



```
import java.util.ArrayList;
import java.util.Date;
public class Agendamento {
    public Profissionais profissional;
    public Clientes cliente;
    private String data;
    private String horario;
    private String status;
    private Servicos servicoAgendado;

    // Construtor do Agendamento
    public Agendamento(Clientes cliente, Profissionais profissional, Servicos
servico, String data, String horario) {
        this.cliente = cliente;
        this.profissional = profissional;
        this.servicoAgendado = servico;
        this.data = data;
        this.horario = horario;
        this.status = "Pendente";
    }

    // Métodos Getters e Setters
    public Profissionais getProfissional() {
        return this.profissional;
    }

    public void setProfissional(Profissionais prof) {
        this.profissional = prof;
    }

    public Clientes getCliente() {
        return this.cliente;
    }

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

    public String getData() {
        return this.data;
    }

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

    public String getHorario() {
        return this.horario;
    }

    public void setHorario(String horario) {
        this.horario = horario;
    }
}
```

```
public String getStatus() {
    return this.status;
}

public void setStatus(String status) {
    this.status = status;
}

public Servicos getServicoAgendado() {
    return this.servicoAgendado;
}

public void setServicoAgendado(Servicos servicoAgendado) {
    this.servicoAgendado = servicoAgendado;
}

// Métodos de lógica de agendamento (anteriormente vazios)
public void pesquisarCidade(String cidade, ArrayList<Profissionais>
listaProfissionais) {
    boolean encontrado = false;
    for (Profissionais p : listaProfissionais) {
        if (p.getCidadeAtend() != null &&
p.getCidadeAtend().equalsIgnoreCase(cidade)) {
            System.out.println("- " + p.getNome() + " (" +
p.getServicoPrincipal() + ")");
            encontrado = true;
        }
    }
    if (!encontrado) {
        System.out.println("Nenhum profissional encontrado nesta cidade.");
    }
}

public void pesquisarServico(String servico, ArrayList<Profissionais>
listaProfissionais) {
    System.out.println("Pesquisando profissionais que oferecem o serviço: " +
servico);
    boolean encontrado = false;
    for (Profissionais p : listaProfissionais) {
        if (p.ofereceServico(servico)) {
            System.out.println("- " + p.getNome() + " (Valor médio: R$" +
p.getValorGeral() + ")");
            encontrado = true;
        }
    }
    if (!encontrado) {
        System.out.println("Nenhum profissional encontrado para este
serviço.");
    }
}

public void apresentarProfissionais(ArrayList<Profissionais>
listaProfissionais) {
    System.out.println("Lista de Todos os Profissionais:");
    if (listaProfissionais.isEmpty()) {
        ...
    }
}
```

```
        } else {
            for (Profissionais p : listaProfissionais) {
                System.out.println("-----");
                System.out.println("Nome: " + p.getNome());
                System.out.println("Categoria: " + p.getCategoria());
                System.out.println("Serviço Principal: " +
p.getServicoPrincipal());
                System.out.println("Cidade de Atendimento: " +
p.getCidadeAtend());
                System.out.println("Avaliação: " + p.getAvaliacao());
                System.out.println("-----");
            }
        }
    }

    public void escolherProfissional(Profissionais profissional) {
        this.setProfissional(profissional);
        System.out.println("Profissional " + profissional.getNome() + " escolhido
para o agendamento.");
    }

    public void verificarDisponibilidade() {
        if (this.profissional != null) {
            System.out.println("Verificando disponibilidade para " +
this.profissional.getNome() + ":");

            this.profissional.verificarAgenda();
            System.out.println("Nenhum profissional selecionado para verificar a
disponibilidade.");
        }
    }

    public void escolherData(String data) {
        this.setData(data);
        System.out.println("Data " + data + " selecionada.");
    }

    public void escolherHorario(String horario) {
        this.setHorario(horario);
        System.out.println("Horário " + horario + " selecionado.");
    }

    public void escolherFormaDePag(String forma) {
        System.out.println("Forma de pagamento " + forma + " selecionada.");
    }

    public void realizarPagamento() {
        if (this.cliente != null && this.servicoAgendado != null) {
            float valorTotal = this.servicoAgendado.getValor();
            if (this.cliente.getSaldo() >= valorTotal) {
