­FACULDADE DE TECNOLOGIA DE SÃO CAETANO DO SUL – ANTONIO RUSSO

Bruna Silvestre do Nascimento

Gabriel Edivaldo Oliveira Bezerra Da Silva

Luana Evelyn Lopes Nascimento

Samantha Nozomi Morimoto

**SMARTTASKS – GERENCIADOR DE PROJETOS**

São Caetano do Sul

2024

Bruna Silvestre do Nascimento

Gabriel Edivaldo Oliveira Bezerra Da Silva

Luana Evelyn Lopes Nascimento

Samantha Nozomi Morimoto

**SMARTTASKS – GERENCIADOR DE PROJETOS**

Trabalho de conclusão de curso submetido como requisito parcial para conclusão do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Faculdade de Tecnologia de São Caetano do Sul – Antônio Russo.

Orientador: Prof. Dr. Rodrigo Amorim Motta Carvalho

São Caetano do Sul

2024

Bruna Silvestre do Nascimento

Gabriel Edivaldo Oliveira Bezerra Da Silva

Luana Evelyn Lopes Nascimento

Samantha Nozomi Morimoto

**SMARTTASKS – GERENCIADOR DE PROJETOS**

Banca examinadora:

|  |
| --- |
| Prof. Dr. Rodrigo Amorim Motta Carvalho  Presidente da Banca |
| Prof. Xxxxxxxxxxxxxxxxxxxxxxx  Professor convidado |
| Prof. Xxxxxxxxxxxxxxxxxxxxxxx  Professor convidado |

São Caetano do Sul

2024

RESUMO

Este trabalho de conclusão de curso tem como objetivo desenvolver o SmartTasks, um website gerenciador de tarefas voltado para a execução de projetos. O sistema oferece recursos inteligentes e intuitivos para a organização e acompanhamento das tarefas diárias, auxiliando os usuários a enfrentarem os desafios do ambiente de trabalho, como alta carga de tarefas, demandas constantes e prazos apertados. Essas ferramentas são essenciais para melhorar a gestão do tempo e das atividades. Para o desenvolvimento do software, foram utilizadas as linguagens de programação PHP e JavaScript no back-end, HTML e CSS no front-end, e o banco de dados MySQL para armazenamento. Além disso, foi adotada a metodologia do PMI (Project Management Institute) para o gerenciamento de projetos, o que foi fundamental no planejamento do SmartTasks. O resultado é um website gratuito com usabilidade explícita, projetado para facilitar a vida dos usuários.

**Palavras-chave**: Gerenciador de tarefas, usuários, PMI

ABSTRACT

The aim of this final project is to develop SmartTasks, a task management website for project execution. The system offers intelligent and intuitive resources for organizing and monitoring daily tasks, helping users to face the challenges of the workplace, such as high task loads, constant demands and tight deadlines. These tools are essential for improving time and activity management. To develop the software, the programming languages PHP and JavaScript were used for the back-end, HTML and CSS for the front-end, and the MySQL database for storage. In addition, the PMI (Project Management Institute) methodology for project management was adopted, which was fundamental in the planning of SmartTasks. The result is a free website with explicit usability, designed to make life easier for users.

**Keywords**: Task Manager, users, PMI

LISTA DE ILUSTRAÇÕES

[Figura 1 – Diagrama de caso de uso do sistema 16](#_4f1mdlm)

[Figura 2 – Interface da tela inicial 17](#_2u6wntf)

Figura 3 – Interface da tela de login

Figura 4 – Interface das notificações

Figura 5 – Interface de perfil

Figura 6 – Interface de progresso

Figura 7 – Interface de acesso ao projeto

Figura 8 – Interface de criar projeto

Figura 9 – Interface de criar tarefas

Figura 10 – Interface de cadastro

Figura 11 – Interface de Login

Figura 12 – Arquitetura Cliente – Servidor

Figura 13 – Cenário de cadastro dos usuários

Figura 14 – Cenário de login dos usuários

Figura 15 – Cenário de cadastro de projeto, consulta, alteração e exclusão

Figura 16 – Cenário de cadastro de tarefas, consulta, alteração e exclusão

Figura 17 – Cenário de estimativa de conclusão de projeto

Figura 18 – Cenário do gestor em cadastrar projeto, consultar, alterar e excluir

Figura 19 – Cenário de admin: adicionar usuário

Figura 20 – Cenário de admin: alterar ou excluir usuário

Figura 21 – Modelo de dados

Figura 22 – Interface de estimativa de prazo iniciando

Figura 23 – Interface de estimativa de prazo com mudança de status

Figura 24 – Interface de estimativa de prazo recalculando sem mudar status

Figura 25 ­– Interface de estimativa de prazo com novo prazo

LISTA DE TABELAS

[Tabela 1 – Especificações de requisitos funcionais 17](#_19c6y18)

[Tabela 2 – Especificações de requisitos não funcionais 18](#_3tbugp1)

Tabela 3 – Especificações de regra de negócio

Tabela 4 – Descrição dos atores

Tabela 5 – Descrição de casos de uso

LISTA DE ABREVIATURAS E SIGLAS

ABNT Associação Brasileira de Normas Técnicas

IBGE Instituto Brasileiro de Geografia e Estatística

PMI Project Management Institute

SGBD Sistema de Gerenciamento de Banco de Dados

HTML HyperText Markup Language

CSS Cascading Style Sheets

SQL Structured Query Language

PHP Hypertext Preprocessor

SUMÁRIO

[**INTRODUÇÃO** 15](#_30j0zll)

[Tema 15](#_1fob9te)

[Problema/público alvo/Contexto do sistema 15](#_1fob9te)

[Objetivos gerais e específicos 15](#_2et92p0)

[Justificativa 15](#_tyjcwt)

[Procedimentos metodológicos 15](#_3dy6vkm)

[Apresentação dos capítulos 15](#_1t3h5sf)

[**2 TECNOLOGIAS** 16](#_4d34og8)

[2.1 PHP 16](#_2s8eyo1)

[2.2 jQuery 16](#_17dp8vu)

[2.3 SQL 16](#_3rdcrjn)

[2.4 Bootstrap 16](#_3rdcrjn)

[2.5 HTML 16](#_3rdcrjn)

[2.6 CSS 16](#_3rdcrjn)

[2.6 XAMPP 16](#_3rdcrjn)

[2.6 Visual Studio Code 16](#_3rdcrjn)

[**3 REQUISITOS** 17](#_26in1rg)

[3.1 Requisitos funcionais 17](#_lnxbz9)

[3.2 Requisitos não funcionais 17](#_35nkun2)

[3.3 Regras de negócio 17](#_1ksv4uv)

[3.5 Documentação dos casos de uso 17](#_44sinio)

[3.5.1 Atores 17](#_2jxsxqh)

[3.5.2 Diagrama de casos de uso 17](#_z337ya)

[3.5.2 Descrição textual dos casos de uso 17](#_3j2qqm3)

[3.5.3 Associação casos de uso x requisitos 17](#_1y810tw)

[3.6 Protótipo das interfaces do sistema 17](#_4i7ojhp)

[**4 PROJETO DO SISTEMA** 18](#_2xcytpi)

[4.1 Arquitetura do software 18](#_1ci93xb)

[4.2 Diagrama e especificação das classes 18](#_3whwml4)

[4.3 Diagrama de sequencia 18](#_2bn6wsx)

[4.4 Persistência de dados 18](#_qsh70q)

[4.4.1 Características do SGDB 18](#_3as4poj)

[4.4.2 Modelo de dados 18](#_1pxezwc)

[- Modelos lógico/Conceitual para bancos relacionais 18](#_49x2ik5)

[- Esquema de armazenamento para banco de dados NoSQL 18](#_2p2csry)

[4.5 Plano de testes 18](#_147n2zr)

[**5 RESULTADOS** 19](#_3o7alnk)

[*\*\* Executar o sistema, capturar as telas e apresenta-las. O objetivo é demonstrar que o sistema está atendendo seus propósitos.* 19](#_23ckvvd)

[5.1 Modulo a 19](#_ihv636)

[5.2 Modulo b 19](#_32hioqz)

[5.3 Modulo n 19](#_1hmsyys)

[5.4 Discussão 19](#_41mghml)

[**6 CONSIDERAÇÕES FINAIS** 20](#_2grqrue)

[**REFERÊNCIAS** 21](#_vx1227)

[APÊNDICE A – Aqui vai o título do apêndice 22](#_3fwokq0)

[ANEXO A – Aqui vai o título do anexo 23](#_1v1yuxt)

1. **INTRODUÇÃO**

A transformação digital tem sido um tema importante para as empresas quando se trata de inovação e mudança de pensamento. No contexto atual, a tecnologia está sendo muito utilizada para transformar processos manuais em processos digitais, a fim de atingir níveis de escala, otimização do tempo, melhor distribuição de recursos, e assim, tendo atualizações constantes, os softwares cada vez mais são utilizados nessa era “digital”, principalmente no ambiente empresarial. Os usos dessas tecnologias vêm sendo visíveis para atender as demandas, tornando eficaz e indispensável dentro da empresa. Desta forma, surge uma questão “Qual é a importância e o benefício de usar um software para gerenciamento de projetos?” e respondendo-o, o software pode facilitar a resolução de problemas, automatizar tarefas repetitivas, administrar o tempo, controlar fluxo financeiro e outros recursos.

Nos próximos capítulos abordaremos a construção de um software para gerenciamento de projetos, dentre alguns conhecidos como Monday e Trello, o sistema SmartTasks inova em estimar o tempo de realização do projeto dividindo-o em marcos.

**Tema**

O gerenciamento de projetos é uma disciplina essencial em diversas áreas do conhecimento, incluindo negócios, eventos e tecnologia. Ele ajuda a garantir que os projetos sejam concluídos com sucesso, dentro do escopo, prazo, custo e qualidade esperados. Empresas e organizações utilizam técnicas de gerenciamento de projetos para aumentar a eficiência, reduzir riscos e alcançar objetivos estratégicos.

Na etapa de planejamento define-se os objetivos, escopo, cronograma, recursos, riscos e orçamento do projeto. Seguindo para etapa de execução onde implementa-se as atividades conforme o plano estabelecido. A etapa seguinte é o monitoramento e controle, acompanha-se o progresso do projeto, identifica-se desvios e toma-se ações corretivas. Por fim, a última etapa é o encerramento, finaliza-se o projeto, entrega-se os resultados e avalia-se o desempenho.

Cada etapa deve ser gerenciada com atenção para garantir que o resultado atenda às expectativas do cliente e seja lançado com sucesso. Se alguma das etapas não for bem executada o projeto corre o risco de insucesso. Se o gestor do projeto não organizar, monitorar, controlar e acompanhar o projeto os prazos podem ser perdidos, a capacidade de pessoas para executar as tarefas pode ser insuficiente dado o prazo estabelecido e o custo pode ser afetado.

