

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, alterar 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, verificando a disponibilidade da agenda.
- Permitir o cancelamento de consultas previamente agendadas.
- Exibir a agenda de consultas de forma clara, permitindo a visualização por dia ou por profissional.

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 impedir o agendamento de mais de uma consulta no mesmo horário para o mesmo Médico (controle de disponibilidade).

RF6: O sistema deverá exibir a agenda de consultas por dia.

RF7: O sistema deverá permitir a consulta, alteração 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 de Login: Tela inicial que solicita um usuário e senha para acessar o sistema. Garante que apenas pessoal autorizado (ex: recepcionista, administrador) possa gerenciar os dados.

Tela Principal (Dashboard): Tela exibida após o login, contendo 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 botões para editar ou 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, edição e exclusão.

Tela de Agendamento (Agenda): A tela principal da operação. Deve exibir um calendário ou uma visualização diária que permita à recepcionista selecionar um médico, ver seus horários vagos e agendar um paciente em um horário específico.

Tela de Relatório de Consultas: Uma tela que permite filtrar consultas por período ou por médico e exibe uma lista dos agendamentos.

4.2. Regras de Negócio

RN-01: O CPF de um Paciente deve ser único no sistema. O sistema não deve permitir o cadastro de dois pacientes com o mesmo CPF.

RN-02: O CRM de um Médico deve ser único. O sistema não deve permitir o cadastro de dois médicos com o mesmo CRM.

RN-03: Um Médico não pode ter duas consultas agendadas no mesmo dia e horário.

RN-04: Não é permitido agendar consultas em datas ou horários que já passaram.

RN-05: Para cancelar uma consulta, é necessário que ela exista.

4.3. Entidades Envolvidas (Classes)

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

Responsabilidade: Autenticar no sistema.

Atributos: login (String), senha (String).

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_USUARIO

id_usuario (INT, PRIMARY KEY, AUTO_INCREMENT)

login (VARCHAR(50), NOT NULL, UNIQUE)

senha (VARCHAR(255), NOT NULL)

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)

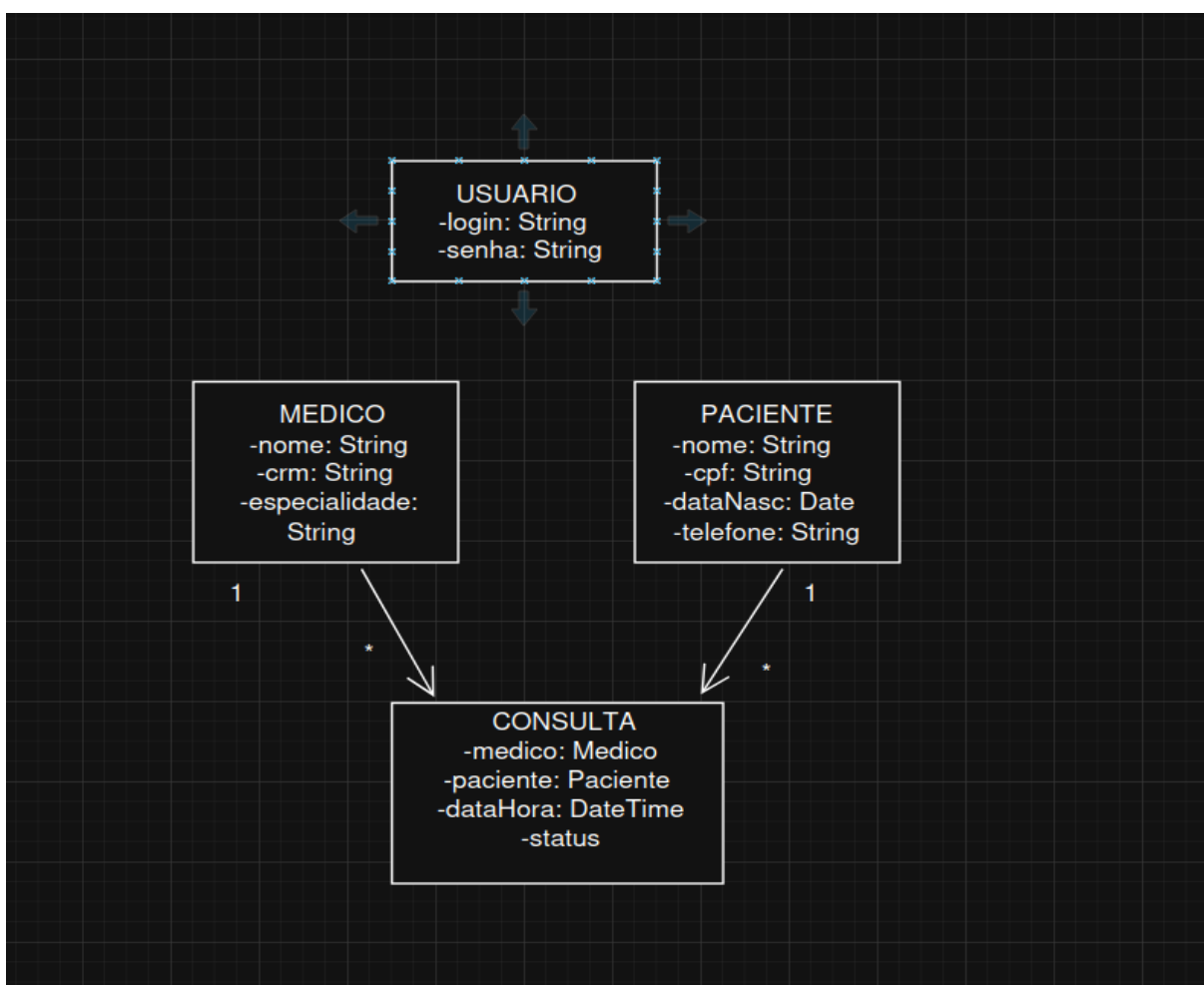
status (VARCHAR(20), NOT NULL)