                this.cliente.setSaldo(this.cliente.getSaldo() - valorTotal);
                this.setStatus("Confirmado");
                System.out.println("Pagamento de R$" + valorTotal + " realizado
com sucesso! Agendamento confirmado.");
                // Aqui o valor pago será transferido para o saldo do profissional
                this.profissional.setSaldo(this.profissional.getSaldo() +
```

```
        System.out.println("Saldo do profissional: R$" +  
this.profissional.getSaldo());  
    } else {  
        System.out.println("Saldo insuficiente para realizar o pagamento.  
Saldo atual: R$" + this.cliente.getSaldo());  
        this.setStatus("Aguardando Pagamento");  
    }  
}  
  
} else {  
    System.out.println("Cliente ou serviço não definidos para  
pagamento.");  
}  
}  
  
}  
  
public void confirmarAgendamento() {  
    System.out.println("Agendamento para " + getCliente().getNome() + " com "  
+ getProfissional().getNome() + " do serviço " +  
getServiçoAgendado().getNomeServiço() +  
        " em " + getData() + " às " + getHorario() + " está " +  
getStatus() + ".");  
}  
}
```

```
import java.util.ArrayList;
import java.util.Scanner;

public class Clientes extends Usuarios {
    // Atributos específicos de Cliente
    protected float avaliacao;
    public String agendar;
    public String servicoSolic;
    public Profissionais profissionalEscolhido;
    public int qntAtendimento;
    public float avaliacaoCliente;
    public float notaCliente;

    //cnstrutor
    public Clientes(String nome, String data, int cpf, String categoria, String
    usuario, String senha) { // Adicionado 'usuario' para o construtor da superclasse
        super(nome, data, cpf, categoria, usuario, senha);
        this.avaliacao = 0.0f;
        this.qntAtendimento = 0;
    }

    // Métodos da classe
    public void cadastrarCliente() {
        System.out.println("CLIENTE CADASTRADO!");
        System.out.println("NOME: " + getNome());
        System.out.println("DATA DE NASCIMENTO: " + getDataNasc());
        System.out.println("CPF/USUÁRIO: " + getCpf());
        System.out.println("CATEGORIA: " + getCategoria());
        System.out.println("USUÁRIO: " + getUsuario());
    }

    public void realizarNovoAgendamento(String s) {
        System.out.println("Gostaria de realizar um novo agendamento?
(SIM/NAO)"); // [cite: 92]
        this.agendar = s;

        if ("SIM".equalsIgnoreCase(this.agendar)) {
            System.out.println("Informe a cidade onde gostaria de ser
atendido(a)"); // [cite: 96]
        }
    }

    // Método para pesquisar cidade e listar profissionais disponíveis nela
    public void pesquisarCidade(String cidade, ArrayList<Profissionais>
    listaDeProfissionais) { //
        this.setCidadeAtend(cidade);
        System.out.println("Pesquise o serviço da sua preferência");
        System.out.println("Seguem os profissionais em " + getCidadeAtend() +
            " que realizam atendimentos:");
        System.out.println("LISTA DE PROFISSIONAIS NA CIDADE:");

        boolean encontrouProfissional = false;
        for (Profissionais p : listaDeProfissionais) {
```

```
// Verifica se o profissional atende na cidade
    if (p.getCidadeAtend() != null &&
p.getCidadeAtend().equalsIgnoreCase(cidade)) {
        apresentarProfissional(p);
        encontrouProfissional = true;
    }
}
if (!encontrouProfissional) {
    System.out.println("Nenhum profissional encontrado nesta cidade.");
}
}

// Método para pesquisar serviço e listar profissionais que o oferecem
public void pesquisarServico(String servico, ArrayList<Profissionais>
listaDeProfissionais) { //
    this.servicoSolic = servico;
    System.out.println("Seguem os profissionais que realizam o atendimento de
" + getServicoSolic() + ":" ); // [cite: 104]
    System.out.println("LISTA DE PROFISSIONAIS POR SERVIÇO:");

    boolean encontrouProfissional = false;
    for (Profissionais p : listaDeProfissionais) {
        // Verifica se o profissional oferece o serviço solicitado
        // Supondo que Profissionais tenha um método para verificar se oferece
um serviço
        if (p.ofereceServico(servico)) {
            apresentarProfissional(p); // Apresenta o profissional
            encontrouProfissional = true;
        }
    }
    if (!encontrouProfissional) {
        System.out.println("Nenhum profissional encontrado que ofereça este
serviço.");
    }
}

// Método para apresentar detalhes de um profissional
public void apresentarProfissional(Profissionais profissional) { //
    this.profissionalEscolhido = profissional;
    System.out.println("--- Dados do Profissional ---");
    System.out.println("Nome: " + profissional.getNome());
    System.out.println("Cidade de Atendimento: " +
