

**GRUPO DE ESTUDO DE EDUCAÇÃO A DISTÂNCIA / CETEC  
TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS - MODALIDADE EaD**

**Adriana da Conceição  
Beatriz Carvalho da Costa  
Diego Franco Raffaele  
Jurema Dionysio Conde  
Otávio Henrique Prestes Nogueira Moraes  
Paula Cristina de Melo Viana**

**SISTEMA DE GERENCIAMENTO DE TAREFAS**

**São Paulo  
2024**

**Adriana da Conceição  
Beatriz Carvalho da Costa  
Diego Franco Raffaele  
Jurema Dionysio Conde  
Otávio Henrique Prestes Nogueira Moraes  
Paula Cristina de Melo Viana**

## **SISTEMA DE GERENCIAMENTO DE TAREFAS**

Trabalho de Conclusão de Curso apresentado ao Curso Técnico em Desenvolvimento de Sistemas – modalidade EaD, orientado pelo Prof. Cintia Batista Pinto da Silva, como requisito para obtenção do título de Técnico em Desenvolvimento de Sistemas.

**São Paulo  
2024**

## RESUMO

O Sistema de Gerenciamento de Tarefas foi projetado para melhorar a organização e execução das atividades diárias, oferecendo funcionalidades como cadastro de usuários, criação, edição, exclusão e conclusão de tarefas. No front-end, o sistema conta com uma interface moderna, amigável e responsiva, desenvolvida com HTML, CSS, proporcionando uma experiência de usuário otimizada em diversos dispositivos. No back-end, o sistema utiliza PHP e MySQL, garantindo processamento eficiente e armazenamento seguro dos dados, além da utilização da linguagem JavaScript. Notificações automáticas e histórico de atividades auxiliam na gestão do tempo e das tarefas, enquanto a Metodologia Ágil Scrum foi adotada para agilizar o desenvolvimento. Essa solução atende à crescente necessidade de ferramentas que aumentam a produtividade e ajudam na organização eficaz do tempo.

**Palavras-chave:** Gerenciamento de tarefas. Produtividade. Metodologia Ágil. Segurança. Acessibilidade.

## **ABSTRACT**

Task Management System is designed to enhance daily task organization and execution. The system offers functionalities such as user registration, task creation, editing, deletion, and completion. The front-end features a modern, user-friendly, and responsive interface developed with HTML, CSS, providing an optimized user experience across various devices. On the back-end, the system utilizes PHP and MySQL, ensuring efficient processing and secure data storage, along with the implementation of JavaScript language. Automated notifications and activity history assist in time and task management, while the Agile Scrum Methodology was adopted to streamline development. This solution addresses the growing demand for tools that boost productivity and facilitate effective time organization.

**Keywords:** Task Management. Productivity. Agile Methodology. Security. Accessibility.

## ÍNDICE DE ILUSTRAÇÕES

|                                                                        |    |
|------------------------------------------------------------------------|----|
| Figure 1 - Banco de dados do sistema de gerenciamento de tarefas. .... | 23 |
| Figure 2 - Estrutura do sistema de gerenciamento de tarefas. ....      | 25 |
| Figure 3 - Tela registrar. ....                                        | 52 |
| Figure 4 - Tela de recuperação de senha. ....                          | 53 |
| Figure 5 - Tela de login. ....                                         | 53 |
| Figure 6 - Tela Dashboard. ....                                        | 54 |
| Figure 7 - Tela Dashboard após personalização. ....                    | 55 |
| Figure 8 - Tela adicionar tarefas. ....                                | 56 |
| Figure 9 - Tela dashboard após adição da tarefa. ....                  | 56 |
| Figure 10 - Tela editar. ....                                          | 57 |
| Figure 11 - Tela Dashboard após edição. ....                           | 57 |
| Figure 12 - Tela excluir. ....                                         | 58 |

## SUMÁRIO

|           |                                                                  |           |
|-----------|------------------------------------------------------------------|-----------|
| <b>1.</b> | <b>INTRODUÇÃO .....</b>                                          | <b>7</b>  |
| 1.1.      | Justificativa .....                                              | 8         |
| 1.2.      | Objetivo .....                                                   | 8         |
| 1.3.      | Descrição Geral .....                                            | 9         |
| 1.4.      | Descrição da Equipe .....                                        | 9         |
| 1.5.      | Partes Interessadas.....                                         | 9         |
| 1.6.      | Premissas .....                                                  | 10        |
| 1.7.      | Restrição .....                                                  | 10        |
| <b>2.</b> | <b>METODOLOGIA .....</b>                                         | <b>11</b> |
| 2.1.      | Atendimento aos Requisitos Funcionais e Não Funcionais .....     | 11        |
| 2.2.      | Descrição do Software Utilizado .....                            | 12        |
| 2.3.      | Descrição da Equipe e Partes Interessadas .....                  | 12        |
| 2.4.      | Equipe.....                                                      | 12        |
| 2.5.      | Partes Interessadas.....                                         | 13        |
| 2.6.      | Recursos Utilizados .....                                        | 13        |
| 2.7.      | Equipamentos.....                                                | 14        |
| 2.8.      | Técnicas e Processos.....                                        | 14        |
| 2.9.      | Ferramentas de Gerenciamento do Projeto .....                    | 14        |
| 2.10.     | Segurança e Privacidade de Dados.....                            | 15        |
| 2.11.     | Plano de Testes.....                                             | 15        |
| <b>3.</b> | <b>DESENVOLVIMENTO .....</b>                                     | <b>16</b> |
| 3.1.      | Casos de Uso (Anexo I).....                                      | 16        |
| 3.2.      | Desenvolvimento das Estruturas do banco de Dados em MySQL .....  | 19        |
| 3.3.      | Implementação do Sistema .....                                   | 25        |
| 3.3.1.    | Conexão com o Banco de Dados (config.php) .....                  | 25        |
| 3.3.2.    | Página de Login (index.php).....                                 | 26        |
| 3.3.3.    | Página de Registro (register.php) .....                          | 28        |
| 3.3.4.    | Página Recuperação de Senha (forgot_password.php) .....          | 30        |
| 3.3.5.    | Painel de Tarefas Com Filtros e Relatórios (dashboard.php).....  | 31        |
| 3.3.6.    | Adicionar Tarefas com Registro no Histórico (add_task.php) ..... | 37        |
| 3.3.7.    | Editar Tarefas (edit_task.php) .....                             | 40        |
| 3.3.8.    | Excluir Tarefas (delete_task.php) .....                          | 44        |
| 3.3.9.    | Histórico de Atividades (histórico.php) .....                    | 47        |
| 3.3.10.   | Encerrar Sessão (logout.php) .....                               | 48        |
| 3.3.11.   | Estilo CSS Customizado (style.css) .....                         | 48        |
| <b>4.</b> | <b>ARQUITETURA DO SOFTWARE .....</b>                             | <b>52</b> |
| 4.1.      | Registrar.....                                                   | 52        |
| 4.2.      | Recuperar Senha .....                                            | 52        |
| 4.3.      | Login .....                                                      | 53        |
| 4.4.      | Dashboard .....                                                  | 54        |
| 4.5.      | Adicionar Tarefas .....                                          | 55        |
| 4.6.      | Editar Tarefa .....                                              | 56        |
| 4.7.      | Excluir .....                                                    | 58        |
| <b>5.</b> | <b>CONSIDERAÇÕES FINAIS .....</b>                                | <b>59</b> |
|           | <b>REFERÊNCIAS.....</b>                                          | <b>61</b> |
|           | <b>ANEXOS .....</b>                                              | <b>63</b> |
|           | <b>ANEXO I – DIAGRAMA DE USO .....</b>                           | <b>63</b> |
|           | <b>ANEXO II – DIAGRAMA DE CLASSE .....</b>                       | <b>64</b> |
|           | <b>ANEXO III – TRELLO .....</b>                                  | <b>65</b> |

## 1. INTRODUÇÃO

A vida moderna exige cada vez mais de nós. Para acompanhar esse ritmo acelerado, é fundamental encontrar formas de otimizar nosso tempo e aumentar nossa produtividade. Ferramentas como a de gerenciamento de tarefas são aliadas poderosas nessa jornada, oferecendo recursos como listas de tarefas, lembretes, integração com calendários e visão geral sobre o status do projeto (seja para um único usuário, ou para times com vários membros). Com uma abordagem organizada e um planejamento cuidadoso, podemos não apenas alcançar melhores resultados, mas também ganhar tempo para nos dedicarmos ao que realmente importa.

O Sistema de Gerenciamento de Tarefas tem como objetivo fornecer uma solução robusta e eficiente para a organização das atividades diárias dos usuários. Através de uma interface intuitiva e fácil de usar, acessível em diversos dispositivos, o sistema integrará funcionalidades como a criação e manutenção de listas de tarefas, lembretes automáticos e sincronização com calendários, além de oferecer funcionalidades que visam aumentar a produtividade, otimizar o tempo e auxiliar no alcance de seus objetivos.

Dada a crescente demanda por ferramentas que facilitam a gestão pessoal e profissional, desenvolver um sistema de gerenciamento de tarefas torna-se essencial para oferecer uma solução eficiente para a organização e monitoramento das atividades diárias. Este sistema não só simplificará a gestão de tarefas, mas também melhorará a eficiência e o controle sobre o uso do tempo, atendendo a uma necessidade crescente de ferramentas que aumentam a produtividade e ajudam na priorização das tarefas.

O sistema em desenvolvimento oferecerá um conjunto de funcionalidades fundamentais para o gerenciamento eficiente de tarefas. Entre elas, destacam-se o cadastro e a autenticação segura de usuários, a criação, edição, organização, a marcação intuitiva de tarefas como concluídas e um sistema de notificações personalizadas para garantir o cumprimento dos prazos. A experiência do usuário será otimizada por meio de uma interface intuitiva e responsiva, acessível em diversos dispositivos, proporcionando uma gestão de tarefas eficaz e prática.

O Projeto será conduzido utilizando a Metodologia Ágil SCRUM, com uma

equipe dedicada composta por seis membros. Entre eles, um Scrum Master que será responsável por facilitar as reuniões do Scrum e remover obstáculos, enquanto o Product Owner coordenará a equipe e garantirá o cumprimento dos prazos. A equipe de desenvolvimento será formada por um Desenvolvedor de Front-end, que criará a interface do usuário, um Desenvolvedor de Back-end, que gerenciará a lógica de negócios e o banco de dados, uma Testadora de Qualidade, que garantirá o correto funcionamento do sistema, e por último um Designer de Interface, que criará layouts atraentes e intuitivos.

Como resultado, esperamos entregar um sistema que não só simplifique a gestão de tarefas, mas também ofereça ao usuário uma interface intuitiva e acessível, otimizando a sua experiência e contribuindo para uma maior organização e produtividade pessoal. No entanto, o desenvolvimento do sistema pressupõe a disponibilidade de acesso à internet por parte dos usuários, a compatibilidade com navegadores modernos e a utilização de dispositivos móveis e desktops. Além disso, a manutenção regular do sistema será realizada para corrigir bugs, implementar melhorias e assegurar a segurança. Também será considerado o feedback dos usuários para orientar a melhoria contínua do sistema e garantir que ele atenda às suas necessidades e expectativas.

### **1.1. Justificativa**

Com a crescente necessidade de organização e produtividade, um sistema de gerenciamento de tarefas eficaz pode melhorar significativamente a vida dos usuários, ajudando-os a priorizar e concluir suas atividades diárias, além possibilitar que a visão sobre a gestão do projeto como um todo esteja centralizada.

### **1.2. Objetivo**

O objetivo é criar um sistema de cadastro de usuários com autenticação segura, garantindo a proteção dos dados. Além disso, será desenvolvida uma interface amigável que permitirá aos usuários adicionar, editar, excluir e marcar tarefas como concluídas ou em andamento, facilitando a gestão das atividades. Um sistema de notificações será implementado para lembrar os usuários de suas tarefas de forma eficiente. Também será priorizada a acessibilidade do aplicativo em

diferentes dispositivos e navegadores, proporcionando uma experiência de usuário consistente e de alta qualidade.

### **1.3. Descrição Geral**

O sistema em desenvolvimento engloba basicamente as seguintes funcionalidades:

- Cadastro de usuários;
- Autenticação na plataforma;
- Adição, edição e exclusão;
- Marcação de tarefas com status de “Concluídas”;

### **1.4. Descrição da Equipe**

O sistema será baseado na metodologia SCRUM, contendo 6 (seis) participantes na Equipe (1 Scrum Master, 1 Product Owner, 4 membros “Teams”), com a sua divisão de funções já efetuada.

### **1.5. Partes Interessadas**

Para viabilização do projeto, teremos o envolvimento de diferentes públicos, sendo estes:

- Clientes do Projeto;
- Equipe de Desenvolvimento;
- Gestores Funcionais;
- Patrocinadores;
- Fornecedores;
- Usuários Finais;

## **1.6. Premissas**

O sistema será desenvolvido com o propósito de proporcionar uma solução completa e eficaz para o gerenciamento de tarefas, com premissas assumidas como verdadeiras para facilitar o planejamento e a execução de tarefas, visando não só atender às necessidades dos usuários, mas também motivá-los a utilizar de forma regular.

