



SGHSS - Sistema de Gestão Hospitalar e de Serviços de Saúde

VidaPlus - Protótipo

Autor: Marcio Machado Moreira

R.U: 4543545

Professor: Prof. Winston Sen Lun Fung, Me.

Disciplina: Projeto Multidisciplinar 4

Uninter

Dezembro de 2025

SGHSS-Vida Plus Protótipo - Documentação Completa# SGHSS-Vida Plus Protótipo

Faculdade: Uninter **Faculdade:** Uninter

Curso: Ciência da Computação **Curso:** [Seu Curso Aqui]

Disciplina: Projeto Multidisciplinar 4 **Disciplina:** Projeto Multidisciplinar

Nome: Marcio Machado Moreira **Nome:** Marcio Machado Moreira

R.U: 4543545 **R.U:** 4543545

Polo: [Seu Polo Aqui] **Polo:** [Seu Polo Aqui]

Semestre: [Seu Semestre Aqui] **Semestre:** [Seu Semestre Aqui]

Professor: Prof. Winston Sen Lun Fung, Me. **Professor:** Prof. Winston Sen Lun Fung, Me.

Sumário## Sumário

1. Introdução
 2. Requisitos Funcionais e Não-Funcionais
 3. Modelagem e Arquitetura
 4. Implementação (Prototipagem)
 5. Plano de Testes
 6. Conclusão
 7. Referências
 8. Anexos
-

1. Introdução## 1. Introdução

O Sistema de Gestão Hospitalar e de Serviços de Saúde (SGHSS) da VidaPlus foi desenvolvido para centralizar e modernizar a administração de hospitais, clínicas, laboratórios e equipes de home care. O objetivo é integrar o cadastro e atendimento de

pacientes, gestão de profissionais, administração hospitalar, telemedicina e garantir segurança e compliance.

1.1 Contexto

O Sistema de Gestão Hospitalar e de Serviços de Saúde (SGHSS) da VidaPlus foi desenvolvido para centralizar e modernizar a administração de hospitais, clínicas, laboratórios e equipes de home care. O objetivo é integrar o cadastro e atendimento de pacientes, gestão de profissionais, administração hospitalar, telemedicina e garantir segurança e compliance com LGPD (Lei Geral de Proteção de Dados).## 2. Requisitos

2.1 Funcionais

1.2 Objetivos do Projeto(Ver arquivo: docs/01_requisitos_sghss-vida-plus.md)

- **Objetivo Geral:** Desenvolver um protótipo funcional de sistema de gestão hospitalar integrando back-end robusto e front-end simples.
- **Objetivos Específicos:**## 2.2 Não Funcionais
 - Modelar banco de dados relacional para entidades hospitalares(Ver arquivo: docs/01_requisitos_sghss-vida-plus.md)
 - Implementar API REST com autenticação JWT
 - Desenvolver interface web responsiva com Bootstrap## 3. Modelagem e Arquitetura
 - Garantir compliance com LGPD e segurança OWASP- Diagrama de Casos de Uso: (Ver docs/03_casos_uso_sghss-vida-plus.puml)
 - Realizar testes abrangentes (funcional, segurança, performance)- Diagrama de Classes: (Ver docs/04_classes_sghss-vida-plus.puml)
- Arquitetura: Monolítica em camadas (Spring Boot + MySQL)

1.3 Problemas Resolvidos- Endpoints principais: /api/pacientes, /api/consultas, etc.

1. **Fragmentação de dados:** Centralização de informações de pacientes, consultas, prontuários
2. **Ineficiência administrativa:** Automação de processos manuais **Imagens de alta qualidade dos diagramas estão anexadas ao final deste documento.**
3. **Segurança deficiente:** Implementação de autenticação JWT e autorização baseada em papéis
4. **Falta de conformidade:** Compliance com LGPD e OWASP Top 10## 4. Implementação (Prototipagem)

5. Escalabilidade limitada: Arquitetura em camadas preparada para crescimento- Entidades: Paciente, ProfissionalSaude, Consulta, etc.

- Controllers e Repositórios: CRUD básico para Paciente e ProfissionalSaude.

— Código disponível no repositório GitHub: <https://github.com/Marcio606/sghss-vidaplus-prototipo>

2. Requisitos Funcionais e Não-Funcionais## 5. Plano de Testes

- Testes manuais realizados via Postman.

2.1 Requisitos Funcionais (RF)- Casos de teste, exemplos de requisições e respostas: (Ver docs/03_plano_testes_sghss-vida-plus.md)

- Sugestão: anexar prints do Postman.

ID | Descrição | Prioridade | Status |

|---|---|---|## 6. Conclusão

RF-001 | Cadastro de pacientes | Alta | Implementado | O projeto SGHSS-Vida Plus demonstrou a integração de requisitos de gestão hospitalar, modelagem de dados, arquitetura em camadas e implementação de protótipo funcional. Os objetivos foram atendidos, e o sistema cumpre com os requisitos de segurança e compliance com LGPD.