profissional.getCidadeAtend());
    System.out.println("Biografia: " + profissional.getBiografia());
    System.out.println("Serviço Principal: " +
profissional.getServicoPrincipal());
    System.out.println("Serviços Oferecidos:");
    profissional.listarServicos();
    System.out.println("Avaliação Média: " + profissional.getAvaliacao());
    System.out.println("-----");
}
}

public void selecionarProfissional(Profissionais profissionalSelecionado) {
// 
    this.profissionalEscolhido = profissionalSelecionado;
```

```
}

public void verificarAgenda(Profissionais profissional) {
    System.out.println("Verificando agenda de " + profissional.getNome() +
":");
    System.out.println("Datas Disponíveis: " + profissional.getDatasDispon());
    System.out.println("Horários Disponíveis: " +
profissional.getHorariosDispon());
    // Aqui o cliente veria as datas e horários e escolheria um
}

public void escolherData(String data) {
    System.out.println("Data escolhida para o agendamento: " + data);
}

public void escolherHorario(String horario) {
    System.out.println("Horário escolhido para o agendamento: " + horario);
}

public void escolherPagamento(String formaPagamento) {
    System.out.println("Forma de pagamento escolhida: " + formaPagamento);
    // Aqui o sistema apresenta opções (cartão, pix, saldo)
}

public void depositar(float valor) {
    setSaldo(getSaldo() + valor);
    System.out.println("Depósito de R$" + valor + " realizado com sucesso.
Novo saldo: R$" + getSaldo());
}

public void confirmarAgendamento(Agendamento agendamento) {
    System.out.println("Agendamento confirmado com sucesso!");
    System.out.println("Detalhes: " + agendamento.getCliente().getNome() + " "
com " + agendamento.getProfissional().getNome() + " em "
+ agendamento.getData() + " às " + agendamento.getHorario());
}

public void realizarCheckIn() {
    System.out.println("Check-in realizado com sucesso!");
}

public void realizarCheckOut() {
    System.out.println("Check-out realizado com sucesso!");
    setQntAtendimento(getQntAtendimento() + 1);
    System.out.println("Por favor, avalie o atendimento");
}

// Métodos especiais (Getters e Setters específicos de Clientes)
public String getAgendar() {
    return this.agendar;
}

public void setAgendar(String s) {
    this.agendar = s;
```

```
    }

    public String getServicoSolic() {
        return this.servicoSolic;
    }

    public void setServicoSolic(String servicoSolic) {
        this.servicoSolic = servicoSolic;
    }

    public int getQntAtendimento() {
        return this.qntAtendimento;
    }

    public void setQntAtendimento(int qntAtendimento) {
        this.qntAtendimento = qntAtendimento;
    }

    public float getAvaliacaoCliente() {
        return this.avaliacaoCliente;
    }

    public void setAvaliacaoCliente(float f) {
        this.avaliacaoCliente = f;
    }

    public float getNotaCliente() {
        return this.notaCliente;
    }

    public void setNotaCliente(float f) {
        this.notaCliente = f;

        if (this.qntAtendimento > 0) {
            this.avaliacaoCliente = (this.avaliacaoCliente * (this.qntAtendimento
- 1) + f) / this.qntAtendimento; // Lógica para média
        } else {
            this.avaliacaoCliente = f;
        }
    }
}
```

```
public abstract class Usuarios {
    protected String nome;
    protected String dataNasc;
    protected int cpf;
    protected String categoria;
    protected String usuario;
    protected String senha;
    protected String cidadeAtend;
    protected float saldo;

    // Construtor Usuarios
    public Usuarios(String nome, String dataNasc, int cpf, String categoria,
String usuario, String senha) {
        this.nome = nome;
        this.dataNasc = dataNasc;
        this.cpf = cpf;
        this.categoria = categoria;
        this.usuario = usuario;
        this.senha = senha;
        this.saldo = 0.0f;
    }

    // Métodos Getters e Setters
    public String getNome() {
        return this.nome;
    }

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

    public String getDataNasc() {
        return this.dataNasc;
    }

    public void setDataNasc(String data) {
        this.dataNasc = data;
    }

    public int getCpf() {
        return this.cpf;
    }

    public void setCpf(int cpf) {
        this.cpf = cpf;
    }

    public String getCategoria() {
        return this.categoria;
    }

    public void setCategoria(String categoria) {
        this.categoria = categoria;
    }
}
```

```
public String getUsuario() {
    return this.usuario;
}

public void setUsuario(String usuario) {
    this.usuario = usuario;
}

protected String getSenha() {
    return this.senha;
}