A aplicação web SmartTasks surge para contribuir com as empresas e os gestores de projetos na organização de seus projetos, criação de tarefas e responsabilização, definição de prazo e estimativa de tempo baseado em marcos de 25%, 50%,75% e 100% do progresso do projeto. Ainda conta com uma página dedicada à *business intelligence*, uma área que transforma dados em conhecimento útil e auxilia empresas e organizações a serem mais ágeis em suas tomadas de decisões. A simplicidade da interface e a clareza dos termos usados para indicar os campos buscam guiar o gerente e a equipe a preencherem corretamente com os dados necessários.

**Problema/público-alvo/Contexto do sistema**

Uma empresa de desenvolvimento de software tem enfrentado dificuldades para entregar os projetos dentro dos prazos acordados. Os projetos frequentemente excedem o tempo estimado e os custos aumentam devido a falta de clareza e objetividade do escopo, capacidade de pessoas insuficiente para executar as tarefas e problemas de comunicação. Refletindo em estresse para os colaboradores e perda de receita para a empresa.

O público-alvo pode ser descrito como:

- Gestores de projetos: Profissionais responsáveis por liderar equipes de desenvolvimento, definir metas, planejar recursos e garantir a entrega bem-sucedida de projetos dentro do escopo, prazo e do orçamento.

- Desenvolvedores e equipes de projeto: Colaboradores que trabalham diretamente na execução das tarefas e na implementação dos projetos. Eles precisam de uma ferramenta eficiente para gerenciar suas atividades diárias, colaborar com colegas de equipe e acompanhar o progresso do projeto.

- Empresas e organizações de outros segmentos: A aplicação SmartTasks pode ser útil para empresas de tecnologia, consultorias, agências de marketing, departamentos de TI e outras organizações que precisam gerenciar múltiplos projetos simultaneamente.

- Clientes e stakeholders (partes interessadas): São clientes indiretos da ferramenta, os clientes e stakeholders são parte interessada nos projetos e desejam transparência, comunicação eficaz e entregas pontuais.

O objetivo é desenvolver uma aplicação web gerenciadora de projetos que possua a capacidade de gerenciar o desenvolvimento de projetos, tarefa e visualização de dados em forma de dashboard. O sistema irá estimar os prazos necessários para completar os projetos, sendo representado por porcentagem de 25, 50, 75 e 100% para representar o progresso do projeto, desta forma, organizando as tarefas e contará com responsáveis para administrar os projetos em que estão envolvidos.

**Justificativa**

A criação de um sistema de gerenciamento de projetos pode contribuir com a organização das etapas, tarefas, prazos e recursos envolvidos em um projeto de forma estruturada. Isso evita confusões, retrabalho e perda de informações importantes.

É possível acompanhar o progresso de cada tarefa, identificar possíveis atrasos ou gargalos e tomar medidas corretivas rapidamente. Isso garante que o projeto avance de acordo com o cronograma estabelecido.

Ao estabelecer processos claros e padrões de trabalho, um sistema de gerenciamento de projetos contribui para a melhoria contínua da qualidade das entregas e do desempenho da equipe.

A criação de um sistema de gerenciamento de projetos pode otimizar processos, aumentar a eficiência operacional, reduzir custos, garantir a qualidade das entregas e alcançar os objetivos estratégicos da empresa.

A implementação do projeto SmartTasks visa contribuir com o trabalho de gestores de projetos, buscando proporcionar uma experiência de organização, planejamento, acompanhamento do projeto e visualização dos dados como informação possibilitando a entrega dos projetos dentro do prazo acordado.

**Metodologia**

Para o plano de teste, testamos a funcionalidade Estimativa de prazo de um projeto, a principal função de destaque no sistema. O objetivo da estimativa é medir o progresso do projeto em relação ao tempo determinado pelo gestor no cadastro do projeto.

A metodologia usada divide a quantidade total de tarefas e a duração do projeto em quatro marcos de 25%. Para as tarefas considera-se sempre um número inteiro maior caso a divisão retorne um número quebrado e para os dias considera-se um dia a menos da data de entrega, pois essa data precisa ser reservada para a entrega acordada do projeto. Para saber se o projeto vai atrasar ou não, a função compara a quantidade de tarefas e a data do marco com o dia que o cálculo é acionado e o que já foi realizado até aquele momento.

Tabela 01: Plano de testes

|  |  |
| --- | --- |
| Plano de testes | |
| Cenário | Descrição |
| Situação 1 | Projeto iniciando |
| Situação 2 | O projeto extrapolou o prazo e os status das tarefas já foram modificados |
| Situação 3 | O prazo é alterado e a estimativa recalculada, sem alterar os status das tarefas |
| Situação 4 | Prazo novo e alteração dos status das tarefas |

FONTE: Elaborada pelos autores (2024)

**Procedimentos metodológicos**

1. Elaboração de requisitos funcionais e não funcionais
2. Definição da arquitetura de informação, dos fluxos de navegação e da aparência visual com base nos requisitos levantados.
3. Implementação das funcionalidades e características do sistema
4. Elaboração e execução de planos de teste
   1. Apresentação dos capítulos

### Tecnologias

Neste capítulo serão apresentadas as tecnologias que foram utilizadas na construção base do sistema. A escolha das ferramentas e tecnologias certas desempenham um papel fundamental na construção eficiente da aplicação. São abordadas as linguagens utilizadas no desenvolvimento de software, o banco de dados, as ferramentas de versionamento, e as bibliotecas usadas para agilizar no processo de construção. Além disso, serão descritas as razões por trás da escolha de tais tecnologias que foram selecionadas com base em sua adequação aos requisitos do projeto e às necessidades dos usuários.

### Requisitos

Este capítulo aborda os requisitos do sistema, enfatizando as necessidades e funcionalidades essenciais identificadas durante a análise e levantamento de requisitos. A clareza na descrição dos requisitos é fundamental para a construção de um sistema que atenda às expectativas do usuário e suas demandas. Inicialmente serão apresentados os requisitos funcionais do sistema, que descrevem as principais operações a serem realizadas, e por fim, serão discutidos os requisitos não funcionais, que definem atributos de qualidade do software.

**Projeto do sistema**

Este capítulo aborda de forma geral, a base da implementação eficaz das funcionalidades propostas. Será descrita a arquitetura geral do sistema, descrevendo os principais componentes e a relação entre eles. Em seguida serão apresentados também as partes de front-end e back-end, o banco de dados e as interfaces de integração. Cada componente será descrito em termos de responsabilidades, interfaces e tecnologias utilizadas.

**Resultados**

Será destacado neste capítulo os resultados positivos alcançados pela implementação bem-sucedida do sistema, evidenciando suas contribuições e funcionalidades. Será apresentado a eficácia do software e como ele pode ser usado para gerenciar melhor projetos e tarefas dentro de organizações, obtendo uma melhor controle e organização, além do aumento da produtividade.

**Considerações finais**

Neste capítulo, serão apresentadas as conclusões sobre o desenvolvimento e a implementação do sistema. Serão analisados os objetivos alcançados e como o software contribuiu para a melhoria da gestão de projetos em organizações. Além disso serão sugeridos caminhos para trabalhos futuros e melhorias contínuas que se adequem a necessidades do usuário, e as tecnologias do momento.

**2 TECNOLOGIAS**

**PHP (Hypertext Preprocessor)**

PHP é uma linguagem de script do lado do servidor amplamente utilizada para desenvolvimento web. É especialmente adequada para criar páginas dinâmicas e interativas, foi escolhido devido à sua popularidade e suporte robusto para interações com bancos de dados, facilitando o desenvolvimento de aplicações web dinâmicas. Além de ser uma das linguagens ensinadas na disciplina de scripts da FATEC São Caetano do Sul.

**jQuery**

jQuery é uma biblioteca de JavaScript rápida, pequena e rica em recursos, que simplifica a manipulação do HTML, gerenciamento de eventos e interações AJAX, escolhido por sua facilidade de uso e capacidade de simplificar tarefas comuns de front-end, como validação de formulários, manipulação de DOM e requisições AJAX.

**SQL (Structured Query Language)**

SQL é uma linguagem de consulta padrão para manipular e gerenciar bancos de dados relacionais. Ela permite realizar operações como inserção, atualização, exclusão e recuperação de dados, é essencial para interagir com bancos de dados relacionais, permitindo armazenar e acessar dados de forma eficiente e estruturada.

**Bootstrap**

Bootstrap é um framework front-end de código aberto que facilita o desenvolvimento de interfaces web responsivas e estilizadas. Ele fornece um conjunto de componentes e estilos predefinidos, é escolhido por oferecer um rápido desenvolvimento de UI/UX, responsividade para dispositivos móveis e uma estrutura consistente de design, economizando tempo e esforço no desenvolvimento front-end.

**HTML (Hypertext Markup Language)**

HTML é a linguagem padrão para estruturar e criar conteúdo para páginas web. Ele define a estrutura básica de uma página, incluindo elementos como cabeçalho, parágrafos, listas e links, é a base fundamental de qualquer página web, permitindo a criação de estruturas e a incorporação de outros elementos como CSS e JavaScript para criar uma experiência interativa.

**CSS (Cascading Style Sheets)**

CSS é uma linguagem de estilo que define a apresentação visual de páginas web. Ela controla o layout, cores, fontes e outros aspectos visuais de elementos HTML, é essencial para a estilização e formatação de páginas web, permitindo criar layouts atraentes, responsivos e consistentes em diferentes dispositivos e navegadores.

**Xampp (Cross-Platform, Apache, MariaDB, PHP, and Perl)**

XAMPP é um pacote de software livre que contém o servidor web Apache, o sistema de gerenciamento de banco de dados MariaDB, e as linguagens de programação PHP e Perl. Ele facilita a instalação e configuração de um ambiente de desenvolvimento local para a criação e teste de websites e aplicações web, foi escolhido por ser uma solução completa e fácil de usar para o desenvolvimento local. Ele permite aos desenvolvedores configurarem rapidamente um ambiente de servidor local, simular um servidor web real e testar funcionalidades antes de implantar os projetos em um servidor de produção.

**Visual Studio Code**

Visual Studio Code é um editor de código-fonte desenvolvido pela Microsoft, que é leve, mas poderoso. Ele suporta uma ampla gama de linguagens de programação e oferece muitos recursos. Foi escolhido por sua facilidade, eficiência de uso. Ele proporciona um ambiente de desenvolvimento prático para a construção de um sistema. Além disso, seu suporte a extensões permite adaptar o editor às necessidades específicas do projeto, facilitando o desenvolvimento.

**Ajax**

Ajax (Asynchronous JavaScript and XML) é uma técnica de programação web que permite realizar requisições assíncronas ao servidor sem a necessidade de recarregar a página inteira. O principal motivo para usar Ajax é criar interfaces web mais dinâmicas e interativas. Com o Ajax, é possível carregar novos dados do servidor, enviar informações para o servidor e atualizar partes específicas da página em tempo real, sem a necessidade de recarregar toda a página.

