

UNIVERSIDADE RUY BARBOSA - WYDEN

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Gabriella Mizrach Benevides

PROJETO WEB/MOBILE: APLICAÇÃO DE ATENDIMENTO MÉDICO

SALVADOR

[2025]

UNIVERSIDADE RUY BARBOSA - WYDEN
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Gabriella Mizrach Benevides

PROJETO WEB/MOBILE: APLICAÇÃO DE ATENDIMENTO MÉDICO

Trabalho apresentado como requisito parcial para aprovação na disciplina de Programação Orientada a Objetos em Java, do curso de Análise e Desenvolvimento de Sistemas do Centro Universitário Ruy Barbosa - Wyden.

Orientação: Prof. Heleno Cardoso

SALVADOR

[2025]

RESUMO

Este trabalho apresenta o desenvolvimento de uma aplicação web/mobile focada no gerenciamento de atendimentos médicos. O projeto parte do problema da gestão manual de agendas em clínicas, um processo frequentemente suscetível a erros e ineficiência. O objetivo principal foi criar um sistema digital que otimizasse o processo de marcação e controle de consultas. Para isso, a aplicação foi desenvolvida utilizando a linguagem Java e aplicando os conceitos fundamentais da Programação Orientada a Objetos (POO), como classes, herança e polimorfismo, para modelar as entidades principais como “Paciente, Medico e Consulta”. O sistema resultante permite o cadastro completo de pacientes e médicos, o agendamento de consultas com verificação de disponibilidade de horário, o cancelamento de procedimentos e a visualização da agenda diária. Conclui-se que o projeto atinge seus objetivos, entregando uma solução funcional que centraliza as informações e facilita a rotina administrativa da clínica.

Palavras-chave: Agendamento Médico, Java, Programação Orientada a Objetos, Gestão de Clínicas, Aplicação Web.

ABSTRACT

This work presents the development of a web/mobile application focused on managing medical appointments. The project arises from the problem of manual schedule management in clinics, a process that is often prone to errors and inefficiency. The main objective was to create a digital system that could optimize the process of scheduling and controlling consultations. To achieve this, the application was developed using the Java programming language and applying the fundamental concepts of Object-Oriented Programming (OOP), such as classes, inheritance, and polymorphism, to model the main entities such as “Patient, Doctor, and Appointment.” The resulting system enables the complete registration of patients and doctors, the scheduling of appointments with availability verification, the cancellation of procedures, and the visualization daily schedule. It is concluded that the project achieves its goals, delivering a functional solution that centralizes information and facilitates the administrative routine of the clinic.

Keywords: Medical Scheduling, Java, Object-Oriented Programming, Clinic Management, Web Application.

SUMÁRIO

Sumário

RESUMO.....	3
ABSTRACT	4
1 INTRODUÇÃO.....	13
2 OBJETIVOS DA APLICAÇÃO.....	14
3 FUNÇÕES/LISTA DE EVENTOS (FUNCIONALIDADES) – RF / RNF.....	15
3.1. Requisitos Funcionais (RF).....	15
3.2. Requisitos Não Funcionais (RNF).....	15
4 ESPECIFICAÇÃO DE PROGRAMAS.....	16
4.1. Layout da Tela.....	16
4.2. Regras de Negócio.....	16
4.3. Entidades Envolvidas (Classes).....	16
4.4. Tabelas.....	17
5 DIAGRAMA DE CLASSE.....	19
5.1. Artefato Gráfico.....	19
5.2. Dicionário de Dados.....	19
6 APLICAÇÃO WEB/MOBILE.....	21
6.1. Menu/Submenu.....	21
6.2. Telas Funcionais.....	21
6.3. Telas de Diálogo.....	21
7 CONSIDERAÇÕES FINAIS.....	23
REFERÊNCIAS.....	24

1 INTRODUÇÃO

O presente projeto tem como objetivo desenvolver uma aplicação web/mobile focada no gerenciamento de atendimentos e consultas em uma clínica médica de pequeno porte. O contexto para a escolha desta aplicação parte da observação de que muitas clínicas e consultórios ainda dependem de sistemas manuais, como agendas de papel ou planilhas, para controlar os horários dos pacientes e médicos. Este método tradicional, embora comum, é propenso a erros operacionais, como agendamentos duplicados, rasuras, dificuldade em localizar rapidamente horários vagos ou o histórico de um paciente. Isso configura um problema específico de falta de eficiência na gestão da agenda.

