



UNIVERSIDADE FEDERAL DE SANTA CATARINA CAMPUS ARARANGUÁ
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE (CTS)
TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO (TIC)

Guilherme Pereira Santana

TaskMaster: Utilização de práticas ágeis para otimização do gerenciamento de projetos por meio de um software

Araranguá

2023

Guilherme Pereira Santana

TaskMaster: Utilização de práticas ágeis para otimização do gerenciamento de projetos por meio de um software

Trabalho de Conclusão de Curso submetido ao curso de Tecnologias da Informação e Comunicação da Universidade Federal de Santa Catarina para a obtenção do título de Bacharel em Tecnologias da Informação e Comunicação.

Orientador: Prof. Fabrício Herpich, Dr.
Coorientador: Prof. Wilson Gruber, Dr.

Araranguá

2023

Santana, Guilherme Pereira

TaskMaster: Utilização de práticas ágeis para otimização do gerenciamento de projetos por meio de um software / Guilherme Pereira Santana ; orientador, Fabrício Herpich, coorientador, Vilson Gruber, 2023.

104 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Campus Araranguá, Graduação em Tecnologias da Informação e Comunicação, Araranguá, 2023.

Inclui referências.

1. Tecnologias da Informação e Comunicação. 2. Gestão de projeto. 3. Software. 4. Template. 5. Gestão ágeis. I. Herpich, Fabrício . II. Gruber, Vilson . III. Universidade Federal de Santa Catarina. Graduação em Tecnologias da Informação e Comunicação. IV. Título.

Guilherme Pereira Santana

TaskMaster: Utilização de práticas ágeis para otimização do gerenciamento de projetos por meio de um software

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do título de “Bacharel em Tecnologias da Informação e Comunicação” e aprovado em sua forma final pelo Curso de Graduação em Tecnologias da Informação e Comunicação.

Araranguá, 27 de Novembro de 2023



Prof. Fernando José Spanhol, Dr.
Coordenação do Curso

Banca examinadora



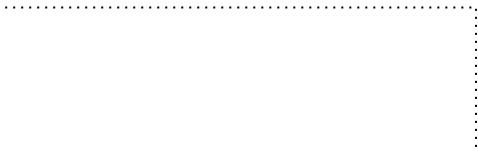
Prof. Fabrício Herpich, Dr.
Orientador



Prof. Vilson Gruber, Dr.
Coorientador



Prof.(a) Marina Carradore Sérgio, Dr.(a)
Universidade Federal de Santa Catarina



Prof. Rangel Machado Simon, Dr.
Universidade Federal de Santa Catarina

Araranguá, 2023.

AGRADECIMENTOS

Em primeiro lugar, desejo expressar minha profunda gratidão a Deus por sua orientação e graça ao longo desta jornada acadêmica. Sua luz e inspiração estiveram sempre presentes em cada passo que dei. Agradeço por me dar forças nos momentos de desafio e por guiar meus passos na busca do conhecimento e da realização pessoal.

Aos dedicados professores que moldaram meu caminho acadêmico, minha admiração e agradecimento. Suas orientações, conhecimento compartilhado e paixão pelo ensino foram fundamentais para a minha formação técnica e intelectual. Cada lição ministrada e cada conselho oferecido contribuíram significativamente para o meu crescimento.

Aos meus queridos pais, dedico este agradecimento. Sem o amor, apoio inabalável e sacrifícios que fizeram por mim, esta conquista não teria sido possível. Cada um de vocês desempenhou um papel crucial na minha jornada, e por isso sou eternamente grato. Aos amigos que compartilharam comigo esta jornada acadêmica, quero expressar minha sincera gratidão. O apoio mútuo e as amizades que desenvolvemos foram um pilar fundamental para nossa perseverança e sucesso.

A todos os mencionados acima e também àqueles que contribuíram de alguma forma na realização deste trabalho, quero expressar meu extremo sentimento de gratidão e apreço. Muito obrigado por seu apoio, orientação e influência positiva ao longo desta jornada. Cada um de vocês desempenhou um papel fundamental na concretização deste trabalho e no meu desenvolvimento pessoal e acadêmico.

RESUMO

Neste trabalho, será realizado uma análise abrangente dos princípios subjacentes ao gerenciamento de projetos, com especial atenção à aplicação prática desses conceitos por meio de um *software* desenvolvido. Será abordado o cenário atual, no qual muitos profissionais de gerenciamento de projetos enfrentam desafios decorrentes da falta de *softwares* especializados, *templates* prontos e agilidade na administração de suas tarefas. Uma das principais lacunas identificadas é a ausência de ferramentas que atendam especificamente às necessidades do gerenciamento de projetos em termos de simplicidade e personalização. Frequentemente, os *softwares* disponíveis no mercado não oferecem os *templates* e a agilidade necessários para se adequarem às particularidades de cada projeto. Isso pode resultar em ineficiências, atrasos e dificuldades na condução dos projetos. O *software* desenvolvido para este estudo visa preencher essa lacuna. Ele oferece uma plataforma flexível que permite aos profissionais de gerenciamento de projetos criar e personalizar *templates* de acordo com as exigências específicas de seus projetos. Além disso, a interface intuitiva e a metodologia ágil incorporada no *software* visam simplificar e acelerar o processo de gerenciamento de projetos. Ao utilizar esta ferramenta, os profissionais podem agilizar a criação de planos de projeto, acompanhar o progresso, alocar recursos de forma eficaz e gerenciar tarefas de maneira mais eficiente. Isso não apenas economiza tempo, mas também contribui para o sucesso geral do projeto, garantindo que todas as etapas sejam executadas com eficiência e de acordo com as metas estabelecidas. Em resumo, este *software* desempenha um papel fundamental na melhoria do gerenciamento de projetos, oferecendo a flexibilidade, os *templates* personalizados e a agilidade necessários para enfrentar os desafios da administração de projetos de maneira eficaz. Isso não apenas beneficia os profissionais envolvidos, mas também aumenta a probabilidade de alcançar com êxito os objetivos de cada projeto.

Palavras-chave: Gerenciamento de projeto; Templates; Agilidade; Software.

ABSTRACT

In this work, a comprehensive analysis of the principles underlying project management will be carried out, with special attention to the practical application of these concepts through software. We address the current scenario in which many project management professionals face challenges due to the lack of specialized software, ready-made templates, and agility in managing their tasks. One of the main gaps identified is the absence of tools that specifically cater to the needs of project management in terms of simplicity and customization. Often, the software available in the market does not provide the necessary templates and agility to adapt to the specifics of each project. This can result in inefficiencies, delays, and difficulties in project execution. The software developed for this study aims to bridge this gap. It offers a flexible platform that allows project management professionals to create and customize templates according to the specific requirements of their projects. Additionally, the intuitive interface and agile methodology incorporated into the software aim to simplify and expedite the project management process. By using this tool, professionals can streamline the creation of project plans, track progress, allocate resources effectively, and manage tasks more efficiently. This not only saves time but also contributes to the overall success of the project, ensuring that all stages are executed efficiently and in accordance with established goals. In summary, this software plays a pivotal role in improving project management by providing the flexibility, customized templates, and agility necessary to effectively address the challenges of project administration. This benefits not only the professionals involved but also increases the likelihood of successfully achieving the objectives of each project.

Keywords: Project management; Templates; Agility; Software.

LISTA DE FIGURAS

Figura 1	- Exemplo de caso de uso no TaskMaster	30
Figura 2	- Menu de personalização	31
Figura 3	- Botão de ativação do menu de personalização	31
Figura 4	- Demonstração do tema 'claro'	32
Figura 5	- Demonstração do tema escuro	32
Figura 6	- Barra de navegação	33
Figura 7	- Barra superior	33
Figura 8	- Tela Home (Dashboard)	34
Figura 9	- Seção avaliação na tela Home	34
Figura 10	- Seção tarefas na tela Home	35
Figura 11	- Relatório dos status das tarefas	36
Figura 12	- Análise gráfica do tempo gasto na execução de tarefas	36
Figura 13	- Exemplo de data grid	38
Figura 14	- Seção Home na tela Landing Page	39
Figura 15	- Seção Serviços na tela Landing Page	39
Figura 16	- Seção Páginas na tela Landing page	40
Figura 17	- Seção Contato na tela Landing Page	40
Figura 18	- Tela Login	42
Figura 19	- Tela Cadastre-se	44
Figura 20	- Tela Recupere sua senha	46
Figura 21	- Data grid da tela Usuário	50
Figura 22	- Aba Principal na tela Usuário	50
Figura 23	- Aba Endereço na tela Usuário	51
Figura 24	- Aba Configurações na tela Usuário	51
Figura 25	- Data grid da tela Template	52
Figura 26	- Aba Principal na tela Template	53
Figura 27	- Aba Atividades / Tarefas na tela Template	54
Figura 28	- Aba Cronograma na tela Template	54
Figura 29	- Data grid da tela Projeto	58
Figura 30	- Aba Principal na tela Projeto	59
Figura 31	- Aba Atividades / Tarefas na tela Projeto	60

Figura 32	- Aba Equipe / Funções na tela Projeto	60
Figura 33	- Aba Configurações na tela Projeto	61
Figura 34	- Data grid da tela Cronograma	66
Figura 35	- Tela Cronograma	67
Figura 36	- Timeline na tela Cronograma	67
Figura 37	- Tela Detalhes da tarefa	69
Figura 38	- Data grid da tela Administração de Tarefas	71
Figura 39	- Tela Administração de Tarefas	72
Figura 40	- Data grid da tela Registro de Tarefas	76
Figura 41	- Tela Registro de Tarefas	76
Figura 42	- Tela Chat	80
Figura 43	- Modelo de banco de dados utilizado no Taskmaster	98

LISTA DE QUADRO

Quadro 1	- Requisitos da tela Landing Page	41
Quadro 2	- Requisitos da tela Login	43
Quadro 3	- Regra de negócio da tela Login	44
Quadro 4	- Requisitos da tela Cadastre-se	45
Quadro 5	- Regra de negócio da tela Cadastre-se	45
Quadro 6	- Requisitos da tela Recupere sua senha	46
Quadro 7	- Regra de negócio da tela Recupere sua senha	47
Quadro 8	- Requisitos da tela Usuário	48
Quadro 9	- Regra de negócio da tela Usuário	50
Quadro 10	- Requisitos da tela Template	55
Quadro 11	- Regra de negócio da tela Template	58
Quadro 12	- Requisitos da tela Projeto	61
Quadro 13	- Regra de negócio da tela Projeto	65
Quadro 14	- Requisitos da tela Cronograma	68
Quadro 15	- Requisitos da tela Detalhes da Tarefa.....	69
Quadro 16	- Requisitos da tela Administração de Tarefas.....	71
Quadro 17	- Regra de negócio da tela Administração de Tarefas.....	73
Quadro 18	- Requisitos da tela Registro de Tarefas.....	76
Quadro 19	- Regra de negócio da tela Registro de Tarefas.....	78
Quadro 20	- Requisitos da tela Chat.....	79
Quadro 21	- Regra de negócio da tela Chat.....	81

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
CPF	Cadastro de Pessoas Físicas
CEP	Código de Endereçamento Postal
CLI	Command Line Interface
CSRF	Cross-Site Request Forgery
CSS	Cascading Style Sheets
DDD	Domain Driven Design
HTML	HyperText Markup Language
IDE	Integrated Development Environment
JIT	Just In Time
JWT	JSON Web Token
LINQ	Language Integrated Query
PDF	Portable Document Format
PMBOK	Project Management Body of Knowledge
PMI	Project Management Institute
PwC	Pulse of the Profession
SGBD	Sistema de Gerenciamento de Banco de Dados
SQL	Structured Query Language
SQLI	Structured Query Language Injection
TCC	Trabalho de Conclusão de Curso
VS	Visual Studio
XLS	Extensible Style Language
WIP	Work in Progress

SUMÁRIO

1 INTRODUÇÃO	14
1.1 CONTEXTUALIZAÇÃO	14
1.2 PROBLEMÁTICA	15
1.3 OBJETIVOS	17
1.3.1 Objetivo geral	17
1.3.2 Objetivos específicos	17
1.4 ESTRUTURA DO TRABALHO	18
2 FUNDAMENTAÇÃO TEÓRICA	19
2.1 GESTÃO DE PROJETOS	19
2.2 METODOLOGIA ÁGIL	21
2.3 MONITORAMENTO DE RISCOS	23
2.4 SOFTWARE DE GESTÃO DE PROJETO	24
3 TASKMASTER: DESENVOLVIMENTO DO SOFTWARE	26
3.1 REQUISITOS	26
3.2 CASOS DE USO	28
3.3 LAYOUT	30
3.3.1 Tema	31
3.3.2 Barra de navegação	32
3.3.3 Barra superior	33
3.4 HOME (<i>DASHBOARD</i>)	33
3.4.1 Avaliação	34
3.4.2 Tarefas	35
3.4.3 Gráficos	35
3.4.4 Data grid	37
3.5 TELAS DESENVOLVIDAS	38
3.5.1 Landing page	39
3.5.2 Login	42
3.5.3 Cadastre-se	44
3.5.4 Recupere sua senha	45
3.5.5 Usuário	47

3.5.6 Template	51
3.5.7 Projeto	58
3.5.8 Cronograma	65
3.5.9 Detalhes da tarefa	69
3.5.10 Administração de tarefas	70
3.5.11 Registro de tarefas	75
3.5.12 Chat	79
4 FUNDAMENTAÇÃO TÉCNICA	83
4.1 IDE (INTEGRATED DEVELOPMENT ENVIRONMENT)	83
4.1.1 Rider	83
4.1.2 Visual Studio Code	84
4.2 BACKEND	85
4.2.1 .NET	85
4.2.2 Jwt Token	86
4.2.3 Swagger	87
4.2.4 Automapper	87
4.2.5 SignalR	88
4.2.6 Entity Framework	89
4.2.7 DDD (Domain driven design)	90
4.3 FRONTEND	91
4.3.1 HTML	91
4.3.2 CSS	92
4.3.3 Bootstrap	92
4.3.4 Angular	93
4.3.5 Typescript	95
4.4 BANCO DE DADOS	95
4.4.1 Microsoft SQL Server	96
4.4.2 Modelo de banco de dados	96
5 CONCLUSÃO	98
5.1 TRABALHOS FUTUROS	100

1 INTRODUÇÃO

O gerenciamento de projetos desempenha um papel fundamental na vida de pessoas e organizações em um mundo cada vez mais dinâmico e complexo. Seja na busca por objetivos pessoais ou no alcance de metas empresariais, a capacidade de planejar, executar e controlar projetos de forma eficaz é um diferencial crucial. Isso porque, em um ambiente em constante transformação, a realização de tarefas sem uma estratégia bem definida pode levar a atrasos, desperdício de recursos e, em última instância, ao fracasso (VARGAS, 2018).