**Bcrypt**

Bcrypt é um algoritmo de hash utilizado para criptografia de senhas. Ele é projetado para ser lento e resistente a ataques de força bruta,utiliza uma combinação de salting (adicionando um valor aleatório único a cada senha antes de aplicar o hash) e repetições de cálculos para gerar um hash seguro que é difícil de quebrar mesmo por máquinas com alto poder computacional.

**Google Charts**

Google Charts é uma biblioteca de visualização de dados desenvolvida pelo Google. Ela permite criar gráficos e visualizações interativas diretamente em páginas da web usando JavaScript e HTML5. A biblioteca oferece uma variedade de tipos de gráficos, como gráficos de linhas, barras, pizza, área, dispersão, entre outros, além de recursos avançados como gráficos dinâmicos e interativos.

**3 REQUISITOS**

3.1 Requisitos funcionais

A tabela 1 apresenta os requisitos funcionais do sistema, descrevendo as funcionalidades que devem obter.

Tabela 2: Requisitos funcionais

|  |  |
| --- | --- |
| **Id** | **Requisitos Funcionais** |
| **RF01** | O sistema deve permitir que o usuário cadastre uma conta. |
| **RF02** | O sistema deve permitir que o usuário efetue o login. |
| **RF03** | O sistema deve permitir que o administrador crie uma conta para usuário. |
| **RF04** | O sistema deve permitir que o administrador edite a conta do usuário. |
| **RF05** | O sistema deve permitir que o administrador exclua a conta do usuário. |
| **RF06** | O sistema deve permitir que o administrador atribua o nível de acesso. |
| **RF07** | O sistema deve permitir que o administrador interrompa o acesso do usuário. |
| **RF08** | O sistema deve permitir que o usuário crie o projeto. |
| **RF09** | O sistema deve permitir que o usuário edite o projeto. |
| **RF10** | O sistema deve permitir que o usuário visualize o projeto. |
| **RF11** | O sistema deve permitir que o gestor exclua o projeto. |
| **RF12** | O sistema deve permitir que o usuário cadastre as tarefas. |
| **RF13** | O sistema deve permitir que o usuário edite as tarefas. |
| **RF14** | O sistema deve permitir que o usuário visualize as tarefas. |
| **RF15** | O sistema deve permitir que o usuário exclua as tarefas. |
| **RF16** | O sistema deve permitir que usuário atribua prioridade da tarefa. |
| **RF17** | O sistema deve permitir que o usuário defina prazo para a tarefa. |
| **RF18** | O sistema deve permitir que o usuário defina categoria de projeto. |
| **RF19** | O sistema deve permitir que o usuário adicione anexos de arquivos no projeto. |
| **RF20** | O sistema deve permitir atribuição de tarefas aos usuários. |
| **RF21** | O sistema deve permitir a visualização do progresso e status da tarefa. |
| **RF22** | O sistema deve permitir que estime os marcos do projeto e data de entrega das tarefas. |
| **RF23** | O sistema deve possibilitar a visualização de dashboard. |

FONTE: Elaborada pelos autores (2024)

Nota-se que no RF11 apenas para o gestor é permitida a exclusão do projeto.

3.2 Requisitos não funcionais

A tabela 2 apresenta os requisitos não funcionais do sistema, descrevendo como as suas funcionalidades devem ser realizadas.

Tabela 3: Requisitos não funcionais

|  |  |  |
| --- | --- | --- |
| **Código** | **Descrição:** | **Classificação** |
| **RNF01** | O sistema deve garantir a proteção dos dados do usuário, utilizando hash para armazenamento de senha e transmissão de informações, além de adotar práticas adequadas de autenticação e controle de acesso. | Segurança |
| **RNF02** | O sistema deve permitir que o usuário entre na sua conta utilizando os dados cadastrados e se autenticado. | Segurança |
| **RNF03** | O sistema deve ter tempos de resposta rápidos, com uma interface ágil e sem atrasos perceptíveis nas interações do usuário, mesmo em situações de alta carga ou grande volume de dados. | Desempenho |
| **RNF04** | A interface do sistema deve ser intuitiva e fácil de usar, com um design limpo e organizado, facilitando a navegação e o uso das funcionalidades. Deve-se considerar a acessibilidade e proporcionar uma experiência consistente em diferentes dispositivos e tamanhos de tela. | Usabilidade |
| **RNF05** | O sistema deve ser estável e não apresentar falhas frequentes, garantindo que as tarefas sejam gerenciadas corretamente e que os dados sejam consistentes. O sistema deve ser capaz de se recuperar de falhas de forma segura e não comprometer a integridade das informações. | Confiabilidade |
| **RNF06** | O sistema deve suportar um grande número de usuários simultaneamente, sem degradação significativa do desempenho. A arquitetura do sistema deve permitir a escalabilidade horizontal, possibilitando a adição de recursos conforme necessário. | Escalabilidade |
| **RNF07** | O sistema deve ser compatível com diferentes navegadores, como Chrome, Firefox e Safari, e dispositivos, incluindo desktops, laptops, tablets e smartphones, garantindo uma experiência consistente e adequada em todas as plataformas. | Compatibilidade |

FONTE: Elaborada pelos autores (2024)

3.3 Regras de negócio

A tabela 3 apresenta as regras de negócio fundamentais que direcionam o funcionamento do sistema. Estas regras são de grande importância para garantir a consistência e a eficácia das operações no sistema.

Tabela 4: Regras de negócio

|  |  |
| --- | --- |
| Id | Regra de negócio |
| **RN01** | O usuário somente deve conseguir completar o cadastro se fornecer informações válidas, como nome, e-mail e senha. |
| **RN02** | Um usuário só pode cadastrar uma única conta no sistema, usando os mesmos dados de identificação como o e-mail. |
| **RN03** | O sistema deve autenticar o usuário somente se fornecer os dados validos, permitido o acesso somente a usuário autorizados. |
| **RN04** | O usuário somente deve ser aprovado se o Admin revisar e aprovar seu cadastro. |
| **RN05** | O usuário deve conseguir criar um projeto/tarefa se fornecer informações válidas e necessárias, como nome e descrição do projeto/tarefa. |
| **RN06** | O usuário somente deve conseguir alterar um projeto se for o criador do projeto, fornecer informações válidas e se o projeto existir no sistema. |
| **RN07** | Todas as informações de projeto dos usuários devem ser armazenadas de forma segura e protegidas contra acessos não autorizados. |
| **RN08** | O Admin somente deve conseguir categorizar usuários como "gestor" ou "funcionário" durante a aprovação se o cadastro estiver pendente. |
| **RN9** | Os usuários podem verificar o progresso do projeto em porcentagem, com base no cálculo feito utilizando os dados de tarefas geral e tarefas realizadas. |
| **RN10** | Pode ser verificado pelo usuário o marco do projeto, e quantas tarefas devem ser realizadas para ser alcançado determinado marco, sendo possível ter uma visão de conclusão dentro do prazo determinado. |

FONTE: Elaborada pelos autores (2024)

|  |
| --- |
| **Usuário Final** |
| **Descrição:** Este ator é quem irá utilizar o sistema para realizar o controle e organização dos projetos e tarefas.  **Responsabilidades:**   * Cadastrar sua conta no sistema. * Realizar login. * Cadastrar os projetos. * Criar as tarefas. * Atribuir tarefas a outros usuários. * Atualizar tarefas atribuídas a ele.   **Ambiente Físico:**  Ambiente de trabalho que possua um aparelho tecnológico com acesso à internet.  **Número e Tipo:**  Não existe um número determinado destes atores, mas deve ser sempre menor ou igual à quantidade de usuários que o sistema possui.  **Frequência de uso:**  Normalmente o usuário utilizara o sistema durante sua jornada de trabalho diária. |

3.5 Documentação dos casos de uso

3.5.1 Atores

A figura abaixo apresenta o ator que possui dentro do sistema, obtendo sua identificação e funcionalidade.

3.5.2 Diagrama de casos de uso

A figura apresentada abaixo, ilustra casos de uso representando as interações entre o usuário e o administrador com as funcionalidades do sistema.

Diagrama

Descrição gerada automaticamente

3.5.2 Descrição textual dos casos de uso

|  |
| --- |
| **CSU01 - Permitir o Cadastro de Usuário** |
| **Descrição**: Este caso de uso descreve o processo de cadastro de um novo usuário no sistema.  **Ator primário**: Usuário.  **Pré-condição**: O sistema está em funcionamento.  **Pós-condição**: O novo usuário é registrado no sistema e recebe as credenciais de acesso.  **Fluxo Principal:**   1. O usuário acessa a página de cadastro do sistema. 2. O usuário preenche todos os campos do formulário. 3. O sistema valida os dados inseridos pelo usuário e verifica se o endereço de e-mail fornecido pelo usuário já está registrado no sistema. 4. Caso já exista um usuário com o mesmo endereço de e-mail, o sistema exibe uma mensagem de erro informando que o endereço de e-mail já está em uso e o fluxo retorna ao passo 2. 5. Caso o endereço de e-mail seja válido e único: O sistema cria um novo registro para o usuário no banco de dados e envia um e-mail de confirmação para o endereço fornecido pelo usuário. 6. O fluxo é concluído.   **Fluxo alternativo:**   1. Caso a senha fornecida pelo usuário não atenda aos critérios de segurança estabelecidos pelo sistema: sistema exibe uma mensagem de erro informando que a senha não é segura o suficiente.   **Fluxo de exceção:**  1. No caso de falha no envio do e-mail de confirmação após o cadastro, o sistema exibe uma mensagem de erro informando que o e-mail de confirmação não pôde ser enviado, mas o cadastro do usuário é concluído normalmente. O usuário terá a opção de solicitar um novo e-mail de confirmação posteriormente. |

|  |
| --- |
| **CSU02 - Permitir o Registro de Projetos** |
| **Descrição:** Este caso de uso descreve o processo de registro de um novo projeto no sistema de gerenciamento de projetos. O usuário fornece informações detalhadas sobre o projeto, como nome, descrição, data de início e data de conclusão prevista.  **Ator primário:** Usuário.  **Pré-condição:** O usuário está autenticado no aplicativo.  O usuário tem permissões para criar projetos.  **Pós-condição:** O novo projeto é registrado no sistema com as informações fornecidas pelo usuário. O usuário é redirecionado para a página de gerenciamento de projetos.  **Fluxo Principal:**   1. O usuário seleciona a opção de criar um projeto. 2. O sistema exibe um formulário de criação de projeto contendo os seguintes campos: título, responsável, categoria, descrição, inícios e términos estimados e reais. 3. O usuário preenche todos os campos obrigatórios do formulário. 4. O usuário confirma todas as informações. 5. O sistema valida os dados inseridos pelo usuário e salva o novo projeto no banco de dados. 6. O sistema exibe uma mensagem de sucesso informando que o projeto foi cadastrado com êxito e redireciona o usuário de volta a página de gerenciamento de projeto. 7. O fluxo é concluído.   **Fluxo alternativo:**   1. Caso o usuário deixe de preencher algum campo obrigatório do formulário: O sistema exibe uma mensagem de erro informando que todos os campos obrigatórios devem ser preenchidos. 2. O fluxo retorna ao passo 3.   **Fluxo de exceção:**   1. Caso ocorra uma falha ao salvar o novo projeto no banco de dados: o sistema exibe uma mensagem de erro informando que o projeto não pôde ser cadastrado.   O fluxo é interrompido. |