A motivação para o desenvolvimento desta aplicação é, portanto, otimizar essa rotina administrativa, reduzindo os erros humanos e o tempo gasto na marcação de consultas. A aplicação permitirá o cadastro centralizado de pacientes e médicos, o agendamento, a consulta e o cancelamento de horários de forma digital.

Desta forma, o projeto busca demonstrar a relevância de uma solução de software para solucionar um problema do mundo real, visando facilitar a rotina dos profissionais de recepção e saúde e otimizar o fluxo de atendimento ao paciente.

2 **OBJETIVOS DA APLICAÇÃO**

Os objetivos principais são:

- Gerenciar o cadastro de Pacientes, permitindo incluir, consultar e excluir seus dados pessoais (como nome, CPF e telefone).
- Gerenciar o cadastro de Médicos, registrando suas informações (como nome, CRM e especialidade).
- Permitir o agendamento de novas consultas, associando um paciente a um médico em uma data e hora específicas.
- Permitir o cancelamento de consultas previamente agendadas.
- Exibir a agenda de consultas de forma clara.

3 FUNÇÕES/LISTA DE EVENTOS (FUNCIONALIDADES) – RF / RNF

3.1. Requisitos Funcionais (RF)

RF1: O sistema deverá permitir o cadastro de Pacientes, com informações essenciais como nome completo, CPF, data de nascimento e telefone.

RF2: O sistema deverá permitir o cadastro de Médicos, com informações como nome completo, CPF, CRM e especialidade.

RF3: O sistema deverá permitir o agendamento de consultas, associando um Paciente, um Médico e uma data/hora.

RF4: O sistema deverá permitir o cancelamento de uma consulta agendada.

RF5: O sistema deverá exibir a agenda de consultas.

RF7: O sistema deverá permitir a consulta e exclusão dos dados cadastrais de Pacientes e Médicos.

3.2. Requisitos Não Funcionais (RNF)

RNF1: A aplicação deverá ser responsiva, adaptando-se para uso tanto em dispositivos móveis (tablets, smartphones) quanto em desktops.

RNF2: A aplicação deve ter um tempo de resposta inferior a 2 segundos para ações críticas, como a confirmação de um agendamento ou a consulta da agenda.

RNF3: A interface do usuário (GUI) deve ser intuitiva e limpa, exigindo um mínimo de treinamento para que um(a) recepcionista possa operá-la.

RNF4: O sistema deverá ser desenvolvido utilizando a linguagem de programação Java, conforme requisito da disciplina.

4 ESPECIFICAÇÃO DE PROGRAMAS

4.1. Layout da Tela

Tela Principal (Dashboard): Contem o menu principal.

Tela de Gerenciamento de Pacientes: Uma tela com formulário para cadastro de novos pacientes (campos para nome, CPF, telefone, etc.) e uma tabela que lista os pacientes existentes, com botão para excluir.

Tela de Gerenciamento de Médicos: Similar à de pacientes, com formulário para cadastro de médicos (campos para nome, CRM, especialidade) e uma tabela para listagem e exclusão.

Tela de Agendamento (Agenda): A tela principal da operação. Deve exibir um calendário ou uma visualização diária que permita à recepcionista a visualização da agenda.

4.2. Regras de Negócio

RN-01 (Cadastro de Pacientes): O sistema deve permitir o registro e a permanência de pacientes informando nome, CPF e telefone, gerando um identificador único (ID) automaticamente.

RN-02 (Cadastro de Médicos): O sistema deve permitir o registro e a permanência de médicos informando nome, CRM e especialidade, gerando um identificador único (ID) automaticamente.

RN-03 (Vínculo de Consulta): O agendamento deve vincular obrigatoriamente um Paciente existente e um Médico existente a uma data e hora selecionada pelo usuário.

RN-04 (Status Automático): Toda nova consulta cadastrada deve receber automaticamente o status inicial "AGENDADA".

RN-05 (Gestão de Exclusão): O sistema deve permitir a exclusão de registros (cancelamento de consultas ou remoção de cadastros) diretamente através das listas de visualização.

4.3. Entidades Envolvidas (Classes)

Classe Usuario: Representa um usuário que pode acessar o sistema (ex: Recepcionista).

Classe Paciente: Representa um paciente da clínica.

Responsabilidade: Armazenar dados cadastrais do paciente.

Atributos: nome (String), cpf (String), dataNascimento (Date), telefone (String).

Classe Medico: Representa um médico que atende na clínica.

