



UNIVERSIDADE
FEDERAL DO CEARÁ

Clínica Dentária

Heloísa Éven Feitoza Ricardino, Italo Fernando Silva Dutra, Marllon Vinicius de Sousa Pinheiro, Lucas Dantas Brandolt, Marcel Bastos Bleasby, Marcus Luan Angelo Ferreira

**Orientador: Profa. Dsc. Jacilane de Holanda Rabelo
(Engenharia de Software)**

**Projeto Detalhado de Software RUS064
Universidade Federal do Ceará**

e-mail dos alunos:

lucasdbrandolt@alu.ufc.br

marcelbleasby@alu.ufc.br

italodutraa@alu.ufc.br

marllonvinicius@alu.ufc.br

heloisaevenfr@alu.ufc.br

O projeto "Clínica Dentária" é um sistema desenvolvido para otimizar a gestão de clínicas odontológicas, facilitando diversas atividades como o agendamento de consultas, o cadastro e acompanhamento de pacientes, o registro de procedimentos e o controle financeiro. Com uma interface intuitiva e funcionalidades bem definidas, o sistema visa melhorar a experiência tanto para os profissionais quanto para os pacientes, garantindo uma gestão eficiente dos recursos da clínica. Ao proporcionar uma plataforma centralizada para todas as operações, o projeto busca simplificar a rotina da clínica, aumentando sua produtividade e qualidade dos serviços prestados. Essa iniciativa surge em resposta à demanda do mercado por soluções que otimizem a administração das clínicas odontológicas, oferecendo maior organização, controle e segurança das informações. Em síntese, o sistema "Clínica Dentária" visa modernizar e aprimorar a gestão clínica, contribuindo para um atendimento mais eficaz e satisfatório aos pacientes.

1. Introdução do Projeto

O projeto "Clínica Dentária" é um sistema desenvolvido para auxiliar na gestão de uma clínica odontológica. Ele foi criado com o objetivo de facilitar o agendamento de consultas, o controle de pacientes, o registro de procedimentos e o gerenciamento financeiro.

1.2. Objetivo do Projeto

1.2.1. Objetivo Geral:

No geral, o objetivo é otimizar a rotina da clínica dentária, proporcionando uma melhor experiência tanto para os profissionais quanto para os pacientes, além de garantir um gerenciamento eficiente dos recursos financeiros do estabelecimento.

1.2.2 Objetivos Específicos:

- Agendamento de Consultas: Permitir que os pacientes agendem consultas de forma fácil e rápida, contribuindo para uma melhor organização da agenda dos profissionais.
- Cadastro de Pacientes: Manter um registro detalhado dos pacientes, incluindo informações pessoais, histórico médico e tratamentos realizados.
- Registro de Procedimentos: Registrar os procedimentos odontológicos realizados em cada consulta, garantindo um acompanhamento preciso do tratamento de cada paciente ao longo do tempo.
- Controle Financeiro: Auxiliar na gestão financeira da clínica, registrando pagamentos, faturamentos, despesas e fornecendo relatórios que auxiliem na tomada de decisões.

1.3. Delimitação do Problema

Considerando os diversos problemas que uma clínica dentária pode enfrentar, com múltiplos atendimentos presenciais, identificam-se os principais desafios enfrentados por clientes e gerentes em certos tópicos:

- Cadastro de Consultas;
- Cadastro de Funcionários e Clientes;
- Consulta de Funcionários e Clientes;
- Modificação de Funcionários e Clientes;
- Reagendamento de Consultas.

1.4. Justificativa da Escolha do Tema

A escolha do tema "Clínica Dentária" é fundamentada em algumas razões claras:

Demanda no Mercado: O setor de saúde, em particular a área odontológica, enfrenta constantes desafios relacionados à gestão eficiente de suas clínicas. A demanda por sistemas de gestão específicos para clínicas dentárias é evidente, uma vez que a administração manual pode ser trabalhosa e propensa a erros.

Viabilidade e Relevância: O desenvolvimento de um sistema de gestão para clínicas dentárias é viável e relevante, uma vez que pode melhorar significativamente a eficiência operacional, a organização dos registros dos pacientes e a gestão financeira das clínicas.

Benefícios para os Usuários Finais: Os profissionais e pacientes das clínicas dentárias serão os principais beneficiários desse sistema, pois terão acesso a uma plataforma que facilita o agendamento de consultas, o registro de pacientes, o acompanhamento de tratamentos e o controle financeiro.

1.5 Definição da reutilização de software

Detalhamento da reutilização:

- **Origem do código:** o código foi reutilizado de um repositório do GitHub. Esse repositório foi escolhido devido à sua robustez e à aderência aos requisitos do nosso projeto.
- **Processo de reutilização:** o processo de reutilização envolveu a clonagem do repositório original, seguido de uma revisão detalhada do código. Após a análise, identificamos as seguintes partes que precisavam ser adaptadas para atender às especificações do nosso projeto:
 - interface: partes da interface gráfica foram adaptadas para se alinhar à identidade visual e usabilidade desejadas para o sistema da clínica odontológica.
 - funcionalidades de cadastro: foram reutilizadas e modificadas funcionalidades relacionadas ao cadastro de pacientes e procedimentos, ajustando-as para atender às necessidades específicas do projeto.
 - alterações na estrutura: a estrutura de pastas e arquivos do projeto original foi reorganizada em um pull request para melhor adequar-se ao nosso fluxo de trabalho e padrões de organização.

1.6. Organização do Trabalho

Este item descreve a organização do documento, delineando as seções e subseções que compõem o projeto. A seguir, é apresentada a estrutura do documento:

Resumo: breve resumo do trabalho, destacando o objetivo, escopo e principais características do sistema.

Introdução do Projeto: contextualização do projeto, explicando a necessidade de um sistema de gestão para clínicas odontológicas e apresentando os objetivos do sistema.

Descrição Geral do Sistema: como seus principais envolvidos e suas características.

Regras de Negócio: descrição das regras de negócio relevantes para o sistema, incluindo restrições e ferramentas de apoio.

Documentação de Casos de Uso: lista e descrição dos atores do sistema, seguido pela documentação do diagrama de caso de uso do sistema.

Diagrama de Classes: descrição do diagrama de classes de projeto do sistema, incluindo todos os elementos da UML.

Diagrama de Sequência: apresentação dos diagramas de sequência do sistema, ilustrando as interações entre os objetos.

Conclusões: resumo das lições aprendidas durante o desenvolvimento do trabalho e reflexão sobre a aplicabilidade dos conceitos estudados na vida profissional. Junto com a percepção individual de cada aluno sobre o projeto.

Ata de Reunião: registro das reuniões realizadas durante o desenvolvimento do projeto, destacando as discussões, decisões e encaminhamentos tomados pela equipe.

Referências: lista das referências utilizadas no trabalho, seguindo o formato APA.

2. Descrição Geral do Sistema

2.1. Principais Envolvidos e suas Características

2.1.1. Usuários do Sistema

O sistema é destinado para clínicas odontológicas, visando os principais usuários, tais como: recepcionista, dentista e administrador financeiro

2.2. Regras de Negócio

Nome	RN001 - Horários para Agendamento de Consultas
Descrição	Os horários disponíveis para agendamento de consultas devem ser definidos previamente pela clínica, levando em consideração a disponibilidade dos profissionais.

Nome	RN002 - Limite de Agendamento de Consultas
Descrição	Cada paciente só pode ter uma consulta agendada por vez.

Nome	RN003 - Política de Cancelamento/Reagendamento
Descrição	O sistema deve permitir o cancelamento e reagendamento de consultas com antecedência mínima de 24 horas.

Nome	RN004 - Relatório de Procedimentos Odontológicos
Descrição	O sistema deve gerar um relatório para cada procedimento realizado durante uma consulta, incluindo detalhes como data, tipo de procedimento e profissional responsável.

Nome	RN005 - Relatórios financeiros
Descrição	Deve ser possível gerar relatórios financeiros periódicos, como balanços mensais e demonstrativos de resultados.

Nome	RN006 - Formas de pagamento
Descrição	O sistema deve oferecer suporte a diferentes formas de pagamento, como dinheiro, cartão de crédito e convênios.

Nome	RN007 - Segurança dos dados
Descrição	O acesso ao sistema deve ser restrito aos profissionais autorizados da clínica, por meio de login e senha individual, em conformidade com a Lei Geral de Proteção de Dados (LGPD) para garantir a segurança e privacidade das informações.

Nome	RN008 - Backup e Recuperação de Dados
Descrição	O sistema deve realizar backups regulares dos dados armazenados, garantindo a disponibilidade e recuperação em caso de falhas ou perdas de dados.

2.1. Técnicas Escolhidas/Ciclo de Vida/Divisão da Equipe

Técnicas de elicitação de requisitos utilizadas:

Observação: foram observadas as funcionalidades de cada parte do sistema e o que elas geram para que fosse possível documentar os requisitos.

