**SENAI ETTORE ZANINI**

**TÉCNICO DESENVOLVIMENTO DE SISTEMAS  
MAAP**

**MedClinic**

**Sertãozinho  
2025**

**LISTA DE FIGURAS**

Figura 1 – Benchmarking ............................................... 1

Figura 2 – Requisitos do Sistema .................................................................. 2

Figura 3 – Modelagem ................................................................................ 3

Figura 4 – KanBan .....................................................................................4

Figura 5 – Dicionário API.............................................................................5

**SUMÁRIO**

1 BENCHMARKETING ................................................................. 4

1.1 Objetivos ...................................................4

1.2 Contextualização ........................................................... 4

2 REQUISITOS DO SISTEMA ................................................. 9

2.1 Requisitos Funcionais ............................................. 10

2.2 Diagrama do sistema .................................................. 18

2.3 Diagrama de classe ....................................................... 19

2.4 Rotas da aplicação backend........................................................23

3 MODELAGEM................................................................... 24

3.1 Contextualização............................................................. 24

4 KANBAN

4.1 Imagens das Sprints no KanBan ..............................................24

5. DICIONÁRIO API

5.1 ............................................................................................... 29

**1. BENCHMARKING**

O objetivo do benchmarking na nossa aplicação é analisar e comparar clinicas medicas para sofisticar nossos serviços, implementando estratégias para uma melhor qualidade e variedade no atendimento.

|  |  |  |
| --- | --- | --- |
| **COLETA DE DADOS** | | |
| **SITES** | **PONTOS POSITIVOS** | **PONTOS NEGATIVOS** |
| **Amor Saúde** | Atendimento diferenciado e personalizado. | Limitações na quantidade de especialidade disponíveis. |
| Acesso facilitado para pessoas com deficiência. | Possibilidades de filas de esperar para consultas em horário de pico. |
| **Meu Doutor** | Agendamento de consultas disponíveis pelo site. | Dificuldades técnicas ocasionadas no sistema de agendamento online. |
| Parcerias com laboratórios para exames específicos e diferenciados. | Tempo de espera para agendamento de consultas pode ser prolongado. |
| **Iclinic** | Sistema de gestão completo para clínicas e consultórios. | Curva de aprendizado para usuários menos familiarizados com tecnologia. |
| Controle financeiro e emissão de notas fiscais. | Funcionalidades avançadas podem não ser utilizadas por clínicas menores. |

Amor Saúde e Meu Doutor são mais voltados para o atendimento direto ao paciente, com foco em consultas acessíveis e agendamento online. Eles competem diretamente entre si, com diferenças sutis em cobertura geográfica e experiência do usuário.

IClinic é uma solução voltada para clínicas e consultórios, ajudando na gestão interna, como agendamentos, prontuários e finanças. Não compete diretamente com Amor Saúde e Meu Doutor, mas pode ser usado por clínicas que fazem parte dessas redes.

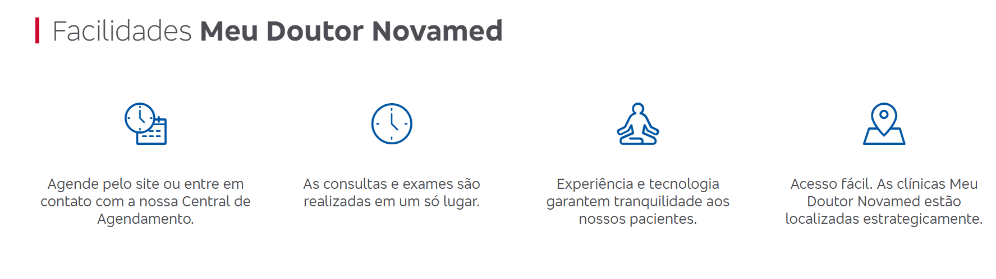
Vantagem: Melhora a eficiência operacional das clínicas.

Desvantagem: Não atende diretamente ao paciente e pode ter custos elevados para pequenas clínicas.

**Referências:**

**AMOR SAÚDE.**



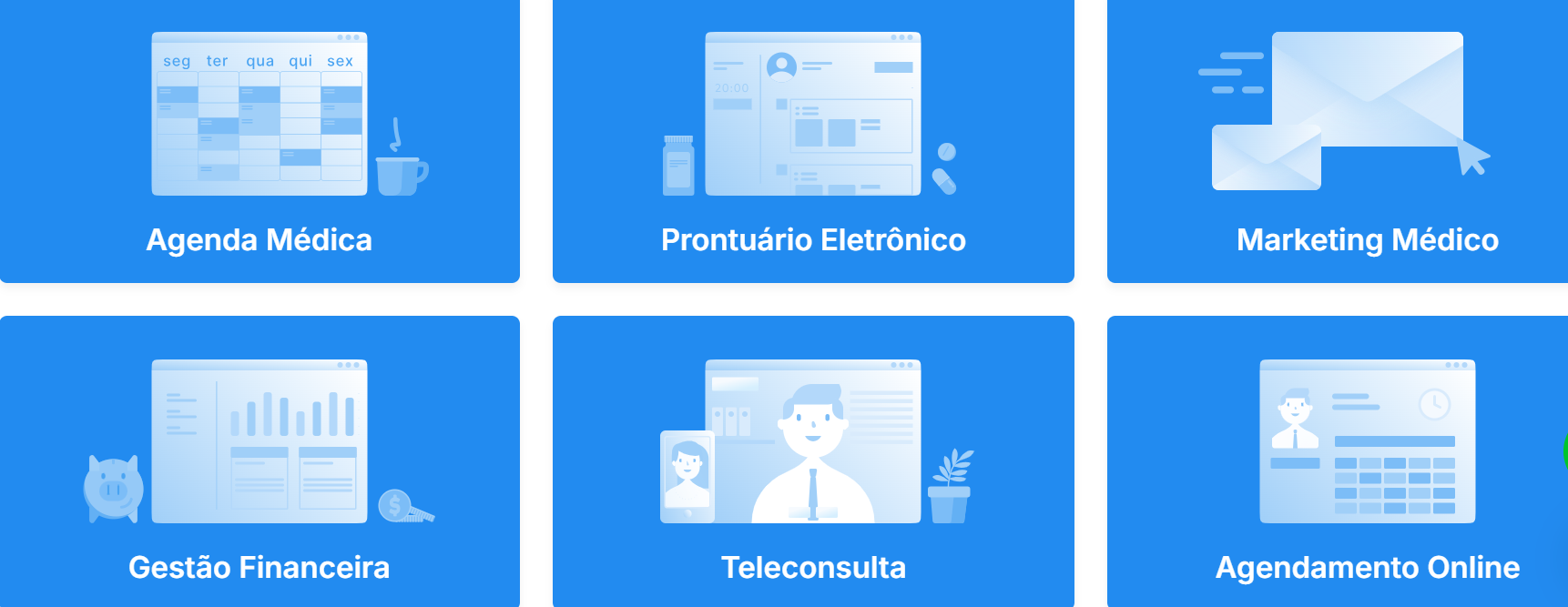


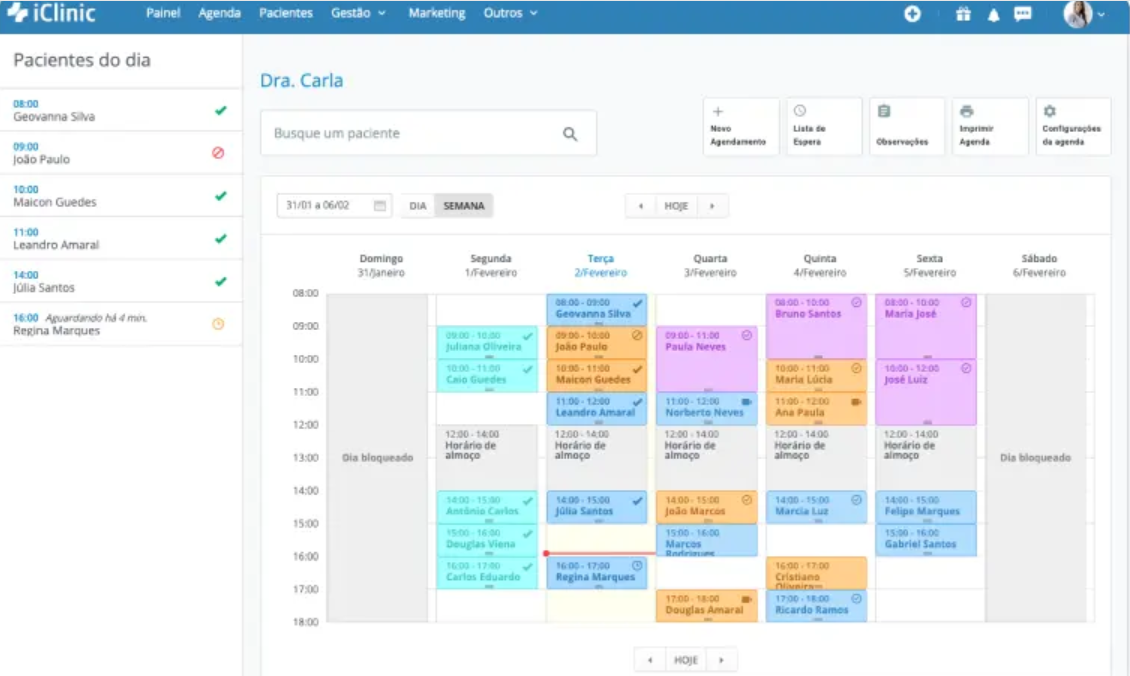
**MEU DOUTOR.**



**ICLINIC**







A tela de agendamento da ICLINIC é uma ferramenta poderosa para otimizar a gestão de agendamentos da sua clínica.

Ao explorar todas as funcionalidades e personalizações disponíveis, você poderá adaptar a ferramenta às necessidades específicas da sua clínica e equipe.

**COLETA DE DADOS DO ICLINIC:**

-Lista os pacientes agendados para o dia, proporcionando uma visão rápida da movimentação da clínica.

- Agenda Médica e Prontuário Eletrônico otimizam a gestão de agendamentos, prontuários de pacientes e informações clínicas, agilizando o fluxo de trabalho e reduzindo a burocracia.

- O Marketing Médico impulsiona o relacionamento com pacientes, fortalece a marca da clínica e atrai novos pacientes.

- A Gestão Financeira oferece controle total sobre as finanças da clínica, enquanto a Tele consulta e Agendamento Online expandem o alcance da clínica e oferecem mais comodidade aos pacientes.

-Agiliza a localização de pacientes na base de dados para agendamentos ou consultas rápidas.

- Define o período de visualização da agenda, oferecendo flexibilidade para verificar agendamentos futuros ou passados.

- A tela de agendamento é integrada com outras funcionalidades da ICLINIC, como prontuários eletrônicos, gestão financeira e marketing, proporcionando uma gestão completa da clínica.

O Benchmarking trata-se de um processo de pesquisa e análise entre empresas do mesmo setor, com o objetivo de comparar o desempenho de seus produtos, serviços e processo em relações aos concorrentes.

Essa estratégia de monitoramento de mercado envolve coleta, interpretação, avaliação e mensuração de informações relevante. Para construir uma inteligência de mercado eficiente, é essencial compreender o significado dos dados obtidos e identificar como eles podem contribuir para o crescimento da empresa.

De modo geral, cada organização define os fatores que influenciam seu desempenho estabelece métricas baseadas nos principais indicadores de performance (KPIs) do setor ou mercado.

Benchmarking pode ser aplicado em diversas áreas de uma organização, como:

Marketing e Vendas: Comparar estratégias de marketing, campanhas publicitárias, processos de vendas, atendimento ao cliente, etc.

Operações e Produção: Comparar processos produtivos, gestão da qualidade, logística, gestão de estoques, etc.

Finanças: Comparar indicadores financeiros, processos de controle, gestão de custos, etc.

Tecnologia da Informação: Comparar infraestrutura de TI, sistema de informação, segurança da informação, etc.

Exemplos de aplicações, são:

Uma empresa de Varejo, que pode fazer Benchmarking com um concorrente para identificar as melhores práticas em gestão de estoque e layout de loja.

Uma empresa de serviços pode fazer Benchmarking com uma empresa de outro setor que é referência em atendimento ao cliente.

Uma empresa de tecnologia pode fazer Benchmarking com empresas líderes em inovação para identificar oportunidades de desenvolvimento de novos produtos.

Benefícios:

O Benchmarking pode trazer diversos benefícios para uma organização, como:

Melhora do Desempenho: Ao identificar e implementar as melhores práticas, a empresa pode melhorar sua eficiência, produtividade e qualidade.

Aumento da Competitividade: O Benchmarking ajuda a empresa a se manter competitiva, identificando e respondendo às mudanças do mercado.

Redução de Custos: Ao otimizar processos e eliminar desperdícios, a empresa pode reduzir seus custos operacionais.

Inovação: O Benchmarking pode estimular a inovação, ao trazer novas ideias e perspectivas de outras empresas.

Aprendizado: O processo de Benchmarking promove o aprendizado e o desenvolvimento de novas habilidades e conhecimentos.

**RESULTADOS**

Iremos implementar ao site da nossa aplicação o acesso facilitado a pessoas com deficiência (baixa visão, dificuldade na escuta e a presença de interprete de libras nas consultas), realizaremos atendimento personalizado a pedidos dos pacientes, nos adaptando de acordo com a sua solicitação.

Implementaremos também uma gestão financeira, para organizarmos e planejarmos as finanças da clínica, para um melhor atendimento e maior qualidade para os pacientes. Para uma melhor experiência da clínica, para um melhor atendimento e maior qualidade para os pacientes. Para uma melhor experiência, os pacientes poderão fazer o agendamento da consulta pelo site, e faremos parecerias com laboratórios com diversas especialidades em exame.

<https://www.amorsaude.com.br/quem-somos/> <https://iclinic.com.br/>

vivaconsulta.com.br/sobre

<https://www.novamedsaude.com.br/wps/portal/novamed/area-publica/inicio>

**2. REQUISITOS DO SISTEMA**

O MedClinic é um sistema completo e inovador desenvolvido para conectar pacientes e médicos de maneira ágil e eficiente, proporcionando um atendimento médico digital de qualidade. A plataforma integra diversas funcionalidades essenciais para a gestão de consultas, prontuários e prescrições, tornando a experiência médica mais moderna e acessível.

Com o MedClinic, pacientes podem agendar consultas, realizar atendimentos por vídeo e acessar seus registros médicos com facilidade, enquanto médicos têm à disposição ferramentas avançadas para organizar sua rotina, acompanhar tratamentos e otimizar o tempo de atendimento.

Os requisitos do sistema são essenciais para estabelecer expectativas, guiar o processo de desenvolvimento e assegurar que o produto satisfaça as necessidades dos usuários, reduzindo retrabalhos e custos usamos.

**Paciente:** id\_paciente, nome, CPF, celular, e-mail, data\_nascimento e endereco **Médico:** id\_medico, id\_paciente, id\_consulta, nome, especialidade, CRM, celular e e-mail.

**Consulta:** id\_consulta, data, horário, diagnostico, receita, sala e status.

**2.1Requisitos funcionais.**

Requisitos funcionais são as funcionalidades que o sistema deve ter para atender às necessidades dos usuários e resolver os problemas propostos. Eles descrevem o que o sistema deve fazer, ou seja, as ações que ele deve realizar, os comportamentos esperados e as interações com o usuário.

Em outras palavras, são as regras de negócio que definem como o sistema deve funcionar.

|  |  |
| --- | --- |
| **REQUISITOS FUNCIONAIS** | |
| **RF001** | Cadastrar Paciente |
| **RF002** | Editar dados do Paciente |
| **RF003** | Excluir Cadastro do Paciente |
| **RF004** | Agendar Consulta |
| **RF005** | Cadastrar Médico. |
| **RF006** | Notificar Paciente. |
| **RF007** | Realizar Consulta online. |
| **RF008** | Registrar Prescrição Médica. |
| **RF009** | Consultar Histórico do Paciente. |
| **RNF001** | Contar quantas pessoas passaram no Iclinic. |

**2.2.1 RF001 – Cadastrar Paciente**

Descrição: O sistema deve permitir que o usuário cadastre novos pacientes no sistema MedClinic.

Prioridade: Alta

Entrada:

* Nome completo do paciente (obrigatório).
* CPF (obrigatório).
* Telefone (obrigatório).
* Data Nascimento (telefone/e-mail) (obrigatório).

Processamento:

* O sistema deve validar os campos obrigatórios.

Saída:

* Mensagem de sucesso ao cadastrar o paciente.
* Mensagem de erro em caso de campos obrigatórios não preenchidos ou duplicidade de cadastro (CPF já existente).

Pós-condição:

* O paciente cadastrado deve ser exibido na lista de registros disponíveis no sistema.

**2.2.2 RF002 – Editar Dados do Paciente**

Descrição: O sistema deve permitir que o usuário edite as informações de um paciente previamente cadastrado.

Prioridade: Alta

Entrada:

* Identificação do paciente a ser editado (ID ou CPF).
* Campos a serem editados (Nome, CPF, Telefone, Data Nascimento).

Processamento:

O sistema deve validar os campos obrigatórios.

Saída:

* Mensagem de sucesso ao salvar as alterações.
* Mensagem de erro caso haja duplicidade de CPF ou campos obrigatórios não preenchidos.

Pós-condição:

* As alterações devem ser refletidas na lista de pacientes e em todos os registros associados ao paciente, como histórico médico e consultas agendadas.

**2.2.3 RF003 – Excluir Cadastro de Paciente**

Descrição: O sistema deve permitir que o usuário exclua um paciente do sistema MedClinic.

Prioridade: Média

Entrada:

* Identificação do paciente a ser excluído (ID ou CPF).

Processamento:

* O sistema deve verificar se há consultas agendadas ou registros médicos associados ao paciente.
* Se houver consultas futuras ou histórico médico vinculado, o sistema deve impedir a exclusão e informar o usuário.

Saída:

* Mensagem de sucesso ao excluir o paciente.
* Mensagem de erro caso existam consultas agendadas ou registros médicos impedindo a exclusão.

Pós-condição:

* O paciente deve ser removido da lista de registros e não deve aparecer em buscas e consultas, salvo em casos de necessidade legal de arquivamento de dados.

**2.2.4 RF004 – Agendar Consulta**

Descrição: O sistema deve permitir que os usuários agendem consultas para obter assistência ou informações relacionadas aos serviços oferecidos pela biblioteca.

Prioridade: Baixa

Entrada:

* Dados do usuário.
* Parâmetros de preferência para horário e tipo de consulta.
* Interação do usuário (opcional).

Processamento:

* O sistema verifica a disponibilidade dos atendentes ou dos horários disponíveis para consultas. O agendamento pode ser feito com base em critérios como a preferência de horário ou o tipo de consulta solicitada.

Saída:

* Confirmação do agendamento com informações sobre data, horário e detalhes adicionais.

Possibilidade de reprogramação ou cancelamento do agendamento, caso necessário.

Pós-condição:

* O usuário recebe uma confirmação do agendamento na interface do sistema, seja via web ou aplicativo móvel.

**2.2.5 RF005 – Cadastrar Médico**

Descrição: O sistema deve permitir o cadastro de novos médicos, com informações pessoais e profissionais, como nome, especialidade, endereço, telefone e email.

Prioridade: Alta

Entrada:

* Dados fornecidos pelo médico ou pela administração da instituição.
* Dados opcionais (id\_paciente, id\_consulta, nome, especialidade, CRM, celular e email)

Processamento:

* Validação dos dados.
* Armazenamento no banco de dados.
* Criação do perfil do médico, com suas especialidades e credenciais.
* Associação de permissões, como acesso a agendamentos e consulta de prontuários.

Saída:

* Confirmação do cadastro.
* Geração de identificação ou código do médico.

Pós-condição:

* O novo médico está cadastrado no sistema da instituição, podendo realizar atendimentos, acessar informações sobre consultas agendadas e gerenciar sua agenda

**2.2.6 RF006 – Notificar Atraso de Consulta**

Descrição: O sistema deve verificar se o paciente não compareceu à consulta marcada e notificá-lo sobre o atraso ou falta.

Prioridade: Alta

Entrada:

* Dados do sistema de agendamento de consultas.
* Critérios de atraso ou falta na consulta (como o horário da consulta e a ausência do paciente).

Processamento:

* Identificação de consultas não realizadas ou com atraso.
* Cálculo de deliberações, caso aplicável (como taxas ou necessidade de reagendamento).
* Seleção do canal de envio da notificação (ex.: e-mail, SMS ou aplicativo).

Saída:

* A notificação é enviada ao paciente com informações sobre a consulta perdida possíveis, conclusões (se houver) e instruções para reagendamento.

Pós-condição:

* A consulta é marcada como “Ausente” ou “Não Realizada” até que o paciente compareça ou reagende.
* Caso tenha sorte, ela é registrada no sistema.

**2.2.7 RF007 – Realizar Consulta Online**

Descrição: O sistema deve permitir que o paciente agende e realize uma consulta médica online com um profissional cadastrado.

Prioridade: Alta

Entrada:

• Data e horário da consulta (obrigatórios).

• Receita da medicação (se aplicada).

• Sala da consulta (Indica o ambiente de atendimento).

• Status de consulta (Agendada, Realizada ou Cancelada).

Processamento:

• O sistema deve verificar a disponibilidade do médico no horário solicitado.

• O sistema deve associar a consulta ao paciente e ao médico.

• O sistema deve garantir que os recursos necessários para a consulta online (como plataforma de vídeo) estejam configurados.

Saída:

• Confirmação do agendamento da consulta online com data, horário, sala de consultas e status.

• Mensagem de erro caso o médico ou horário solicitado não esteja disponível.

Pós-condição:

• A consulta é registrada no sistema como "Agendada".

• O paciente e o médico têm acesso às informações da consulta na plataforma.

• O sistema registra o histórico da consulta, incluindo status e informações relevantes.

**2.2.8 RF008 – Registrar Prescrição Médica**

Descrição: O sistema deve permitir que o médico registre uma prescrição médica para um paciente durante ou após uma consulta.

Prioridade: Alta

Entrada:

* Identificação do paciente (ID ou CPF).
* Detalhes da prescrição (medicamentos, dosagem, instruções).
* Data e hora da prescrição.
* Sala da consulta (Indica o ambiente de atendimento).
* Status de consulta (Agendada, Realizada ou Cancelada).

Processamento:

* O sistema deve validar se o paciente está cadastrado.
* O sistema deve garantir que todos os campos obrigatórios da prescrição estejam preenchidos.

Saída:

* Mensagem de sucesso ao registrar a prescrição.
* Mensagem de erro caso o paciente não seja encontrado ou campos obrigatórios não estejam preenchidos.

Pós-condição:

* A prescrição deve ser associada ao histórico do paciente e estar disponível para consulta futura.

**2.2.9 RF009 – Consultar Histórico do Paciente**

Descrição: O sistema deve permitir que os usuários consultem o histórico completo de um paciente, incluindo consultas anteriores, prescrições e informações pessoais.

Prioridade: Média

Entrada:

* Identificação do paciente (ID ou CPF).

Processamento:

* O sistema deve buscar e exibir todas as informações relacionadas ao paciente, incluindo consultas, prescrições e dados pessoais.

Saída:

* Exibição do histórico completo do paciente.
* Mensagem de erro caso o paciente não seja encontrado.

Pós-condição:

* O histórico do paciente deve ser exibido de forma clara e organizada para o usuário.

**2.2.1 RNF001 – Tempo de Resposta do Sistema**

Descrição: O sistema deve responder a quantidade de pessoas que passaram na clínica.

Prioridade: Alta

Entrada:

* Qualquer requisição feita ao sistema.

Processamento:

* O sistema deve processar a requisição e retornar uma resposta dentro do tempo estipulado.

Saída:

* Resposta à requisição dentro de 2 segundos.

Pós-condição:

* A experiência do usuário deve ser fluida, sem atrasos perceptíveis.
* Esses requisitos garantem que o sistema seja eficiente e atenda às necessidades dos usuários de forma adequada.

**2.DIAGRAMA DO SISTEMA**

Os diagramas desempenham um papel fundamental no processo de desenvolvimento de software, sendo ferramentas essenciais para os programadores e equipes de desenvolvimento. Eles permitem visualizar de forma clara e estruturada como um sistema deve funcionar, mesmo antes de qualquer linha de código ser escrita. Essa visualização antecipada é crucial para garantir que todos os envolvidos no projeto tenham uma compreensão compartilhada do funcionamento do sistema, evitando mal-entendidos e inconsistências que poderiam surgir durante a implementação.

Um dos principais benefícios dos diagramas é a capacidade de representar como as diferentes partes do sistema se interconectam. Por exemplo, diagramas de arquitetura, como diagramas de componentes ou de implantação, mostram como os módulos do sistema se relacionam entre si e com sistemas externos. Já diagramas de sequência ou de fluxo de dados ajudam a entender como as informações circulam entre os componentes, identificando pontos de integração e dependências. Essa clareza é especialmente útil em projetos complexos, onde múltiplas equipes trabalham em diferentes partes do sistema.

Além disso, os diagramas são ferramentas poderosas para identificar problemas de design ainda na fase de planejamento. Ao visualizar o sistema de forma abstrata, é possível detectar falhas de lógica, gargalos de desempenho ou inconsistências no fluxo de dados antes que eles se tornem problemas reais durante a codificação. Isso economiza tempo e recursos, pois corrigir erros na fase de design é significativamente mais barato e menos trabalhoso do que fazê-lo após a implementação.

Outro aspecto importante é que os diagramas facilitam a comunicação entre os membros da equipe e stakeholders. Eles servem como uma linguagem comum que pode ser compreendida por desenvolvedores, gerentes de projeto, designers e até mesmo clientes, garantindo que todos estejam alinhados com os objetivos e funcionalidades do sistema. Essa transparência é essencial para o sucesso do projeto, especialmente em equipes distribuídas ou multidisciplinares.

Por fim, os diagramas também contribuem para a manutenção do sistema a longo prazo. Eles funcionam como documentação viva, fornecendo uma visão geral do sistema que pode ser consultada por novos desenvolvedores ou para revisitar decisões de design no futuro. Isso torna o processo de manutenção e escalabilidade do sistema mais eficiente, já que a estrutura e o fluxo de dados estão claramente documentados.

Em resumo, os diagramas são muito mais do que simples representações visuais; eles são ferramentas estratégicas que ajudam a planejar, comunicar, identificar problemas e documentar sistemas de software, garantindo que o desenvolvimento seja mais organizado, eficiente e colaborativo.

**2.1 DIAGRAMA DE CLASSE**

O diagrama de classes é um dos diagramas mais importantes na modelagem de sistemas orientados a objetos, sendo parte da linguagem UML (Unified Modeling Language). Ele é utilizado para representar a estrutura estática de um sistema, mostrando as classes, seus atributos, métodos e os relacionamentos entre elas. Esse tipo de diagrama é essencial para entender a arquitetura do software e como os objetos interagem entre si.

O diagrama de classe da MedClinic, descreve três classes principais, **Médico,** **Consulta** e **Paciente.**

**Classe Médico**

* **Atributos:**
* **Id\_medico (number):** Identificador único do médico.
* **Nome (string):** nome do médico.
* **Especialidade (string):** Especialidade do médico.
* **CRM (string):** Número do CRM (Conselho Regional de Medicina) medico.
* **Telefone (string):** Número de telefone do médico.
* **E-mail (string):** Endereço de e-mail do médico.

**Métodos:**

* **listarMedicos(): Array<Medico>:** Retorna uma lista de todos os médicos cadastrados.
* **cadastrarMedico(medico: Medico):** Boolean: Cadastra um novo médico e retorna true se o cadastro for bem-sucedido.
* **removerMedico(idMedico: number): Boolean**: Remove um médico com base no idMedico e retorna true se a remoção for bem-sucedida.
* **AtualizarCadastroMedico (medico: Medico): Boolean:** Atualiza os dados de um médico e retorna true se a atualização for bem-sucedida.

**Classe Paciente**

Atributos:

* **IdPaciente (number):** Identificador único do paciente.
* **Nome (string):** Nome do paciente
* **CPF (string)**: CPF do paciente.
* **Telefone (string):** Número de telefone do paciente.
* **DataNascimento (date):** Data de nascimento do paciente.
* **Endereco (string):** Endereço do paciente.

**Métodos:**

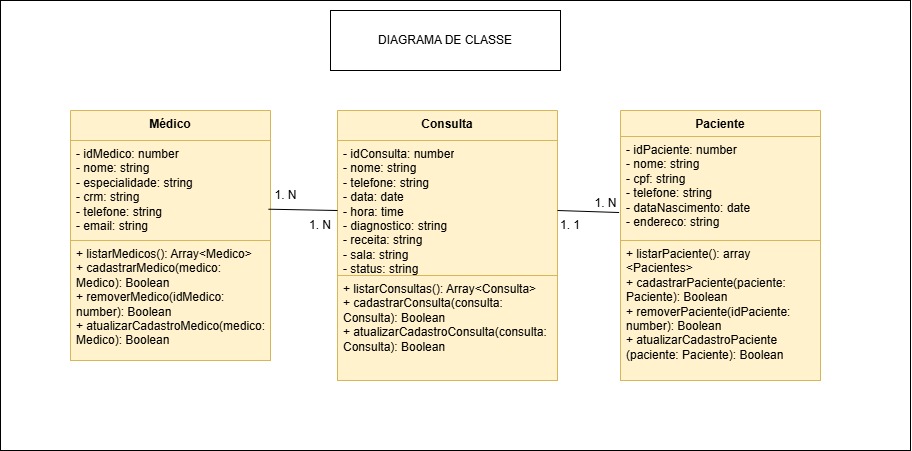
* **listarPaciente(): Array<Paciente>:** Retorna uma lista de todos os pacientes cadastrados.
* **cadastrarPaciente(paciente: Paciente): Boolean:** Cadastra um novo paciente e retorna true se o cadastro for bem-sucedido.
* **removerPaciente(idPaciente: number): Boolean:** Remove um paciente com base no idPaciente e retorna true se a remoção for bem-sucedida.
* **AtualizarCadastroPaciente (paciente: Paciente): Boolean:** Atualiza os dados de um paciente e retorna true se a atualização for bem-sucedida.

Embora não estejam explicitamente descritos no diagrama, é possível inferir alguns relacionamentos entre as classes:

**Médico e Consulta:** Um médico pode estar associado a várias consultas.

**Paciente e Consulta:** Um paciente pode ter várias consultas agendadas ou realizadas.

**Médico e Paciente:** Um médico pode atender vários pacientes, e um paciente pode ser atendido por vários médicos (dependendo da especialidade).



**2.2 DER**

O DER é uma ferramenta indispensável para o planejamento e desenvolvimento de sistemas, pois oferece uma visão estruturada e organizada do banco de dados. Ele não apenas facilita a implementação técnica, mas também garante que o sistema atenda às necessidades do negócio de forma eficiente e escalável. Sem um DER bem elaborado, o risco de problemas como redundância, inconsistência e dificuldade de manutenção aumenta significativamente.

Aqui estão os principais componentes do DER da MedClinic.

**Entidades:**

**Paciente**: Representa os pacientes do sistema.

* Atributos: **id\_paciente,** **nome, cpf, celular, email, data\_nascimento, endereco.**

**Médico**: Representa os médicos do sistema.

* Atributos: **id\_medico, id\_paciente, id\_consulta, nome, especialidade, crm, celular, email.**

**Consulta**: Representa as consultas médicas.

* Atributos: **id\_consulta, data, horario, diagnostico, receita, sala, status.**

**Relacionamentos:**

* **Médico-Paciente**: Um médico pode estar associado a um paciente através do **id\_paciente.**
* **Médico-Consulta**: Um médico está associado a uma consulta através do **id\_consulta.**

**Chaves**

**Chaves Primárias (PK):**

* **id\_paciente** na entidade Paciente.
* **id\_medico** na entidade Médico.
* **id\_consulta** na entidade Consulta.

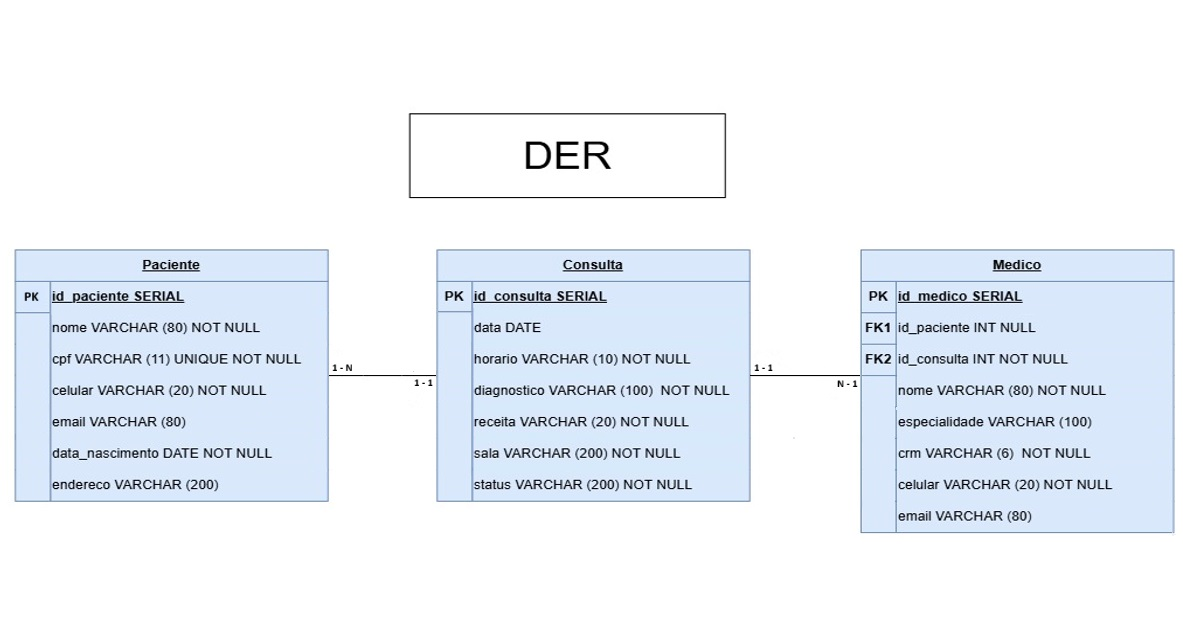
**Chaves Estrangeiras (FK):**

* **id\_paciente** na entidade Médico, referenciando **id\_paciente** na entidade Paciente.
* **id\_consulta** na entidade Médico, referenciando **id\_consulta** na entidade Consulta.

**Atributos:**

* Cada entidade possui atributos que descrevem suas propriedades, como nome, CPF, especialidade, etc.

Este DER fornece uma visão geral das entidades e como elas se relacionam em um sistema de saúde, facilitando a compreensão e a implementação do banco de dado



**2.DICIONÁRIO API**

|  |  |  |
| --- | --- | --- |
| **DICIONÁRIO ENDPOINT** | | |
| **ROTAS** | **REQUISIÇÃO** | **PARÂMETRO** |
| / | GET |  |
| /listar/medico | GET | idMedico  statusMedico  nome  especialidade  crm  telefone  email |
| /listar/paciente | GET | idPaciente  statusPaciente  nome  cpf  telefone  email  dataNascimento  endereco |
| /listar/consulta | GET | idConsulta  statusConsultaRegistro  nome   data   hora  diagnostico   receita  salaAtendimento  consultaStatus  idPaciente  idMedico |
| /cadastro/medico | POST | idMedico  statusMedico  nome  especialidade  crm  telefone  email |
| /cadastro/paciente | POST | nome  cpf  telefone  email  dataNascimento  endereco |
| /cadastro/consulta | POST | idConsulta  statusConsultaRegistro  nome   data   hora  diagnostico   receita  salaAtendimento  consultaStatus  idPaciente  idMedico |
| /remover/medico | PUT | Remover médico. |
| /remover/paciente | PUT | Remover paciente. |
| /remover/consulta | PUT | Remover consulta. |
| /atualizar/medico/4 | PUT | nome  especialidade  crm  telefone  email |
| /atualizar/paciente/4 | PUT | nome  cpf  telefone  email  dataNascimento  endereco |
| /atualizar/consulta/4 | PUT | nome  cpf  telefone  email  dataNascimento  endereco |
| /remover/paciente/1 | DELETE | “Paciente removido com sucesso!” |
| /remover/medico/1 | DELETE | “Medico removido com sucesso!” |
| /remover/consulta/1 | DELETE | “Consulta removida com sucesso!” |
| /login | POST | Página do login cliente. |

**3. MODELAGEM**

A modelagem de software consiste na criação de modelos que descrevem o comportamento e as características de um sistema, como um guia para o projeto, e seus objetivos são, identificar funcionalidades e características de software, planejar a construção do seu sistema e explicar o funcionamento e a interação entre as partes do software.

Suas vantagens são, organizar e melhorar o fluxo de desenvolvimento, facilitar a compreensão do projeto, permitir e capturar requisitos com precisão. Melhorar a colaboração entre equipes e de exemplo, uma ferramenta que utilizamos em aula foi o fluxograma/diagrama de atividade.

Alguns tipos de modelagem de software são, modelo de diagrama UML, modelagem dirigida a eventos, fluxogramas, diagramas de atividades e alguns softwares de modelagem de processos (e que alguns usamos em aula) são, Figma, Miro, Draw.io, Microsoft e etc.

Esses modelos foram utilizados para projetar um determinado sistema (diagrama de classe e fluxograma), facilitando o entendimento de sistemas complexos e a afeiçoar diferentes interações.