Responsabilidade: Armazenar dados cadastrais do médico e sua especialidade.

Atributos: nome (String), cpf (String), crm (String), especialidade (String).

Classe Consulta: Representa o agendamento de um atendimento.

Responsabilidade: Associar um Paciente a um Medico em uma data e hora específicas.

Atributos: medico (Medico), paciente (Paciente), dataHora (DateTime), status (String, ex:"Agendada", "Cancelada", "Realizada").

4.4. Tabelas

Tabela: TB_PACIENTE

```

id_paciente (INT, PRIMARY KEY, AUTO_INCREMENT)
nome (VARCHAR(200), NOT NULL)
cpf (VARCHAR(11), NOT NULL, UNIQUE)
data_nascimento (DATE)
telefone (VARCHAR(15))

```

Tabela: TB_MEDICO

```

id_medico (INT, PRIMARY KEY, AUTO_INCREMENT)
nome (VARCHAR(200), NOT NULL)
crm (VARCHAR(20), NOT NULL, UNIQUE)
especialidade (VARCHAR(100), NOT NULL)

```

Tabela: TB_CONSULTA

```

id_consulta (INT, PRIMARY KEY, AUTO_INCREMENT)
data_hora (DATETIME, NOT NULL)
status (VARCHAR(20), NOT NULL)

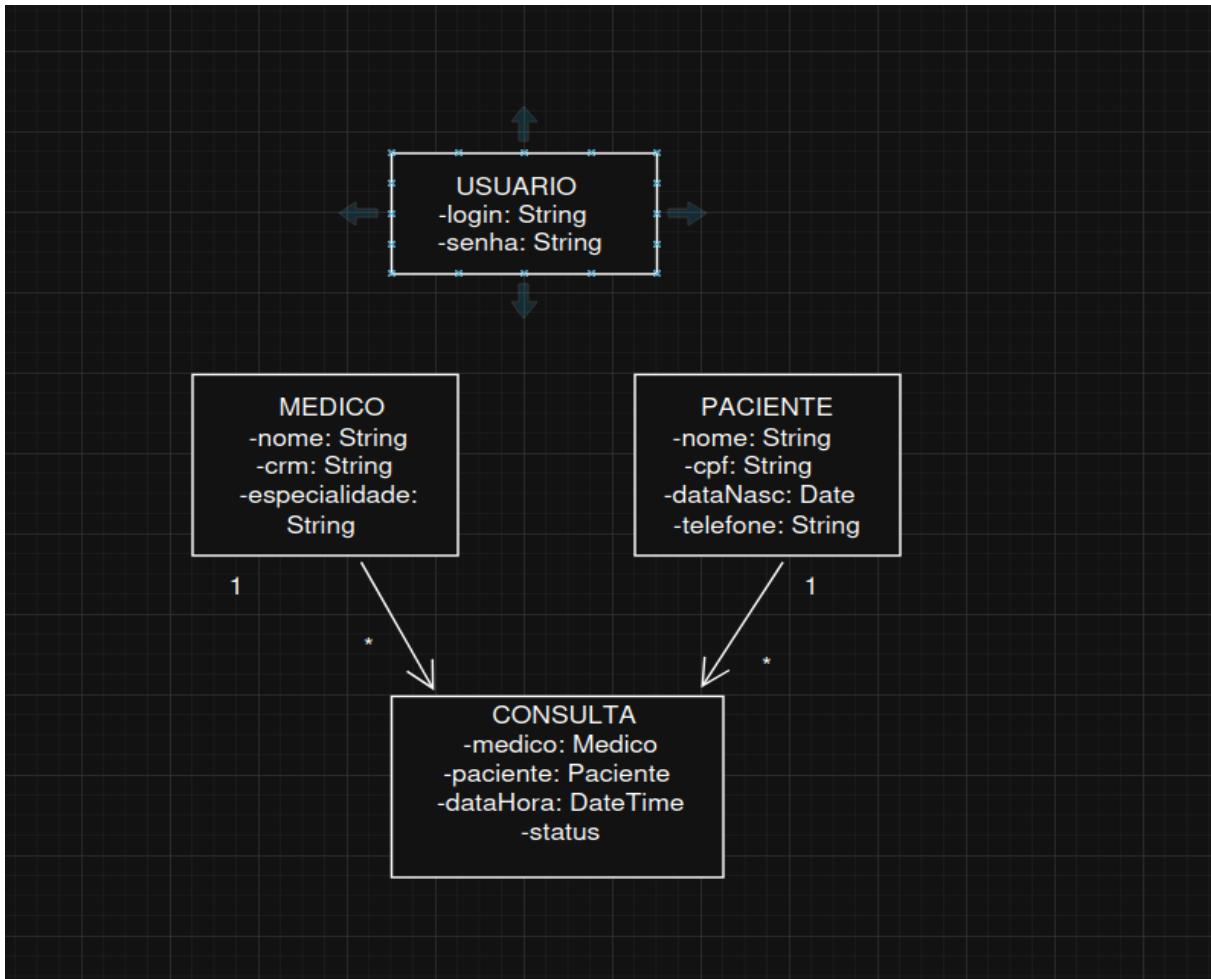
```