|  |
| --- |
| **CSU03 - Permitir a Criação de tarefas** |
| **Descrição:** Este caso de uso descreve o processo de criação de tarefas dentro dos projetos no sistema de gerenciamento de projetos.  **Ator primário:** Usuário.  **Pré-condição:** O usuário é o proprietário do projeto ou tem permissões de gerenciamento para o projeto.  **Pós-condição:** A tarefa é atribuída ao responsável proprietário do projeto, ou ao especificado e registrada no sistema. A lista de tarefas do projeto é atualizada com a nova tarefa. E o usuário é redirecionado de volta à página de detalhes do projeto.  **Fluxo Principal:**   1. O usuário seleciona o projeto ao qual deseja atribuir tarefa na interface do sistema. 2. O sistema exibe os detalhes do projeto, incluindo informações sobre o projeto e a lista de membros que fazem parte da equipe. 3. O usuário seleciona a opção de criar tarefa na página de detalhes do projeto 4. O sistema exibe um formulário para a criação de tarefas contendo os seguintes campos: título, responsáveis, prioridade, descrição inícios e términos estimados e reais. 5. O sistema exibe as opções de prioridade disponíveis, como alta, média e baixa. 6. O usuário confirma as informações e seleciona a opção criar tarefa. 7. O sistema exibe uma mensagem de confirmação e atualiza a lista de tarefas do projeto na página de detalhes. 8. O fluxo é concluído.   **Fluxo alternativo:**   1. Se o sistema detectar que os dados fornecidos pelo usuário estão incompletos ou   incorretos, ele exibirá uma mensagem de erro apropriada e permitirá que o usuário faça as correções necessárias.  **Fluxo de exceção:**   1. Após a atribuição da tarefa, o sistema pode notificar automaticamente o membro da equipe responsável e atualizar o status da tarefa.   O fluxo é interrompido. |

|  |
| --- |
| **CSU04 - Visualizar Tarefas** |
| **Descrição:** Este caso de uso descreve o processo pelo qual um usuário visualiza as tarefas atribuídas a ele no sistema.  **Ator primário:** Usuário.  **Pré-condição:** O usuário está autenticado no sistema.  **Pós-condição:** As tarefas atribuídas ao usuário são exibidas na interface do sistema.  **Fluxo Principal:**   1. O usuário acessa a página de tarefas no sistema. 2. O sistema recupera todas as tarefas atribuídas ao usuário a partir do banco de dados. 3. O sistema exibe a lista de tarefas na interface do usuário. 4. O fluxo é concluído.   **Fluxo alternativo:**  Caso não haja tarefas atribuídas ao usuário, o sistema informa que não há tarefas para serem exibidas. |

|  |
| --- |
| **CSU04 - Permitir que os Usuários Acompanhem o Progresso do Projeto** |
| **Descrição:** Este caso de uso permite que um usuário acompanhe o progresso de um projeto específico no sistema de gerenciamento de projetos.  **Ator primário:** Usuário.  **Pré-condição:** O usuário está autenticado no sistema. O usuário é o proprietário do projeto ou tem permissões de visualização para o projeto.  **Pós-condição:** O usuário pode visualizar o progresso atual do projeto e suas tarefas associadas. As informações sobre o projeto e suas tarefas são atualizadas refletindo o estado mais recente.  **Fluxo Principal:**   1. O usuário seleciona o projeto que deseja acompanhar na interface do sistema. 2. O sistema exibe a página de detalhes do projeto, incluindo informações sobre o projeto, a lista de tarefas e o status geral do projeto. 3. O usuário pode visualizar o status atual do projeto, que inclui a conclusão geral, as tarefas pendentes e concluídas. 4. O usuário pode navegar para a seção de tarefas para obter informações detalhadas sobre cada tarefa, incluindo descrição, data de vencimento, responsável e status 5. O usuário pode selecionar tarefas individuais para obter mais detalhes, marcar tarefas como concluídas ou atualizar informações relacionadas à tarefa, se tiver permissões apropriadas. 6. O fluxo é concluído.   **Fluxo alternativo:**   1. Se o usuário atualizar informações relacionadas a uma tarefa (por exemplo, marcar uma tarefa como concluída), o sistema atualizará o status da tarefa e recalcula o status geral do projeto.   **Fluxo de exceção:**   1. Caso ocorra uma falha ao atualizar o progresso do projeto no banco de dados: O sistema exibe uma mensagem de erro informando que o progresso não pôde ser atualizado. 2. O fluxo é interrompido. |

|  |
| --- |
| **CSU05 – Consultar estimativa do projeto ser concluído no prazo** |
| **Descrição:** Este caso de uso permite que o usuário consulte as estatísticas do projeto para determinar a probabilidade de conclusão dentro do prazo estipulado.  **Ator primário:** Usuário.  **Pré-condição:** O usuário está autenticado no sistema. O usuário tem acesso a projetos registrados no sistema.  **Pós-condição:** O usuário recebe informações estatísticas sobre o projeto, incluindo a probabilidade de conclusão dentro do prazo.  **Fluxo Principal:**   1. O usuário seleciona o projeto desejado no aplicativo. 2. O sistema exibe os detalhes o projeto, incluindo sua prioridade atual. 3. O sistema exibe um botão logo ao lado das informações do projeto. 4. Ao selecionar o botão, o sistema inicia o processo de coleta de dados. 5. Com base nos dados coletados, o sistema realiza o cálculo para determinar a probabilidade de conclusão do projeto dentro do prazo estipulado 6. O usuário então visualiza o relatório em forma de porcentagem 7. O usuário analisa as estatísticas fornecidas para avaliar a situação atual do projeto. 8. Se necessário, o usuário toma decisões informadas, como reatribuir tarefas, ajustar prazos ou alocar mais recursos com base nas informações apresentadas. 9. O fluxo é concluído.   **Fluxo alternativo:**   1. 1. Se o usuário não estiver satisfeito com a probabilidade de conclusão apresentada, ele pode optar por acessar funcionalidades adicionais, como relatório para obter informações mais específicas sobre o andamento do projeto.   **Fluxo de exceção:**   1. Se o sistema não conseguir coletar dados relevantes devido a problemas técnicos ou falhas, exibir uma mensagem de erro e oferecer opções de tentar novamente ou contatar o suporte técnico. 2. O fluxo é interrompido. |

|  |
| --- |
| **CSU06 – Gerenciar contas de usuários** |
| **Descrição:** Este caso de uso permite que o administrador do sistema possa gerenciar as contas de usuários, incluindo a criação, edição, exclusão e visualização de informações associadas a cada conta de usuário.  **Ator primário:** Administrador.  **Pré-condição:** O administrador está autenticado no aplicativo.  **Pós-condição:** As alterações nas contas de usuários são refletidas no sistema.  **Fluxo Principal:**   1. O Administrador do Sistema acessa a interface principal do sistema após realizar o login. 2. Na interface principal, o Administrador do Sistema seleciona a opção de gerenciar contas de usuários. 3. O sistema exibe a lista de contas de usuários existentes, com opções para criar, editar ou excluir contas. 4. O administrador pode criar contas, editar e excluir contas 5. O fluxo é concluído.   **Fluxo alternativo:**   1. Se o Administrador desejar visualizar detalhes adicionais de uma conta de usuário sem realizar alterações, ele pode selecionar a opção de visualização detalhada em vez de edição.   **Fluxo de exceção:**   1. Se ocorrer um erro durante a criação, edição ou exclusão de uma conta de usuário (por exemplo, campos obrigatórios não preenchidos corretamente), o sistema exibe mensagens de erro apropriadas e permite que o Administrador do Sistema corrija as informações. 2. O fluxo é interrompido. |