De acordo com o Project Management Institute (PMI), organização líder na promoção de boas práticas em gerenciamento de projetos (PMI, 2017, p.10) “o gerenciamento de projetos é a aplicação de conhecimento, habilidades, ferramentas e técnicas às atividades do projeto a fim de atender aos seus requisitos”. Isso destaca a complexidade intrínseca à gestão de projetos, que envolve uma gama diversificada de disciplinas, desde o planejamento estratégico até a execução detalhada.

Nas últimas décadas, houve uma mudança significativa na maneira como projetos são abordados. A crescente volatilidade dos mercados, a evolução rápida da tecnologia e as crescentes expectativas dos clientes têm impulsionado a transição de métodos tradicionais de gerenciamento de projetos para abordagens mais ágeis e adaptáveis.

1.1 CONTEXTUALIZAÇÃO

Para alcançar o sucesso na gestão de projetos requer não apenas uma abordagem metodológica sólida, mas também a utilização de ferramentas modernas, como *softwares* de gestão de projetos. O uso desses *softwares* oferece uma série de vantagens que podem impactar positivamente o resultado de um projeto. Para ilustrar a importância do uso de *softwares* de gestão de projetos na busca pelo sucesso, em um estudo realizado por James P. Lewis em seu livro "Project Planning, Scheduling & Control", Lewis destaca que o uso de ferramentas modernas de gerenciamento de projetos, como *softwares* específicos, pode aumentar significativamente a eficiência da equipe, melhorar a comunicação entre os membros

do projeto e proporcionar uma visão mais clara do andamento das tarefas e do cumprimento de prazos (LEWIS, 2011).

Além disso, Lewis ressalta que essas ferramentas permitem uma análise mais precisa dos dados do projeto, facilitando a tomada de decisões informadas e a identificação precoce de possíveis desvios ou problemas. Ele argumenta que, ao adotar uma abordagem metodológica sólida e combinar isso com o uso de *softwares* de gestão de projetos, as organizações podem melhorar sua capacidade de entrega de projetos dentro do prazo e do orçamento, o que, por sua vez, contribui para o sucesso geral da gestão de projetos.

1.2 PROBLEMÁTICA

No cenário atual, o fracasso e o não cumprimento de projetos representam desafios significativos tanto para indivíduos que buscam atingir metas pessoais quanto para grandes empresas que buscam realizar empreendimentos estratégicos. Independentemente do tamanho ou escopo do projeto, uma série de fatores contribui para esses desafios, revelando a complexidade inerente à gestão de projetos na era moderna. Neste contexto, é essencial explorar as causas subjacentes e as implicações dos fracassos e não cumprimentos de projetos em ambas as esferas.

Um estudo da Pulse of the Profession (PWC) em 2018 revelou que apenas 64% dos projetos são concluídos no prazo e 67% dentro do orçamento (PMI, 2018). Isso ressalta a necessidade de práticas de gerenciamento de projetos mais eficazes. A entrada de *softwares* de gestão de projetos desempenhou um papel fundamental na solução desse problema, melhorando o gerenciamento com análise de dados em tempo real, resultando em maior eficiência e conformidade com prazos e orçamentos.

As ferramentas de gestão de projetos representam uma verdadeira revolução no mercado, proporcionando melhorias significativas na maneira como os projetos são planejados, executados e monitorados. Essas ferramentas têm desempenhado um papel crucial no aprimoramento da eficiência, na economia de recursos e na redução de erros em uma variedade de setores.

Uma das maiores evoluções que essas ferramentas trouxeram ao mercado e aos projetos é a capacidade de facilitar a colaboração e a comunicação entre equipes distribuídas geograficamente. Isso permitiu que projetos complexos fossem gerenciados de maneira mais eficaz, com todas as partes envolvidas tendo acesso às informações relevantes em tempo real. Além disso, a automação de tarefas rotineiras e a geração de relatórios automatizados economizaram tempo e reduziram a probabilidade de erros humanos.

No entanto, apesar desses avanços notáveis, ainda há uma carência no mercado quando se trata de ferramentas que oferecem *templates* prontos para uso. A falta de *templates* predefinidos pode ser uma barreira para aqueles que não têm experiência prévia em gestão de projetos ou que desejam iniciar rapidamente um novo projeto.

Por outro lado, há também um excesso de ferramentas que adotam abordagens excessivamente detalhadas, com a necessidade de preencher inúmeros campos antes mesmo de começar um projeto. Isso pode ser avassalador e desencorajador para muitos usuários, especialmente aqueles que estão procurando simplicidade e agilidade em suas ferramentas de gestão de projetos.

Considerando o desafio que muitas pessoas enfrentam ao iniciar ou desenvolver um projeto sem possuir o conhecimento adequado, além da necessidade crescente de adotar processos ágeis que sejam menos onerosos em termos de preenchimento de campos surge então a necessidade de desenvolver um *software* que possa aprimorar esse processo. A pergunta-chave que surge é: "Como desenvolver um *software* de gestão de projetos que possa atender este nicho de mercado?"

Dado o exposto acima, neste projeto busca-se que o *software* considere os seguintes aspectos:

- Cada projeto é único, com requisitos e características distintas. O desenvolvimento do *software* deve levar em conta essas particularidades para garantir que atenda às necessidades específicas de cada projeto.
- O *software* deve permitir o controle completo dos projetos, incluindo o acompanhamento do andamento, a identificação de possíveis atrasos e a gestão eficaz dos recursos disponíveis.

- Projetos personalizados muitas vezes exigem adaptações constantes. O *software* deve ser flexível o suficiente para permitir a personalização de processos e fluxos de trabalho de acordo com as demandas de cada projeto.
- Além do gerenciamento básico, a integração de funcionalidades como gráficos interativos, comunicação em tempo real, controle de atividades e cronograma pode agregar valor ao *software*.
- A adoção do *software* depende da usabilidade e da interface amigável. Os usuários precisam sentir que a ferramenta é intuitiva e facilita suas tarefas.

A abordagem metodológica para o desenvolvimento desse *software* de gestão de projetos personalizados deve considerar esses desafios e garantir que a solução seja eficaz, adaptável e fácil de usar. Ao final deste projeto, espera-se entregar um *software* testado e validado que melhore significativamente a gestão de projetos personalizados, facilitando a comunicação, o acompanhamento e o sucesso dessas iniciativas.

1.3 OBJETIVOS

1.3.1 Objetivo geral

O objetivo deste Trabalho de Conclusão de Curso (TCC) é conceber, projetar e desenvolver um *software* de gestão de projetos que atenda às necessidades específicas de organizações e equipes que buscam uma abordagem mais simples e eficiente para o gerenciamento de projetos.

1.3.2 Objetivos específicos

- Desenvolver um protótipo funcional do *software* de gestão de projetos, que inclua recursos necessário para alcançar a melhor de gerenciar e administrar um projeto.
- Documentar todo o processo de desenvolvimento e apresentar os resultados do TCC, destacando as contribuições do *software* para o campo de gerenciamento de projetos.

- Desenvolver uma arquitetura e *design* de *software*, fundamentados em conceitos técnicos sólidos e fazendo uso de alguns dos principais *frameworks* e tecnologias adotados no mercado profissional.

1.4 ESTRUTURA DO TRABALHO

O atual capítulo aborda as considerações iniciais sobre do tema deste trabalho, ou seja, apresenta a proposta de trabalho, delimitando a problemática, o tema e o objetivo deste trabalho.

O segundo capítulo aborda a fundamentação teórica que foi seguida para o desenvolvimento deste trabalho, a base e conceitos que foram usados para o desenvolvimento do *software*.

No terceiro capítulo aborda o que foi reproduzido no *software*, apresenta as telas desenvolvidas com os requisitos e regras de negócio da mesma, além de ressaltar alguns conceitos de gestão de projeto para o desenvolvimento da tela.

No quarto capítulo, será abordado a fundamentação técnica para o desenvolvimento, nele será abordado as tecnologias e os conceitos utilizados para a criação do *software* de forma organizada e eficiente. No último capítulo, será realizada as considerações finais a respeito deste trabalho, concluindo e ressaltando a importância dele e o quanto contribui para o desenvolvimento pessoal.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, serão explorados os conceitos essenciais relacionados à gestão de projetos, gerenciamento de riscos e metodologias ágeis. Esses conceitos serão posteriormente aplicados na criação e desenvolvimento do *software* “TaskMaster”.

2.1 GESTÃO DE PROJETOS

Segundo o Project Management Body of Knowledge (PMBOK) “projeto é um esforço temporário empreendido para criar um produto, serviço ou resultado único”. (PMI, 2017, p.4). Os projetos são de natureza temporária, assim, devem possuir data de início e término definidas. Eles se dão por concluídos quando os objetivos são alcançados ou quando o projeto falha por algum fator como, por exemplo, recursos esgotados, mudança de estratégia ou motivos legais. Um projeto que obtenha sucesso no cumprimento dos objetivos gerará, então, um produto, serviço ou resultado único.

A gestão de projetos tem como finalidade garantir a execução eficaz e eficiente desses empreendimentos temporários, com o objetivo de que as organizações alcancem êxito na consecução dos requisitos dos projetos. Esse suporte é concretizado por meio da aplicação de conhecimentos, habilidades, ferramentas e técnicas nas atividades do projeto (PMI, 2017).

Além disso, o PMBOK (PMI, 2017) estabelece cinco grupos de processos de gerenciamento de projetos e dez áreas de conhecimento em gerenciamento de projetos para categorizar e organizar os processos de gerenciamento. Estes grupos de processos estão organizados da seguinte forma:

- **Grupo de processos de iniciação:** Esses processos visam obter a aprovação para novos projetos ou fases de projeto. São responsáveis por auxiliar as partes interessadas na avaliação da viabilidade do projeto e no alinhamento das expectativas quanto aos resultados.

- **Grupo de processos de planejamento:** Este grupo compreende os processos para refinar os objetivos estabelecidos na fase de iniciação e definir o escopo de trabalho detalhado do projeto.
- **Grupo de processos de execução:** Envolve os processos necessários para executar e completar o trabalho definido no planejamento.
- **Grupo de processos de monitoramento e controle:** Este grupo lida com o acompanhamento da evolução do projeto, incluindo a análise do desempenho e a identificação de possíveis modificações necessárias.
- **Grupo de processos de encerramento:** Envolve os processos que formalizam o encerramento adequado do projeto, fase ou contrato.

As áreas de conhecimento identificadas pelo PMBOK (PMI, 2017) categorizam os processos de acordo com os requisitos de conhecimento necessários.

Estas áreas de conhecimento são organizadas da seguinte forma:

- **Gerenciamento da integração do projeto:** Compreende processos que unificam e coordenam as atividades dos grupos de processos de gerenciamento do projeto, sendo aplicados durante todo o ciclo de vida do projeto.
- **Gerenciamento do escopo do projeto:** Trata dos processos que definem o que está incluído ou excluído no projeto, assegurando que todas as atividades necessárias sejam contempladas para atingir os objetivos.
- **Gerenciamento do cronograma do projeto:** Envolve os processos necessários para garantir que o projeto cumpra os prazos estabelecidos.
- **Gerenciamento dos custos do projeto:** Estes processos têm como foco garantir que o projeto seja concluído dentro do orçamento estipulado.
- **Gerenciamento da qualidade do projeto:** Garante que tanto o processo de gerenciamento do projeto quanto o produto atendam aos critérios de qualidade alinhados com as expectativas das partes interessadas.
- **Gerenciamento dos recursos do projeto:** Compreende os processos responsáveis por identificar, adquirir e gerenciar os recursos essenciais para o sucesso do projeto.

- **Gerenciamento das comunicações do projeto:** Envolve processos que endossam a coleta, armazenamento e distribuição de informações cruciais para o projeto.
- **Gerenciamento dos riscos do projeto:** Inclui os processos responsáveis por minimizar ameaças ao desenvolvimento do projeto e maximizar as oportunidades.
- **Gerenciamento das aquisições do projeto:** É a área responsável por obter produtos ou serviços externos à equipe do projeto.
- **Gerenciamento das partes interessadas do projeto:** Compreende os processos que identificam as partes interessadas no projeto e também planejam, gerenciam e monitoram o envolvimento de cada uma delas com o projeto.

2.2 METODOLOGIA ÁGIL

A Metodologia ágil é uma abordagem de gestão de projetos e desenvolvimento de *software* que se caracteriza pela flexibilidade, colaboração e entrega iterativa de valor ao cliente. Uma definição concisa pode ser encontrada nas palavras de Guilherme Miranda (MIRANDA, 2023, p.3)

"a metodologia ágil é uma abordagem para gerenciamento de projetos que se concentra na entrega contínua e incremental de valor para o cliente. Ela foi desenvolvida originalmente para gerenciamento de projetos de software, mas tem sido amplamente aplicada em vários outros campos."

A filosofia dos métodos ágeis foi inicialmente formulada em 2001 por um grupo composto por 17 desenvolvedores de *software* por meio do Manifesto Ágil (MATHIS, 2018). Esse manifesto estabeleceu quatro valores fundamentais:

- **Indivíduos e interações mais que processos e ferramentas:** Destaca a importância da colaboração e comunicação entre as pessoas envolvidas no projeto em vez de depender excessivamente de processos ou ferramentas.
- **Software em funcionamento mais que documentação abrangente:** Prioriza a entrega de *software* funcional e utilizável em detrimento da documentação extensa, colocando o foco na entrega de valor real ao cliente.

- **Colaboração com o cliente mais que negociação de contratos:** Enfatiza a importância da colaboração ativa com o cliente ao longo do processo, em vez de depender de contratos rígidos.
- **Responder a mudanças mais que seguir um plano:** Reconhece que a flexibilidade e a capacidade de se adaptar a mudanças são essenciais para atender às necessidades do cliente e ao ambiente em constante evolução.