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

public String getCidadeAtend() {
    return this.cidadeAtend;
}

public void setCidadeAtend(String cidade) {
    this.cidadeAtend = cidade;
}

public float getSaldo() {
    return this.saldo;
}

public void setSaldo(float saldo) {
    this.saldo = saldo;
}
}
```

```
public class AvaliarCliente {
    public Profissionais profissional;
    public Clientes cliente;
    public float nota;
    public String comentario;

    public AvaliarCliente(Profissionais p, Clientes c, float n, String comentario)
    {
        this.profissional = p;
        this.cliente = c;
        this.nota = n;
        this.comentario = comentario;

        if (c.getQntAtendimento() > 0) {
            float novaAvaliacaoMedia = ((c.getAvaliacaoCliente() * c.getQntAtendimento()) + n) / (c.getQntAtendimento() + 1);
            c.setAvaliacaoCliente(novaAvaliacaoMedia);
        } else {
            c.setAvaliacaoCliente(n);
            c.setQntAtendimento(1);
        }

        System.out.println("Cliente " + c.getNome() + " avaliado com nota " + n +
" e comentário: '" + comentario + "'");
        System.out.println("Nova avaliação média do cliente: " +
c.getAvaliacaoCliente());
    }

    // Métodos Getters e Setters
    public Profissionais getProfissional() {
        return profissional;
    }

    public void setProfissional(Profissionais profissional) {
        this.profissional = profissional;
    }

    public Clientes getCliente() {
        return cliente;
    }

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

    public float getNota() {
        return nota;
    }

    public void setNota(float nota) {
        this.nota = nota;
    }

    public String getComentario() {
        return comentario;
    }
}
```

```
    }

    public void setComentario(String comentario) {
        this.comentario = comentario;
    }

    // Método toString
    @Override
    public String toString() {
        return "Avaliação de " + cliente.getNome() + " para " +
profissional.getNome() + // [cite: 458]
                ": nota " + nota + " - " + comentario;
    }
}
```

```
public class Servicos {

    public String nomeServiço;
    public float valor;
    public Profissionais profissional;

    // Construtor
    public Servicos(Profissionais profissional, String nomeServiço, float valor) {
        this.profissional = profissional;
        this.nomeServiço = nomeServiço;
        this.valor = valor;
    }

    // Métodos
    public void cadastrarServicos() {
        System.out.println("SERVIÇO CADASTRADO!");
        System.out.println("Profissional: " + getProfissional().getNome());
        System.out.println("Serviço cadastrado: " + getNomeServiço());
        System.out.println("Valor: R$" + getValor());
    }

    public String getNomeServiço() {
        return this.nomeServiço;
    }

    public void setNomeServiço(String nomeServiço) {
        this.nomeServiço = nomeServiço;
    }

    public float getValor() {
        return this.valor;
    }

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

    public Profissionais getProfissional() {
        return this.profissional;
    }

    public void setProfissional(Profissionais profissional) {
        this.profissional = profissional;
    }
}
```

```
public class AvaliarProfissional {
    private Profissionais profissional;
    private Clientes cliente;
    private float nota;
    private String comentario;

    // Construtor
    public AvaliarProfissional(Clientes cliente, Profissionais prof, float n,
String comentario) {
        this.cliente = cliente;
        this.profissional = prof;
        this.nota = n;
        this.comentario = comentario;

        if (prof.getQntAtend() > 0) {
            float novaAvaliacaoMedia = ((prof.getAvaliacao() * prof.getQntAtend())
+ n) / (prof.getQntAtend() + 1);
            prof.setAvaliacao(novaAvaliacaoMedia);
            prof.setQntAtend(prof.getQntAtend() + 1);
        } else {
            prof.setAvaliacao(n);
            prof.setQntAtend(1);
        }
        System.out.println("Profissional " + prof.getNome() + " avaliado com nota "
+ n + " e comentário: '" + comentario + "'");
        System.out.println("Nova avaliação média do profissional: " +
prof.getAvaliacao());
    }

    // Métodos Getters e Setters
    public float getNota() {
        return this.nota;
    }

    public void setNota(float n) {
        this.nota = n;
    }

    public Profissionais getProfissional() {
        return this.profissional;
    }

    public void setProfissional(Profissionais prof) {
        this.profissional = prof;
    }

    public Clientes getCliente() {
        return this.cliente;
    }

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