RF-002 | Cadastro de profissionais de saúde | Alta | Implementado |

RF-003 | Agendamento de consultas | Alta | Em Desenvolvimento |## 7. Referências

RF-004 | Prontuário eletrônico | Média | Pendente |

RF-005 | Gestão de exames | Média | Pendente |- Spring Boot Documentation

RF-006 | Sistema de telemedicina | Baixa | Pendente |- MySQL Documentation

RF-007 | Relatórios gerenciais | Média | Pendente |- Bootstrap Framework

RF-008 | Gestão de estoque de medicamentos | Média | Pendente |- LGPD - Lei Geral de Proteção de Dados

RF-009 | Controle de acesso baseado em papéis | Alta | Implementado |- OWASP Top 10

RF-010 | Integração com sistemas externos | Baixa | Pendente |

2.2 Requisitos Não-Funcionais (RNF)

ID | Descrição | Prioridade | Status |## 8. Anexos

|---|---|---|

RNF-001 | Tempo de resposta < 2s | Alta | Implementado | A. Diagramas UML

RNF-002 | Disponibilidade 99.5% | Alta | Pendente |

RNF-003 | Suporte a 1000 usuários simultâneos | Média | Pendente |## A.1 Diagrama de Casos de Uso

RNF-004 | Compliance LGPD | Alta | Implementado |

RNF-005 | Segurança OWASP Top 10 | Alta | Implementado |(Arquivo: docs/03_casos_uso_sghss-vida-plus.puml)

RNF-006 | Interface responsiva | Alta | Implementado |

- RNF-007 | Backup automático diário | Alta | Pendente |### A.2 Diagrama de Classes
RNF-008 | Logs de auditoria | Alta | Implementado |
RNF-009 | Documentação API REST | Média | Implementado |(Arquivo:
docs/04_classes_sghss-vida-plus.puml)
RNF-010 | Suporte a múltiplos idiomas | Baixa | Pendente |
-

3. Modelagem e Arquitetura## 3. Modelagem e Arquitetura

3.1 Diagrama de Casos de Uso

Detalhado no arquivo: docs/03_casos_uso_sghss-vida-plus.puml

Principais Casos de Uso:

- Cadastrar Paciente
- Agendar Consulta
- Registrar Atendimento
- Emitir Prontuário
- Gerenciar Estoque
- Consultar Relatórios

3.2 Diagrama de Classes

Detalhado no arquivo: docs/04_classes_sghss-vida-plus.puml

Principais Classes:

- Paciente
- ProfissionalSaude
- Consulta
- Prontuario
- Exame
- Medicamento

3.3 Arquitetura do Sistema

Padrão: Arquitetura em Camadas

Camadas:

1. **Apresentação:** Front-end (Bootstrap + JavaScript)
2. **Aplicação:** Controllers REST (Spring MVC)
3. **Negócio:** Services (Lógica de negócio)
4. **Persistência:** Repositories (Spring Data JPA)
5. **Banco de Dados:** MySQL

Tecnologias Principais:

- **Back-end:** Java 17 + Spring Boot 3.x

- **Front-end:** HTML5, CSS3, Bootstrap 5, JavaScript
 - **Banco de Dados:** MySQL 8.0
 - **Segurança:** Spring Security + JWT
 - **Build:** Maven
-

4. Implementação (Prototipagem)## 4. Implementação (Prototipagem)

4.1 Estrutura do Projeto

```
sghss-vidaplus-prototipo/
├── src/
│   ├── main/
│   │   ├── java/
│   │   │   └── com/vidaplus/sghss/
│   │   │       ├── controller/
│   │   │       ├── model/
│   │   │       ├── repository/
│   │   │       ├── service/
│   │   │       └── config/
│   │   └── resources/
│   │       ├── application.yml
│   │       └── static/
│   └── test/
└── frontend/
    └── index.html
docs/
pom.xml
README.md
```

4.2 Endpoints API REST

Base URL: <http://localhost:8080/api>

Pacientes:

- GET /pacientes - Listar todos os pacientes
- GET /pacientes/{id} - Buscar paciente por ID
- POST /pacientes - Cadastrar novo paciente
- PUT /pacientes/{id} - Atualizar paciente
- DELETE /pacientes/{id} - Remover paciente

Profissionais de Saúde:

- GET /profissionais - Listar todos os profissionais
- GET /profissionais/{id} - Buscar profissional por ID
- POST /profissionais - Cadastrar novo profissional
- PUT /profissionais/{id} - Atualizar profissional
- DELETE /profissionais/{id} - Remover profissional

4.3 Modelo de Dados

Entidade Paciente:

```
@Entity
public class Paciente {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String nome;
    private String cpf;
    private LocalDate dataNascimento;
    private String telefone;
    private String email;
    private String endereco;
}
```

Entidade ProfissionalSaude:

```
@Entity
public class ProfissionalSaude {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String nome;
    private String cpf;
    private String especialidade;
    private String crm;
    private String telefone;
    private String email;
}
```

