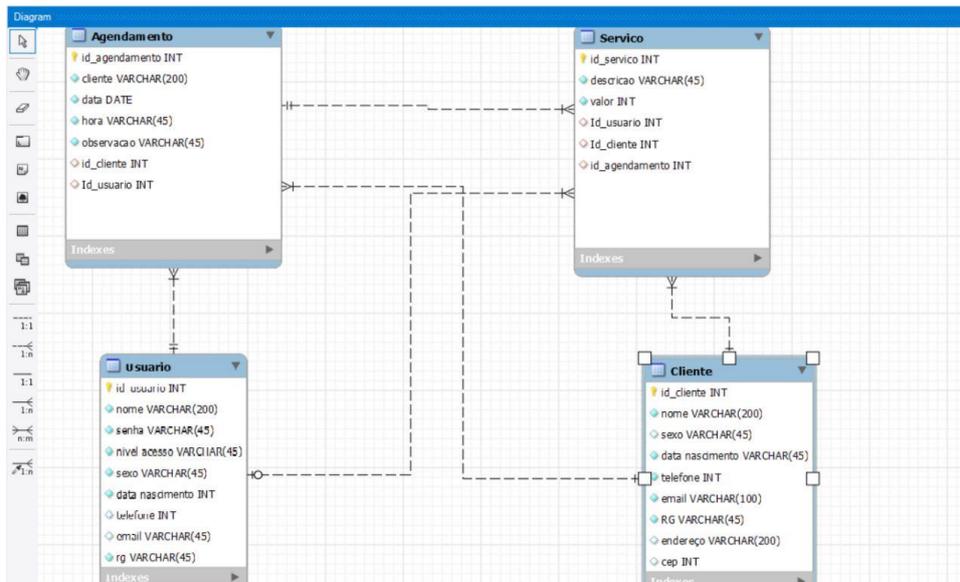
- Tabela Cliente
- Usuario
- Agendamento
- Serviço

3.1.1 MER - MODELAGEM ENTIDADE RELACIONAMENTO

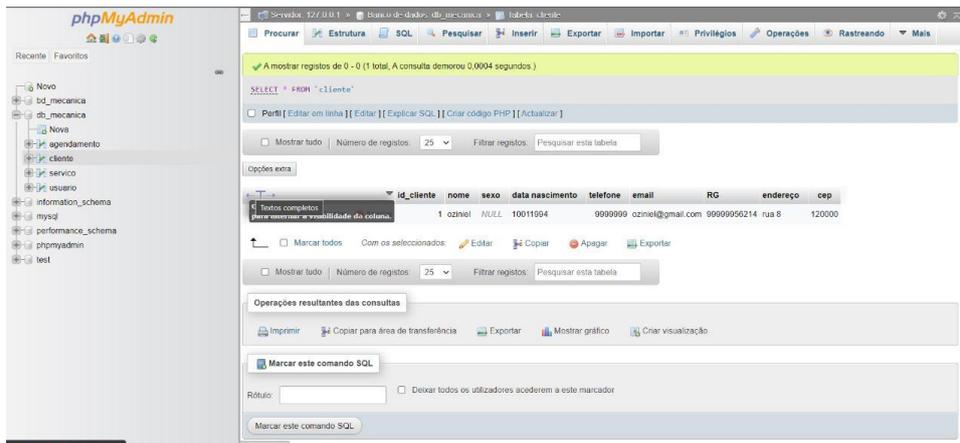


OBJ OBJ

DER - DIAGRAMA ENTIDADE RELACIONAMENTO



3.1.3 FÍSICO



3.1.1 e 3.1.2. Tirar print dos comandos de criação e das tabelas do banco criado.

```

-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES';

-----
-- Schema DB_Mecanica
-----

-----
-- Schema DB_Mecanica
-----

CREATE SCHEMA IF NOT EXISTS `DB_Mecanica` DEFAULT CHARACTER SET utf8 ;

-- Schema new_schema1

USE `DB_Mecanica` ;

-----
-- Table `DB_Mecanica`.`Cliente`
-----