## **1.7. Restrição**

O desenvolvimento será realizado em 90 dias, utilizando tecnologias que garantam um produto robusto e de baixo custo. No entanto, o projeto enfrenta algumas restrições importantes, como limitações de tempo que podem impactar a quantidade de trabalho realizado e a qualidade do produto. Restrições tecnológicas podem surgir devido às limitações de hardware e software, afetando a escolha das ferramentas de desenvolvimento e a arquitetura do sistema. Além disso, o projeto deve cumprir padrões rigorosos de segurança para proteger os dados dos usuários, o que pode aumentar a complexidade e o tempo de desenvolvimento.

## 2. METODOLOGIA

O desenvolvimento do Sistema de Gerenciamento de Tarefas adota uma abordagem ágil baseada no Scrum, combinada com métodos atuais de desenvolvimento de software. Essa abordagem visa entregar um produto de qualidade, alinhado às necessidades dos usuários, de forma iterativa e funcional.

A escolha do Scrum se justifica pela necessidade de flexibilidade para adaptar o projeto às mudanças de requisitos e pela importância da colaboração entre a equipe e os stakeholders. Através de sprints, a equipe entregará funcionalidades incrementáveis a cada ciclo, permitindo obter feedback constante e realizar ajustes de forma ágil.

Para a implementação do sistema, foram selecionadas tecnologias robustas e eficientes, como PHP, MySQL, HTML, CSS, Java Script e AWS. A combinação dessas tecnologias permite construir um sistema flexível, seguro e com uma interface intuitiva.

### 2.1. Atendimento aos Requisitos Funcionais e Não Funcionais

Para garantir o sucesso do nosso projeto, priorizamos a consideração tanto dos requisitos funcionais quanto dos não funcionais. A combinação cuidadosamente escolhida de tecnologias assegura que o sistema seja eficiente, seguro e capaz de atender às necessidades dos usuários. A divisão das tecnologias subentende-se da seguinte forma:

- Back-end: Utilização do PHP devido à sua flexibilidade e eficiência para desenvolvimento web.
- Banco de dados: Escolha do MySQL pela sua confiabilidade e velocidade em consultas de grandes volumes de dados.
- Front-end: Uso do HTML, CSS e Java Script para permitir a criação de uma interface de usuário dinâmica, responsiva e com layout amigável.
- Infraestrutura e plataforma: Hospedagem na AWS (Amazon Web Services), que oferece escalabilidade e segurança.

## **2.2. Descrição do Software Utilizado**

A escolha de cada tecnologia para o desenvolvimento deste sistema foi estratégica para otimizar seu desempenho e atender aos requisitos específicos do projeto. A divisão dos softwares subentende-se da seguinte forma:

- PHP: Linguagem de programação server-side utilizada para criar páginas dinâmicas e processar formulários.
- MySQL: Banco de dados relacional escolhido pela capacidade de armazenar grandes volumes de dados de maneira eficiente.
- HTML: Linguagem de marcação utilizada para estruturar e apresentar o conteúdo da web.
- CSS: Linguagem de estilos utilizada para customizar a aparência das telas.
- Java Script: Linguagem de programação que adiciona interatividade e funcionalidades dinâmicas às páginas web.
- AWS: Utilizada para garantir a segurança e acessibilidade da aplicação por meio da hospedagem em nuvem.

## **2.3. Descrição da Equipe e Partes Interessadas**

Nesta seção, detalharemos a composição da equipe de desenvolvimento e as partes interessadas no projeto. A estrutura e o funcionamento da equipe, juntamente com a identificação e gestão das expectativas das partes interessadas, são elementos cruciais para atingirmos os objetivos. Ao compreender os papéis de cada membro da equipe e as necessidades dos stakeholders, garantiremos que o projeto seja entregue dentro do prazo, com qualidade em conformidade com as metas estratégicas do negócio.

## **2.4. Equipe**

Para garantir a agilidade e a qualidade do desenvolvimento, adotaremos a metodologia ágil Scrum. A equipe será composta por seis membros, distribuídos da seguinte forma:

- Scrum Master: Facilitará o processo Scrum e removerá impedimentos.
- Product Owner: Coordenará a equipe e garantirá o cumprimento dos prazos.
- Desenvolvedor Front-end: Desenvolverá a interface gráfica do sistema.
- Desenvolvedor Back-end: Gerenciará a lógica do sistema e integração com o banco de dados.
- Testadora de Qualidade: Realizará testes para garantir a integridade do sistema.
- Designer de Interface: Criará layouts intuitivos e focará na experiência do usuário.

## **2.5. Partes Interessadas**

O sucesso de um projeto depende em grande maioria da satisfação das partes interessadas. Por isso, ele está estruturado da seguinte maneira:

- Clientes do Projeto: Usuários finais do sistema, cuja expectativa será gerenciar tarefas de forma eficiente.
- Equipe de Desenvolvimento: Responsável pela implementação e entrega do sistema.
- Gestores Funcionais, Patrocinadores e Usuários Finais: Interessados diretos no sucesso e na qualidade da solução.

## **2.6. Recursos Utilizados**

O sucesso de um projeto depende em grande maioria da satisfação das partes interessadas. Por isso, ele está estruturado da seguinte maneira:

- Clientes do Projeto: Usuários finais do sistema, cuja expectativa será gerenciar tarefas de forma eficiente.
- Equipe de Desenvolvimento: Responsável pela implementação e entrega do sistema.
- Gestores Funcionais, Patrocinadores e Usuários Finais: Interessados diretos

no sucesso e na qualidade da solução.

## **2.7. Equipamentos**

A escolha dos equipamentos certos é um fator determinante para o sucesso de um projeto de software. Afinal, são eles que fornecem a base para o desenvolvimento, testes e, posteriormente, a operação do sistema, para isso utilizaremos:

- Computadores (desktops e notebooks) para o desenvolvimento e teste do sistema.
- Servidores na nuvem (AWS) para hospedagem do sistema, garantindo acessibilidade e segurança.

## **2.8. Técnicas e Processos**

O desenvolvimento será baseado na metodologia Scrum, com ciclos de Sprints. A cada Sprint, o time realizará um planejamento detalhado, reuniões diárias para acompanhar o progresso, uma demonstração do incremento do produto e uma retrospectiva para identificar melhorias. Os papéis de Product Owner, Scrum Master e time de desenvolvimento estão bem definidos e colaborarão de forma ágil para garantir o sucesso do projeto.

## **2.9. Ferramentas de Gerenciamento do Projeto**

Para garantir a entrega bem-sucedida e a qualidade do projeto, adotamos um conjunto estratégico de ferramentas que integram as diversas fases do desenvolvimento:

- Trello: Para o acompanhamento das tarefas e sprints.
- GitHub: Controle de versão para gerenciar o código-fonte.
- Microsoft Teams: Comunicação contínua entre os membros da equipe.

## **2.10. Segurança e Privacidade de Dados**

A segurança dos dados dos usuários é uma de nossas prioridades. Por isso, nosso sistema será desenvolvido com várias camadas de proteção, garantindo que a utilização de nossa plataforma seja totalmente segura. Para isso, adotaremos as seguintes medidas:

- Criptografia: Todos os dados sensíveis, como senhas, serão criptografados.
- Conformidade com a LGPD: O sistema será desenvolvido em conformidade com a legislação de proteção de dados (LGPD), garantindo privacidade dos usuários.

## **2.11. Plano de Testes**

A segurança dos dados dos usuários é uma de nossas prioridades. Por isso, nosso sistema será desenvolvido com várias camadas de proteção, garantindo que a utilização de nossa plataforma seja totalmente segura. Para isso, adotaremos as seguintes medidas:

- Criptografia: Todos os dados sensíveis, como senhas, serão criptografados.
- Conformidade com a LGPD: O sistema será desenvolvido em conformidade com a legislação de proteção de dados (LGPD), garantindo privacidade dos usuários.

## **Critérios de Sucesso e Métricas de Desempenho**

Para avaliar o sucesso do projeto, serão aplicados critérios e métricas de desempenho específicos. Esses indicadores permitirão monitorar de perto o progresso e a qualidade do sistema, incluindo:

- Número de usuários: Crescimento do número de usuários ativos no sistema.
- Satisfação dos usuários: Avaliação contínua de feedbacks dos usuários para melhorias.
- Performance: O sistema deve responder rapidamente, mesmo com alta demanda.

### 3. DESENVOLVIMENTO

#### 3.1. Casos de Uso (Anexo I)

Visando proporcionar uma experiência intuitiva e eficiente para o usuário, foram definidos os casos de uso. Desta forma, o sistema permitirá que os usuários se cadastrem, acessem e gerenciem suas tarefas, além do recebimento de notificações de forma prática, por meio dos dispositivos. Sendo assim os principais casos de uso são:

1. Cadastro de Usuários: O sistema permite que os usuários se registrem, criando uma conta com nome de usuário, e-mail e senha.
2. Autenticação de Usuários: Usuários registrados poderão fazer login e acessar suas listas de tarefas.
3. Gerenciamento de Tarefas: Os usuários podem criar, editar, excluir e marcar tarefas como concluídas.
4. Notificações: Os usuários receberão lembretes automáticos sobre suas tarefas pendentes.
5. Acesso Multidispositivo: O sistema será acessível por meio de dispositivos móveis e desktops.

Esses casos de uso fornecem a base funcional para o desenvolvimento do sistema, conforme descrito no backlog do produto e detalhado nas sprints do projeto.

#### Estrutura do Diagrama de Classe – (Anexo II)

Este diagrama de classes é baseado no modelo Entidade-Relacionamento (EER) no MySQL representando uma estrutura visual do sistema de gerenciamento de tarefas, mostrando como as diferentes entidades (classes) estão relacionadas. Cada classe tem atributos (dados) e métodos (funções) que definem seu comportamento. Os relacionamentos entre as classes indicam como elas interagem ou dependem umas das outras:

##### **Tabela de Usuários Atributos:**

- id: Identificador único do usuário (chave primária, tipo INT).

- nome\_usuario: Nome do usuário (VARCHAR).
- email: Endereço de e-mail do usuário (VARCHAR, deve ser único).
- senha: Senha do usuário (VARCHAR, criptografada).
- data\_criacao: Data de criação do registro (DATETIME).

**Função:** Armazena as informações de login e cadastro dos usuários.

#### **Tabela de Tarefas Atributos:**

- id: Identificador único da tarefa (chave primária, tipo INT).
- id\_usuario: Relaciona a tarefa ao usuário que a criou (chave estrangeira para usuarios.id).
- titulo: Título da tarefa (VARCHAR).
- descricao: Descrição detalhada da tarefa (TEXT).
- prazo: Data limite para conclusão da tarefa (DATETIME).
- status: Enum (ENUM) para indicar o status da tarefa (ex: "pendente", "concluída").
- prioridade: Nível de prioridade da tarefa (ENUM: "alta", "média", "baixa").
- categoria: Categoria da tarefa (VARCHAR, opcional).
- data\_criacao: Data de criação da tarefa (DATETIME).
- data\_atualizacao: Data da última atualização da tarefa (DATETIME).

**Função:** Armazena as informações das tarefas que os usuários criam e gerenciam.

#### **Tabela de Recuperação de Senha Atributos:**

- id: Identificador único do registro de recuperação de senha (chave primária, tipo INT).
- id\_usuario: Referência ao usuário que solicitou a recuperação (chave estrangeira para usuarios.id).

- token: Token gerado para a recuperação de senha (VARCHAR).
- valido\_ate: Data de validade do token (DATETIME).
- utilizado: Indica se o token foi utilizado ou não (BOOLEAN).

**Função:** Gerencia o processo de recuperação de senha.

#### **Tabela de Notificações Atributos:**

- id: Identificador único da notificação (chave primária, tipo INT).
- id\_tarefa: Relaciona a notificação a uma tarefa específica (chave estrangeira para tarefas.id).
- mensagem: Mensagem da notificação (TEXT).
- data\_notificacao: Data e hora em que a notificação foi gerada (DATETIME).

**Função:** Armazena as notificações relacionadas às tarefas para alertar os usuários de eventos importantes.

#### **Tabela Histórico de Atividades Atributos:**

- id: Identificador único do registro de atividade (chave primária, tipo INT).
- id\_usuario: Referência ao usuário que realizou a ação (chave estrangeira para usuarios.id).
- acao: Descrição da ação realizada (VARCHAR, ex: "criação", "atualização", "exclusão").
- id\_tarefa: Referência à tarefa associada à ação (chave estrangeira para tarefas.id).
- data\_acao: Data e hora em que a ação foi realizada (DATETIME).

**Função:** Mantém um registro das atividades realizadas pelos usuários no sistema, permitindo rastrear todas as ações.

#### **Relações:**

- Usuários e Tarefas: Um usuário pode criar várias tarefas, representado pela

chave estrangeira `id\_usuario` na tabela `tarefas`.

- Tarefas e Notificações: Cada tarefa pode gerar várias notificações, representado pela chave estrangeira `id\_tarefa` na tabela `notificacoes`.
- Usuários e Recuperação de Senha: Cada usuário pode ter vários registros de recuperação de senha, representado pela chave estrangeira `id\_usuario` na tabela `recuperacao\_senha`.
- Usuários e Histórico de Atividades: O histórico de atividades também está ligado aos usuários, onde cada atividade registrada refere-se a uma ação específica realizada por um usuário em uma tarefa.

