

# **ARACLIN: CATÁLOGO DE CLÍNICAS MÉDICAS**

**ARTUR SOUSA LUZZI  
ROBERTA SANTOS VIEIRA  
THIAGO RAMOS DE ANDRADE**

**Araçuaí - MG  
5 de outubro de 2023**

INSTITUTO FEDERAL DO NORTE DE MINAS GERAIS  
CAMPUS ARAÇUAÍ  
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

## **ARACLIN: CATÁLOGO DE CLÍNICAS MÉDICAS**

ARTUR SOUSA LUZZI  
ROBERTA SANTOS VIEIRA  
THIAGO RAMOS DE ANDRADE

Relatório final apresentado aos professores  
como requisito avaliativo para o Trabalho Prá-  
tico Interdisciplinar.

Araçuaí - MG  
5 de outubro de 2023

# Sumário

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>4</b>
<b>1.1</b>	<b>Motivação . . . . .</b>	<b>4</b>
<b>1.2</b>	<b>Objetivos . . . . .</b>	<b>5</b>
<b>2</b>	<b>DOCUMENTAÇÃO . . . . .</b>	<b>6</b>
<b>2.1</b>	<b>Engenharia de Software . . . . .</b>	<b>6</b>
<b>2.2</b>	<b>Programação Orientada a Objetos . . . . .</b>	<b>11</b>
<b>2.3</b>	<b>Programação Web . . . . .</b>	<b>12</b>
<b>2.4</b>	<b>Banco de dados II . . . . .</b>	<b>22</b>
<b>3</b>	<b>TECNOLOGIAS UTILIZADAS . . . . .</b>	<b>28</b>
<b>4</b>	<b>CONCLUSÕES . . . . .</b>	<b>29</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>30</b>

# Resumo

Este trabalho apresenta o desenvolvimento de um projeto acadêmico interdisciplinar que teve como objetivo a concepção e desenvolvimento de um website, denominado "AraClin", voltado para fornecer informações sobre clínicas médicas privadas situadas em Araçuaí-MG. A abordagem aplicada integra conhecimentos de Engenharia de Software, Programação Orientada a Objetos, Programação Web e Banco de Dados I. A metodologia envolveu análise de demandas dos usuários, escolha de tecnologias, definição da arquitetura, implementação e testes. Apesar de não ter sido publicado, o website proporcionou a aplicação prática dos conhecimentos teóricos em um contexto desafiador, resultando no aprimoramento das habilidades profissionais da equipe. Isso inclui o trabalho colaborativo, aplicação de conhecimentos técnicos, gestão de projetos e melhores práticas de programação.

**Palavras-chave:** Projeto interdisciplinar; website; AraClin; clínicas médicas; desenvolvimento.

# 1 Introdução

Este projeto é resultado da síntese dos conhecimentos adquiridos nas disciplinas de Engenharia de Software, Programação Orientada a Objetos, Programação Web e Banco de Dados I. O foco é a concepção e desenvolvimento de um website que desempenha um papel crucial como uma plataforma aberta, permitindo que os usuários acessem de maneira descomplicada informações essenciais sobre as clínicas médicas privadas em Araçuaí-MG. A gama de informações disponíveis inclui detalhes sobre os médicos, suas especialidades, informações de contato, localizações e outros atributos relevantes.

Por meio de uma abordagem interdisciplinar, o projeto une saberes provenientes de diversas áreas de estudo. A expectativa é que essa abordagem resulte em uma solução robusta e completa, capaz de atender de maneira eficaz às demandas dos usuários finais, enriquecendo a busca por serviços médicos na região de Araçuaí. Neste relatório, cada fase do processo de concepção e desenvolvimento do projeto será minuciosamente delineada, destacando as tecnologias empregadas para a realização bem-sucedida desse objetivo.

## 1.1 Motivação

A concepção deste projeto se origina de uma lacuna no acesso às informações pertinentes aos serviços de saúde privada em Araçuaí-MG. Conduzimos uma pesquisa com o intuito de avaliar o nível de conhecimento da população sobre as clínicas médicas particulares nessa localidade. Os resultados apontaram para um notório déficit de conhecimento, o que ressalta a necessidade de aprimorar o acesso à informações essenciais.

Durante a pesquisa, foram coletados dados de 82 participantes oriundos das comunidades de Araçuaí, Jenipapo de Minas, Virgem da Lapa, Itinga, Francisco Badaró e Berilo. Uma das descobertas centrais desta pesquisa é que 84,3% dos participantes expressou um desejo de buscar atendimento médico particular em Araçuaí-MG.

Contudo, a pesquisa revelou que, apesar do desejo de buscar atendimento médico particular, uma parcela significativa dos participantes não possuem as informações necessárias para usufruir destes serviços. Apenas 2,4% dos entrevistados afirmaram conhecer todos os médicos que atuam nas clínicas particulares, enquanto a grande maioria, 68,3%, conhece apenas alguns, e notavelmente, 29,3% não possuem nenhum conhecimento sobre os médicos atuantes.

Ademais, a falta de informações sobre as especialidades médicas disponíveis em Araçuaí MG é notória. Somente 6,1% dos participantes detêm conhecimento de todas as especialidades oferecidas, ao passo que 76,8% conhecem apenas algumas, e 17,1% desconhecem por completo as especialidades existentes.

Os desafios não se restringem à falta de conhecimento sobre os médicos e especialidades. No que diz respeito às informações de contato, 43,9% possuem o telefone de contato de apenas

algumas clínicas e 56,1% não possuem o de nenhuma. Em relação à localização, 9,8% dos participantes conhecem de todas as clínicas, 65,9% conhecem apenas de algumas e 24,4% não conhecem de nenhuma.

Adicionalmente, a pesquisa incluiu um aspecto qualitativo, no qual os participantes expressaram suas frustrações, enfatizando as dificuldades em encontrar informações sobre as clínicas e especialidades desejadas, resultando em busca ineficiente e falta de direcionamento.

Estes resultados solidificam a motivação por trás deste projeto, tendo em vista que o website desenvolvido busca preencher essas lacunas, oferecendo uma solução abrangente e amigável médicas particulares na região.

## 1.2 Objetivos

O objetivo primordial deste projeto é a criação do website "AraClin", um catálogo online que concentra informações cruciais sobre as clínicas médicas privadas situadas em Araçuaí-MG. Esse catálogo abarca informações essenciais, como localização, especialidades médicas, médicos em atuação, detalhes de contato e modalidades de pagamento. O propósito é suprir a lacuna de informações identificada durante a pesquisa, oferecendo uma plataforma de fácil navegação e intuição que viabiliza aos usuários explorar, buscar e acessar uma variedade de informações pertinentes relacionadas às clínicas na região.

Para facilitar a tomada de decisões, foi implementado um sistema de avaliação por estrelas, no qual os pacientes ou responsáveis podem expressar seu grau de satisfação em uma escala de 1 a 5 e elaborar comentários detalhados sobre suas experiências com as clínicas. Além disso, na página inicial, a listagem das clínicas foi ordenada por avaliações mais elevadas, e um campo de busca possibilita aos usuários inserir palavras-chave, como o nome da clínica, especialidade médica ou nome do médico, facilitando assim a busca por serviços que melhor atendam às suas necessidades.

Mediante essa abordagem, espera-se proporcionar aos usuários uma visão ampla e detalhada das opções de atendimento médico particular. Isso permitirá decisões embasadas em relação à saúde e ao bem-estar, ao mesmo tempo que otimiza a experiência de busca por serviços médicos particulares na cidade de Araçuaí-MG.

## 2 Documentação

A documentação que apresentamos a seguir traz uma narrativa completa e detalhada do processo de concepção e desenvolvimento do site "AraClin". Cada etapa abordada nesta documentação é um reflexo direto do conhecimento e das competências que foram adquiridas ao longo das aulas de Engenharia de Software, Programação Orientada a Objetos, Banco de Dados e Programação Web.

### 2.1 Engenharia de Software

A disciplina de Engenharia de Software desempenhou um papel fundamental no desenvolvimento do projeto, orientando desde a fase inicial de concepção até a entrega final do software. Através do fornecimento de metodologias, práticas e ferramentas, ela possibilitou a estruturação de requisitos, a modelagem de interações, a estimativa de prazos, o planejamento de atividades e a realização de testes.

Em consonância com os princípios de desenvolvimento colaborativo, o projeto incorporou o GitHub como um sistema de controle de versão de código fonte. Por meio dessa plataforma, foi possível gerenciar de forma eficiente as modificações realizadas no código, garantindo um acompanhamento detalhado do histórico de alterações ao longo de todo o processo de desenvolvimento. O repositório do projeto pode ser acessado em: <https://github.com/vertplay/integrador>.