Id_paciente (INT, NOT NULL, FOREIGN KEY REFERENCES
TB_PACIENTE(id_paciente))

id_medico (INT, NOT NULL, FOREIGN KEY REFERENCES
TB_MEDICO(id_medico))

5 DIAGRAMA DE CLASSE

5.1. Artefato Gráfico



5.2. Dicionário de Dados

Tabela: TB_PACIENTE | Campo | Tipo | Descrição || :--- | :--- | :--- || id_paciente | INT (PK) | Identificador numérico único para o paciente. || nome | VARCHAR(200) | Nome completo do paciente. || cpf | VARCHAR(11) | CPF do paciente (único, 11 dígitos). || data_nascimento | DATE | Data de nascimento do paciente (formato AAAA-MM-DD). || telefone | VARCHAR(15) | Telefone de contato do paciente (com DDD). |

Tabela: TB_MEDICO | Campo | Tipo | Descrição || :--- | :--- | :--- || id_medico | INT (PK) | Identificador numérico único para o médico. || nome | VARCHAR(200) | Nome completo do médico. || cpf | VARCHAR(11) | CPF do médico (único, 11 dígitos). || crm | VARCHAR(20) | Número do registro do médico no Conselho Regional de Medicina (único).

|| especialidade | VARCHAR(100) | Especialidade principal do médico (ex: Cardiologia, Pediatria). |

Tabela: TB_CONSULTA | Campo | Tipo | Descrição || :--- | :--- | :--- | id_consulta | INT (PK) | Identificador numérico único para a consulta. || data_hora | DATETIME | Data e hora exatas do agendamento (AAAA-MM-DD HH:MM). || status | VARCHAR(20) | Situação atual da consulta (ex: "Agendada", "Cancelada", "Realizada"). || id_paciente | INT (FK) | Chave estrangeira que referencia o id_paciente na TB_PACIENTE. || id_medico | INT (FK) | Chave estrangeira que referencia o id_medico na TB_MEDICO.

6 APLICAÇÃO WEB/MOBILE

6.1. Menu/Submenu

O sistema possui um menu principal centralizado na tela inicial, que direciona o usuário para as operações de cadastro e consulta. O menu está estruturado da seguinte forma:

Pacientes:

- Cadastrar Paciente: Acesso ao formulário de registro de novos pacientes.
- Listar Pacientes: Visualização da tabela de pacientes cadastrados, com opção de exclusão.

Médicos:

- Cadastrar Médico: Acesso ao formulário de registro de novos profissionais.
- Listar Médicos: Visualização da tabela de médicos cadastrados, com opção de exclusão.

Consultas (Agendamentos):

- Agendar Consulta: Formulário para registrar um novo atendimento, vinculando um Paciente a um Médico.

Ver Agenda: Relatório geral que lista todas as consultas agendadas no sistema.

6.2. Telas Funcionais

Tela Inicial (Menu Principal): A porta de entrada da aplicação, contendo os botões de navegação para todas as funcionalidades do sistema.

Telas de Cadastro (Formulários): Cadastro de Paciente: Formulário contendo campos de texto para inserção de Nome, CPF, Telefone e Data de Nascimento.

Cadastro de Médico: Formulário para registro de Nome, CRM, Especialidade e Telefone.

Agendamento de Consulta: Formulário dinâmico que carrega automaticamente a lista de Médicos e Pacientes cadastrados, permitindo a seleção e a definição da Data/Hora.

Lista de Pacientes: Tabela que exibe todos os pacientes registrados, com um botão de "Excluir" para cada registro.

Lista de Médicos: Tabela que exibe os profissionais cadastrados, também com opção de exclusão.