Além desses quatro valores, o Manifesto Ágil também inclui doze princípios que fundamentam a abordagem ágil:

- Priorizar a satisfação do cliente por meio da entrega contínua de *software* com valor agregado.
- Receber bem mudanças nos requisitos, mesmo tardiamente no desenvolvimento, em busca de vantagens competitivas.
- Entregar *software* funcionando com frequência, preferencialmente em intervalos curtos de tempo.
- Promover a colaboração diária entre pessoas de negócio e desenvolvedores durante todo o projeto.
- Construir projetos em torno de indivíduos motivados, proporcionando-lhes um ambiente de suporte e confiança.
- Favorecer a comunicação face a face como o meio mais eficaz de transmitir informações dentro da equipe de desenvolvimento.
- Usar *software* funcionando como a medida principal de progresso.
- Promover o desenvolvimento sustentável, permitindo que a equipe mantenha um ritmo constante indefinidamente.
- Dedicar atenção contínua à excelência técnica e ao bom *design* como meio de aumentar a agilidade.
- Valorizar a simplicidade, maximizando a quantidade de trabalho não realizado.
- Reconhecer que as melhores arquiteturas, requisitos e *designs* emergem de equipes auto-organizáveis.
- Em intervalos regulares, a equipe deve refletir sobre sua eficácia e fazer ajustes em seu comportamento.

Outras características notáveis dos processos ágeis, conforme descritas por Bruno Souza (2018), incluem a sua natureza altamente adaptativa, o foco nas pessoas, uma abordagem com baixa formalidade que valoriza a qualidade do trabalho em detrimento da documentação excessiva que poderia atrasar o projeto. Além disso, os processos ágeis se destacam pelo uso de ciclos curtos de interações com prazos geralmente definidos em períodos inferiores a um mês.

Jeff Sutherland (2016), um dos co-criadores do Scrum, enfatiza que a abordagem ágil é altamente eficaz não apenas no desenvolvimento de *software*, mas também na gestão de projetos empresariais. Isso se deve à sua capacidade de se adaptar rapidamente às mudanças nas necessidades do cliente e no ambiente de negócios. Em um mundo empresarial em constante evolução, a capacidade de adaptação é essencial para o sucesso.

2.3 MONITORAMENTO DE RISCOS

De acordo com Roger Pressman e Bruce Maxim (PRESSMAN; MAXIM, 2021, p.1048), renomados especialistas em engenharia de *software*:

“Um risco é um problema em potencial para um projeto de software, e que pode ou não acontecer. Independentemente do resultado, é aconselhável identificá-lo, avaliar sua probabilidade de ocorrência, estimar seu impacto e estabelecer um plano de contingência.”

O monitoramento de riscos é um componente vital da gestão de projetos, uma vez que reconhece a natureza volátil e imprevisível dos riscos. Conforme mencionado por Wazlawick (2019), é fundamental compreender que, ao longo do projeto, novos riscos podem surgir ou a exposição aos riscos previamente identificados pode se agravar. Portanto, é imperativo revisar regularmente a avaliação de riscos.

Esse processo não se limita apenas a acompanhar a implementação das respostas aos riscos, mas também envolve a identificação e análise contínuas de novos riscos. Além disso, requer a elaboração de planos adicionais para lidar com essas novas ameaças, como indicado pelo PMI (2017).

É importante ressaltar que a documentação adequada dos riscos desempenha um papel crucial nesse contexto, como apontado por Wazlawick (2019).

Tanto a equipe de projeto quanto as partes interessadas devem ter acesso a informações atualizadas sobre os riscos e seus níveis de exposição, de modo a tomar decisões informadas.

O processo de monitoramento de riscos se baseia na coleta de informações durante a execução do projeto para avaliar diversos aspectos. Isso inclui verificar mudanças nos status dos riscos, identificar novos riscos que possam surgir e avaliar a eficácia das medidas de mitigação implementadas, conforme orientado pelo PMI (2017).

O PMI (2017) disponibiliza uma variedade de ferramentas e técnicas para facilitar o monitoramento de riscos. Duas abordagens comuns incluem auditorias e reuniões periódicas de monitoramento de riscos. As reuniões servem como fóruns para analisar e documentar a eficácia das respostas aos riscos, além de possibilitar a reavaliação contínua dos riscos existentes e a identificação de novos desafios. A auditoria, por sua vez, pode ser conduzida como parte das reuniões de monitoramento ou separadamente, dependendo da complexidade do projeto e das necessidades específicas de controle.

É importante notar que, ao final do processo de monitoramento de riscos, assim como no planejamento de respostas aos riscos, podem surgir necessidades de mudanças nos custos, cronograma ou em outros componentes do plano de gerenciamento do projeto (PMI, 2017). Portanto, a revisão constante e a adaptação são elementos cruciais para garantir o sucesso na gestão de riscos ao longo do ciclo de vida do projeto.

2.4 SOFTWARE DE GESTÃO DE PROJETO

A utilização de *software* de gestão de projetos desempenha um papel crucial no cenário empresarial contemporâneo. O aumento da complexidade das iniciativas, a necessidade de gerenciar equipes dispersas geograficamente e o foco na otimização operacional levaram as organizações a adotarem soluções tecnológicas para aprimorar a gestão de projetos.

Os projetos atuais frequentemente envolvem uma rede intrincada de tarefas interdependentes, equipes multifuncionais e diversas partes interessadas. A utilização de *software* de gestão de projetos auxilia as organizações a enfrentarem

essa complexidade, oferecendo ferramentas para o planejamento, acompanhamento e controle.

Harold Kerzner (2017), autor de destaque no campo do gerenciamento de projetos, argumenta que o uso de sistemas automatizados de gerenciamento de projetos é crucial para a obtenção de sucesso em projetos complexos, uma vez que eles viabilizam uma monitorização eficiente de recursos, custos e cronogramas.

A utilização de *software* de gestão de projetos possibilita a coleta de dados precisos sobre o progresso do projeto e o desempenho da equipe. Esses dados são fundamentais para embasar decisões informadas e ajustar o curso do projeto conforme necessário. Segundo Alessandra Rech Tortelli (TORTELLI, 2023) "com a digitalização, as empresas podem automatizar tarefas rotineiras, reduzindo o tempo gasto em processos manuais". A automatização de tarefas e processos repetitivos por meio de *software* de gestão de projetos contribui para a melhoria da eficiência e produtividade das equipes em um projeto, contribuindo assim para o sucesso dele.

3 TASKMASTER: DESENVOLVIMENTO DO SOFTWARE

Para aqueles que não têm experiência prévia, começar com um *software* de gestão de projeto pode parecer assustador. Muitas pessoas já se viram diante da incerteza sobre por onde começar ou como abordar tarefas complicadas. Entretanto, existe uma solução que pode simplificar e otimizar consideravelmente essa tarefa, a utilização de um *software* de gestão de projetos ágil, que incorpora *templates* e requer apenas um número reduzido de campos a serem preenchidos, complementado por funcionalidades com notificações automáticas e gerenciáveis tornando o processo menos complexo e mais eficaz.

O "TaskMaster" é um software de gerenciamento de projetos com ênfase em agilidade, desenvolvido para simplificar o planejamento, execução e acompanhamento de tarefas de maneira eficaz e flexível. Ele é especialmente útil quando se lida com projetos complexos e em constante mudança. A abordagem ágil permite a adaptação rápida a novos requisitos e prioridades, resultando em entregas mais rápidas e de maior qualidade.

A utilização de *templates*, que podem ser criados tanto por você para reutilização futura quanto por outras pessoas, simplifica consideravelmente o processo de elaboração de projetos. Você pode começar selecionando um *template* que melhor se adapte ao seu tipo de projeto e, em seguida, personalizá-lo de acordo com as suas necessidades específicas.

Além de iniciar, manter um projeto é igualmente importante. Pensando nisso, o "TaskMaster" facilita o acompanhamento do progresso, a colaboração da equipe e a tomada de decisões informadas. Ele oferece uma visão clara do que está sendo feito, o que precisa ser feito e como o projeto está avançando em relação aos prazos e metas estabelecidas.

3.1 REQUISITOS

Os requisitos de *software* desempenham um papel central na engenharia de *software*, atuando como uma peça fundamental que liga as expectativas dos usuários às soluções tecnológicas. Eles são a base sobre a qual um sistema de *software* é projetado, desenvolvido e avaliado. Para compreender plenamente a

necessidade e a utilização dos requisitos de *software*, é importante explorar a definição e o propósito desses elementos, bem como as práticas recomendadas em sua gestão.

"Ao estabelecer um conjunto de requisitos, você obtém um entendimento de qual será o impacto do software sobre o negócio, o que o cliente quer e como os usuários vão interagir com o software." (PRESSMAN; MAXIM, 2021, p.234). Em outras palavras, eles delineiam as funcionalidades que um sistema deve oferecer, bem como quaisquer restrições ou critérios de desempenho que ele deve atender. Esses requisitos podem ser derivados de diversas fontes, incluindo as necessidades dos usuários, regulamentações governamentais, práticas do setor e objetivos organizacionais.

A necessidade de requisitos de *software* é clara quando se considera o ambiente complexo em que a engenharia de *software* opera. Sem uma especificação clara e detalhada dos requisitos, os desenvolvedores podem enfrentar dificuldades na compreensão do que deve ser construído e como isso deve ser feito.

Além disso, os requisitos atuam como um contrato implícito entre as partes interessadas (PRESSMAN; MAXIM, 2021). Eles fornecem uma base sólida para a comunicação entre equipes de desenvolvimento, clientes e usuários finais, garantindo que todas as partes compartilhem uma compreensão comum do que o sistema deve realizar.

A utilização dos requisitos de *software* abrange todas as fases do ciclo de vida de desenvolvimento de *software*. Durante a elicitação de requisitos, os analistas trabalham em estreita colaboração com as partes interessadas para identificar, documentar e priorizar os requisitos (Sommerville, 2011).

Essa fase é crucial para estabelecer a base sólida sobre a qual o *software* será construído. Os requisitos também desempenham um papel essencial no *design* de *software*, pois fornecem as especificações que os arquitetos e designers usam para criar a estrutura do sistema.

Durante o desenvolvimento e os testes, os requisitos são traduzidos em código e usados como referência para verificar se o sistema está atendendo às expectativas (PRESSMAN; MAXIM, 2021). A gestão de requisitos é uma prática contínua que envolve o controle de mudanças nos requisitos à medida que eles

evoluem ao longo do tempo. Isso garante que o sistema continue a atender às necessidades em constante mudança dos usuários e das organizações.

Em resumo, os requisitos de *software* são elementos vitais na engenharia de *software*, conectando as necessidades dos usuários às soluções tecnológicas. Eles desempenham um papel crucial na compreensão do que deve ser construído e fornecem a base para o sucesso de qualquer projeto de *software*. "Construir um programa de computador elegante que resolve o problema errado não adianta nada para ninguém". (PRESSMAN; MAXIM, 2021, p.234). Portanto, a necessidade e a utilização eficaz dos requisitos são aspectos fundamentais para o desenvolvimento de *software* de qualidade.

Ao abordar as telas criadas no "Taskmaster", irá ser realizado a coleta e apresentação dos requisitos individuais de cada tela, juntamente com suas correspondentes regras de negócio.

3.2 CASOS DE USO

Na engenharia de *software*, a compreensão das necessidades dos usuários é fundamental para o sucesso de um projeto. Uma das abordagens amplamente adotadas para entender, documentar e comunicar essas necessidades é a técnica de casos de uso. Segundo Sommerville (Sommerville, 2011, p.74), os casos de uso "identificam as interações individuais entre o sistema e seus usuários ou outros sistemas".

Os casos de uso são uma ferramenta valiosa que ajuda a capturar os requisitos funcionais de um sistema de *software*, bem como a modelar as interações entre os usuários e o sistema. Essas interações são representadas em termos de ações ou cenários que descrevem como um usuário (ou ator) interage com o sistema para realizar uma tarefa específica. Os casos de uso podem variar em complexidade, desde tarefas simples até fluxos de trabalho complexos que envolvem múltiplos atores e etapas.

A necessidade de casos de uso é evidente quando se considera a complexidade inerente à maioria dos sistemas de *software*. Os requisitos dos sistemas muitas vezes não podem ser compreendidos apenas por meio de uma lista de funcionalidades ou especificações técnicas. Os casos de uso fornecem uma

abordagem orientada pelo usuário para a captura de requisitos, permitindo que as partes interessadas vejam como o sistema será utilizado na prática.