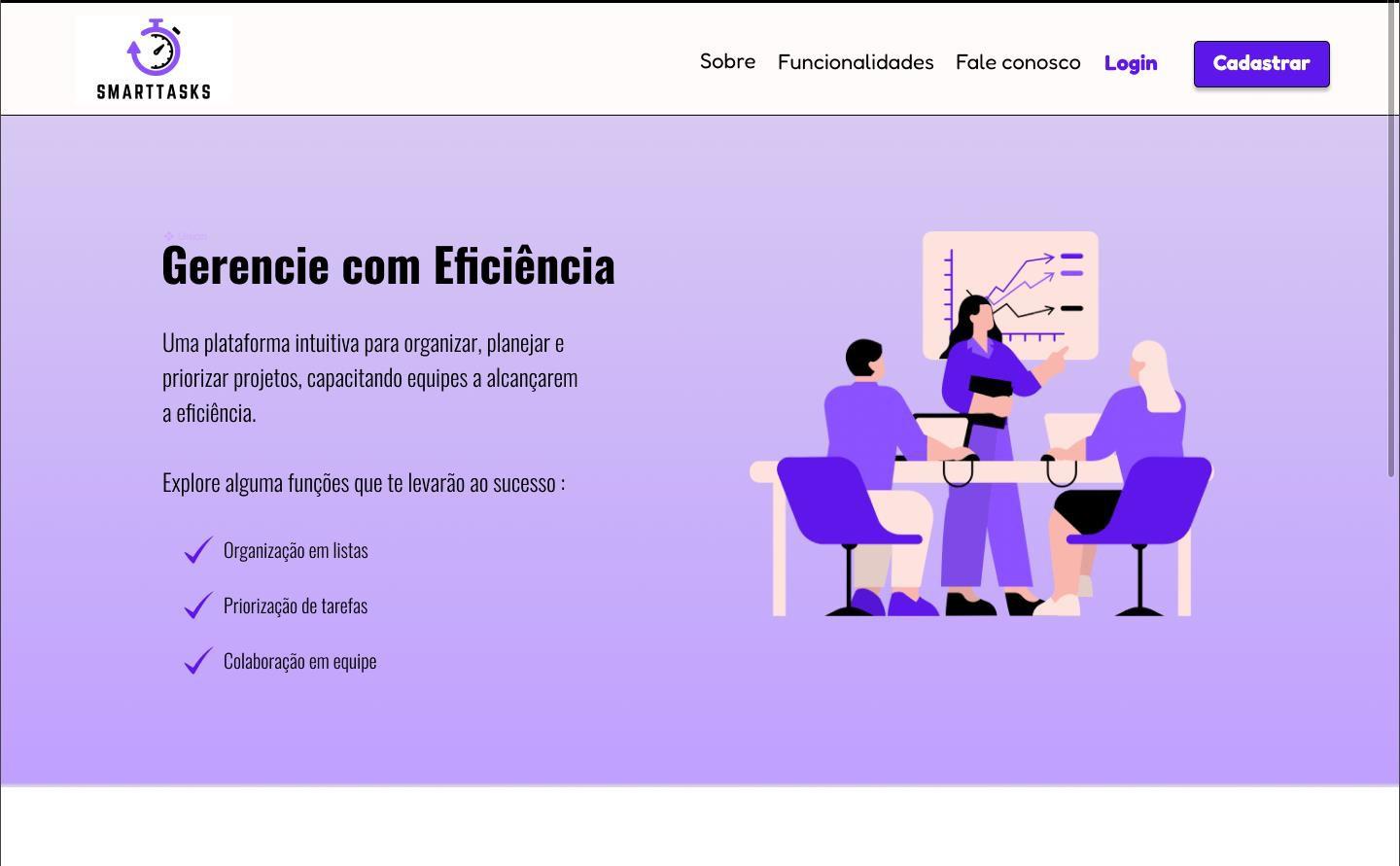
|  |
| --- |
| **CSU07 – Conceder permissões** |
| **Descrição:** Este caso de uso descreve como os administradores do sistema podem conceder permissões específicas a determinados usuários, permitindo-lhes acessar recursos ou funcionalidades específicas no sistema.  **Ator primário:** Administrador.  **Pré-condição:** O administrador está autenticado no sistema. O administrador possui permissões de administração.  **Pós-condição:** As alterações nas permissões são refletidas no sistema, permitindo ou restringindo o acesso conforme configurado.  **Fluxo Principal:**   1. O Administrador do Sistema acessa a interface principal do sistema após realizar o login. 2. Na interface principal, o Administrador do Sistema seleciona a opção de gerenciar permissões. 3. O sistema exibe a lista de usuários cadastrados, com opções para conceder ou revogar permissões. 4. O fluxo é concluído.   **Fluxo alternativo:**   1. Se o Administrador precisar revisar as permissões concedidas a um usuário antes de fazer alterações, ele pode selecionar a opção de visualização detalhada das permissões antes de conceder ou revogar.   **Fluxo de exceção:**   1. Se ocorrer um erro durante a concessão ou revogação de permissões (por exemplo, permissões inexistentes ou indisponíveis), o sistema exibe mensagens de erro apropriadas e permite que o Administrador do Sistema corrija as informações. |

3.5.3 Associação casos de uso x requisitos

|  |  |
| --- | --- |
| Requisito | Caso de uso |
| RNF01 | UC01 |
| RF01 | UC02, UC03 |
| RF02, RF03 | UC05 |

3.6 Protótipos das interfaces do sistema

Tela inicial – Figura 2



FONTE: Elaborada pelos autores (2024)

A tela inicial é onde o usuário começará após entrar no site, nela temos um pequeno texto sobre os usos da plataforma e algumas de suas funções. Na parte superior haverá um cabeçalho com cinco links: “Sobre”, que levara para uma página com informações mais detalhadas sobre a plataforma no geral; “Funcionalidades”, que levará para uma página com mais detalhes sobre as funções da plataforma; “Fale Conosco”, que levará para uma página onde o usuário poderá requisitar suporte aos desenvolvedores; “Login”, onde o usuário poderá entrar em uma conta já cadastrada; E “Cadastrar”, onde o usuário poderá se cadastrar caso ainda não possua uma conta ou cadastrar uma nova conta caso já possua uma.

**Tela de Login** – Figura 3

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

FONTE: Elaborada pelos autores (2024)

A tela de login é onde o usuário irá colocar seu e-mail e senha previamente cadastrados para acessar sua conta, logo após colocar essas informações haverá o botão “Entrar” para acessar a conta do usuário.

# Tela de Cadastro – Figura 4

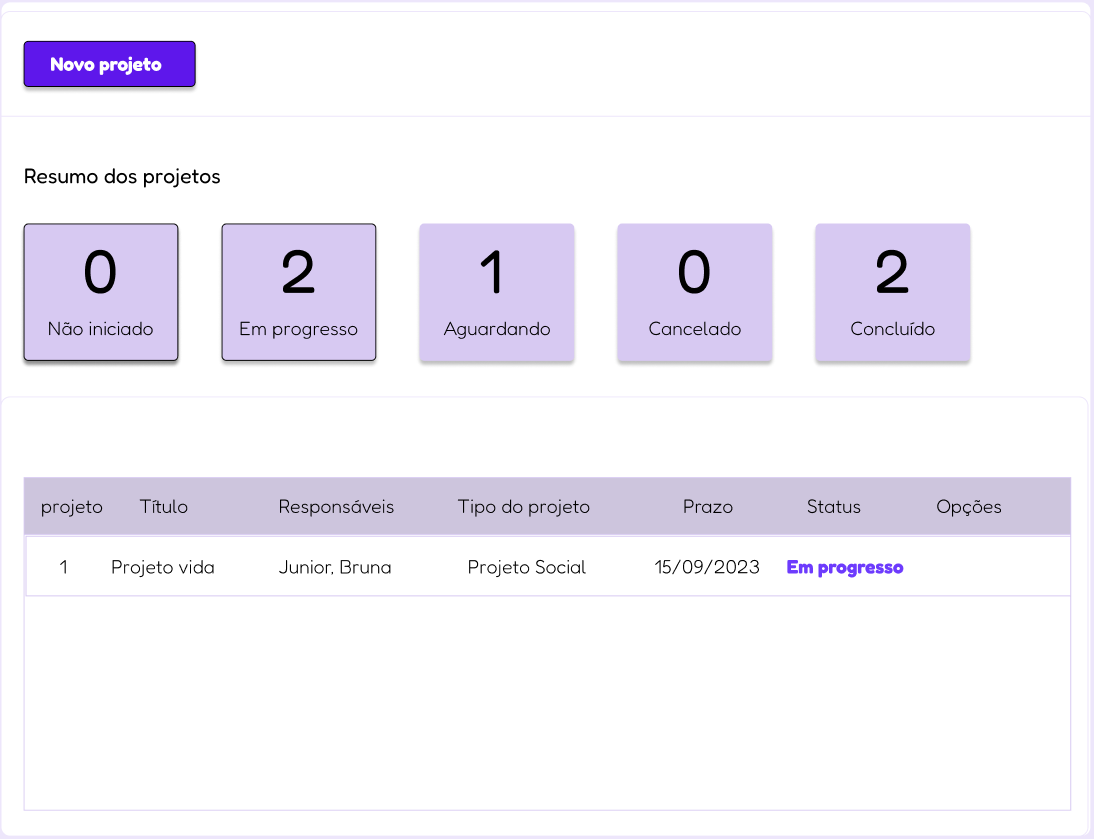
Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

FONTE: Elaborada pelos autores (2024)

Essa tela é onde o usuário irá se cadastrar suas informações quando criar uma conta na plataforma, ela terá os campos: “Nome”, nome do usuário; “Email”, e-mail do usuário; e “Senha”, a senha usada para a conta na plataforma. E por fim um botão “Cadastrar” para confirmar o cadastro.

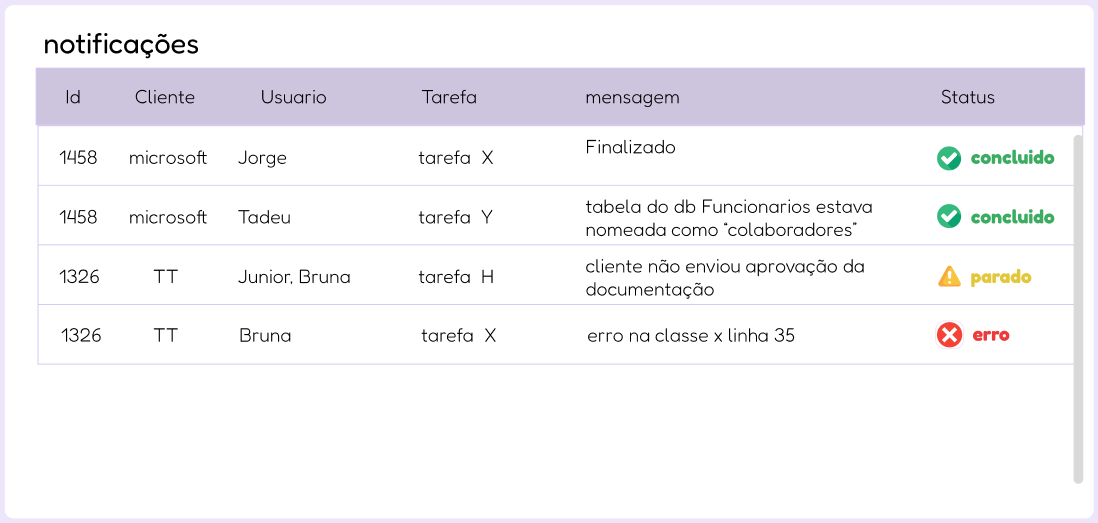
##### Dashboard – Figura 5

****

FONTE: Elaborada pelos autores (2024)

A tela Dashboard é onde os usuários poderão visualizar a lista com os projetos cadastrados na plataforma. A parte superior da tela terá um cabeçalho com um botão “Novo projeto” onde o usuário poderá criar um novo projeto que será adicionado a lista, logo abaixo haverá cinco quadros indicando o número de *projetos* não iniciados, em progresso, em aguardo, cancelados e concluídos. E por fim, abaixo haverá um quadro com a lista de projetos, com informações como numeração do projeto, título, os usuários responsáveis pela supervisão do projeto, categoria do projeto, prazo, status e opções.

