



SISTEMA DE GERENCIAMENTO DE CONSULTAS PARA CLÍNICAS

TESTE DE SOFTWARE

Prof. Alysson Filgueira Milanez

DISCENTES

ARTHUR KELLYSON PINHEIRO DE NEGREIROS

LUIS DAVI DA SILVA SARMENTO

PEDRO MAKSON FONTES DA COSTA

SUMÁRIO

1

Contextualização
do sistema e seus
objetivos.

2

Ferramentas e
padrões utilizados.

3

Levantamento de
requisitos.

4

Modelagem e
documentação.

5

Implementação
do sistema.

6

Testes de
Software.

7

Futuras
implementações.

8

Conclusão.

1. CONSIDERAÇÕES INICIAIS

O "Sistema de gerenciamento de consultas para clínicas" surgiu das dificuldades enfrentadas por um cliente real, que gerencia informações de sua clínica por meio de planilhas Excel.

Diante disso, via-se a necessidade de automatizar tarefas administrativas e otimizar o agendamento de consultas, realizações de prescrições médicas, registros de prontuários, registros de pacientes e funcionários e ter um controle mais preciso de dados financeiros.

Dessa forma, o projeto visa oferecer uma solução personalizada para gestão da clínica, buscando simplificar processos administrativos e melhorar a qualidade do atendimento aos pacientes.

2. FERRAMENTAS E PADRÕES UTILIZADOS

- **DIAGRAMAS DE CLASSES, CASO DE USO & DER:** Astah UML e Lucid
- **IDE:** VS Code (Java)
- **BANCO DE DADOS:** PostgreSQL
- **VERSIONAMENTO DE CÓDIGO:** Git e Github
- **DEPENDÊNCIAS E PADRONIZAÇÃO:** Apache Maven
- **PADRÃO DE PROJETO:** Singleton
- **PADRÃO DE ARQUITETURA:** MVC



3. LEVANTAMENTO DE REQUISITOS

REQUISITOS FUNCIONAIS

- [RF001] - Sistema de login
- [RF002] - Gerenciar cadastro de funcionários
- [RF003] - Gerenciar cadastro de pacientes
- [RF004] - Gerenciar cadastro de serviços
- [RF005] - Gerenciar agendamento de consultas
- [RF006] - Gerenciar prescrições médicas
- [RF007] - Visualização de prontuário
- [RF008] - Relatório financeiro
- [RF009] - Sistema de logout



