

Home&Help: Desenvolvimento de um sistema web para assistência residencial

Murilo Henrique de Oliveira Moreira, Daiane Mastrangelo Tomazeti

¹Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP) – Câmpus Hortolândia – São Paulo - SP - Brasil

`murilo.moreira@aluno.ifsp.gov.br, daianetomazeti@ifsp.edu.br`

Abstract. *Technological advances have driven the creation of digital platforms to facilitate the hiring of maintenance professionals, such as plumbers and electricians. This work proposes the development of a web-based home assistance system using PHP for data processing, HTML for structuring, CSS for styling, JavaScript for interactive features, and MySQL for storing information. The system aims to optimize the hiring of services, offering evaluation of providers, search by specialty, user profiles, and information security. The proposal seeks greater efficiency and appreciation of professionals in the residential market. Therefore, it can be stated that the results obtained met the proposed expectations, providing significant learning aligned with the skills developed in the Systems Analysis and Development course.*

Resumo. *O avanço tecnológico tem impulsionado a criação de plataformas digitais para facilitar a contratação de profissionais de manutenção, como encanadores e eletricitas. Este trabalho propõe o desenvolvimento de um sistema web de assistência residencial, utilizando PHP para o processamento de dados, HTML para estruturação, CSS para estilização, JavaScript para funcionalidades interativas e MySQL para o armazenamento de informações. O sistema visa otimizar a contratação de serviços, oferecendo avaliação de prestadores, busca por especialidades, perfis de usuários e segurança das informações. A proposta busca maior eficiência e valorização dos profissionais no mercado residencial. Com isso, pode-se afirmar que os resultados obtidos atenderam às expectativas propostas, proporcionando um aprendizado significativo e alinhado às competências desenvolvidas no curso de Análise e Desenvolvimento de Sistemas.*

1. Introdução

O progresso tecnológico tem gerado muitas mudanças e inovações em diferentes esferas, e o setor de serviços residenciais não é uma exceção. Com a procura exigente de soluções práticas e eficientes para a contratação de funcionários de manutenção, como encanadores e eletricitas, têm sido desenvolvidas várias plataformas *online* especializadas. Por exemplo, o serviço “marido de aluguel” em São Paulo cresceu consideravelmente, à medida que as pessoas procuram desesperadamente alternativas ao desemprego e empregos mal remunerados (Exame, 2023). Este artigo visa criar um sistema *web* de assistência residencial, destinado a unir clientes e provedores de serviço, para o qual será realizada uma análise de plataformas existentes e objetivos selecionados para as necessidades do mercado brasileiro.

Assim, um sistema de assistência residencial do tipo *web* atenderia a necessidade real de otimização para demanda e oferta de contratação de serviços, como foi demonstrado na pesquisa (Exame, 2023). Por exemplo, a plataforma concebida por Oliveira (2023), demonstra

que é perfeitamente possível criar um sistema que atua como central de vagas para atender demandantes de assistência residencial e a prestadores de serviço, agilizando e facilitando em muito a vida de ambas as partes. Outro exemplo interessante é do estudo realizado por Andrade e Favaretto (2019), que aborda um âmbito complementar importante, de interface e experiência do usuário.

Além disso, a análise de sistemas como o *Workbook*, desenvolvido pelo IFRN, demonstra a necessidade crucial de fortes mecanismos de pesquisa e divulgação para os prestadores de serviços, o que permitiria assegurar que os profissionais qualificados possam ser encontrados de forma fácil e segura pelos utilizadores. A importância da segurança e da confiabilidade dos sistemas baseados na *Web* é também sublinhada por Oliveira (2022), que observa que a proteção da privacidade dos pacientes e a segurança das transações devem ser uma grande preocupação para os criadores.

Nesse cenário, o desenvolvimento de um sistema *web* de assistência residencial apresenta como objetivo não apenas a criação de uma ferramenta tecnológica, mas a solução de problemas práticos enfrentados por consumidores e profissionais. Dentre as funcionalidades previstas estão a possibilidade de avaliação e *feedback* dos prestadores de serviços, busca e oferta de pessoas capacitadas em áreas do ramo doméstico, filtro de prestadores voltada a sua especialidade, perfil estilizado de cada usuário, cadastro e a instalação de mecanismos de segurança para proteção de informações sensíveis. Com isso, o sistema apresentará a expectativa de solucionar não apenas a ineficiência dos serviços prestados na manutenção residencial, mas contribuir para a profissionalização e valorização dos trabalhadores do ramo.

1.2 Objetivos Gerais

Desenvolver um sistema *web* de assistência residencial que conecte clientes a prestadores de serviços de manutenção, como eletricitas e encanadores, otimizando a demanda e a oferta por meio de uma plataforma eficiente, segura e acessível.

1.3 Objetivos Específicos

- Analisar plataformas existentes para identificar funcionalidades essenciais e melhores práticas no setor de assistência residencial.
- Projetar e desenvolver uma interface intuitiva e acessível, garantindo uma boa experiência de usuário para clientes e prestadores de serviço.
- Implementar um sistema de pesquisa e filtragem eficiente para facilitar a localização de profissionais qualificados.
- Criar um sistema de avaliações e *feedbacks*, permitindo que clientes avaliem os serviços prestados e promovam a transparência na plataforma.
- Garantir a segurança dos dados protegendo as informações dos usuários e prevenindo fraudes.
- Facilitar a divulgação e visibilidade dos prestadores de serviço, contribuindo para sua profissionalização e valorização no mercado.

2. Referencial Teórico

2.1. HTML5

Conforme o *site* (Mozilla, 2021), HTML (*Hypertext Markup Language*) é uma linguagem de marcação utilizada para criar a estrutura de conteúdo de uma página *web*. Ela é composta por elementos que representam diferentes tipos de conteúdo, como texto, imagens, *links* e formulários. Esses elementos são organizados em uma hierarquia de árvore, em que os elementos pai contêm elementos filhos. Ao criar uma página HTML, é possível definir a aparência do conteúdo utilizando CSS (*Cascading Style Sheets*) e adicionar interatividade com JavaScript. Além disso, HTML é uma linguagem acessível, permitindo a utilização de recursos para tornar a página mais inclusiva para usuários com deficiências visuais ou auditivas. Em resumo, HTML é a base da *web* e é essencial para qualquer desenvolvedor que deseja criar conteúdo para a *Internet*.

2.2. CSS3

De acordo com o publicado no *site* (Mozilla, 2023), CSS é uma linguagem de folha de estilo usada para descrever a apresentação de um documento escrito em HTML ou XML (*eXtensible Markup Language*). Com o CSS, é possível definir o *layout*, as fontes, as cores e outros aspectos visuais do conteúdo da página. O CSS permite que os desenvolvedores separem a apresentação da estrutura do documento, tornando a manutenção e atualização da aparência visual do site mais fácil e eficiente. Além disso, o CSS permite que diferentes dispositivos e navegadores possam interpretar a página da mesma maneira, proporcionando uma experiência consistente para os usuários finais.

2.3. PHP

PHP (*Hypertext Preprocessor*) é uma linguagem para criação de *scripts* para a *web* do lado servidor, embutidos em HTML, cujo código fonte é aberto, e que é compatível com os mais importantes servidores *web* (especificamente o Apache). O PHP permite incorporar fragmentos de código em páginas HTML normais – código esse que é interpretado à medida que suas páginas são oferecidas aos usuários. (Converse, 2013, p. 27). Além do PHP conter essas vantagens de código aberto e a compatibilidade com servidores *web*, é de fácil instalação em qualquer plataforma, e se integra a quase todos os bancos de dados usados na atualidade. (Dev Fúria, 2015)

2.4. MySQL

Segundo a Oracle, o MySQL é um sistema de gerenciamento de banco de dados relacional de código aberto. Ele é amplamente utilizado para armazenar, gerenciar e recuperar dados para uma ampla variedade de aplicações, desde *sites* simples até sistemas complexos de armazenamento de dados. MySQL é conhecido por sua alta performance, confiabilidade e facilidade de uso. É projetado para suportar uma carga de trabalho intensa e pode ser escalado para grandes volumes de dados e usuários (Oracle Brasil, 2024).

O MySQL utiliza a linguagem SQL (*Structured Query Language*) para realizar operações de banco de dados, como inserção, atualização, exclusão e consulta de dados. Ele é compatível com várias plataformas e sistemas operacionais, incluindo Windows, Linux e macOS, e pode ser integrado com diversas linguagens de programação, como PHP, Java, Python e C++ (Oracle Brasil, 2024).

A Oracle destaca que o MySQL é uma solução ideal para aplicações *web*, devido à sua capacidade de manusear grandes volumes de dados e usuários simultâneos. Ele é utilizado por muitas grandes empresas e serviços *online* populares, como Facebook, Twitter e YouTube, para gerenciar suas operações de banco de dados (Oracle Brasil, 2024).

2.5. Visual Studio Code(VS Code)

O *Visual Studio Code* (VS Code) é um editor de código. Ele é de código aberto, foi desenvolvido pela *Microsoft*, é uma poderosa ferramenta capaz de compilar dados em alta escala, foi feito para ser intuitivo e de fácil manuseio. Possui suporte a diversas linguagens de programação, terminal de comandos integrado e controle de versão. (Hanashiro, 2021).

2.6. JAVASCRIPT

Segundo o Tecnoblog (s.d.), JavaScript é uma linguagem de programação de alto nível voltada para o desenvolvimento *web*, criada originalmente para funcionar ao lado do cliente, ou seja, nos navegadores. Junto ao HTML e ao CSS, é uma das principais tecnologias da *web*, permitindo a criação de páginas interativas com elementos dinâmicos e boa performance.

3. Trabalhos Correlatos

Nesta seção estão resumidos três trabalhos correlatos encontrados em pesquisas na *web* através das palavras-chave: Marido de Aluguel, Plataformas residenciais, Ajuda residencial, Prestadores de serviços domésticos, Reparos domésticos, Assistência residencial. Apesar da busca feita com tais palavras, foi possível encontrar trabalhos que fizeram uso de outras referências.

3.1. Doutor Resolve

A Figura 1 mostra a página *web* da Doutor Resolve, uma plataforma *online* dedicada à prestação de serviços de reparos e reformas residenciais. Criada para facilitar o acesso dos clientes a profissionais especializados, a Doutor Resolve oferece uma interface intuitiva que permite a navegação fácil e rápida pelos diversos serviços disponíveis. A plataforma, lançada em 2010, foi desenvolvida para atender tanto a clientes residenciais quanto comerciais, proporcionando uma solução completa para necessidades de manutenção e melhorias.

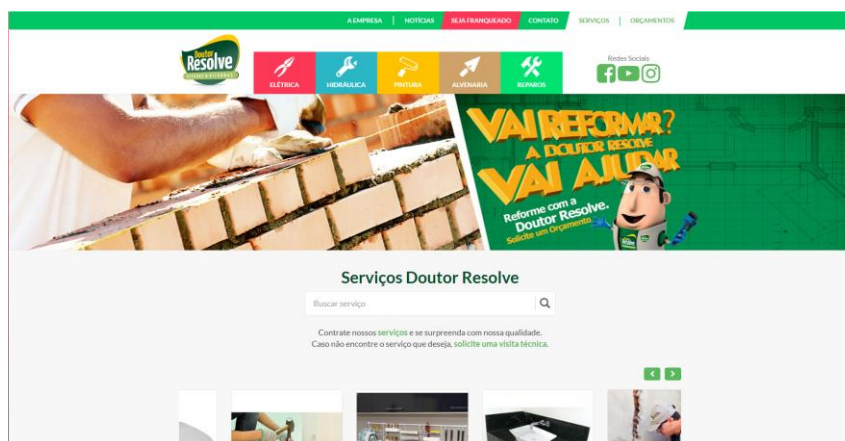


Figura 1. Doutor Resolve

3.2. GetNinjas

A Figura 2 mostra a página *web* do GetNinjas, uma plataforma online que conecta clientes a prestadores de serviços em diversas áreas, como reformas, limpeza, assistência técnica, entre outros. Criada em 2011, a GetNinjas foi desenvolvida para facilitar o acesso a profissionais qualificados, oferecendo uma interface intuitiva que permite aos usuários solicitar orçamentos e contratar serviços com facilidade. Disponível tanto na *web* quanto em aplicativos para Android e iOS, a plataforma visa atender às necessidades de clientes residenciais e comerciais.



Figura 2. GetNinjas

3.3. Triider

A Figura 3 apresenta a página *web* da Triider, uma plataforma digital que conecta clientes a profissionais especializados em reformas, manutenção e pequenos reparos residenciais. Inaugurada em 2016, a Triider foi concebida para simplificar o acesso a serviços confiáveis, proporcionando uma interface intuitiva que permite aos usuários solicitar orçamentos, comparar profissionais e contratar serviços de forma rápida e segura. Disponível tanto na *web* quanto em aplicativos para Android e iOS, a plataforma atende às necessidades de clientes residenciais e pequenos negócios.



Figura 3. Triider

3.4 Comparativo entre trabalhos

Para a melhor compreensão das semelhanças e diferenças entre os trabalhos correlatos e o que foi desenvolvido neste trabalho, a Tabela 1 apresenta um comparativo entre as funcionalidades oferecidas entre tais sistemas.

Tabela 1. Comparativos trabalhos

Aspectos Avaliativos	Doutor Resolve	GetNinjas	Triider	Home&Help
Solicitação em tempo real	Não	Não	Não	Sim
Foco na demanda doméstica	Não	Não	Sim	Sim
Timeline estilizada rede social	Não	Não	Não	Sim
Feedback na timeline	Não	Não	Não	Sim
Página focada aos prestadores	Não	Não	Não	Sim

O Home&Help se destaca dos concorrentes por oferecer solicitações em tempo real, uma interface interativa com *timeline* (linha do tempo) estilizada e *feedback* direto, algo que as outras plataformas não possuem. O foco na demanda doméstica é ofertado pelo sistema Triider e Home&Help, porém ainda não há uma página voltada aos prestadores de serviço, com isso, o sistema Home&Help se destaca pois conta com a facilidade de captação de clientes devido ao seu foco nos prestadores.

4. Metodologias

A proposta consiste em apresentar o desenvolvimento de um aplicativo *web* voltado para a assistência residencial, oferecendo uma interface intuitiva que permite aos usuários dessa aplicação buscar de uma forma simples o acesso a serviços confiáveis, com isso, permitir aos usuários solicitar orçamentos, comparar profissionais e contratar serviços de forma rápida e segura.

4.1 Pesquisa bibliográfica

Como fontes de pesquisa para a conclusão dessa aplicação, serão utilizados artigos, documentos, autores e aplicações já em uso no mercado para ter um referencial e um ponto de partida para esse projeto, incluindo revisão bibliográfica para a apresentação e utilização de ferramentas técnicas para o desenvolvimento do Sistema Home&Help.

4.2 Modelo Incremental

A Figura 4 mostra o desenvolvimento do sistema. Optou-se pelo modelo incremental, devido à sua flexibilidade em incorporar mudanças à medida que o projeto avança. Isso permitirá a entrega de funcionalidades de forma interativa, possibilitando a validação precoce das soluções e a obtenção de *feedback*. O desenvolvimento é dividido em etapas, denominadas “incrementos”, que produzirão incrementalmente o sistema, até a sua versão final. Em cada incremento é realizado todo o ciclo do desenvolvimento do *software*, e planejamento dos testes do sistema já em funcionamento (Dias, 2019).

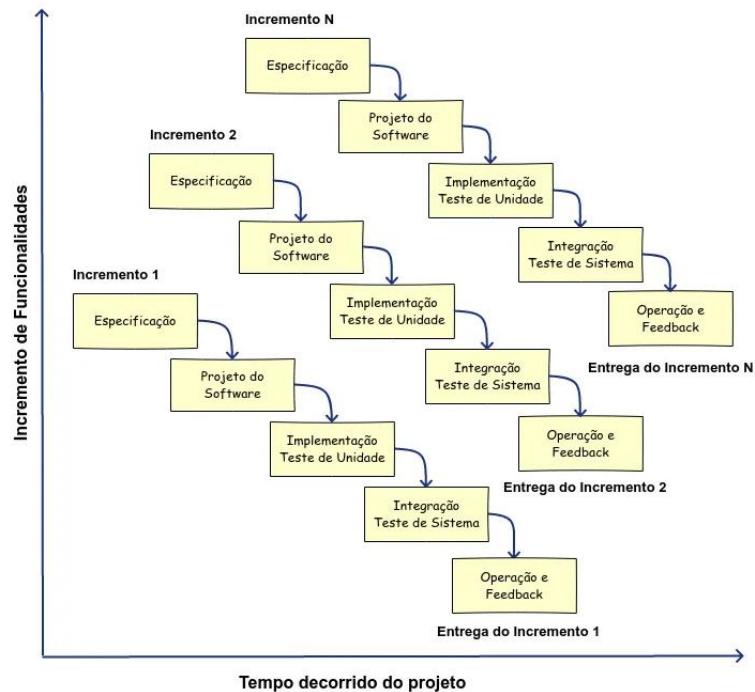


Figura 4. Modelo Incremental

5. Análise

A análise do sistema Home&Help foi baseada na UML, utilizando o Diagrama de Casos de Uso para representar suas funcionalidades. Os requisitos foram divididos em funcionais, que detalham as operações essenciais, e não funcionais, que definem características como usabilidade e compatibilidade. Além disso, um modelo lógico foi aplicado para estruturar os dados de forma clara e independente da tecnologia, garantindo um desenvolvimento eficiente e alinhado às necessidades do projeto.

5.1. Diagrama de Casos de Uso

Para a análise do sistema Home&Help foi inserido o modelo UML (*Unified Modeling Language*), que ajuda na modelagem e documentação de sistemas como está sendo mostrado na figura 5. Entre seus diagramas, o Diagrama de Casos de Uso é essencial para representar as funcionalidades do sistema do ponto de vista do usuário, sem detalhes técnicos (Devmedia, 2012). Abaixo está o diagrama de caso de uso utilizado na implementação do sistema Home&Help:

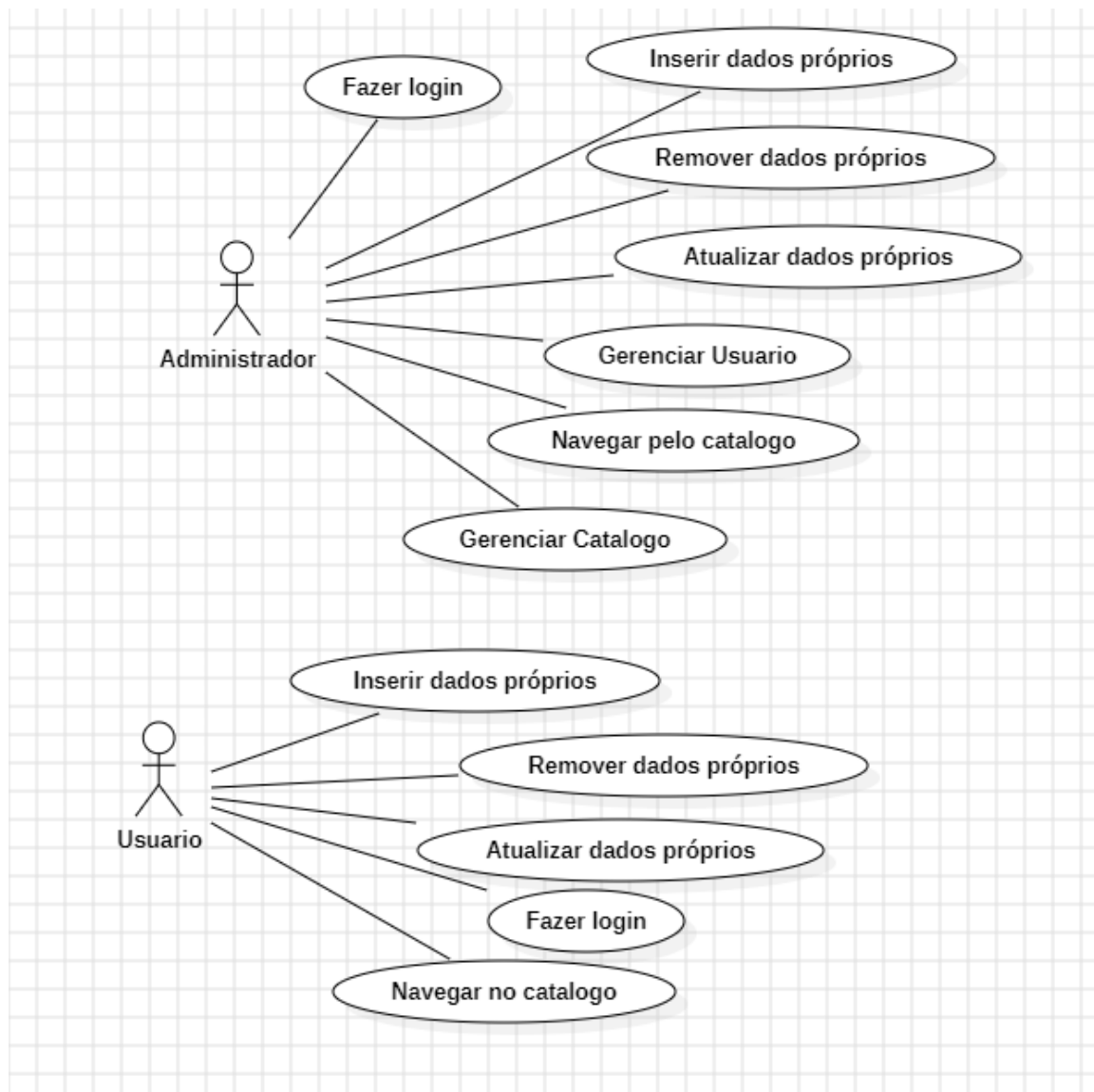


Figura 5. Diagrama de Caso de Uso

5.2. Requisitos

Requisitos de *software* são especificações que definem as funcionalidades e restrições que um *software* deve atender. Eles orientam o processo de desenvolvimento, assegurando que o produto final cumpra as necessidades identificadas na fase de concepção e atenda aos objetivos do negócio (Softdesign, 2025).

5.2.1 Requisitos Funcionais

Segundo Softdesign (2025), os requisitos de *software* funcionais são a base de qualquer projeto de desenvolvimento de *software*. São especificações detalhadas das funcionalidades que o sistema deve possuir para resolver o problema para o qual ele foi projetado. Abaixo foi inserido os Requisitos Funcionais do sistema:

- **Cadastro de Usuários;**
- **Busca por Prestadores de serviço;**

- **Perfil de Prestadores;**
- **Filtro para os Prestadores;**
- **Diálogo direto com o Prestador.**

5.2.2 Requisitos Não Funcionais

Os requisitos de *software* não funcionais definem as características, qualidades e restrições que o sistema deve ter. De forma mais objetiva, ele se refere a como o sistema se comportará e exercerá tais funcionalidades, e não exatamente o que ele fará (Softdesign, 2025). Abaixo foi inserido os Requisitos Não Funcionais do sistema:

- **Usabilidade;**
- **Disponibilidade;**
- **Documentação;**
- **Manutenibilidade;**
- **Privacidade;**
- **Compatibilidade Multiplataforma.**

5.3 Modelo Lógico

Na Figura 6 é mostrado o modelo lógico, que é uma representação abstrata dos requisitos de dados de uma organização, focando em entidades, atributos e relacionamentos, sem detalhar a implementação. Ele captura as necessidades e regras de negócios de forma compreensível, independente da tecnologia, e facilita a organização dos dados no sistema, antes da implementação física, garantindo flexibilidade na escolha da tecnologia (Alura, 2024). Abaixo estão todas as funcionalidades das tabelas criadas para o sistema Home&Help.

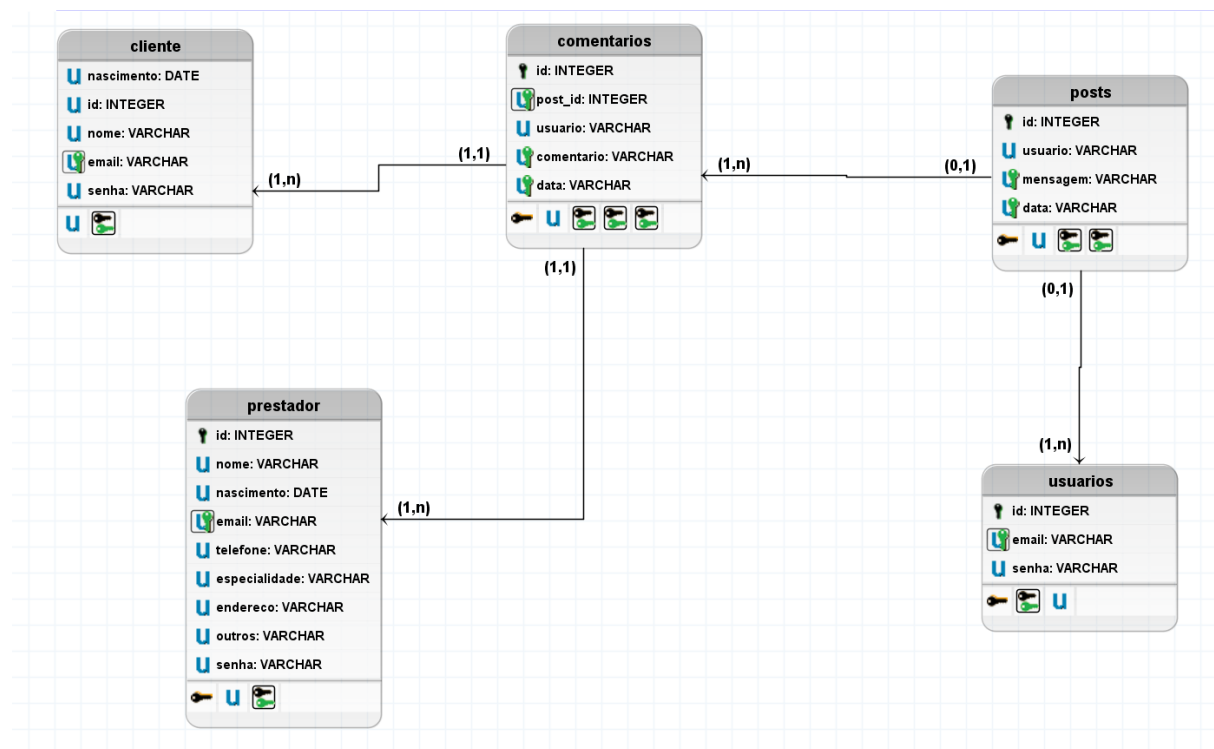


Figura 6. Modelo Lógico

6. Desenvolvimento

Nesta seção, será detalhado o processo de desenvolvimento do projeto, abordando as tecnologias e ferramentas utilizadas para sua implementação. Inicialmente, será apresentada as etapas de desenvolvimento, a implementação das funcionalidades e a estilização da interface. Além disso, serão exibidos trechos de código e imagens ilustrativas que demonstram o funcionamento do sistema, permitindo uma compreensão clara e objetiva da sua construção.

6.1 Tela de Login

Na Figura 7 é apresentada a página de *login* do sistema, onde basicamente, ela permite que um usuário entre com seu *e-mail* e senha para acessar a plataforma. Além disso, tem dois botões, para identificação, entre "Cliente" e "Prestador". Esses botões levam para páginas de cadastro específicas. No final, tem um *link* "Voltar", que leva o usuário para uma página anterior.

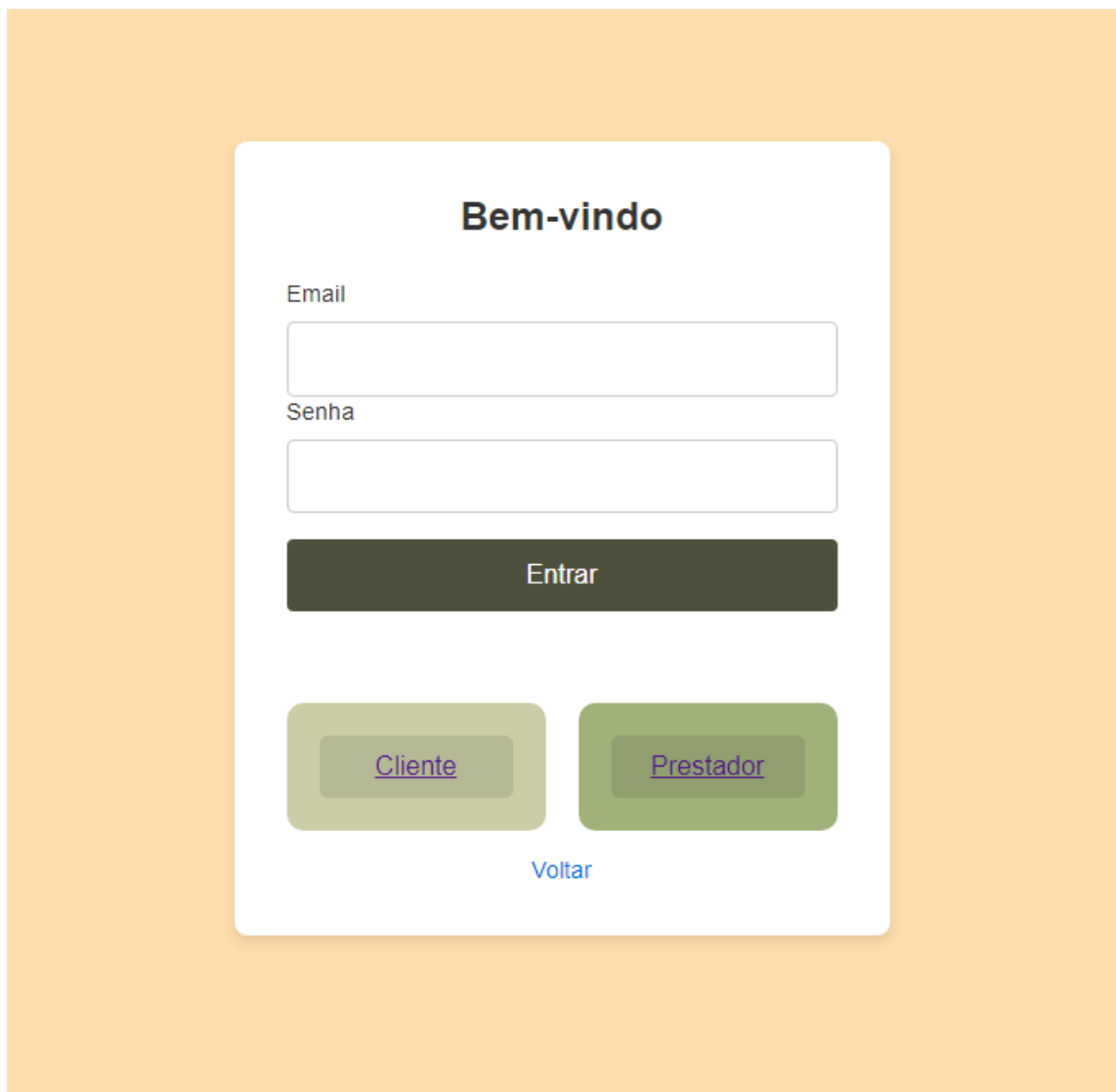
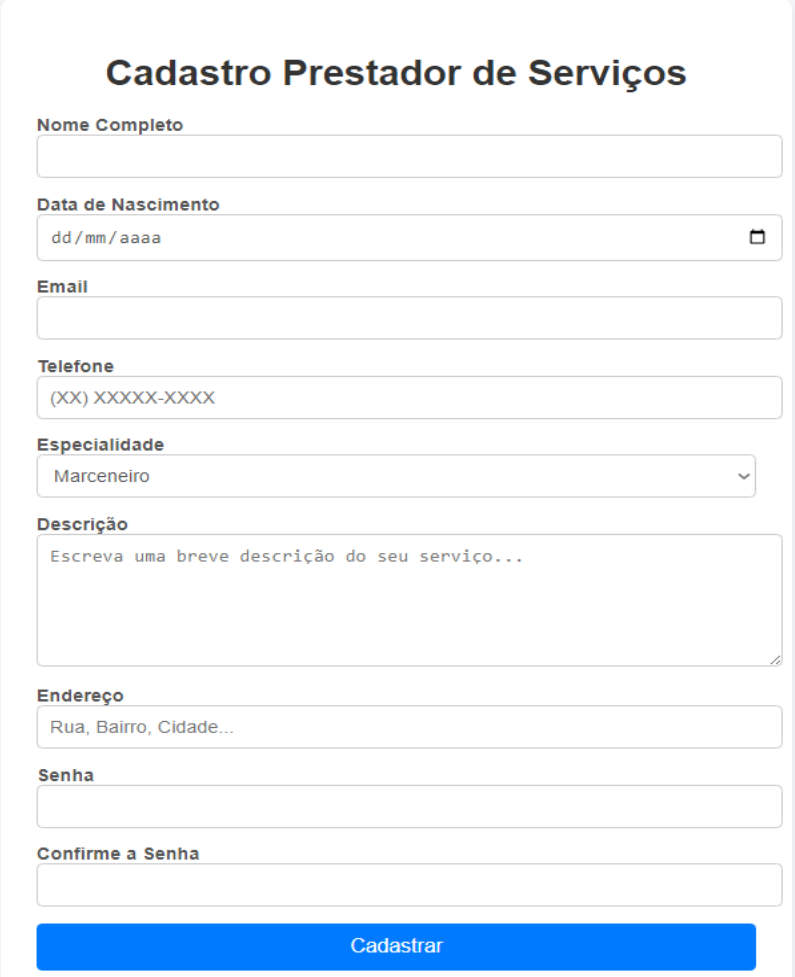
A imagem mostra a interface de login de um sistema. O fundo é uma cor sólida de laranja claro. No centro, há um formulário branco com uma sombra suave. No topo do formulário, o texto "Bem-vindo" está em uma fonte preta, negrito e tamanho médio. Abaixo dele, há dois campos de entrada: o primeiro é rotulado "Email" e o segundo "Senha", ambos em uma fonte cinza pequena. Cada rótulo está à esquerda do seu respectivo campo de entrada, que é um retângulo branco com uma borda cinza muito fina. Abaixo dos campos, há um botão "Entrar" de cor escura, quase preta, com o texto "Entrar" em branco. Mais abaixo, há dois botões retangulares de cor verde-oliva. O primeiro botão contém o texto "Cliente" em uma fonte verde, e o segundo contém "Prestador" em uma fonte verde. Ambos os botões têm uma borda verde mais escura. Na base do formulário, há um link "Voltar" em uma fonte azul, que funciona como um texto clicável sem uma borda ou fundo.

Figura 7. Tela de Login

6.2 Tela de Cadastro Prestadores

Na Figura 8 é apresentada a página de cadastro para prestadores de serviços. Aqui, os profissionais podem registrar suas informações para oferecer seus serviços. O formulário solicita dados básicos como nome completo, data de nascimento, *e-mail* e telefone. Além disso, é possível selecionar uma especialidade (como marceneiro, por exemplo) e escrever uma breve descrição sobre o serviço oferecido. Também é necessário informar o endereço e criar uma senha, que deve ser confirmada para garantir que foi digitada corretamente. Após preencher tudo, basta clicar no botão "Cadastrar" para finalizar o registro. Caso o usuário já tenha uma conta, há um *link* para fazer *login*, além da opção de voltar para a página anterior.



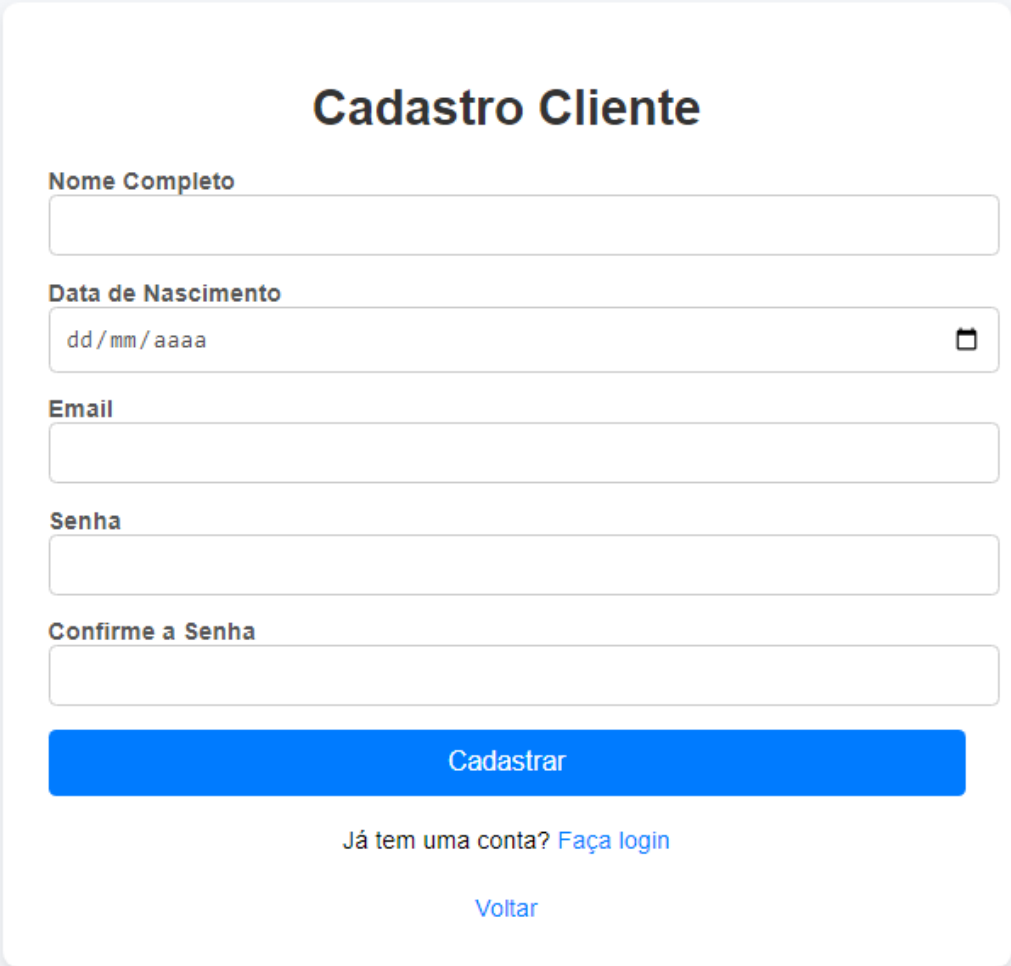
O formulário, intitulado "Cadastro Prestador de Serviços", contém os seguintes campos e elementos:

- Nome Completo:** Campo de texto único.
- Data de Nascimento:** Campo com máscara "dd/mm/aaaa" e ícone de calendário.
- Email:** Campo de texto único.
- Telefone:** Campo com máscara "(XX) XXXXX-XXXX".
- Especialidade:** Menu suspenso com "Marceneiro" selecionado.
- Descrição:** Área de texto grande com o placeholder "Escreva uma breve descrição do seu serviço..." e ícone de lápis.
- Endereço:** Campo com placeholder "Rua, Bairro, Cidade...".
- Senha:** Campo de texto único.
- Confirme a Senha:** Campo de texto único.
- Cadastrar:** Botão azul destacado.
- Links de navegação:** "Já tem uma conta? [Faça login](#)" e "[Voltar](#)".

Figura 8. Tela de Cadastro Prestadores

6.3 Tela de Cadastro de Cliente

Na Figura 9 é apresentada a página de cadastro para clientes. Aqui, os usuários podem criar uma conta para acessar a plataforma. O formulário solicita informações básicas como nome completo, data de nascimento e e-mail. Além disso, é necessário criar uma senha e confirmá-la para garantir que foi digitada corretamente. Depois de preencher os campos, basta clicar no botão "Cadastrar" para concluir o registro. Se o usuário já tiver uma conta, há um link para fazer login. Também existe a opção de voltar para a página anterior.



Cadastro Cliente

Nome Completo

Data de Nascimento

dd/mm/aaaa

Email

Senha

Confirme a Senha

Cadastrar

Já tem uma conta? [Faça login](#)

[Voltar](#)

Figura 9. Tela de Cadastro de Cliente

6.4 Tela Principal (Home)

Na Figura 10 é apresentada a página inicial do sistema Home&Help - Serviços Residenciais, onde os usuários podem interagir publicando mensagens e comentando em postagens. O sistema dá as boas-vindas ao usuário logado, exibindo seu nome no topo. Há um campo onde o usuário pode escrever e publicar mensagens. Abaixo, aparecem postagens com a opção de comentar onde os usuários podem interagir entre si. No topo da página, há um menu de navegação com *links* para Home, Serviços, Perfil, além das opções de Entrar e Sair. No rodapé, a plataforma exibe seus direitos autorais.



Figura 10. Tela Principal (Home)

6.5 Tela de Serviços

Na Figura 11 é apresentada a página de listagem dos Prestadores de Serviços na plataforma Home Help - Serviços Residenciais. Aqui, os usuários podem visualizar profissionais disponíveis para diferentes tipos de serviços. No topo, há um filtro que permite selecionar uma especialidade específica ou visualizar todas. Abaixo, são exibidos cartões com informações dos prestadores, incluindo nome, especialidade, e-mail, telefone, endereço e uma breve descrição dos serviços oferecidos. Cada profissional tem um cartão individual, facilitando a visualização e a busca pelo serviço desejado. No menu superior, há opções para navegar entre as páginas Home, Serviços e Perfil, além dos botões de Entrar e Sair.

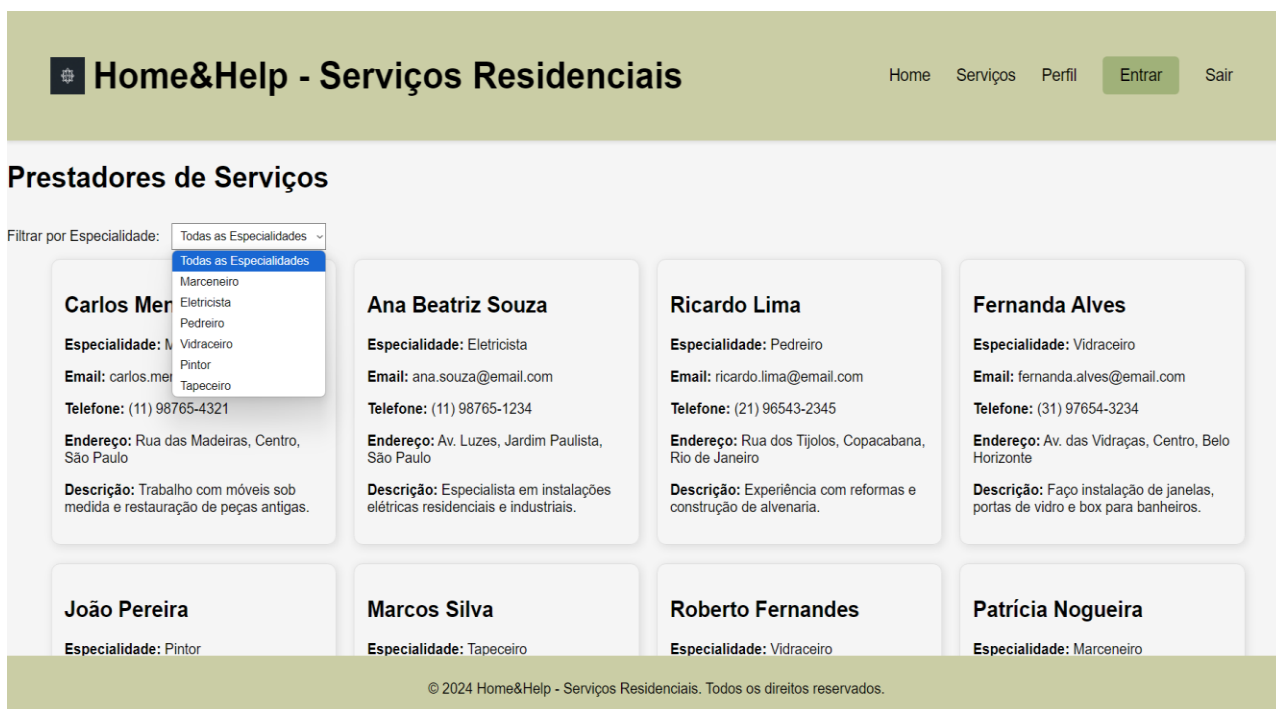


Figura 11. Tela de Serviços

6.6 Tela de Perfil

Na Figura 12 é apresentada a página que exibe o perfil do prestador de serviços/cliente na plataforma Home&Help - Serviços Residenciais. Aqui, o profissional pode visualizar seus dados cadastrados de forma clara e organizada. Além disso, há um botão "Editar" e "Salvar", permitindo que o profissional atualize suas informações sempre que necessário. A navegação da plataforma permanece acessível no topo da página, com atalhos para Home, Serviços, Perfil, além das opções de Entrar ou Sair, garantindo praticidade e uma experiência intuitiva para o usuário.

A imagem mostra a interface de perfil de usuário no sistema Home&Help - Serviços Residenciais. No topo, há uma barra de navegação com links para Home, Serviços, Perfil, Entrar e Sair. O conteúdo principal é o 'Perfil de Prestador', que contém um formulário com os seguintes campos: Nome (Carlos Mendes), Data de Nascimento (12/06/1985), Email (carlos.mendes@email.com), Especialidade (Marceneiro) e Endereço (Rua das Madeiras, Centro, São Paulo). Abaixo do formulário, há um botão 'Editar'. No rodapé, há uma linha de copyright: '© 2024 Home&Help - Serviços Residenciais. Todos os direitos reservados.'

Figura 12. Tela de Perfil

6.7 Código - Conexão tela de Cliente

Na Figura 13, é apresentado um código PHP que realiza o cadastro de um cliente em um banco de dados MySQL. Esse script recebe os dados enviados por um formulário, valida as informações e insere os registros na tabela correspondente, garantindo a persistência das informações no sistema.

```
TCC > TCC > conexaoBD_cliente.php
1  <?php
2  include ("conexao.php"); // Conectar ao banco de dados
3
4
5  // Capturar dados do formulário
6  $nome = $_POST['name'];
7  $nascimento = $_POST['date'];
8  $email = $_POST['email'];
9  $senha = $_POST['password'];
10 $confirmSenha = $_POST["confirm-password"];
11
12 $mensagem_status = "";
13
14 // Verificar se as senhas são iguais
15 if ($senha != $confirmSenha) {
16     die("Erro: As senhas não coincidem.");
17 }
18
19 // Inserir no banco de dados
20 $sql = "INSERT INTO cliente (nome, nascimento, email, senha) VALUES ('$nome', '$nascimento', '$email', '$senha')";
21
22
23 if (mysqli_query($conexao, $sql)) {
24     echo ("Dados salvos com sucesso!");
25     echo ("<a href='login.php'> Voltar para o Login</a>");
26 } else {
27     echo ("Erro ao salvar os dados: " . $conexao->error);
28     echo ("<a href='login.php'>Voltar para o Login</a>");
29 }
30
31
32 ?>
33
```

Figura 13. Código - Conexão tela de cliente

6.8 Código - Conexão com o Banco de Dados

Na Figura 14, é exibido o código do arquivo conexão, responsável por estabelecer a conexão com um banco de dados MySQL. Esse *script* é essencial para criar a comunicação entre a aplicação e o banco de dados, permitindo a execução de consultas, inserções e outras operações para o funcionamento do sistema.

```
TCC > TCC > 🐞 conexao.php
1  <?php
2  $host = "127.0.0.1"; // Servidor do banco de dados
3  $usuario = "root";   // Usuário do MySQL (padrão: root)
4  $senha = "";        // Senha do MySQL (padrão: vazio no XAMPP)
5  $banco = "homehelp_bd"; // Nome do banco de dados
6
7  // Criar a conexão
8  $conexao = mysqli_connect($host, $usuario, $senha, $banco);
9
10 // Verificar conexão
11 if (!$conexao) {
12     die("Erro na conexão com o banco de dados: " . $mysqli_connect_errno());
13 }
14 ?>
```

Figura 14. Código – Conexão com Banco de Dados

6.9 Código - Formulário HTML

Na Figura 15, é apresentado um código HTML que representa um formulário de cadastro de clientes. Esse formulário inclui elementos essenciais, como campos de entrada para nome, e-mail e data de nascimento, caixas de seleção para opções específicas, botões para envio dos dados, além de rótulos que auxiliam na identificação de cada campo, garantindo uma experiência mais intuitiva para o usuário.

```

<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Cadastro</title>
  <link rel="stylesheet" href="styles_cadastro.css">
</head>
<body>

  <div class="register-container">
    <h1>Cadastro Cliente</h1>

    <form class="register-form" action="conexaoBD_cliente.php" method="POST">
      <div class="input-group">
        <label for="name">Nome Completo</label>
        <input type="text" id="name" name="name" required>
      </div>

      <div class="input-group">
        <label for="date">Data de Nascimento</label>
        <input type="date" id="date" name="date" required>
      </div>

      <div class="input-group">
        <label for="email">Email</label>
        <input type="email" id="email" name="email" required>
      </div>

      <div class="input-group">
        <label for="password">Senha</label>
        <input type="password" id="password" name="password" required>
      </div>

      <div class="input-group">
        <label for="confirm-password">Confirme a Senha</label>
        <input type="password" id="confirm-password" name="confirm-password" required>
      </div>

      <button type="submit">Cadastrar</button>
      <p class="login-link">Já tem uma conta? <a href="./login.php">Faça login</a></p>
      <a href="./index.php" class="forgot-password">Voltar</a>
    </form>
  </div>
</body>
</html>

```

Figura 15. Código – Formulário HTML

6.10 Código - Edição no CSS

Na Figura 16, é exibida a tela que ilustra a estilização do formulário do cliente utilizando a linguagem CSS. A imagem apresenta algumas estilizações aplicadas, as quais estão vinculadas à página principal exibida ao usuário. Nela, são definidas características como cores, tamanhos e formatos, determinando a aparência final do ambiente.

```

TCC > TCC > # styles_cadastro.css > body
1  body {
2      font-family: Arial, sans-serif;
3      background-color: #f4f6f9;
4      margin: 0;
5      padding: 0;
6  }
7
8  .register-container {
9      width: 100%;
10     max-width: 600px;
11     margin: 50px auto;
12     background-color: #fff;
13     padding: 30px;
14     border-radius: 10px;
15     box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
16 }
17
18 h1 {
19     text-align: center;
20     font-size: 2rem;
21     margin-bottom: 20px;
22     color: #333;
23 }
24
25 .register-form {
26     display: flex;
27     flex-direction: column;
28 }
29
30 .input-group {
31     margin-bottom: 15px;
32 }
33
34 .input-group label {
35     font-size: 1rem;
36     font-weight: bold;
37     margin-bottom: 5px;
38     color: #555;
39 }
40
41 .input-group input, .input-group select, .input-group textarea {
42     width: 100%;
43     padding: 10px;
44     border: 1px solid #ccc;
45     border-radius: 5px;
46     font-size: 1rem;
47     color: #555;
48 }

```

Figura 16. Código estilização no CSS

6.11 Teste de Funcionalidade

É um tipo de teste de *software* que verifica se um sistema ou aplicação está funcionando conforme esperado, de acordo com os requisitos estabelecidos. Ele avalia se cada funcionalidade implementada atende corretamente às especificações definidas, garantindo que o *software* cumpra seus objetivos (CIRIUS QUALITY, 2023).

Principais Características do Teste de Funcionalidade:

- Foca no espaço que o sistema deve fazer, e não em como ele faz.
- Pode ser feito de forma manual ou automatizada.

- Baseia-se nos requisitos e especificações do software.
- Testa entradas, processamentos e saídas esperadas.

Exemplo de teste abaixo. Nas Figuras 17 e 18, é apresentado o teste da funcionalidade do botão de cadastro do cliente. Nesse caso, ao clicar no botão “Cliente”, o usuário deve ser transferido para a tela de cadastro do cliente, caso isso não ocorresse, deveria ser voltado um passo anterior e com isso, trabalhar para arrumar essa funcionalidade. Durante o projeto, os testes funcionaram dessa forma, testando todas as funcionalidades e checando o seu retorno positivo ou negativo.

A tela de login, intitulada "Bem-vindo", possui campos para "Email" e "Senha". Abaixo dos campos, há um botão "Entrar" em um fundo escuro. Logo abaixo do botão "Entrar", há uma seta para baixo. Na base da tela, há dois botões: "Cliente" (destacado com um fundo mais claro) e "Prestador". Abaixo desses botões, há um link "Voltar" em azul.

Figura 17. Tela Login “Teste”

A tela de cadastro do cliente, intitulada "Cadastro Cliente", possui campos para "Nome Completo", "Data de Nascimento" (com máscara dd/mm/aaaa), "Email", "Senha" e "Confirme a Senha". Abaixo dos campos, há um botão "Cadastrar" em azul. Na base da tela, há um link "Já tem uma conta? Faça login" e um link "Voltar" em azul.

Figura 18. Tela Cadastro Cliente “Teste”

7. Conclusão

Com a crescente demanda por soluções ágeis e eficientes, as plataformas digitais que conectam clientes e prestadores de serviços têm se mostrado essenciais para otimizar processos. Este projeto visou desenvolver um sistema web de assistência residencial, com o objetivo de atender a essa demanda.

Durante o desenvolvimento deste projeto, foram utilizadas diversas ferramentas e tecnologias essenciais para a área de Análise e Desenvolvimento de Sistemas, incluindo HTML, CSS, PHP, JAVASCRIPT, Linguagem SQL e Modelagem do sistema. A aplicação dessas ferramentas possibilitou a implementação das soluções propostas e a resolução dos desafios encontrados ao longo do processo.

Dentre as principais dificuldades enfrentadas, destacam-se a parte escrita do projeto, a falta de documentação na *web* e em alguns momentos a falta de conhecimento em alguns determinados assuntos em relação a escrita do código para a formação do sistema. No entanto, por meio da aplicação dos conhecimentos adquiridos ao longo do curso e da pesquisa contínua, foi possível superar esses obstáculos e atingir os objetivos estabelecidos no início do projeto.

Com isso, após a conclusão do projeto, foi apresentado que os conhecimentos adquiridos com o tempo foram essenciais para a finalização do projeto, proporcionando um aprendizado significativo e alinhado às competências desenvolvidas no curso de Análise e Desenvolvimento de Sistemas. Dentre essas competências, destacam-se a capacidade de análise e modelagem de sistemas, programação, gerenciamento de banco de dados e resolução de problemas técnicos.

Por fim, este trabalho abre caminho para futuras melhorias e novas implementações. Projetos futuros deverão envolver a expansão das funcionalidades, otimização do desempenho e melhoria do código, garantindo uma evolução contínua e aprimoramento da solução desenvolvida.

8. Trabalhos Futuros

Para as melhorias futuras deste projeto, estávamos pensando em otimizar o código, refatorando e aplicando melhores práticas para melhorar o desempenho. Também será importante criar mais funcionalidades, como integração com notificações, *chat* entre clientes e prestadores, e geolocalização. Além disso, o projeto deverá ser colocado na *web*, garantindo a escalabilidade, segurança e bom desempenho em um ambiente de produção. A versão *mobile* poderá ser desenvolvida com um *layout* responsivo ou um aplicativo nativo para iOS e Android, visando ampliar a acessibilidade e melhorar a experiência do usuário. Essas ações garantirão a evolução contínua do sistema.

9. Referências

ALURA. Modelo lógico. *Alura*. Disponível em: <https://www.alura.com.br/artigos/diferencas-modelo-logico-modelo-fisico?srsId=AfmBOopdlMiJ0FIQpexcG8548PEi6WRzhDtYOdWA1zJfJAu4g9FVCg-s>.

CIRIUS QUALITY. Teste de funcionalidade. *Cirius Quality*. Disponível em: https://ciriusquality.com.br/glossario/o-que-e-teste-de-funcionalidade/?srsId=AfmBOopBibXnhNosPTS9Ggr8Ux8NMU8B0V_Ppt5F1r1nd2r-F0-8u1Wq.

DEV FÚRIA. O que é o PHP. *DevFuria*. Disponível em: <http://devfuria.com.br/php/o-que-e-php/>.

DEV MEDIA. Diagrama de caso de uso. *DevMedia*. Disponível em: <https://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>.

DIAS, R. O modelo incremental. Contexto Delimitado, 2019. Disponível em: <https://medium.com/contexto-delimitado/o-modelo-incremental-b41fc06cac04>.

DOUTOR RESOLVE. *Doutor Resolve*. Disponível em: <https://www.doutorresolve.com.br/>.

EXAME. Marido de aluguel cresce em SP como opção ao desemprego e salários baixos. Disponível em: <https://exame.com/pme/marido-de-aluguel-cresce-em-sp-como-opcao-a-desemprego-e-salarios-baixos/>.

GETNINJAS. *GetNinjas*. Disponível em: <https://www.getninjas.com.br/>.

INSTITUTO FEDERAL DO RIO GRANDE DO NORTE. Sistema Workbook para busca e divulgação de prestadores de serviços. 2017. Disponível em: https://memoria.ifrn.edu.br/bitstream/handle/1044/2540/SISTEMA_WORKBOOK_PARA_BUSCA_E_DIVULGA_O_DE_PRESTADORES_DE_SERVIÇOS%20%281%29.pdf?sequence=1&isAllowed=y.

MELLO, F.; SILVA, R. Desenvolvimento de sistemas web para serviços residenciais. *Revista Thema*, v. 15, n. 2, p. 45-60, 2018. Disponível em: <https://periodicos.ifsul.edu.br/index.php/thema/article/view/1050/1060>.

MOURA, J. Plataforma web para contratação de serviços. *Universidade de Uberaba*, 2019. Disponível em: <https://dspace.uniube.br:8443/bitstream/123456789/2676/1/Plataforma%20web%20para%20contrata%c3%a7%c3%a3o%20de%20servi%c3%a7os.pdf>.

MOZILLA. Getting started with HTML. *Mozilla Developer Network*, 2021. Disponível em: https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics.

MOZILLA. What is CSS? *Mozilla Developer Network*, 2023. Disponível em: https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/What_is_CSS.

ORACLE BRASIL. O que é o MySQL. *Oracle Brasil*. Disponível em: <https://www.oracle.com/br/mysql/what-is-mysql/>.

SEBRAE. Reformas e reparos: marido de aluguel é a solução. *Sebrae*. Disponível em: <https://sebrae.com.br/sites/PortalSebrae/artigos/reformas-e-reparos-marido-de-aluguel-e-a-solucao,d2cf103bc7d1b610VgnVCM1000004c00210aRCRD>.

SOFTDESIGN. O que são requisitos? *SoftDesign*. Disponível em: <https://softdesign.com.br/blog/requisitos-de-software-funcionais-e-nao-funcionais/#h-requisitos-funcionais>.

SOUZA, A. Segurança em sistemas web para serviços residenciais. *Revista Educ@mais*, v. 12, n. 3, p. 35-48, 2017. Disponível em: <https://periodicos.ifsul.edu.br/index.php/educarmais/article/view/1145/928>.

TECNOBLOG. O que é JavaScript? Guia para iniciantes. *Tecnoblog*. Disponível em: <https://tecnoblog.net/responde/o-que-e-javascript-guia-para-iniciantes/>.

TREINA WEB. VS Code: o que é e por que você deve usar. *TreinaWeb*. Disponível em: <https://www.treinaweb.com.br/blog/vs-code-o-que-e-e-por-que-voce-deve-usar>.

TRIIDER. *Triider*. Disponível em: <https://www.triider.com.br/>.

Documento Digitalizado Restrito

Artigo do TCC do aluno Murilo Henrique de Oliveira Moreira

Assunto: Artigo do TCC do aluno Murilo Henrique de Oliveira Moreira
Assinado por: Daiane Tomazeti
Tipo do Documento: Comprovante
Situação: Finalizado
Nível de Acesso: Restrito
Hipótese Legal: Informação Pessoal - dados pessoais e dados pessoais sensíveis (Art. 31 da Lei nº 12.527/2011)
Tipo do Conferência: Cópia Simples

Documento assinado eletronicamente por:

- **Daiane Mastrangelo Tomazeti, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 12/03/2025 14:24:32.

Este documento foi armazenado no SUAP em 12/03/2025. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifsp.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1963789
Código de Autenticação: b526373494