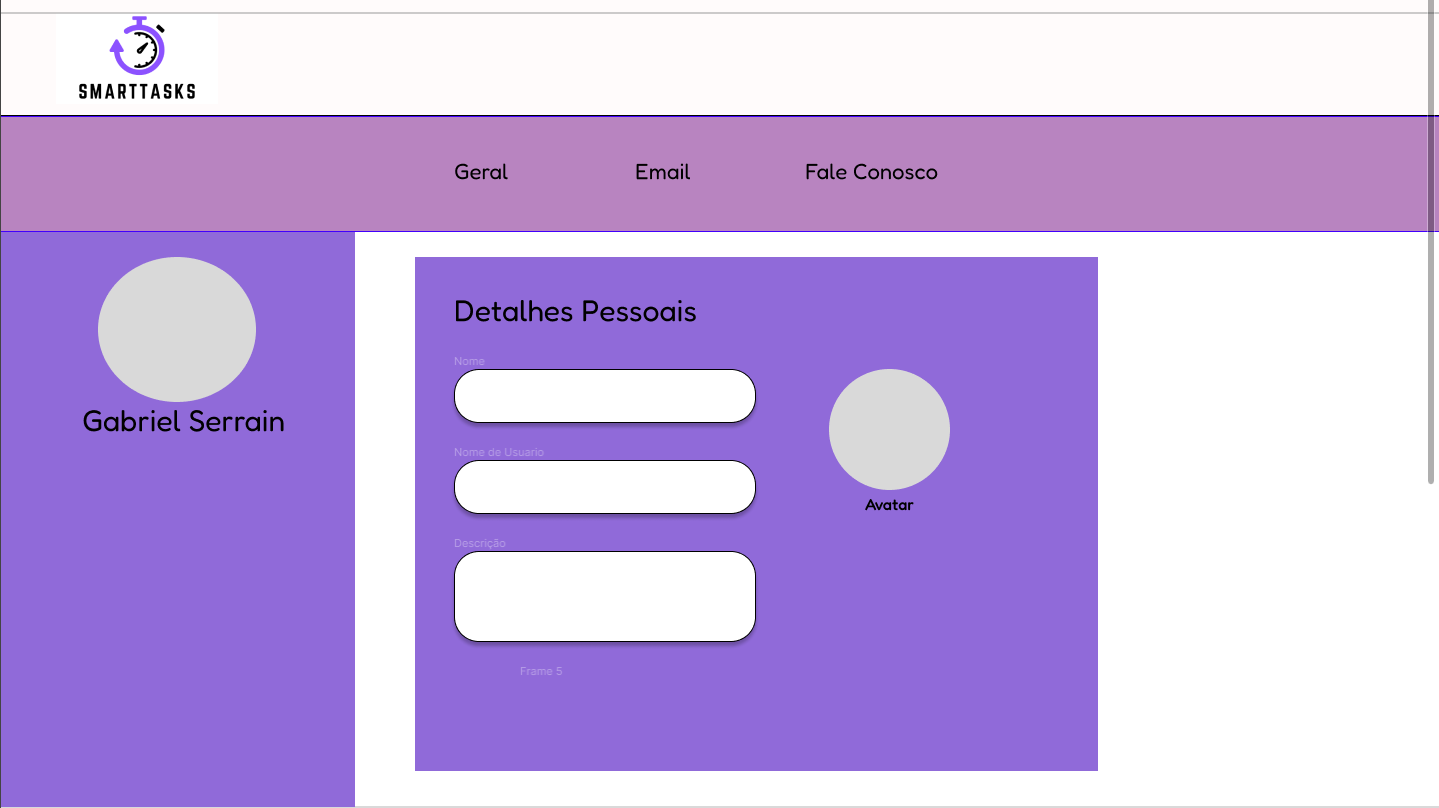
###### Notificações – Figura 6



FONTE: Elaborada pelos autores (2024)

A tela de notificações é onde os usuários poderão ser notificados de atualizações nas tarefas dos projetos aos quais estão associados. Essa tela mostrará as seguintes informações: “id”, que seria a numeração do projeto a que a tarefa pertence; “Cliente”, que seria o cliente que solicitou o projeto; “Usuário”, que seria o usuário que atualizou a tarefa; “Tarefa”, seria a tarefa específica que foi atualizada; “mensagem”, seria uma breve mensagem descrevendo o motivo da atualização da tarefa; e “Status”, que indica se a tarefa, está concluída, parada e etc.

**Tela de Perfil** – Figura 7

****

FONTE: Elaborada pelos autores (2024)

A tela de perfil é onde os usuários podem visualizar informações que serão expostas a outros usuários da plataforma, como avatar/foto do usuário, nome, nome de usuário e uma breve descrição sobre o usuário dono do perfil, o dono do perfil também poderá alterar suas informações caso queira. No cabeçalho também haverá um botão “Email” para caso o usuário dono do perfil queira mudar sua senha ou o e-mail vinculado a sua conta.

#### Tela de Progresso – Figura 8

FONTE: Elaborada pelos autores (2024)

Tela de progresso é onde os usuários poderão ver o progresso de conclusão de seus projetos. Os projetos serão mostrados em quadros que terão um botão “Acessar” que irá direcionar o usuário ao projeto, e o progresso será mostrado em forma de porcentagem de acordo com o número de tarefas já finalizadas no projeto especificado. No cabeçalho na parte superior haverá uma mensagem “Bem-vindo(a), usuário (onde haverá o nome de usuário)” e no canto esquerdo do cabeçalho haverá um símbolo de sino que abrira um pequeno quadro com as notificações de forma resumida, caso o usuário clique em “MOSTRAR TODAS” ele será enviado para a tela de notificações onde poderá ver as notificações completas.

Acessar Projeto – Figura 9

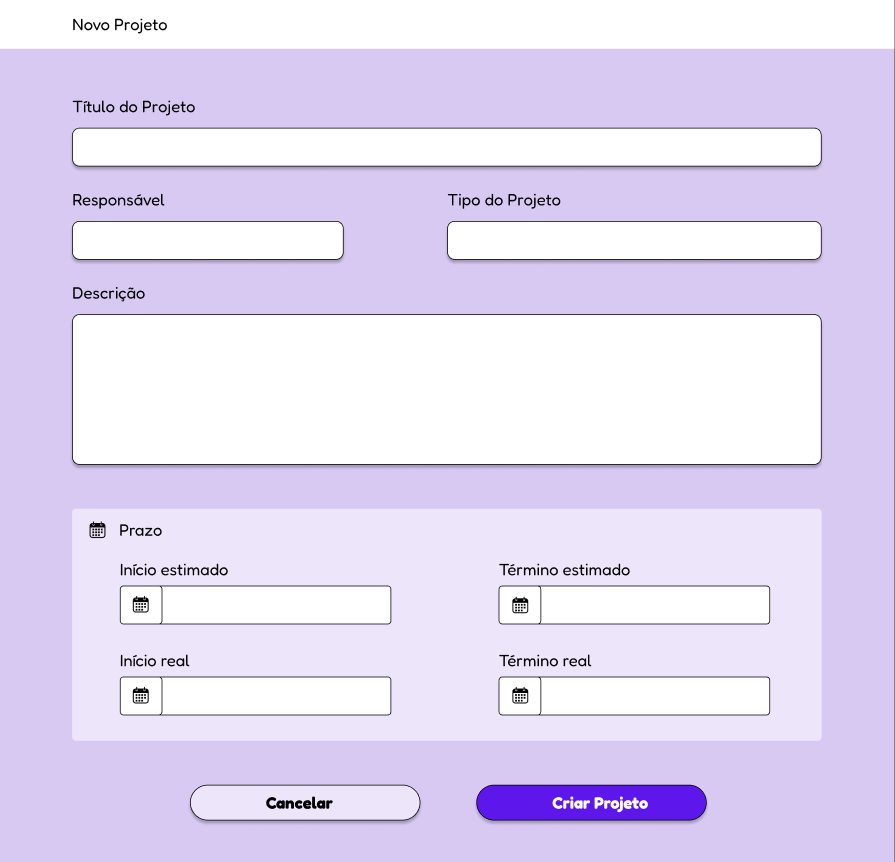
Interface gráfica do usuário, Aplicativo, Teams

Descrição gerada automaticamente

FONTE: Elaborada pelos autores (2024)

A tela “Acessar projeto” é onde o usuário poderá ver a lista de tarefas de um determinado projeto. No cabeçalho estará escrito o nome do projeto, abaixo haverá um botão “Descrição” que mostraria a descrição do projeto ao usuário e um botão “Nova tarefa” para caso um dos usuários responsáveis pelo projeto queira adicionar uma nova tarefa. E abaixo haverá um quadro com as tarefas do projeto, com informações como: “id”, que seria a numeração da tarefa; “Titulo”, se refere ao nome dado a tarefa; “Autor”, seria o usuário que criou a tarefa; “Prioridade”, seria o nível de urgência da tarefa; “Prazo”, seria a data para finalização da tarefa; “Status”, seria se a tarefa está concluída, ativa e etc.; e “Opções”, que terá um botão “Visualizar” que levara o usuário para uma tela onde ele poderá ver as especificações da tarefa.

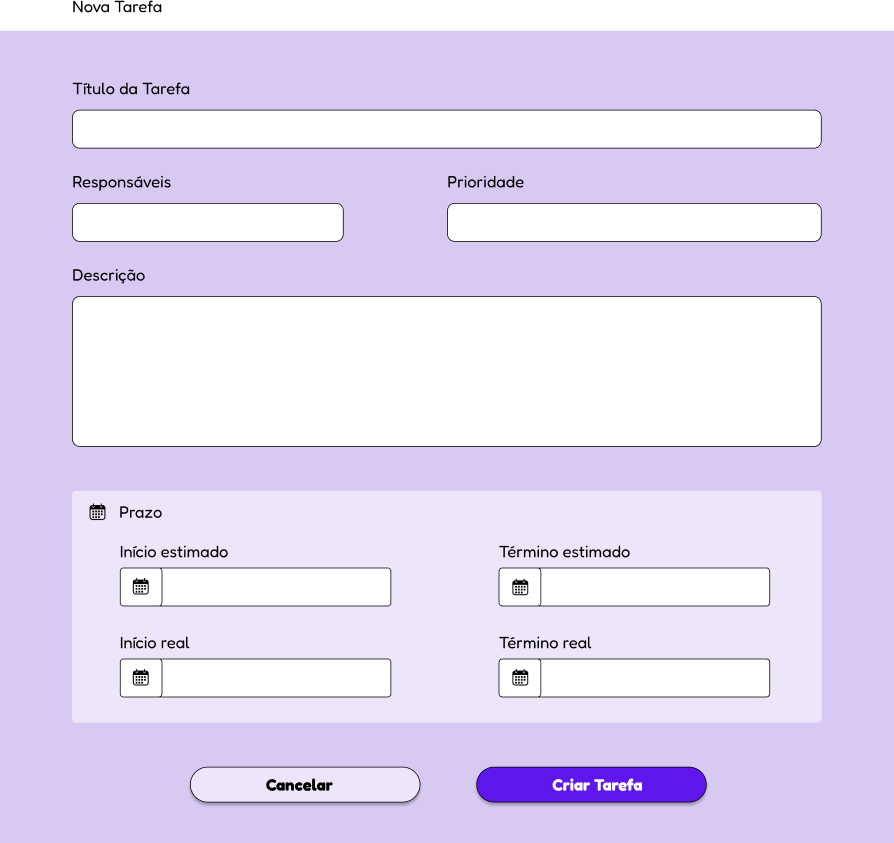
##### Tela de Criar Projeto – Figura 10

****

FONTE: Elaborada pelos autores (2024)

A tela de criação de projeto é onde o usuário poderá criar um novo projeto, na parte superior dessa tela haverão os campos: “Título do Projeto”, onde ele irá colocar o nome do projeto; “Responsável”, onde o usuário responsável pelo projeto será evidenciado; “Tipo do projeto”, que seria a categoria a que o projeto pertence; “Descrição”, que seria uma descrição breve ou detalhada sobre o projeto. Logo abaixo haverá um quadro “Prazo” com os campos: “Inicio estimado”, que seria a data estimada para início do projeto; “Término estimado”, que seria a data estimada para o término do projeto; “Inicio real”, que seria a data em que o projeto de fato foi iniciado; e “Término real”, que seria a data em que o projeto de fato foi finalizado. E abaixo do quadro “Prazo” há um botão cancelar para caso o usuário queira cancelar a criação do projeto e um botão criar projeto para finalizar a criação do projeto.