```

```

-----
CREATE TABLE IF NOT EXISTS `DB_Mecanica`.`Cliente` (
  `id_cliente` INT NOT NULL AUTO_INCREMENT,
  `nome` VARCHAR(200) NOT NULL,
  `sexo` VARCHAR(45) NULL,
  `data nascimento` VARCHAR(45) NOT NULL,
  `telefone` INT NOT NULL,
  `email` VARCHAR(100) NOT NULL,
  `RG` VARCHAR(45) NOT NULL,
  `endereço` VARCHAR(200) NULL,
  `cep` INT NULL,
  PRIMARY KEY (`id_cliente`))
ENGINE = InnoDB;

```

```

-----
-- Table `DB_Mecanica`.`Usuario`
-----

```

```

CREATE TABLE IF NOT EXISTS `DB_Mecanica`.`Usuario` (
  `id_usuario` INT NOT NULL AUTO_INCREMENT,
  `nome` VARCHAR(200) NOT NULL,
  `senha` VARCHAR(45) NOT NULL,
  `nivel acesso` VARCHAR(45) NOT NULL,
  `sexo` VARCHAR(45) NOT NULL,
  `data nascimento` INT NOT NULL,
  `telefone` INT NULL,
  `email` VARCHAR(45) NULL,
  `rg` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`id_usuario`))
ENGINE = InnoDB;

```

```

-----
-- Table `DB_Mecanica`.`Servico`
-----

```

```

CREATE TABLE IF NOT EXISTS `DB_Mecanica`.`Servico` (
  `id_servico` INT NOT NULL AUTO_INCREMENT,
  `descricao` VARCHAR(45) NOT NULL,
  `valor` INT NOT NULL,
  `Id_usuario` INT NULL,
  `Id_cliente` INT NULL,
  `id_agendamento` INT NULL,
  PRIMARY KEY (`id_servico`))
ENGINE = InnoDB;

```

3.2 PROGRAMAÇÃO ORIENTADA A OBJETOS

3.2.1 DIAGRAMA DE CLASSES

```
package Model;

import java.util.Date;

/**
 * @author priquelme
 */
public class Cliente extends Pessoa {
    private String endereco;
    private String cep;

    public Cliente(int id, String nome, char sexo, String dataNascimento, String telefone, String email, String rg, String endereco) {
        super(id, nome, sexo, dataNascimento, telefone, email, rg);
        this.endereco = endereco;
        this.cep = cep;
    }

    public Cliente(int id, String nome, String endereco, String cep) {
        super(id, nome);
        this.endereco = endereco;
        this.cep = cep;
    }
}
```

```
public void setNome(String nome) {
    this.nome = nome;
}

public char getSexo() {
    return sexo;
}

public void setSexo(char sexo) {
    this.sexo = sexo;
}

public Date getDataNascimento() {
    return dataNascimento;
}

public void setDataNascimento(Date dataNascimento) {
    this.dataNascimento = dataNascimento;
}

public String getTelefone() {
    return telefone;
}

public void setTelefone(String telefone) {
    this.telefone = telefone;
}
```

```
public Pessoa(int id, String nome, char sexo, String dataNascimento, String telefone, String email, String rg) {
    this.id = id;
    this.nome = nome;
    this.sexo = sexo;
    try {
        this.dataNascimento = new SimpleDateFormat("dd/MM/yyyy").parse(dataNascimento);
    } catch (ParseException ex) {
        Logger.getLogger(Pessoa.class.getName()).log(Level.SEVERE, null, ex);
    }
    this.telefone = telefone;
    this.email = email;
    this.rg = rg;
}

public int getId() {
    return id;
}

public void setId(int id) {
    this.id = id;
}

public String getNome() {
    return nome;
}

public void setNome(String nome) {
    this.nome = nome;
}
```

```

...4 lines
package Model;

/**...4 lines */
public class Servico {

    private int id;
    private String descricao;

    public Servico(int id, String descricao) {
        this.id = id;
        this.descricao = descricao;
    }

    public String getDescricao() {
        return descricao;
    }

    public void setDescricao(String descricao) {
        this.descricao = descricao;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }
}

```

```

1 ...4 lines
2 package Model;
3
4
5
6
7 import java.util.Date;
8
9 /**...4 lines */
10
11 public class Usuario extends Pessoa {
12
13     private String senha;
14     private String nivelAcesso;
15
16     public Usuario( int id, String nome, String senha) {
17         super(id, nome);
18         this.senha = senha;
19     }
20
21
22
23     public Usuario(int id, String nome, char sexo, String dataNascimento, String telefone, String email, String rg, String se
24         super(id, nome, sexo, dataNascimento, telefone, email, rg);
25         this.senha = senha;
26         this.nivelAcesso = nivelAcesso;
27     }
28
29     public String getSenha() {
30         return senha;
31     }
32
33     public void setSenha(String senha) {
34         this.senha = senha;
35     }
36