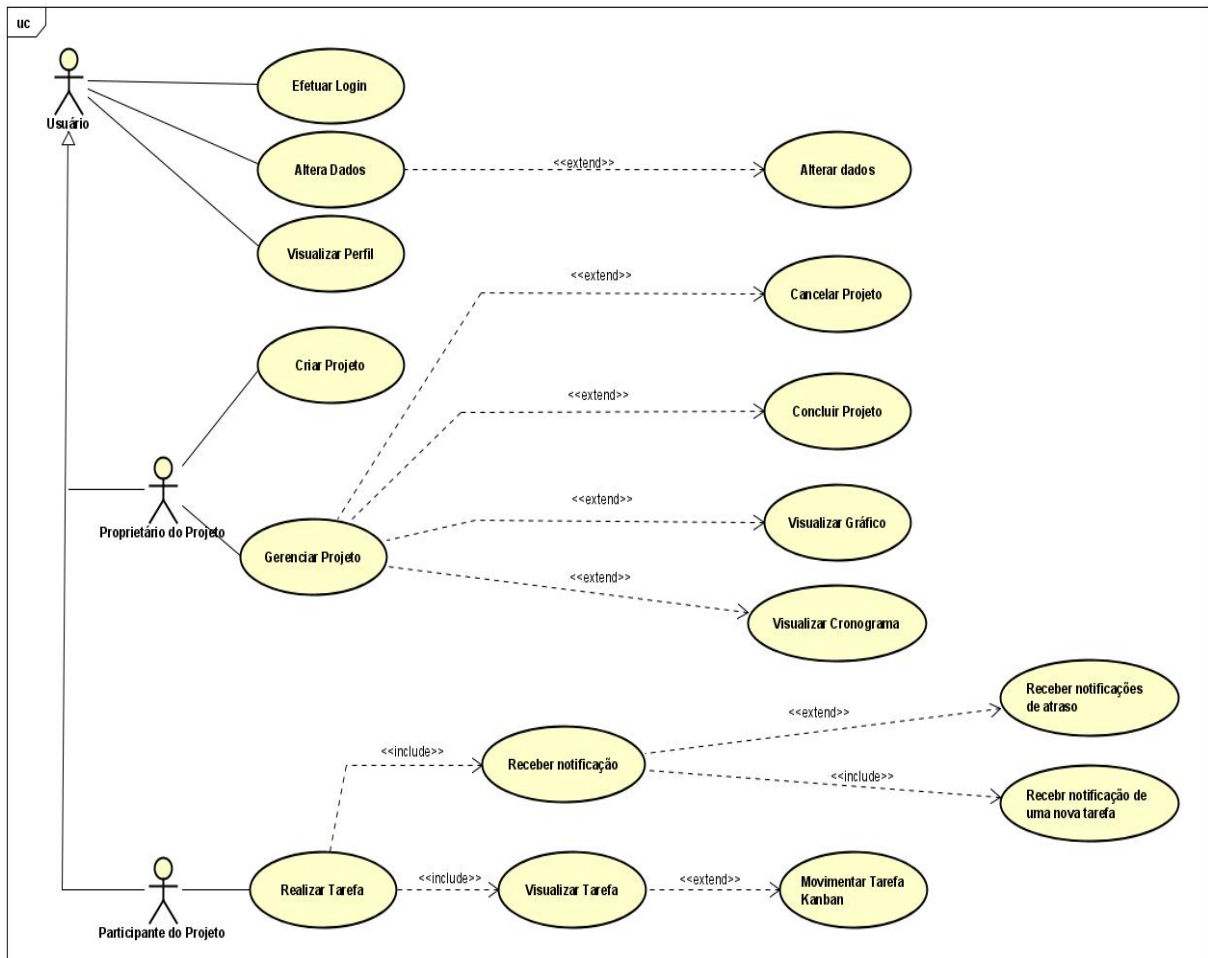
A utilização dos casos de uso na engenharia de *software* é abrangente. Eles desempenham um papel crítico em várias fases do ciclo de vida de desenvolvimento de *software*. Durante a elicitação de requisitos, os casos de uso ajudam os analistas a identificarem e definir as funcionalidades essenciais do sistema, garantindo que os desenvolvedores tenham uma compreensão clara das metas do usuário (VAZQUEZ; SIMÕES, 2016).

Além disso, os casos de uso são usados como base para o *design* de *software*. Os arquitetos de sistemas podem mapear os casos de uso para componentes e interfaces de *software*, criando uma estrutura que atenda às necessidades dos usuários. Durante o desenvolvimento, os casos de uso se tornam um guia para a implementação, permitindo que os desenvolvedores traduzam as interações do usuário em código funcional.

Os casos de uso também têm um papel importante nos testes de *software*, pois fornecem cenários realistas que podem ser usados para validar se o sistema atende aos requisitos do usuário. Além disso, eles são uma ferramenta valiosa para a comunicação entre as equipes de desenvolvimento e as partes interessadas, garantindo que todos tenham uma compreensão clara das funcionalidades e interações esperadas do sistema.

Em resumo, os casos de uso desempenham um papel crítico na engenharia de *software*, permitindo a captura e a comunicação eficaz das necessidades dos usuários. Dessa forma, foi elaborado um caso de uso, no qual será demonstrado os cenários de utilização mais frequentes no “TaskMaster” evidenciado na Figura 1.

Figura 1 - Exemplo de caso de uso no TaskMaster



Fonte: Elaborado pelo o Autor

3.3 LAYOUT

A personalização de produtos e sistemas de acordo com as preferências do usuário tornou-se uma prática cada vez mais relevante na era digital. Com a crescente quantidade de informações e opções disponíveis, a capacidade de oferecer experiências sob medida para os usuários tem se mostrado essencial para aprimorar a intuitividade e a satisfação.

Nesse contexto, o "Taskmaster" exemplifica essa tendência ao aplicar um *layout* "autoajustável". Ao clicar no botão azul, indicado na Figura 3, os usuários tem acesso a um menu de personalização (Figura 2), onde o mesmo pode personalizar o *software* de acordo com suas preferências e necessidades. Essa abordagem permite que o usuário adapte a interface, escolhendo elementos visuais, disposição

de informações e funcionalidades de acordo com seu gosto pessoal, tornando a experiência mais envolvente e intuitiva.

Figura 2 - Menu de personalização



Fonte: Elaborado pelo o Autor

Figura 3 - Botão de ativação do menu de personalização

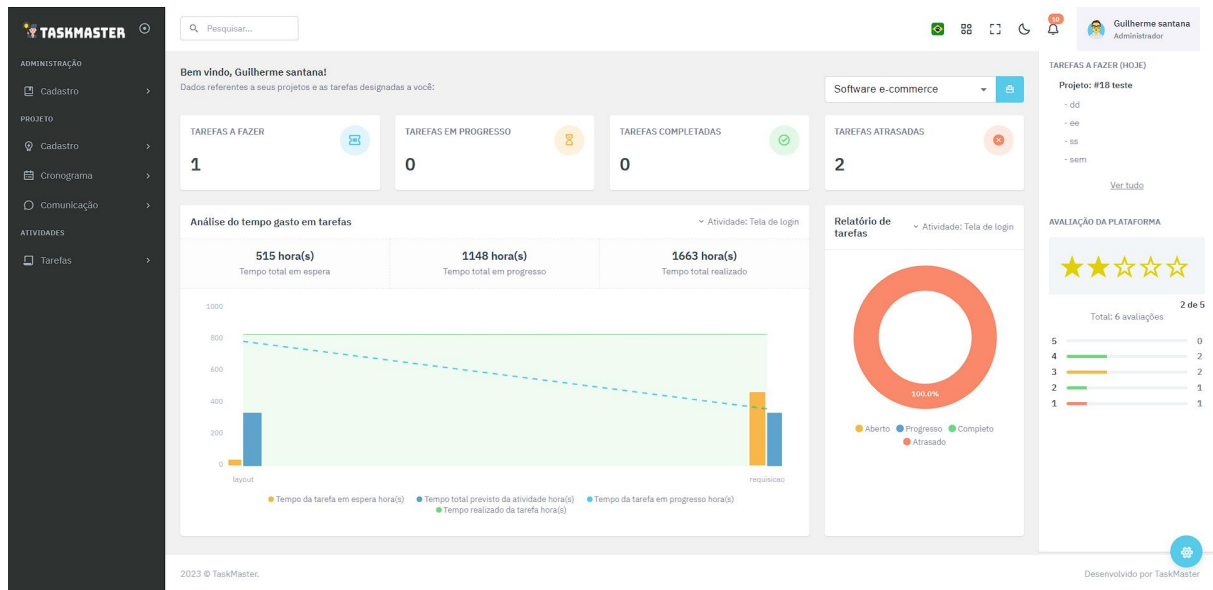


Fonte: Elaborado pelo o Autor

3.3.1 Tema

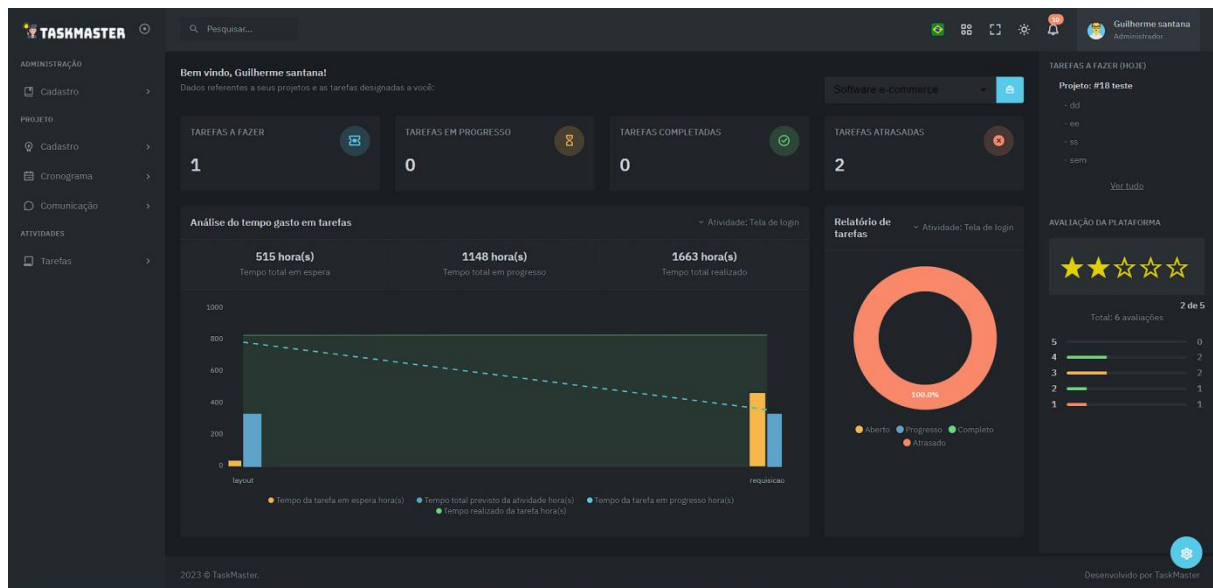
O usuário possui a possibilidade de escolher entre dois temas, podendo ser claro (Figura 4) ou escuro (Figura 5).

Figura 4 - Demonstração do tema claro



Fonte: Elaborado pelo o Autor

Figura 5 - Demonstração do tema escuro

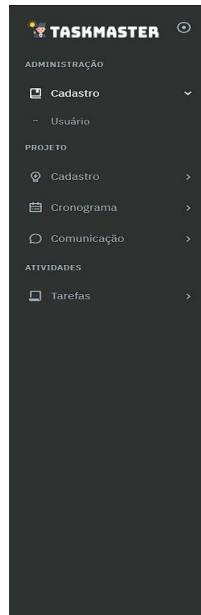


Fonte: Elaborado pelo o Autor

3.3.2 Barra de navegação

A barra de navegação (Figura 6) é a área em que os *links* de acesso as páginas do site estão disponíveis para os usuários. Ela pode ser personalizada por meio do menu de personalização e oferece menus pré-definidos, que variam de acordo com o perfil do usuário.

Figura 6 - Barra de navegação



Fonte: Elaborado pelo o Autor

3.3.3 Barra superior

A barra superior (Figura 7) é o local em que o usuário pode visualizar notificações e mensagens recebidas. A partir dela, é possível acessar diferentes telas por meio de uma barra de pesquisa e também é possível acessar o próprio cadastro a partir do menu.

Figura 7 - Barra superior

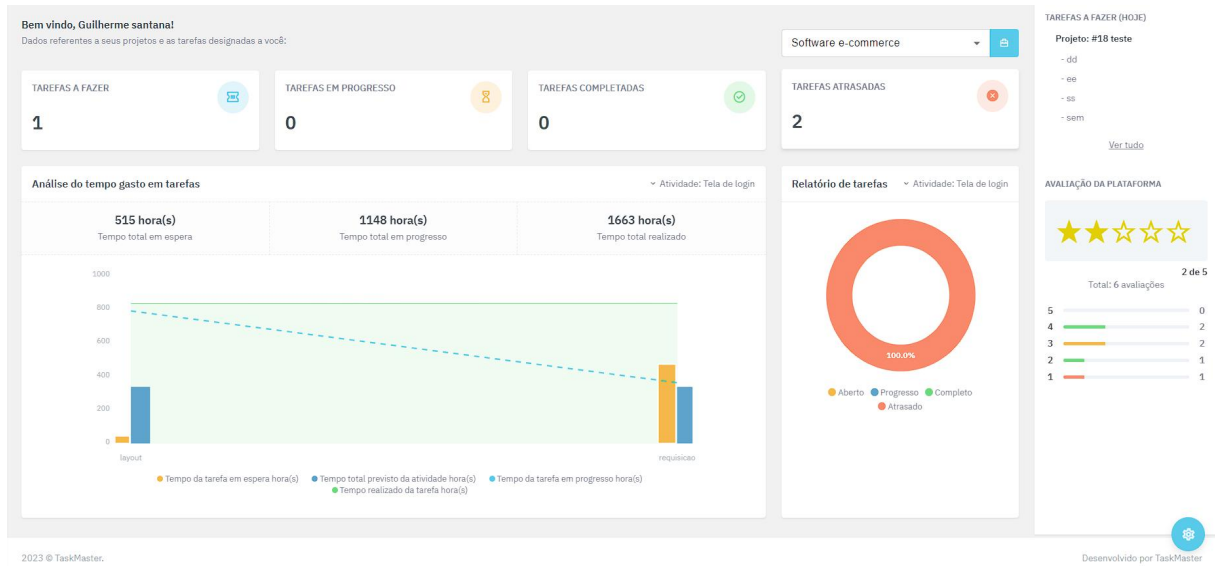


Fonte: Elaborado pelo o Autor

3.4 HOME (DASHBOARD)

A página inicial (Figura 8), ou "home", de um *software* desempenha um papel fundamental na experiência do usuário, pois é muitas vezes a primeira impressão que um usuário tem ao interagir com a aplicação. Ela serve como um ponto de entrada e um centro de informações essencial, transmitindo a identidade e funcionalidade do *software* de forma clara e atrativa. Com isso, será abordado os elementos presentes na página inicial do "TaskMaster".

Figura 8 - Tela Home (Dashboard)



Fonte: Elaborado pelo o Autor

3.4.1 Avaliação

A avaliação de um site web não é apenas uma forma de medir a satisfação do usuário, mas também uma fonte valiosa de informações sobre o desempenho, a usabilidade e as necessidades dos usuários. Por meio de avaliações, os usuários podem expressar suas opiniões, identificar problemas e sugerir melhorias. Levando em conta isso foi realizado uma seção dentro da “Dashboard” (Figura 9) onde é possível realizar e visualizar avaliações já feitas na plataforma.

Figura 9 - Seção avaliação na tela Home

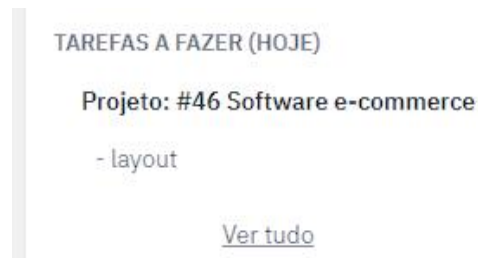


Fonte: Elaborado pelo o Autor

3.4.2 Tarefas

Região da “Dashboard” responsável por exibir as tarefas que estão designadas ao usuário naquele dia como demonstrado na Figura 10.

Figura 10 - Seção tarefas na tela Home



Fonte: Elaborado pelo o Autor

3.4.3 Gráficos

A gestão eficaz de projetos é uma tarefa complexa que requer a coleta, análise e interpretação de uma quantidade significativa de informações. Para os gestores de projetos, a capacidade de tomar decisões informadas desempenha um papel fundamental no sucesso de seus empreendimentos.

Os gráficos não são apenas uma ferramenta de apresentação, eles também são uma ferramenta para realizar cálculos, conforme enfatizado por Alessandro Nicoli, um renomado estatístico que disse que "os gráficos são poderosos mapas que nós guiam através dos números". (MATTOS, 2009, p.141). Isso significa que gráficos bem elaborados podem não apenas representar dados de maneira visualmente eficaz, mas também auxiliar os gerentes de projeto no cálculo rápido de métricas críticas, como a taxa de conclusão de tarefas ou a utilização de recursos.

Com base na importância dos gráficos na gestão de projetos e na tomada de decisões informadas, o “TaskMaster” foi desenvolvido com recursos avançados de visualização de dados e indicadores de tarefas em sua “Dashboard”. Essa abordagem permite aos usuários não apenas acompanhar o progresso do projeto, mas também interagir com as informações de maneira dinâmica e significativa, fornecendo *insights* valiosos para a gestão eficaz de projetos.

Figura 11 - Relatório dos status das tarefas

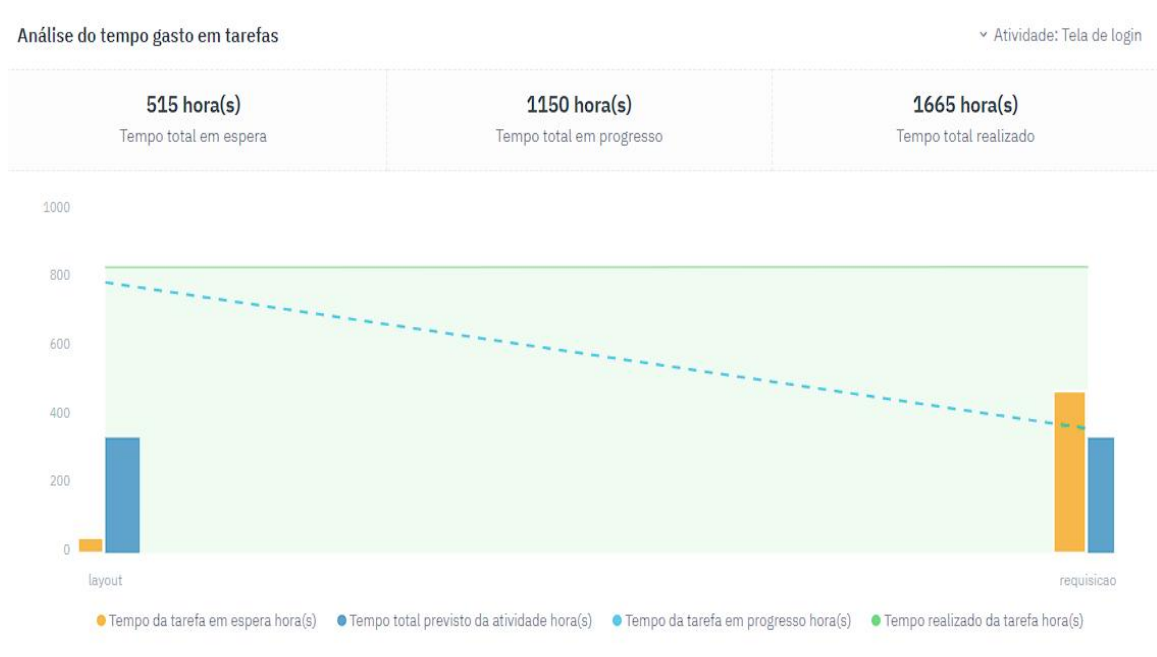


Fonte: Elaborado pelo o Autor

O gráfico de pizza é uma representação visual que ajuda a identificar a distribuição das tarefas em quatro estados diferentes: aberto, atrasado, completo e em progresso. Esse tipo de gráfico permite uma rápida compreensão da proporção de tarefas em cada um desses estados, facilitando a visualização do status geral de um projeto ou conjunto de tarefas conforme evidenciado na Figura 11.

Além do gráfico de pizza, foi elaborado, com a junção de gráfico de linhas e barras, um gráfico para a análise de tempo gasto em tarefas dentro de um atividade.

Figura 12 - Análise gráfica do tempo gasto na execução de tarefas



Fonte: Elaborado pelo o Autor

O gráfico da Figura 12 possui as seguintes representações:

- **Gráficos de barras para avaliar o tempo investido:** As linhas tracejadas azuis oferecem uma representação visual das tarefas em relação ao tempo gasto. Eles permitem que os usuários identifiquem rapidamente qual tarefa consumiu a maior quantidade de tempo para ser concluída. Isso é essencial para a identificação de atividades críticas e a alocação de recursos de maneira mais eficiente.
- **Gráficos de barras para rastrear o tempo de espera:** As barras amarelas destacam o tempo de espera entre as tarefas. Eles permitem que os usuários identifiquem a tarefa que permaneceu mais tempo aguardando para ser realizada. Isso é crucial para eliminar gargalos e melhorar o fluxo de trabalho, garantindo que as tarefas sejam executadas de forma mais eficaz e sem atrasos desnecessários.
- **Gráficos de área para comparar com estimativas prévias:** A área verde do gráfico que destaca o tempo total da tarefa e as barras azuis que evidencia o tempo estimado da atividade oferecem uma maneira eficaz de comparar o tempo real gasto em uma tarefa com a estimativa prévia. Isso permite que os usuários identifiquem as discrepâncias entre o planejamento inicial e a execução real. Ao fazer isso, os gerentes de projeto podem ajustar suas estratégias de planejamento e melhorar a precisão das estimativas futuras.

A combinação desses gráficos possibilita uma análise profunda das operações do projeto. Os usuários podem rapidamente identificar quais tarefas estão desviando do cronograma planejado, quais estão aguardando ação e como as estimativas estão alinhadas com o desempenho real. Essas informações são fundamentais para tomar decisões informadas e ajustar o curso do projeto conforme necessário.

3.4.4 Data grid

A gestão e visualização eficaz de grandes volumes de dados são desafios críticos em uma ampla variedade de campos, desde análise de negócios até o

desenvolvimento de *software*. Uma das ferramentas mais poderosas e amplamente adotadas para enfrentar essa complexidade é o uso de *data grid* (grade de dados).

Os *data grid* são estruturas que dividem uma interface em células organizadas em linhas e colunas, criando uma grade que facilita a organização e exibição de dados. Essa técnica se revela particularmente valiosa quando se lida com grandes volumes de informações, pois permite uma exibição estruturada e a navegação fácil por meio desses dados.

Nesse contexto, tornou-se evidente a necessidade de incorporar a utilização de *data grid* no “TaskMaster”. Sua adaptabilidade, flexibilidade e capacidade de reutilização se destacaram como fatores essenciais na gestão e no desenvolvimento eficaz da aplicação. Os *data grid* do “Taskmaster” contam com diversos filtros na parte superior, uma coluna que permite realizar alguma ação e na parte inferior contam com botões para gerar relatório.

Segue abaixo na Figura 13 um exemplar de *data grid* encontrado no “TaskMaster”.

Figura 13 - Exemplo de data grid

The screenshot displays a data grid with the following structure:

Ações	Cód projeto	Projeto	Data de início	Data de conclusão	Status Projeto	Andamento do projeto
[Icons]	47	Aplicativo mobile	03/11/2023	12/01/2024	Aberto	[Progress bar]
[Icons]	46	Software e-commerce	15/10/2023	28/11/2023	Aberto	[Progress bar]
[Icons]	44	Software e-commerce	04/09/2023	18/10/2023	Atrasado	[Progress bar]

Additional interface elements include: search filters for 'Cód projeto', 'Projeto', 'Data de início', 'Data de conclusão', and 'Status Projeto'; a 'Filtrar...' dropdown; a pagination bar showing '1 - 3 of 3' items; and export buttons for 'Exportar em PDF' and 'Exportar em EXCEL'.

Fonte: Elaborado pelo o Autor

3.5 TELAS DESENVOLVIDAS

Nesta seção, será abordado as diferentes telas criadas no “Taskmaster”, cada uma com seus requisitos e regras de negócios específicos, os quais serão detalhados a seguir. A criação e concepção de algumas telas são fundamentadas em conceitos da gestão de projetos, que serão destacados sua aplicação em cada tela.

3.5.1 Landing page

Uma *landing page*, também conhecida como página de destino, é um elemento fundamental no mundo do marketing digital. É uma página da web projetada de forma estratégica com um propósito específico: converter visitantes em leads, clientes ou direcioná-los para a próxima etapa da jornada do comprador. A *landing page* do “TaskMaster” é organizada em quatro seções distintas: “Home”, “Serviços”, “Páginas” e “Contato”.

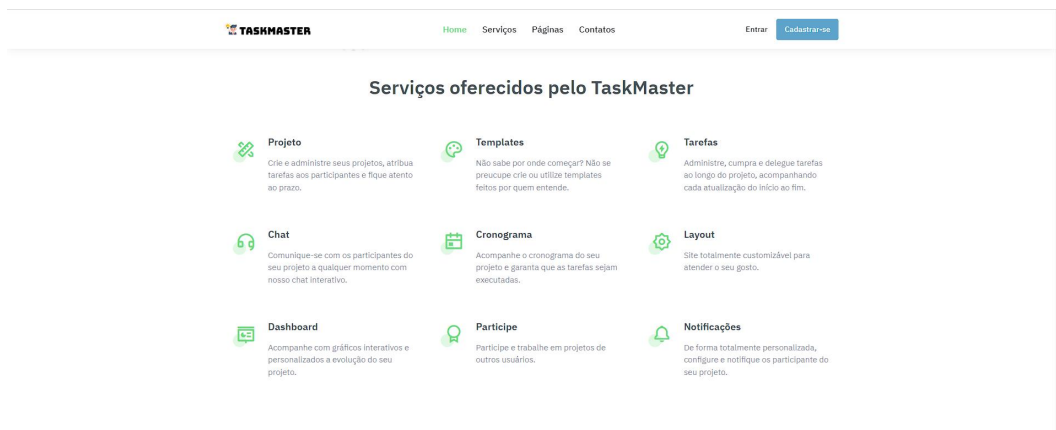
Figura 14 - Seção Home na tela Landing Page



Fonte: Elaborado pelo o Autor

Na seção “Home” (Representada pela Figura 14), os visitantes encontrarão *links* de redirecionamento para diversas partes do *software*, bem como um “carrossel” de imagens que apresenta o *software* de forma visualmente atrativa.

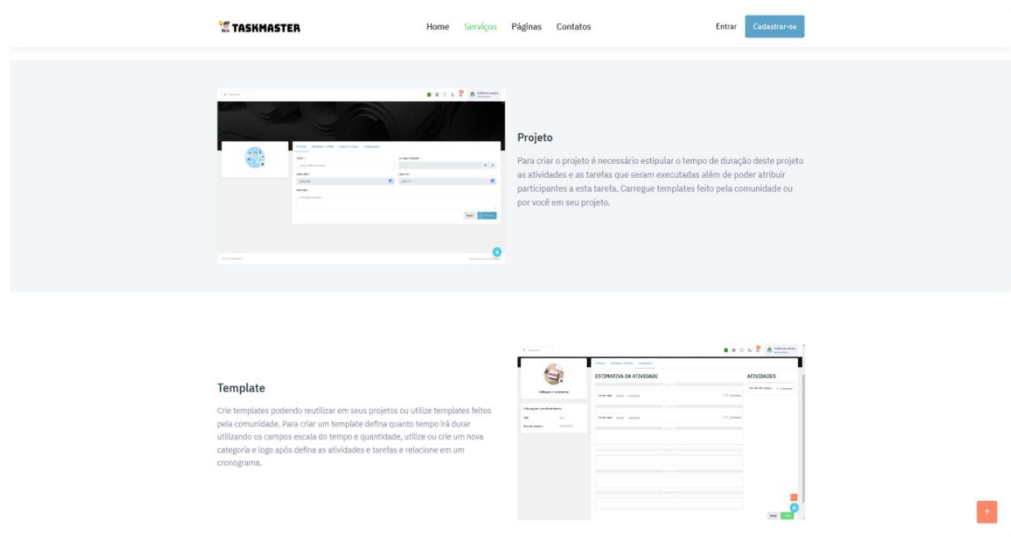
Figura 15 - Seção Serviços na tela Landing Page



Fonte: Elaborado pelo o Autor

A seção “Serviços” (Representada pela Figura 15) oferece uma prévia de todos os recursos e funcionalidades oferecidos pelo “TaskMaster”. Isso permite que os visitantes tenham uma visão geral dos benefícios e capacidades do *software*.

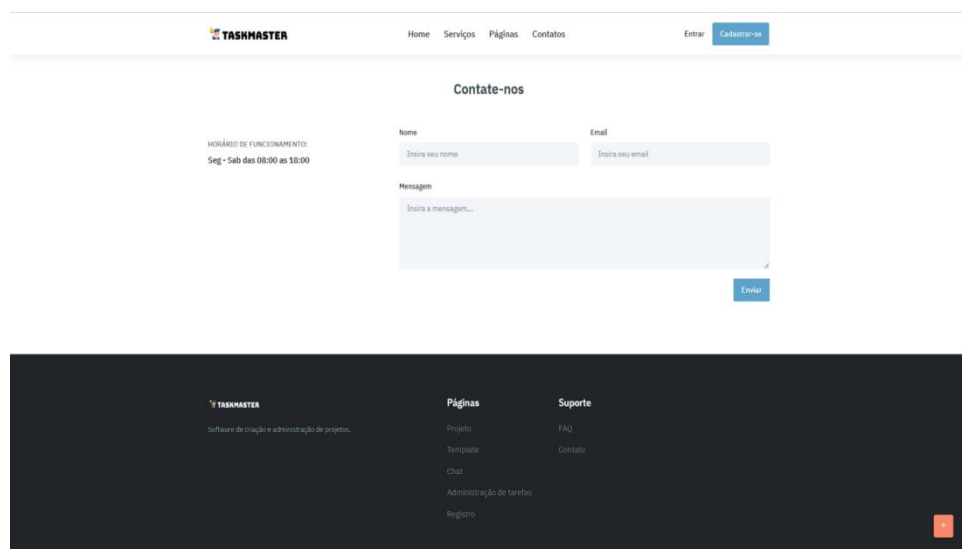
Figura 16 - Seção Páginas na tela Landing page



Fonte: Elaborado pelo o Autor

Na seção “Páginas” (Representada pela Figura 16), os usuários podem explorar e compreender o funcionamento de cada tela do *software*, além de entender como essas telas interagem entre si. Isso oferece uma perspectiva mais detalhada sobre a experiência de uso do “TaskMaster”.

Figura 17 - Seção Contato na tela Landing Page



Fonte: Elaborado pelo o Autor

Na seção “Contato” (Representada pela Figura 17), os visitantes têm a oportunidade de se comunicar com os administradores do site por meio de um formulário. Isso permite que eles enviem mensagens ou dúvidas diretamente para a equipe por trás do “TaskMaster”, facilitando a interação e o suporte aos usuários. O Quadro 1 apresenta todos os requisitos que foram usados para a construção da tela “Landing page”.

Quadro 1 - Requisitos da tela Landing Page

(continua)

Requisitos	Tipo	Importância	Complexidade
[RF001] - A página inicial (<i>landing page</i>) deve disponibilizar informações relacionadas ao <i>software</i> .	Funcional	Alta	Baixa
[RF002] - A página inicial (<i>landing page</i>) deve ser dividida em quatro seções, permitindo que os usuários cliquem em um item para serem redirecionados para a parte correspondente da página.	Funcional	Alta	Média
[RF003] - Na seção inicial (<i>home</i>), deve haver um “carrossel” de imagens do <i>software</i> .	Funcional	Baixa	Média
[RF004] - Na seção “Serviços”, deve ser apresentada uma prévia das funcionalidades oferecidas do <i>software</i> .	Funcional	Alta	Baixa
[RF005] - Na seção “Páginas”, deve haver uma imagem e uma breve descrição que explicam o propósito e a funcionalidade de cada tela para o usuário.	Funcional	Alta	Baixa

Fonte: Elaborado pelo o Autor

Quadro 1 - Requisitos da tela Landing Page

(conclusão)

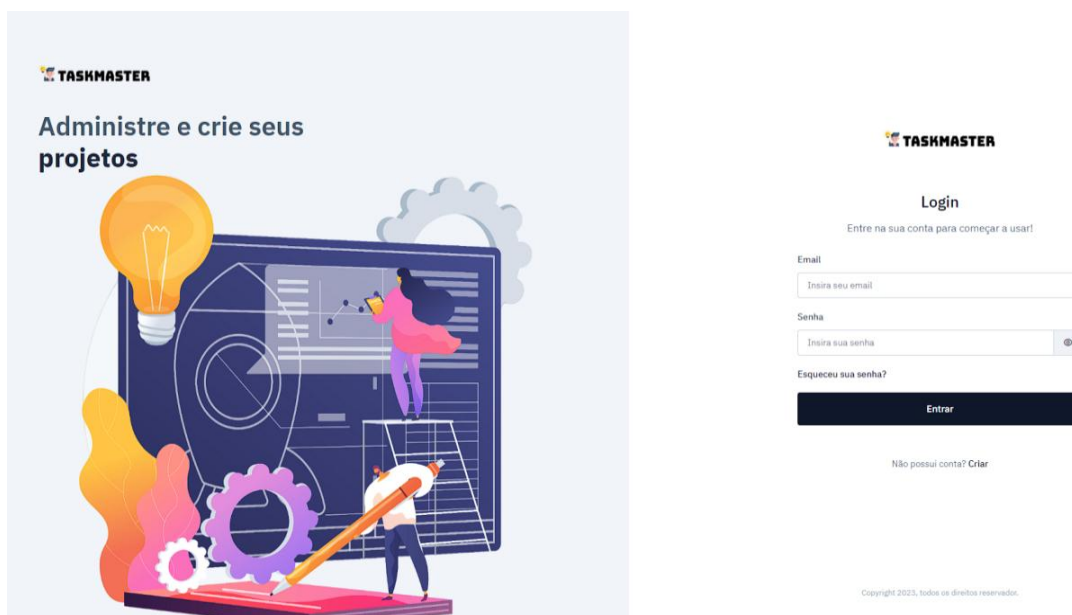
Requisitos	Tipo	Importância	Complexidade
[RF006] - Na seção “Contatos”, deve ser disponibilizado um formulário que permita aos usuários que não possuem cadastro enviar mensagens aos administradores do site.	Funcional	Baixa	Média
[RF007] - Na “Home”, deve haver botões que redirecionem os usuários para a tela “Login”, ou, caso não possuam cadastro, para a tela “Cadastre-se”.	Funcional	Alta	Baixa

Fonte: Elaborado pelo o Autor

3.5.2 Login

A Tela “Login” (Figura 18) é responsável pela a autenticação do usuário.

Figura 18 - Tela Login



Fonte: Elaborado pelo o Autor

A Tela “Login” possui os seguintes requisitos abaixo (Quadro 2):

Quadro 2 - Requisitos da tela Login

Requisitos	Tipo	Importância	Complexidade
[RF001] - A tela “Login” tem a responsabilidade de verificar a identidade do usuário e conceder acesso ao sistema após autenticação bem-sucedida.	Funcional	Alta	Baixa
[RF002] - A autenticação na tela de “Login” deve ser realizada com base nas informações inseridas nos campos de email e senha.	Funcional	Alta	Baixa
[RF003] - A tela “Login” deve direcionar o usuário para o painel principal (<i>home</i>) após a autenticação ser válida.	Funcional	Alta	Baixa
[RF004] - Após o usuário realizar o <i>login</i> , é necessário armazenar o <i>token</i> no “Local Storage” (Armazenamento interno <i>web</i>) para permitir o seu uso em requisições futuras, juntamente com os dados do usuário que está autenticado.	Não Funcional	Alta	Média

Fonte: Elaborado pelo o Autor

O Quadro 3 irá representar todas as regras de negócio que a tela “Login” possui para seu funcionamento:

Quadro 3 - Regra de negócio da tela Login

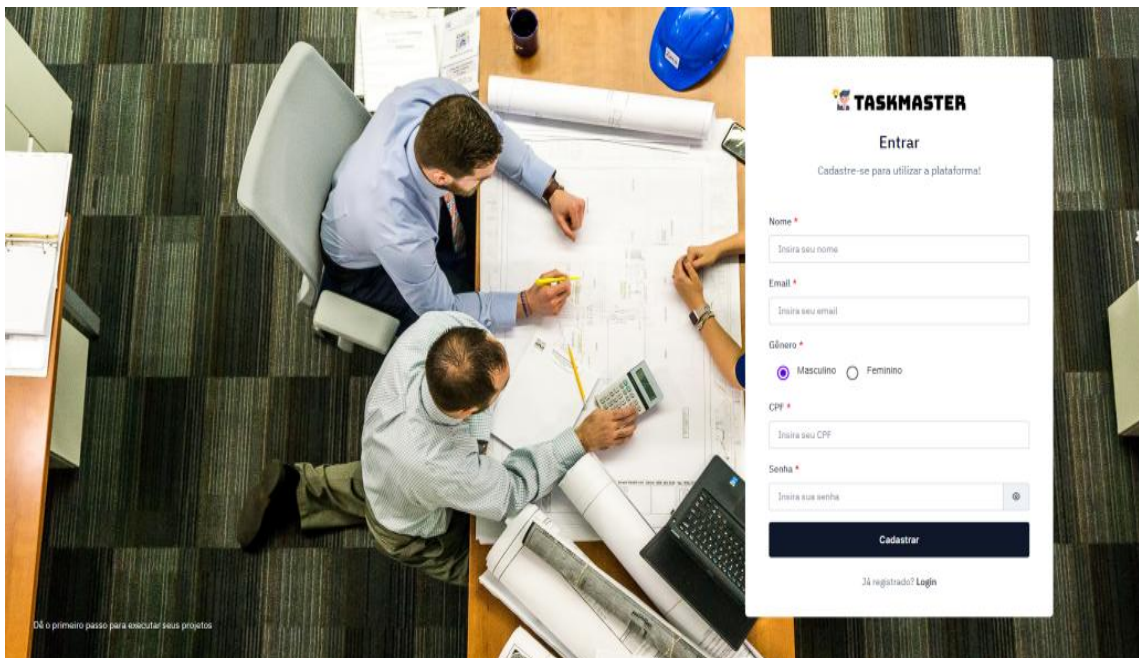
Regra de negócio	Tipo
[RN001] - Realizar o "login"	O redirecionamento para a <i>home</i> ocorrerá apenas se as informações de email e senha fornecidas estiverem registradas no sistema. Se autenticado com sucesso, será exibida a seguinte mensagem: "Seja bem-vindo de volta: [Nome do usuário]". Caso a autenticação não seja bem-sucedida, será apresentada a mensagem: "Usuário ou senha inválido".

Fonte: Elaborado pelo o Autor

3.5.3 Cadastre-se

Tela responsável pelo cadastro inicial do usuário na plataforma (Figura 19).

Figura 19 - Tela Cadastre-se



Fonte: Elaborado pelo o Autor

O cadastro inicial do usuário na plataforma tem os seguintes requisitos:

Quadro 4 - Requisitos da tela Cadastre-se

Requisitos	Tipo	Importância	Complexidade
[RF001] - A tela “Cadastre-se” deve proporcionar ao usuário a capacidade de criar sua conta na plataforma.	Funcional	Alta	Alta
[RF002] - Se o cadastro for válido, o usuário deve ser direcionado para a tela “Home”.	Funcional	Alta	Baixa

Fonte: Elaborado pelo o Autor

Quadro 5 - Regra de negócio da tela Cadastre-se

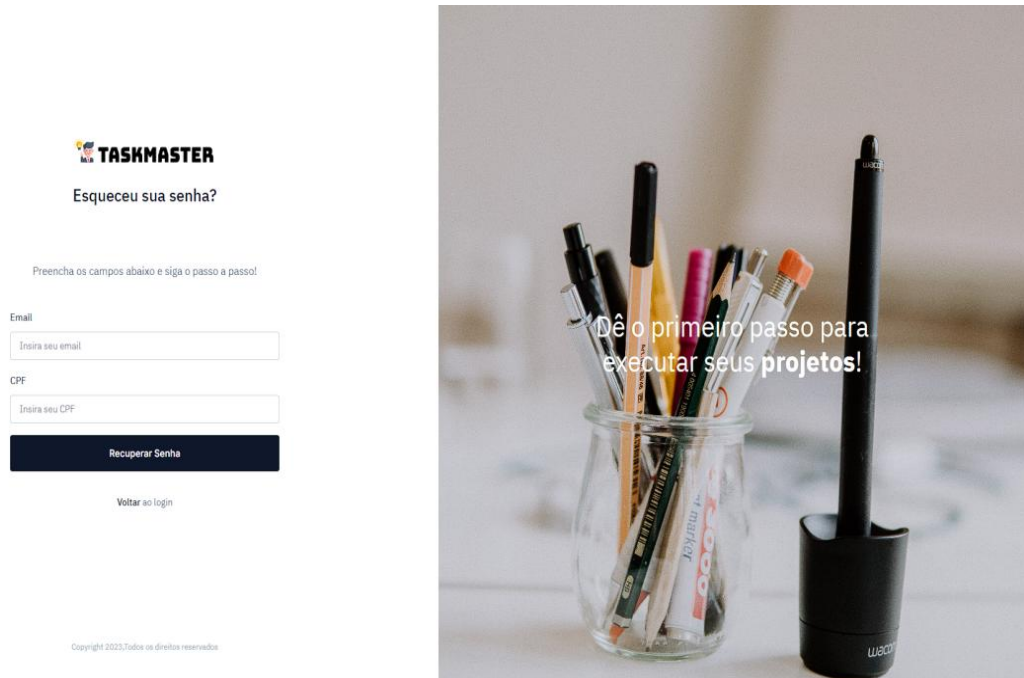
Regra de negócio	Tipo
[RN001] - Cadastrar-se	Durante o processo de cadastro inicial, é essencial validar se o email fornecido já está associado a outro usuário na plataforma. Se for o caso, deve ser exibida a seguinte mensagem: "Email já vinculado a outro usuário" e o cadastro não deve ser concluído.
[RN002] - Cadastrar-se	Somente cadastros com Cadastro de Pessoas Físicas (CPF) válidos e senhas contendo no mínimo 8 caracteres, incluindo letras maiúsculas, minúsculas e pelo menos um caractere especial, serão aceitos.

Fonte: Elaborado pelo o Autor

3.5.4 Recupere sua senha

Tela responsável pela recuperação da senha do usuário, caso o mesmo esqueça, a tela esta sendo representada pela Figura 20.

Figura 20 - Tela Recupere sua senha



Fonte: Elaborado pelo o Autor

A tela para a recuperação da senha possui os seguintes requisitos:

Quadro 6 - Requisitos da tela Recupere sua senha

(continua)

Requisitos	Tipo	Importância	Complexidade
[RF001] - A tela de "Recuperação de Senha" deve permitir que o usuário cadastre uma nova senha após inserir corretamente o código recebido por email.	Funcional	Alta	Alta
[RF002] - Deve ser enviado um código randômico para o email do usuário depois que ele fornece seu CPF e o email associado ao seu cadastro.	Funcional	Alta	Alta
[RF003] - É necessário validar se o código foi inserido corretamente.	Funcional	Alta	Baixa

Fonte: Elaborado pelo o Autor

Quadro 6 - Requisitos da tela Recupere sua senha

(conclusão)

Requisitos	Tipo	Importância	Complexidade
[RF004] - O usuário deverá aguardar um minuto antes de solicitar o envio de um novo código para o seu email.	Funcional	Média	Média

Fonte: Elaborado pelo o Autor

A tela “Recupere sua senha” possui as seguintes regras de negócio:

Quadro 7 - Regra de negócio da tela Recupere sua senha

Regra de negócio	Tipo
[RN001] - Recuperar senha	<p>Se o usuário inserir o código incorretamente três vezes, a plataforma deverá informá-lo com a seguinte mensagem: "Número máximo de tentativas alcançado, tente novamente mais tarde!"</p> <p>O usuário deverá aguardar uma hora antes de fazer uma nova tentativa de recuperação de senha.</p>

Fonte: Elaborado pelo o Autor

3.5.5 Usuário

A tela “Usuário” foi desenvolvida com o propósito de permitir a atualização e inclusão de informações adicionais relacionadas aos perfis dos usuários. Ela está acessível a todos os usuários por meio da barra superior, na seção “Perfil”. No entanto, vale ressaltar que somente os usuários com privilégios de administrador têm permissão para visualizar o *data grid* da tela Usuário (Figura 21), bem como para exibir e editar os detalhes de qualquer usuário no sistema.

Quadro 8 - Requisitos da tela Usuário

(continua)

Requisitos	Tipo	Importância	Complexidade
[RF001] - A tela "Usuário" deve possibilitar ao usuário adicionar ou editar as informações que foram previamente cadastradas.	Funcional	Alta	Alta
[RF002] - A tela "Usuário" deve incluir uma barra de progresso que indica o quão completo está o seu cadastro.	Funcional	Baixa	Baixa
[RF003] - Se uma nova senha for informada e estiver em conformidade com os requisitos, a senha do usuário deve ser atualizada para a nova senha fornecida.	Funcional	Média	Baixa
[RF004] - Ao consultar o Código de Endereçamento Postal (CEP), a plataforma deve realizar uma consulta à Application Programming Interface (API) para verificar se o CEP é válido. Se for válido, deve preencher automaticamente os campos: Estado, Cidade, Bairro e Rua com as informações retornadas pela API.	Funcional	Alta	Alta
[RF005] - Deve se utilizar a API "ViaCep" para realizar a consulta do CEP.	Não Funcional	Alta	Média
[RF006] - Para acessar o seu cadastro, deve ser adicionado um item no menu da barra superior com o nome "Perfil".	Funcional	Alta	Baixa

Fonte: Elaborado pelo o Autor

Quadro 8 - Requisitos da tela Usuário

(conclusão)

Requisitos	Tipo	Importância	Complexidade
[RF007] - Apenas os usuários com perfil de administrador devem ter acesso ao menu de usuário pela barra de navegação.	Funcional	Alta	Baixa
[RF008] - O <i>data grid</i> da tela “Usuário” deve oferecer a opção de tanto cadastrar um novo usuário quanto editar um usuário previamente cadastrado.	Funcional	Alta	Baixa
[RF009] - Somente os usuários com perfil de administrador têm permissão para editar ou criar outros usuários com perfil de administrador.	Funcional	Alta	Baixa
[RF010] - O <i>data grid</i> da tela “Usuário” deve disponibilizar a opção que gere relatório dos usuários criados na plataforma em formato PDF ou XLS.	Funcional	Baixa	Média
[RF010] - O tela “Usuário” deve dispor das seguintes abas: <ul style="list-style-type: none"> ● Principal (Figura 21) ● Endereço (Figura 22) ● Configurações (Figura 23) 	Funcional	Média	Baixa

Fonte: Elaborado pelo o Autor

A tela “Usuário” possui as seguintes regras de negócio:

Quadro 9 - Regra de negócio da tela Usuário

Regra de negócio	Tipo
[RN001] - Edição do usuário	<p>Se o usuário optar por alterar o seu email, será necessário validar se o novo email não está sendo utilizado por outro usuário.</p> <p>Se o novo email já estiver em uso por outro usuário, a plataforma deve notificar com a mensagem: "Email já vinculado a outro usuário".</p>

Fonte: Elaborado pelo o Autor

Figura 21 - Data grid da tela Usuário

Ações	Cód	Nome	Email	CPF	Data de nascimento	Perfil
<input type="checkbox"/>	3	teste2	teste2@gmail.com	288.356.498-85		Administrador
<input type="checkbox"/>	2	teste	teste@gmail.com	288.356.498-85	01/01/1970	Administrador
<input type="checkbox"/>	1	Guilherme santana	guilhermesantana199357@hotmail.com	470.356.278-20	01/01/1970	Administrador

Fonte: Elaborado pelo o Autor

Figura 22 - Aba Principal na tela Usuário

Principal	Endereço	Configurações
<p>Nome *</p> <input type="text" value="Guilherme santana"/>	<p>Email *</p> <input type="text" value="guilhermesantana199357@hotmail.com"/>	
<p>Nome do pai</p> <input type="text" value="Daniel Santana"/>	<p>Nome da mãe *</p> <input type="text" value="Carla Costa"/>	
<p>CPF *</p> <input type="text" value="470.356.278-20"/>	<p>Telefone</p> <input type="text" value="(11) 96278-1668"/>	
<p>Data de nascimento *</p> <input type="text" value="01/01/1970"/>	<p>Gênero *</p> <input checked="" type="radio"/> Masculino <input type="radio"/> Feminino	
<p>Habilidades</p> <input type="text"/>		
<p>Sobre Mim</p> <input type="text" value="algo"/>		

Fonte: Elaborado pelo o Autor

Figura 23 - Aba Endereço na tela Usuário

The screenshot shows the 'Endereço' tab of the user profile. On the left, there is a user profile card for 'Guilherme santana' (Administrator) with a 'Complete seu perfil' progress bar. The main content area has three tabs: 'Principal', 'Endereço', and 'Configurações'. The 'Endereço' tab is active, displaying a form with the following fields: 'Cep' (888044-00), 'País' (Brasil), 'Estado' (SC), 'Cidade' (Criciúma), 'Bairro' (Pinheirinho), 'Rua' (Rua Nações Unidas), and 'Número' (75). There are 'Voltar' and 'Próximo' buttons at the bottom right of the form. The footer includes '2023 © TaskMaster' and 'Desenvolvido por TaskMaster'.

Fonte: Elaborado pelo o Autor

Figura 24 - Aba Configurações na tela Usuário

The screenshot shows the 'Configurações' tab of the user profile. On the left, there is a user profile card for 'Guilherme santana' (Administrator) with a 'Complete seu perfil' progress bar. The main content area has three tabs: 'Principal', 'Endereço', and 'Configurações'. The 'Configurações' tab is active, displaying a form with the following fields: 'Perfil do Usuário' (Tipo de perfil: Usuário com perfil administrador, with a toggle switch), and 'Autenticação do Usuário' (Nova Senha: Insira sua senha). There are 'Voltar' and 'Editar' buttons at the bottom right of the form. The footer includes '2023 © TaskMaster' and 'Desenvolvido por TaskMaster'.

Fonte: Elaborado pelo o Autor



3.5.6 Template

Os *templates* são frequentemente empregados para agilizar a criação e o acompanhamento de projetos complexos. Eles permitem que os gerentes de projeto

comecem com uma estrutura predefinida que incorpora as melhores práticas do setor. Isso não apenas economiza tempo, mas também garante que os projetos sejam planejados e executados de acordo com padrões reconhecidos. Como observado por Paul Tinnirello, "Os templates podem fornecer orientações muito explícitas sobre quais processos são para ser seguido. Isso permite que uma organização, que é nova no gerenciamento de projetos, implemente rapidamente um processos padrão" (TINNIRELLO, 2001, p.453, tradução nossa).

A tela "Template" é onde é cadastrado parâmetros pré-configurados para a utilização em um projeto. O *data grid* da tela "Template" (Figura 25) é acessível a todos os usuários do sistema e exibe informações essenciais, incluindo o código, título, duração estimada, categoria e autor do *template*.

Figura 25 - Data grid da tela Template

Ações	Cód template	Template	Duração estimada template do projeto	Categoria do template	Autor do template
 	22	Software e-commerce	6 Semanas	Desenvolvimento	Guilherme santana

Fonte: Elaborado pelo o Autor

Dentro deste *data grid*, três ações podem ser executadas:

- **Edição e Exclusão:** Se você for o autor do *template* ou possuir perfil administrador, terá a capacidade de editar ou excluir o *template*. Essa funcionalidade permite ajustar ou remover modelos de acordo com as necessidades em constante evolução.

- **Visualização:** Para todos os usuários, independentemente do perfil, é possível visualizar os *templates*. No entanto, a visualização não inclui a capacidade de realizar edições. Isso serve como uma ferramenta valiosa para revisar e obter informações sobre os modelos disponíveis no sistema.

A tela “Template” é dividida em três abas: “Principal”, “Atividade / Tarefas” e “Cronograma”.

Figura 26 - Aba Principal na tela Template

A imagem mostra a interface de usuário para a criação de um template. O formulário é dividido em seções:

- Título:** Campo de texto com o valor "Software e-commerce".
- Descrição:** Campo de texto com o valor "Software e-commerce".
- PERÍODO ESTIMADO:** Seção com "Escala de tempo" (Semanas) e "Quantidade" (6).
- CATEGORIA:** Seção com "Categoria" (Desenvolvimento).

Na parte inferior direita do formulário, há botões "Voltar" e "Próximo".

Fonte: Elaborado pelo o Autor

Na aba “Principal” (Figura 26), você configura os parâmetros iniciais do *template*. Aqui, você especifica o título e a descrição do *template*, juntamente com a categoria à qual ele pertence. Você tem a flexibilidade de escolher entre categorias predefinidas ou criar uma nova categoria, caso a opção "Outros" seja mais apropriada para o *template* em questão. Na seção "Período de Tempo", é onde você define a duração estimada do projeto, podendo escolher entre dias ou semanas como unidade de tempo. Ao especificar a duração necessária, um cronograma será automaticamente gerado na aba "Cronograma". Isso permitirá que você configure o período durante o qual as atividades do projeto serão realizadas, proporcionando uma visão clara e organizada do fluxo de trabalho planejado.

Figura 27- Aba Atividades / Tarefas na tela Template

Fonte: Elaborado pelo o Autor

Na aba “Atividade /Tarefas” (Figura 27), você efetua o cadastro das atividades e das tarefas correspondentes. Cada atividade é composta por um conjunto de tarefas que estão intimamente relacionadas e serão posteriormente ajustadas no cronograma, na próxima aba.

Figura 28 - Aba Cronograma na tela Template

Fonte: Elaborado pelo o Autor

Na aba “Cronograma” (Figura 28), é onde é construído o cronograma das atividades durante o período estabelecido. Na área de "Atividades", você encontrará as atividades que foram cadastradas, juntamente com a sua duração prevista. Na seção "Estimativa da Atividade", é onde você define os dias ou semanas previstos para a execução de cada atividade. Aqui, é possível arrastar as atividades para a semana ou dia de início desejado. Ao repetir esse processo com as demais atividades, você construirá o cronograma detalhado do projeto, estabelecendo as datas específicas em que cada atividade será realizada. Essa abordagem proporciona uma visão clara e organizada do fluxo de trabalho ao longo do período determinado.

A tela “Template” possui os seguintes requisitos:

Quadro 10 - Requisitos da tela Template

(continua)

Requisitos	Tipo	Importância	Complexidade
[RF001] - A tela “Template” deve possibilitar o cadastro de informações prévias de um projeto para serem usadas como um modelo.	Funcional	Alta	Alta
[RF002] - O <i>data grid</i> da tela “Template” deve oferecer a opção de visualizar o <i>template</i> se o usuário não for o proprietário do mesmo. No entanto, se o usuário for o proprietário do <i>template</i> , ele terá as opções de editar ou excluir.	Funcional	Alta	Baixa
[RF003] - A tela “Template” deve ser organizada em três abas distintas: <ul style="list-style-type: none"> ● Principal ● Atividades / Tarefas ● Cronograma 	Funcional	Alta	Baixa

Fonte: Elaborado pelo o Autor

Quadro 10 - Requisitos da tela Template

(continuação)

Requisitos	Tipo	Importância	Complexidade
<p>[RF004] - Na aba “Principal”, devem ser incluídas as informações pertinentes ao <i>template</i>, tais como:</p> <ul style="list-style-type: none"> ● Foto ● Título ● Descrição ● Escala de tempo (Dia ou Semana) ● Categoria 	Funcional	Alta	Média
<p>[RF005] - Na aba “Principal”, deve haver um campo para especificar a escala de tempo (dia ou semana) e a quantidade desejada. Quando esses detalhes forem especificados, a tela deve ser capaz de gerar automaticamente um cronograma na aba “Cronograma” com base nas informações fornecidas.</p>	Funcional	Alta	Alta
<p>[RF006] - Na aba “Principal”, deve haver um campo categoria, e se o usuário não encontrar uma categoria adequada, ele deverá ter a opção de selecionar "Outros" no campo e então cadastrar uma nova categoria.</p>	Funcional	Alta	Média
<p>[RF007] - Na aba “Atividades / Tarefas”, deve ser possível cadastrar, editar ou excluir uma atividade. Cada atividade deve incluir pelo menos uma tarefa, na qual será necessário definir uma data de início e uma data de fim.</p>	Funcional	Alta	Alta

Fonte: Elaborado pelo o Autor

Quadro 10 - Requisitos da tela Template

(conclusão)

Requisitos	Tipo	Importância	Complexidade
[RF008] - Na aba “Atividades / Tarefas”, deve ser possível aplicar filtros para encontrar as atividades e tarefas cadastradas de forma mais eficiente.	Funcional	Alta	Alta
[RF009] - Ao clicar em uma tarefa, uma <i>modal</i> (componente do bootstrap) deve ser aberta com um formulário, permitindo que o usuário especifique informações adicionais para aquela tarefa, tais como prioridade, <i>tags</i> e descrição.	Funcional	Alta	Média
[RF010] - Na aba “Cronograma”, haverá duas colunas, uma para as atividades e outra para os dias ou semanas, dependendo da escala de tempo escolhida. O usuário poderá arrastar as atividades para o período de início desejado, e o sistema irá automaticamente preencher e criar um cronograma com base nas escolhas do usuário.	Funcional	Alta	Alta

Fonte: Elaborado pelo o Autor

A tela “Template” possui as seguintes regras de negócio:

Quadro 11 - Regra de negócio da tela Template

Regra de negócio	Tipo
[RN001] - Criação do cronograma	Para concluir o cadastro do <i>template</i> , é necessário que todas as atividades cadastradas estejam relacionadas no cronograma.
[RN002] - Criação de atividades e tarefas	Em um <i>template</i> , é obrigatório ter pelo menos uma atividade que contém pelo menos uma tarefa.