### Tela de Criar Tarefa – Figura 11



FONTE: Elaborada pelos autores (2024)

A tela de criação de tarefa é similar a tela criação de projeto, ela é onde o usuário poderá criar uma nova tarefa, na parte superior dessa tela haverão os campos: “Título da Tarefa”,

onde ele irá colocar o nome da tarefa; “Responsáveis”, onde os usuários responsáveis pela tarefa serão evidenciados; “Prioridade”, que seria o nível de urgência da tarefa definido pelo criador dela, após a criação da tarefa, a IA da plataforma irá analisar as informações sobre a tarefa registrada e sugerir uma prioridade que ela julgue mais adequada para a tarefa; “Descrição”, que seria uma descrição breve ou detalhada sobre a tarefa. Logo abaixo haverá um quadro “Prazo” com os campos: “Inicio estimado”, que seria a data estimada para início da tarefa; “Término estimado”, que seria a data estimada para o término da tarefa; “Inicio real”, que seria a data em que a tarefa de fato foi iniciada; e “Término real”, que seria a data em que a tarefa de fato foi finalizada. E abaixo do quadro “Prazo” há um botão cancelar para caso o usuário queira cancelar a criação da tarefa e um botão criar projeto para finalizar a criação da tarefa.

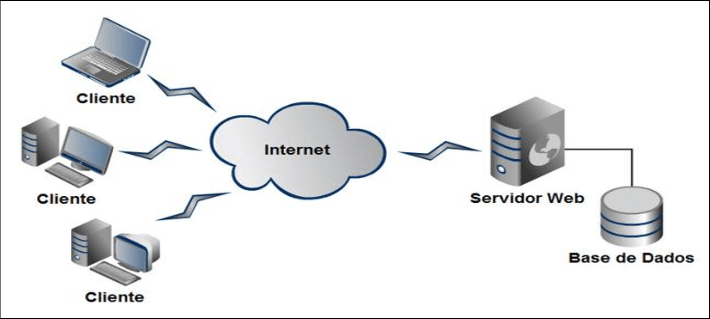
**4 PROJETO DO SISTEMA**

4.1 Arquitetura do software

Arquitetura Cliente - Servidor

A arquitetura cliente - servidor foi definida para atender o requisito de aplicação web. O lado do cliente é definido pela interação do cliente com a interface por meio do navegador. O cliente faz as requisições para o servidor, lado que processa as requisições, busca os dados no banco de dados e os carrega para a interface. A figura abaixo ilustra essa arquitetura, os computadores representam o lado do cliente, os “raios” ilustram a rede via conexão com a internet, representada pela nuvem, e com o servidor. O servidor faz conexão com banco de dados para buscar os dados e retornar para o cliente.

Figura 12



Diagrama

Descrição gerada automaticamenteA Figura 13 apresenta o diagrama de pacotes do sistema, destacando a organização de seus componentes. Além disso, a arquitetura do software é retratada na figura, evidenciando uma arquitetura em camadas. Isso se deve à composição do sistema por diversas tecnologias distintas.

Figura 13 - Diagrama de Pacotes

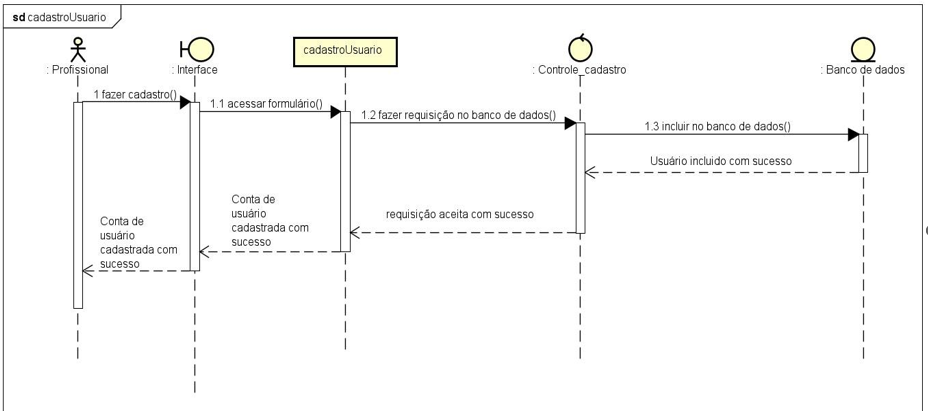
FONTE: Elaborada pelos autores (2024)

4.3. Diagramas de sequência

As figuras de 19 a 26 ilustram cenários que descrevem como objetos se comunicam ao longo do tempo, revelando a sequência de mensagens trocadas entre eles.

### 4.3.1 Cenário: Cadastro funcionário, gestor e admin.

Figura 14: Cenário de cadastro de usuário



FONTE: Elaborada pelos autores (2024)

4.3.2 Cenário: Login funcionário, gestor e admin.

Figura 15: Cenário de login de usuários

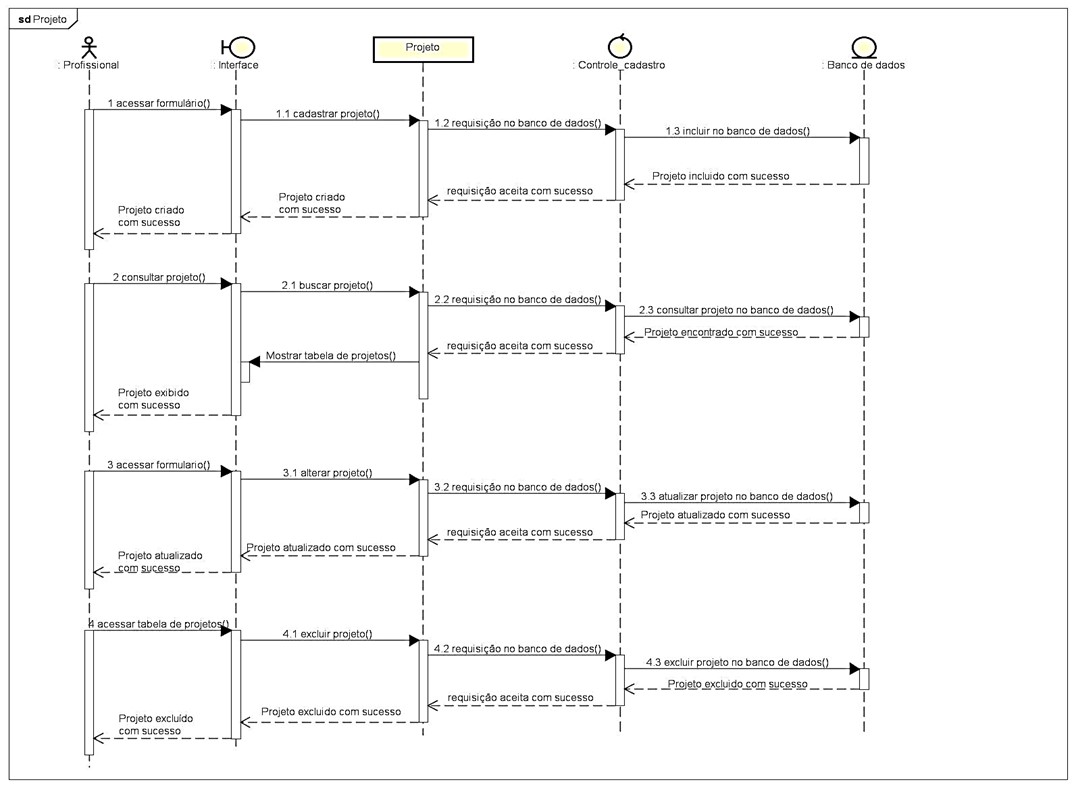
Diagrama

Descrição gerada automaticamente

:FONTE: Elaborada pelos autores (2024)

4.3.3 Cenário: Cadastro de projeto, consulta, alteração e exclusão.

Figura 16: Cenário de cadastro de projeto



FONTE: Elaborada pelos autores (2024)

4.3.4 Cenário: Cadastro de tarefa, consulta, alteração e exclusão.

Figura 17: Cenário de cadastro de tarefal

Calendário

Descrição gerada automaticamente com confiança média

FONTE: Elaborada pelos autores (2024)

4.2.5 Cenário: Estimativa de conclusão de projeto

Figura 18: Cenário de estimativa de conclusão

Diagrama

Descrição gerada automaticamente

FONTE: Elaborada pelos autores (2024)

4.2.6 Cenário Gestor: Cadastro de projeto, consulta, alteração e exclusão.

Diagrama

Descrição gerada automaticamente

4.2.7 Gestor: Cadastro de tarefa, atribuição de responsáveis consulta, alteração e exclusão.

Diagrama

Descrição gerada automaticamente

4.2.8 Cenário: Estimativa de conclusão de projeto

Diagrama

Descrição gerada automaticamente

4.2.9 Cenário Admin: Adicionar Usuário.

**Diagrama

Descrição gerada automaticamente**

4.2.10 Cenário Admin: Alterar ou excluir usuário.

Diagrama

Descrição gerada automaticamente

4.3 Persistência de dados

Para gerenciar as tabelas, criar ou modificar estruturas, fazer consultas SQL, inserir, atualizar e deletar dados, foi definido o banco de dados relacional MySQL. Este banco de dados foi escolhido por compor as ferramentas do XAMPP, um simulador de ambiente de desenvolvimento web local que possui uma interface amigável e compatível com as ferramentas entre si, fornecendo componentes importantes para o desenvolvimento de projeto web como o servidor web Apache, MySQL e compatibilidade com a linguagem PHP.