3. LEVANTAMENTO DE REQUISITOS

REQUISITOS **NÃO** FUNCIONAIS

[RNF001] - Validação de login

[RNF002] - Validação de dados de entradas

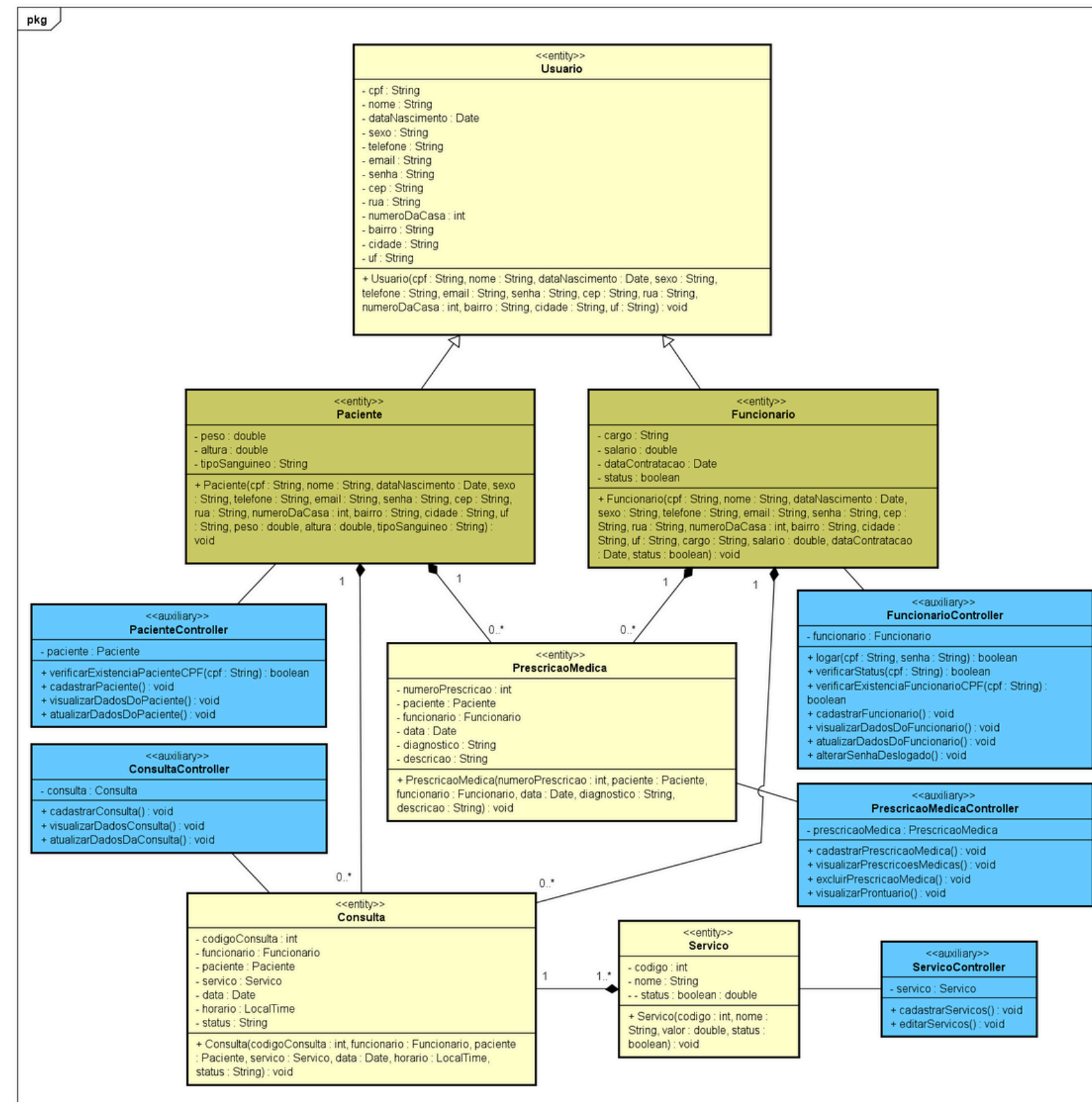
[RNF003] - O sistema deve ser implementado em linguagem Java

