

## TÍTULO DO TRABALHO: Sys Odonto

### Equipe:

Cauã Grigolatto Domingos – AQ3022323 – [caua.grigolatto@aluno.ifsp.edu.br](mailto:caua.grigolatto@aluno.ifsp.edu.br) - Líder

Gabriel de Pauli Santos – AQ302282X – gabriel.pauli@aluno.ifsp.edu.br

Gabriel Dellatore Ezequiel - AQ3022561 - dellatore.gabriel@aluno.ifsp.edu.br

Gabriel Ventura Pires – AQ3023672 – g.ventura@aluno.ifsp.edu.br

João Pedro da Silva Vieira - AQ3022366 - vieira.joao1@aluno.ifsp.edu.br

**Repositório GitHub<sup>1</sup>:** [https://github.com/projetologicas/PRSI\\_2025\\_SysOdonto](https://github.com/projetologicas/PRSI_2025_SysOdonto)

**Professora Orientadora:** Janaina Cintra Abib [janaina@ifsp.edu.br](mailto:janaina@ifsp.edu.br)

### 1. INTRODUÇÃO

No competitivo mercado odontológico brasileiro, a gestão eficiente de pacientes tornou-se um fator crucial para o sucesso das clínicas. Atualmente, um desafio comum a muitas clínicas é a ausência de um sistema seguro para o gerenciamento de dados e agendamentos de consultas, uma lacuna que resulta em falhas de comunicação, perda de informações vitais e altas taxas de não comparecimento, que impactam diretamente o lucro.

Para resolver este problema, o Sys Odonto propõe transformar a rotina da clínica, ao automatizar o envio de lembretes via WhatsApp e centralizar os dados dos pacientes em nuvem com acesso facilitado. Nossa software elimina processos manuais, permitindo que a equipe dedique seu tempo integralmente ao atendimento e cuidado com o paciente.

### 2. OBJETIVO

Desenvolver um sistema, o Sys Odonto, para centralizar e otimizar os processos administrativos dos pacientes, como armazenamento e consulta de dados, e automatizar o envio de lembretes antecipados de consultas via WhatsApp. Com isso, o sistema foca em substituir os

métodos manuais existentes por ferramentas digitais eficazes, permitindo que a equipe priorize o atendimento e o acolhimento dos pacientes.

### 3. PÚBLICO ALVO

O Sys Odonto é projetado para atender às necessidades de clínicas odontológicas de pequeno a médio porte, que buscam centralizar e otimizar o armazenamento dos dados de pacientes, eliminando o esforço manual para lembrá-los de suas consultas. Além disso, o Sys Odonto é voltado para usuários com pouca experiência em tecnologia, como secretários(as) e dentistas, que procuram uma ferramenta de uso natural e intuitivo.

### 4. METODOLOGIA

Para o desenvolvimento do sistema Sys Odonto, será adotada uma abordagem ágil, visando garantir a flexibilidade, a adaptação contínua às necessidades da clínica e a entrega de valor em ciclos curtos. A escolha desta metodologia se justifica pela necessidade de validação constante com a stakeholder, Dra. Nicole, assegurando que o produto final resolva de forma eficaz as dores identificadas em seu ambiente de trabalho.

A fase inicial do projeto consistiu em um levantamento aprofundado de requisitos, que teve como ponto de partida uma entrevista detalhada com a Dra. Nicole da Clínica Olenscki. Essa conversa foi fundamental para mapear os processos manuais vigentes, identificar os principais gargalos operacionais, como a dependência de agendas físicas e a ausência de lembretes automáticos de consulta, e compreender as necessidades do nicho. A partir dessa análise, os requisitos foram classificados em funcionais, que descrevem as ações que o sistema deve executar, e não funcionais, que definem os critérios de qualidade como segurança, desempenho e usabilidade, pensando sempre em um público-alvo com pouca experiência em tecnologia.

O desenvolvimento será organizado em ciclos iterativos de duas semanas, seguindo os princípios ágeis para promover transparência e melhoria contínua. Cada ciclo incluirá um planejamento para definir as prioridades, sincronizar a equipe e, ao final, uma sessão de revisão. Esta revisão será o momento chave para a validação, onde as funcionalidades desenvolvidas serão apresentadas diretamente à Dra. Nicole para coleta de feedback, garantindo que o sistema evolua de acordo com suas expectativas. O ciclo se encerra com uma retrospectiva, focada em aprimorar os processos de trabalho da equipe.

Para a construção da solução, será utilizado um conjunto de tecnologias modernas. A lógica de negócios e os serviços do sistema (backend) serão desenvolvidos em Java (JDK 21) e Spring Boot, enquanto a interface do usuário (frontend) será criada com JavaScript/TypeScript e a biblioteca React, focando em uma experiência interativa e responsiva. O armazenamento e a persistência dos dados serão gerenciados pelo banco de dados Firestore. Não há um ambiente de desenvolvimento integrado (IDE) padrão, porém, todo o código-fonte será versionado através do Git e hospedado em um repositório no GitHub. O gerenciamento do projeto e a visualização do

fluxo de trabalho serão realizados na ferramenta Trello, por meio de um quadro Kanban que permitirá o acompanhamento claro do progresso de todas as tarefas.

## 5. RESULTADOS ESPERADOS

Espera-se que, com a implementação do Sys Odonto, clínicas odontológicas de pequeno e médio porte experimentem uma transformação significativa em sua gestão administrativa e na relação com seus pacientes. O sistema visa proporcionar ganhos concretos em termos de eficiência operacional, qualidade do atendimento e aumento da produtividade das equipes.

Um dos principais resultados almejados é a redução expressiva das taxas de absenteísmo, melhorando o aproveitamento da agenda dos profissionais e, consequentemente, aumentando a receita da clínica, impactando diretamente e positivamente no faturamento da empresa.

Além disso, espera-se uma melhoria na organização e segurança dos dados clínicos e cadastrais dos pacientes, minimizando riscos de perda de dados e possibilitando uma gestão mais precisa do histórico clínico dos pacientes.

Outro resultado esperado é a diminuição da carga de trabalho manual das equipes administrativas, que poderão direcionar seus esforços a atividades mais estratégicas e de contato humano, como o acolhimento ao paciente e o suporte às atividades clínicas, acarretando também, na diminuição de falhas humanas em geral.

Por fim, o projeto visa gerar impactos positivos de longo prazo, como o fortalecimento da relação entre clínica e paciente, a consolidação da imagem da clínica como um ambiente organizado e por fim, a parte financeira da empresa minimizando prejuízos.

## 6. REQUISITOS FUNCIONAIS

Tabela 1. Requisitos Funcionais

Requisito Funcional	Descrição
<b>RF1. Consultar conta por email</b>	<b>Entrada:</b> Email. <b>Processamento:</b> O sistema deve verificar a existência de uma conta no banco de dados através do email válido fornecido. <b>Saída:</b> <ul style="list-style-type: none"><li>• <b>Sucesso:</b> Mensagem de conta existe ou não existe</li></ul>

	<ul style="list-style-type: none"> <li><b>Erro:</b> Mensagem de erro</li> </ul> <p><b>Pré-requisito:</b> -</p>
<b>RF2. Enviar email de confirmação</b>	<p><b>Entrada:</b> Email.</p> <p><b>Processamento:</b> O sistema deve enviar uma mensagem de confirmação ao email fornecido para validar um usuário.</p> <p><b>Saída:</b></p> <ul style="list-style-type: none"> <li><b>Sucesso:</b> Enviar mensagem</li> <li><b>Erro:</b> Mensagem de erro e solicitar reenvio do email</li> </ul> <p><b>Pré-requisito:</b> -</p>
<b>RF3. Cadastrar contas</b>	<p><b>Entrada:</b> Nome da conta, email, senha, confirmação de senha, foto (opcional).</p> <p><b>Processamento:</b> O sistema deve enviar uma mensagem de confirmação no email da conta e então cadastrá-la no banco.</p> <p><b>Saída:</b></p> <ul style="list-style-type: none"> <li>Sucesso: Mensagem de sucesso</li> <li>Erro: Mensagem de erro</li> </ul> <p><b>Pré-requisito:</b> RF1, RF2.</p>
<b>RF4. Consultar conta por credenciais</b>	<p><b>Entrada:</b> Email, senha.</p> <p><b>Processamento:</b> O sistema deve permitir a consulta a contas no banco de dados através do email e senha.</p> <p><b>Saída:</b></p> <ul style="list-style-type: none"> <li>Sucesso: Retorno da conta selecionada.</li> <li>Erro: Mensagem de erro</li> </ul> <p><b>Pré-requisito:</b> -</p>
<b>RF5. Criar sessão</b>	<p><b>Entrada:</b> Email.</p> <p><b>Processamento:</b> O sistema deve criar uma sessão ativa com o email fornecido.</p> <p><b>Saída:</b></p> <ul style="list-style-type: none"> <li>Sucesso: Mensagem de sucesso</li> <li>Erro: Mensagem de erro</li> </ul> <p><b>Pré-requisito:</b> -</p>

<b>RF6. Login no sistema</b>	<p><b>Entrada:</b> Email, senha.</p> <p><b>Processamento:</b> O sistema deve permitir acesso ao sistema, criando uma nova sessão, por meio da autenticação com email e senha.</p> <p><b>Saída:</b></p> <ul style="list-style-type: none"> <li>● Sucesso: Mensagem de sucesso</li> <li>● Erro: Mensagem de erro e solicitar entrada novamente</li> </ul> <p><b>Pré-requisito:</b> RF4, RF5.</p>
<b>RF7. Invalidar sessão</b>	<p><b>Entrada:</b> Sessão.</p> <p><b>Processamento:</b> O sistema deve invalidar a sessão fornecida.</p> <p><b>Saída:</b></p> <ul style="list-style-type: none"> <li>● Sucesso: Mensagem de sucesso</li> <li>● Erro: Mensagem de erro</li> </ul> <p><b>Pré-requisito:</b> -</p>
<b>RF8. Logout</b>	<p><b>Entrada:</b> Escolher &lt;sair&gt; do sistema.</p> <p><b>Processamento:</b> O sistema deve permitir ao usuário encerrar sua sessão e sair da aplicação.</p> <p><b>Saída:</b></p> <ul style="list-style-type: none"> <li>● Sucesso: Mensagem de sucesso</li> <li>● Erro: Mensagem de erro</li> </ul> <p><b>Pré-requisito:</b> RF7.</p>
<b>RF9. Validar CPF</b>	<p><b>Entrada:</b> CPF.</p> <p><b>Processamento:</b> O sistema deve validar o formato do CPF inserido.</p> <p><b>Saída:</b></p> <ul style="list-style-type: none"> <li>● Sucesso: Mensagem de CPF válido/inválido</li> <li>● Erro: Mensagem de erro</li> </ul> <p><b>Pré-requisito:</b> -</p>
<b>RF10. Verificar existência de paciente</b>	<p><b>Entrada:</b> CPF.</p> <p><b>Processamento:</b> O sistema deve checar a existência de pacientes cadastrados com os dados fornecidos.</p> <p><b>Saída:</b></p> <ul style="list-style-type: none"> <li>● Sucesso: Mensagem de paciente existe/não existe</li> <li>● Erro: Mensagem de erro</li> </ul> <p><b>Pré-requisito:</b> RF9.</p>

<b>RF11. Validar telefone</b>	<p><b>Entrada:</b> Telefone.</p> <p><b>Processamento:</b> O sistema deve validar o formato do telefone inserido.</p> <p><b>Saída:</b></p> <ul style="list-style-type: none"> <li>• Sucesso: Mensagem de telefone válido/inválido</li> <li>• Erro: Mensagem de erro</li> </ul> <p><b>Pré-requisito:</b> -</p>
<b>RF12. Checar cadastro de telefone</b>	<p><b>Entrada:</b> Telefone.</p> <p><b>Processamento:</b> O sistema deve checar a existência do telefone inserido.</p> <p><b>Saída:</b></p> <ul style="list-style-type: none"> <li>• Sucesso: Mensagem de telefone não cadastrado/cadastrado</li> <li>• Erro: Mensagem de erro</li> </ul> <p><b>Pré-requisito:</b> RF11.</p>
<b>RF13. Cadastro de paciente</b>	<p><b>Entrada:</b> Código, foto (opcional), nome completo, CPF, número de telefone, data de nascimento, data de início do tratamento (opcional), observações (opcional), odontograma (opcional).</p> <p><b>Processamento:</b> O sistema deve gerar um código único para o paciente e cadastrá-lo no banco com os dados fornecidos.</p> <p><b>Saída:</b></p> <ul style="list-style-type: none"> <li>• Sucesso: Mensagem de paciente cadastrado</li> <li>• Erro: Mensagem de erro</li> </ul> <p><b>Pré-requisito:</b> RF10, RF12.</p>
<b>RF14. Consulta de paciente</b>	<p><b>Entrada:</b> Código, nome do paciente, CPF (apenas número), telefone (apenas número), data de nascimento, data do início de tratamento, observações.</p> <p><b>Processamento:</b> O sistema deve consultar um paciente com os dados fornecidos.</p> <p><b>Saída:</b></p> <ul style="list-style-type: none"> <li>• Sucesso: Retorno dos dados dos pacientes</li> <li>• Erro: Mensagem de erro</li> </ul> <p><b>Pré-requisito:</b> -</p>

<b>RF15. Alterar dados do paciente</b>	<p><b>Entrada:</b> Foto (opcional), nome completo, CPF, número de telefone, data de nascimento, data de início do tratamento (opcional), observações (opcional), odontograma (opcional).</p> <p><b>Processamento:</b> O sistema deve sobreescrivar as informações do paciente editado.</p> <p><b>Saída:</b></p> <ul style="list-style-type: none"> <li>• Sucesso: Paciente editado</li> <li>• Erro: Mensagem de erro</li> </ul> <p><b>Pré-requisito:</b> RF14.</p>
<b>RF16. Obter consulta</b>	<p><b>Entrada:</b> Dia, horário, paciente</p> <p><b>Processamento:</b> O sistema deve retornar a consulta conforme os dados fornecidos.</p> <p><b>Saída:</b></p> <ul style="list-style-type: none"> <li>• Sucesso: Retorno da consulta</li> <li>• Erro: Mensagem de erro</li> </ul> <p><b>Pré-requisito:</b> -</p>
<b>RF17. Enviar lembretes automáticos via WhatsApp</b>	<p><b>Entrada:</b> Telefone, mensagem.</p> <p><b>Processamento:</b> O sistema deve enviar uma mensagem de lembrete ao telefone informado.</p> <p><b>Saída:</b></p> <ul style="list-style-type: none"> <li>• Sucesso: Feedback de mensagem enviada</li> <li>• Erro: Mensagem de erro</li> </ul> <p><b>Pré-requisito:</b> -</p>
<b>RF18. Cadastrar consulta</b>	<p><b>Entrada:</b> Dia, horário, paciente, flag de envio de lembretes via WhatsApp, observações (opcional)</p> <p><b>Processamento:</b> O sistema deve verificar conflitos de horários e cadastrar uma nova consulta no banco com os dados fornecidos.</p> <p><b>Saída:</b></p> <ul style="list-style-type: none"> <li>• Sucesso: Mensagem de consulta cadastrada</li> <li>• Erro: Mensagem de erro</li> </ul> <p><b>Pré-requisito:</b> RF16, RF17.</p>

<b>RF19. Alterar consulta</b>	<p><b>Entrada:</b> Dia, horário, paciente, flag de envio de lembretes via WhatsApp, observações (opcional)</p> <p><b>Processamento:</b> O sistema deve sobreescriver as informações da consulta no banco.</p> <p><b>Saída:</b></p> <ul style="list-style-type: none"> <li>• Sucesso: Mensagem de consulta atualizada</li> <li>• Erro: Mensagem de erro</li> </ul> <p><b>Pré-requisito:</b> RF16, RF17.</p>
<b>RF20. Deletar consulta</b>	<p><b>Entrada:</b> Dia, horário</p> <p><b>Processamento:</b> O sistema deve deletar do banco de dados a consulta.</p> <p><b>Saída:</b></p> <ul style="list-style-type: none"> <li>• Sucesso: Mensagem de consulta deletada</li> <li>• Erro: Mensagem de erro</li> </ul> <p><b>Pré-requisito:</b> RF16.</p>
<b>RF21. Alterar senha do usuário</b>	<p><b>Entrada:</b> Nova senha, confirmação nova senha.</p> <p><b>Processamento:</b> O sistema deve atualizar a senha do usuário.</p> <p><b>Saída:</b></p> <ul style="list-style-type: none"> <li>• <b>Sucesso:</b> Mensagem de sucesso</li> <li>• <b>Erro:</b> Mensagem de erro</li> </ul> <p><b>Pré-requisito:</b> RF3.</p>
<b>RF22. Importar dados de pacientes</b>	<p><b>Entrada:</b> CSV, JSON.</p> <p><b>Processamento:</b> O sistema deve importar dados de pacientes oriundos das entradas listadas.</p> <p><b>Saída:</b></p> <ul style="list-style-type: none"> <li>• <b>Sucesso:</b> mensagem de sucesso</li> <li>• <b>Erro:</b> mensagem de erro</li> </ul> <p><b>Pré-requisito:</b> RF13.</p>

## 7. REQUISITOS NÃO FUNCIONAIS

Tabela 2. Requisitos Não Funcionais

Requisito Não Funcional	Descrição
-------------------------	-----------

<b>RNF1. Portabilidade</b>	<ol style="list-style-type: none"> <li>1. O sistema deve ser compatível com os navegadores Chrome, Firefox e Safari.</li> <li>2. O sistema deve possuir uma arquitetura independente de plataforma, permitindo o sistema ser executado em diferentes sistemas operacionais.</li> </ol>
<b>RNF2. Usabilidade</b>	<ol style="list-style-type: none"> <li>1. O sistema deve possuir uma interface intuitiva e de fácil uso, com uma boa navegação.</li> <li>2. O sistema deve ter uma curva de aprendizado curta para os profissionais na área odontológica.</li> </ol>
<b>RNF3. Confiabilidade</b>	<ol style="list-style-type: none"> <li>1. O sistema deve demonstrar uma consistência nas suas respostas e em sua operação.</li> <li>2. O sistema deve implementar mecanismos de tratamento de erros e exceções, exibindo mensagens de erro claras e informativas para o usuário.</li> </ol>
<b>RNF4. Disponibilidade</b>	<ol style="list-style-type: none"> <li>1. A disponibilidade do sistema deve ser de 99,9% durante todo o horário comercial, permitindo inatividade planejada ou não planejada.</li> <li>2. Deve ser feito um plano de contingência para cenários de inatividade.</li> </ol>
<b>RNF5. Integração com API de terceiros</b>	<ol style="list-style-type: none"> <li>1. O sistema deve ser desenvolvido de uma maneira que propicie a comunicação com o Firebase, do Google, e a API de mensageria do WhapiCloud.</li> </ol>

## 8. ARQUITETURA E MODELOS DO SISTEMA

O sistema Sys Odonto será edificado sobre uma arquitetura em camadas, distribuída de forma que um lado (o que o usuário vê) se comunica com outro lado (o que processa os dados e regras), conforme o padrão Cliente-Servidor. Esta escolha arquitetônica foi feita para assegurar uma separação nítida das responsabilidades de cada parte do sistema, o que facilita sobremaneira a sua manutenção e permite que ele possa crescer e se adaptar a futuras demandas sem grandes dificuldades. A estrutura é composta, então, por três grandes seções principais: a Camada de Apresentação, que é a interface visível ao usuário; a Camada de Aplicação, onde reside toda a inteligência do negócio; e a Camada de Dados, responsável pelo armazenamento seguro das informações.

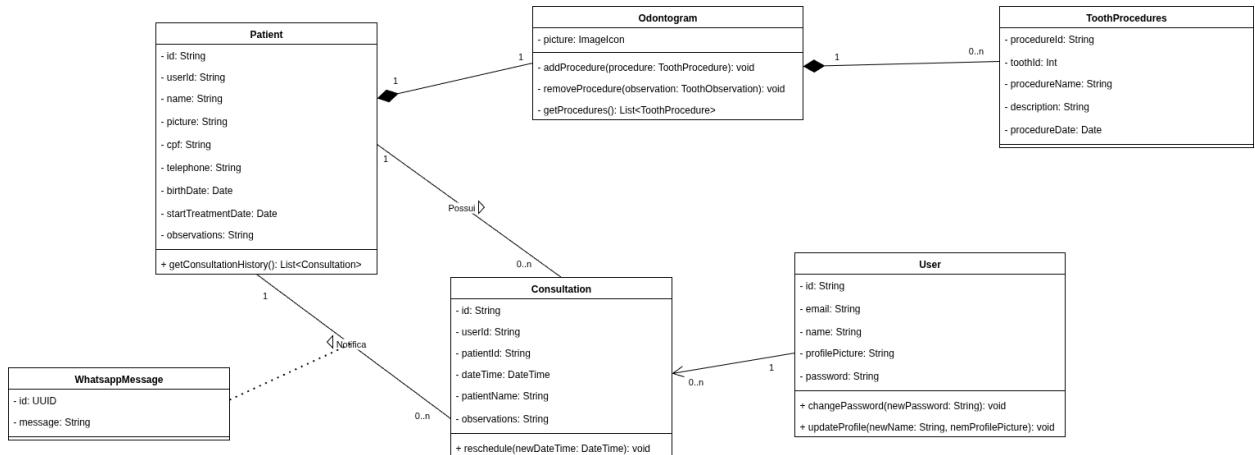
Iniciando pela Camada de Apresentação, esta é a porção do sistema com a qual o utilizador, no caso a Dra. Nicole, interagirá diretamente. Ela será desenvolvida como um aplicativo web moderno, empregando tecnologias como JavaScript e a biblioteca React. Sua incumbência primordial é desenhar todas as telas do sistema, desde o painel inicial e a agenda até os formulários de cadastro, garantindo que sejam intuitivas e se adaptem a diferentes tamanhos de tela. Além disso, esta camada tem a função de capturar as ações do usuário, como cliques em botões ou o preenchimento de dados, e enviar essas informações para a camada seguinte, o "cérebro" do sistema, por meio de comunicações específicas (a chamada API REST).

Avançando para a Camada de Aplicação, encontramos o coração do sistema, o lugar onde todas as regras de negócio e a lógica de processamento são executadas. Esta camada será construída em Java e é composta por alguns elementos chave. Primeiramente, temos os Controllers, que agem como porteiros, recebendo as solicitações da interface do usuário, verificando se os dados estão corretos e as encaminhando para os componentes adequados. Em seguida, vêm os Services, que são os verdadeiros centros de inteligência. Por exemplo, haverá um Serviço de Agendamento, cuja missão será verificar se há horários disponíveis, validar todos os dados de uma nova consulta e coordenar todo o processo de salvamento das informações, conversando para isso com a camada de dados. Integrando-se a tudo isso, está o Domain, que é a essência desta camada. Ele é formado pelas classes desenhadas no diagrama, como a classe User, Consultation, Patient, Odontogram e ToothProcedure. Essas classes funcionam como a representação digital das entidades da clínica, guardando os dados e definindo os comportamentos essenciais do sistema, assegurando assim que as regras de negócio sejam seguidas à risca.

Por fim, a Camada de Dados é a guardiã das informações, encarregando-se de toda a interação com o local onde os dados são armazenados, sem que as outras partes do sistema precisem saber os detalhes técnicos desse armazenamento. Utilizaremos o Google Firestore, um sistema de banco de dados que organiza as informações em documentos, ao invés de tabelas tradicionais. Para facilitar a comunicação, esta camada contará com os Repositories, que são classes específicas (como um PatientRepository) que atuam como intermediários. Os Serviços da camada de Aplicação usarão esses Reppositórios para solicitar que os dados sejam buscados, salvos, alterados ou apagados, sem se preocupar com a maneira exata como o Firestore realiza essas operações.

Para ilustrar o fluxo de informações, pense no agendamento de uma nova consulta: o usuário interage com o aplicativo na Camada de Apresentação. O aplicativo envia a solicitação para o Controller no Backend. Este Controlador aciona o Serviço de Agendamento, que por sua

vez utiliza o Repositório de Consultas para registrar os dados no Firestore. A resposta do sucesso ou de algum eventual erro, então, percorre o caminho inverso, sendo exibida novamente na interface do usuário.



O diagrama de classes que elaboramos estabelece uma rede de relações essenciais que refletem a complexidade e a interconexão das entidades do negócio de uma clínica odontológica. Cada ligação entre as classes define como elas interagem e se complementam para formar o todo do sistema Sys Odonto.

Começando pela entidade **User**, que representa o dentista que opera o sistema, observamos que ele possui uma relação direta com a consulta (**Consultation**). Esta conexão é do tipo um para muitos, o que significa que um único **User** pode ter agendado e gerenciado múltiplas consultas, mas cada consulta individual está sempre associada a apenas um usuário específico. Essa modelagem é crucial para o propósito de um sistema individual, onde o usuário que está logado é, por definição, o dentista responsável por todas as ações e registros.

Passando para a entidade **Patient**, que é o cliente da clínica, ela mantém uma relação fundamental com **Odontogram**. Esta ligação é de composição, um tipo de relacionamento muito forte, caracterizado por uma cardinalidade de um para um. Isso implica que um paciente possui exatamente um odontograma, e este odontograma não pode existir independentemente do paciente ao qual pertence. Caso o registro de um paciente seja removido do sistema, seu odontograma será consequentemente apagado junto, reforçando a ideia de que o odontograma é uma parte intrínseca do paciente.

Dando continuidade a essa cadeia de informações clínicas, o Odontogram, por sua vez, está relacionado à entidade de procedimento dentários (ou ToothProcedures em sua representação). Esta também é uma relação de composição, mas com uma cardinalidade de um para muitos. Isso quer dizer que um odontograma pode conter nenhum ou vários procedimentos, e cada procedimento dentário é uma parte integrante de um odontograma específico. Esta estrutura permite registrar detalhadamente cada intervenção realizada nos dentes do paciente, mantendo um histórico preciso dentro do prontuário visual.

Considerando o fluxo de atendimento, a Patient também se conecta a Consultation. Aqui, estabeleceu-se uma relação nomeada “possui” no diagrama. Isso sugere que a relação entre Patient e Consultation pode ser vista como um para muitos, implicando que um paciente poderia estar ligado a nenhuma ou várias consultas, e uma consulta atrelada a um único paciente.

Finalmente, há uma relação importante entre Consultation e Patient, dada pela classe intermediária WhatsappMessage. Esta ligação é nomeada “notifica”, indicando um relacionamento mais dinâmico ou de evento. Assim, um paciente pode ser notificado por zero ou  $n$  consultas, mas a consulta notifica apenas um paciente. A classe intermediária representa a mensagem de Whatsapp para o paciente, as quais seriam, por exemplo, os lembretes ou as confirmações de agendamento. Este mecanismo é fundamental para automatizar a comunicação e reduzir as faltas, sendo uma funcionalidade chave do Sys Odonto.

Em síntese, as relações desenhadas neste diagrama de classes criam um modelo interconectado que sustenta todas as funcionalidades do Sys Odonto, desde a gestão básica dos usuários e pacientes até o detalhamento dos procedimentos e a automação da comunicação.

## 9. ESTRUTURA DE ARMAZENAMENTO DO SISTEMA



```
{  
  "users": [  
    {  
      "id": "MrGAXRoLlrhXia9ktwCe",  
      "email": "nicole@gmail.com",  
      "name": "Nicole Olenski",  
      "profilePicture": "",  
      "password": "12345"  
    }  
  ],  
  "patients": [  
    {  
      "id": "aF3nQ1vZxP9bRt7KcJ4L",  
      "userId": "MrGAXRoLlrhXia9ktwCe",  
      "name": "Pedro Silva",  
      "picture": "",  
      "cpf": "123.456.789-00",  
      "telephone": "11997665544",  
      "birthDate": "01/01/2000",  
      "startTreatmentDate": "01/01/2020",  
      "observations": "Pendências nas parcelas R$400: 4/12",  
      "odontogram": {  
        "picture": "",  
        "procedures": [  
          {  
            "id": "U7yT2mLpXz4sVq9BfE8R",  
            "toothId": 4,  
            "procedureName": "Extração",  
            "description": "Não houve complicações",  
            "procedureDate": "10/01/2020"  
          }  
        ]  
      }  
    }  
  ],  
  "consultations": [  
    {  
      "id": "k6HjP0rWqN1tVx9uGd3M",  
      "userId": "MrGAXRoLlrhXia9ktwCe",  
      "patientId": "aF3nQ1vZxP9bRt7KcJ4L",  
      "patientName": "Pedro Silva",  
      "observations": "Paciente disse que pode atrasar 5  
min.", "dateTime": "10/01/2020 14:30"  
    }  
  ]  
}
```

O esquema de banco de dados do Sys Odonto foi concebido para um ambiente não-relacional, utilizando o Google Firestore, e prioriza a eficiência e a escalabilidade. Diferente dos bancos de dados tradicionais que utilizam tabelas e uniões, este modelo organiza as informações em coleções de documentos, conectando-os por meio de referências. Essa abordagem permite leituras de dados muito rápidas, fundamentais para a fluidez do sistema.

A estrutura é composta por três coleções principais de nível superior, que representam as grandes categorias de informações do sistema: "users" (usuários), "patients" (pacientes) e "consultations" (consultas).

A coleção "users" serve como o repositório central para os dados dos profissionais de odontologia. Cada documento nesta coleção representa um dentista, no caso, a Dra. Nicole. Armazena suas informações de identificação, como "id", "email", "name" e "profilePicture", além da "password". É uma coleção enxuta, focada estritamente nos dados do usuário, garantindo que o documento principal não exceda limites de tamanho, o que é crucial para a escalabilidade.

Em seguida, temos a coleção "patients", onde cada documento representa um paciente específico da clínica. Dentro de cada documento de paciente, são guardados seus dados pessoais como "id", "name", "cpf", "telephone", e "birthDate". Um detalhe importante é a presença do campo "userId", que atua como uma referência para o documento do Usuário na coleção "users". É por meio desta referência que sabemos a qual dentista aquele paciente pertence, estabelecendo uma clara ligação entre o profissional e seus pacientes. Além disso, o prontuário odontológico do paciente, que inclui a "startTreatmentDate", as "observations" gerais e o "odontogram" (com seus "procedures" embutidos), é armazenado diretamente dentro do próprio documento do paciente. Essa decisão de aninhamento otimiza a recuperação completa do prontuário ao buscar um paciente, pois todas as informações relevantes já vêm juntas, minimizando o número de operações no banco.

Por fim, a coleção "consultations" abriga todos os agendamentos realizados no sistema. Cada documento nesta coleção detalha uma Consulta específica, contendo informações como "id", "dateTime" e "observations" da consulta. Assim como na coleção "patients", aqui também encontramos as referências cruciais: o campo "userId" aponta para o dentista responsável pela consulta, e o campo "patientId" aponta para o Paciente associado a ela. Um aspecto notável nesta coleção é a presença do campo "patientName". Embora o nome do paciente já esteja disponível na coleção "patients" (referenciada por "patientId"), a decisão de duplicar esta informação diretamente no documento da Consulta foi estratégica. Isso permite que a interface da agenda, ao exibir a lista de consultas do dia, possa carregar rapidamente o nome do paciente junto com os detalhes da consulta, sem a necessidade de realizar uma busca adicional na coleção "patients" para cada item da lista. Essa desnормalização pontual é uma prática comum em bancos NoSQL para otimizar a performance de leituras frequentes.

Em suma, este esquema de banco de dados é um exemplo robusto de como os dados são organizados e interconectados em um ambiente não-relacional. Ao separar as entidades principais em coleções de nível superior e utilizar referências para estabelecer as relações, o Sys Odonto garante alta escalabilidade e um desempenho ágil na recuperação e manipulação das informações

## 10. PROTÓTIPOS DO SISTEMA

A validação do protótipo foi realizada por meio de uma sessão de feedback assíncrona com a Dra. Nicole. Durante a sessão, foi apresentada uma demonstração guiada das principais telas e funcionalidades do sistema, detalhando a função de cada componente e o fluxo de navegação entre eles. Este processo resultou na coleta de valiosos insights e sugestões de ajuste, que servirão como diretriz para as próximas etapas de desenvolvimento do Sys Odonto. Durante a discussão, a Dra. explicou que a funcionalidade de ativar/inativar paciente é desnecessária, uma vez que a ausência do paciente no consultório por longos períodos não altera seu status ou ficha. A mesma sugeriu, por outro lado, que o Sys Odonto apresentasse um odontograma personalizável para cada paciente, a fim de registrar os procedimentos realizados em cada dente, e ressaltou a necessidade de importar os dados do papel para o sistema de forma eficiente.

### Link para o protótipo:

[https://www.canva.com/design/DAGzKt4tTqw/5DXDDH1nKrm29de0V4s1LA/edit?utm\\_content=DAGzKt4tTqw&utm\\_campaign=designshare&utm\\_medium=link2&utm\\_source=sharebutton](https://www.canva.com/design/DAGzKt4tTqw/5DXDDH1nKrm29de0V4s1LA/edit?utm_content=DAGzKt4tTqw&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton)

## 11. RESULTADOS ATINGIDOS

O projeto Sys Odonto alcançou marcos significativos no desenvolvimento de uma solução tecnológica para a gestão de clínicas odontológicas, iniciando pela consolidação de um escopo alinhado às necessidades reais da prática clínica. Através de reuniões de feedback com a stakeholder, Dra. Nicole, foi possível refinar os requisitos do sistema, resultando na inclusão de funcionalidades críticas como o odontograma por exemplo. Essas interações permitiram eliminar processos que não agregam valor, garantindo que o software foque na produtividade e na facilidade de uso para profissionais com pouca experiência tecnológica.

No âmbito técnico, a arquitetura do sistema foi integralmente estabelecida seguindo o padrão cliente-servidor em camadas, utilizando tecnologias modernas como Java com Spring Boot no backend e React no frontend. A modelagem de dados foi concluída com sucesso para o banco de dados NoSQL Google Firestore, priorizando a escalabilidade e a rapidez na recuperação de informações. Um resultado de destaque foi a aplicação de técnicas de desnormalização estratégica, como a duplicação do nome do paciente na coleção de consultas, o que permite uma visualização da agenda muito mais ágil e eficiente.

Quanto à funcionalidade de comunicação, que é um dos pilares do projeto, o sistema operacionalizou o envio de lembretes via WhatsApp. Embora o planejamento inicial previsse a integração direta via API, a equipe implementou com sucesso um bot de automação para realizar os disparos. Essa solução cumpre o objetivo de substituir os métodos manuais de contato.

## 12. REFERÊNCIAS BIBLIOGRÁFICAS

**PRESSMAN**, Roger S.; **MAXIM**, Bruce R. *Engenharia de Software: uma abordagem profissional*. 8. ed. Porto Alegre: AMGH, 2016.

**SOMMERVILLE**, Ian. *Engenharia de Software*. 9. ed. São Paulo: Pearson, 2011.

**GOOGLE**. Cloud Firestore Documentation. Disponível em: <https://firebase.google.com/docs/firestore>.

**REACT**. React Documentation. Disponível em: <https://react.dev>.

## 13. CONTROLE DE VERSÃO DO DOCUMENTO

O controle de versão deve ser mantido para registrar as alterações realizadas e acompanhar os itens modificados.

Tabela 3. Controle de versão

Versão	Data	Seção	Descrição da Alteração
V1	03/09/2025	Introdução, objetivo, público alvo	Primeira versão do documento, contendo o escopo, metas e público alvo.
V2	09/09/2025	Metodologia, resultados esperados, requisitos funcionais, requisitos não funcionais	Definição da metodologia a ser utilizada para a construção do projeto; detalhamento dos resultados esperados com o projeto; primeira versão dos requisitos funcionais e não funcionais.
V3	09/09/2025	Requisitos funcionais, requisitos não funcionais	Alteração do requisito funcional RF21 para reativação do paciente, e inclusão do requisito funcional RF22. Deleção dos requisitos não funcionais de desempenho, segurança e escalabilidade. Inclusão do requisito não funcional RNF5.

V4	17/09/2025	Requisitos funcionais	Adição do campo “observação” como entrada para cadastro de pacientes. Adição do RF23.
V5	30/09/2025	Requisitos funcionais	Remoção dos requisitos funcionais RF21 e RF22 à respeito da inativação e reativação do paciente, respectivamente. Adição da entrada opcional <b>odontograma</b> para os requisitos RF13 (cadastro de paciente) e RF15 (edição de paciente). Adição do requisito RF22, para importação de dados de pacientes.