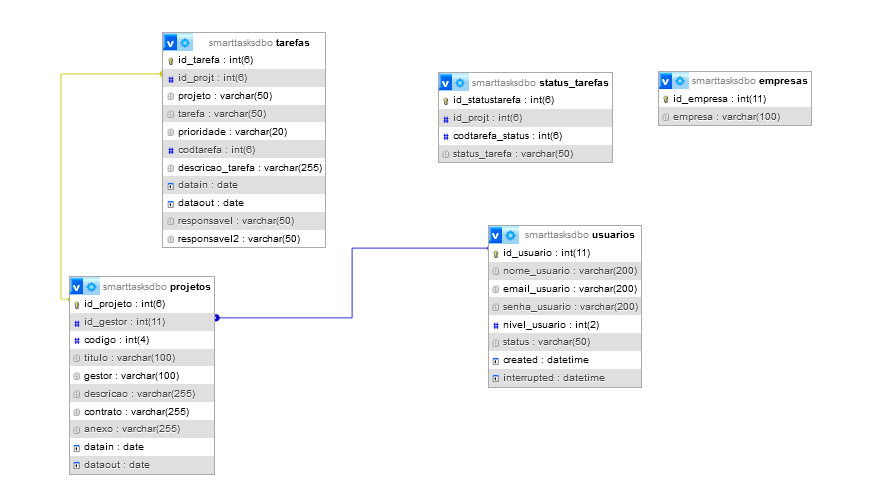
Na aplicação web SmartTasks os dados persistem no banco a partir das entradas via cadastro de usuários, de projetos, tarefas e empresas. Por meio do registro dessas entradas de dados é possível realizar toda operação de contagens de projetos, de tarefas e de pessoas, estimar marcos e prazos para projetos. Após os processamentos, recuperar e carregar dados para visualização.

4.3.1 Características do SGDB

O banco de dados MySQL é estruturado, modelo relacional organizado por tabelas que se relacionam por meio de chaves primárias e estrangeiras. Suporta vários tipos de dados como strings, inteiros, double, datas entre outros. Utiliza linguagem SQL para fazer consultas e manipulação de dados.

4.3.2 Modelo de dados

O modelo a seguir está em processo de remodelagem de banco de dados



4.4 Plano de testes

Neste capítulo a função testada foi a Estimativa de prazo de um projeto, a principal função de destaque no sistema. O objetivo da estimativa é medir o progresso do projeto em relação ao tempo determinado pelo gestor no cadastro do projeto.

Para medir o progresso do projeto baseamos nossa estimativa em tarefas a serem realizadas em um tempo específico de dias. A metodologia usada divide a quantidade total de tarefas e a duração do projeto em quatro marcos de 25%. Para as tarefas considera-se sempre um número inteiro maior caso a divisão retorne um número quebrado e para os dias considera-se um dia a menos da data de entrega, pois essa data precisa ser reservada para a entrega acordada do projeto. Para saber se o projeto vai atrasar ou não, a função compara a quantidade de tarefas e a data do marco com o dia que o cálculo é acionado e o que já foi realizado até aquele momento.

Toda tarefa deve ser designada a dois responsáveis, sendo um obrigatório e o outro é opcional, a razão deste modelo de responsabilização é para que a tarefa não sofra atraso em caso de um dos responsáveis se ausentar do projeto. É dever do responsável atualizar o status da tarefa, é baseado nesse status o progresso ou não do projeto. Caso o responsável pela tarefa não o faça, a visão do gestor em relação ao projeto ficará distorcida.

As tarefas possuem uma classificação de prioridade e a definição desta é estabelecida entre gestor e equipe. O tempo de execução da tarefa é responsabilidade do responsável pela tarefa, este indica no cadastro da tarefa o tempo de realização em dias.

Essa funcionalidade retorna para o gestor informações sobre o projeto como o título, descrição, quantidade total de tarefas, quantidades de tarefas a serem concluídas, a data de início, a data de entrega, duração do projeto, o progresso e o resultado do cálculo exibido na tela. A proposta do sistema é auxiliar o gestor no acompanhamento do projeto para que este tome a decisão mais adequada para concluir o projeto dentro do prazo estabelecido.

Abaixo foram simuladas quatro situações para testar a função

Situação 1: Projeto iniciando

Quando se inicia um projeto, o cálculo deve retornar o progresso do projeto em 0% e a relação entre quantidade de tarefas a serem concluídas e o total de tarefas. Nos marcos são indicadas as

faixas percentuais do projeto para serem alcançadas, a quantidade mínima de tarefas que devem ser realizadas até uma data estimada pelo sistema.



Situação 2: O projeto extrapolou o prazo e os status das tarefas já foram modificados

Quando os status são concluídos o progresso do projeto avança, no entanto, se faz necessário alcançar um mínimo estimado de tarefas até o prazo também estimado. Caso o mínimo esperado não seja concluído, o progresso estará estagnado em uma faixa percentual do projeto até que a quantidade mínima de tarefas seja cumprida. O cálculo do sistema também retornará uma

mensagem avisando o atraso do projeto e a faixa percentual em que está estagnado, em razão do marco não ter sido cumprido no tempo estimado.



Situação 3: Novo prazo, o prazo é alterado e a estimativa recalculada, sem alterar os status das tarefas

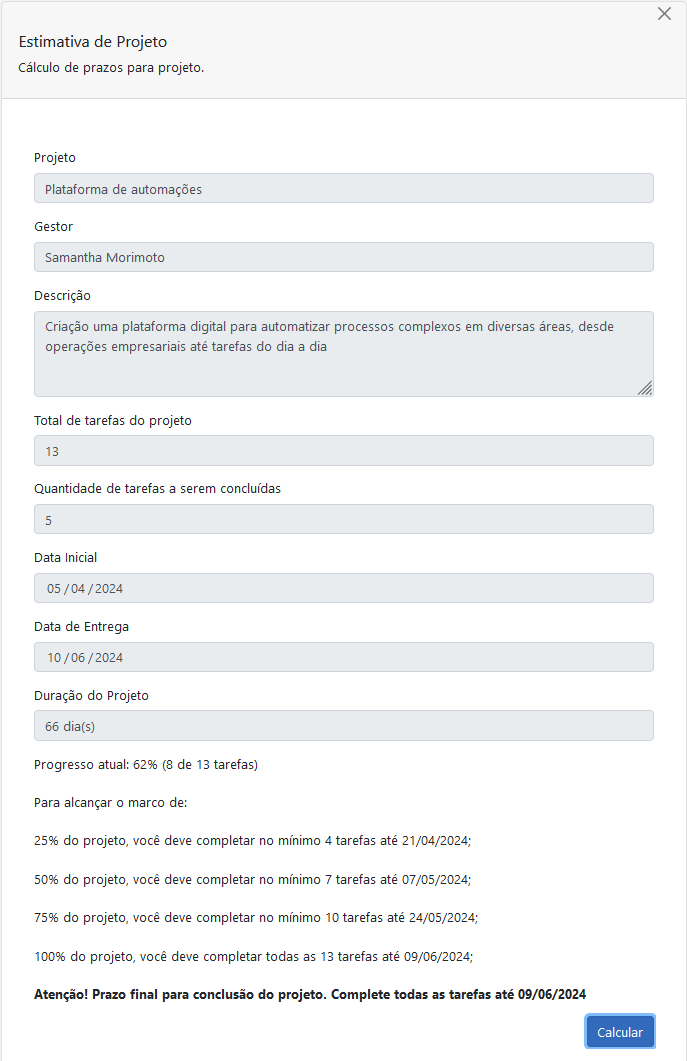
Quando o prazo é esticado, as estimativas de datas também acompanham o movimento, no entanto, a quantidade de tarefas permanece intacta em relação ao marco, isso ocorre porque o que movimenta o progresso do projeto são as tarefas com status concluído. Nessa situação os status das tarefas não sofreram alteração.

O cálculo do sistema também retornará uma mensagem avisando o atraso do projeto e a faixa percentual em que está estagnado, em razão do marco não ter sido cumprido no tempo estimado.



Situação 4: Prazo novo e alteração dos status das tarefas.

Quando o prazo é esticado e o status das tarefas é alterado para concluído o progresso do projeto avança. Se o projeto não estiver atrasado em relação ao dia que o cálculo é acionado, uma mensagem será exibida avisando a data limite para finalizar todas as tarefas.



**5 RESULTADOS**

Este capítulo aborda os resultados alcançados com a implementação do sistema SmartTasks, bem como a análise desses resultados em relação aos objetivos previamente definidos. Para uma melhor compreensão, os resultados foram organizados em módulos distintos.

5.1 Módulo de autenticação e conta de usuário

Nesta seção, será exibida e detalhada a tela de cadastro de usuário do sistema. A interface de cadastro é projetada para ser intuitiva e fácil de usar, permitindo que novos usuários se registrem no sistema de forma rápida e eficiente. Serão apresentados os principais componentes da tela, incluindo os campos obrigatórios para o preenchimento, como nome, e-mail, senha e confirmação de senha. O sistema garante a exclusividade dos nomes de usuário e a validade dos endereços de e-mail. Além disso, são destacados os elementos de usabilidade incorporados, como mensagens de validação. Esta seção também abordará o fluxo de navegação.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

FONTE: Elaborada pelos autores (2024)

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamenteA figural **tal** exibe e apresenta a tela de login do sistema. A interface de login é simples e direta, permitindo que os usuários existentes acessem suas contas de maneira rápida e segura. Serão apresentados os principais componentes da tela, incluindo os campos para inserção do e-mail e senha do usuário, que foram implementados buscando atingir os tópicos mencionados sobre os requisitos funcionais e não-funcionais. Esta seção também cobrirá o fluxo de navegação, orientando o usuário desde a abertura da tela de login até a confirmação do acesso ao sistema.

FONTE: Elaborada pelos autores (2024)

5.2 Modulo b

5.3 Modulo n

5.4 Discussão

**6 CONSIDERAÇÕES FINAIS**

Adicionar paragrafo de sugestões de trabalhos futuros

Como sugestões de trabalhos futuros...

**REFERÊNCIAS**

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT** **NBR 6024**: Informação e documentação – Numeração progressiva das seções de um documento – Apresentação. Rio de Janeiro, 2012.

\_\_\_\_\_\_. **NBR 14724**: Informação e documentação – Trabalhos acadêmicos – Apresentação. Rio de Janeiro, 2011.

\_\_\_\_\_\_. **NBR 10520**: Informação e documentação – Citações em documentos – Apresentação. Rio de Janeiro, 2002.

NEVES, Eduardo Borba; DOMINGUES, Clayton Amaral (Orgs.). **Manual de metodologia da pesquisa científica**. Rio de Janeiro: EB/CEP, 2007.

APÊNDICE A – Aqui vai o título do apêndice

O apêndice é um elemento opcional do trabalho, sendo um texto ou documento elaborado pelo autor necessário para completar sua argumentação sem prejuízo da unidade do trabalho.

Se houver muitos apêndices de forma que se esgotem as letras do alfabeto no título devem ser usadas letras maiúsculas dobradas (AA, AB, AC...).

ANEXO A – Aqui vai o título do anexo

O anexo é um texto ou documento não elaborado pelo autor que serve de fundamentação, comprovação ou ilustração. Trata-se de um elemento opcional do trabalho.

Se houver muitos anexos de forma que se esgotem as letras do alfabeto no título devem ser usadas letras maiúsculas dobradas (AA, AB, AC...).