```
public String getComentario() {
    return comentario;
}

public void setComentario(String comentario) {
    this.comentario = comentario;
}

@Override
public String toString() {
    return "Avaliação do Cliente " + cliente.getNome() + " para o Profissional "
+ profissional.getNome() +
        ": Nota " + nota + " - Comentário: \\" + comentario + "\"";
}
```

```
public class CancelamentoAtendimento {
    public Agendamento agendamento;
    private String motivoCancelamento;

    public CancelamentoAtendimento(Agendamento ag) {
        this.agendamento = ag;
        if (this.agendamento != null) {
            this.agendamento.setStatus("Cancelado");
            System.out.println("Agendamento de " + ag.getCliente().getNome() + " com " + ag.getProfissional().getNome() + " cancelado.");
        } else {
            System.out.println("Agendamento não encontrado para cancelamento.");
        }
    }

    public CancelamentoAtendimento(Agendamento ag, String motivo) {
        this(ag);
        this.motivoCancelamento = motivo;
        System.out.println("Motivo do cancelamento: " + motivo);
    }

    public Agendamento getAgendamento() {
        return this.agendamento;
    }

    public void setAgendamento(Agendamento ag) {
        this.agendamento = ag;
    }

    public String getMotivoCancelamento() {
        return motivoCancelamento;
    }

    public void setMotivoCancelamento(String motivoCancelamento) {
        this.motivoCancelamento = motivoCancelamento;
    }
}
```

```
// Pacote, se houver
// package com.spafy.app;

import java.util.ArrayList;
import java.util.Scanner;

public class Main {
    private static ArrayList<Clientes> listaClientes = new ArrayList<>();
    private static ArrayList<Profissionais> listaProfissionais = new
ArrayList<>();
    private static ArrayList<Agendamento> listaAgendamentos = new ArrayList<>();

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Bem-vindo ao Spafy!");

        System.out.println("\n--- Cadastro de Usuários ---");

        Clientes cliente1 = new Clientes("Ana Silva", "15/03/1990", 123456789,
"Cliente", "anasilva", "senha123");
        cliente1.cadastrarCliente();
        cliente1.setSaldo(100.0f);
        listaClientes.add(cliente1);

        Profissionais prof1 = new Profissionais("Pedro Mendes", "20/07/1985",
987654321, "Profissional", "pedrom", "senhaabc", "Massagem Relaxante", "Rua das
Flores, 100", "Especialista em relaxamento e bem-estar.", "São Paulo");
        prof1.cadastrarProfissional();
        prof1.setSaldo(50.0f);
        listaProfissionais.add(prof1);