O ponto de partida desse projeto envolveu a criação de um fluxo de planejamento, em que se estabeleceu as bases para orientação e coordenação das etapas subsequentes. Nem sempre tudo transcorreu conforme o planejado, porém, ao enfrentarmos as adversidades de maneira proativa, conseguimos ajustar o curso e manter o projeto em direção ao êxito. O Planejamento do Fluxo de Atividades para cada período está disponível na Figura 1.

Figura 1 – Planejamento do Fluxo de Atividades

FLUXO DE ATIVIDADES			
MAIO	JUNHO	JULHO	AGOSTO
Seleção de um problema;	Levantamento dos Requisitos;	Criação do Banco de Dados - Modelo Conceitual, Lógico e Físico;	Realização de Testes;
Realização de pesquisa para comprovar a relevância do problema;	Elaboração do Diagrama de Arquitetura da Informação;	Desenvolvimento do Front-end;	Correção de Erros;
Concepção de uma solução para o problema.	Criação da Identidade Visual;	Desenvolvimento do Back-end.	Elaboração do Diagrama de Casos de Uso;
	Avaliação das Heurísticas de Usabilidade.		Elaboração do Diagrama de Classes;
			Escrita do Relatório Final.

No processo de desenvolvimento de software, o documento de requisitos assume uma função

primordial. Ao englobar todos os requisitos funcionais e de qualidade essenciais para o software, este documento opera como uma ponte de comunicação entre os projetistas do sistema e os usuários, permitindo a definição consensual do escopo e das características do software desejado. (Turine; Masiero, 1996)

Dada a sua significativa relevância no ciclo de desenvolvimento, a organização metódica do documento de requisitos é essencial para otimizar a compreensão e a legibilidade dos requisitos, prevenindo potenciais problemas e erros na etapa de implementação do software (Turine; Masiero, 1996). Nesse contexto, procedemos com o levantamento dos requisitos, delineando tanto os requisitos funcionais quanto os de qualidade, os quais são apresentados detalhadamente nas figuras 2, 3 e 4.

Figura 2 – Requisitos Funcionais

REQUISITOS DO SISTEMA			
REFERÊNCIA	REQUISITO	DESCRIÇÃO	RESPONSÁVEL
RF01	Cadastro de Clínicas	No processo de cadastro de clínicas serão solicitados os seguintes dados: CNPJ; nome fantasia; fotos; descrição; endereço; horários de funcionamento; informações de contato, tais como e-mail, número de telefone, WhatsApp e Instagram; especialidades ofertadas; médicos especialistas, com seus respectivos números de registros do CRM; formas de pagamento; planos de saúde aceitos; convênios aceitos; política de cancelamento e não comparecimento; e acessibilidade. Após a inserção dos dados, o sistema efetuará uma verificação para determinar se a clínica já possui cadastro. Caso possua, essa informação será comunicada. Caso não possua, o sistema possibilitará a criação da conta e os dados serão armazenados.	Thiago Ramos de Andrade
RF02	Cadastro de Paciente ou Responsável	No processo de cadastro de pacientes ou responsáveis serão solicitados os seguintes dados: foto, CPF, RG; nome completo; gênero; data de nascimento; informações de contato, tais como e-mail, número de telefone e WhatsApp; e endereço. Após a inserção dos dados, o sistema efetuará uma verificação para determinar se o cliente já possui cadastro. Caso possua, essa informação será comunicada. Caso não possua, o sistema possibilitará a criação da conta e os dados serão armazenados.	Thiago Ramos de Andrade
RF03	Política de senha forte	Será implementado uma política de senha forte para assegurar que a senha utilizada pelo usuário atenda a critérios de segurança específicos, incluindo um comprimento mínimo de 8 caracteres; combinação de letras maiúsculas, minúsculas, números e caracteres especiais; restrição de sequências numéricas; e restrição de informações pessoais, como nome e data de nascimento.	Thiago Ramos de Andrade

Figura 3 – Requisitos Funcionais

RF04	Recuperação de Senha	O usuário poderá recuperar o acesso à sua conta caso tenha esquecido a senha. Ao fazer essa solicitação, o sistema enviará automaticamente um e-mail ao endereço de e-mail cadastrado na conta do usuário, com um link que o direcionará a uma página onde poderá redefinir sua senha.	Artur Sousa Luzzi
RF05	Login e Logout	O paciente ou responsável deverá inserir o e-mail e senha, suas credenciais de login, para ter acesso ao seu perfil e avaliar as clínicas que o atenderam. Quando desejar deixar o site, poderá realizar o logout, para que sua sessão seja encerrada com segurança. A clínica, por sua vez, deverá inserir o e-mail e senha, suas credenciais de login, para ter acesso ao seu perfil, cadastrar a agenda de seus médicos e confirmar os requerimentos de consulta. Quando desejar deixar o site, poderá realizar o logout, para que sua sessão seja encerrada com segurança. Tanto o paciente ou responsável quanto a clínica poderão acessar uma área dedicada ao gerenciamento de suas informações pessoais. Nessa seção, será possível visualizar e alterar seus dados cadastrados, para que as informações estejam sempre atualizadas. Além disso, haverá a possibilidade de encerramento de conta, que resultará na remoção permanente de todas as informações pessoais associadas à conta do usuário, caso confirme a exclusão.	Artur Sousa Luzzi



Figura 4 – Requisitos Funcionais

RF06	Configurações de perfil do usuário	Tanto o paciente ou responsável quanto a clínica poderão acessar uma área dedicada ao gerenciamento de suas informações pessoais. Nessa seção, será possível visualizar e alterar seus dados cadastrados, para que as informações estejam sempre atualizadas. Além disso, haverá a possibilidade de encerramento de conta, que resultará na remoção permanente de todas as informações pessoais associadas à conta do usuário, caso confirme a exclusão.	Artur Sousa Luzzi
RF07	Mecanismo de busca por palavras-chave	O sistema possuirá um campo de busca que permitirá aos usuários digitar palavras-chave, como o nome da clínica, a especialidade, o médico especialista e outros termos relacionados ao atendimento que procuram. As clínicas que correspondam às pesquisas realizadas serão retornadas.	Roberta Santos Vieira
RF08	Avaliação das clínicas	Ao realizar o login, o paciente ou responsável terá a oportunidade de avaliar a clínica, e sua avaliação ficará disponível para a visualização de todos. Para isso, será utilizado um sistema de estrelas, em que o paciente ou responsável expressará o seu nível de satisfação em uma escala de 1 a 5. Além disso, também poderá escrever comentários detalhados sobre sua experiência com a clínica. Por sua vez, a clínica poderá responder aos comentários.	Roberta Santos Vieira
RF09	Ordenação das clínicas por maior pontuação	Para ajudar o paciente ou responsável a fazer escolhas informadas, a lista de clínicas exibidas no catálogo será ordenada de acordo com a média da pontuação obtida nas avaliações. As clínicas com as melhores classificações serão exibidas primeiro, com sua pontuação acompanhada por um ícone de estrela.	Roberta Santos Vieira

Figura 5 – Requisitos Não Funcionais

REQUISITOS NÃO FUNCIONAIS			
REFERÊNCIA	REQUISITO	DESCRIÇÃO	RESPONSÁVEL
RF01	Integração com Whatsapp	O sistema fornecerá um link direto para a conta do WhatsApp da clínica, permitindo aos usuários entrar em contato de forma rápida e direta através do aplicativo.	Artur Sousa Luzzi
RF02	Integração com Instagram	O sistema deve disponibilizar um link direto para o perfil do Instagram da clínica, permitindo aos usuários acessar e explorar conteúdos adicionais e informações relevantes na plataforma.	Artur Sousa Luzzi

O Diagrama de Casos de Uso, dentre os diversos diagramas da UML, se destaca como o mais abstrato, flexível e informal. A principal finalidade desse diagrama é modelar as funcionalidades e os serviços que o sistema oferece. Utilizando uma linguagem simples, é possível ilustrar o comportamento externo do sistema do ponto de vista do usuário. O diagrama criado para o catálogo de clínicas médicas é apresentado na figura 5.

Após a criação do Diagrama de Casos de Uso, criamos um Diagrama de Classes. Este, por sua vez, possui a finalidade de possibilitar a representação das classes empregadas pelo sistema e suas interações, exibindo uma perspectiva estática da organização dessas classes. Seu foco é estritamente na delimitação da estrutura lógica das classes, sem adentrar nos detalhes dinâmicos. O resultado é apresentado na figura 6.

Figura 6 – Diagrama de Caso de Uso

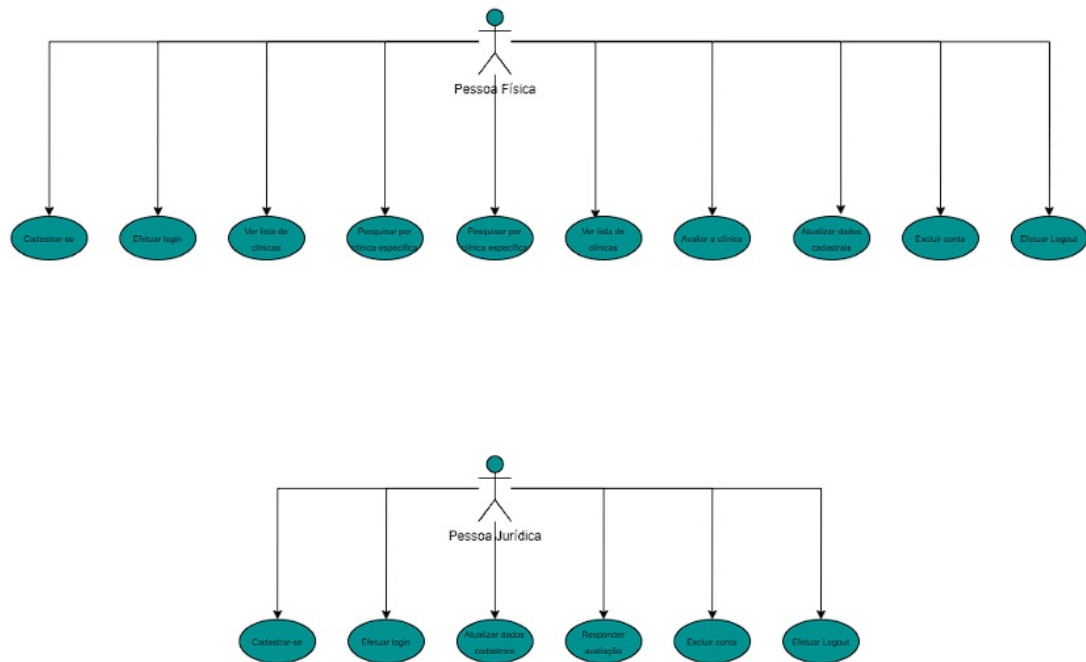
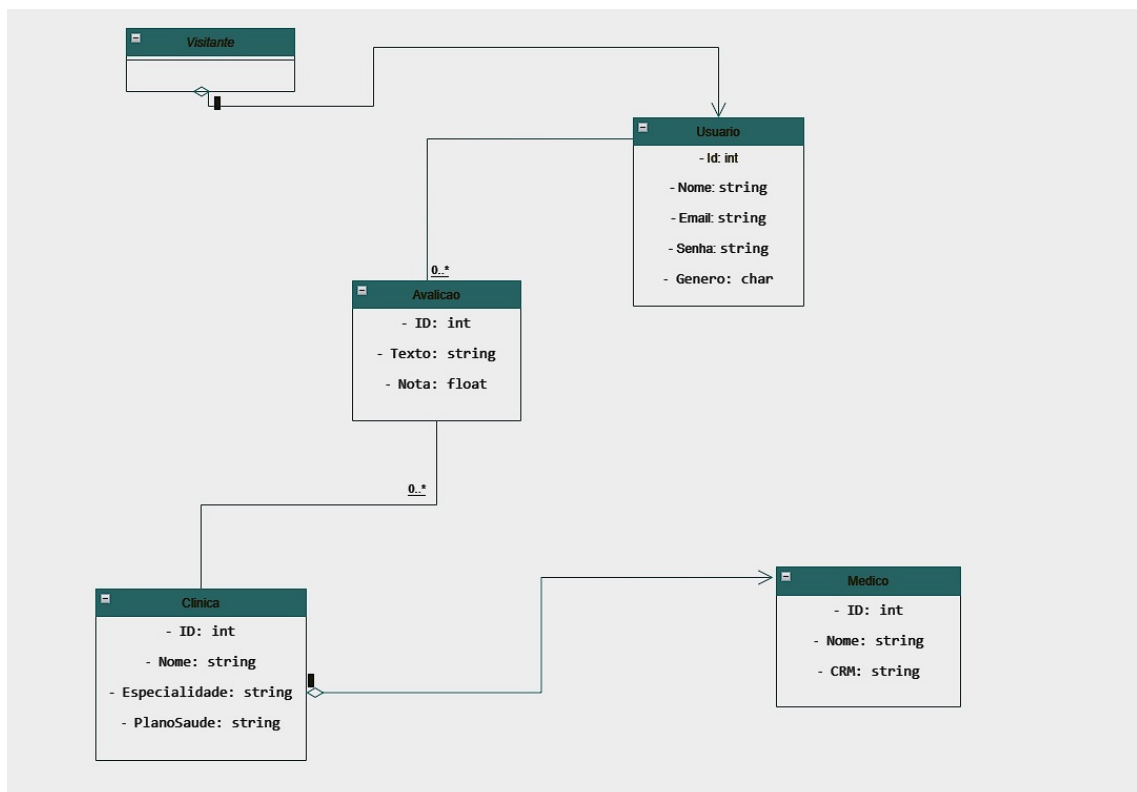


Figura 7 – Diagrama de classes



Para a avaliação do tempo essencial para a criação do nosso sistema, optamos por utilizar o método Program Evaluation Review Technique (PERT). Esse método envolve a análise de três pontos de estimativa, englobando os valores da Estimativa Otimista (O), Estimativa Mais Provável (M) e Estimativa Pessimista (P). Por meio desses dados, procedemos ao cálculo da estimativa

PERT para a tarefa, empregando a seguinte fórmula:  $PERT = P + 4M + O$  (SOUSA, 2018). O resultado obtido como estimativa da duração do desenvolvimento de cada funcionalidade se encontra na figura 7.

Figura 8 – Tempo de Desenvolvimento das Funcionalidades

TEMPO DE DESENVOLVIMENTO			
FUNÇÃO	RESPONSÁVEL	FÓRMULA $PERT = (P+4M+O) / 6$	TEMPO ESTIMADO
Cadastrar clínica	Thiago Ramos de Andrade	$(18 + 4 * 12 + 10) / 6$	12h35
Cadastrar paciente ou responsável	Thiago Ramos de Andrade	$(16 + 4 * 10 + 8) / 6$	10h35
Garantir que a senha atenda critérios de segurança	Thiago Ramos de Andrade	$(10 + 4 * 5 + 3) / 6$	05h30
Realizar login da clínica e do paciente ou responsável	Artur Souza Luzzi	$(10 + 4 * 5 + 3) / 6$	05h30
Realizar logout da clínica e do paciente ou responsável	Artur Souza Luzzi	$(10 + 4 * 4 + 3) / 6$	04h50
Recuperar senha	Artur Souza Luzzi	$(20 + 4 * 16 + 10) / 6$	15h35
Vizualizar e alterar dados cadastrais	Artur Souza Luzzi	$(12 + 4 * 10 + 8) / 6$	10h00
Encerrar a conta	Artur Souza Luzzi	$(8 + 4 * 5 + 4) / 6$	05h20
Avaliar a clínica com uma pontuação de 1 a 5 estrelas	Roberta Santos Vieira	$(15 + 4 * 10 + 8) / 6$	10h30
Adicionar um comentário na avaliação	Roberta Santos Vieira	$(15 + 4 * 10 + 8) / 6$	10h30
Responder um comentário na avaliação	Roberta Santos Vieira	$(15 + 4 * 10 + 8) / 6$	10h30
Exibir lista de clínicas ordenadas por maior média de pontuação	Roberta Santos Vieira	$(10 + 4 * 8 + 6) / 6$	08h00
Realizar busca por palavras-chave	Roberta Santos Vieira	$(10 + 4 * 8 + 6) / 6$	08h00
			<b>119h45</b>

A realização de testes rigorosos desempenha um papel crucial no desenvolvimento de qualquer sistema, garantindo a sua funcionalidade, eficácia e confiabilidade. Os testes permitem identificar possíveis falhas e assegurar que todas as funcionalidades operam conforme o esperado, resultando em uma experiência de usuário aprimorada e satisfatória. Neste contexto, os resultados dos testes realizados para avaliar os requisitos do nosso sistema estão apresentados nas figuras 8 e 9.

Figura 9 – Relatório de testes

RELATÓRIO DE TESTES			
NOME DO TESTE	DESCRIÇÃO	EXPECTATIVA	RESULTADO
Cadastro de clínicas e cadastro de pacientes ou responsáveis	O teste foi realizado com o intuito de verificar se o sistema impede o envio do formulário vazio; se permite a criação da conta, armazenando os dados adequadamente; e se impossibilita a criação de uma nova conta com dados que já foram cadastrados.	Os dados são armazenados corretamente e a conta é criada para novos usuários	Conforme expectativa.
Política de senha forte	O teste foi realizado com o objetivo de verificar se o sistema impõe a política de senha forte, exigindo um comprimento mínimo de 8 caracteres, combinação de letras maiúsculas, minúsculas, números e caracteres especiais.	O sistema exige o cumprimento da política de senha forte	Conforme expectativa.
Recuperação de Senha	O teste foi realizado com o propósito de verificar se o sistema permite a recuperação de senha por meio do envio de e-mail com um link para redefinição de senha.	O sistema envia o e-mail de recuperação com o link para redefinição de senha.	Conforme expectativa.
Login e Logout	O teste foi realizado para verificar se o sistema permite o login das clínicas e dos pacientes ou responsáveis com suas credenciais de e-mail e senha, dando acesso aos seus respectivos perfis. Também foi verificado se o sistema permite o logout de forma segura.	O sistema permite o login e logout	Conforme expectativa.
Configurações de perfil do usuário	O teste foi realizado visando verificar se o sistema permite que clínicas e pacientes ou responsáveis acessem e alterem suas informações pessoais, além de possibilitar o encerramento de conta com a remoção permanente de dados associados.	O sistema permite o acesso, alteração de informações e encerramento de conta de forma adequada.	Conforme expectativa.

Figura 10 – Relatório de testes.

Mecanismo de busca por palavras-chave	O teste foi realizado a fim de verificar se o campo de busca permite aos usuários digitar palavras-chave para pesquisar clínicas com base no nome da clínica, especialidade e médico especialista, retornando as clínicas correspondentes	O sistema permite a busca por palavras-chave e retorna as clínicas correspondentes.	Conforme expectativa.
Avaliação das clínicas	O teste foi realizado com o objetivo de verificar se o sistema permite que pacientes ou responsáveis avaliem clínicas, atribuindo uma pontuação de 1 a 5 estrelas e inserindo comentários sobre as experiências vividas. Também foi verificado se as avaliações ficam disponíveis para visualização e se as clínicas podem responder aos comentários.	O sistema permite avaliações com sistema de estrelas, comentários e respostas às avaliações.	Conforme expectativa.
Ordenação das clínicas por maior pontuação	O teste foi realizado com a finalidade de verificar se o sistema ordena corretamente a lista de clínicas exibidas no catálogo de acordo com a média da pontuação das avaliações, exibindo as clínicas com melhores classificações primeiro.	O sistema ordena a lista de clínicas conforme a média da pontuação das avaliações.	Conforme expectativa.

## 2.2 Programação Orientada a Objetos

A disciplina de Programação Orientada a Objetos (POO) foi de extrema importância para a concretização deste projeto, oferecendo conhecimentos fundamentais sobre linguagens de programação, conceitos de POO, bem como a compreensão do padrão de arquitetura Modelo-View-Controller e a utilização prática do framework CodeIgniter.

Para dar vida à nossa aplicação web, recorremos ao padrão arquitetural de projeto MVC (Model-View-Controller), uma abordagem que visa dividir e organizar de forma eficiente as diferentes funcionalidades de uma aplicação. Dentro da estrutura MVC, existem três componentes cruciais, cada qual desempenhando um papel distintivo (ANDRADE, 2011).

O Model constitui a camada que cuida dos dados da aplicação, incluindo informações como as regras de negócio que guiam o acesso e a alteração desses dados. Por meio dessa camada, gerenciam-se os dados e conduzem-se as operações fundamentais relacionadas a eles (ANDRADE, 2011).

A View concentra-se na representação dos dados e na interação com o usuário. Ela é responsável por renderizar e exibir os dados do Model de maneira adequada ao usuário. Além disso, essa camada redireciona as ações do usuário para o Controlador, permitindo a este saber quando e como agir em resposta às interações do usuário (ANDRADE, 2011).

O Controller, por sua vez, desempenha o papel de intérprete das ações do usuário, traduzindo-as em operações no Model. Atuando como uma ponte entre a Camada de View e o Model, o Controller assegura que as solicitações do usuário sejam tratadas devidamente. Com base nas ações do usuário e nos resultados obtidos do Model, o Controller determina qual visualização deve ser apresentada como resposta à solicitação do usuário, (ANDRADE, 2011).

Para aplicar o padrão MVC de forma eficaz, adotamos o framework CodeIgniter. Quando um desenvolvedor se depara com o desafio de criar uma nova aplicação, a estruturação eficiente ao longo do desenvolvimento se torna uma preocupação primordial. É aqui que os frameworks têm papel de destaque, ao proporcionar uma estrutura básica pré definida, evitando que o código se torne confuso para a manutenção posterior, (ANDRADE, 2011).

O CodeIgniter, como um framework de desenvolvimento de aplicações destinado a construir sites em PHP, é munido de um rico conjunto de bibliotecas, acelerando o desenvolvimento em comparação com a criação a partir do zero. Além de agilizar o processo, o CodeIgniter ajuda a manter o foco no desenvolvimento das ideias, reduzindo a quantidade de código necessário para executar tarefas específicas. Isso não só impulsiona o desenvolvimento, mas também estimula a manutenção e a evolução sustentável do projeto, conferindo eficiência e sustentabilidade a longo prazo, (ANDRADE, 2011).

Ao que se refere ao uso de boas práticas de Programação Orientada a Objetos, é relevante destacar que a aplicação dessas práticas foi fundamental para garantir a qualidade, eficiência e manutenibilidade do projeto. Ao seguir as boas práticas da POO, pudemos criar um código com uma estrutura coesa e bem organizada, facilitando a compreensão, colaboração entre a equipe e a introdução de futuras melhorias. A adoção de padrões de projeto, encapsulamento, herança, polimorfismo e outras técnicas de POO contribuiu para minimizar a complexidade do código, reduzir potenciais erros e agilizar o desenvolvimento, alinhando-se assim aos princípios de qualidade e excelência estabelecidos pelo projeto.

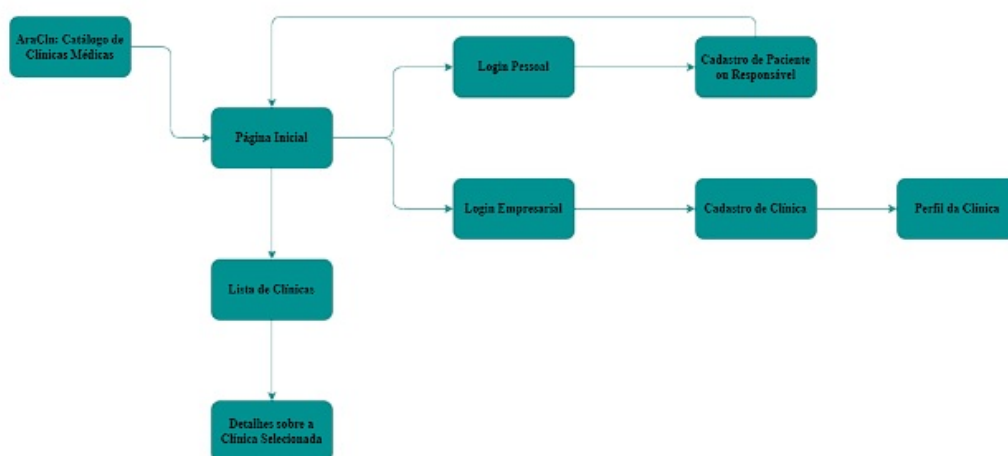
## 2.3 Programação Web

A contribuição da web no projeto abrangeu desde o desenvolvimento do diagrama de arquitetura de informação, identidade visual e incorporação de heurísticas de usabilidade até a implementação de HTML, CSS e JavaScript. Esses elementos culminaram em uma experiência do usuário mais atrativa, intuitiva e funcional.

O Diagrama de Arquitetura da Informação é uma representação visual que delineia a estrutura e organização das informações em um sistema, tornando mais claro como os elementos estão interligados e como os usuários navegam para acessar conteúdo. É uma ferramenta crucial para projetar interfaces intuitivas e facilitar a experiência do usuário ao interagir com um sistema ou site. Tendo isso em mente, desenvolvemos um diagrama de arquitetura da informação para o catálogo de clínicas médicas. O resultado pode ser visto na figura 10.



Figura 11 – Diagrama de arquitetura da informação



A identidade visual abarca um conjunto completo de elementos visuais que desempenham um papel fundamental na identificação de uma empresa ou produto. Esses elementos, que abrangem logotipos, símbolos, cores, tipografia, layouts e arranjos gráficos, são meticulosamente planejados para seguir um padrão que confere solidez e reconhecimento à marca (TEIXEIRA; SILVA; BONA, 2007).

Nesse contexto, a importância da identidade visual transcende a estética, tendo o poder de instaurar um reconhecimento instantâneo, transmitir uma mensagem visual coesa e criar uma impressão memorável na mente dos consumidores (TEIXEIRA; SILVA; BONA, 2007). Por essas razões, decidimos criar uma identidade visual para o website AraClin.

Para esse fim, foi essencial compreender as relações entre as cores e seus significados culturais, considerando atentamente o público-alvo ao qual o produto se destina. A escolha da cor desempenha um papel crucial na formação de impressões e associações. (TEIXEIRA; SILVA; BONA, 2007). Nesse sentido, a cor verde, culturalmente associada à medicina e à saúde, foi a opção natural para simbolizar esses valores.

As cores individuais têm o poder de evocar associações psicológicas e desencadear respostas emocionais naqueles que as percebem. Contudo, a combinação estratégica das cores pode enriquecer ainda mais a construção de uma identidade distintiva (TEIXEIRA; SILVA; BONA, 2007). Com esse objetivo, utilizamos a ferramenta Palletton para gerar uma paleta de cores harmônica, uma escolha fundamentada que reforça a coesão visual desejada. A Figura 11 ilustra a paleta de cores definida para esse propósito.

No processo da criação do logotipo, utilizamos um ícone disponível no Canva e uma fonte manuscrita disponível no Google Fonts. O resultado desse processo é evidenciado na Figura 12.

Após a definição da identidade visual, o próximo passo consistiu em realizar uma análise minuciosa das heurísticas de usabilidade propostas por Jakob Nielsen e Rolf Molich em 1990. O intuito desse processo foi estabelecer diretrizes essenciais a serem seguidas durante o

Figura 12 – Paleta de cores

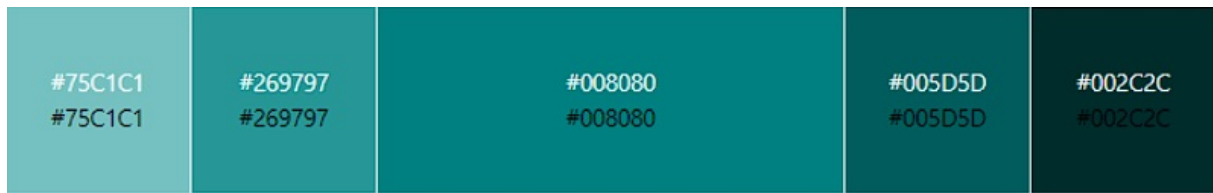


Figura 13 – Logo



desenvolvimento do site, com o objetivo de proporcionar aos usuários uma experiência mais satisfatória.

A incorporação dessas heurísticas revela-se como um meio eficaz de conferir ao ambiente digital uma funcionalidade não apenas pragmática, mas também intuitiva, eficiente e livre de frustrações para seus usuários. Nesse contexto, nossa atenção concentrou-se na integração das seguintes heurísticas em nosso sistema:

- **Controle e Liberdade para o Usuário:** Fornecemos aos usuários a capacidade de remover seus próprios comentários em avaliações, permitindo a correção de enganos e, consequentemente, aprimorando a sensação de segurança ao interagir com as funcionalidades do sistema.
- **Prevenção de Erros:** Para evitar ações irreversíveis, implementamos um mecanismo de confirmação antes da exclusão definitiva de uma conta de usuário. Essa precaução adicional mitigará erros acidentais.
- **Estética e Design Minimalista:** Com foco na clareza e na centralização de informações vitais, nossa interface adota um design minimalista, proporcionando uma experiência de usuário mais limpa e direcionada.
- **Consistência e Padronização:** Adotamos uma abordagem uniforme em relação ao layout e aos elementos de navegação em todas as telas, com o objetivo de garantir uma experiência coesa e natural, promovendo uma usabilidade intuitiva e confortável.
- **Ajude os Usuários a Reconhecerem, Diagnosticarem e Recuperarem-se de Erros:** Implementamos a validação do CPF, alertando os usuários sobre caracteres incorretos e fornecendo a oportunidade de correção. Além disso, notificamos quando campos obrigatórios não são preenchidos e quando uma senha escolhida não atende aos requisitos

mínimos de segurança. Dessa forma, facilitamos o reconhecimento e a resolução de erros, melhorando a experiência global do usuário.

Na etapa de programação, utilizamos o HTML como linguagem de marcação. O trecho de código da figura 13 ilustra a estrutura básica do site, composta por um bloco contendo o menu de navegação, o bloco central onde o conteúdo das páginas será exibido e, por último, o rodapé.

Figura 14 – Utilização do HTML para estrutura básica do site

```
<title>AraClin</title>
</head>
<body>
    <!--MENU-->
    <nav id="menu-bar">
        <a id="home-link" href="{<?=-base_url()?>}">AraClin</a>
        <?php
            if($session->has('ID_clinica') && $session->get('Nome_fantasia_clinica') != null){
                echo '
                <div id="btn-group">
                    <a href="{.base_url('pe/perfil')}" id="menu-btn-emp">Perfil</a>
                    <form action="{.base_url('/pe/logout')}" method="post">
                        <button type="submit" name="sair">LogOut</button>
                    </form>
                </div>

                ';
            }
            else{
                echo'
                <div id="btn-group">
                    <a href="{.base_url('pp/login')}" id="menu-btn-user">Perfil Pessoal</a>
                    <a href="{.base_url('pe/login')}" id="menu-btn-emp">Perfil Empresarial</a>
                </div>

                ';
            }
        ?>
    </nav>
    <!--Bloco central-->
    <div id="container-principal">
        <?= $this->renderSection('content') ?>
    </div>
    <!--FOOTER-->
    <footer>
        &copy;2023
    </footer>
</body>
```



A estrutura do site foi moldada por meio do uso de DIVs estilizadas através do CSS, de modo a garantir a flexibilidade do layout. O trecho de CSS na figura 14 é utilizado para definir a forma do menu, destacando-se:

- “position:fixed”: Garante que o menu permanecerá em uma posição fixa em relação à tela do usuário, independente da utilização do scroll para navegar.
- “background-image...”: Permite usar imagens como plano de fundo dos elementos, foi utilizado para anexar imagens aos links do menu.

Utilização do CSS:

Figura 15 – Utilização do CSS na barra de navegação

```
#menu-bar{/*BARRA DE MENU*/
  min-width:750px;
  width:100%;
  height:var(--menu-height);
  position:fixed;
  top:0;
  background-color:var(--menu-color);
  color:□ #000000;
  border-bottom-width:4px;
  border-bottom-style:solid;
  border-bottom-color:□ #005d5d;
  z-index:20;
  transition:0.3s;
}
#menu-bar > #home-link{/*LINKS DO MENU -> ('HOME')*/
  font-size:38px;
  color:var(--menu-font-color);
  transition:0.3s;
  font-family:"Sofia", sans-serif;
  font-style:italic;
  font-weight: normal;
  float:left;
  width:300px;
  height:100%;
  padding:22px 0px 0px 80px;
  margin-left:40px;
  background-image:url("../img/logo-darker.png");
  background-repeat: no-repeat;
  background-attachment:scroll;
  background-position-x:10%;
  background-position-y:85%;
  background-size:50px;
}
#menu-bar > #home-link:hover{
  color:□ #000000;
  text-decoration:underline 2px;
}
#btn-group{/* MENU -> link de login/registo*/
  width:320px;
  height:100%;
  float:right;
}
```

Por outro lado, conforme observado na figura 15, o bloco central da página possui menos atributos para tomar forma, sendo os seguintes os mais relevantes:

- `margin-top:var(--menu-height);`: O bloco possui uma margem em relação ao topo da página, equivalente à altura do menu de navegação, para evitar que um elemento fique sobre o outro.
- `"min-height"`: Define a altura mínima do elemento, usado para manter a página visualmente agradável mesmo sem possuir conteúdo algum para preencher a interface.

Figura 16 – Utilização do CSS no bloco principal

```
/*-----BLOCO CENTRAL-----*/  
#container-principal{/*BLOCO PRINCIPAL (NO CENTRO)*/  
  float:left;  
  width:100%;  
  margin-top:var(--menu-height);  
  background-color:var(--background-color) ;  
  min-height:calc(100vh - 160px);  
  border-width:0px;  
}
```

Com o intuito de reforçar a segurança e a integridade dos dados dos usuários, implementamos o JavaScript em um formulário HTML para avaliar a robustez das senhas inseridas antes de serem submetidas. O segmento de código está ilustrado na figura 16.

Figura 17 – Utilização do JavaScript para validação de senha

```
window.onload = function() {  
  var userform = document.getElementById("userform");  
  userform.onsubmit = function() {  
    // Verificar os critérios da senha  
    var senha = document.getElementById("formsenha").value;  
  
    const temNumero = /[0-9]/.test(senha);  
    const temLetraMaiuscula = /[A-Z]/.test(senha);  
    const temLetraMinuscula = /[a-z]/.test(senha);  
    const temCaractereEspecial = /[!@#$%^&*()_+{}\\[\];<>,.?~]/.test(senha);  
  
    // Verificar se a senha atende a todos os critérios  
    const senhaValida =  
      temNumero &&  
      temLetraMaiuscula &&  
      temLetraMinuscula &&  
      temCaractereEspecial &&  
      senha.length >= 8; // A senha precisa ter pelo menos 8 caracteres  
  
    if (!senhaValida) {  
      var alertaSenha = document.getElementById("alerta_senha");  
      alertaSenha.innerHTML = "A senha não atende aos critérios:<br>" +  
        "- Deve conter pelo menos um número<br>" +  
        "- Deve conter pelo menos uma letra maiúscula<br>" +  
        "- Deve conter pelo menos uma letra minúscula<br>" +  
        "- Deve conter pelo menos um caractere especial<br>" +  
        "- Deve ter no mínimo 8 caracteres";  
      scrollTo(0,0);  
      return false;  
    }  
  
    if(senhaValida){  
      var alertaSenha = document.getElementById("alerta_senha");  
      alertaSenha.innerHTML = " ";  
      return true;  
    }  
  }  
}
```

Esse código possui a seguinte estrutura lógica:

- “window.onload = function() ...”: Isso garante que o código JavaScript seja executado após a página HTML ser completamente carregada.
- “var userform = document.getElementById("userform");”: Essa linha seleciona o elemento HTML do formulário com o ID "userform" e o atribui à variável userform.
- “userform.onsubmit = function() ...”: Aqui um manipulador de eventos é definido para o evento de envio do formulário. Quando o formulário for enviado, a função fornecida será executada.

- “var senha = document.getElementById("formsenha").value;”: Essa linha obtém o valor da senha inserida pelo usuário no campo de entrada com o ID "formsenha" e armazena esse valor na variável senha.
- As quatro constantes seguintes, “temNumero”, “temLetraMaiuscula”, “temLetraMinuscula” e “temCaractereEspecial”, usam expressões regulares (regex) para verificar se a senha atende a diferentes critérios de complexidade: conter pelo menos um número, uma letra maiúscula, uma letra minúscula e um caractere especial, respectivamente
- const “senhaValida = ...”: Nesta linha, todos os critérios de validação são agrupados, além de verificar se a senha possui pelo menos 8 caracteres. A variável “senhaValida” receberá “true” se todos os critérios forem atendidos, caso contrário, receberá “false”.

A partir daqui, o código começa a lidar com os resultados da validação da senha:

- Se a senha não for válida, ele exibe uma mensagem de alerta com os critérios que não foram atendidos e retorna false. A página também é rolada para o topo para exibir o alerta.
- Se a senha for válida, a mensagem de alerta é limpa e a função retorna true.

No geral, o código visa melhorar a segurança da senha, garantindo que ela atenda a critérios mínimos de complexidade antes de permitir o envio do formulário.

Posteriormente, foi utilizado o PHP para criação de uma função denominada "validarSenha", que tem por finalidade examinar se uma senha cumpre determinados requisitos de segurança, como ter pelo menos um número, uma letra maiúscula, uma letra minúscula e um caractere especial. O trecho do código pode ser visualizado nas imagens 17 e 18.

Figura 18 – Utilização de PHP

```
public function validarSenha($senha) {  
    $temNumero = preg_match('/[0-9]/', $senha);  
    $temLetraMaiuscula = preg_match('/[A-Z]/', $senha);  
    $temLetraMinuscula = preg_match('/[a-z]/', $senha);  
    $temCaractereEspecial = preg_match('/[!@#$%^&*()_+{}\\[\]:;<>,.?~\|\/]', $senha);  
  
    if ($temNumero && $temLetraMaiuscula && $temLetraMinuscula && $temCaractereEspecial) {  
        return true; // A senha atende aos requisitos de validação  
    } else {  
        return false; // A senha não atende aos requisitos de validação  
    }  
}
```

Figura 19 – Utilização de PHP

```

if (!$this->validarSenha($senha)) {
    $this->session->setFlashdata('error_message', 'A senha não atende aos requisitos de segurança.');
```

*(The image shows a PHP code snippet for password validation. It includes a function call to setFlashdata, an echo statement for an alert, and a JavaScript interval for a countdown timer. The code is wrapped in a dark-themed code editor window.)*

```

    echo view("clinicas/erro");
    echo '<div class="alert alert-danger">A senha não atende os requisitos de segurança. Você será redirecionado em breve...</div>';
    echo '<script>
        var seconds = 5; // Tempo de espera em segundos
        var message = document.querySelector(".alert-danger");
        message.innerHTML += " Aguarde " + seconds + " segundos...";
        setInterval(function() {
            seconds--;
            if (seconds > 0) {
                message.innerHTML = "A senha não atende os requisitos de segurança. Aguarde " + seconds + " segundos...";
            } else {
                window.location.href = "'.base_url().'/pe/registro";
            }
        }, 1000);
    </script>';
    return;
}

```

A seguir, apresentaremos uma explicação detalhada do código:

- `preg_match('/[0-9]/', $senha);`: Nesta linha, a função `preg_match` é usada para procurar por um padrão de caractere dentro da senha. O padrão `/[0-9]/` busca por qualquer dígito numérico de 0 a 9 na senha. A função retorna 1 se o padrão for encontrado e 0 caso contrário. O resultado é atribuído à variável `$temNumero`.
- `preg_match('/[A-Z]/', $senha);`: Aqui, a função `preg_match` é usada novamente, mas desta vez o padrão `/[A-Z]/` busca por qualquer letra maiúscula na senha. O resultado é atribuído à variável `$temLetraMaiuscula`.
- `preg_match('/[a-z]/', $senha);`: Da mesma forma que nas linhas anteriores, essa linha verifica se a senha contém pelo menos uma letra minúscula, utilizando o padrão `/[a-z]/`. O resultado é atribuído à variável `$temLetraMinuscula`.
- `preg_match('/[!@#%$^&*()_+{\}\[\] ;<> , . ? ~ \ / ]/', $senha);`: Novamente, é usado `preg_match`, mas desta vez para verificar se a senha contém pelo menos um caractere especial. O padrão `/[!@#%$^&*()_+{\}\[\] ;<> , . ? ~ \ / ]/` procura por qualquer um dos caracteres especiais listados na senha. O resultado é atribuído à variável `$temCaractereEspecial`.
- O código verifica se todas as quatro variáveis (`$temNumero`, `$temLetraMaiuscula`, `$temLetraMinuscula` e `$temCaractereEspecial`) são verdadeiras. Se todas forem verdadeiras, isso significa que a senha atende a todos os requisitos de validação.

Se todas as condições forem atendidas, a função retorna `true`, indicando que a senha é considerada válida. Caso contrário, ou seja, se pelo menos uma das condições não for atendida, a função retorna `false`, indicando que a senha não atende aos requisitos de validação.

O objetivo geral do código é garantir que as senhas fornecidas atendam a critérios mínimos de segurança, exigindo a inclusão de números, letras maiúsculas, letras minúsculas e caracteres

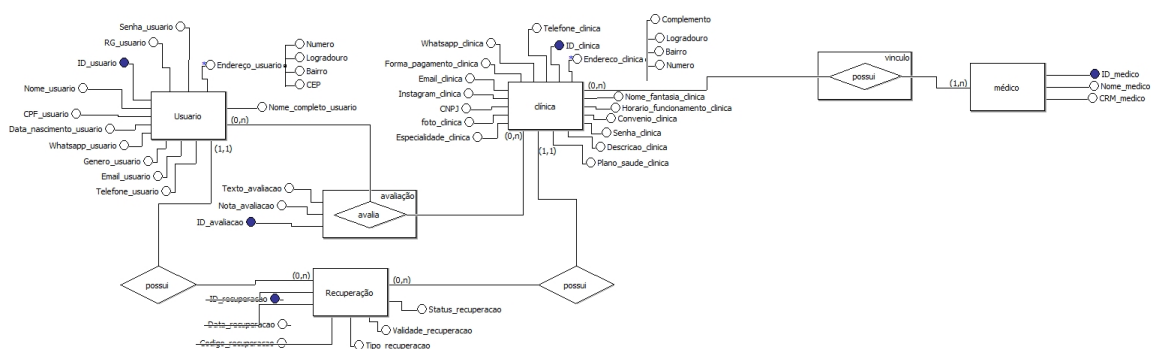
especiais. No entanto, é importante notar que a definição de requisitos de segurança para senhas pode variar de acordo com as políticas de segurança da aplicação ou sistema em que o código será utilizado.

## 2.4 Banco de dados II

A disciplina de Banco de Dados II desempenhou um papel de grande relevância no desenvolvimento deste projeto, fornecendo as bases e as habilidades necessárias para a criação e gestão eficaz de um sistema de armazenamento de dados. Ao abordar a criação do modelo conceitual, lógico e físico, bem como a realização de inserções e consultas no banco de dados utilizando a linguagem SQL (Structured Query Language), a disciplina proporciona a estrutura essencial para garantir a integridade e a organização dos dados.

Dentro do escopo do projeto de banco de dados, duas fases de modelagem são frequentemente consideradas: o modelo conceitual e o modelo lógico. Na fase de modelagem conceitual é concebida uma descrição independente de implementação do banco de dados, capturando as necessidades organizacionais sem se preocupar com os detalhes de armazenamento em um Sistema de Gerenciamento de Banco de Dados (SGBD) específico (Heuser, 1998). Nessa etapa, adotamos a abordagem entidade-relacionamento para construir um modelo conceitual materializado por meio de um diagrama entidade-relacionamento. A Figura 19 exibe o modelo conceitual elaborado para o catálogo de clínicas médicas.

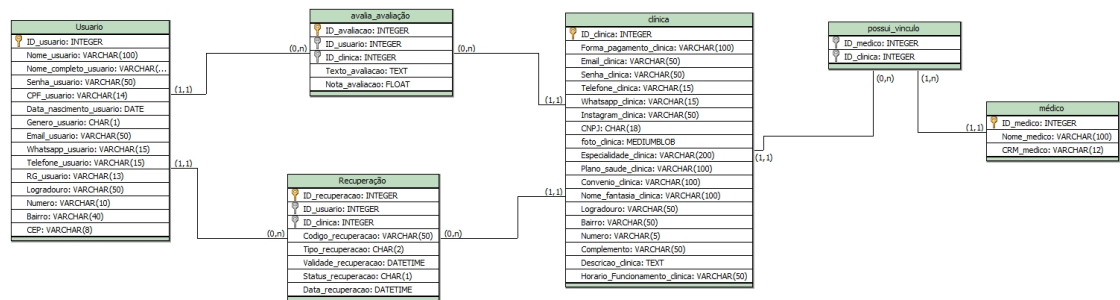
Figura 20 – Modelo Conceitual Entidade-Relacionamento



Na fase de modelagem lógica, o modelo conceitual é transformado em um modelo que define como o banco de dados será concretizado em um SGBD específico. Ao considerar as particularidades do SGBD, as informações são organizadas de acordo com a estrutura adotada, tal como a disposição em tabelas de um SGBD relacional. O produto resultante é um modelo lógico que espelha a organização do banco de dados no nível percebido pelo usuário do sistema (Heuser 1998). A Figura 20 apresenta o modelo lógico relacional gerado a partir do modelo conceitual para este projeto.



Figura 21 – Modelo Lógico



Após a conclusão do modelo conceitual, avançamos para a etapa de criação do modelo físico do banco de dados. Essa transformação do abstrato para o concreto é fundamental para garantir a funcionalidade e a eficiência do sistema, permitindo que os dados sejam armazenados, recuperados e manipulados de maneira coerente e organizada.

Nesse estágio crucial do projeto, as entidades, relacionamentos e atributos definidos no modelo lógico foram concretizados em estruturas reais no banco de dados. As figuras a seguir apresentam de forma visual como o banco de dados foi efetivamente implementado, exibindo as tabelas, colunas, chaves primárias e estrangeiras, bem como as relações entre elas.

Figura 22 – Criação do Banco de Dados

```
CREATE DATABASE araclin;
USE araclin;
```

Figura 23 – Criação da tabela "Recuperacao"

```
CREATE TABLE Recuperacao (
  ID_recuperacao INTEGER PRIMARY KEY AUTO_INCREMENT,
  ID_usuario INTEGER DEFAULT NULL,
  ID_clinica INTEGER DEFAULT NULL,
  Codigo_recuperacao VARCHAR(50) NOT NULL,
  Tipo_recuperacao CHAR(2) NOT NULL,
  Validade_recuperacao DATETIME NOT NULL,
  Status_recuperacao CHAR(1) NOT NULL,
  Data_recuperacao DATETIME NOT NULL
);
```



Figura 24 – Criação da tabela "avalia\_avaliacao"

```
CREATE TABLE avalia_avaliacao (  
  ID_avaliacao INTEGER PRIMARY KEY AUTO_INCREMENT,  
  ID_usuario INTEGER NOT NULL,  
  ID_clinica INTEGER NOT NULL,  
  Texto_avaliacao TEXT NOT NULL,  
  Nota_avaliacao FLOAT NOT NULL  
);
```

Figura 25 – Criação da tabela clínica.

```
CREATE TABLE clinica (  
  ID_clinica INTEGER PRIMARY KEY AUTO_INCREMENT,  
  Forma_pagamento_clinica VARCHAR(100) NOT NULL,  
  Email_clinica VARCHAR(50) NOT NULL,  
  Senha_clinica VARCHAR(50) NOT NULL,  
  Telefone_clinica VARCHAR(15) NOT NULL,  
  Whatsapp_clinica VARCHAR(15),  
  Instagram_clinica VARCHAR(50),  
  CNPJ CHAR(18) NOT NULL,  
  foto_clinica MEDIUMBLOB NOT NULL,  
  tipo_de_imagem_clinica VARCHAR(10) NOT NULL,  
  Especialidade_clinica VARCHAR(200) NOT NULL,  
  Plano_saude_clinica VARCHAR(100) NOT NULL,  
  Convenio_clinica VARCHAR(100) NOT NULL,  
  Nome_fantasia_clinica VARCHAR(100) NOT NULL,  
  Logradouro VARCHAR(50) NOT NULL,
```

Figura 26 – Continuação da tabela "clínica"

```
  Bairro VARCHAR(50) NOT NULL,  
  Numero VARCHAR(5) NOT NULL,  
  Complemento VARCHAR(50),  
  Descricao_clinica TEXT NOT NULL,  
  Cep VARCHAR(11) NOT NULL  
);
```

Figura 27 – Criação da tabela "médico"

```
CREATE TABLE medico (
  ID_medico INTEGER PRIMARY KEY AUTO_INCREMENT,
  Nome_medico VARCHAR(100) NOT NULL,
  CRM_medico VARCHAR(12) NOT NULL
);
```

Figura 28 – Criação da tabela "possui\_vinculo"

```
CREATE TABLE possui_vinculo (
  ID_medico INTEGER NOT NULL,
  ID_clinica INTEGER NOT NULL
);
```

Figura 29 – Criação da tabela "usuário"

```
CREATE TABLE Usuario (
  ID_usuario INTEGER PRIMARY KEY AUTO_INCREMENT,
  Nome_usuario VARCHAR(100) NOT NULL,
  Nome_completo_usuario VARCHAR(100) NOT NULL,
  Senha_usuario VARCHAR(50) NOT NULL,
  CPF_usuario VARCHAR(14) NOT NULL,
  Data_nascimento_usuario DATE NOT NULL,
  Genero_usuario CHAR(1) NOT NULL,
  Email_usuario VARCHAR(50) NOT NULL,
  Whatsapp_usuario VARCHAR(15),
  Telefone_usuario VARCHAR(15),
  RG_usuario VARCHAR(13) NOT NULL,
  Logradouro VARCHAR(50),
  Numero VARCHAR(10),
  Bairro VARCHAR(40),
  CEP VARCHAR(8));
```

Figura 30 – Inserção de dados em PHP.

```
public function cadastrarClinica($parametros) : array{
    $erros = [];

    $query = 'INSERT INTO `clinica`(`forma_pagamento_clinica`, `Email_clinica`, `Senha_clinica`, `Telefone_clinica`, `Whatsapp_clinica`, `Instagram_clinica`,
    `CNPJ`, `foto_clinica`, `tipo_de_imagem_clinica`, `Especialidade_clinica`, `Plano_saude_clinica`, `Convenio_clinica`, `Nome_fantasia_clinica`, `Logradouro`, `Bairro`, `Numero`,
    `Complemento`, `Descricao_clinica`, `Cep`)
    VALUES(:forma_pagamento:, :email:, :senha:, :telefone:, :whatsapp:, :instagram:, :cnpj:, :img:, :imgtype:, :especialidade:, :plano_saude:,
    :convenio:, :nome:, :logradouro:, :bairro:, :numero:, :complemento:, :descricao:, :cep: )';

    $dados = $this->db->query($query, $parametros);
}
```

Figura 31 – Alterações nas tabelas.

```
ALTER TABLE Recuperacao ADD FOREIGN KEY(ID_usuario) REFERENCES Usuario (ID_usuario) ON DELETE CASCADE ON UPDATE CASCADE;  
ALTER TABLE Recuperacao ADD FOREIGN KEY(ID_clinica) REFERENCES clinica (ID_clinica) ON DELETE CASCADE ON UPDATE CASCADE;  
ALTER TABLE avalia_avalicao ADD FOREIGN KEY(ID_usuario) REFERENCES Usuario (ID_usuario) ON DELETE CASCADE ON UPDATE CASCADE;  
ALTER TABLE avalia_avalicao ADD FOREIGN KEY(ID_clinica) REFERENCES clinica (ID_clinica) ON DELETE CASCADE ON UPDATE CASCADE;
```

Durante essa fase, a linguagem SQL foi utilizada para inserir dados no banco de dados, conforme demonstrado na figura 29. Além disso, o comando `ALTER TABLE` foi empregado para modificar tabelas. Especificamente no trecho apresentado na figura 30, o uso do `ON DELETE CASCADE` e `ON UPDATE CASCADE` no `ALTER TABLE` garante integridade referencial, permitindo exclusões e atualizações automáticas em registros filhos quando os pais são modificados ou removidos. Isso mantém a consistência dos dados e fortalece a confiabilidade do sistema.

O uso do Banco de Dados físicos foi recorrente durante todo o desenvolvimento do website. Um dos requisitos essenciais do nosso sistema é a capacidade de realizar buscas por palavras-chave, proporcionando aos usuários a flexibilidade de conduzir pesquisas específicas relacionadas a clínicas, médicos ou especialidades, de acordo com suas necessidades particulares. Para atender a esse objetivo, os conhecimentos adquiridos na disciplina de Banco de Dados II desempenharam um papel de destaque e relevância.

A concretização desses conhecimentos envolveu a aplicação prática do operador `LIKE` para realizar correspondência de padrões parciais nos campos de texto das tabelas, como o nome das clínicas, as especialidades e os nomes dos médicos. Dessa forma, quando um usuário insere um termo de pesquisa na função pesquisar, o operador `LIKE` é empregado para verificar se o termo fornecido aparece em qualquer parte dos campos relevantes das tabelas. Isso significa que, mesmo que o usuário insira apenas uma parte do nome da clínica, especialidade ou médico, o sistema será capaz de identificar correspondências que contenham aquela parte específica do termo.

Adicionalmente, a cláusula `INNER JOIN` foi utilizada para estabelecer associações entre as tabelas `clinica`, `avalia_avalicao`, `possui_vinculo` e `medico`. Isso é necessário para garantir que as avaliações possam ser agregadas às clínicas com base em seus IDs correspondentes, bem como para vincular as clínicas aos médicos responsáveis. Dessa forma, os resultados da pesquisa incluirão apenas as clínicas que possuem avaliações e vínculos com médicos.

A cláusula `GROUP BY`, por sua vez, foi empregada para agrupar os resultados com base nos IDs e nomes de fantasia das clínicas, permitindo a obtenção da média de avaliações para cada entidade clínica. Por fim, a cláusula `ORDER BY` foi utilizada para classificar as clínicas de acordo com a média de avaliação, em ordem decrescente. Isso garante que as clínicas com avaliações mais positivas sejam apresentadas de forma proeminente nos resultados da pesquisa. A figura abaixo ilustra a função que foi desenvolvida

Figura 32 – Utilização de cláusulas variadas.

```
public function pesquisar($termo) {  
    $db = \Config\Database::connect();  
    $dados = $this->db->query("  
        SELECT c.ID_clinica, c.Nome_fantasia_clinica, AVG(a.Nota_avaliacao) AS Media_Avaliacao  
        FROM clinica c  
        LEFT JOIN avalia_avaliacao a ON c.ID_clinica = a.ID_clinica  
        LEFT JOIN possui_vinculo pv ON c.ID_clinica = pv.ID_clinica  
        LEFT JOIN medico m ON pv.ID_medico = m.ID_medico  
        WHERE c.Nome_fantasia_clinica LIKE '%$termo%'  
        OR c.Especialidade_clinica LIKE '%$termo%'  
        OR m.Nome_medico LIKE '%$termo%'  
        GROUP BY c.ID_clinica, c.Nome_fantasia_clinica  
        ORDER BY Media_Avaliacao DESC  
    ")->getResultArray();  
    $db->close();  
    return $dados;  
}
```

### 3 Tecnologias utilizadas

Durante a trajetória deste projeto, utilizamos um conjunto diversificado de ferramentas tecnológicas. O Visual Studio Code, uma ambiente de desenvolvimento integrado (IDE), emergiu como a peça central para a criação e edição de código, oferecendo eficiência e recursos avançados que agilizaram o processo de codificação e depuração.

O GitHub, por sua vez, desempenhou um papel essencial como plataforma central para a colaboração remota. Atuando como repositório do código-fonte do sistema, o GitHub facilitou a colaboração organizada entre membros da equipe, com cada alteração registrada como um "commit", permitindo o rastreamento detalhado das mudanças e a contribuição coletiva.

Também é importante mencionar a adoção do CodeIgniter. Aliado ao padrão MVC, o framework trouxe uma estrutura sólida para o desenvolvimento do website. A clara separação entre Model, View e Controller não apenas organizou o código, mas também otimizará a manutenção e a escalabilidade do sistema futuramente.

Além disso, o XAMPP desempenhou um papel vital como ambiente de desenvolvimento. Com sua combinação de servidores Apache, MySQL e PHP, o XAMPP ofereceu um ambiente local para a construção, testes e ajustes do sistema, eliminando a dependência de um servidor externo.

Por fim, o draw.io foi uma ferramenta crucial na criação de diagramas, enquanto os ícones do Google e o Canva contribuíram significativamente para aprimorar a experiência do usuário, adicionando um toque visualmente atrativo e intuitivo ao sistema. Juntas, essas tecnologias formaram um conjunto integrado e poderoso que permitiu a criação do site AraClin: Catálo de Clínicas Médicas

## 4 Conclusões

Com a abordagem descrita ao longo deste relatório, alcançamos o desenvolvimento bem-sucedido do website "AraClin". Embora não tenha sido lançado para o público em geral, esse projeto representou uma oportunidade valiosa para aplicarmos as teorias e técnicas que absorvemos em nossas salas de aula em um contexto genuíno e desafiador. Enfrentamos uma série de obstáculos ao longo de todo o processo de concepção e desenvolvimento, cada um deles exigindo uma abordagem criativa e dedicada para superá-los.

Ao fim, esses desafios não foram apenas superados, mas também nos proporcionaram uma rica experiência de aprendizado. Os benefícios obtidos transcendem o âmbito acadêmico, pois fomos instigados a aprimorar nossas habilidades profissionais. Aprendemos a colaborar eficazmente como uma equipe coesa, a aplicar nosso conhecimento técnico de forma pragmática, a gerenciar o projeto de forma eficiente e a implementar as melhores práticas de programação para garantir a funcionalidade e confiabilidade do site.

Este projeto não apenas solidificou nossa compreensão dos conceitos teóricos, mas também nos permitiu experimentar em primeira mão como aplicar esses conceitos em um cenário real. As lições aprendidas e as habilidades adquiridas durante esse processo são inestimáveis e certamente nos servirão bem em nossas futuras carreiras profissionais. Portanto, apesar de não termos lançado o site publicamente, consideramos esse projeto como um marco significativo em nossa jornada educacional e profissional, demonstrando nossa capacidade de transformar conhecimento acadêmico em realizações tangíveis.

# Referências

Andrade, F. F. d. *Desenvolvimento de aplicações web com a utilização dos frameworks CodeIgniter e Doctrine*. 2011.

Andrade, T. R., Luzzi, A. S., and Vieira, R. S. Clínicas médicas em araguaí-mg. Questionário, 2023. 82 participantes.

Heuser, C. A. *Projeto de Banco de Dados*. No. 4 in Série Livros Didáticos. Instituto de Informática da UFRGS, Sagra Luzzato, Porto Alegre, 2001.

Moma, G. 10 heurísticas de nielsen para o design de interface, 2017.

Sousa, F. C. d. Análise de técnicas para estimativa de esforço de tempo em projeto de software desenvolvidos na fábrica de tecnologias turing - unievangélica, 2018.

Teixeira, F. C., Silva, R. D.-V. d. O. e., and Bona, R. J. O processo de desenvolvimento de uma identidade visual. In *VIII Congresso Brasileiro de Ciências da Comunicação da Região Sul* (Passo Fundo, 2017).

Turine, M. A. S., and Masiero, P. C. Especificação de requisitos: uma introdução. Tech. rep., Relatórios Técnicos do ICMC. Universidade de São Paulo, São Carlos, 1996.

(??) (??) (??) (??) (??) (??) (??)