Agenda de Consultas: Tabela principal que mostra os agendamentos, exibindo a Data/Hora, o nome do Paciente, o nome do Médico e o Status, com opção para cancelar (excluir) o agendamento.

6.3. Telas de Diálogo

Mensagem de Confirmação :

Ao clicar no botão "Excluir" ou "Cancelar", o sistema exibe um alerta (pop-up) no navegador perguntando: "Tem certeza?" ou "Cancelar consulta?". A exclusão só ocorre se o usuário confirmar.

Alerta de Campos Obrigatórios:

Caso o usuário tente salvar um formulário com campos vazios, o próprio navegador exibe balões de alerta indicando: "Preencha este campo", impedindo o envio de dados incompletos.

Redirecionamento Automático:

Após salvar um registro com sucesso, o sistema redireciona o usuário automaticamente para o Menu Principal, confirmando a operação.

7 CONSIDERAÇÕES FINAIS

O desenvolvimento da aplicação de atendimento médico foi proposto para solucionar o problema de gerenciamento manual de agendas em clínicas, um processo propenso a falhas e ineficiência. Através da aplicação dos conceitos da Programação Orientada a Objetos (POO) em Java, foi possível modelar e construir um sistema funcional. As entidades principais, como Paciente, Medico e Consulta, foram implementadas como classes, permitindo um código organizado, coeso e de fácil manutenção.

O sistema desenvolvido atinge todos os objetivos principais propostos neste documento. A aplicação é capaz de gerenciar os cadastros de pacientes e médicos, controlar o agendamento de consultas e permitir cancelamentos, cumprindo os requisitos funcionais essenciais.

O maior desafio encontrado foi a necessidade de integrar múltiplos conceitos (design de interface, lógica de negócios e persistência de dados) em uma única aplicação. Como trabalho futuro, o sistema poderia ser expandido para incluir prontuários eletrônicos, controle financeiro (pagamentos) e integração com envio de lembretes de consulta por WhatsApp ou SMS.

Conclui-se que o projeto foi bem-sucedido, demonstrando a aplicação prática da POO Java na criação de uma solução de software relevante para um problema do mundo real e consolidando o aprendizado obtido na disciplina.

REFERÊNCIAS

Livros:

DEITEL, Paul; DEITEL, Harvey. **Java: Como Programar.** 10. ed. São Paulo: Pearson Prentice Hall, 2016.

GUEDES, Gilleanes T. A. **UML 2: Guia Prático.** 8. ed. São Paulo: Novatec Editora, 2020.

KOSCIELNY, Greg L.; BOUSETTA, Badr. **Spring Boot: Up and Running:** Build cloud-native Java and Kotlin applications. 1. ed. Sebastopol, CA: O'Reilly Media, 2021.

Artigos e Documentação Online:

BAELDUNG. What Is Spring Boot?. **Baeldung**, 2024. Disponível em:

<https://www.baeldung.com/spring-boot-introduction>. Acesso em: 02 nov. 2025.

ORACLE. **Java SE 17 Documentation.** Oracle Help Center, 2025. Disponível em:

<https://docs.oracle.com/en/java/javase/17/>. Acesso em: 05 nov. 2025.

STACK OVERFLOW. **Como conectar Java ao banco de dados MySQL (JDBC)?** Stack Overflow, 2013. Disponível em: <https://pt.stackoverflow.com/questions/18318/como-conectar-java-ao-banco-de-dados-mysql-jdbc>. Acesso em: 04 nov. 2025.

Ferramentas (Software):

DIAGRAMS.NET. **draw.io.** Versão 24.4.0. 2024. Disponível em:

<https://app.diagrams.net/>. Acesso em: 05 nov. 2025.

Vídeos e Tutoriais:

CÓDIGO FONTE TV. **Curso de PROGRAMAÇÃO JAVA para INICIANTES |**

Fundamentos + Programação Orientada Objetos. YouTube, 2024. Disponível em:

<https://www.youtube.com/watch?v=nODe5lFcGpg>. Acesso em: 10 nov. 2025.

ATTECNOLOGIA. [ADI #1] - **Criando um CRUD com Java Spring Boot e MySQL.**

YouTube, 2023. Disponível em: <https://www.youtube.com/watch?v=Tnl4YnB6E54>. Acesso em: 10 nov. 2025.

DEV EFICIENTE. **Como fazer uma API Rest (CRUD) com Spring Boot.** YouTube, 2022.

Disponível em: <https://www.youtube.com/watch?v=SVfkKa0xUbU>. Acesso em: 01 nov. 2025.