        // Adicionando serviços ao profissional
        prof1.cadastrarServico("Massagem Relaxante", 80.0f);
        prof1.cadastrarServico("Massagem Terapêutica", 120.0f);
        prof1.setDataDispon("Seg, Qua, Sex");
        prof1.setHorariosDispon("09:00-12:00, 14:00-18:00");

        Profissionais prof2 = new Profissionais("Maria Oliveira", "10/01/1992",
112233445, "Profissional", "mariaoli", "pass123", "Manicure", "Av. Principal,
500", "Dedicação e arte para suas unhas.", "São Paulo");
        prof2.cadastrarProfissional();
        prof2.cadastrarServico("Manicure", 30.0f);
        prof2.cadastrarServico("Pedicure", 40.0f);
        prof2.setDataDispon("Ter, Qui");
        prof2.setHorariosDispon("10:00-13:00, 15:00-19:00");
        listaProfissionais.add(prof2);

        // 2. Cliente realiza agendamento ---
        System.out.println("\n--- Cliente realizando agendamento ---");
        cliente1.realizarNovoAgendamento("SIM");
        String cidadeCliente = "São Paulo";
        cliente1.pesquisarCidade(cidadeCliente, listaProfissionais);

        String servicoDesejado = "Massagem Relaxante";
```

```
        Servicos servicoMassagem = null;
        for (Servicos s : prof1.listaServicos) { // Acessando listaServicos
            diretamente (pode ser via getter)
                if (s.getNomeServico().equalsIgnoreCase("Massagem Relaxante")) {
                    servicoMassagem = s;
                    break;
                }
            }

        if (servicoMassagem != null) {
            cliente1.selecionarProfissional(prof1);
            cliente1.verificarAgenda(prof1);

            String dataAgendamento = "2025-07-01";
            String horarioAgendamento = "10:00";
            cliente1.escolherData(dataAgendamento);
            cliente1.escolherHorario(horarioAgendamento);

            Agendamento agendamento1 = new Agendamento(cliente1, prof1,
servicoMassagem, dataAgendamento, horarioAgendamento);
            listaAgendamentos.add(agendamento1);

            System.out.println("\n--- Realizando Pagamento ---");
            cliente1.escolherPagamento("Saldo Spafy");
            agendamento1.realizarPagamento(); // O pagamento atualiza o status do
agendamento e saldos

            cliente1.confirmarAgendamento(agendamento1);
        } else {
            System.out.println("Serviço 'Massagem Relaxante' não encontrado para o
profissional " + prof1.getNome());
        }

        // --- 3. Simulação de Check-in/Check-out e Avaliação ---
        System.out.println("\n--- Simulação de Check-in/Check-out e Avaliação
---");
        if (!listaAgendamentos.isEmpty()) {
            Agendamento ag = listaAgendamentos.get(0); // Pegando o primeiro
agendamento para simular

            ag.getCliente().realizarCheckIn(); // Cliente faz check-in
            ag.getProfissional().realizarCheckIn(); // Profissional faz check-in

            System.out.println("\nAtendimento em andamento...");
            System.out.println("...");

            ag.getCliente().realizarCheckOut(); // Cliente faz check-out
            ag.getProfissional().realizarCheckOut(); // Profissional faz check-out

            // Cliente avalia o profissional
            System.out.println("\n--- Cliente avaliando profissional ---");
            AvaliarProfissional avaliaProf = new AvaliarProfissional(cliente1,
prof1, 4.5f, "Excelente massagem, muito relaxante!");
        }
    }
}
```

```
        System.out.println("\n--- Profissional avaliando cliente ---");
        AvaliarCliente avaliaCli = new AvaliarCliente(prof1, cliente1, 5.0f,
"Cliente muito pontual e educado!");

        System.out.println("\nAvaliação do Profissional: " +
prof1.getAvaliacao());
        System.out.println("Avaliação do Cliente: " +
cliente1.getAvaliacaoCliente());
    }