Análise de métricas de desempenho: foram analisadas métricas de desempenho do sistema, como tempo de resposta, taxa de erro e uso de recursos, para identificar áreas de melhoria e requisitos de desempenho adicionais.

2.2. Aplicação das Técnicas

Ciclo de Vida do Projeto:

O ciclo de vida do projeto foi estruturado seguindo uma abordagem iterativa e incremental, permitindo adaptações conforme novos requisitos são identificados e priorizados ao longo do desenvolvimento. Isso garante uma maior flexibilidade e capacidade de resposta às mudanças, essenciais em projetos de desenvolvimento de software. Fases de desenvolvimento:

1. Fase de iniciação e planejamento:
 - objetivo: definir o escopo inicial do projeto, papéis e responsabilidades da equipe, e estabelecer um cronograma preliminar.
 - atividades:
 - apresentação do projeto e definição de papéis (reunião 1).
 - discussão sobre o escopo e principais funcionalidades do sistema (reunião 1).
 - definição das técnicas iniciais de elicitação de requisitos, como entrevistas e análise documental (reunião 1).
 - divisão da equipe em desenvolvimento e teste, e distribuição das primeiras tarefas (reunião 1).
2. Fase de elicitação e análise de requisitos:
 - objetivo: levantar e analisar os requisitos do sistema de forma colaborativa e contínua, ajustando as funcionalidades conforme necessário.
 - atividades:
 - realização de pesquisas individuais sobre sistemas similares e levantamento de requisitos iniciais (reunião 2).
 - discussão e identificação de lacunas nos requisitos levantados (reunião 2).
 - utilização de técnicas adicionais, como brainstorming, para complementar a elicitação de requisitos (reunião 2).
 - início da elaboração da documentação do trabalho, com definição das seções e subseções (reunião 2).

3. Fase de projeto e documentação:

- objetivo: consolidar as informações levantadas e elaborar o documento de projeto, organizando e estruturando os elementos do sistema.
- atividades:
 - revisão e consolidação das informações para inclusão no documento de projeto (reunião 3).
 - discussão sobre a estrutura do documento e designação de responsáveis por cada seção (reunião 3).
 - elaboração dos diagramas de caso de uso, classes e sequência, conforme as atividades encaminhadas na reunião (reunião 3).

4. Desenvolvimento iterativo e incremental:

- objetivo: desenvolver o sistema em ciclos iterativos, onde cada iteração entrega uma versão funcional do software, incorporando novos requisitos e melhorias identificadas.
- atividades:
 - implementação das funcionalidades principais em ciclos curtos e revisões constantes.
 - integração de novos requisitos à medida que são identificados durante o desenvolvimento.
 - realização de testes contínuos para garantir a estabilidade e qualidade do sistema.
 - ajustes no cronograma conforme os resultados das iterações e feedback das partes interessadas (reunião 3).

5. Iterações

Cada ciclo de desenvolvimento é seguido por uma revisão detalhada do progresso, onde os feedbacks são analisados e novos requisitos são integrados no planejamento das próximas iterações.

6. Conclusão

A adoção do ciclo de vida iterativo e incremental, definida na terceira reunião, permite à equipe de 6 pessoas desenvolver um sistema flexível e alinhado às necessidades dinâmicas da clínica, com entregas incrementais que asseguram o contínuo alinhamento do projeto com os objetivos estabelecidos.

Divisão da Equipe:

Desenvolvimento de Software (Marllon Vinícius de Sousa Pinheiro, Marcel Bastos Bleasby):

Responsável pela implementação das funcionalidades do sistema, seguindo as especificações de requisitos.

Testes e Qualidade (Heloisa Even Feitoza Ricardino, Italo Fernando Silva Dutra):

Encarregado de garantir a qualidade do software por meio da realização de testes rigorosos e da identificação e correção de defeitos.

Gerenciamento de Projeto(Marcel Bastos Bleasby):

Responsável pelo planejamento, coordenação e acompanhamento do progresso do projeto, garantindo que os prazos sejam cumpridos e os recursos alocados de forma eficiente.

Documentação e Comunicação (Lucas Dantas Brandolt, Marcel Bastos Bleasby) :

Responsável por documentar todos os aspectos do projeto, incluindo requisitos, especificações, procedimentos e comunicações internas e externas. Além disso, a documentação foi compartilhada com todos os membros do grupo para garantir a transparência e o alinhamento.

2.3. Resultados Obtidos

A análise para as funcionalidades veio a partir do repositório e sua hierarquia de objetos, portanto as funcionalidades foram elicitadas e validadas de acordo com o escopo definido acima.

Tabela 1. Requisitos Funcionais do Sistema

Código	Descrição	Prioridade
RF01	O sistema deve permitir que o Administrador realize o cadastro de um paciente, registrando seu nome, CPF, telefone, rua, cidade, estado, senha e indicação.	Essencial
RF02	O sistema deve permitir que o Administrador realize o cadastro de novos Funcionários registrando seu nome, CPF, telefone, rua, cidade, estado, salário e função.	Essencial
RF03	O sistema deve permitir ao Recepcionista consultar e alterar os dados cadastrais dos pacientes, exceto CPF.	Essencial
RF04	O sistema deve permitir ao Paciente consultar seus próprios dados.	Essencial
RF05	O sistema deve permitir ao Recepcionista a alteração de seus dados cadastrados no sistema, exceto salário e função.	Essencial
RF06	O sistema deve permitir aos pacientes a alteração de seus dados cadastrados no sistema, exceto CPF.	Essencial
RF07	O sistema deve permitir ao Administrador a consulta de dados dos Funcionários e Pacientes cadastrados no sistema.	Essencial
RF08	O sistema deve permitir ao Administrador a alteração de dados dos Funcionários e Pacientes cadastrados, exceto CPF.	Essencial
RF09	O sistema deve permitir ao usuário realizar o login no sistema por meio do seu usuário e senha.	Essencial
RF10	O sistema deve permitir ao Administrador gerar relatórios semanais e mensais sobre as movimentações do sistema.	Importante

RF11	O sistema deve permitir ao Financeiro gerar relatórios semanais e mensais sobre os lucros da clínica.	Importante
RF12	O sistema deve permitir ao Dentista confirmar sua disponibilidade para exames solicitados por um paciente.	Essencial
RF13	O sistema deve permitir que o Financeiro faça alterações nos salários dos Dentistas e Recepcionistas.	Importante
RF14	O sistema deve permitir ao Recepcionista confirmar as consultas.	Essencial
RF15	O sistema deve permitir que o Recepcionista realize o cadastro de um paciente, registrando seu nome, CPF, telefone, rua, cidade, estado, senha e indicação.	Essencial
RF16	O sistema deve permitir que o Administrador faça a exclusão da conta de Pacientes e Funcionários se baseando no CPF cadastrado, desde que o mesmo não tenha nenhuma pendência.	Essencial
RF17	O sistema deve permitir que o Recepcionista faça a exclusão da conta de Pacientes se baseando no CPF cadastrado, desde que o mesmo não tenha nenhuma pendência.	Essencial

Tabela 2. Requisitos Não-Funcionais do Sistema

Código	Descrição	Categoria	Prioridade
RNF01	O sistema deve registrar todos os dados gerados em um sistema de banco de dados em Mongo. AES 128bits	Segurança	Essencial
RNF02	A interface do usuário deve permitir que 90% dos novos usuários conclua uma tarefa de agendamento de consulta em menos de 5 minutos. O tempo médio de aprendizado do sistema não deve exceder 30 minutos para os novos usuários.	Usabilidade	Importante
RNF03	O sistema deve utilizar o sistema de criptografia AES 128bits para o armazenamento dos dados.	Segurança	Essencial
RNF04	O sistema deve estar funcionando no período de 7h da manhã até as 20h.	Confiabilidade	Essencial
RNF05	O sistema deve fazer backup dos dados do sistema diariamente.	Confiabilidade	Essencial
RNF06	O tempo médio de resolução de problemas não deve exceder 24 horas. A taxa de retrabalho não deve ser superior a 10% do esforço total de desenvolvimento.	Manutenibilidade	Importante

3. Documentação de Casos de Uso

3.1. Atores

Tabela 3. Atores do Sistema

#	Ator	Definição
1	Funcionário	Visualizar seus dados, alterar os dados(Exceto função e salário), gerar arquivos e salvar arquivos.
2	Admin	Cadastrar, Editar, Listar e excluir dados de Paciente e Funcionário.
3	Paciente	Visualizar seus dados, alterar os dados, agendar consultas e visualizar consultas.
4	Recepcionista	Responsável por confirmar exames solicitados pelo paciente, altera e consulta dados do paciente (exceto o CPF) e seus próprios dados.
5	Dentista	Responsável por confirmar sua disponibilidade para exames.
6	Financeiro	Responsável por manter dados dos funcionários, dos pacientes,

3.2. Diagrama de Caso de Uso

O diagrama de caso de uso é uma ferramenta crucial para o entendimento, documentação e comunicação dos requisitos do sistema, contribuindo para o sucesso do projeto ao garantir que o sistema atenda às necessidades dos usuários de forma eficaz e eficiente. O diagrama gerado foi o seguinte:

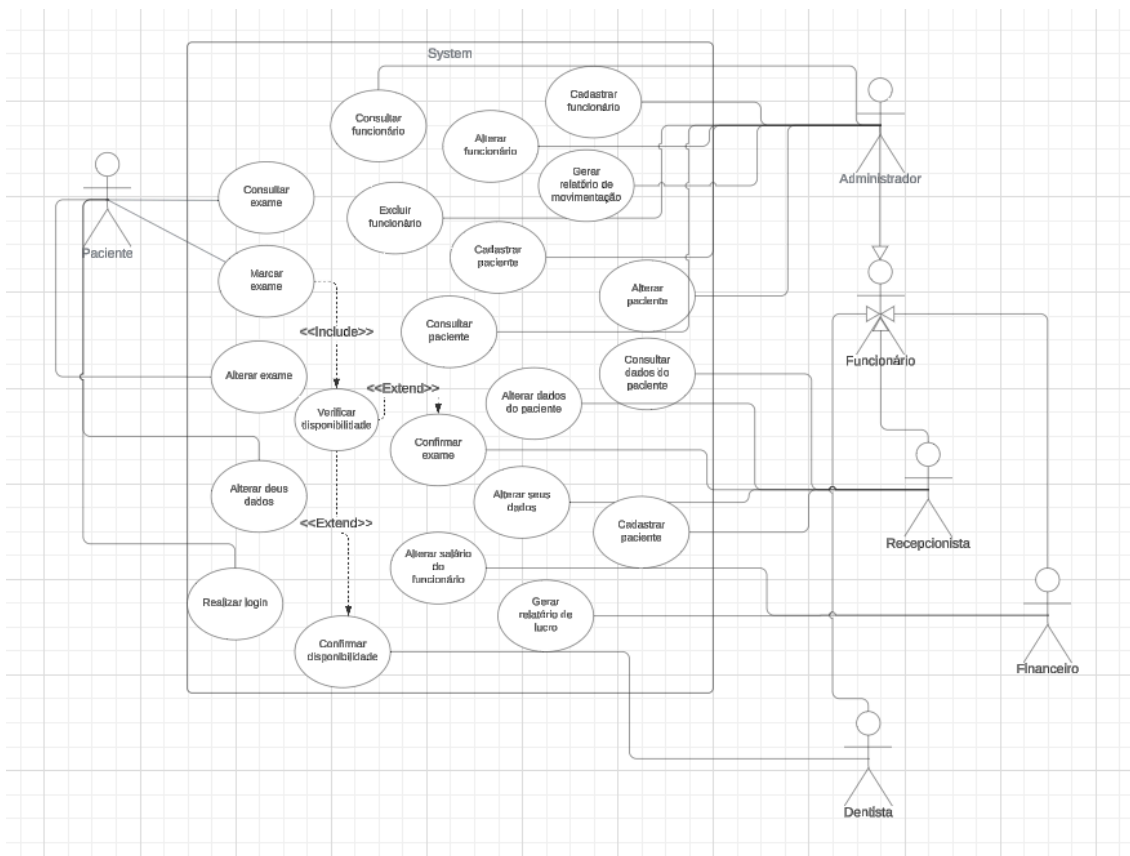


Figura 1. Diagrama de Caso de Uso

3.2.1. Especificações de caso de uso

ECUS01 – CONSULTAR EXAMES

Sumário	Esta opção permite cadastrar, remover, atualizar, deletar e consultar informações sobre o produto.
Ator(es):	Paciente
Pré-condições:	O paciente deve estar autenticado no sistema. O paciente deve ter exames registrados no sistema.

FLUXO PRINCIPAL

1. O paciente acessa a opção "Consultar Exame" no sistema.
2. O sistema exibe uma lista de exames disponíveis para o paciente.
3. O paciente seleciona um exame para visualizar.
4. O sistema exibe os detalhes do exame selecionado.

FLUXO ALTERNATIVO : Exame não encontrado

- a. Caso não haja exames registrados, o sistema exibe uma mensagem de erro e retorna ao menu principal.

Pós-condições:	O paciente visualiza as informações detalhadas sobre o exame selecionado.
-----------------------	---

ECUS02 – MARCAR EXAMES

Sumário	Esta opção permite marcar exames.
Ator(es):	Paciente, Recepcionista
Pré-condições:	O paciente deve estar autenticado no sistema. O paciente deve ter uma consulta associada ao exame.

FLUXO PRINCIPAL

1. O paciente acessa a opção "Marcar Exame".
2. O sistema solicita a data e o horário desejados para o exame.
3. O sistema verifica a disponibilidade do horário e local.
4. Caso disponível, o sistema confirma o agendamento do exame.
5. O paciente recebe uma confirmação com os detalhes do exame.

FLUXO ALTERNATIVO: Horário não disponível

- a. O paciente visualiza as informações detalhadas sobre o exame selecionado.

Pós-condições:	O exame é agendado e registrado no sistema.
-----------------------	---

ECUS03 – ALTERAR DADOS DO PACIENTE

Sumário	Permite que os dados do paciente sejam atualizados no sistema.
Ator(es):	Administrador, Recepcionista
Pré-condições:	O ator deve ter permissão para editar informações dos pacientes.

FLUXO PRINCIPAL

1. O ator seleciona o paciente que deseja atualizar.
2. O sistema exibe as informações atuais do paciente.
3. O ator modifica os dados necessários (nome, endereço, contato, etc.).
4. O sistema salva as alterações e confirma a atualização.

FLUXO ALTERNATIVO

- a. Erro de validação: Caso algum dado inserido não esteja conforme, o sistema exibe uma mensagem de erro e permite a correção.

Pós-condições:	As informações do paciente são atualizadas com sucesso no sistema.
-----------------------	--

ECUS04 – ALTERAR EXAME

Sumário	Permite que um exame agendado seja alterado para outra data ou horário.
Ator(es):	Paciente, Recepcionista
Pré-condições:	O exame deve estar previamente agendado. O ator deve ter permissão para realizar a alteração.

FLUXO PRINCIPAL

1. O ator acessa a opção "Alterar Exame".
2. O sistema exibe os exames agendados para o paciente.
3. O ator seleciona o exame que deseja alterar.
4. O sistema solicita a nova data e horário desejados.
5. O sistema verifica a disponibilidade do novo horário e local.
6. Caso disponível, o sistema confirma a alteração.
7. O ator recebe a confirmação com os detalhes do novo agendamento.

FLUXO ALTERNATIVO

- a) Horário não disponível: Se o novo horário ou local não estiver disponível, o sistema sugere alternativas..

Pós-condições:	O exame é atualizado no sistema com os novos dados de agendamento.
-----------------------	--

ECUS05 – ALTERAR SEUS DADOS	
Sumário	Esta opção permite que o paciente atualize suas próprias informações pessoais no sistema.
Ator(es):	Paciente
Pré-condições:	O paciente deve estar autenticado no sistema.
FLUXO PRINCIPAL	
5. O paciente acessa a opção "Alterar seus Dados". 6. O sistema exibe as informações atuais do paciente. 7. O paciente modifica os dados desejados (telefone, rua, cidade, estado, senha.). 8. O sistema salva as alterações e confirma a atualização.	
FLUXO ALTERNATIVO	
b. Erro de validação: Caso algum dado inserido não esteja conforme, o sistema exibe uma mensagem de erro e permite a correção.	
Pós-condições:	As informações do paciente são atualizadas com sucesso no sistema.

ECUS06 – CADASTRAR FUNCIONÁRIO	
Sumário	Permite que o administrador registre um novo funcionário no sistema..
Ator(es):	Administrador
Pré-condições:	O ator deve ter permissão para adicionar novos funcionários.
FLUXO PRINCIPAL	
1. O administrador acessa a opção "Cadastrar Funcionário". 2. O sistema solicita as informações necessárias do funcionário (nome, cargo, contato, etc.). 3. O administrador preenche os dados e confirma o cadastro. 4. O sistema salva as informações e registra o novo funcionário.	
FLUXO ALTERNATIVO	
a. Erro de validação: Caso algum dado inserido não esteja conforme, o sistema exibe uma mensagem de erro e permite a correção.	
Pós-condições:	O novo funcionário é registrado com sucesso no sistema.

ECUS07 – EXCLUIR FUNCIONÁRIO

Sumário	Esta opção permite que o administrador remova um funcionário do sistema.
Ator(es):	Administrador
Pré-condições:	O funcionário deve estar registrado no sistema. O ator deve ter permissão para excluir funcionários.

FLUXO PRINCIPAL

1. O administrador acessa a opção "Excluir Funcionário".
2. O sistema exibe a lista de funcionários registrados.
3. O administrador seleciona o funcionário que deseja excluir.
4. O sistema solicita confirmação para a exclusão.
5. Após confirmação, o sistema remove o funcionário do registro.

FLUXO ALTERNATIVO

- a. Funcionário com dados vinculados:
Se o funcionário tiver dados vinculados que impeçam a exclusão direta (por exemplo, exames associados), o sistema deve solicitar ações adicionais.

Pós-condições:	O funcionário é removido com sucesso do sistema.
-----------------------	--

ECUS08 – CONSULTAR FUNCIONÁRIO

Sumário	Permite visualizar as informações de um funcionário registrado no sistema.
Ator(es):	Administrador, Recepcionista
Pré-condições:	O funcionário deve estar registrado no sistema.

FLUXO PRINCIPAL

1. O ator acessa a opção "Consultar Funcionário".
2. O sistema exibe a lista de funcionários.
3. O ator seleciona o funcionário desejado.
4. O sistema exibe as informações detalhadas do funcionário selecionado.

FLUXO ALTERNATIVO

a) Funcionário não encontrado: Caso o funcionário não esteja registrado, o sistema exibe uma mensagem de erro.
--

Pós-condições:	As informações do funcionário são exibidas ao ator.
-----------------------	---

ECUS09 – Cadastrar Paciente	
Sumário	Permite que o ator cadastre um novo paciente no sistema.
Ator(es):	Recepcionista, Administrador
Pré-condições:	O ator deve estar autenticado e autorizado.

FLUXO PRINCIPAL
<p>O ator acessa a opção "Cadastrar Paciente".</p> <p>O sistema solicita as informações do paciente (nome, endereço, contato, etc.).</p> <p>O ator preenche as informações necessárias.</p> <p>O sistema verifica a unicidade dos dados (por exemplo, CPF).</p> <p>O sistema registra o paciente e confirma o cadastro.</p>

FLUXO ALTERNATIVO
Dados duplicados: O sistema alerta sobre a duplicidade e solicita a correção.

Pós-condições:	O paciente é cadastrado com sucesso.
-----------------------	--------------------------------------

ECUS10 – Alterar Funcionário	
Sumário	Permite que o administrador altere as informações de um funcionário.
Ator(es):	Administrador
Pré-condições:	<p>O administrador deve estar autenticado no sistema.</p> <p>O funcionário deve estar registrado no sistema.</p>

FLUXO PRINCIPAL
<p>O administrador acessa a opção "Alterar Funcionário".</p> <p>O sistema exibe a lista de funcionários.</p> <p>O administrador seleciona o funcionário a ser alterado.</p> <p>O sistema exibe as informações atuais do funcionário.</p> <p>O administrador altera os dados necessários.</p> <p>O sistema salva as alterações.</p>

FLUXO ALTERNATIVO
Dados inválidos: O sistema exibe uma mensagem de erro e solicita a correção.

Pós-condições:	As informações do funcionário são atualizadas no sistema.
-----------------------	---

ECUS11 – Realizar Login	
Sumário	Permite que os usuários se autentiquem no sistema.
Ator(es):	Todos os usuários (Paciente, Recepcionista, Administrador, Funcionário, Dentista, Financeiro)
Pré-condições:	O usuário deve estar registrado no sistema.

FLUXO PRINCIPAL
<p>O usuário acessa a tela de login.</p> <p>O sistema solicita as credenciais (usuário e senha).</p> <p>O usuário insere as credenciais e confirma.</p> <p>O sistema verifica as credenciais.</p> <p>Se corretas, o sistema concede acesso ao usuário.</p>

FLUXO ALTERNATIVO
Credenciais incorretas: O sistema exibe uma mensagem de erro e permite uma nova tentativa.

Pós-condições:	O usuário é autenticado e tem acesso às funcionalidades permitidas.
-----------------------	---

ECUS12 – Gerar Relatório de Movimentação	
Sumário	Permite que o administrador gere um relatório das movimentações realizadas no sistema.
Ator(es):	Administrador
Pré-condições:	O administrador deve estar autenticado.

FLUXO PRINCIPAL
<p>O administrador acessa a opção "Gerar Relatório de Movimentação".</p> <p>O sistema solicita critérios de geração (período, tipo de movimentação, etc.).</p> <p>O administrador insere os critérios e solicita a geração do relatório.</p> <p>O sistema processa os dados e gera o relatório.</p> <p>O sistema exibe o relatório para o administrador.</p>

FLUXO ALTERNATIVO
Nenhuma movimentação encontrada: O sistema exibe uma mensagem e permite a redefinição dos critérios.

Pós-condições:	O relatório é gerado e exibido para o administrador.
-----------------------	--

ECUS13 – Gerar Relatório de Lucro	
Sumário	Permite que o departamento financeiro gere um relatório detalhado dos lucros.
Ator(es):	Financeiro
Pré-condições:	O ator deve estar autenticado no sistema. Devem existir dados de transações financeiras registradas no sistema.

FLUXO PRINCIPAL

O ator deve estar autenticado no sistema.
Devem existir dados de transações financeiras registradas no sistema.
Fluxo Principal de Eventos:
O financeiro acessa a opção "Gerar Relatório de Lucro".
O sistema solicita critérios de geração (período, tipo de serviço, etc.).
O ator define os critérios e solicita a geração do relatório.
O sistema processa os dados e gera o relatório.
O sistema exibe o relatório detalhado para o ator.

FLUXO ALTERNATIVO

Nenhum dado encontrado: O sistema informa que não há dados para o período solicitado e permite a redefinição dos critérios.

Pós-condições:	O relatório de lucro é gerado e exibido para o ator.
-----------------------	--

ECUS14 – Alterar Paciente	
Sumário	Permite que os dados de um paciente sejam atualizados no sistema.
Ator(es):	Administrador, Recepcionista
Pré-condições:	O ator deve estar autenticado e autorizado. O paciente deve estar registrado no sistema.

FLUXO PRINCIPAL

O ator acessa a opção "Alterar Paciente".
O sistema exibe a lista de pacientes registrados.
O ator seleciona o paciente que deseja alterar.
O sistema exibe as informações atuais do paciente.
O ator faz as alterações necessárias (nome, endereço, telefone, etc.).
O sistema salva as alterações e confirma a atualização.

FLUXO ALTERNATIVO

Dados inválidos: O sistema solicita correções se alguma informação não for válida.

Pós-condições:	As informações do paciente são atualizadas com sucesso.
-----------------------	---

ECUS15 – Consultar Dados do Paciente	
Sumário	Permite que os dados de um paciente específico sejam consultados.
Ator(es):	Administrador, Recepcionista, Funcionário
Pré-condições:	O ator deve estar autenticado e autorizado.

FLUXO PRINCIPAL
<ul style="list-style-type: none"> O ator acessa a opção "Consultar Dados do Paciente". O sistema solicita critérios de busca (nome, CPF, etc.). O ator insere os critérios e inicia a busca. O sistema exibe os dados do paciente encontrado. O ator visualiza as informações do paciente.

FLUXO ALTERNATIVO
Paciente não encontrado: O sistema informa que não há registros para o paciente e permite uma nova busca.

Pós-condições:	O ator visualiza as informações detalhadas do paciente.
-----------------------	---

ECUS16 – Confirmar Exame	
Sumário	Permite que o paciente ou a recepcionista confirme o agendamento de um exame.
Ator(es):	Paciente, Recepcionista
Pré-condições:	O exame deve estar previamente agendado.

FLUXO PRINCIPAL
<ul style="list-style-type: none"> O ator acessa a opção "Confirmar Exame". O sistema exibe os detalhes do exame agendado. O ator verifica e confirma o exame. O sistema registra a confirmação e atualiza o status do exame.

FLUXO ALTERNATIVO
Exame não encontrado: O sistema informa a ausência de exames a serem confirmados.

Pós-condições:	O exame é confirmado e está pronto para ser realizado.
-----------------------	--

ECUS17 – Alterar Salário do Funcionário	
Sumário	Permite que o administrador altere o salário de um funcionário.
Ator(es):	Administrador
Pré-condições:	O administrador deve estar autenticado e autorizado. O funcionário deve estar registrado no sistema.

FLUXO PRINCIPAL
<ul style="list-style-type: none"> O sistema exibe a lista de funcionários. O administrador seleciona o funcionário que deseja alterar o salário. O sistema exibe o salário atual do funcionário. O administrador insere o novo salário. O sistema salva as alterações e confirma a atualização.

FLUXO ALTERNATIVO
Salário inválido: O sistema solicita correções se o valor inserido não for válido.

Pós-condições:	O salário do funcionário é atualizado no sistema.
-----------------------	---

ECUS18 – Verificar Disponibilidade	
Sumário	Permite que a disponibilidade de horários para exames ou consultas seja verificada.
Ator(es):	Paciente, Recepcionista
Pré-condições:	O ator deve estar autenticado.

FLUXO PRINCIPAL
<ul style="list-style-type: none"> O ator acessa a opção "Verificar Disponibilidade". O sistema solicita os critérios (data, horário, tipo de exame). O ator insere os critérios e solicita a verificação. O sistema verifica a disponibilidade conforme os critérios. O sistema exibe os horários disponíveis.

FLUXO ALTERNATIVO
Nenhuma disponibilidade: O sistema informa a indisponibilidade e permite a redefinição dos critérios.

Pós-condições:	O ator visualiza os horários disponíveis para agendamento.
-----------------------	--

ECUS19 – Confirmar Disponibilidade	
Sumário	permite que a disponibilidade de um recurso (sala, equipamento, profissional) seja confirmada para um exame ou consulta.
Ator(es):	Recepcionista, Administrador
Pré-condições:	O ator deve estar autenticado e autorizado.

FLUXO PRINCIPAL
<ul style="list-style-type: none"> O ator acessa a opção "Confirmar Disponibilidade". O sistema exibe as opções de disponibilidade previamente verificadas. O ator seleciona a opção desejada e confirma. O sistema registra a confirmação e bloqueia o horário/recurso para uso.

FLUXO ALTERNATIVO
O sistema informa que o recurso já foi reservado por outro usuário.

Pós-condições:	O recurso é reservado e confirmado para o uso no horário definido.
-----------------------	--

ECUS20 – Consultar Paciente	
Sumário	Permite que o ator consulte informações básicas de um paciente, como nome, CPF e data de nascimento.
Ator(es):	Administrador, Recepcionista, Funcionário
Pré-condições:	O ator deve estar autenticado e autorizado.

FLUXO PRINCIPAL
<ul style="list-style-type: none"> O ator acessa a opção "Consultar Paciente". O sistema solicita critérios de busca (nome, CPF, etc.). O ator insere os critérios e inicia a busca. O sistema exibe uma lista de pacientes que correspondem aos critérios. O ator seleciona um paciente da lista. O sistema exibe as informações básicas do paciente selecionado.

FLUXO ALTERNATIVO
O sistema informa que não há registros que correspondem aos critérios e permite uma nova busca.

Pós-condições:	O ator visualiza as informações básicas do paciente.
-----------------------	--

ECUS21 – Consultar Dados do Paciente	
Sumário	Permite que o ator consulte informações detalhadas de um paciente, como histórico de consultas, exames, e tratamentos.
Ator(es):	Administrador, Recepcionista, Funcionário
Pré-condições:	O ator deve estar autenticado e autorizado. O paciente deve estar registrado no sistema.

FLUXO PRINCIPAL

- O ator acessa a opção "Consultar Dados do Paciente".
- O sistema solicita critérios de busca (nome, CPF, etc.).
- O ator insere os critérios e inicia a busca.
- O sistema exibe a lista de pacientes correspondentes.
- O ator seleciona um paciente da lista.
- O sistema exibe os dados detalhados do paciente, incluindo histórico médico.

FLUXO ALTERNATIVO

- O ator visualiza as informações detalhadas do paciente.

Pós-condições:	O ator visualiza as informações detalhadas do paciente.
-----------------------	---

4. Diagrama de Classes

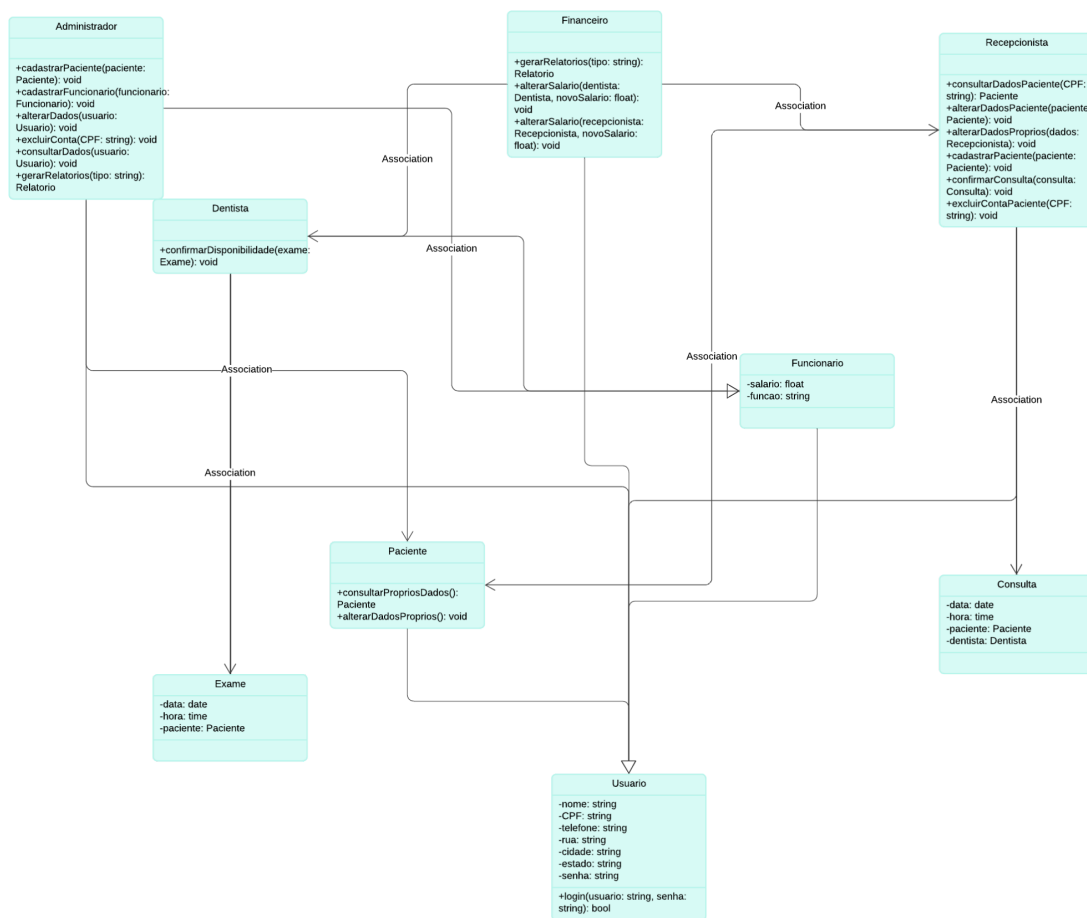
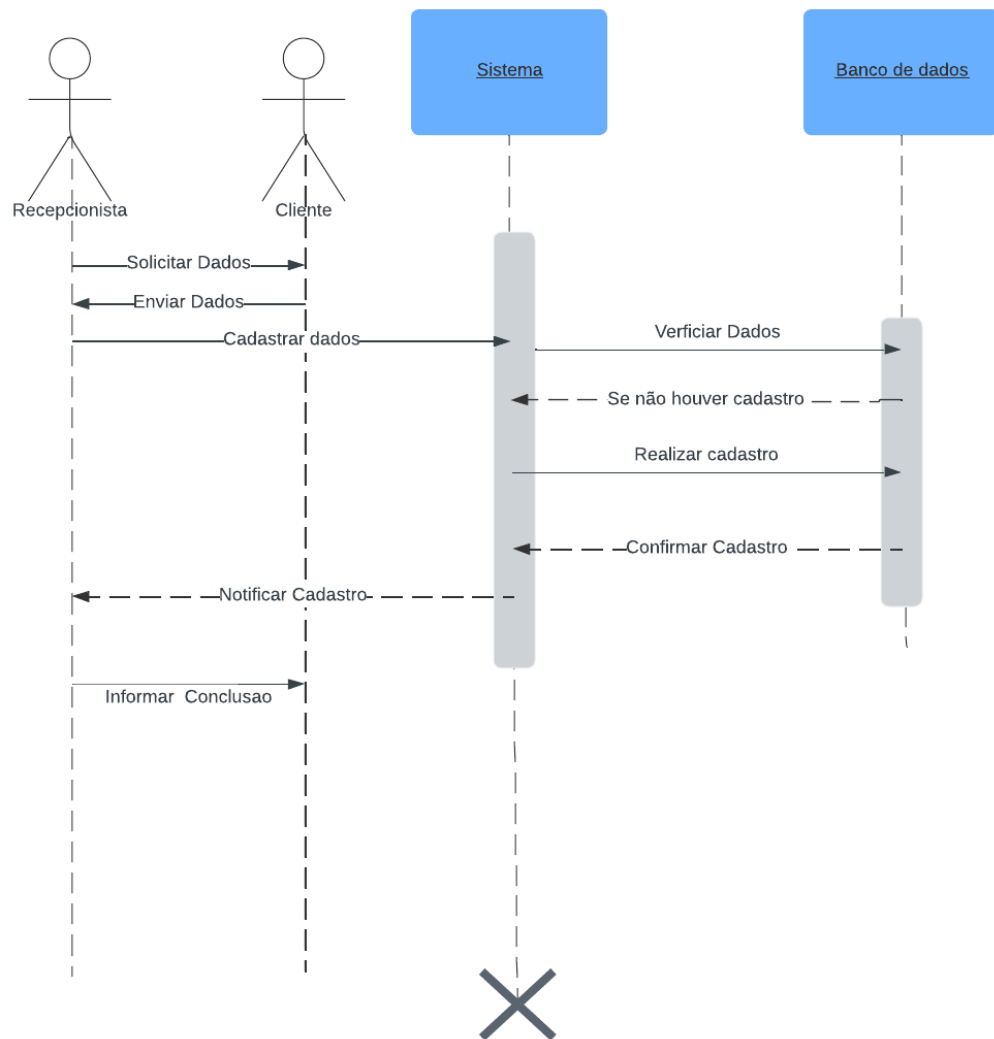


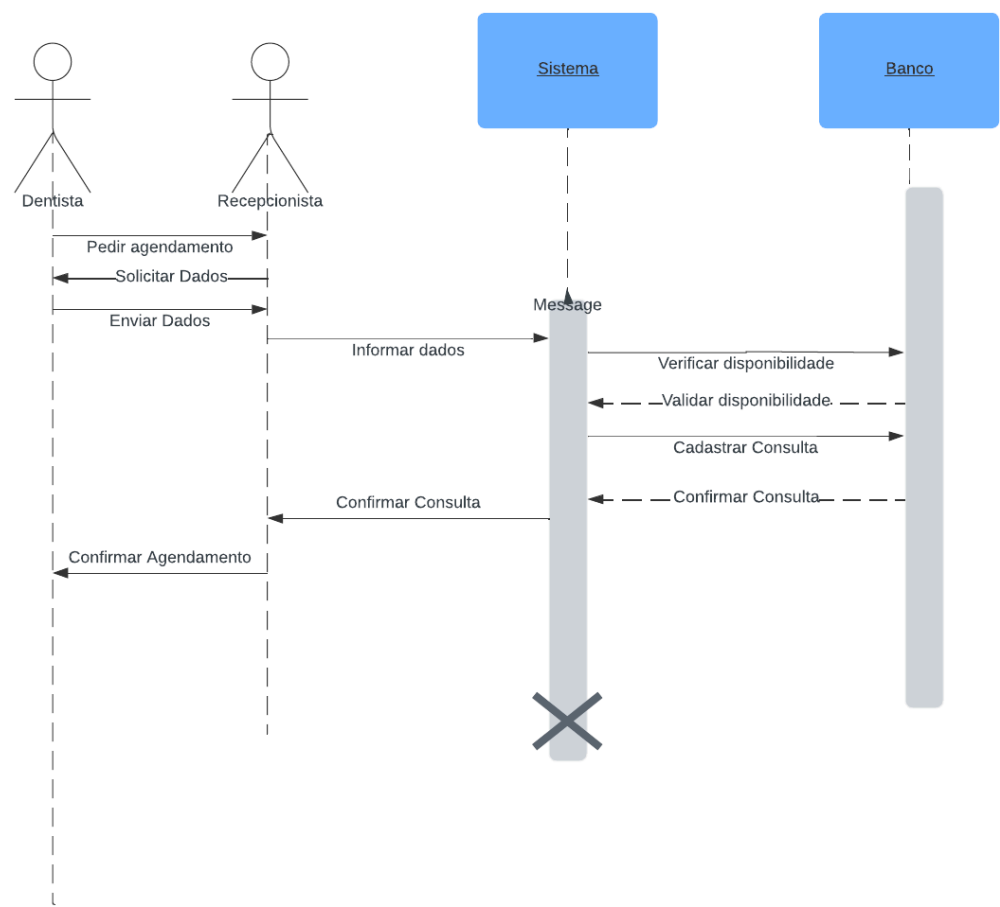
Figura 3. Diagrama de Classes

5. Diagrama de Sequência

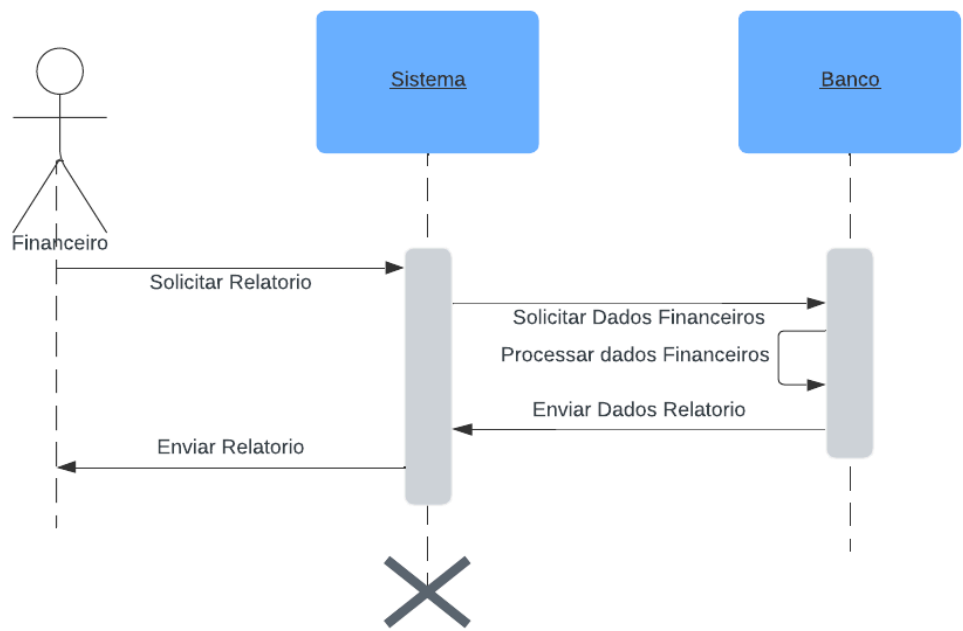
5.1. Cadastrar paciente



5.1.2 Agendar Consulta



5.1.3 Gerar Relatórios Financeiros



6. Decisões Arquiteturais

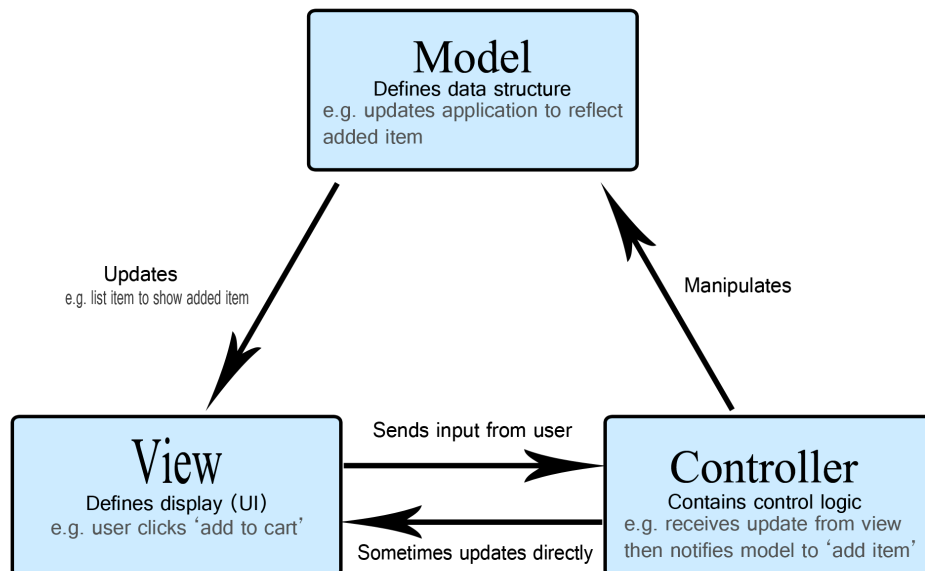
6.1. Decisão 1 - MVC (Model-View-Controller)

Por que escolher MVC?

- Separação de responsabilidades: no contexto da clínica dentária, onde a interface do usuário deve ser clara e acessível tanto para profissionais quanto para pacientes, a separação entre Model, View e Controller permite um desenvolvimento organizado. Por exemplo, a lógica de negócios para agendamentos e a gestão de pacientes é separada da interface de usuário, facilitando a manutenção e as futuras expansões do sistema.
- Facilidade de atualização e manutenção: como as funcionalidades da clínica, como agendamento de consultas e gestão de prontuários, podem evoluir, o MVC permite que as mudanças na interface (View) sejam feitas sem afetar a lógica de negócios (Model) ou a forma como os dados são manipulados (Controller).

Por que é a melhor escolha?

- O MVC é ideal para este sistema porque a clínica precisa de uma interface robusta e fácil de usar, ao mesmo tempo que requer uma lógica de negócios sólida para gerenciar dados de pacientes, consultas e outros serviços. A separação clara facilita o trabalho de diferentes membros da equipe, permitindo um desenvolvimento mais eficiente e ágil.



a

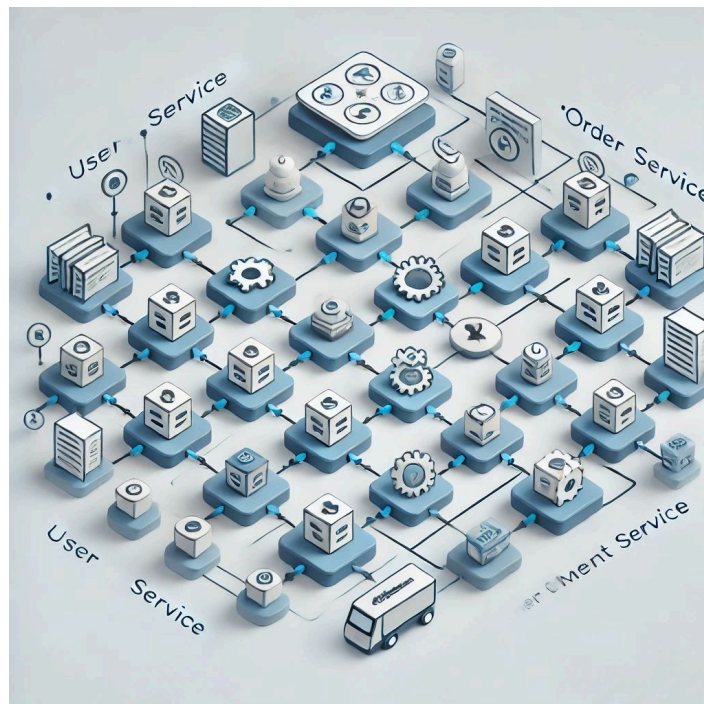
6.2. Decisão 2 - SOA (Service-Oriented Architecture)

Por que escolher SOA?

- Modularidade e escalabilidade: a clínica dentária pode precisar integrar seu sistema com outros serviços, como sistemas de faturamento, seguros de saúde ou fornecedores de materiais. a modularidade da SOA permite que esses serviços sejam adicionados ou modificados sem interromper o funcionamento do sistema principal.
- Interoperabilidade: com SOA, o sistema da clínica pode facilmente se comunicar com sistemas externos, como plataformas de gestão de saúde pública ou fornecedores de serviços terceirizados, utilizando serviços independentes que podem ser desenvolvidos e mantidos separadamente.

Por que é a melhor escolha?

- Para uma clínica que pode expandir seus serviços ou integrar-se a outras plataformas de saúde, SOA oferece a flexibilidade necessária para implementar novos serviços rapidamente e sem riscos de impacto em outros módulos do sistema. Essa arquitetura também facilita o crescimento do sistema à medida que a clínica aumenta seu escopo de operações.



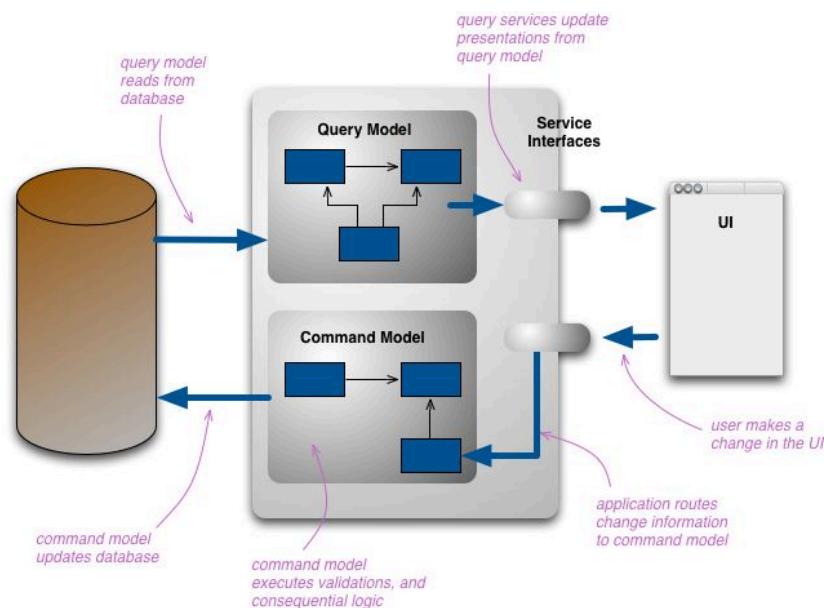
6.3. Decisão 3 - CQRS (Command Query Responsibility Segregation)

Por que escolher CQRS?

- Eficiência em leitura e escrita: em uma clínica dentária, onde as operações de leitura (por exemplo, consulta de histórico médico) são mais frequentes e distintas das operações de escrita (por exemplo, registro de uma nova consulta ou tratamento), o CQRS otimiza o desempenho dividindo essas responsabilidades.
- Escalabilidade: à medida que a clínica cresce, o CQRS permite que as operações de consulta sejam escaladas de forma independente das operações de escrita. Isso é especialmente útil quando há um grande número de usuários acessando o sistema simultaneamente para visualizar ou atualizar informações dos pacientes.

Por que é a melhor escolha?

- Para uma clínica dentária que lida com volumes crescentes de dados e onde a eficiência no acesso à informação é crítica, o CQRS permite um sistema mais ágil e responsivo. Essa arquitetura é ideal para garantir que o sistema possa lidar com demandas de leitura e escrita de maneira otimizada, mantendo o desempenho mesmo em situações de alta carga.



7. Matriz de estilos arquiteturais

A matriz de estilos arquiteturais é essencial para o sistema da clínica dentária, pois permite comparar e avaliar as diferentes arquiteturas (como MVC, SOA e CQRS) em relação aos requisitos específicos da clínica. Ela ajuda a alinhar as escolhas arquiteturais com os objetivos funcionais e não funcionais do sistema, como desempenho e escalabilidade, e mitiga riscos antecipando desafios. Além disso, a matriz facilita a comunicação e documentação das decisões, garantindo clareza e justificativa para todos os envolvidos no projeto.

Estilo Arquitetural/Atributos de Qualidade	Desempenho	Escalabilidade	Manutenibilidade	Flexibilidade	Simplicidade
MVC			F	F	F
SOA		F	F	F	P
CQRS	F	F	P	F	P

7.1 Justificativa

- MVC:
 - Manutenibilidade: favorece, pois separa as responsabilidades entre modelo, visão e controle, facilitando a manutenção do código.
 - Flexibilidade: favorece, pois permite mudanças na interface de usuário ou lógica de negócios sem impactar outras camadas.
 - Simplicidade: favorece, pois é uma arquitetura relativamente simples de entender e implementar.
- SOA:
 - Escalabilidade: favorece, pois permite escalar serviços individualmente.
 - Manutenibilidade: favorece, pois a modularidade dos serviços facilita a manutenção e atualização.
 - Flexibilidade: favorece, pois permite a substituição ou atualização de serviços sem afetar todo o sistema.
 - Simplicidade: prejudica, pois a orquestração e a integração entre serviços podem ser complexas.
- CQRS:
 - Desempenho: favorece, pois separa comandos de consultas, otimizando cada operação para o seu uso específico.
 - Escalabilidade: favorece, pois permite escalar leitura e escrita de maneira independente.

- Manutenibilidade: prejudica, pois aumenta a complexidade do código com dois modelos distintos (comandos e consultas).
- Flexibilidade: favorece, pois possibilita diferentes implementações para leitura e escrita.
- Simplicidade: prejudica, pois adiciona complexidade na separação de responsabilidades.

Por fim, devido a baixa experiência da equipe na linguagem Python, é levado em consideração algo que traga uma familiaridade com a orientação a objetos da sua forma mais simples e direta, por isso a escolha do MVC, onde se pode separar as responsabilidades de cada funcionalidade ou conjunto de funções, além de poder ser mais flexível para novas regras de negócio com baixo impacto, e por ser uma aplicação desktop, favorece a manutenibilidade para novas atualizações e funcionalidades

8. Linha de Produto

Funcionalidade	Start	Premium	Plus
Cadastro de Pacientes	Registro básico de pacientes	Registro com validação de dados	Registro com validação e histórico de alterações
Cadastro de Funcionários	Registro básico de funcionários	Registro com validação de dados	Registro com controle de funções e permissões
Consulta de Dados Cadastrais	Consulta básica de pacientes	Consulta de pacientes e funcionários com filtros	Consulta avançada com relatórios e exportação
Alteração de Dados Cadastrais	Alteração básica para pacientes	Alteração com logs para pacientes e recepcionistas	Alteração com logs completos para todos os usuários
Login e Autenticação	Login e recuperação de senha	Login com autenticação de dois fatores	Login com autenticação de dois fatores e auditoria de sessões
Relatórios e Monitoramento	Relatórios básicos (mensal)	Relatórios semanais e mensais com exportação	Relatórios customizados com filtros avançados
Gestão de Consultas	Confirmação de consultas	Agenda básica para dentistas	Agenda avançada com integração de calendário externo
Gestão Financeira	Controle básico de salários	Controle de salários com relatórios financeiros básicos	Controle de salários com relatórios detalhados e análise financeira
Exclusão de Contas	Exclusão de contas de pacientes	Exclusão de pacientes com verificação de pendências	Exclusão completa com verificação de pendências e histórico

9. Notação F.O.D.A

9.1 Link de acesso:

https://lucid.app/lucidchart/dfb0915b-93fd-4b05-9cc2-e7746395949d/edit?viewport_loc=1517%2C-381%2C2694%2C1138%2C0eqBfotfVGIV&invitationId=inv_36c61f92-e5fa-4713-97f2-9ae2bc29d012

10. Percepções dos alunos:

10.1. Percepção do Aluno Marllon Vinicius:

Participar do desenvolvimento deste software de gerenciamento de uma clínica dentária foi uma experiência enriquecedora para mim como estudante de engenharia de software. Durante o processo, pude perceber a importância da documentação detalhada, da colaboração em equipe e da compreensão das necessidades do cliente. Além disso, aprendi a importância do planejamento cuidadoso e da flexibilidade para lidar com mudanças de requisitos durante o desenvolvimento do projeto.

10.2. Percepção do Italo Dutra:

Mexer em um sistema sem documentação provou o quanto ela é importante e essencial em qualquer projeto de software. A falta de documentação tornou alguns entendimentos de funções simples do sistema bem mais complexos. Aprendi o quanto a documentação é essencial para o software, não só para quem está trabalhando nele no momento, mas também para alguém que vai trabalhar nele posteriormente.

10.3. Percepção da aluna Heloisa Éven:

Uma das primeiras lições que aprendi foi a importância de manter uma boa comunicação dentro da equipe. Trabalhar em grupo foi fundamental para entendermos o escopo do projeto, discutirmos ideias e tomarmos decisões que iriam melhorar o projeto como um todo. Além disso, o desenvolvimento do diagrama de caso de uso foi uma parte crucial do processo. Ele nos permitiu visualizar de forma clara e concisa as interações entre os diferentes usuários e as funcionalidades do sistema, facilitando muito o entendimento dos requisitos e a adição de novas funcionalidades.

10.4. Percepção do Marcel Bleasby:

Devido a dificuldade de entendimento do repositório, havendo muito retrabalho e interpretação não tão clara dos requisitos, a expertise foi essencial em utilizar uma linguagem de programação que fosse comum a todos da equipe, visto que

foi negociado desde o primeiro momento das reuniões pré trabalho.

10.5. Percepção do Lucas Dantas Brandolt:

Apreendi que a documentação é tão importante quanto o código em si, um código bem documentado é um código limpo e de fácil entendimento, tanto para o próprio desenvolvedor, quanto para outros desenvolvedores que forem fazer a manutenção do software posteriormente.

Conclusão

A partir das diversas percepções compartilhadas pelos alunos envolvidos no desenvolvimento do software de gerenciamento de uma clínica dentária, podemos concluir que alguns pontos cruciais emergem como aprendizados valiosos. A importância da documentação detalhada ficou evidente, não apenas para facilitar o entendimento das funcionalidades do sistema, mas também para garantir uma transição suave entre diferentes membros da equipe e facilitar futuras manutenções e atualizações. Além disso, a colaboração em equipe e a comunicação eficaz foram destacadas como fundamentais para o sucesso do projeto, permitindo a compreensão do escopo, a discussão de ideias e a tomada de decisões que beneficiaram o software como um todo. A flexibilidade e a habilidade de adaptação diante de mudanças de requisitos também se mostraram essenciais para lidar com os desafios durante o desenvolvimento. Em resumo, essas percepções destacam a importância não apenas do conhecimento técnico, mas também das habilidades interpessoais e da abordagem metodológica para o sucesso de projetos de engenharia de software.

11. Ata de Reunião

N. da Reunião: 1ª. Reunião	
Data e horário	10/03/2024. Noite (20h)
Tipo	Meet
Participantes:	Heloísa, Ítalo, Lucas, Marllon, Marcel, Marcus
Atividade:	<p>Apresentação do projeto "Clínica Dentária" e definição dos papéis de cada membro da equipe.</p> <p>Discussão sobre o escopo do sistema, suas principais funcionalidades e objetivos.</p> <p>Definição preliminar das técnicas a serem utilizadas para eliciação de requisitos, como entrevistas com potenciais usuários e análise de documentos relacionados à gestão de clínicas odontológicas.</p> <ul style="list-style-type: none">• Distribuição de tarefas iniciais, como a divisão da equipe em responsáveis pelo desenvolvimento e teste, pesquisa sobre sistemas similares, levantamento de requisitos iniciais, elaboração de um cronograma preliminar para o desenvolvimento do projeto.

N. da Reunião: 2ª Reunião	
Data e horário	26/03/2024. Noite (21h)
Tipo	Meet
Participantes:	Heloísa, Ítalo, Lucas, Marllon, Marcel, Marcus
Atividade:	<p>Apresentação dos resultados das pesquisas individuais sobre sistemas similares e requisitos iniciais.</p>

	<p>Discussão sobre os resultados obtidos e identificação de lacunas nos levantamentos de requisitos.</p> <p>Decisão sobre as técnicas adicionais a serem utilizadas para complementar a elicitação de requisitos, como brainstorming.</p> <p>Início da elaboração do documento de Projeto Detalhado de Software, com a definição das seções e subseções que comporão o documento.</p>
--	---

N. da Reunião: 3ª Reunião	
Data e horário	30/03/2024. Noite (18h)
Tipo	Meet
Participantes:	Heloísa, Ítalo, Lucas, Marllon, Marcel, Marcus
Atividade:	<p>Revisão e consolidação das informações levantadas até o momento para inclusão no Projeto Detalhado de Software.</p> <p>Discussão sobre a estrutura e organização do documento, com a definição de responsáveis por cada seção.</p> <p>Definição do ciclo de vida para o desenvolvimento do sistema, optando pelo modelo iterativo e incremental.</p> <p>Ajustes finais no cronograma de desenvolvimento, levando em consideração as etapas de elicitação de requisitos e análise de resultados.</p> <p>Encaminhamentos sobre as próximas atividades, como a elaboração dos diagramas de caso de uso, classes e sequência, e a preparação para a apresentação do projeto.</p>

N. da Reunião: 1ª. Reunião da parte 2	
Data e horário	01/08/2024. Noite (20h)
Tipo	Meet
Participantes:	Heloísa, Ítalo, Lucas, Marllon e Marcel.
Atividade:	<ul style="list-style-type: none"> Os principais estilos arquiteturais foram debatidos e as três decisões foram tomadas, sendo decididas as seguintes arquiteturas: MVC, SOA e CQRS. Logo após, houve a divisão de tarefas que foram definidas desta maneira: <ul style="list-style-type: none"> - Heloísa e Ítalo: preparar as justificativas de decisões das arquiteturas escolhidas. - Lucas e Marllon: analisar o impacto dessas arquiteturas nos atributos de qualidade. - Marcel: preparar a apresentação dessas decisões para a próxima reunião.

N. da Reunião: 2ª Reunião da parte 2	
Data e horário	05/08/2024. Noite (21h)
Tipo	Meet
Participantes:	Heloísa, Ítalo, Lucas, Marllon e Marcel
Atividade:	<p>A equipe discutiu e revisou o impacto de cada estilo arquitetural nos atributos de qualidade, sendo gerada a matriz de estilos arquiteturais.</p> <p>Divisão de tarefas:</p> <ul style="list-style-type: none"> - Heloísa e Ítalo: documentar a matriz e preparar uma explicação sobre sua importância. - Lucas e Marllon: começar a desenvolver a base da Linha de Produtos de Software (LPS) considerando as arquiteturas.

	<ul style="list-style-type: none"> - Marcel: Revisar a documentação e preparar feedback para a próxima reunião.
--	--

N. da Reunião: 3ª Reunião da parte 2	
Data e horário	10/08/2024. Noite (19h30)
Tipo	Meet
Participantes:	Heloísa, Ítalo, Lucas, Marllon e Marcel.
Atividade:	<p>Discussão sobre a criação de uma LPS utilizando as arquiteturas escolhidas, com foco em adicionar novas features; Identificação das features; início da montagem do mapa de produto e do modelo F.O.D.A.</p> <p>Divisão de tarefas:</p> <p>-Heloísa e Ítalo: finalizar modelo F.O.D.A e o mapa de produto.</p> <p>Lucas e Marllon: desenvolver as features identificadas para a LPS.</p> <p>Marcel: preparar a documentação e uma prévia da apresentação final.</p>

N. da Reunião: 4ª Reunião da parte 2	
Data e horário	17/08/2024. Noite (20h30)
Tipo	Meet
Participantes:	Heloísa, Ítalo, Lucas, Marllon e Marcel.
Atividade:	<p>Foram apresentados o modelo F.O.D.A. e o mapa de produto, discutindo a configuração das features e a flexibilidade na criação de produtos.</p>

	<p>Pequenos ajustes foram feitos no mapa de produto para refletir melhor as dependências entre as features.</p> <p>O documento foi finalizado pela equipe completa.</p>
--	---

12. Referências

ROSCELLE, Laura. Projeto-Clínica: Projeto de desenvolvimento de um sistema CRUD de clínica dentária em Python com interface gráfica. [S. l.], 2023. Disponível em: <https://github.com/Laura-rosce/Projeto-Clinica-dent-ria>. Acesso em: 4 abr. 2024.