

NEXUMTECH

Turma 1TDSPY
Felipe Ribeiro Salles de Camargo - 565224

Pamella Christiny Chaves Brito - 565206



São Paulo

2025

Sumário

Este é o Sumário (Table of Contents) revisado, com a hierarquia e numeração lógica ajustadas para um documento profissional, baseado no conteúdo do arquivo "NexumTech - Java.docx".

Você pode copiar e colar esta estrutura para o seu relatório.

SUMÁRIO

Eu entendo perfeitamente! Você está focado no conteúdo e na apresentação final do seu relatório.

Abaixo está o Sumário (Table of Contents) completo, organizado e profissional, **sem nenhuma citação**, pronto para você copiar e colar diretamente no seu documento.

SUMÁRIO

1. IDENTIFICAÇÃO DO PROJETO E EQUIPE
 - 1.1. NEXUMTECH
 - 1.2. Turma 1TDSPY
 - 1.3. Local e Ano (São Paulo, 2025)
2. OBJETIVO E ESCOPO DO PROJETO
3. DESCRIÇÃO DAS FUNÇÕES
 - 3.1. Processo de notificação fixa e pontual
 - 3.2. Processo de notificação pré-consulta
 - 3.3. Processo de notificação de consulta atrasada
 - 3.4. Endpoints para realizar as requisições (API REST)
4. FERRAMENTAS UTILIZADAS
 - 4.1. Linguagens e Plataformas
 - 4.2. Bibliotecas e Frameworks Principais
 - 4.3. Ferramenta de Desenvolvimento (Inferida)
5. DIAGRAMAS
 - 5.1. DER (Diagrama de Entidade-Relacionamento)
 - 5.2. UML (Diagrama de Classes)
6. GUIA DE EXECUÇÃO DO PROJETO ÁGATA
 - 6.1. Pré-Requisitos do Ambiente
 - 6.2. Scripts SQL para Criação do Schema
 - 6.3. Passo a Passo para Execução
 - 6.4. Observações Importantes

Objetivo e escopo do projeto:

O Ágata é uma solução inovadora que visa transformar o acesso à saúde para pacientes com desafios motores, cognitivos ou tecnológicos. Em vez de se adaptarem a sistemas complexos de agendamento e teleconsulta, eles agora têm uma ferramenta desenvolvida pensando em suas necessidades específicas.

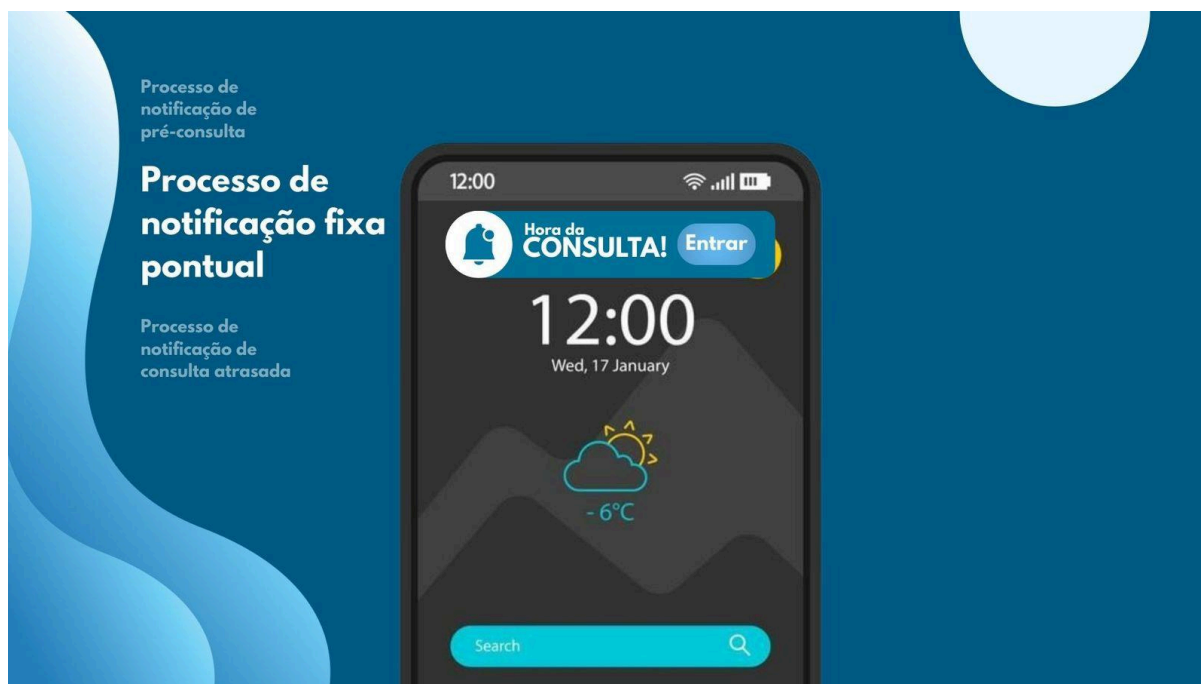
Nosso principal objetivo é conectar esses pacientes com os profissionais de saúde ideais de forma simples e direta. Utilizando o aplicativo do HC, os pacientes receberão notificações intuitivas e fixas que servem como atalhos diretos para as teleconsultas, eliminando barreiras e simplificando o processo do início ao fim. Com o Ágata, a saúde digital se torna acessível a todos.

Descrição das funções

Temos três opções de atuar com o ágata:

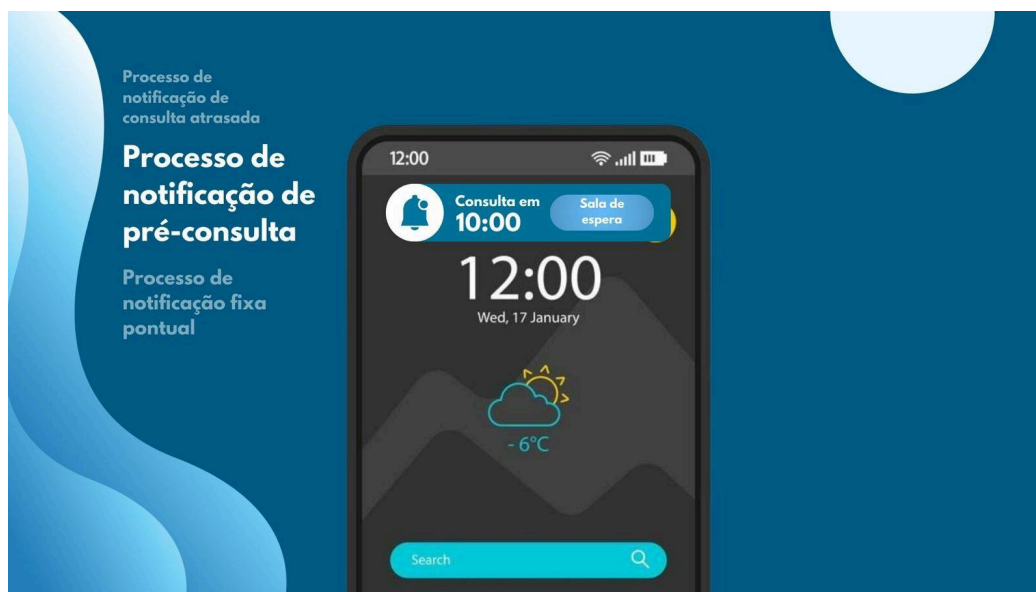
1 - Processo de notificação fixa e pontual

Para interação é necessário que clique no botão **ENTRAR**, que levará diretamente para a sala virtual.



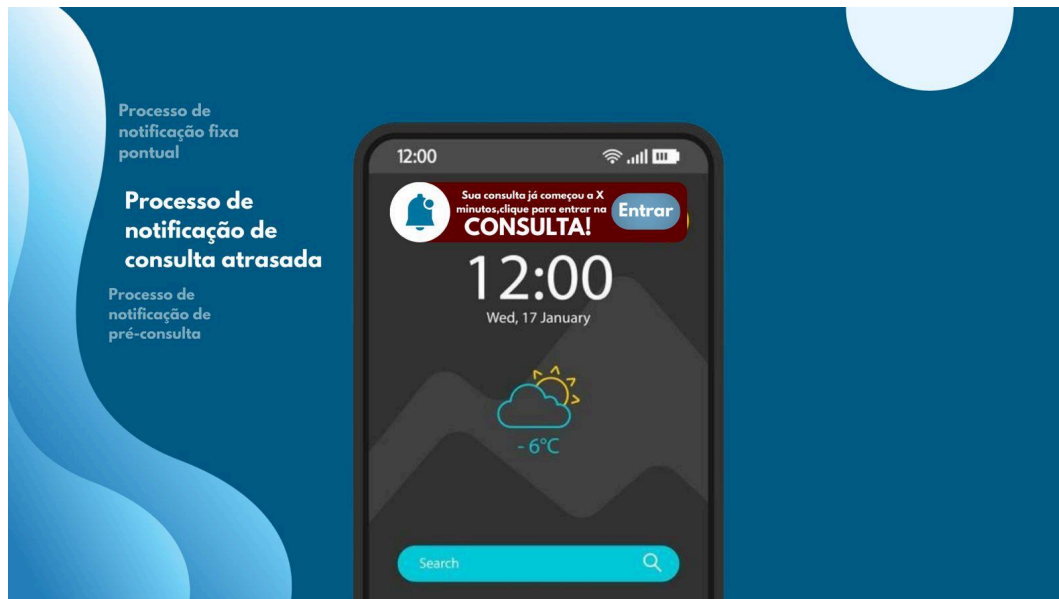
2 - Processo de notificação pré-consulta

Para interação é necessário que clique no botão **ENTRAR**, que levará diretamente para a sala de espera virtual.



3 - Processo de notificação de consulta atrasada

Para interação é necessário que clique no botão **ENTRAR**, que levará diretamente para a sala virtual.



Ferramentas utilizadas

Linguagens e Plataformas

Java (Linguagem de Programação): É a linguagem principal de todo o projeto.

Oracle Database / SQL (Banco de Dados): Usado para persistir os dados (pacientes, médicos, consultas), conforme evidenciado pelo seu script DDL (CREATE TABLE, CREATE SEQUENCE) e pela natureza das classes DAO (Data Access Object).

Bibliotecas e Frameworks Principais

JDBC (Java Database Connectivity): Embora o código exato da conexão (dentro dos DAOs) não esteja visível, as classes DAO (PacienteDAO, MedicoDAO, ConsultaDAO) são a interface padrão para usar o JDBC para interagir com o Oracle Database.

Java Swing: Utilizado na função iniciarConsulta() para criar uma interface gráfica básica (janela, botões, labels e um cronômetro) para a simulação de teleconsulta. Isso é indicado pelos imports como javax.swing.JFrame, javax.swing.JButton, etc.

java.util.Scanner: Usada na classe Main para lidar com a entrada de dados pela linha de comando (CLI - Command Line Interface), permitindo que o usuário navegue pelos menus.

java.time (API de Data e Hora): Usada para manipulação de data e hora (LocalDateTime, DateTimeFormatter).

java.sql.Timestamp: Usada especificamente para interagir com os tipos de dados de tempo do SQL no banco de dados.

Ferramenta de Desenvolvimento (Inferida)

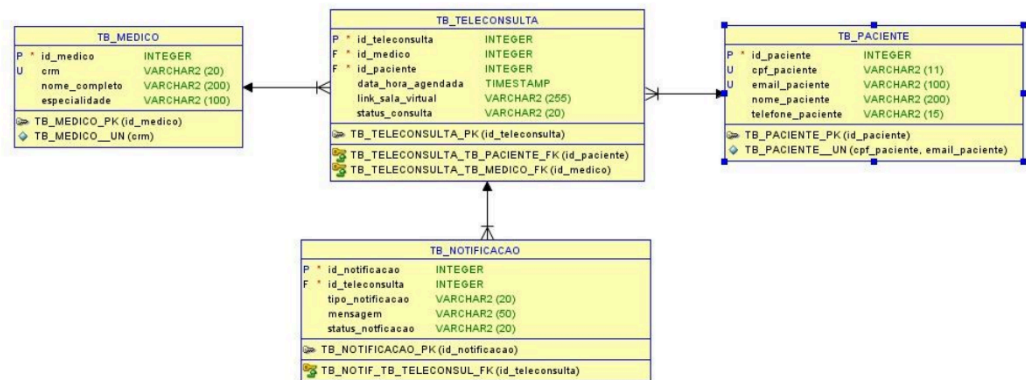
IDE (Ambiente de Desenvolvimento Integrado): Pelas capturas de tela, você está utilizando o IntelliJ IDEA (ou um IDE similar como Eclipse/VS Code com plugins Java), que fornece o ambiente para escrever, compilar e executar o código.

Endpoints para realizar as requisições

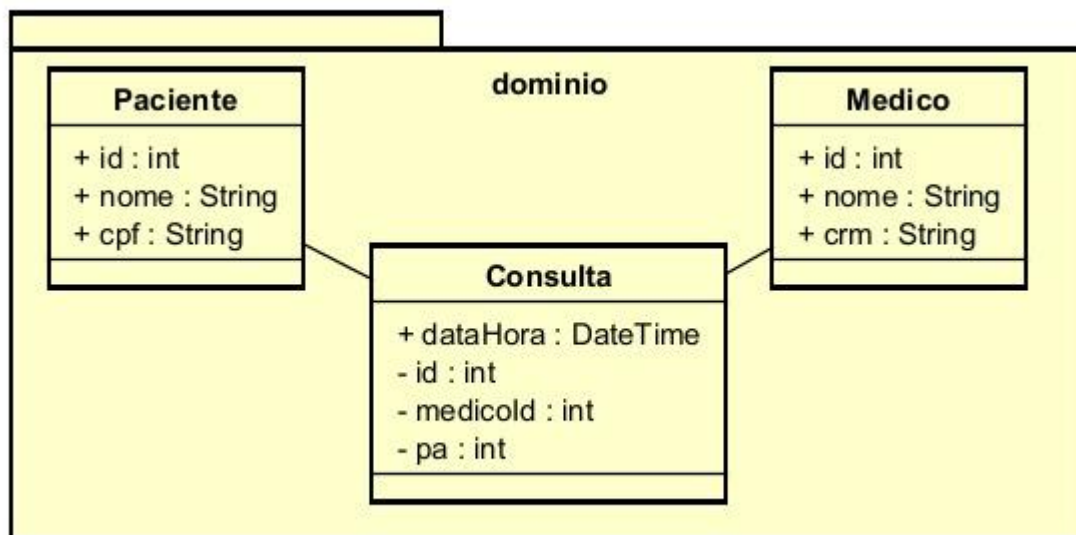
Recurso	Método HTTP	URI Completa	Descrição	Status de Sucesso	Status de Erro/Regra de Negócio
Consulta	POST	/consultas	Agenda uma nova consulta (verifica conflitos de horário).	201 Created	409 Conflict, 500 Internal Server Error
Consulta	GET	/consultas	Lista todas as consultas agendadas.	200 OK	404 Not Found
Consulta	DELETE	/consultas/{id}	Cancela/deleta uma consulta pelo ID.	204 No Content	404 Not Found
Paciente	POST	/pacientes	Adiciona um novo paciente (Verificar unicidade de CPF).	201 Created	409 Conflict
Paciente	GET	/pacientes/{id}	Busca paciente por ID.	200 OK	404 Not Found

Pacient e	PUT	/pacientes/{id}	Atualiza todos os dados de um paciente pelo ID.	200 OK	404 Not Found
Pacient e	DELETE	/pacientes/{id}	Remove um paciente pelo ID.	204 No Content	404 Not Found
Médico	POST	/medicos	Adiciona um novo médico (Verificar unicidade de CRM).	201 Created	409 Conflict
Médico	GET	/medicos	Lista todos os médicos.	200 OK	404 Not Found
Médico	GET	/medicos/{crm}	Busca médico pelo CRM.	200 OK	404 Not Found
Médico	PUT	/medicos/{crm}	Atualiza todos os dados de um médico pelo CRM.	200 OK	404 Not Found
Médico	DELETE	/medicos/{crm}	Remove um médico pelo CRM.	204 No Content	404 Not Found

MER



UML



Guia de Execução do Projeto Agata (Gerenciador de Teleconsultas)

Este guia detalha os pré-requisitos e o passo a passo para configurar e rodar o projeto utilizando Java, Oracle Database e a IDE IntelliJ IDEA.

1. Pré-Requisitos do Ambiente

Para garantir a execução correta, os seguintes itens devem estar prontos em seu sistema:

1. Java Development Kit (JDK): O JDK 22 deve estar instalado e seu caminho (PATH) configurado no sistema operacional.

2. Oracle Database: Uma instância do Oracle Database deve estar rodando e acessível.
 - Um usuário/schema deve ser criado e populado com as tabelas e sequências necessárias (vide scripts SQL abaixo).
3. Driver JDBC: O arquivo JAR do driver JDBC do Oracle (ojdbc*.jar) deve ser obtido e adicionado às bibliotecas do projeto na IDE.
4. Configuração de Conexão: A classe banco/ConnectionFactory.java deve ser configurada com os dados de acesso ao seu banco de dados (URL, usuário e senha).

2. Scripts SQL para Criação do Schema

Execute os seguintes comandos no seu schema do Oracle para criar a estrutura do banco de dados:

SQL

-- SEQUÊNCIAS

```
CREATE SEQUENCE SEQ_PACIENTE START WITH 1 INCREMENT BY 1 NOCACHE  
NOCYCLE;
```

```
CREATE SEQUENCE SEQ_CONSULTA START WITH 1 INCREMENT BY 1 NOCACHE  
NOCYCLE;
```

-- TABELAS

```
CREATE TABLE PACIENTE (  
    ID NUMBER PRIMARY KEY,  
    NOME VARCHAR2(100) NOT NULL,  
    CPF VARCHAR2(14) UNIQUE NOT NULL  
);
```

```
CREATE TABLE MEDICO (  
    CRM VARCHAR2(20) PRIMARY KEY,  
    NOME_MEDICO VARCHAR2(100) NOT NULL,  
    ESPECIALIDADE_MEDICO VARCHAR2(50) NOT NULL  
);
```

```
CREATE TABLE CONSULTA (  
    ID NUMBER PRIMARY KEY,  
    ID_PACIENTE NUMBER NOT NULL,  
    CRM_MEDICO VARCHAR2(20) NOT NULL,  
    DATA_HORA_CONSULTA TIMESTAMP NOT NULL,  
    STATUS VARCHAR2(20) NOT NULL,  
    DURACAO NUMBER NOT NULL,  
    CONSTRAINT FK_CONSULTA_PACIENTE FOREIGN KEY (ID_PACIENTE)  
REFERENCES PACIENTE(ID),  
    CONSTRAINT FK_CONSULTA_MEDICO FOREIGN KEY (CRM_MEDICO)  
REFERENCES MEDICO(CRM)  
);
```

3. Passo a Passo para Execução

Siga estes passos dentro da IDE IntelliJ IDEA para rodar o projeto:

Passo 1: Importar e Configurar o Projeto

1. No IntelliJ IDEA, vá em File > Open e selecione a pasta raiz do projeto (Agata).
2. Confirme se a pasta src foi automaticamente marcada como Source Root.
3. Adicione o driver JDBC do Oracle ao projeto (geralmente em File > Project Structure > Libraries).

Passo 2: Configurar a Conexão com o Banco

1. Abra o arquivo banco/ConnectionFactory.java.
2. Altere os valores das seguintes variáveis para os dados de acesso ao seu Oracle Database:

Java

```
String urlDeConexao = "jdbc:oracle:thin:@oracle.fiap.com.br:1521:ORCL"; // (Altere se necessário)
```

```
String login = "rm565206"; // (Seu usuário)
```

```
String senha = "200806"; // (Sua senha)
```

Passo 3: Executar a Aplicação

1. Abra a classe principal app.Main.java.
2. Clique no botão "Run" (o ícone de seta verde) para iniciar a aplicação.
3. A interação com o sistema será feita pelo console (terminal) da sua IDE.

Observações Importantes

- Regras de Negócio: Fique atento à classe service.ConsultaService.java para entender a lógica de negócio do sistema.
- Acesso a Dados: Todos os DAOs utilizam a ConnectionFactory para gerenciar a conexão com o Oracle de forma automática.
- Interação: O projeto é menu-driven, então siga as opções numéricas apresentadas no console.