    // --- 4. Cancelamento de Agendamento ---
    System.out.println("\n--- Teste de Cancelamento de Agendamento ---");
    // Criar um novo agendamento para cancelar
    Clientes cliente2 = new Clientes("Bruno Costa", "01/01/2000", 998877665,
"Cliente", "brunoc", "segura");
    listaClientes.add(cliente2);
    cliente2.setSaldo(200.0f);

    Profissionais prof3 = new Profissionais("Carla Souza", "11/11/1988",
223344556, "Profissional", "carlas", "segredos", "Corte de Cabelo", "Rua da
Beleza, 77", "Estilista com anos de experiência.", "São Paulo");
    prof3.cadastrarProfissional();
    prof3.cadastrarServico("Corte Feminino", 70.0f);
    listaProfissionais.add(prof3);

    Servicos servicoCorte = null;
    for (Servicos s : prof3.listaServicos) {
        if (s.getNomeServico().equalsIgnoreCase("Corte Feminino")) {
            servicoCorte = s;
            break;
        }
    }

    if (servicoCorte != null) {
        Agendamento agCancelavel = new Agendamento(cliente2, prof3,
servicoCorte, "2025-07-05", "16:00");
        listaAgendamentos.add(agCancelavel);
        System.out.println("Agendamento criado para teste de cancelamento.
Status: " + agCancelavel.getStatus());

        CancelamentoAtendimento cancelamento = new
CancelamentoAtendimento(agCancelavel, "Mudança de planos do cliente.");
        System.out.println("Status do agendamento após tentativa de
cancelamento: " + agCancelavel.getStatus());
    }

    // --- 5. Verificando saldos ---
    System.out.println("\n--- Verificação de Saldos Finais ---");
    System.out.println("Saldo de " + cliente1.getNome() + ": R$" +
cliente1.getSaldo());
    System.out.println("Saldo de " + prof1.getNome() + ": R$" +
prof1.getSaldo());
    System.out.println("Saldo de " + cliente2.getNome() + ": R$" +
cliente2.getSaldo());
    System.out.println("Saldo de " + prof3.getNome() + ": R$" +
prof3.getSaldo());
```

```
        } scanner.close();  
    }
```

```
import java.util.ArrayList;

public class Profissionais extends Usuarios {
    protected float avaliacao;
    public String servicosGeral;
    public float valorGeral;
    public String endereco;
    public String biografia;
    public String datasDispon;
    public String horariosDispon;
    public String datasAgend;
    public String horariosAgend;
    public int qntAtend;
    public String servicoPrincipal;

    public ArrayList<Servicos> listaServicos;

    public Profissionais(String nome, String dataNasc, int cpf, String categoria,
String usuario, String senha) {
        super(nome, dataNasc, cpf, categoria, usuario, senha);
        this.qntAtend = 0;
        this.avaliacao = 0.0f;
        this.listaServicos = new ArrayList<>();
    }

    public Profissionais(String nome, String dataNasc, int cpf, String categoria,
String usuario, String senha, String servicoPrincipal, String endereco, String
biografia, String cidadeAtend) {
        this(nome, dataNasc, cpf, categoria, usuario, senha);
        this.servicoPrincipal = servicoPrincipal;
        this.endereco = endereco;
        this.biografia = biografia;
        this.setCidadeAtend(cidadeAtend);
    }

    public void cadastrarProfissional() {
        System.out.println("PROFISSIONAL CADASTRADO!");
        System.out.println("NOME: " + getNome());
        System.out.println("DATA DE NASCIMENTO: " + getDataNasc());
        System.out.println("CPF/USUÁRIO: " + getCpf());
        System.out.println("CATEGORIA: " + getCategoria());
        System.out.println("SERVIÇO PRINCIPAL: " + getServicoPrincipal());
        System.out.println("CIDADE DE ATENDIMENTO: " + getCidadeAtend());
    }