#### **Fluxo do Processo:**

- Criação de Usuário: Um usuário é criado e suas informações são armazenadas na tabela `usuarios`.
- Criação de Tarefa: O usuário pode criar uma tarefa, que será armazenada na tabela `tarefas`, junto com a data de criação, status e prioridade.
- Notificação: Quando algo importante relacionado à tarefa acontece, uma notificação é gerada e armazenada em `notificacoes`.
- Recuperação de Senha: Se o usuário esquecer sua senha, o sistema gera um token de recuperação, armazenado em `recuperacao\_senha`.
- Histórico de Atividades: Todas as ações realizadas pelos usuários, como a criação ou atualização de uma tarefa, são registradas no `historico\_atividades`.

### **3.2. Desenvolvimento das Estruturas do banco de Dados em MySQL**

-- Criação do banco de dados

```
CREATE DATABASE IF NOT EXISTS sistema_gerenciamento_tarefas;
```

```
USE sistema_gerenciamento_tarefas;
```

-- Tabela de Usuários

```
CREATE TABLE IF NOT EXISTS usuarios (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nome_usuario VARCHAR(50) NOT NULL,
    email VARCHAR(100) NOT NULL UNIQUE,
    senha VARCHAR(255) NOT NULL, -- Armazenamento de senha com hash
    data_criacao TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

-- Tabela de Tarefas

```
CREATE TABLE IF NOT EXISTS tarefas (
    id INT AUTO_INCREMENT PRIMARY KEY,
    id_usuario INT NOT NULL,
    titulo VARCHAR(255) NOT NULL,
    descricao TEXT,
    prazo DATE,
    status ENUM('pendente', 'concluida', 'atrasada') DEFAULT 'pendente',
    prioridade ENUM('baixa', 'media', 'alta') DEFAULT 'media',
    categoria VARCHAR(100), -- Categoria da tarefa
    data_criacao TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    data_atualizacao TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON
UPDATE CURRENT_TIMESTAMP,
    FOREIGN KEY (id_usuario) REFERENCES usuarios(id) ON DELETE
CASCADE
);
```

-- Tabela de Notificações

```

CREATE TABLE IF NOT EXISTS notificacoes (
    id INT AUTO_INCREMENT PRIMARY KEY,
    id_tarefa INT NOT NULL,
    mensagem VARCHAR(255),
    data_notificacao TIMESTAMP,
    FOREIGN KEY (id_tarefa) REFERENCES tarefas(id) ON DELETE
CASCADE
);

```

-- Tabela de Histórico de Atividades

```

CREATE TABLE IF NOT EXISTS historico_atividades (
    id INT AUTO_INCREMENT PRIMARY KEY,
    id_usuario INT NOT NULL,
    acao VARCHAR(255), -- Descrição da ação (ex: "Tarefa criada", "Tarefa
concluída")
    id_tarefa INT, -- A tarefa relacionada (opcional)
    data_acao TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (id_usuario) REFERENCES usuarios(id) ON DELETE
CASCADE,
    FOREIGN KEY (id_tarefa) REFERENCES tarefas(id) ON DELETE
CASCADE
);

```

-- Tabela de Tokens para Recuperação de Senha

```

CREATE TABLE IF NOT EXISTS recuperacao_senha (
    id INT AUTO_INCREMENT PRIMARY KEY,
    id_usuario INT NOT NULL,

```

```

        token VARCHAR(255) NOT NULL UNIQUE,    -- Token único para
recuperação

        valido_ate TIMESTAMP, -- Data de expiração do token

        utilizado BOOLEAN DEFAULT FALSE, -- Indica se o token já foi utilizado

        FOREIGN KEY (id_usuario) REFERENCES usuarios(id) ON DELETE
CASCADE

    );

```

-- Índices para Filtros e Ordenação

-- Índice para melhorar a busca por status

```
CREATE INDEX idx_status ON tarefas(status);
```

-- Índice para busca por data de criação (ordenar tarefas por data)

```
CREATE INDEX idx_data_criacao ON tarefas(data_criacao);
```

-- Índice para busca por prioridade

```
CREATE INDEX idx_prioridade ON tarefas(prioridade);
```

-- Índice para busca por categoria

```
CREATE INDEX idx_categoria ON tarefas(categoria);
```

-- Consulta de Tarefas com Filtro e Ordenação (Exemplo)

-- Tarefas pendentes, ordenadas por prazo e prioridade

```
SELECT titulo, descricao, prazo, prioridade
```

```
FROM tarefas
```

```
WHERE id_usuario = 1 AND status = 'pendente'
```

```
ORDER BY prazo ASC, prioridade DESC;
```

**Figure 1** - Banco de dados do sistema de gerenciamento de tarefas.

```

1  -- Criação do banco de dados
2  • CREATE DATABASE IF NOT EXISTS sistema_gerenciação_tarefas;
3  • USE sistema_gerenciação_tarefas;
4
5  -- Tabela de Usuários
6  • CREATE TABLE IF NOT EXISTS usuarios (
7      id INT AUTO_INCREMENT PRIMARY KEY,
8      nome_usuario VARCHAR(50) NOT NULL,
9      email VARCHAR(100) NOT NULL UNIQUE,
10     senha VARCHAR(255) NOT NULL, -- Armazenamento de senha com hash
11     data_criacao TIMESTAMP DEFAULT CURRENT_TIMESTAMP
12 );
13
14 -- Tabela de Tarefas
15 • CREATE TABLE IF NOT EXISTS tarefas (
16     id INT AUTO_INCREMENT PRIMARY KEY,
17     id_usuario INT NOT NULL,
18     titulo VARCHAR(255) NOT NULL,
19     descricao TEXT,
20     prazo DATE,
21     status ENUM('pendente', 'concluida', 'atrasada') DEFAULT 'pendente',
22     prioridade ENUM('baixa', 'media', 'alta') DEFAULT 'media',
23     categoria VARCHAR(100), -- Categoria da tarefa
24     data_criacao TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
25     data_atualizacao TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
26     FOREIGN KEY (id_usuario) REFERENCES usuarios(id)
27 );
28
29 -- Tabela de Notificações
30 • CREATE TABLE IF NOT EXISTS notificacoes (
31     id INT AUTO_INCREMENT PRIMARY KEY,
32     id_tarefa INT NOT NULL,
33     mensagem VARCHAR(255),
34     data_notificacao TIMESTAMP,
35     FOREIGN KEY (id_tarefa) REFERENCES tarefas(id)
36 );

```

```

38  -- Tabela de Histórico de Atividades
39  ● ⊖ CREATE TABLE IF NOT EXISTS historico_atividades (
40      id INT AUTO_INCREMENT PRIMARY KEY,
41      id_usuario INT NOT NULL,
42      acao VARCHAR(255), -- Descrição da ação (ex: "Tarefa criada", "Tarefa concluída")
43      id_tarefa INT, -- A tarefa relacionada (opcional)
44      data_acao TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
45      FOREIGN KEY (id_usuario) REFERENCES usuarios(id),
46      FOREIGN KEY (id_tarefa) REFERENCES tarefas(id) on delete cascade
47  );
48
49  -- Tabela de Tokens para Recuperação de Senha
50  ● ⊖ CREATE TABLE IF NOT EXISTS recuperacao_senha (
51      id INT AUTO_INCREMENT PRIMARY KEY,
52      id_usuario INT NOT NULL,
53      token VARCHAR(255) NOT NULL UNIQUE, -- Token único para recuperação
54      valido_ate TIMESTAMP, -- Data de expiração do token
55      utilizado BOOLEAN DEFAULT FALSE, -- Indica se o token já foi utilizado
56      FOREIGN KEY (id_usuario) REFERENCES usuarios(id)
57  );
58
59  -- Índices para Filtros e Ordenação
60  -- Índice para melhorar a busca por status
61  ● CREATE INDEX idx_status ON tarefas(status);
62  -- Índice para busca por data de criação (ordenar tarefas por data)
63  ● CREATE INDEX idx_data_criacao ON tarefas(data_criacao);
64  -- Índice para busca por prioridade
65  ● CREATE INDEX idx_prioridade ON tarefas(prioridade);
66  -- Índice para busca por categoria
67  ● CREATE INDEX idx_categoria ON tarefas(categoria);

```

**Fonte:** De autoria própria (2024).

### 3.3. Implementação do Sistema

#### Estrutura do Projeto

**Figure 2** - Estrutura do sistema de gerenciamento de tarefas.

```
/sistema-gerenciamento-tarefas
|-- /css
|   |-- style.css           # Arquivo de estilo
|-- /js
|   |-- script.js          # Arquivo JavaScript
|-- config.php              # Arquivo de configuração com conexão ao BD
|-- index.php               # Página de login
|-- register.php            # Página de registro
|-- dashboard.php           # Painel de tarefas
|-- add_task.php            # Adicionar tarefa
|-- edit_task.php           # Editar tarefa
|-- delete_task.php         # Excluir tarefa
|-- historico.php           # Histórico de atividades
|-- logout.php              # Logout
```

Fonte: De autoria própria (2024).

#### 3.3.1. Conexão com o Banco de Dados (config.php)

Este arquivo estabelece a conexão com o banco de dados MySQL.

```
<?php
// config.php - Conexão com o banco de dados MySQL
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "sistema_gerenciamento_tarefas";

// Conexão com o MySQL
$conn = new mysqli($servername, $username, $password, $dbname);

// Verificar conexão
if ($conn->connect_error) {
    die("Falha na conexão: " . $conn->connect_error);
}
?>
```

### 3.3.2. Página de Login (index.php)

A página de login autentica o usuário.

```
<?php
include 'config.php';
session_start();

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $email = filter_var($_POST['email'], FILTER_SANITIZE_EMAIL);
    $senha = $_POST['senha'];
    $lembrar = isset($_POST['lembrar']);

    $stmt = $conn->prepare("SELECT * FROM usuarios WHERE email = ?");
    $stmt->bind_param("s", $email);
    $stmt->execute();
    $result = $stmt->get_result();

    if ($result->num_rows > 0) {
        $row = $result->fetch_assoc();
        if (password_verify($senha, $row['senha'])) {
            $_SESSION['id_usuario'] = $row['id'];
            $_SESSION['nome_usuario'] = $row['nome_usuario'];
            session_regenerate_id(true);

            // Lógica para lembrar o email
            if ($lembrar) {
                setcookie('email', $email, time() + (30 * 24 * 60 * 60), "/",
"", true, true);
            } else {
                // Remover o cookie se não quiser lembrar
                setcookie('email', '', time() - 3600, "/");
            }

            header("Location: dashboard.php");
            exit();
        } else {
            $error_message = "Credenciais inválidas!";
        }
    } else {
        $error_message = "Credenciais inválidas!";
    }
}

// Verificação do cookie para auto-login
$email = isset($_COOKIE['email']) ? htmlspecialchars($_COOKIE['email']) : '';
?>

<!DOCTYPE html>
```

```

<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Login</title>
  <link rel="stylesheet" href="style.css">
  <link href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css'
rel='stylesheet'>
</head>
<body>
  <div class="wrapper">
    <form action="index.php" method="POST">
      <h1>Login</h1>
      <?php if (isset($error_message)): ?>
        <div class="alert alert-danger"><?php echo $error_message;
?></div>
      <?php endif; ?>
      <div class="input-box">
        <input type="email" name="email" placeholder="E-mail"
value="<?php echo $email; ?>" required>
        <i class='bx bxs-user'></i>
      </div>
      <div class="input-box">
        <input type="password" name="senha" placeholder="Senha"
required>
        <i class='bx bxs-lock-alt'></i>
      </div>

      <div class="remember-forgot">
        <label><input type="checkbox" name="lembrar" <?php echo
isset($_COOKIE['email']) ? 'checked' : ''; ?>> Lembrar-me</label>
        <a href="forgot_password.php" class="forgot-password">Esqueceu
a senha?</a>
      </div>

      <button type="submit" class="btn">Login</button>

      <div class="register-link">
        <p>Ainda não tem uma conta? <a href="register.php">Criar
conta</a></p>
      </div>
    </form>
  </div>
</body>
</html>

```

### 3.3.3. Página de Registro (register.php)

Essa página permite registrar novos usuários.

```
<?php
// Registro de usuários
include 'config.php';
session_start();

$message = "";

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $nome_usuario = trim($_POST['nome_usuario']);
    $email = trim($_POST['email']);
    $senha = trim($_POST['senha']);

    if (empty($nome_usuario) || empty($email) || empty($senha)) {
        $message = "Todos os campos são obrigatórios.";
    } elseif (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        $message = "E-mail inválido.";
    } else {
        $senha_hashed = password_hash($senha, PASSWORD_DEFAULT);

        // Verificar se o e-mail já está em uso
        $stmt = $conn->prepare("SELECT COUNT(*) FROM usuarios WHERE email = ?");
        $stmt->bind_param("s", $email);
        $stmt->execute();
        $stmt->bind_result($count);
        $stmt->fetch();
        $stmt->close();

        if ($count > 0) {
            $message = "Este e-mail já está registrado. Faça o login.";
        } else {
            // Inserir o novo usuário
            $stmt = $conn->prepare("INSERT INTO usuarios (nome_usuario, email, senha) VALUES (?, ?, ?)");
            $stmt->bind_param("sss", $nome_usuario, $email, $senha_hashed);

            if ($stmt->execute()) {
                header("Location: index.php");
                exit();
            } else {
                $message = "Erro ao registrar usuário: " . $conn->error;
            }
            $stmt->close();
        }
    }
}
```

```

    // Armazena a mensagem na sessão para exibição posterior
    $_SESSION['message'] = $message;
    header("Location: register.php");
    exit();
}
?>

<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Registrar</title>
    <link rel="stylesheet" href="style.css">
    <link href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css'
rel='stylesheet'>
</head>
<body>
    <div class="wrapper">
        <form action="register.php" method="POST">
            <h1>Registrar</h1>
            <?php
                // Exibe a mensagem se existir
                if (isset($_SESSION['message']) && $_SESSION['message']) {
                    echo '<div class="alert alert-warning">' .
htmlspecialchars($_SESSION['message']) . '</div>';
                    unset($_SESSION['message']); // Limpa a mensagem após exibição
                }
            ?>
            <div class="input-box">
                <input type="text" name="nome_usuario" placeholder="Nome de
Usuário" class="form-control" required>
                <i class='bx bxs-user'></i>
            </div>
            <div class="input-box">
                <input type="email" name="email" placeholder="E-mail"
class="form-control" required>
                <i class='bx bxs-envelope'></i>
            </div>
            <div class="input-box">
                <input type="password" name="senha" placeholder="Senha"
class="form-control" required>
                <i class='bx bxs-lock-alt'></i>
            </div>
            <button type="submit" class="btn btn-primary w-
100">Registrar</button>
        </form>
    </div>
</body>
</html>

```

```

        <div class="register-link">
            <p>Já tem uma conta? <a href="index.php">Faça login</a></p>
        </div>
    </div>
</body>
</html>

```

### 3.3.4. Página Recuperação de Senha (forgot\_password.php)

Essa página permite o usuário recuperar a senha.

```

<?php
// Recuperação de senha
include 'config.php';
session_start();

$message = "";

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $email = $_POST['email'];

    // Verificar se o e-mail existe
    $stmt = $conn->prepare("SELECT COUNT(*) FROM usuarios WHERE email = ?");
    $stmt->bind_param("s", $email);
    $stmt->execute();
    $stmt->bind_result($count);
    $stmt->fetch();
    $stmt->close();

    if ($count == 0) {
        $message = "Este e-mail não está registrado.";
    } else {

        $token = bin2hex(random_bytes(50));

        $message = "Um e-mail foi enviado para você com instruções de
recuperação.";
    }

    // Armazena a mensagem na sessão para exibição posterior
    $_SESSION['message'] = $message;
    header("Location: forgot_password.php");
    exit();
}
?>

```

```

<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Esqueceu a Senha</title>
    <link rel="stylesheet" href="style.css">
    <link href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css'
rel='stylesheet'>
</head>
<body>
    <div class="wrapper">
        <form action="forgot_password.php" method="POST">
            <h1>Recuperar Senha</h1>
            <?php
                // Exibe a mensagem se existir
                if (isset($_SESSION['message']) && $_SESSION['message']) {
                    echo '<div class="alert alert-warning">' .
htmlspecialchars($_SESSION['message']) . '</div>';
                    unset($_SESSION['message']); // Limpa a mensagem após exibição
                }
            ?>
            <div class="input-box">
                <input type="email" name="email" placeholder="E-mail"
class="form-control" required>
                <i class='bx bxs-envelope'></i>
            </div>
            <button type="submit" class="btn btn-primary w-100">Recuperar
Senha</button>
        </form>
        <div class="register-link">
            <p><a href="index.php">Voltar ao login</a></p>
        </div>
    </div>
</body>
</html>

```

### 3.3.5. Painel de Tarefas Com Filtros e Relatórios (dashboard.php)

O painel exibe todas as tarefas do usuário, com filtros, ordenação, e relatórios de progresso.

```

<?php
// Painel de tarefas
include 'config.php';
session_start();

```

```

if (!isset($_SESSION['id_usuario'])) {
    header("Location: index.php");
    exit();
}

$id_usuario = $_SESSION['id_usuario'];

// Consultar tarefas
$filtro_status = isset($_GET['status']) ? $_GET['status'] : '';
$ordem = isset($_GET['ordem']) ? $_GET['ordem'] : 'prazo ASC';

$sql = "SELECT * FROM tarefas WHERE id_usuario = $id_usuario";
if ($filtro_status) {
    $sql .= " AND status = '$filtro_status'";
}
$sql .= " ORDER BY $ordem";
$result = $conn->query($sql);

// Consultar estatísticas
$sql_estatisticas = "SELECT
                        COUNT(IF(status = 'pendente', 1, NULL)) AS pendentes,
                        COUNT(IF(status = 'concluida', 1, NULL)) AS
concluidas,
                        COUNT(IF(prazo < CURDATE() AND status = 'pendente', 1,
NULL)) AS atrasadas
                        FROM tarefas
                        WHERE id_usuario = $id_usuario";
$result_estatisticas = $conn->query($sql_estatisticas);
$estatisticas = $result_estatisticas->fetch_assoc();

// Recuperar cor do tema do cookie
$themeColor = isset($_COOKIE['theme_color']) ? $_COOKIE['theme_color'] :
'F5B7B1';
?>

<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Painel de Tarefas</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
    <link href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css'
rel='stylesheet'>
    <style>
        body {

```

```

        background-color: <?php echo $themeColor; ?>;
        transition: background-color 0.3s ease;
        margin: 0;
        padding: 0;
        font-family: Arial, sans-serif;
        display: flex;
        justify-content: center;
        align-items: center;
        min-height: 100vh;
    }

    .container {
        max-width: 1080px;
        padding: 40px;
        background-color: #FFF;
        border: 2px solid rgba(255, 255, 255, 0.2);
        backdrop-filter: blur(20px);
        box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);
        border-radius: 10px;
        box-shadow: 0px 4px 12px rgba(0, 0, 0, 0.1);
    }

    .btn i {
        margin-right: 5px;
    }

    .theme-toggle i {
        font-size: 20px;
    }

    .form-label {
        font-weight: bold;
    }

    .btn-adicionar {
        width: 35%;
        height: 45px;
        padding: 0 10px;
        background: #198754;
        border: none;
        outline: none;
        border-radius: 40px;
        box-shadow: 0 0 10px rgba(0, 0, 0, .1);
        cursor: pointer;
        font-size: 16px;
        color: #FFF;
        font-weight: 600;
        display: flex;
        justify-content: center;
    }

```

```

        align-items: center;
    }

    .btn-adicionar i {
        font-size: 18px;
    }

    table {
        width: 100%;
    }

    .card {
        margin-top: 20px;
    }

    .text-center {
        text-align: center;
    }

    .filter-container {
        margin-bottom: 20px;
    }

    .form-select {
        width: 100%;
    }
</style>
</head>
<body id="body">

    <div class="container">
        <h2 class="text-center">Bem-vindo, <?php echo
$_SESSION['nome_usuario']; ?>!</h2>
        <br>

        <!-- Botão Alterar Tema -->
        <button id="theme-toggle" class="btn mb-3" style="background-color:
<?php echo $themeColor; ?>; color: white; font-weight: bold;">
            Alterar Tema <i class='bx bx-palette'></i>
        </button>

        <!-- Link Adicionar Tarefa -->
        <a href="add_task.php" class="btn btn-success mb-3" style="color:
white; font-weight: bold;">
            Adicionar Tarefa <i class='bx bx-plus'></i>
        </a>

        <!-- Filtros e Ordenação -->
        <form method="GET" action="dashboard.php" class="filter-container">

```

```

        <div class="row">
            <div class="col-md-5">
                <label for="status" class="form-label">Filtrar por
Status</label>
                <select name="status" class="form-select">
                    <option value="">Todos</option>
                    <option value="pendente" <?php if ($filtro_status ==
'pendente') echo 'selected'; ?>>Pendente</option>
                    <option value="concluida" <?php if ($filtro_status ==
'concluida') echo 'selected'; ?>>Concluída</option>
                </select>
            </div>
            <div class="col-md-5">
                <label for="ordem" class="form-label">Ordenar por</label>
                <select name="ordem" class="form-select">
                    <option value="prazo ASC" <?php if ($ordem == 'prazo
ASC') echo 'selected'; ?>>Prazo Crescente</option>
                    <option value="prazo DESC" <?php if ($ordem == 'prazo
DESC') echo 'selected'; ?>>Prazo Decrescente</option>
                </select>
            </div>
            <div class="col-md-2 d-flex align-items-end">
                <button type="submit" class="btn btn-primary w-
100">Aplicar</button>
            </div>
        </div>
    </form>

    <!-- Tabela de Tarefas -->
    <table class="table table-striped">
        <thead>
            <tr>
                <th class="text-center">Título</th>
                <th class="text-center">Descrição</th>
                <th class="text-center">Prazo</th>
                <th class="text-center">Status</th>
                <th class="text-center">Ações</th>
            </tr>
        </thead>
        <tbody>
            <?php while($row = $result->fetch_assoc()): ?>
                <tr>
                    <td class="text-center"><?php echo $row['titulo']; ?></td>
                    <td class="text-center"><?php echo $row['descricao'];
?></td>
                    <td class="text-center"><?php echo $row['prazo']; ?></td>
                    <td class="text-center"><?php echo $row['status']; ?></td>
                    <td class="text-center">

```

```

        <a href="edit_task.php?id=<?php echo $row['id']; ?>"
class="btn btn-warning btn-sm">Editar</a>
        <a href="delete_task.php?id=<?php echo $row['id']; ?>"
class="btn btn-danger btn-sm">Excluir</a>
    </td>
</tr>
<?php endwhile; ?>
</tbody>
</table>

<!-- Resumo de Tarefas -->
<div class="card mt-4">
    <div class="card-body">
        <h5 class="card-title">Resumo de Tarefas</h5>
        <ul>
            <li>Pendentes: <?php echo $estatisticas['pendentes'];
?></li>
            <li>Concluídas: <?php echo $estatisticas['concluidas'];
?></li>
            <li>Atrasadas: <?php echo $estatisticas['atrasadas'];
?></li>
        </ul>
    </div>
</div>

<!-- Botão Sair -->
<div class="text-center">
    <a href="logout.php" class="btn btn-adicionar mt-3" style="width:
30%; border-radius: 5px;">
        <i class='bx bx-log-out' style="font-size: 18px; margin-right:
5px;"></i> Sair
    </a>
</div>

<script>
    // Alternar tema
    const themeToggleButton = document.getElementById('theme-toggle');
    const body = document.getElementById('body');

    // Lista de cores a serem aplicadas no fundo
    const colors = ['#BF2C98', '#9512A6', '#1D0259', '#150140', '#090126',
'#F5B7B1', '#FF9F66', '#FF5F00', '#002379', '#00C896', '#A9DFBF', '#E8DAEF',
'#FAD7A1'];

    let currentColorIndex = colors.indexOf("<?php echo $themeColor; ?>");
    // Define a cor atual

    // Função de troca de cor ao clicar no botão
    themeToggleButton.addEventListener('click', () => {

```

```

        currentColorIndex = (currentColorIndex + 1) % colors.length;
        const newColor = colors[currentColorIndex];

        body.style.backgroundColor = newColor;
        themeToggleButton.style.backgroundColor = newColor; // Atualiza a
        cor do botão para o novo fundo

        // Salva a cor no cookie por 365 dias
        document.cookie = "theme_color=" + newColor + "; path=/; max-age="
+ (365 * 24 * 60 * 60);
    });
</script>

</body>
</html>

```

### 3.3.6. Adicionar Tarefas com Registro no Histórico (add\_task.php)

O usuário pode adicionar novas tarefas, e a ação será registrada no histórico.

```

<?php
// Adicionar tarefas com histórico
include 'config.php';
session_start();

if (!isset($_SESSION['id_usuario'])) {
    header("Location: index.php");
    exit();
}

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $id_usuario = $_SESSION['id_usuario'];
    $titulo = mysqli_real_escape_string($conn, $_POST['titulo']);
    $descricao = mysqli_real_escape_string($conn, $_POST['descricao']);
    $prazo = mysqli_real_escape_string($conn, $_POST['prazo']);

    // Inserir a nova tarefa
    $sql_tarefa = "INSERT INTO tarefas (id_usuario, titulo, descricao, prazo,
status)
                    VALUES ('$id_usuario', '$titulo', '$descricao', '$prazo',
'pendente')";

    if ($conn->query($sql_tarefa) === TRUE) {
        // Obter o ID da tarefa recém-inserida
        $id_tarefa = $conn->insert_id;
    }
}

```

```

        // Registrar a ação no histórico
        $acao = "Criou a tarefa: $titulo";
        $sql_historico = "INSERT INTO historico_atividades (id_usuario,
id_tarefa, acao)
                        VALUES ('$id_usuario', '$id_tarefa', '$acao')";
        $conn->query($sql_historico);

        header("Location: dashboard.php");
    } else {
        echo "Erro ao adicionar tarefa: " . $conn->error;
    }
}

// Recuperar cor do tema do cookie
$themeColor = isset($_COOKIE['theme_color']) ? $_COOKIE['theme_color'] :
'#F5B7B1';
?>

<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <title>Adicionar Tarefa</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
    <link href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css'
rel='stylesheet'>
    <style>
        body {
            background-color: <?php echo $themeColor; ?>;
            transition: background-color 0.3s ease;
            margin: 0;
            padding: 0;
            font-family: Arial, sans-serif;
            display: flex;
            justify-content: center;
            align-items: center;
            min-height: 100vh;
        }

        .container {
            max-width: 680px;
            padding: 40px;
            background-color: #FFF;
            border: 2px solid rgba(255, 255, 255, 0.2);
            backdrop-filter: blur(20px);
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);
            border-radius: 10px;

```

```

        box-shadow: 0px 4px 12px rgba(0, 0, 0, 0.1);
    }

    .btn i {
        margin-right: 5px;
    }

    .theme-toggle i {
        font-size: 20px;
    }

    .form-label {
        font-weight: bold;
    }

    .btn-adicionar {
        width: 100%;
        height: 45px;
        background: #198754;
        border: none;
        outline: none;
        border-radius: 40px;
        box-shadow: 0 0 10px rgba(0, 0, 0, .1);
        cursor: pointer;
        font-size: 16px;
        color: #FFF;
        font-weight: 600;
    }
</style>
</head>
<body>
    <div class="container">
        <h2>Adicionar Tarefa</h2>
        <form action="add_task.php" method="POST">
            <div class="mb-3">
                <label for="titulo" class="form-label">Título</label>
                <input type="text" name="titulo" class="form-control"
required>
            </div>
            <div class="mb-3">
                <label for="descricao" class="form-label">Descrição</label>
                <textarea name="descricao" class="form-control"></textarea>
            </div>
            <div class="mb-3">
                <label for="prazo" class="form-label">Prazo</label>
                <input type="date" name="prazo" class="form-control" required>
            </div>
            <button type="submit" class="btn-adicionar">Adicionar
Tarefa</button>

```

```

        </form>
    </div>

    <script>
        // Alternar tema
        const body = document.body;

        // Lista de cores a serem aplicadas no fundo
        const colors = ['#BF2C98', '#9512A6', '#1D0259', '#150140', '#090126',
            '#F5B7B1', '#FF9F66', '#FF5F00', '#002379', '#00C896', '#A9DFBF', '#E8DAEF',
            '#FAD7A1'];

        let currentIndex = colors.indexOf("<?php echo $themeColor; ?>"); //
        Define a cor atual

        // Função de troca de cor ao clicar no botão
        document.addEventListener('keydown', (event) => {
            if (event.key === 't') { // Muda a cor ao pressionar a tecla 't'
                currentIndex = (currentIndex + 1) % colors.length;
                body.style.backgroundColor = colors[currentIndex];

                // Salva a cor no cookie por 365 dias
                document.cookie = "theme_color=" + colors[currentIndex] + ";
                path=/; max-age=" + (365 * 24 * 60 * 60);
            }
        });
    </script>

</body>
</html>

```

### 3.3.7. Editar Tarefas (edit\_task.php)

Permite ao usuário editar as informações de uma tarefa existente.

```

<?php
//Editar tarefas
include 'config.php';
session_start();

$id_tarefa = $_GET['id'];
$id_usuario = $_SESSION['id_usuario'];

$sql = "SELECT * FROM tarefas WHERE id = $id_tarefa AND id_usuario =
$id_usuario";
$result = $conn->query($sql);
$tarefa = $result->fetch_assoc();

```

```

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $titulo = $_POST['titulo'];
    $descricao = $_POST['descricao'];
    $prazo = $_POST['prazo'];
    $status = $_POST['status'];

    $sql = "UPDATE tarefas SET titulo = '$titulo', descricao = '$descricao',
prazo = '$prazo', status = '$status'
        WHERE id = $id_tarefa AND id_usuario = $id_usuario";

    if ($conn->query($sql) === TRUE) {
        // Atualizar histórico
        $acao = "Editou a tarefa: $titulo";
        $sql_historico = "INSERT INTO historico_atividades (id_usuario,
id_tarefa, acao)
                        VALUES ('$id_usuario', '$id_tarefa', '$acao')";
        $conn->query($sql_historico);

        header("Location: dashboard.php");
    } else {
        echo "Erro ao atualizar tarefa: " . $conn->error;
    }
}

// Recuperar cor do tema do cookie
$themeColor = isset($_COOKIE['theme_color']) ? $_COOKIE['theme_color'] :
'#F5B7B1';
?>

<!DOCTYPE html>
<html Lang="pt-br">
<head>
    <meta charset="UTF-8">
    <title>Editar Tarefa</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
    <link href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css'
rel='stylesheet'>
    <style>
        body {
            background-color: <?php echo $themeColor; ?>;
            transition: background-color 0.3s ease;
            margin: 0;
            padding: 0;
            font-family: Arial, sans-serif;
            display: flex;
            justify-content: center;
            align-items: center;
            min-height: 100vh;

```

```

    }

    .container {
        max-width: 680px;
        padding: 40px;
        background-color: #FFF;
        border: 2px solid rgba(255, 255, 255, 0.2);
        backdrop-filter: blur(20px);
        box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);
        border-radius: 10px;
        box-shadow: 0px 4px 12px rgba(0, 0, 0, 0.1);
    }

    .btn i {
        margin-right: 5px;
    }

    .theme-toggle i {
        font-size: 20px;
    }

    .form-label {
        font-weight: bold;
    }

    .btn-atualizar {
        width: 100%;
        height: 45px;
        background: #FFC107;
        border: none;
        outline: none;
        border-radius: 40px;
        box-shadow: 0 0 10px rgba(0, 0, 0, .1);
        cursor: pointer;
        font-size: 16px;
        color: #333;
        font-weight: 600;
    }
</style>
</head>
<body>
    <div class="container">
        <h2>Editar Tarefa</h2>
        <form action="edit_task.php?id=<?php echo $id_tarefa; ?>"
method="POST">
            <div class="mb-3">
                <label for="titulo" class="form-label">Título</label>
                <input type="text" name="titulo" class="form-control"
value="<?php echo $tarefa['titulo']; ?>" required>

```

```

    </div>
    <div class="mb-3">
        <label for="descricao" class="form-label">Descrição</label>
        <textarea name="descricao" class="form-control"><?php echo
$tarefa['descricao']; ?></textarea>
    </div>
    <div class="mb-3">
        <label for="prazo" class="form-label">Prazo</label>
        <input type="date" name="prazo" class="form-control"
value="<?php echo $tarefa['prazo']; ?>" required>
    </div>
    <div class="mb-3">
        <label for="status" class="form-label">Status</label>
        <select name="status" class="form-select">
            <option value="pendente" <?php if ($tarefa['status'] ==
'pendente') echo 'selected'; ?>>Pendente</option>
            <option value="concluida" <?php if ($tarefa['status'] ==
'concluida') echo 'selected'; ?>>Concluída</option>
        </select>
    </div>
    <button type="submit" class="btn-atualizar">Atualizar
Tarefa</button>
</form>
</div>

<script>
// Alternar tema
const body = document.body;

// Lista de cores a serem aplicadas no fundo
const colors = ['#BF2C98', '#9512A6', '#1D0259', '#150140', '#090126',
'#F5B7B1', '#FF9F66', '#FF5F00', '#002379', '#00C896', '#A9DFBF', '#E8DAEF',
'#FAD7A1'];

let currentIndex = colors.indexOf("<?php echo $themeColor; ?>"); //
Define a cor atual

// Função de troca de cor ao clicar no botão
document.addEventListener('keydown', (event) => {
    if (event.key === 't') { // Muda a cor ao pressionar a tecla 't'
        currentIndex = (currentIndex + 1) % colors.length;
        body.style.backgroundColor = colors[currentIndex];

        // Salva a cor no cookie por 365 dias
        document.cookie = "theme_color=" + colors[currentIndex] + ";
path=/; max-age=" + (365 * 24 * 60 * 60);
    }
});
</script>

```

```
</body>
</html>
```

### 3.3.8. Excluir Tarefas (delete\_task.php)

Permite ao usuário excluir uma tarefa, registrando a ação no histórico.

```
<?php
// Excluir tarefas
include 'config.php';
session_start();

$id_tarefa = $_GET['id'];
$id_usuario = $_SESSION['id_usuario'];

// Verificar se a tarefa existe
$sql = "SELECT * FROM tarefas WHERE id = $id_tarefa AND id_usuario =
$id_usuario";
$result = $conn->query($sql);
$tarefa = $result->fetch_assoc();

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    // Registrar a ação no histórico antes de excluir a tarefa
    if ($tarefa) {
        $acao = "Excluiu a tarefa: " . $tarefa['titulo'];
        $sql_historico = "INSERT INTO historico_atividades (id_usuario,
id_tarefa, acao)
                        VALUES ('$id_usuario', '$id_tarefa', '$acao')";
        $conn->query($sql_historico);
    }

    // Excluir a tarefa
    $sql = "DELETE FROM tarefas WHERE id = $id_tarefa AND id_usuario =
$id_usuario";

    if ($conn->query($sql) === TRUE) {
        header("Location: dashboard.php");
    } else {
        echo "Erro ao excluir tarefa: " . $conn->error;
    }
}

// Recuperar cor do tema do cookie
$themeColor = isset($_COOKIE['theme_color']) ? $_COOKIE['theme_color'] :
'#F5B7B1';
?>

<!DOCTYPE html>
```

```

<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Excluir Tarefa</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
  <link rel="stylesheet" href="css/style.css">
  <style>
    body {
      background-color: <?php echo $themeColor; ?>;
      transition: background-color 0.3s ease;
      margin: 0;
      padding: 0;
      font-family: Arial, sans-serif;
      display: flex;
      justify-content: center;
      align-items: center;
      min-height: 100vh;
    }

    .container {
      max-width: 680px;
      padding: 40px;
      background-color: #FFF;
      border: 2px solid rgba(255, 255, 255, 0.2);
      backdrop-filter: blur(20px);
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);
      border-radius: 10px;
      box-shadow: 0px 4px 12px rgba(0, 0, 0, 0.1);
    }

    .btn-danger, .btn-secondary {
      width: 45%;
      height: 45px;
      border-radius: 40px;
      box-shadow: 0 0 10px rgba(0, 0, 0, .1);
      font-size: 16px;
      font-weight: 600;
    }

    .button-container {
      display: flex;
      justify-content: space-between;
      margin-top: 20px;
    }

    .note {
      margin-top: 15px;
      font-size: 14px;

```

```

        font-weight: bold;
        color: #dc3545;
    }

    .question {
        margin-top: 10px;
        font-size: 16px;
    }
</style>
</head>
<body>
    <div class="container">
        <h2>Excluir Tarefa</h2>
        <p>Você tem certeza que deseja excluir a tarefa: <strong><?php echo
$task['titulo']; ?></strong>?</p>
        <p class="note">Após a exclusão, esta ação não poderá ser
desfeita.</p>
        <p class="question">Deseja continuar?</p>
        <form action="delete_task.php?id=<?php echo $id_task; ?>"
method="POST">
            <div class="button-container">
                <button type="submit" class="btn btn-danger">Excluir
Tarefa</button>
                <a href="dashboard.php" class="btn btn-secondary">Cancelar</a>
            </div>
        </form>
    </div>

    <script>
        // Alternar tema
        const body = document.body;

        // Lista de cores a serem aplicadas no fundo
        const colors = ['#BF2C98', '#9512A6', '#1D0259', '#150140', '#090126',
'#F5B7B1', '#FF9F66', '#FF5F00', '#002379', '#00C896', '#A9DFBF', '#E8DAEF',
'#FAD7A1'];

        let currentColorIndex = colors.indexOf("<?php echo $themeColor; ?>"); //
Define a cor atual

        // Função de troca de cor ao clicar no botão
        document.addEventListener('keydown', (event) => {
            if (event.key === 't') { // Muda a cor ao pressionar a tecla 't'
                currentColorIndex = (currentColorIndex + 1) % colors.length;
                body.style.backgroundColor = colors[currentColorIndex];

                // Salva a cor no cookie por 365 dias
                document.cookie = "theme_color=" + colors[currentColorIndex] + ";
path=/; max-age=" + (365 * 24 * 60 * 60);

```

```

    }
  });
</script>
</body>
</html>

```

### 3.3.9. Histórico de Atividades (histórico.php)

Exibe todas as ações realizadas pelo usuário no sistema.

```

<?php
// historico.php - Histórico de atividades
include 'config.php';
session_start();

if (!isset($_SESSION['id_usuario'])) {
    header("Location: index.php");
    exit();
}

$id_usuario = $_SESSION['id_usuario'];

// Consultar histórico de atividades
$sql = "SELECT h.*, t.titulo
        FROM historico_atividades h
        JOIN tarefas t ON h.id_tarefa = t.id
        WHERE h.id_usuario = $id_usuario
        ORDER BY h.data_acao DESC";
$result = $conn->query($sql);
?>

<!DOCTYPE html>
<html Lang="pt-br">
<head>
    <meta charset="UTF-8">
    <title>Histórico de Atividades</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet" href="css/style.css">
</head>
<body class="bg-light">
    <div class="container mt-5">
        <h2>Histórico de Atividades</h2>
        <table class="table table-striped">
            <thead>

```

```

        <tr>
            <th>Tarefa</th>
            <th>Ação</th>
            <th>Data</th>
        </tr>
    </thead>
    <tbody>
        <?php while($row = $result->fetch_assoc()): ?>
            <tr>
                <td><?php echo $row['titulo']; ?></td>
                <td><?php echo $row['acao']; ?></td>
                <td><?php echo $row['data_acao']; ?></td>
            </tr>
        <?php endwhile; ?>
    </tbody>
</table>
<a href="dashboard.php" class="btn btn-secondary mt-3">Voltar ao
Painel</a>
</div>
</body>
</html>

```

### 3.3.10. Encerrar Sessão (logout.php)

Encerra a sessão e redireciona o usuário para a página de login.

```

<?php
// logout.php - Encerrar sessão
session_start();
session_destroy();
header("Location: index.php");
exit();
?>

```

### 3.3.11. Estilo CSS Customizado (style.css)

Aqui estão os estilos personalizados para o sistema.

```

@import
url("https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700;800;900&display=swap");

* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: "Poppins", sans-serif;
}

```

```

}

body {
  display: flex;
  justify-content: center;
  align-items: center;
  min-height: 100vh;
  background: url('img.jpg') no-repeat;
  background-size: cover;
  background-position: center;
}

.wrapper {
  width: 420px;
  background: transparent;
  border: 2px solid rgba(255, 255, 255, .2);
  backdrop-filter: blur(20px);
  box-shadow: 0 0 10px rgba(0, 0, 0, .2);
  color: #FFF;
  border-radius: 10px;
  padding: 30px 40px;
}

.wrapper h1{
  font-size: 36px;
  text-align: center;
}

.wrapper .input-box {
  position: relative;
  width: 100%;
  height: 50px;
  margin: 30px 0;
}

.input-box input {
  width: 100%;
  height: 100%;
  background: transparent;
  border: none;
  outline: none;
  border: 2px solid rgba(255, 255, 255, .2);
  border-radius: 40px;
  font-size: 16px;
  color: #FFF;
  padding: 20px 45px 20px 20px;
}

```

```

.input-box input::placeholder {
  color: #FFF;
}

.input-box i {
  position: absolute;
  right: 20px;
  top: 50%;
  transform: translateY(-50%);
  font-size: 20px;
}

.wrapper .remember-forgot {
  display: flex;
  justify-content: space-between;
  font-size: 14.5px;
  margin: -15px 0 15px;
}

.remember-forgotLabel input {
  accent-color: #FFF;
  margin-right: 3px;
}

.remember-forgot a {
  color: #FFF;
  text-decoration: none;
}

.remember-forgot a:hover {
  text-decoration: underline;
}

.wrapper .btn {
  width: 100%;
  height: 45px;
  background: #FFF;
  border: none;
  outline: none;
  border-radius: 40px;
  box-shadow: 0 0 10px rgba(0, 0, 0, .1);
  cursor: pointer;
  font-size: 16px;
  color: #333;
  font-weight: 600;
}

.wrapper .register-link {

```

```
font-size: 14.5px;
text-align: center;
margin: 20px 0 15px;
}

.register-link p a {
  color: #FFF;
  text-decoration: none;
  font-weight: 600;
}

.register-link p a:hover {
  text-decoration: underline;
}
```

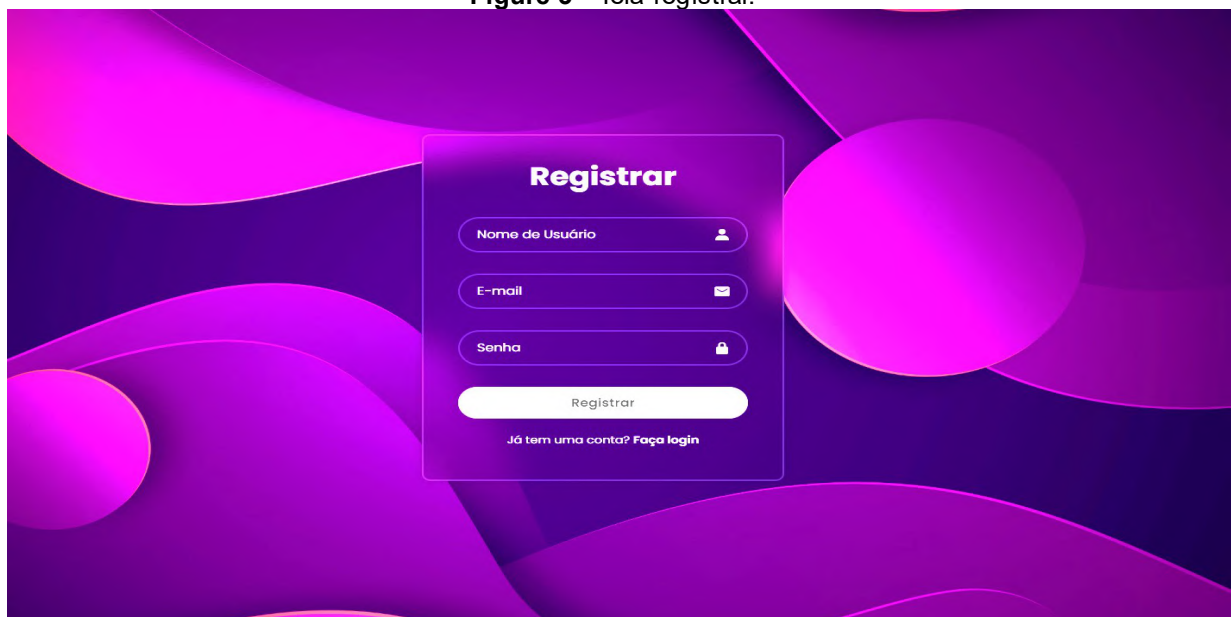
## 4. ARQUITETURA DO SOFTWARE

A arquitetura de software é a espinha dorsal de um sistema, definindo sua estrutura e comportamento. Ao projetar o sistema de gerenciamento de tarefas, a arquitetura escolhida influencia diretamente a interface do usuário, a organização interna dos dados e a eficiência das operações. Nesta apresentação, vamos explorar como os princípios de design se integram à arquitetura, resultando em uma interface intuitiva e uma experiência de usuário fluida, sem comprometer a funcionalidade e a escalabilidade do sistema.

### 4.1. Registrar

A tela de registrar é o primeiro passo para novos usuários se integrarem ao nosso sistema de gerenciamento de tarefas. Nesta tela, os usuários podem criar suas contas de forma simples e segura, fornecendo informações essenciais como nome, e-mail e senha.

Figure 3 - Tela registrar.

A imagem mostra a interface de usuário para o registro de uma nova conta. O formulário é centralizado e possui um fundo abstrato com formas orgânicas em tons de verde e amarelo. O formulário contém o título "Registrar" em negrito. Abaixo dele, há três campos de entrada: "Nome de Usuário" com um ícone de pessoa, "E-mail" com um ícone de envelope e "Senha" com um ícone de cadeado. Abaixo dos campos, há um botão "Registrar" em amarelo. Na base do formulário, há um link "Já tem uma conta? Faça login" em azul.

Fonte: De autoria própria (2024).

### 4.2. Recuperar Senha

A tela de recuperação de senha é uma etapa fundamental para garantir que os usuários possam acessar suas contas de forma segura. Nela, os usuários podem solicitar a redefinição de suas senhas, inserindo seu e-mail registrado.

**Figure 4** - Tela de recuperação de senha.

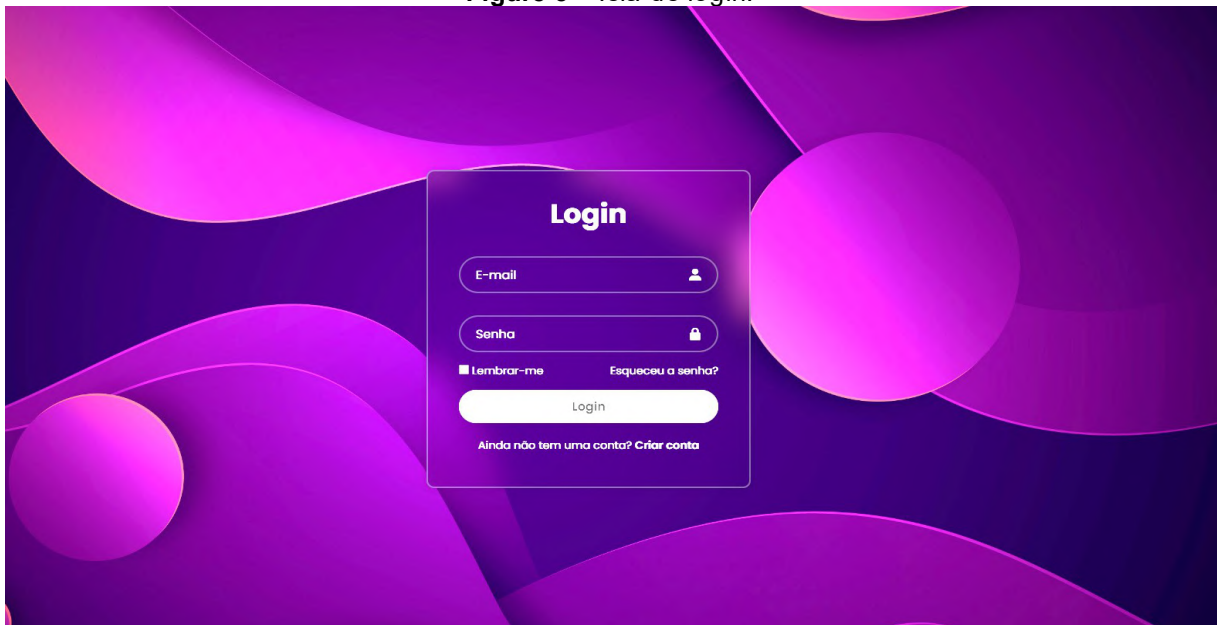


**Fonte:** De autoria própria (2024).

### 4.3. Login

A tela de login permite que os usuários acessem suas contas de forma segura. Nela, o usuário deve inserir suas credenciais como e-mail e senha, garantindo que apenas pessoas autorizadas tenham acesso ao sistema.

**Figure 5** - Tela de login.

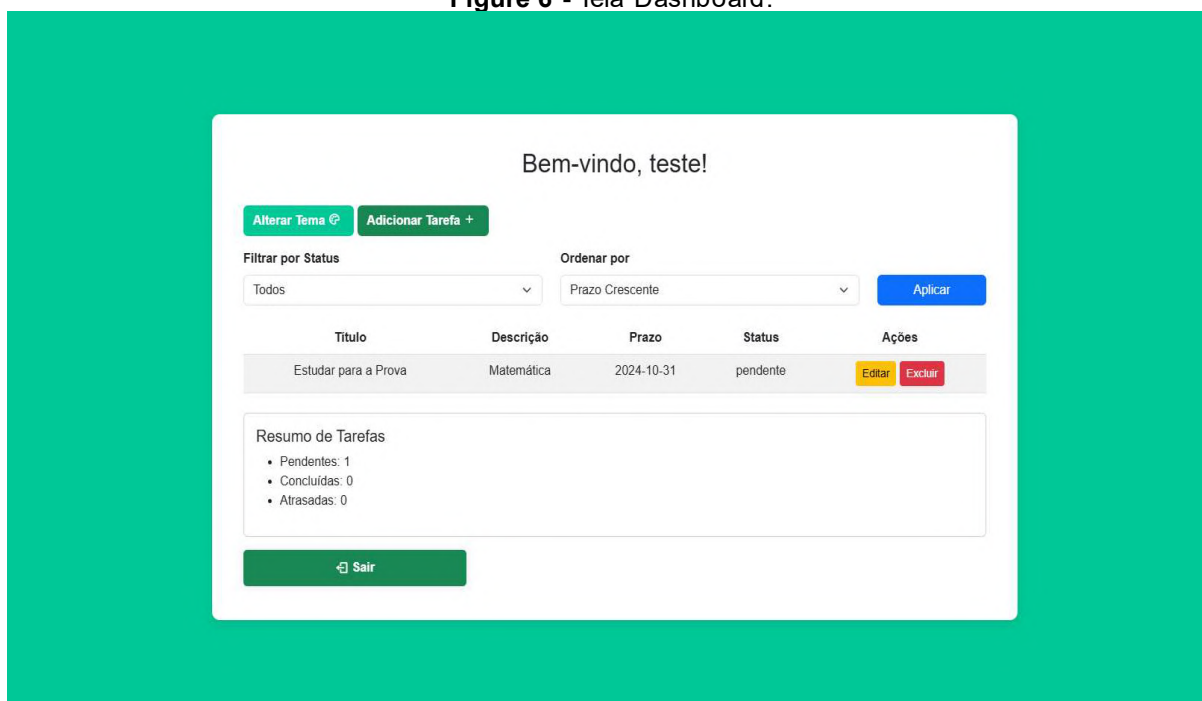


**Fonte:** De autoria própria (2024).

#### 4.4. Dashboard

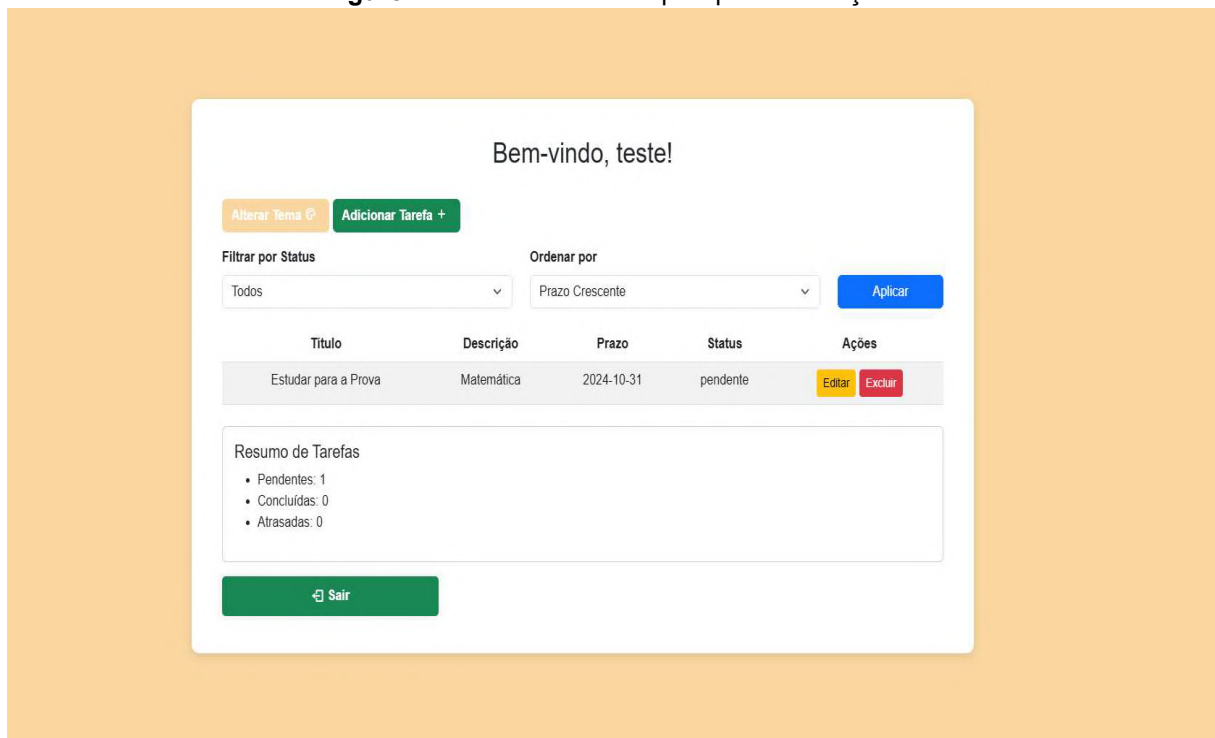
A tela Dashboard é o coração do nosso sistema de gerenciamento de tarefas, oferecendo uma visão clara e organizada de todas as atividades do usuário. Neste painel, os usuários podem incluir e visualizar suas tarefas de forma dinâmica, utilizando filtros e opções de ordenação para verificarem se suas atividades estão pendentes, atrasadas ou concluídas e encontrar rapidamente o que precisam.

**Figure 6** - Tela Dashboard.



**Fonte:** De autoria própria (2024).

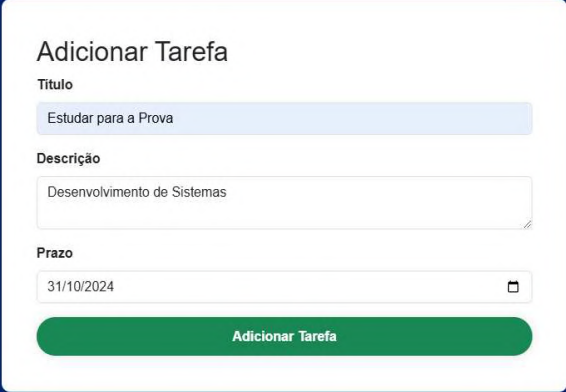
Ao clicar no botão "Alternar Tema", o usuário poderá personalizar sua tela com uma seleção de cores pré-definidas.

**Figure 7** - Tela Dashboard após personalização.

**Fonte:** De autoria própria (2024).

#### 4.5. Adicionar Tarefas

A tela de adicionar tarefas é uma funcionalidade essencial do nosso sistema de gerenciamento, integrada ao Dashboard. Aqui, os usuários podem inserir novas atividades de maneira intuitiva, garantindo que nada fique esquecido. Essa interface dinâmica facilita a organização das atividades, permitindo que os usuários mantenham o foco no que realmente importa e gerenciem seu tempo de forma eficiente.

**Figure 8** - Tela adicionar tarefas.

**Adicionar Tarefa**

**Título**  
Estudar para a Prova

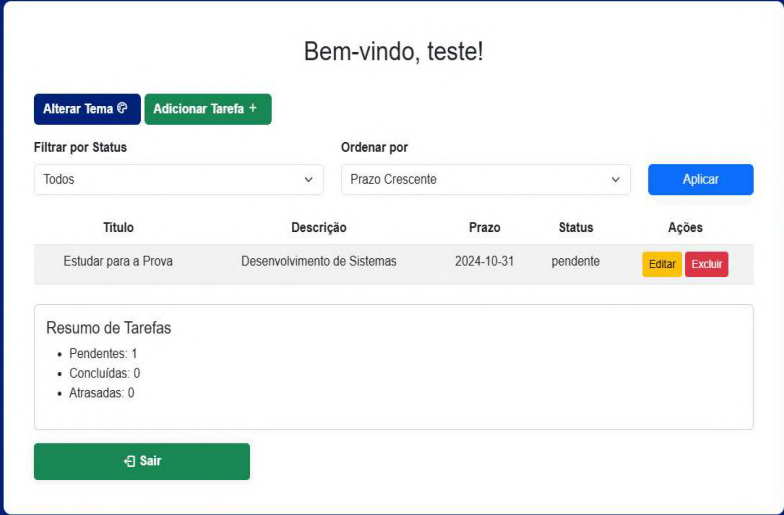
**Descrição**  
Desenvolvimento de Sistemas

**Prazo**  
31/10/2024

**Adicionar Tarefa**

**Fonte:** De autoria própria (2024).

Dashboard após a adição da tarefa:

**Figure 9** - Tela dashboard após adição da tarefa.

Bem-vindo, teste!

**Alterar Tema** **Adicionar Tarefa +**

Filtrar por Status: Todos | Ordenar por: Prazo Crescente | **Aplicar**

| Título               | Descrição                   | Prazo      | Status   | Ações                        |
|----------------------|-----------------------------|------------|----------|------------------------------|
| Estudar para a Prova | Desenvolvimento de Sistemas | 2024-10-31 | pendente | <b>Editar</b> <b>Excluir</b> |

**Resumo de Tarefas**

- Pendentes: 1
- Concluídas: 0
- Atrasadas: 0

**Sair**

**Fonte:** De autoria própria (2024).

#### 4.6. Editar Tarefa

A tela editar permite aos usuários atualizarem e modificar suas tarefas de forma rápida e eficiente. Ao clicar nesse botão, os usuários podem ajustar detalhes como título, descrição, prazo e status, garantindo que as informações estejam

sempre atualizadas. Essa funcionalidade proporciona flexibilidade na gestão de tarefas, permitindo que os usuários adaptem suas listas de atividades conforme as mudanças nas demandas do dia a dia.

**Figure 10** - Tela editar.



**Editar Tarefa**

**Título**  
Estudar para a Prova

**Descrição**  
Desenvolvimento de Sistemas

**Prazo**  
31/10/2024

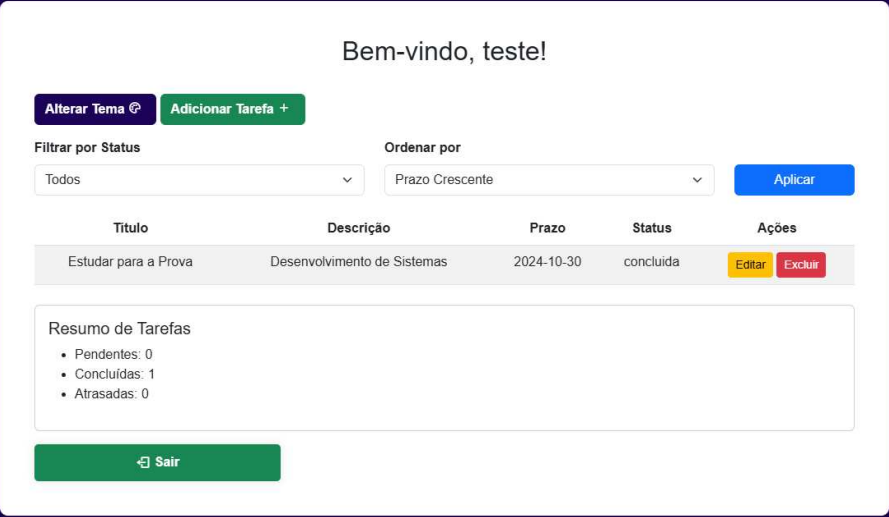
**Status**  
Concluída

Atualizar Tarefa

**Fonte:** De autoria própria (2024).

Dashboard após edição:

**Figure 11** - Tela Dashboard após edição.



Bem-vindo, teste!

Alterar Tema Adicionar Tarefa +

Filtrar por Status Ordenar por

Todos Prazo Crescente Aplicar

| Título               | Descrição                   | Prazo      | Status    | Ações          |
|----------------------|-----------------------------|------------|-----------|----------------|
| Estudar para a Prova | Desenvolvimento de Sistemas | 2024-10-30 | concluída | Editar Excluir |

Resumo de Tarefas

- Pendentes: 0
- Concluídas: 1
- Atrasadas: 0

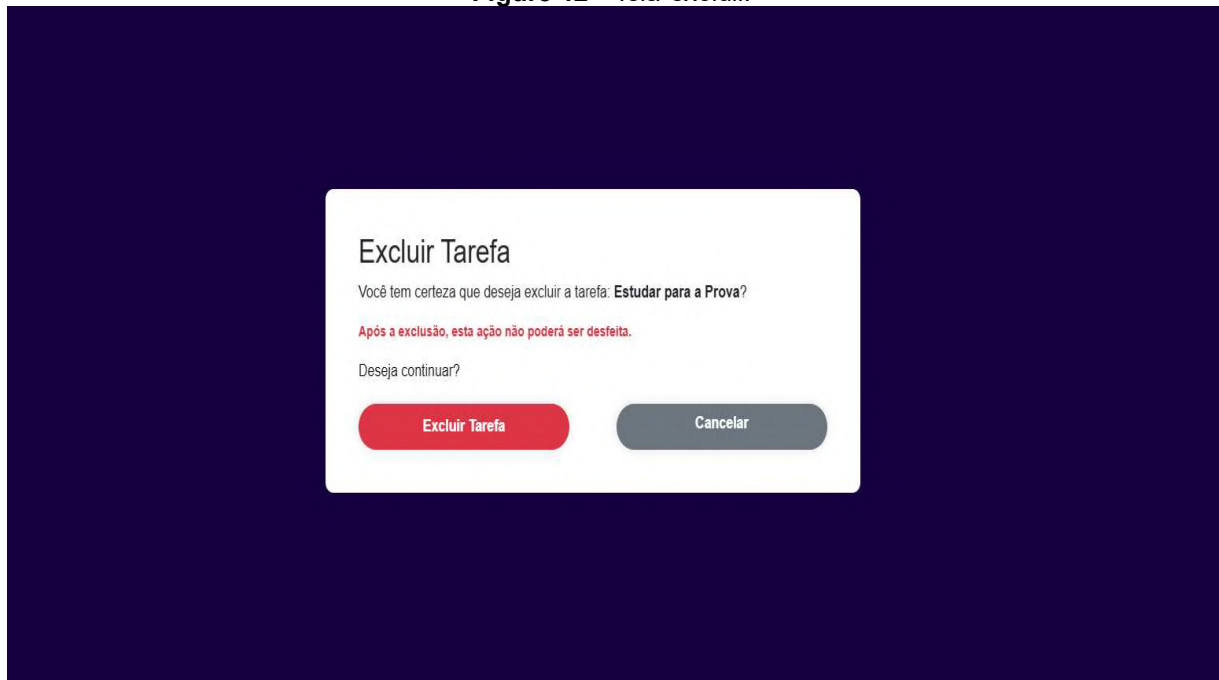
Sair

**Fonte:** De autoria própria (2024).

#### 4.7. Excluir

A tela excluir permite os usuários removam tarefas que não são mais relevantes ou necessárias. Ao clicar nesse botão, os usuários podem eliminar as atividades concluídas ou desatualizadas, após a confirmação da perguntando de segurança exibida pelo sistema, ajudando-os a manterem sua lista de tarefas organizada e focada.

**Figure 12** - Tela excluir.



**Fonte:** De autoria própria (2024).

## 5. CONSIDERAÇÕES FINAIS

O Sistema de Gerenciamento de Tarefas foi desenvolvido com o objetivo de oferecer uma solução prática e eficaz para o gerenciamento de atividades diárias, ajudando os usuários a organizarem melhor suas tarefas e otimizarem o uso do tempo. Por meio de uma interface amigável e funcionalidades intuitivas, o sistema visa facilitar a criação, edição, exclusão e priorização de tarefas, oferecendo recursos que aumentam a produtividade e simplificam o planejamento.

A aplicação foi desenvolvida utilizando uma combinação de tecnologias e linguagens robustas tanto para o back-end, como a linguagem PHP e MySQL para o gerenciamento de dados e utilizando linguagens front-end, como HTML, CSS e JavaScript, proporcionando um sistema seguro, rápido e acessível em diversos dispositivos. O uso da Metodologia Ágil Scrum possibilitou a adaptação do projeto às mudanças nos requisitos e garantiu a entrega contínua de funcionalidades, alinhadas às necessidades dos usuários. Ao oferecer uma solução digital para gerenciamento de tarefas, o sistema atende a uma demanda crescente por ferramentas de produtividade e gestão do tempo, tanto para uso pessoal quanto profissional.

Durante o desenvolvimento do projeto, alguns desafios foram enfrentados, como a necessidade de garantir a segurança dos dados e a integração com diversas funcionalidades, incluindo notificações e histórico de atividades. Esses desafios proporcionaram um valioso aprendizado para a equipe, que aprimorou suas habilidades em desenvolvimento web e técnicas de gerenciamento de projetos ágeis.

O projeto apresenta um grande potencial para expansão. Entre as melhorias futuras, destacam-se a adição de novas funcionalidades, como integração com aplicativos de calendário, suporte para tarefas recorrentes, controle de permissões por função, e a possibilidade de anexar arquivos. Essas melhorias visam oferecer uma experiência mais completa e personalizada, mantendo o sistema em constante evolução.

O desenvolvimento do Sistema de Gerenciamento de Tarefas atingiu os objetivos propostos, fornecendo uma ferramenta útil e acessível para a gestão de atividades diárias. A solução entrega uma interface simples e funcionalidades

práticas, contribuindo para uma melhor organização e produtividade dos usuários.

Com a implementação de futuras melhorias e a coleta contínua de feedback dos usuários, o sistema poderá evoluir e se consolidar como uma referência em ferramentas de gerenciamento de tarefas.

Com isso, esperamos que o sistema contribua significativamente para o aumento da produtividade dos usuários, reduzindo o tempo gasto com o gerenciamento de tarefas e otimizando a utilização de recursos.

## REFERÊNCIAS

DE MELO, MARCOS. **PHP – Criando Páginas Dinâmicas**. Disponível em:

<[https://marcosdemelo.com/aulas/files/Desenvolvimento%20Back-](https://marcosdemelo.com/aulas/files/Desenvolvimento%20Back-End/Apostilas/Livro%20PHP%20-%20P%C3%A1ginas%20Din%C3%A2micas.pdf)

[End/Apostilas/Livro%20PHP%20-%20P%C3%A1ginas%20Din%C3%A2micas.pdf](https://marcosdemelo.com/aulas/files/Desenvolvimento%20Back-End/Apostilas/Livro%20PHP%20-%20P%C3%A1ginas%20Din%C3%A2micas.pdf)>

Acesso em: 22/09/2024.

DEVMEDIA. **HTML básico - códigos HTML**. Disponível em:

<<https://www.devmedia.com.br/html-basico-codigos-html/16596>>. Acesso em:

18/09/2024.

DEVMEDIA. **Primeiros Passos no Mysql**. Disponível em:

<<https://www.devmedia.com.br/primeiros-passos-no-mysql/28438>>. Acesso em:

18/09/2024.

IFFR. **Portal do Instituto Federal Farroupilhas**. Disponível em:

<[https://www.iffarroupilha.edu.br/component/k2/attachments/download/5702/87f24f4c](https://www.iffarroupilha.edu.br/component/k2/attachments/download/5702/87f24f4c8f7ed3238ef6f4a71b010a14)

[8f7ed3238ef6f4a71b010a14](https://www.iffarroupilha.edu.br/component/k2/attachments/download/5702/87f24f4c8f7ed3238ef6f4a71b010a14)>. Acesso em: 06/09/2024.

MDN WEB DOCS. **JavaScript**. Disponível em: <[https://developer.mozilla.org/pt-](https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript)

[BR/docs/Learn/JavaScript](https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript)>. Acesso em: 06/09/2024.

SKLAR, DAVID. **Aprendendo PHP**. Novatec. Disponível em:

<[https://books.google.com.br/books?id=9bfeDAAQBAJ&printsec=frontcover&hl=pt-](https://books.google.com.br/books?id=9bfeDAAQBAJ&printsec=frontcover&hl=pt-BR#v=onepage&q&f=false)

[BR#v=onepage&q&f=false](https://books.google.com.br/books?id=9bfeDAAQBAJ&printsec=frontcover&hl=pt-BR#v=onepage&q&f=false)>. Acesso em 16/09/2024.

UNICESUMAR. **Introdução ao Gerenciamento Ágil de Software**. Disponível em:

<[https://www.unicesumar.edu.br/wp-content/uploads/degustacao/ebook/ebook-](https://www.unicesumar.edu.br/wp-content/uploads/degustacao/ebook/ebook-material-didatico-engenharia-de-software.pdf)

[material-didatico-engenharia-de-software.pdf](https://www.unicesumar.edu.br/wp-content/uploads/degustacao/ebook/ebook-material-didatico-engenharia-de-software.pdf)>. Acesso em: 06/09/2024.

YOUTUBE. **Curso completo e atual de HTML5 e CSS3 - Módulo 1 de 5**.

Disponível em:< [https://www.youtube.com/playlist?list=PLHz\\_AreHm4dkZ9-](https://www.youtube.com/playlist?list=PLHz_AreHm4dkZ9-atkcmcBaMZdmLHft8n)

[atkcmcBaMZdmLHft8n](https://www.youtube.com/playlist?list=PLHz_AreHm4dkZ9-atkcmcBaMZdmLHft8n)>. Acesso em: 20/10/2024.

YOUTUBE. **Curso de PHP Moderno – Módulo 1 – Curso em Vídeo**. Disponível

em:<[https://www.youtube.com/watch?v=TfsO0BGvGn0&list=PLHz\\_AreHm4dIFPrCX](https://www.youtube.com/watch?v=TfsO0BGvGn0&list=PLHz_AreHm4dIFPrCX)

[Cmd5g92860x\\_Pbr\\_](https://www.youtube.com/watch?v=TfsO0BGvGn0&list=PLHz_AreHm4dIFPrCX)>. Acesso em: 20/10/2024.

YOUTUBE. **Login Form in HTML & CSS**. Disponível em:<

<https://www.youtube.com/watch?v=hlwIM4a5rxg>>. Acesso em: 26/10/2024.

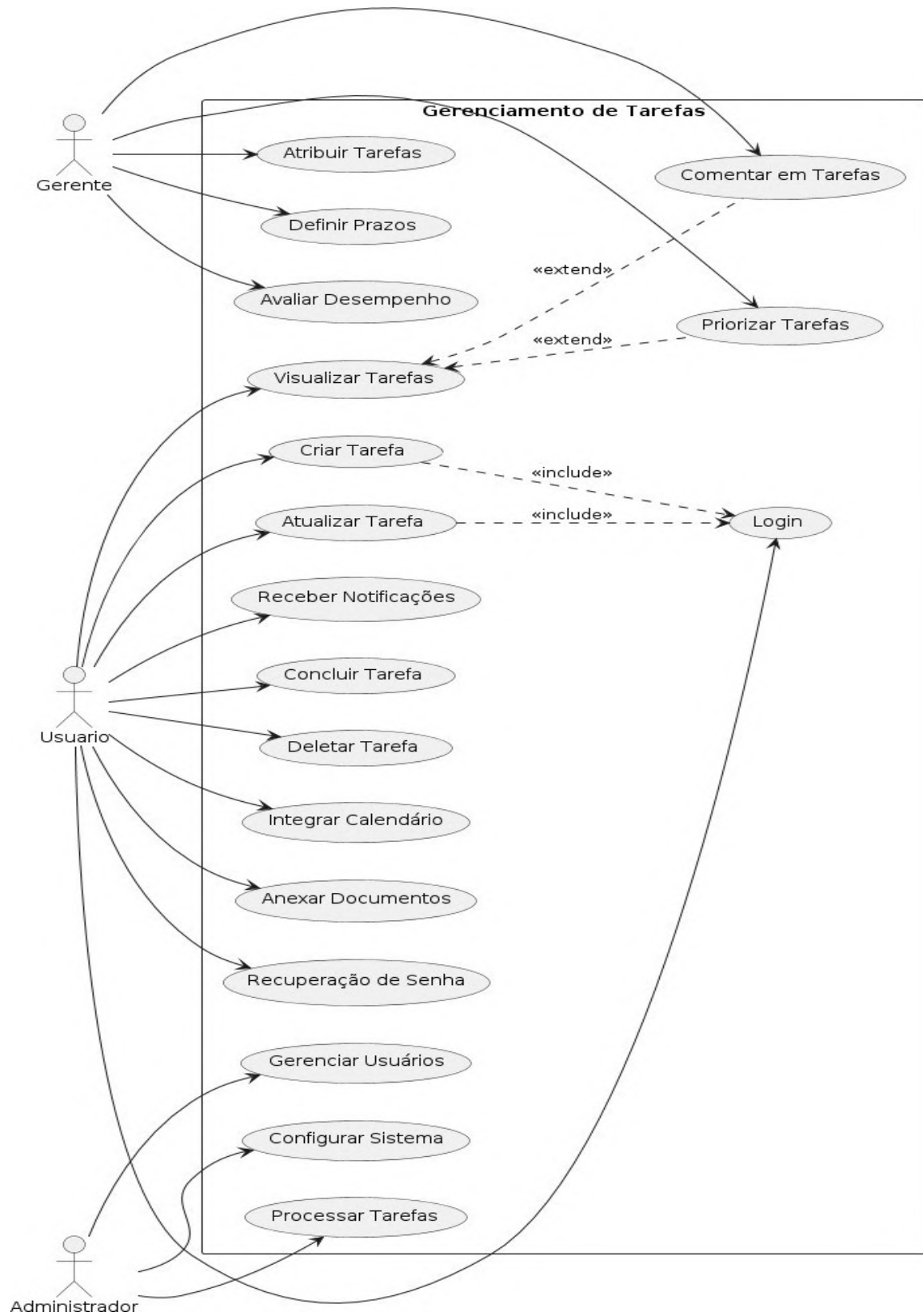
WK TECHNOLOGY. **Passo a passo para desenvolver softwares do zero.**

Disponível em: <<https://wktechnology.com.br/desenvolver-softwares-do-zero/>>.

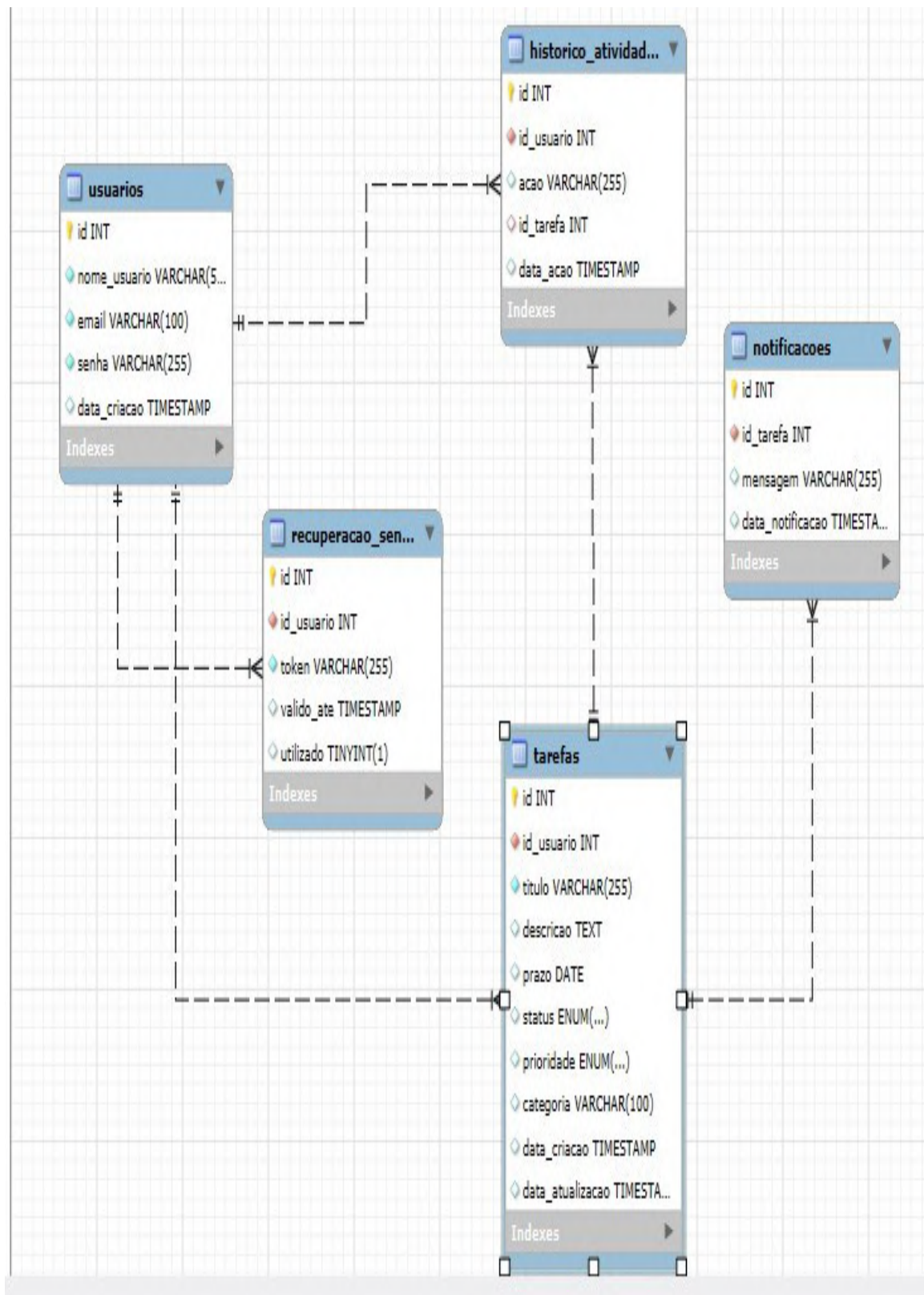
Acesso em: 26/09/2024.

## ANEXOS

## ANEXO I – DIAGRAMA DE USO



## ANEXO II – DIAGRAMA DE CLASSE



## ANEXO III – TRELLO