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_USUARIO | Campo | Tipo | Descrição | | :--- | :--- | :--- | | id_usuario | INT (PK) | Identificador numérico único para o usuário. | | login | VARCHAR(50) | Nome de usuário (único) usado para acessar o sistema. | | senha | VARCHAR(255) | Senha criptografada para acesso ao sistema. |

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 fixo que direciona o usuário para as principais funcionalidades da aplicação. O menu está estruturado da seguinte forma:

Início: Tela principal.

Agendamentos:

Agendar Nova Consulta

Visualizar Agenda Completa

Cadastros:

Gerenciar Pacientes

Gerenciar Médicos

Relatórios:

Consultas por Período

Sistema:

Gerenciar Usuários

Sair (Logout)

6.2. Telas Funcionais

Tela de Login: Interface inicial que solicita usuário e senha para garantir o acesso seguro.

Tela de Agenda : A tela principal da aplicação. Apresenta uma visualização em colunas (por médico) dos horários do dia.

Tela de Cadastro de Paciente: Contém um formulário para inserir ou editar dados de pacientes (nome, CPF, telefone, etc.) e uma tabela que lista todos os pacientes cadastrados, com botões de "Editar" e "Excluir".

Tela de Cadastro de Médico: Similar à de pacientes, com formulário e tabela para gerenciar os médicos (nome, CRM, especialidade).

6.3. Telas de Diálogo

Mensagem de Confirmação:

"Deseja realmente cancelar a consulta do paciente [Nome] no dia [Data]?"

"Paciente [Nome] cadastrado com sucesso!"

Mensagem de Erro (Validação):

"Erro: O CPF informado já existe no cadastro."

"Erro: Medico inexistente"

Mensagem de Alerta:

"Atenção: Todos os campos obrigatórios (*) devem ser preenchidos."

"O usuário ou senha informados estão incorretos."

6.4. Layout Relatórios

Layout do Relatório de Consultas: O principal relatório do sistema permite ao usuário definir um período (ex: 01/11/2025 a 30/11/2025), o sistema então gera uma tabela simples contendo:

Data e Hora da Consulta

Nome do Paciente

Nome do Médico

Status (Realizada, Cancelada)

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 (validando a disponibilidade de horário) 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 (Exemplos):

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.