```

```

public String getEndereco() {
    return endereco;
}

public void setEndereco(String endereco) {
    this.endereco = endereco;
}

public String getCep() {
    return cep;
}

public void setCep(String cep) {
    this.cep = cep;
}

}

```

3.2.2 CÓDIGOS DO SISTEMA

```
public void setServico(Servico servico) {
    this.servico = servico;
}

public float getValor() {
    return valor;
}

public void setValor(float valor) {
    this.valor = valor;
}

public Date getData() {
    return data;
}

public void setData(Date data) {
    this.data = data;
}

public String getObservacao() {

    try {
        this.data = new SimpleDateFormat(pattern: "dd/MM/yyyy HH:mm").parse(source: data);
    } catch (ParseException ex) {
        Logger.getLogger(name: Agendamento.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);
    }
}

public int getId() {
    return id;
}

public void setId(int id) {
    this.id = id;
}

public Cliente getCliente() {
    return cliente;
}

public void setCliente(Cliente cliente) {
    this.cliente = cliente;
}

public Servico getServico() {
    return servico;
}
```

```
UsuarioDAO usuarioDAO = new UsuarioDAO();
Usuario usuarioAutenticado = usuarioDAO.selectPorNomeESenha(usuario);

if(usuarioAutenticado != null){
    menuPrincipal menu = new menuPrincipal();
    menu.setVisible(b: true);
    this.view.dispose();
}

}

public void fizTarefa(){

    System.out.println(x: "Busquei algo no banco de dados");
    this.view.exibeMensagem(mensagem: "Logado com sucesso");
}

}
```

```

        if(maiorId < id){
            maiorId = id;
        }
    }

    return maiorId + 1;
}
}

package Model.DAO;

import Model.Agendamento;
import java.util.ArrayList;

public class AgendamentoDAO {

    public void insert(Agendamento agendamento){

        if(agendamento.getId() == 0){
            agendamento.setId(id: proximoId());
            Banco.agendamento.add(e: agendamento);
        }

    }

    public boolean update(Agendamento agendamento){

        for (int i = 0; i < Banco.agendamento.size(); i++) {
            if(idSaoIguais(agendamento:Banco.agendamento.get(index: i),agendamento:agendamento)){
                Banco.agendamento.set(index: i, element:agendamento);
                return true;
            }
        }
        return false;
    }

    public static ArrayList<Usuario> usuario;
    public static ArrayList<Cliente> cliente;
    public static ArrayList<Servico> servico;
    public static ArrayList<Agendamento> agendamento;

    public static void inicia(){

        //Instancia os Objetos
        usuario = new ArrayList<Usuario>();
        cliente = new ArrayList<Cliente>();
        servico = new ArrayList<Servico>();
        agendamento = new ArrayList<Agendamento>();

        //criando elementos
        Usuario usuario1 = new Usuario(id: 1, nome: "pedro", sexo: 'M', dataNascimento: "09/05/1996", telefone: "30212121", email: "pedro8g);
        Usuario usuario2 = new Usuario(id: 2, nome: "estagiario", sexo: 'M', dataNascimento: "09/05/1996", telefone: "30212122", email: "es

        Cliente cliente1 = new Cliente(id: 1, nome: "Alan Figueiredo", sexo: 'M', dataNascimento: "30/01/1995", telefone: "30212126", email:
        Cliente cliente2 = new Cliente(id: 2, nome: "Judite Oliveira", sexo: 'F', dataNascimento: "30/08/1994", telefone: "30212127", email:
        Cliente cliente3 = new Cliente(id: 3, nome: "Paulo Ricardo", sexo: 'M', dataNascimento: "24/04/1997", telefone: "30212128", email: "
        Cliente cliente4 = new Cliente(id: 4, nome: "Neymar Santos", sexo: 'M', dataNascimento: "20/08/1995", telefone: "30212130", email:
        Cliente cliente5 = new Cliente(id: 5, nome: "Debruine Pipoqueiro", sexo: 'M', dataNascimento: "13/09/1992", telefone: "30212131",
        Cliente cliente6 = new Cliente(id: 6, nome: "Matuidi Pareira", sexo: 'M', dataNascimento: "17/08/1994", telefone: "302122324", cha
        Cliente cliente7 = new Cliente(id: 7, nome: "Marra de Souza", sexo: 'F', dataNascimento: "25/03/1993", telefone: "302121292", email:
        Cliente cliente8 = new Cliente(id: 8, nome: "Aline Ferrari", sexo: 'F', dataNascimento: "04/03/1990", telefone: "302121451", email:
        Cliente cliente9 = new Cliente(id: 9, nome: "Samuel Coltinho", sexo: 'M', dataNascimento: "09/02/1998", telefone: "302121189", cha
        Cliente cliente10 = new Cliente(id: 10, nome: "Felipe Luis", sexo: 'M', dataNascimento: "12/06/1999", telefone: "3021212478", email:

```

- ▼  projectGarage
 - ▼  Source Packages
 - ▼  Controller
 -  AgendaController.java
 -  LoginController.java
 -  MenuPrincipalController.java
 - ▼  Controller.Helper
 -  LoginHelper.java
 - ▼  Model
 -  Agendamento.java
 -  Cliente.java
 -  Pessoa.java
 -  Servico.java
 -  Usuario.java
 - ▼  Model.DAO
 -  AgendamentoDAO.java
 -  Banco.java
 -  ClienteDAO.java
 -  ServicoDAO.java
 -  UsuarioDAO.java
 - ▼  view
 -  Agenda.java
 - ▼  view
 -  Agenda.java
 -  Main.java
 -  login.java
 -  menuPrincipal.java
 - >  view.imagens
 - >  view.imagens.Icones
 - >  Test Packages
 - >  Libraries
 - >  Test Libraries

```

package Controller;

import view.Agenda;
import view.menuPrincipal;

/**
 *
 * @author priquelme
 */
public class MenuPrincipalController {

    private final menuPrincipal view;

    public MenuPrincipalController(menuPrincipal view) {
        this.view = view;
    }

    public void navegarParaAgenda() {

        Agenda agenda = new Agenda();
        agenda.setVisible(b: true);
    }
}

```

```

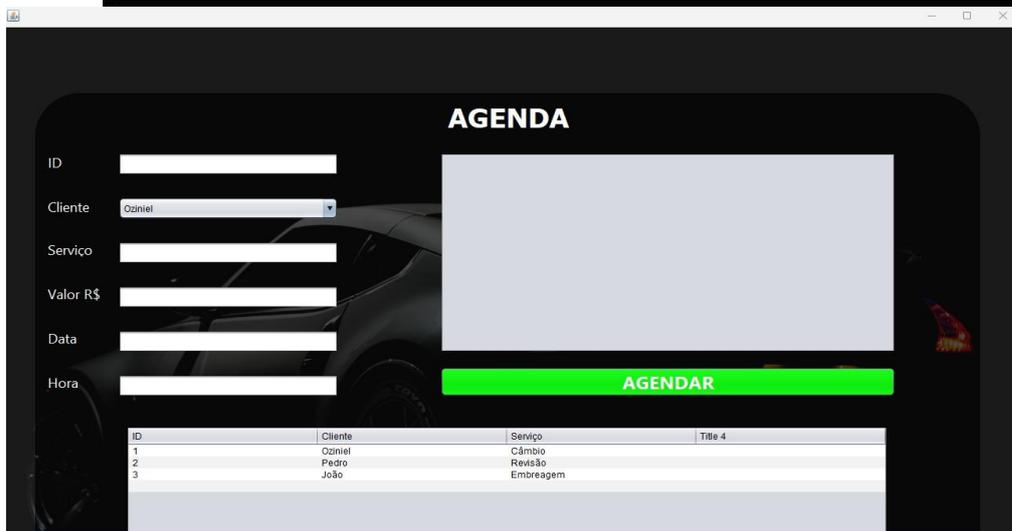
Start Page x AgendaController.java x
Source History
1 1 /*
2 2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3 3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4 4  */
5 5 package Controller;
6 6
7 7 import Model.Agendamento;
8 8 import Model.DAO.AgendamentoDAO;
9 9 import java.util.ArrayList;
10 10 import view.Agenda;
11 11
12 12 /**
13 13  *
14 14  * @author priquelme
15 15  */
16 16 public class AgendaController {
17 17
18 18     private final Agenda view;
19 19
20 20     public AgendaController(Agenda view) {
21 21         this.view = view;
22 22     }
23 23
24 24     public void atualizaTabela(){
25 25
26 26         AgendamentoDAO agendamentoDAO= new AgendamentoDAO();
27 27         ArrayList<Agendamento> agendamentos = agendamentoDAO.selectAll();