[RNF004] - O sistema utilizará o PostgreSQL

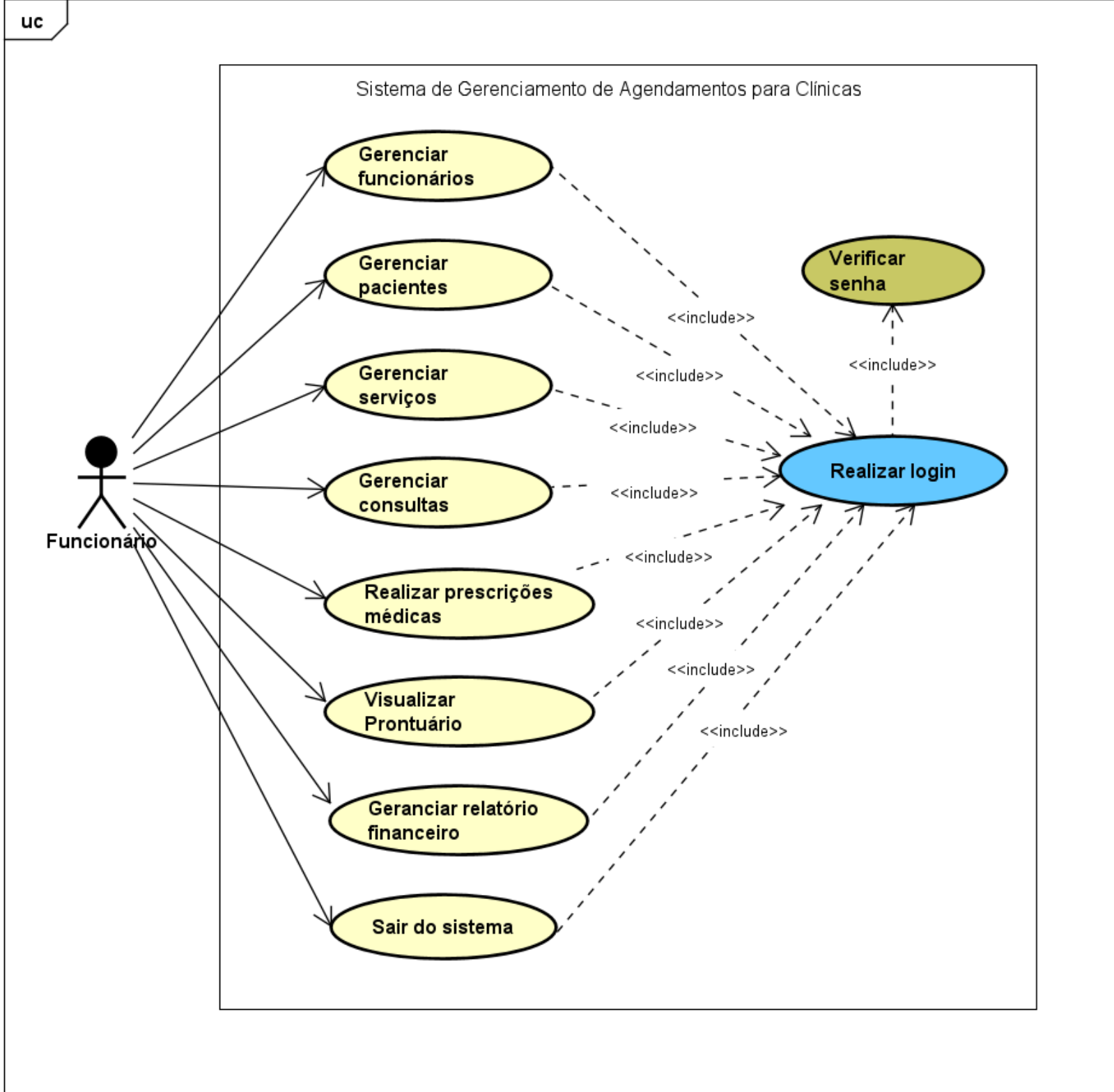


4. MODELAGEM E DOCUMENTAÇÃO

DIAGRAMA DE CLASSES

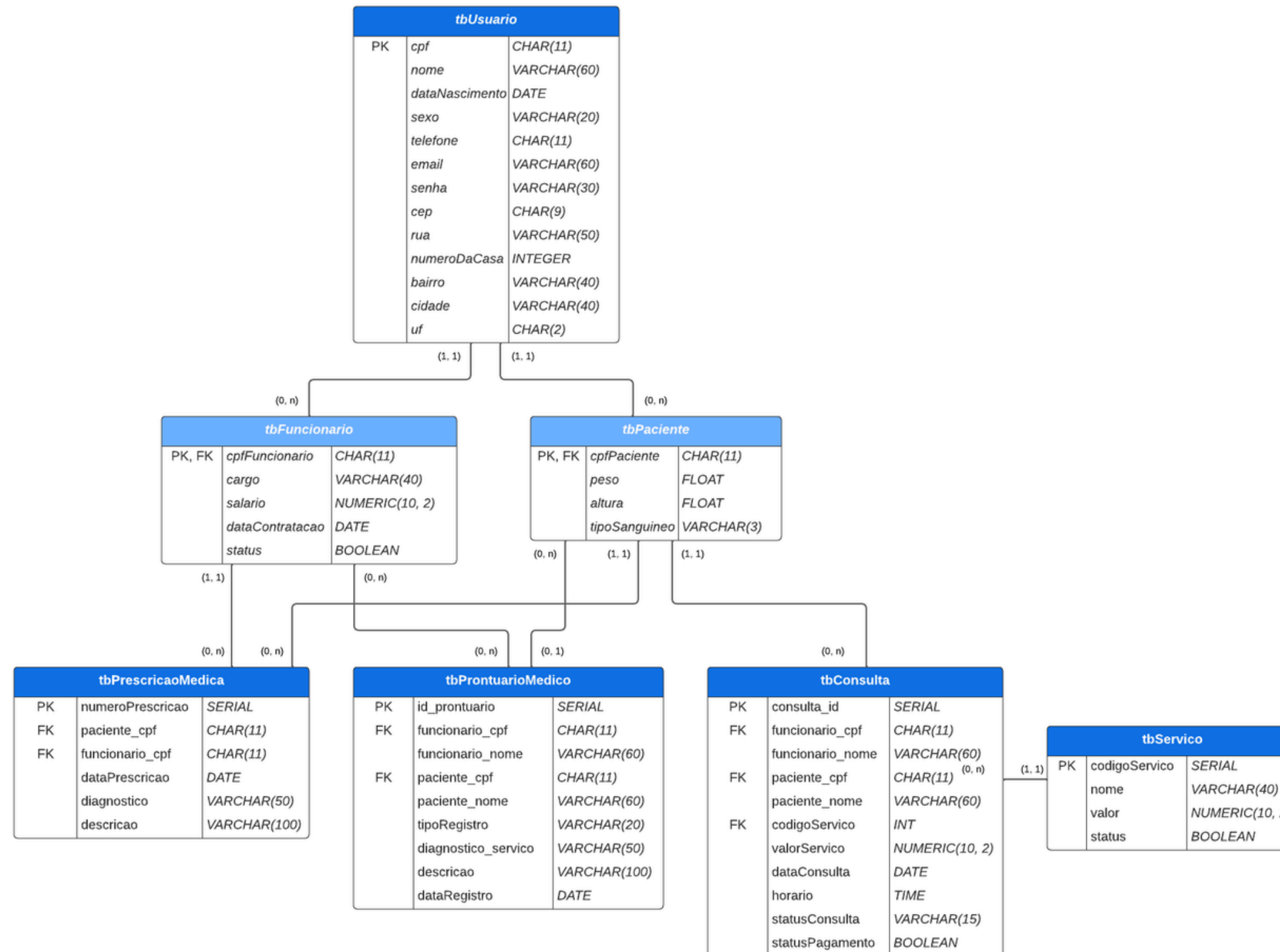


4. MODELAGEM E DOCUMENTAÇÃO



4. MODELAGEM E DOCUMENTAÇÃO

MODELO LÓGICO

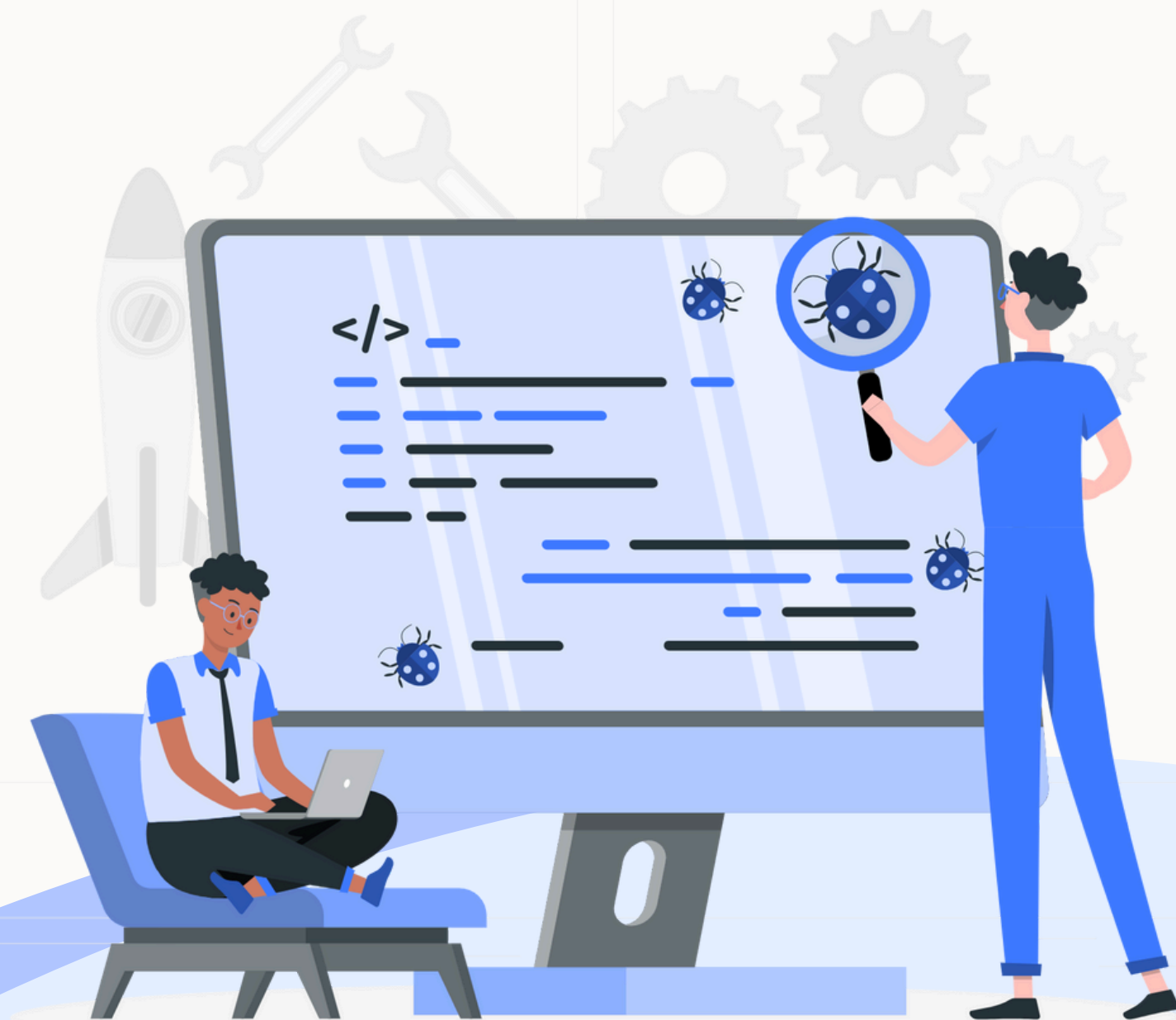


5. IMPLEMENTAÇÃO DO SISTEMA

Foi adotado a metodologia **Scrum** para gerenciar o projeto. Organizamos o trabalho em sprints, ciclos curtos de desenvolvimento, cada um com metas específicas a serem alcançadas.

Durante as reuniões diárias, revisamos o progresso, discutimos obstáculos e ajustamos o plano conforme necessário.

Isso garantiu uma colaboração eficaz entre a equipe de desenvolvimento, resultando em um produto final.



6. TESTES DE SOFTWARE