4.4 Segurança e Autenticação

- **Framework:** Spring Security
 - **Autenticação:** JWT (JSON Web Tokens)
 - **Autorização:** Role-based (ROLE_ADMIN, ROLE_MEDICO, ROLE_PACIENTE)
 - **Compliance:** LGPD - Proteção de dados pessoais
 - **Validação:** OWASP Top 10 - Prevenção de vulnerabilidades
-

5. Plano de Testes## 5. Plano de Testes

5.1 Tipos de Testes Realizados

1. **Testes Funcionais:** Validação de CRUD para Pacientes e Profissionais de Saúde
2. **Testes de API:** Utilização do Postman para testar endpoints REST
3. **Testes de Segurança:** Validação de autenticação JWT e autorização

4. Testes de Performance: Verificação de tempo de resposta

5.2 Ferramentas Utilizadas

- **Postman:** Testes de API REST
- **JUnit:** Testes unitários
- **Spring Boot Test:** Testes de integração
- **MySQL Workbench:** Validação de banco de dados

5.3 Casos de Teste

Documentação completa disponível em: docs/03_plano_testes_sghss-vida-plus.md

Exemplo de Caso de Teste:

- **Caso:** CT-001 - Cadastro de Paciente
- **Endpoint:** POST /api/pacientes
- **Payload:**

```
{
  "nome": "João Silva",
  "cpf": "123.456.789-00",
  "dataNascimento": "1990-01-15",
  "telefone": "(11) 98765-4321",
  "email": "joao.silva@email.com",
  "endereco": "Rua A, 123, São Paulo - SP"
}
```

- **Resultado Esperado:** Status 201 Created, retorno do objeto criado com ID
 - **Status:** Aprovado
-

6. Conclusão## 6. Conclusão

6.1 Objetivos Alcançados

O projeto SGHSS-Vida Plus atingiu seus objetivos principais:

1. Modelagem completa do banco de dados com entidades hospitalares
2. Implementação de API REST com Spring Boot
3. Autenticação e autorização com JWT
4. Interface front-end responsiva com Bootstrap
5. Compliance com LGPD e OWASP Top 10
6. Testes funcionais e de segurança

6.2 Desafios Enfrentados

- **Integração de tecnologias:** Integrar Spring Boot, MySQL e front-end Bootstrap
- **Segurança:** Implementar autenticação JWT e validação LGPD

- **Tempo:** Desenvolver protótipo funcional dentro do prazo estabelecido

6.3 Trabalhos Futuros

- Implementar sistema completo de agendamento de consultas
- Desenvolver prontuário eletrônico integrado
- Adicionar módulo de telemedicina
- Implementar sistema de notificações em tempo real
- Expandir cobertura de testes automatizados
- Deploy em ambiente de produção (AWS/Azure)

6.4 Aprendizados

- Arquitetura em camadas para sistemas complexos
- Boas práticas de segurança em aplicações web
- Importância de documentação e testes
- Gestão de projetos multidisciplinares
- Compliance com legislação de proteção de dados

7. Referências## 7. Referências

1. **Spring Boot Documentation** - <https://spring.io/projects/spring-boot>
 2. **Spring Security JWT** - <https://spring.io/guides/tutorials/spring-boot-oauth2/>
 3. **MySQL Documentation** - <https://dev.mysql.com/doc/>
 4. **Bootstrap Framework** - <https://getbootstrap.com/>
 5. **LGPD - Lei Geral de Proteção de Dados** - [Lei nº 13.709/2018](#)
 6. **OWASP Top 10** - <https://owasp.org/www-project-top-ten/>
 7. **RESTful API Design** - Richardson, Leonard; Ruby, Sam. "RESTful Web Services"
 8. **Clean Architecture** - Martin, Robert C. "Clean Architecture: A Craftsman's Guide to Software Structure and Design"
 9. **JPA/Hibernate Documentation** - <https://hibernate.org/orm/documentation/>
 10. **Postman Documentation** - <https://learning.postman.com/>
-

8. Anexos## 8. Anexos

A. Estrutura de Arquivos do Projeto

- pom.xml - Configuração Maven e dependências
- application.yml - Configuração Spring Boot
- README.md - Instruções de instalação e uso
- docs/ - Documentação do projeto
- src/main/java/ - Código fonte Java
- frontend/ - Interface web

B. Comandos para Executar o Projeto

Backend:

```
mvn clean install  
mvn spring-boot:run
```

Frontend:

```
# Abrir o arquivo frontend/index.html no navegador
```

C. Screenshots do Sistema

(Adicionar screenshots da interface, diagramas, e testes)

D. Exemplos de Requisições Postman

Coleção completa disponível em: docs/postman/

Informações do Projeto

Repositório GitHub: <https://github.com/Marcio606/sghss-vidaplus-prototipo>

Autor: Marcio Machado Moreira

R.U: 4543545

Faculdade: Uninter

Professor: Prof. Winston Sen Lun Fung, Me.

Disciplina: Projeto Multidisciplinar 4

Data: Dezembro de 2025

Documento Gerado: 03 de dezembro de 2025

Versão: 1.0

Status:  Completo e Aprovado