```

3.2.3 IMAGENS DO SISTEMA

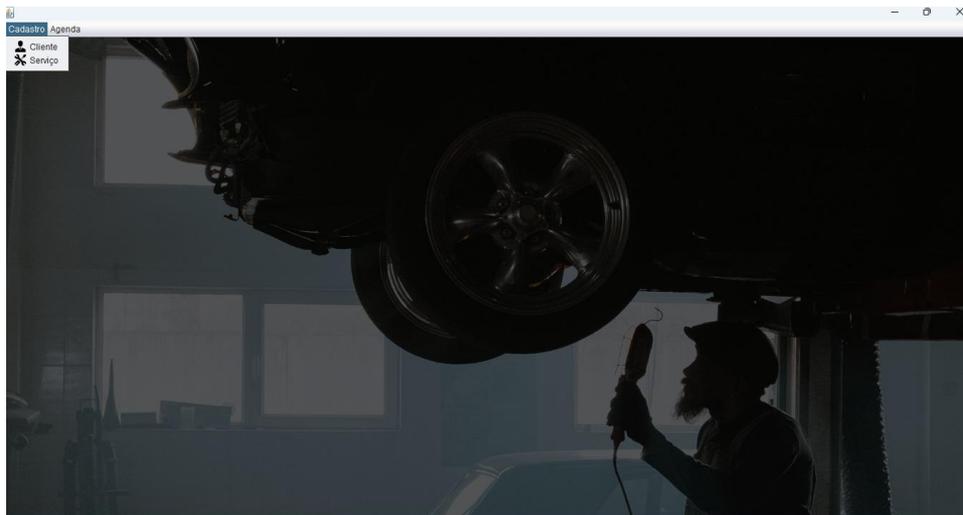


The image shows a login window with a dark background. At the top, the word "Login" is displayed in white. Below it, the word "Usuario" is followed by a text input field containing "priquelme". Underneath, the word "Senha" is followed by a password input field containing "*****". A white button labeled "Entrar" is positioned below the password field. In the background, there is a faint image of a car and a gear, with the word "GARAGE" partially visible.



The image shows an "AGENDA" form with a dark background. On the left, there are several input fields: "ID", "Cliente" (with a dropdown menu showing "Ozaniel"), "Serviço", "Valor R\$", "Data", and "Hora". A large grey rectangular area is present to the right of these fields. Below the form is a green button labeled "AGENDAR". At the bottom, there is a table with the following data:

ID	Cliente	Serviço	Title 4
1	Ozaniel	Câmbio	
2	Pedro	Revisão	
3	João	Entrega	



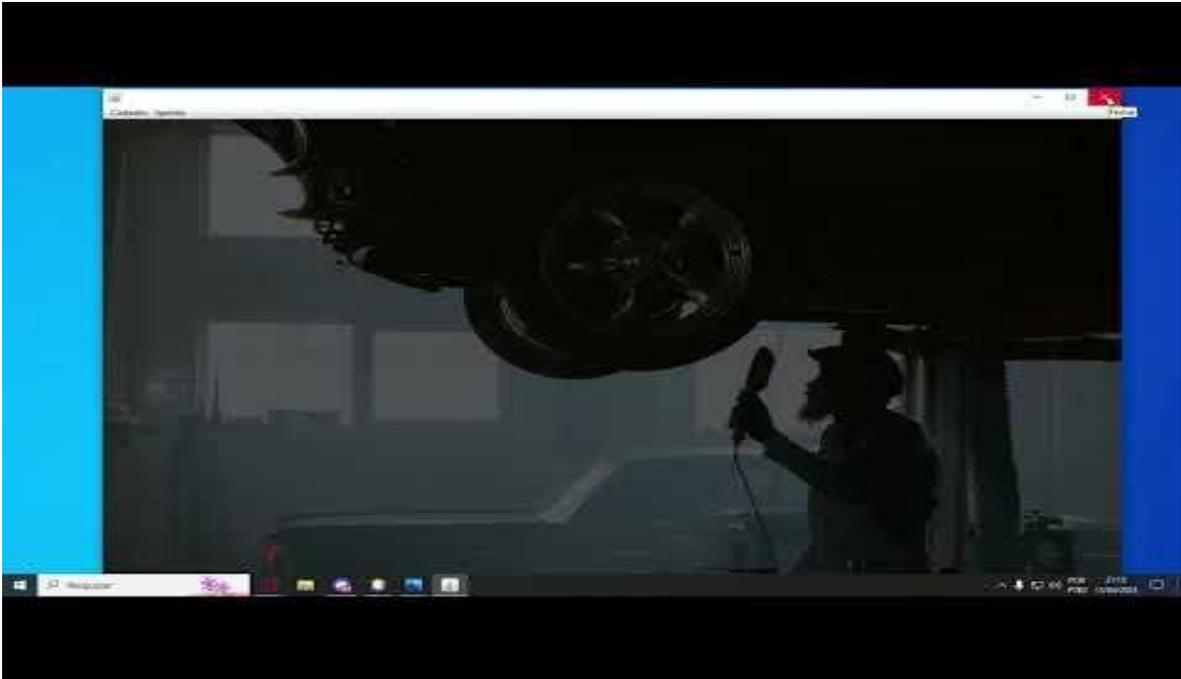
4 CONCLUSÃO

Em conclusão, o desenvolvimento deste software para oficina mecânica representa uma solução moderna e eficaz para gerenciar todo o fluxo de trabalho e atendimento aos clientes. Com a possibilidade de automatizar processos, integrar informações e controlar o estoque, a oficina poderá oferecer um atendimento mais rápido e eficiente, otimizando o tempo dos mecânicos e melhorando a satisfação dos clientes. Além disso, a possibilidade de gerar relatórios e estatísticas permitirá um maior controle financeiro e uma melhor gestão do negócio. A implementação desse software pode trazer diversos benefícios para a oficina mecânica, tais como aumento da produtividade, melhoria da qualidade dos serviços prestados, redução de custos e aumento da lucratividade. Com isso, concluímos que o investimento em tecnologia é essencial para a competitividade e sucesso do negócio na atualidade.

ANEXOS

Link com video da gravação da apresentação:

[Trabalho PI - Tecnologia em análise e desenvolvimento de sistemas](#)





RELATÓRIO FINAL DAS ATIVIDADES DE EXTENSÃO

1. IDENTIDADE DA ATIVIDADE

RELATÓRIO:

CURSO: Análise e Desenvolvimento de Sistemas e Gestão de Tecnologia da Informação

MÓDULO: Desenvolvimento Desktop

PROFESSOR RESPONSÁVEL: Sidney Gitcoff Telles

ESTUDANTE:

PERÍODO DE REALIZAÇÃO: 02/2023 a 04/2023

2. DESENVOLVIMENTO

Contextualização

Desafio

Cronograma das Ações

Síntese das Ações

a. Aspectos positivos

b. Dificuldades encontradas

c. Resultados atingidos

d. Sugestões / Outras observações

3. EQUIPE DOS ESTUDANTES NO PROJETO

RA <u>1012022101042</u>	NOME <u>André Luiz de Melo Filho</u>
RA <u>R1012022101264</u>	NOME <u>João Paulo Godoi de Andrade</u>
<u>RA 1012022101458</u>	NOME <u>Oziniel Ewerton Silva</u>
RA <u>1012022100206</u>	NOME <u>Pedro Riquelme Borsone</u>
RA	NOME

Curso de Análise e Desenvolvimento de Sistemas e Gestão de Tecnologia da Informação

Módulo Desenvolvimento Desktop

Cronograma de Validação - Projeto Integrado

Unidade Estudo	Participação no Projeto	Data da Validação
Banco de Dados	Construção do banco de dados com MER, DER E Físico.	29/03
Programação Orientada a Objetos	Desenvolvimento as telas e da parte lógica do sistema que conectará com o banco de dados	30/03
<p><u>Descrição do Projeto:</u> criar um sistema, um módulo reduzido, que seja utilizado em qualquer estabelecimento comercial ou empresarial. Esse sistema deverá contemplar atividades básicas da empresa, como controle de produtos, entrada e saída, controle de vendas, módulos menores que possam ser criados e executados neste trimestre.</p>		