    // Método para adicionar um novo serviço oferecido pelo profissional
    public void cadastrarServico(String nomeServico, float valor) {
        Servicos novoServico = new Servicos(this, nomeServico, valor);
        listaServicos.add(novoServico);
        System.out.println("SERVIÇO ADICIONADO COM SUCESSO!");
        System.out.println("Profissional: " + this.getNome());
        System.out.println("Serviço: " + nomeServico);
        System.out.println("Valor: R$" + valor);
    }
}
```

```
// Método para listar todos os serviços oferecidos pelo profissional
public void listarServicos() {
    System.out.println("Serviços oferecidos por " + this.getNome() + ":");

    if (listaServicos.isEmpty()) {
        System.out.println("Nenhum serviço cadastrado.");
    } else {
        for (Servicos s : listaServicos) {
            System.out.println("- " + s.getNomeServico() + " (R$" +
s.getValor() + ")");
        }
    }
}

// Método para verificar a agenda do profissional
public void verificarAgenda() {
    System.out.println("Agendamento para o dia: " + getDatasAgend() + " às "
+
getHorariosAgend() + " horas");
}

public void editarPerfil() { //
    System.out.println("Editando perfil do profissional " + getNome() +
"...");
    //posteriormente colocaremos a função para editar dados
}

public void realizarCheckIn() {
    System.out.println("Check-in realizado com sucesso!");
}

public void realizarCheckOut() {
    System.out.println("Check-out realizado com sucesso!");
    setQntAtend(getQntAtend() + 1);
}

public void verificarCarteira() {
    System.out.println("SALDO ATUAL: R$" + getSaldo());
}

public void sacar(float valorSaque) {
    if (getSaldo() >= valorSaque) {
        setSaldo(getSaldo() - valorSaque);
        System.out.println("Saque de R$" + valorSaque + " realizado com
sucesso. Novo saldo: R$" + getSaldo());
    } else {
        System.out.println("Valor insuficiente, o saque não pode ser
realizado! Saldo atual: R$" + getSaldo());
    }
}

// Método para verificar se o profissional oferece um determinado serviço
public boolean ofereceServiço(String nomeServiço) {
    for (Servicos s : listaServicos) {
```

```
        return true;
    }
}
return false;
}

// Métodos Getters e Setters
public String getServicosGeral() {
    return this.servicosGeral;
}

public void setServicosGeral(String servicosGeral) {
    this.servicosGeral = servicosGeral;
}

public float getValorGeral() {
    return this.valorGeral;
}

public void setValorGeral(float v) {
    this.valorGeral = v;
}

public String getEndereco() {
    return this.endereco;
}

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

public String getBiografia() {
    return this.biografia;
}

public void setBiografia(String biografia) {
    this.biografia = biografia;
}

public String getDataDispon() {
    return this.datasDispon;
}

public void setDataDispon(String datasDispon) {
    this.datasDispon = datasDispon;
}

public String getHorariosDispon() {
    return this.horariosDispon;
}

public void setHorariosDispon(String horariosDispon) {
    this.horariosDispon = horariosDispon;
}
```

```
public String getDataAgend() {
    return this.datasAgend;
}

public void setDataAgend(String diasAgend) {
    this.datasAgend = diasAgend;
}

public String getHorariosAgend() {
    return this.horariosAgend;
}

public void setHorariosAgend(String horariosAgend) {
    this.horariosAgend = horariosAgend;
}

public float getAvaliacao() {
    return this.avaliacao;
}

public void setAvaliacao(float n) {
    this.avaliacao = n;
}

public int getQntAtend() {
    return this.qntAtend;
}

public void setQntAtend(int n) {
    this.qntAtend = n;
}

public String getServicoPrincipal() {
    return this.servicoPrincipal;
}

public void setServicoPrincipal(String servicoPrincipal) {
    this.servicoPrincipal = servicoPrincipal;
}
```

Este é o arquivo README do projeto. Você deve descrever aqui o seu projeto. Informe ao usuário (algum que não sabe nada sobre este projeto!) tudo que ele/ela precisa saber. Os comentários devem incluir, pelo menos:

TÍTULO DO PROJETO:

OBJETIVO DO PROJETO:

VERSÃO ou DATA:

COMO INICIAR O PROJETO:

AUTORES:

INSTRUÇÕES PARA O USUÁRIO: