

CENTRO UNIVERSITÁRIO UNINTER  
TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

PROJETO: DESENVOLVIMENTO BACK-END

AMANDA MAJCZAK THEODORO DA SILVA SPADER  
RU.: 2481377

PAP ABERLADO LUZ  
Prof. Winston Sen Lun Fung, Me.  
2025.

## Sumário

1. INTRODUÇÃO .....	3
1.1. Contexto do Estudo de Caso .....	3
1.2. Objetivos do Projeto .....	4
1.3. Principais Usuários .....	4
2. ANÁLISE E REQUISITOS .....	5
3. MODELAGEM E ARQUITETURA .....	8
3.1. Arquitetura do Sistema .....	8
3.2. Diagrama de Classes .....	9
3.3. Diagrama Entidade-Relacionamento (DER) .....	10
3.4. Endpoints da API .....	10
3.5. Tecnologias de Persistência .....	12
4. IMPLEMENTAÇÃO .....	12
4.1. Tecnologias Utilizadas .....	12
4.2. Estrutura do Sistema .....	13
4.3. Telas da Prototipação .....	16
4.4. Trechos de Códigos .....	17
4.5. Exemplo de Fluxo .....	20
5. PLANO DE TESTES .....	21
5.1. POST /administrador .....	22
5.2. GET /administrador .....	22
5.3. GET /paciente/:idAdm .....	23
6. CONCLUSÃO .....	23
REFERÊNCIAS .....	24
ANEXO A – CASO DE USO .....	25
ANEXO B – FICHA DE ANAMNESE .....	26
ANEXO C – PRONTUÁRIO DE TRIAGEM .....	27
ANEXO D – MODELO DE CARTEIRA DE ENFERMEIRO .....	28
ANEXO E – MODELO CARTEIRA DE FARMACÊUTICO .....	29
ANEXO F – MODELO CARTEIRA DE MÉDICO .....	30

# 1. INTRODUÇÃO

Nos últimos anos, o crescimento da indústria de Saúde e Bem-estar tem aumentado cada vez mais, contendo grande influência das mídias sociais. Com isso, as empresas procuram sempre melhorar seus serviços para conseguir se sobressair a suas concorrentes no mercado (Agência O Globo, 2023).

Além disso, o tema de gestão hospitalar tem sido analisado e trabalhado cada vez mais por conta dos avanços tecnológicos, conforme (Lima *et al.*, 2024):

Com o surgimento de novas tecnologias, modelos de gestão inovadores e uma maior conscientização sobre a importância da qualidade no atendimento, os gestores hospitalares estão buscando constantemente maneiras de melhorar a eficiência e eficácia dos serviços de saúde no país.

A gestão Hospitalar é um tema cada vez mais falado por conta da necessidade que vem sendo trabalhado e levado mais a sério cada vez mais, principalmente após a pandemia. Essa área trabalha no contexto da união entre saúde, administração e tecnologia, onde “processos e estratégias voltadas para administrar e organizar os recursos de uma instituição de saúde, como hospitais e clínicas.” (Riviere, 2024).

Para que a Gestão Hospitalar esteja dentro do seu propósito, o ideal é algo que traga a integração entre todas as áreas envolvidas e realize um fluxo eficiente, com todos os envolvidos com seus papéis desenvolvidos. A implementação desse sistema é de extrema relevância para garantir a eficiência operacional, a segurança no atendimento e a conformidade com normas sanitárias. A informatização dos processos contribui para a redução de erros, otimização de recursos e tomada de decisões mais assertivas por parte da gestão hospitalar.

## 1.1. Contexto do Estudo de Caso

O estudo de caso para esse projeto foi para a criação de um Sistema de Gestão Hospitalar e de Serviços de Saúde (SGHSS), onde a instituição fictícia VidaPlus necessita de um sistema que atenda aos requisitos de sua administração, que envolve hospitais, clínicas de bairro, laboratórios e equipes de home care. Isso trás a necessidade crescente de sistemas informatizados no ambiente hospitalar, visando a melhoria da gestão de pessoal, pacientes e suprimentos.

O Sistema de Gestão tem como proposta de:

- Gerenciar os seus pacientes, controlando suas consultas, exames e prontuários.

- Gerenciar seus profissionais de saúde, como médicos, enfermeiros e farmacêuticos.
- Permitir que os médicos realizem consultas e monitorem suas agendas, assim como enfermeiros controlem suas atividades.
- Gerenciamento da Administração Hospitalar como Leitos, Suprimentos e realização de relatórios.
- Possibilitação de realização de Telemedicinas.

Para que esse projeto fosse melhor executado, realizou-se pesquisas para saber como funciona o sistema hospitalar, bem como as funções de cada pessoa ali dentro e quais são os dados fundamentais para poder realizar as manipulações de dados de forma mais próxima a realidade.

## 1.2. Objetivos do Projeto

Os objetivos desse projeto, conforme o Estudo de Caso, é o desenvolvimento de um sistema que realize esse gerenciamento hospitalar, permitindo registros e monitoramentos de seus pacientes e seus profissionais de saúde, além da parte estrutural como Leito e Internações, e suprimentos em estoque. Com isso, haverá um único local em que se mantenha todo o gerenciamento em único local.

## 1.3. Principais Usuários

Os principais usuários foram definidos conforme o estudo de caso:

- Administrador: responsável pelos cuidados estruturais e gerenciamento de pessoal, como Médicos, Enfermeiros e Farmacêuticos cadastrados, além dos leitos e suprimentos em estoque, monitorando todas as áreas necessárias para um bom gerenciamento.
- Profissionais de Saúde: os profissionais de saúde estão envolvidos diretamente nesse gerenciamento, trabalhando conforme sua área:
  - Médicos: gerenciando sua agenda médica, realizando consultas e prescrições médicas, realizando internações e organizando seus pacientes da melhor forma possível.
  - Enfermeiros: Tendo um maior controle sobre o que é necessário realizar em cada paciente e visualizar as atividades para aquele dia, seguindo as orientações médicas passadas.



RF002	Administração: Cadastrar Profissionais de Saúde (Médico, Enfermeiros, Farmacêuticos)	Funcional	Alta
RF003	Administração: Cadastrar Leitos	Funcional	Alta
RF004	Administração: Cadastrar Suprimentos	Funcional	Alta
RF005	Administração: Visualizar Pacientes	Funcional	Alta
RF006	Administração: Visualizar Profissionais de Saúde cadastrados	Funcional	Alta
RF007	Administração: Atender solicitações de internação e escolher leito	Funcional	Alta
RF008	Médico: Agenda - Cadastrar datas e horas para consultas	Funcional	Alta
RF009	Médico: Agenda – Remover horários de consultas	Funcional	Alta
RF010	Médico: Agenda – Visualizar Agenda Completa	Funcional	Alta
RF011	Médico: Prontuário – Atualizar Prontuário do Paciente	Funcional	Alta
RF012	Médico: Consultas – Realizar consultas médicas	Funcional	Alta
RF013	Médico: Consultas – Emitir Receitas Digitais	Funcional	Média
RF014	Médico: Anamnese – Realizar Anamnese do Paciente	Funcional	Média
RF015	Médico: Anamnese – Alterar Anamnese do Paciente a cada consulta	Funcional	Média
RF016	Médico: Internação – Solicitar Internação do Paciente	Funcional	Alta
RF017	Médico: Internação – Criar Prescrições durante a internação do Paciente	Funcional	Alta

RF018	Médico: Internação – Realizar Alta	Funcional	Alta
RF019	Enfermeiro: Internação – Solicitar Suprimentos para realizar as Prescrições	Funcional	Alta
RF020	Enfermeiro: Internação – Realizar o procedimento solicitado	Funcional	Alta
RF021	Enfermeiro: Internação – Atualizar o Prontuário do Paciente após a realização do procedimento	Funcional	Alta
RF022	Farmacêutico: Suprimentos – Receber solicitação dos suprimentos	Funcional	Alta
RF023	Farmacêutico: Suprimento – Registrar entrega de Suprimentos ao Enfermeiro	Funcional	Alta
RF024	Farmacêutico: Suprimento – Dar baixa no estoque de Suprimentos	Funcional	Alta
RF025	Paciente: Marcar Consultas com um Médico específico	Funcional	Alta
RF026	Paciente: Visualizar próximas consultas marcadas	Funcional	Alta
RF027	Paciente: Receber Notificação de Consultas Próximas	Funcional	Alta
RF028	Paciente: Visualizar Prontuário Médico	Funcional	Alta
RF029	Teleconsulta: Realizar Videochamadas seguras	Funcional	Alta
RF030	Teleconsulta: Atualizar Prontuário do Paciente	Funcional	Alta
RF031	Teleconsulta: Realizar prescrições online	Funcional	Alta
RNF001	Controle de Acesso de Perfis	Não Funcional	Alta

RNF002	Criptografia de Dados Sensíveis	Não Funcional	Alta
RNF003	Registros de logs para auditoria	Não Funcional	Alta
RNF004	Suportar várias unidades hospitalares	Não Funcional	Alta
RNF005	Tempo de resposta rápido	Não Funcional	Médio
RNF006	Interface amigável e responsiva	Não Funcional	Médio

Fonte: Elaborado pelo autor com base no estudo de caso e diagrama de caso de uso

### 3. MODELAGEM E ARQUITETURA

A modelagem e arquitetura escolhida para esse projeto foi desenvolver um sistema em Java com Maven e com JDBC (Java Database Connectivity), utilizando juntamente o SparkleJava, que é um microframework web para construção de API Rest ou páginas web leves e o banco de dados MySQL, trazendo em sua junção um software de forma a separar responsabilidades, facilitar manutenção, reutilização de partes do código e escalabilidade.

#### 3.1. Arquitetura do Sistema

O sistema foi desenvolvido em 5 camadas, que são compostas por:

- **Camada de Apresentação:** titulada no sistema como interfaces, onde ficam as interfaces gráficas responsáveis por interagir com os usuários, feitos com o JFrame Swing.
- **Camada de Controle:** titulada no sistema como controles, eles controlam o fluxo de interação do sistema feito pelas interfaces com o banco de dados, recebendo as informações digitadas pelo usuário e devolve uma resposta.
- **Camada de Classes:** titulada no sistema como classes, ali contém toda a base de informações que serão recebidas e enviadas tanto pelas views como pelo banco de dados.



- ### 3.2. Diagrama de Classes

[illegible]

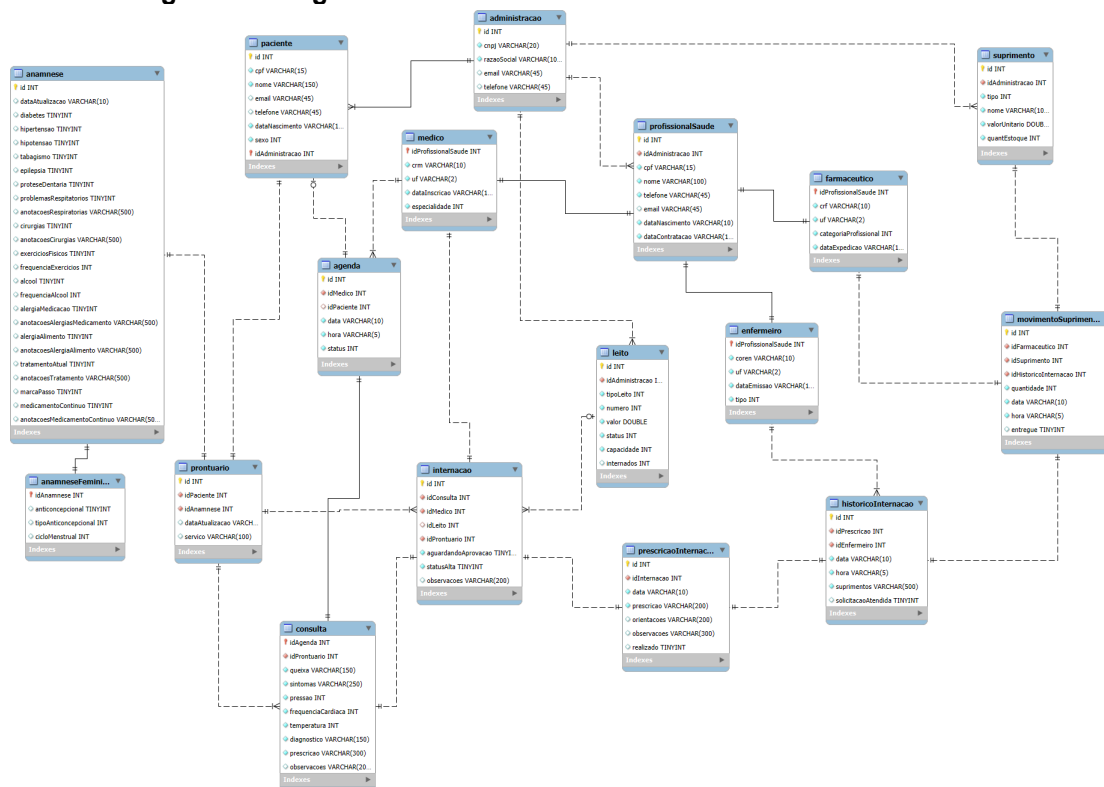
O Diagrama de Classes acima foi desenvolvido de acordo com pesquisas sobre o temas e ligados aos requisitos dos sistemas que foram retirados do estudo de caso. Esse tipo de diagrama é muito útil para estágio de design do sistema, e ajudar a entender melhor os requisitos e componentes para identificar problemas. No âmbito de desenvolvimento de sistemas, as classes se tornam classes e objetos reais durante os estágios iniciais do projeto, podendo ser modificado conforme o passar do tempo, dependendo da lógica, interfaces e testes (IBM, 2021).

9

agenda e as enfermeiras e farmacêuticas podendo monitorar e cuidar dos procedimentos e estoques necessários.

### 3.3. Diagrama Entidade-Relacionamento (DER)

Figura 3 – Diagrama DER do modelo relacional do Banco de Dados



Fonte: Próprio Autor

O Diagrama Entidade-Relacionamento trabalha como uma ferramenta de modelagem que é utilizada para planejar a estrutura de um Banco de Dados antes de sua implementação. Ele mostra quais são as informações que serão armazenadas em cada tabela e como elas interagem entre si, facilitando a visualização e o planejamento antes da implementação, ajudando a evitar erros antes da criação do banco de dados.

Nesse DER, vemos que algumas tabelas influenciam as outras através das chaves, como o Profissional de Saúde que passa sua chave primária como chave para Médicos, Farmacêuticos e Enfermeiros. Aqui, é possível ver que o Administrador tem gerenciamento de todo os processos e a possibilidade de que, quando houver novas filiais, poder separar quem está ligado a cada administração.

### 3.4. Endpoints da API

A API contém os seguintes endpoints principais:

Tabela 2 – Endpoints da API

<b>Método</b>	<b>Rota</b>	<b>Descrição</b>
POST	/administrador	Cadastra Administrador
GET	/administrador	Retorna Administrador
POST	/paciente/novo	Cadastrar Paciente
GET	/paciente/:idAdm	Retorna Pacientes da Unidade
GET	/paciente/busca/:cpf	Retorna Paciente com o CPF específico
POST	/profissional	Cadastrar Profissional de Saúde
GET	/profissional/:idAdm	Busca Profissionais de Saúde da Unidade
POST	/medico	Cadastra Médico
GET	/medico	Retorna todos os Médicos
GET	/medico/busca/:crm	Retorna o Médico do CRM específico
POST	/enfermeiro	Cadastra Enfermeiro
GET	/enfermeiro	Retorna todos os Enfermeiros
GET	/enfermeiro/busca/:coren	Retorna o Enfermeiro do COREN específico
POST	/farmaceutico	Cadastra Farmacêutico
GET	/farmaceutico	Retorna todos os Farmacêuticos
GET	/farmaceutico/busca/:crf	Busca Farmacêutico pelo CRF
POST	/leito	Cadastra Leito na Unidade
GET	/leito/:idAdm	Retorna Leitos da Unidade
GET	/leito/busca/:num	Retorna o Leito de acordo com o número
GET	/leito/disponíveis/:idAdm	Retorna Leitos Disponíveis
POST	/suprimento	Cadastra Suprimento na Unidade
GET	/suprimento	Recupera Suprimentos da Unidade

GET	/suprimento/busca/:nome	Recupera o Suprimento de acordo com o Nome
-----	-------------------------	--

Fonte: Próprio Autor

### 3.5. Tecnologias de Persistência

Para a persistência dos dados, foi utilizado o **MySQL** que é um banco de dados relacional gratuito e open source, tem alta performance, suporta o padrão de SQL, integra facilmente com outras plataformas, além da compatibilidade com a linguagem escolhida para esse projeto que foi Java.

Tabela 3 – Tecnologias de Persistência

Camada	Tecnologia
Backend/API REST	Java + Spark Java + Gson + JDBC
Banco de Dados	MySQL
Frontend Desktop	Java Swing (JFrame)

Fonte: Próprio Autor

## 4. IMPLEMENTAÇÃO

Aqui será apresentado a implementação do protótipo funcional parcial do sistema. O objetivo é demonstrar como o backend foi desenvolvido e estruturado e como as requisições foram tratadas. Por questão de tempo, foi optado por desenvolver uma versão mínima viável com um frontend simples para melhor entendimento e exemplificação.

### 4.1. Tecnologias Utilizadas

As tecnologias utilizadas para o desenvolvimento desse protótipo foram:

- Linguagem: Java
- Framework web: Spark Java
- Banco de Dados: MySQL
- Biblioteca JSON: Gson
- Gerenciador de Dependências: Maven
- Ferramenta de Testes: Postman
- Frontend: Java Swing (JFrame)

## 4.2. Estrutura do Sistema

A estrutura de pastas e do sistema foi feita da seguinte forma:

```
/src
+-- classes/
| +-- Administracao.java
| +-- Agenda.java
| +-- Anamnese.java
| +-- AnamneseFeminina.java
| +-- Consulta.java
| +-- Enfermeiro.java
| +-- Farmaceutico.java
| +-- HistoricoInternacao.java
| +-- Internacao.java
| +-- Leito.java
| +-- Medico.java
| +-- MovimentoSuprimento.java
| +-- Paciente.java
| +-- PrescricaoInternacao.java
| +-- ProfissisonalSaude.java
| +-- ProntuarioMedico.java
| +-- Suprimento.java
+--com/
| +-- Main.java
+--controles/
| +-- ControleAdministracao.java
| +-- ControleAgenda.java
| +-- ControleConsulta.java
| +-- ControleEnfermeiro.java
| +-- ControleFarmaceutico.java
| +-- ControleHistoricoInternacao.java
| +-- ControleInternacao.java
| +-- ControleLeito.java
| +-- ControleMedico.java
| +-- ControleMovimentoSuprimento.java
```

```

| +-- ControlePaciente.java
| +-- ControlePrescricao.java
| +-- ControleProfissional.java
| +-- ControleProntuario.java
| +-- ControleSuprimento.java
+--dao/
| +-- AdministracaoDAO.java
| +-- AgendaDAO.java
| +-- AnamneseDAO.java
| +-- ConsultaDAO.java
| +-- DB.java
| +-- EnfermeiroDAO.java
| +-- FarmaceuticoDAO.java
| +-- HistoricoInternacaoDAO.java
| +-- InternacaoDAO.java
| +-- LeitoDAO.java
| +-- MedicoDAO.java
| +-- MovimentoSuprimentoDAO.java
| +-- PacienteDAO.java
| +-- PrescricaoInternacaoDAO.java
| +-- ProfissionalSaudeDAO.java
| +-- ProntuarioDAO.java
| +-- SuprimentoDAO.java
+--enums/
| +-- CategoriaProfissional.java
| +-- CicloMenstrual.java
| +-- DisponibilidadeLeito.java
| +-- Especialidades.java
| +-- IntervaloConsultas.java
| +-- ServicoConsulta.java
| +-- ServicoProntuario.java
| +-- Sexo.java
| +-- TipoAnticoncepcional.java
| +-- TipoEnfermeiro.java

```

| +-- *TipoLeito.java*  
| +-- *TipoSuprimento.java*  
+--*interfaces/*  
| +-- *TelaAgendaCompleta.java*  
| +-- *TelaAnamnesePaciente.java*  
| +-- *TelaCNPJ.java*  
| +-- *TelaCPFPaciente.java*  
| +-- *TelaCPFProfissional.java*  
| +-- *TelaCadastrarEnfermeiro.java*  
| +-- *TelaCadastrarFarmaceutico.java*  
| +-- *TelaCadastrarMedico.java*  
| +-- *TelaCadastrarPaciente.java*  
| +-- *TelaCadastrarDatasConsultas.java*  
| +-- *TelaCadastrarLeito.java*  
| +-- *TelaCadastrarSuprimento.java*  
| +-- *TelaConsulta.java*  
| +-- *TelaEnfermeirosCadastrados.java*  
| +-- *TelaFarmaceuticosCadastrados.java*  
| +-- *TelaInicial.java*  
| +-- *TelaInicialAdministrador.java*  
| +-- *TelaInicialEnfermeiro.java*  
| +-- *TelaInicialFarmaceutico.java*  
| +-- *TelaInicialMedico.java*  
| +-- *TelaInicialPaciente.java*  
| +-- *TelaInternacoesAtuais.java*  
| +-- *TelaLeitosCadastrados.java*  
| +-- *TelaMarcarConsulta.java*  
| +-- *TelaMedicosCadastrados.java*  
| +-- *TelaPacientesCadastrados.java*  
| +-- *TelaPrescricaoInternacao.java*  
| +-- *TelaPrimeiroAcessoADM.java*  
| +-- *TelaSolicitacoesInternacoes.java*  
| +-- *TelaSolicitarSuprimentos.java*  
| +-- *TelaSuprimentosCadastrados.java*

| +-- *TelaVerificarInternacoes.java*

Essa estrutura foi definida e separada dessa forma para que cada parte fique em sua pasta e seja mais fácil de localizar o que desejar sem tanto trabalho.

### 4.3. Telas da Prototipação

O projeto foi criado juntamente com a parte visual simples para facilitar o entendimento das conexões que são realizadas durante o processo. Então, separei algumas telas que achei interessante pra demonstrar um pouco do processo.

Para a compilação do projeto, é necessário digitar no prompt de comando na pasta dentro do projeto: `mvn clean compile exec:java`

**Figura 4 – Tela Inicial do Sistema**



Fonte: Próprio Autor

Essa será a Primeira tela apresentada caso já haja algum Administrador cadastrado no sistema. Caso o Banco de Dados esteja limpo, irá ser uma tela para realizar esse cadastro.

O protótipo que está sendo apresentado está considerando como somente um único Administrador, mas com planejamento no banco de dados e no sistema a possibilidade de trabalhar com filiais.

Tanto o Cadastro de Leitos e Suprimentos seguem o mesmo padrão de tela, que é apresentado somente para o Administrador do sistema, pois é de sua responsabilidade os cuidados desse gerenciamento:

**Figura 5 – Telas de Cadastro de Leitos e Suprimentos**



Leito	Tipo de Leito	Valor	Capacidade	Status
1	QUARTO INDIVIDUAL	RS 800.0	1	Indisponível
2	SUITE	RS 1000.0	1	Disponível
3	UTI	RS 1500.0	10	Disponível

Tipo	Nome	Valor Unitário	Estoque
3	Clopidogrel 300 mg	200.0	10
3	Diprona Sódica 500 mg	0.46	1200
3	Paracetamol 750 mg	0.3	5000
3	Ceftriaxona 1 g	3.5	800
3	Cloreto de Sódio 0.9% (500 mL)	2.2	2000
2	Algodão de Gaze (Rolos)	1.1	3000
0	Seringa Descartável	0.25	10000
3	Lidocaina 2% (Frasco)	1.8	500
3	Omeprazol 20 mg (Cápsula)	0.4	2500
5	Alcool Etilico 70% (500 mL)	2.0	1800
4	Luvas de Procedimento (Par)	0.35	12000

Fonte: Próprio Autor

Os médicos cadastrados podem visualizar diretamente da sua tela Inicial suas próximas consultas além dos pacientes internados.

**Figura 6 – Tela Inicial do Médico**

Fonte: Próprio Autor

AAAA

## 4.4. Trechos de Códigos

### 4.4.1. Recuperação dos Administradores Cadastrados

```
get("/administrador", (req, res) -> { //retorna administrador
    return controleAdministracao.recuperaAdministracao();
});

public ArrayList<Administracao> recuperaAdministracao() {
    ArrayList<Administracao> administradores = admDAO.recuperaAdministracao();
    return administradores;
}
```

```

public ArrayList<Administracao> recuperaAdministracao(){
    ArrayList<Administracao> administradores = new ArrayList<>();
    Connection conn = null;
    Statement stmt = null;
    ResultSet rs = null;

    try{

        conn = DB.getConeConnection();

        stmt = conn.createStatement();

        rs = stmt.executeQuery("select * from administracao");

        System.out.println("----->Administradores Recuperados:");

        while(rs.next()){
            int id = rs.getInt("id");
            String cnpj = rs.getString("cnpj");
            String razaoSocial = rs.getString("razaoSocial");
            String email = rs.getString("email");
            String telefone = rs.getString("telefone");

            Administracao adm = new Administracao(id, cnpj, razaoSocial,
                email, telefone);
            administradores.add(adm);
            System.out.println(adm);
        }

        return administradores;

    } catch (SQLException e){
        System.out.println("!!!!!!Erro ao RECUPERAR dados "
            + "dos Administradores!!!!!!");
    } finally {
        DB.closeResultSet(rs);
        DB.closeStatement(stmt);
        DB.closeConnection();
    }

    return null;
}

```

#### 4.4.2. Cadastro de Novos Pacientes

```

post("/paciente/novo", (req,res) -> { //cadastra paciente
    Paciente paciente = gson.fromJson(req.body(), Paciente.class);
    return controlePaciente.cadastrarPaciente(paciente);
});

public boolean cadastrarPaciente(Paciente paciente){
    boolean confirmacao = pacienteDAO.cadastrarPaciente(paciente);
    return confirmacao;
}

```

```

public boolean cadastrarPaciente(Paciente paciente){
    Connection conn = null;
    PreparedStatement pstmt = null;

    try{

        conn = DB.getConnection();

        pstmt = conn.prepareStatement("INSERT INTO paciente" +
            "(cpf,nome,email,telefone,dataNascimento,sexo,"
            + "idAdministracao) VALUES(?,?,?,?,?,?,?)",
            Statement.RETURN_GENERATED_KEYS);

        pstmt.setString(1, paciente.getCpf());
        pstmt.setString(2, paciente.getNome());
        pstmt.setString(3, paciente.getEmail());
        pstmt.setString(4, paciente.getTelefone());
        pstmt.setString(5, paciente.getDataNascimento());
        pstmt.setInt(6, paciente.getSexo());
        pstmt.setInt(7, paciente.getIdAdministrador());

        int rollsAffected = pstmt.executeUpdate();

        if(rollsAffected > 0){
            ResultSet rs = pstmt.getGeneratedKeys();
            while (rs.next()){
                int id = rs.getInt(1);
                paciente.setId(id);
            }
            System.out.println("----->Paciente Cadastrado:");
            System.out.println(paciente);
            return true;
        }

    } catch (SQLException e){
        System.out.println("!!!!!!Erro ao CADASTRAR o Paciente!!!!!!");
    } finally {
        DB.closeStatement(pstmt);
        DB.closeConnection();
    }

    return false;
}

```

#### 4.4.3. Busca Médico via CRM

```

get("/medico/busca/:crm", (req,res) -> { //retorna medico via CRM
    return controleMedico.buscaMedicoCRM(req.params(":crm"));
});

public Medico buscaMedicoCRM(String crm){
    Medico medico = medicoDAO.buscaMedicoPorCRM(crm);
    return medico;
}

```

```

public Medico buscaMedicoPorCRM(String crm){
    Connection conn = null;
    Statement stmt = null;
    ResultSet rs = null;

    try{

        conn = DB.getConnection();

        stmt = conn.createStatement();

        rs = stmt.executeQuery("select * from medico where "
            + "crm = '" + crm + "'");

        while(rs.next()){
            int id = rs.getInt("idProfissionalSaude");
            String uf = rs.getString("uf");
            String dataInscricao = rs.getString("dataInscricao");
            int especialidade = rs.getInt("especialidade");

            Medico medico = new Medico(crm, uf, dataInscricao,
                especialidade, id, "", "", "", "", "", "");
            System.out.println("---->Enfermeiro Localizado pelo CRM:");
            System.out.println(medico);
            return medico;
        }

    } catch (SQLException e){
        System.out.println("!!!!Erro ao BUSCAR o Médico!!!!");
    } finally {
        DB.closeResultSet(rs);
        DB.closeStatement(stmt);
        DB.closeConnection();
    }

    return null;
}

```

## 4.5. Exemplo de Fluxo

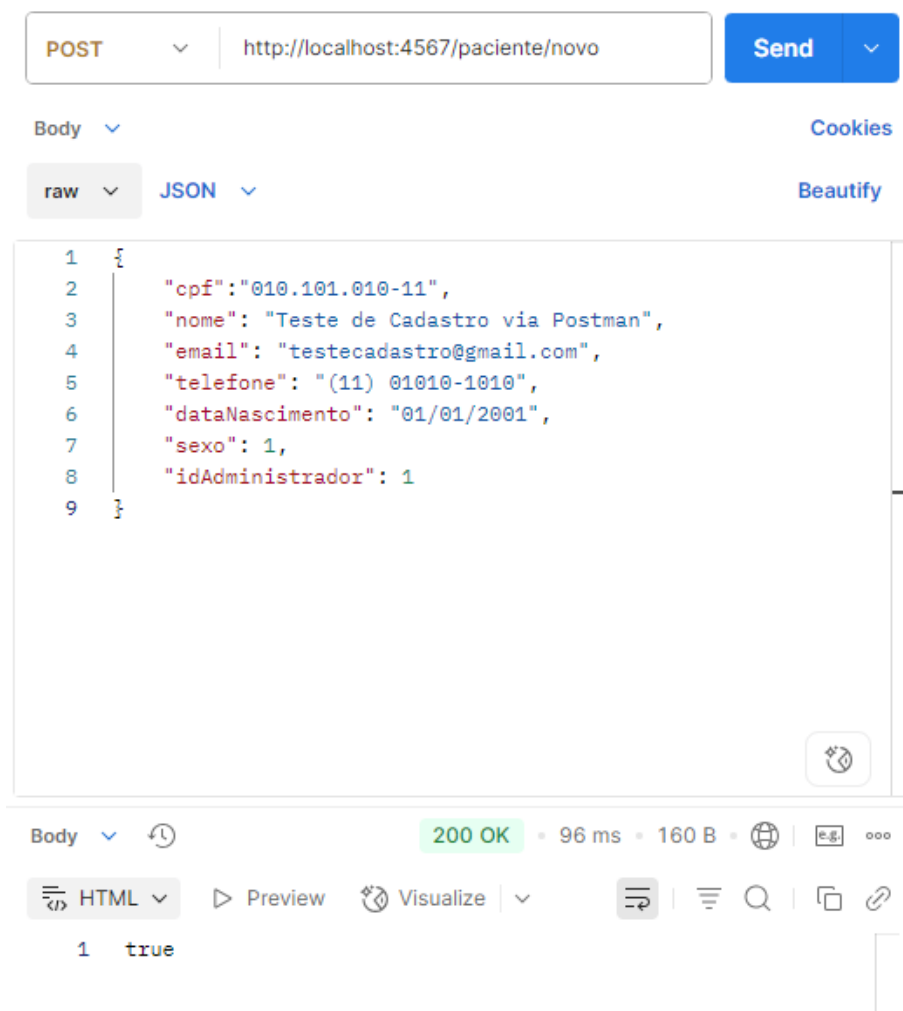
O endpoint GET /administrador retorna todos os administradores cadastrados no sistema. Resposta no Postman:

Body Cookies Headers (4) Test Results (1/1) | ①

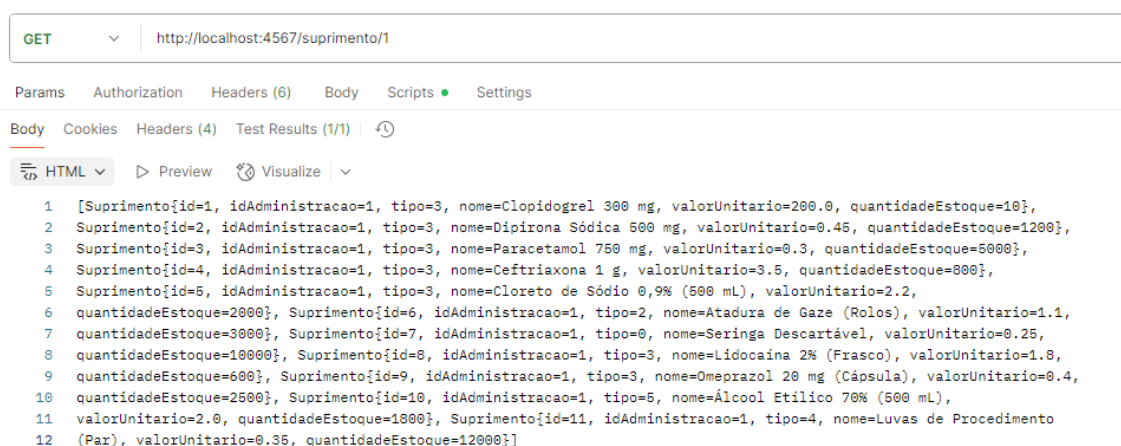
HTML Preview Visualize | v

[Administracao{id=1, cnpj=12.345.678/0001-90, razaoSocial=VidaPlus Saúde, email=administracao@vidaplus.com, telefone=(46) 12345-6789}]

Já o endpoint POST /paciente/novo cadastra um novo paciente e retorna a confirmação como forma de boolean.



O endpoint GET `/suprimento/:idAdm` mostra todos os suprimentos cadastrados em uma unidade:

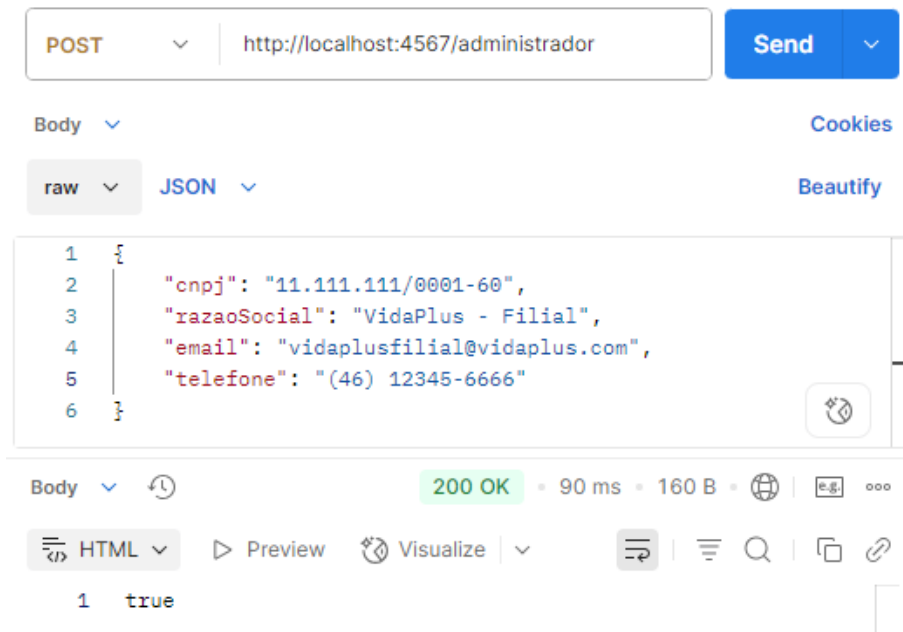


## 5. PLANO DE TESTES

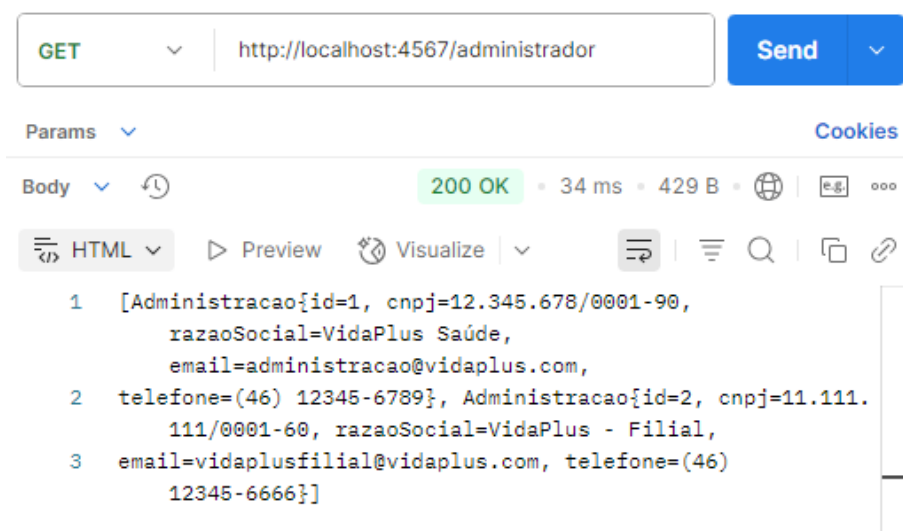
Esse plano de teste teve como objetivo mostrar algumas funcionalidades e validar as principais delas via API REST do sistema desenvolvido em com Spark

Framework e banco de dados MySQL. Neles foram testados métodos como GET e POST, de forma manual diretamente com o Postman.

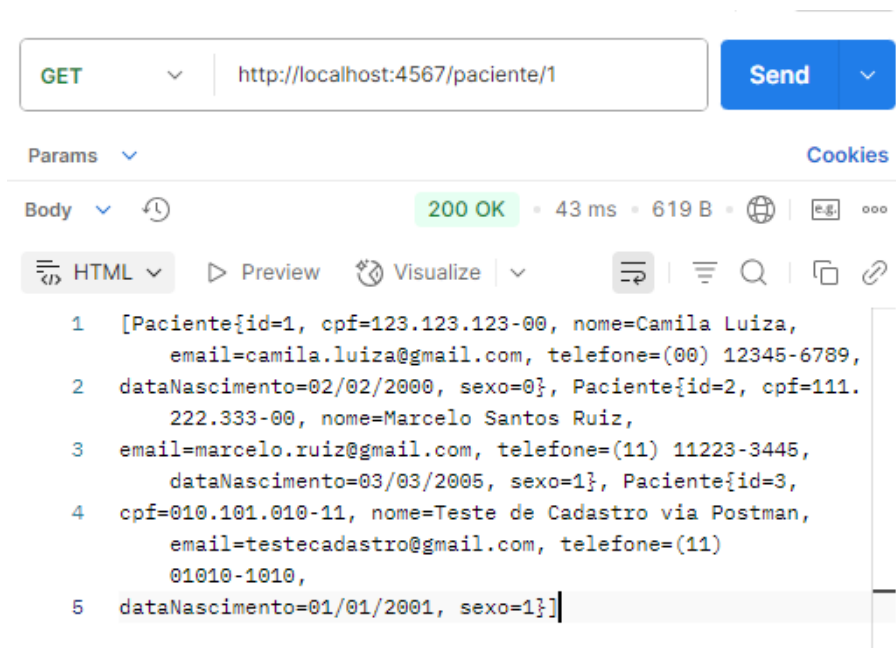
### 5.1. POST /administrador



### 5.2. GET /administrador



### 5.3. GET /paciente/:idAdm



## 6. CONCLUSÃO

Foi um projeto bem complexo pra mim, principalmente por nunca ter utilizado ferramentas como Postman ou frameworks como o Spark, mas fico feliz por ter conseguido atender ao mínimo do que foi proposto, com os conhecimentos que eu já tinha juntamente com novos desafios.

O prazo do projeto foi um pouco apertado pra mim pra que eu conseguisse fazer etapas como de verificações de outros itens, mas ao mesmo tempo, é gratificante ver que o trabalho aconteceu realmente.

Deixo aqui meus agradecimentos e o link do GitHub com o projeto: [Amandickz/VidaPlus: Projeto final do curso de TADS da UNINTER.](https://github.com/Amandickz/VidaPlus)

## REFERÊNCIAS

DESENVOLVIMENTO, C. do. **Requisitos funcionais e não funcionais: o que são e como identificar?**. Casa do Desenvolvedor. Disponível em: <https://blog.casadodesenvolvedor.com.br/requisitos-funcionais-e-nao-funcionais/>.

Acesso em: 15 jun. 2025.

Lima, L. A. O. *et al.* **Gestão Hospitalar E Os Desafios Na Saúde Pública No Brasil**. Disponível em: <https://www.iosrjournals.org/iosr-jbm/papers/Vol26-issue3/Ser-6/G2603063338.pdf>. Acesso em: 13 jul. 2025.

Riviere, R. **Como funciona a gestão hospitalar e qual a sua importância?**. Disponível em: <https://versatilis.com.br/como-funciona-a-gestao-hospitalar/>. Acesso em: 13 jul. 2025.

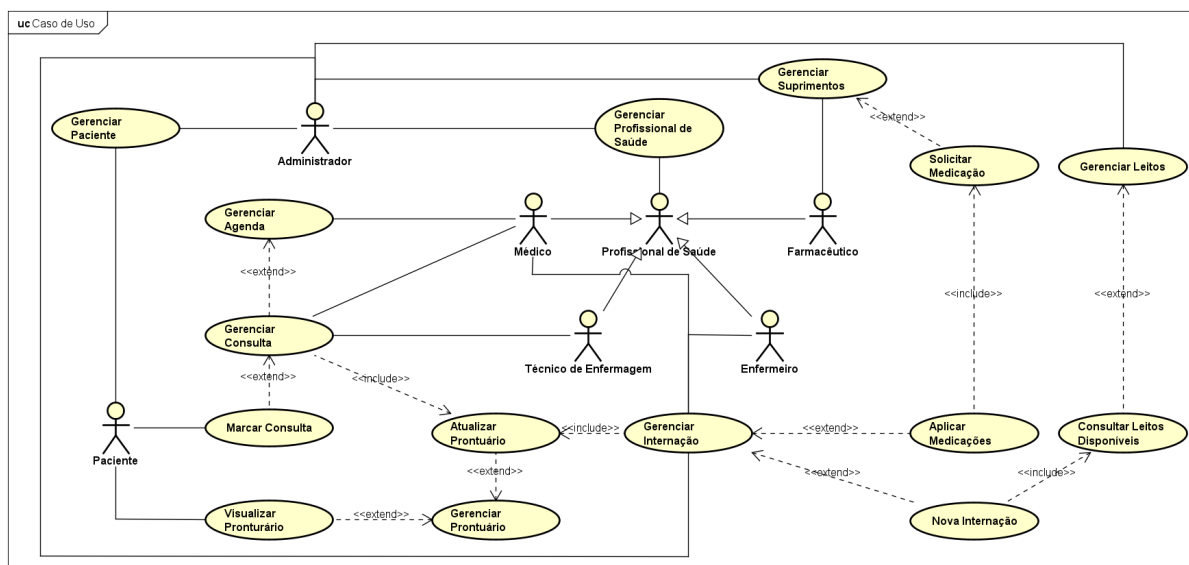
**DIAGRAMAS de Classes**. [S. I.], 05 mar 2021. Disponível em: <https://www.ibm.com/docs/pt-br/rsas/7.5.0?topic=structure-class-diagrams>. Acesso em: 14 jul. 2025.

MEMED, **Hierarquia na área da saúde: como funciona na prática?**. Disponível em: <https://blog.memed.com.br/hierarquia-na-area-da-saude/>. Acesso em 18/04/2025



## ANEXO A – CASO DE USO

### MODELO DE CASO DE USO COMPLETO DO SISTEMA DE GESTÃO HOSPITALAR E DE SERVIÇOS DE SAÚDE (SGHSS)



## ANEXO B – FICHA DE ANAMNESE

### EXEMPLO DE FICHA DE ANAMNESE PARA APLICAÇÃO ANTES DA CONSULTA

#### FICHA DE ANAMNESE

1º) Dados gerais do paciente:	
Nome: _____	
Idade: _____	Sexo: _____ Data de Nasc: ____/____/____ Etnia: _____
Estado Civil: _____	Filhos: ( ) _____ Idade: _____
End: _____	
Tel: _____	e-mail: _____
Profissão: _____	Indicação: _____

2º) Em caso de Emergência avisar:	
Nome: _____	Telefone: _____
Médico: _____	Telefone: _____
Convênio Médico: _____	Cart: _____ Hospital: _____

Queixa Principal: _____
-------------------------

2º) Avaliação do paciente:	
Sinais Vitais: PA: _____ FC: _____ T: _____ Peso: _____ Alt: _____ IMC: _____	
Diabetes: ( )	Hipertensão Arterial ( )
Tabagismo ( )	Varizes ( )
Epilepsia: ( )	Hipotensão Arterial ( )
Próteses Dentárias: ( )	Lesões ( )
Cirurgias: ( ) _____	Data: _____

<p>Tratamento estéticos anteriores: ( ) S ( ) N</p> <p>Exercícios Físicos: ( ) S ( ) N</p> <p>Frequência: _____</p> <p>Ingere líquido com frequência: ( ) S ( ) N</p> <p>Alimentação Balanceada: ( ) S ( ) N</p> <p>Ingere Álcool ( ) S ( ) N</p> <p>Tipo: _____</p> <p>Ingere Chás ( ) S ( ) N</p> <p>Tipo: _____</p> <p>Antecedentes Alérgicos: ( ) S ( ) N</p> <p>Antecedentes Oncológicos: ( ) S ( ) N</p>	<p>É gestante: ( ) S ( ) N</p> <p>Faz algum tratamento médico: ( ) S ( ) N</p> <p>Usa ou já usou ácido? ( ) S ( ) N</p> <p>Cuidados diários? ( ) S ( ) N</p> <p>Usa anticoncepcional? ( ) S ( ) N</p> <p>Ciclo menstrual regular? ( ) S ( ) N</p> <p>Esquimose (sangue acum" roxo"): ( ) S ( ) N</p> <p>Presença de Metais: ( ) S ( ) N</p> <p>Tem algum problema Ortopédico: ( ) S ( ) N</p> <p>Portador de marca passo? ( ) S ( ) N</p> <p>Costuma ficar muito T sentado? ( ) S ( ) N</p> <p>Sensibilidade à dor: ( ) S ( ) N</p>
--	---

# ANEXO C – PRONTUÁRIO DE TRIAGEM

## MODELO DE PRONTUÁRIO DE TRIAGEM



SECRETARIA MUNICIPAL DE SAÚDE DE GRAÇA  
CENTRO DE ATENÇÃO PSICOSSOCIAL – CAPS I  
**PRONTUÁRIO DO CIDADÃO / TRIAGEM**



IDENTIFICAÇÃO		
CNS:	ADMISSÃO: ____ / ____ / ____	PRONTUÁRIO:
NOME DO(A) PACIENTE:		
ENDEREÇO:	Nº:	BAIRRO:
REFERÊNCIA: CASA ( ) / APARTAMENTO ( ) OUTRO: _____		TEL / CEL:
CPF:	SEXO:	NASCIMENTO: ____ / ____ / ____
RAÇA: NEGRA ( ) / AMARELA ( ) / PARDA ( ) / BRANCA ( ) / INDÍGENA ( )		
OCUPAÇÃO:	RENDA FAMILIAR:	
ESCOLARIDADE:	NACIONALIDADE:	
NATURALIDADE:	RELIGIÃO:	
PROCEDÊNCIA:		
ESTADO CIVIL – CASADO ( ) / AMASIADO ( ) / SEPARADO ( ) / VIUVO ( ) / DIVORCIADO ( )		
NOME DO CONJUGUE:		
FILIAÇÃO / FAMÍLIA / RESPONSÁVEL		
MÃE:	PAI:	
RESPONSÁVEL:	TEL / CEL:	
HISTÓRIA CLÍNICA / TRIAGEM		
ORIGEM DO ENCAMINHAMENTO: _____		
QUEIXAS E DURAÇÃO: _____ _____		
HISTÓRIA DA DOENÇA ATUAL: _____ _____		
ANTECEDENTES FAMILIARES: _____		
ANTECEDENTES PESSOAIS: _____		
HISTÓRICO DE USO MEDICAMENTOSO		
ANTERIOR: _____ _____ _____	ATUAL: _____ _____ _____	
HISTÓRIA MÉDICA		
_____ _____ _____ _____		

**MODELO DE CARTEIRA DE ENFERMEIRO EMITIDA PELO CONSELHO REGIONAL DE ENFERMAGEM DO PARANÁ (COREN/PR)**

28



## ANEXO E – MODELO CARTEIRA DE FARMACÊUTICO

### MODELO DA CÉDULA DE IDENTIFICAÇÃO DE FARMACÊUTICO EMITIDA PELO CONSELHO FEDERAL DE FARMÁCIA (CRF)

#### FRENTE



Pant 376

P.7412 P.377

CMYK

Pant. 2945

IR

#### VERSO



Pant 376

P.7412 P.377

Pant. 2945

Antiscanner

OVI

Invisível

## ANEXO F – MODELO CARTEIRA DE MÉDICO

### MODELO DA CÉDULA DE IDENTIDADE MÉDICA EMITIDA PELO CONSELHO FEDERAL DE MEDICINA (CFM)