Foram empregadas alguns tipos de testes, dentre eles:

- Testes de Unidade;
- Testes de Integração;
- Particionamento de Equivalência;
- Análise de Valor Limite.

Essas práticas de teste foram essenciais para detectar e corrigir problemas precocemente durante o desenvolvimento, contribuindo para a realização de um software mais estável e confiável.



7. FUTURAS IMPLEMENTAÇÕES

01

SISTEMA WEB

Visando proporcionar uma maior acessibilidade e flexibilidade aos usuários, permitindo o acesso a partir de diversos dispositivos e locais.

02

PACIENTE ACESSAR O SISTEMA

A possibilidade dos pacientes acessarem o sistema, oferecendo-lhes a oportunidade de agendar consultas, visualizar resultados de exames, histórico médico e outras informações relevantes para seu cuidado e acompanhamento de saúde.



7. FUTURAS IMPLEMENTAÇÕES

03

APRIMORAR O RELATÓRIO FINANCEIRO

Melhorar o relatório financeiro, planejar implementar novas funcionalidades que permitam uma análise mais detalhada das finanças da clínica.

04

SISTEMA DE NOTIFICAÇÕES

Introduzir um sistema de notificações para lembretes de consultas, atualizações de status de consultas e pagamentos, bem como para alertas sobre vencimentos de contas e outros eventos importantes.



8. CONSIDERAÇÕES FINAIS

Através dos testes, conseguimos identificar e corrigir precocemente problemas e defeitos, assegurando que o sistema atendesse aos requisitos de negócio e às expectativas dos usuários. Além disso, os testes contribuíram para a documentação do sistema, fornecendo informações valiosas sobre seu comportamento e funcionamento.