Fonte: Elaborado pelo o Autor

3.5.7 Projeto

A tela “Projeto” é o ponto de partida para iniciar um novo projeto. Nesta interface, os usuários têm a opção de carregar modelos previamente registrados (*templates*) ou criar um modelo personalizado que atenda às suas necessidades específicas.

No *data grid* da tela, os usuários têm acesso a uma visão abrangente de todos os projetos registrados no *software* (caso seja administrador) ou apenas aos seus projetos (caso seja um usuário comum). O *data grid* da tela “Projeto” (Figura 29) exibe detalhes cruciais, incluindo o status atual do projeto, seu progresso representado por uma barra de avanço, a data de início e término, o nome do projeto e um código de identificação único.

Figura 29 - Data grid da tela Projeto

Ações	Cód projeto	Projeto	Data de início	Data de conclusão	Status Projeto	Andamento do projeto
[Icons]	45	Software e-commerce	15/10/2023	28/11/2023	Aberto	[Progress Bar]
[Icons]	44	Software e-commerce	04/09/2023	16/10/2023	Atrasado	[Progress Bar]

Fonte: Elaborado pelo o Autor

Dentro deste *data grid*, os usuários têm a capacidade de atualizar o status do projeto de acordo com suas necessidades. Se o projeto estiver em andamento, as opções disponíveis incluem concluir, cancelar ou excluir o projeto. No entanto, se o projeto não estiver mais em andamento e possuir um status diferente de "aberto", o *data grid* permite excluir o projeto ou apenas a visualização do projeto, funcionando como um registro histórico valioso. Quando um projeto está em andamento e com status "aberto", os usuários também têm a opção de editá-lo conforme necessário, permitindo ajustes e atualizações em tempo real.

Na tela "Projeto" é onde fica o cadastro dos detalhes dos projetos, esta tela é dividida em 4 abas: "Principal", "Atividades/Tarefas", "Equipe/Funções" e "Notificações".

Figura 30 - Aba Principal na tela Projeto

A imagem mostra a interface de usuário para o cadastro de um projeto. No topo, há uma barra de busca com o texto "Pesquisar...". À direita, há ícones para WhatsApp, uma grade, uma lupa, um relógio e uma notificação, além do nome de usuário "Guilherme santana Administrador".

O formulário principal está dividido em abas: "Principal", "Atividades / Tarefas", "Equipe / Funções" e "Configurações". A aba "Principal" está selecionada e contém os seguintes campos:

- Título ***: Um campo de texto com o placeholder "Insira o título do projeto...".
- Carregar Template**: Um campo de seleção com um ícone de lupa.
- Data Início ***: Um campo de data com o placeholder "Data início" e um ícone de calendário.
- Data Fim ***: Um campo de data com o placeholder "Data fim" e um ícone de calendário.
- Descrição**: Um campo de texto com o placeholder "Descrição do projeto...".

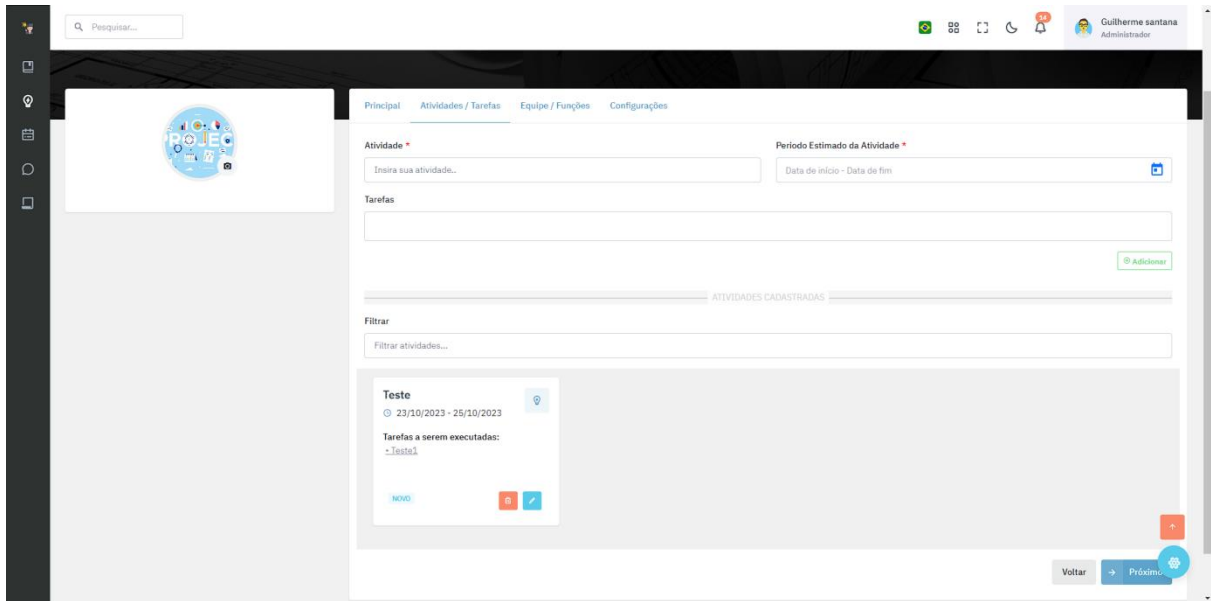
Na base do formulário, há dois botões: "Voltar" e "Próximo".

Na barra inferior esquerda, há o texto "2023 © TaskMaster". Na barra inferior direita, há o texto "Desenvolvido por TaskMaster" e um ícone de engrenagem.

Fonte: Elaborado pelo o Autor

Na aba "Principal" (Figura 30), é onde você pode registrar as informações-chave do seu projeto. Aqui, é possível fornecer o título do projeto, uma descrição detalhada, bem como as datas de início e término previstas. Se desejar utilizar um *template* previamente cadastrado, basta selecioná-lo no campo "Carregar Template". Uma vez selecionado, a tela será automaticamente preenchida com todas as informações contidas no *template*, adaptando-as ao projeto, incluindo o período estimado e as tarefas associadas.

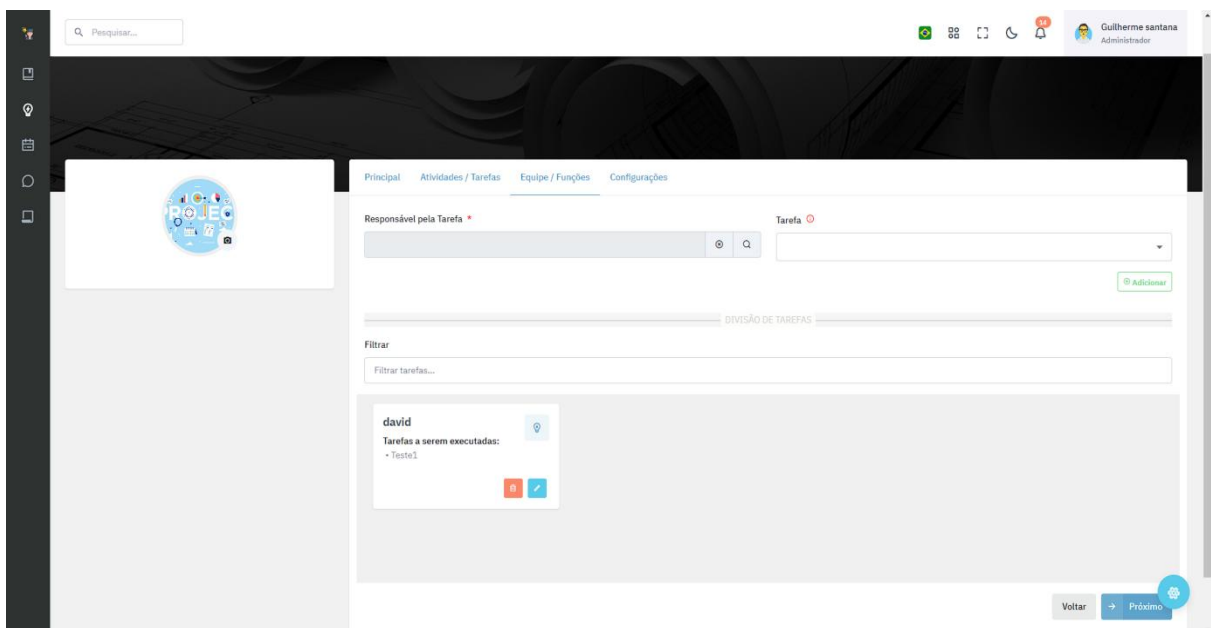
Figura 31 - Aba Atividades / Tarefas na tela Projeto



Fonte: Elaborado pelo o Autor

Na aba “Atividades/Tarefas” (Figura 31), você encontrará as atividades e suas tarefas correspondentes. Cada atividade possui um conjunto de tarefas associadas, e é necessário concluir todas essas tarefas para finalizar a atividade com sucesso. Cada atividade também é caracterizada por uma data de início e uma data de término, que devem estar alinhadas com o cronograma geral do projeto.

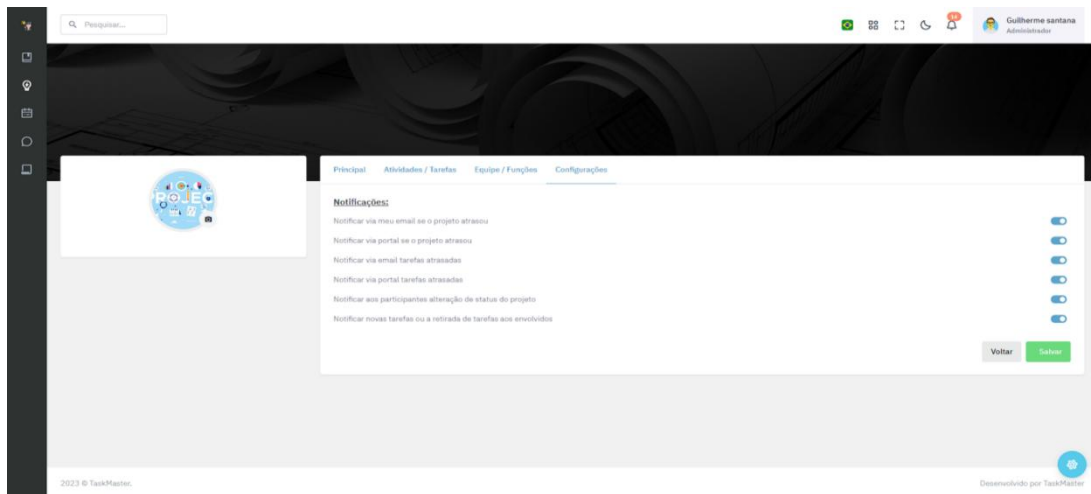
Figura 32 - Aba Equipe / Funções na tela Projeto



Fonte: Elaborado pelo o Autor

Na aba “Equipe / Funções” (Figura 32), você terá a oportunidade de atribuir uma tarefa a um ou mais usuários da plataforma. Se você optar por não vincular uma tarefa a outra pessoa, ela ficará atribuída ao responsável do projeto por padrão. Após o cadastro do projeto, todos os usuários receberão uma notificação informando sobre a tarefa que foi atribuída no projeto, garantindo que todos estejam cientes de suas responsabilidades.

Figura 33 - Aba Configurações na tela Projeto



Fonte: Elaborado pelo o Autor

Na aba “Notificações” (Figura 33), será onde você poderá configurar as notificações em seu projeto. Nela é possível ativar e desativar quando quiser as seguintes notificações: “Notificar via meu email se o projeto atrasou”, “Notificar via portal se o projeto atrasou”, “Notificar via email tarefas atrasadas”, “Notificar via portal tarefas atrasadas”, “Notificar aos participantes alteração de status do projeto” e “Notificar novas tarefas ou a retirada de tarefas aos envolvidos”.

A tela “Projeto” possui o seguintes requisitos:

Quadro 12 - Requisitos da tela Projeto

(continua)

Requisitos	Tipo	Importância	Complexidade
[RF001] - A tela “Projeto” deve possibilitar que o usuário cadastre as informações do seu projeto.	Funcional	Alta	Alta

Fonte: Elaborado pelo o Autor

Quadro 12 - Requisitos da tela Projeto

(continuação)

Requisitos	Tipo	Importância	Complexidade
[RF002] - O <i>data grid</i> da tela “Projeto” deve listar todos os projetos que foram criados pelo usuário.	Funcional	Alta	Média
[RF003] - O <i>data grid</i> de projetos deve oferecer ao usuário a opção de concluir, cancelar ou excluir um projeto.	Funcional	Alta	Alta
[RF004] - Se o projeto estiver com um status diferente de "aberto" ou "atrasado", o <i>data grid</i> da tela “Projeto” deve habilitar apenas o botão de visualização, permitindo que o usuário veja as informações do projeto sem a opção de edição (servindo de histórico).	Funcional	Alta	Baixa
[RF005] - O <i>data grid</i> da tela “Projeto” deve oferecer ao usuário a opção de gerar um relatório dos projetos criados no formato PDF ou XLS.	Funcional	Baixa	Média
[RF006] - A tela “Projeto” deve ser organizada em quatro abas distintas: <ul style="list-style-type: none"> ● Principal ● Atividades / Tarefas ● Equipe / Funções ● Configurações 	Funcional	Alta	Baixa

Fonte: Elaborado pelo o Autor

Quadro 12 - Requisitos da tela Projeto

(continuação)

Requisitos	Tipo	Importância	Complexidade
[RF007] - Na aba "Principal", devem estar contidas as informações principais do projeto, incluindo: Foto, Título, Descrição, Data de Início e Data de Término.	Funcional	Alta	Baixa
[RF008] - Na aba "Principal", deve haver um campo chamado "Template", onde, após ser selecionado, as informações relativas àquele <i>template</i> serão trazidas, criando assim um projeto com base nas informações do <i>template</i> selecionado.	Funcional	Alta	Alta
[RF009] - Na aba "Atividades / Tarefas", deve ser possível cadastrar, editar ou excluir uma atividade. Cada atividade deve incluir pelo menos uma tarefa, com datas de início e fim definidas.	Funcional	Alta	Alta
[RF010] - Na aba "Atividades / Tarefas", deve ser possível aplicar filtros para encontrar as atividades e tarefas cadastradas de forma mais eficiente.	Funcional	Baixa	Alta
[RF011] - Ao clicar em uma tarefa, uma modal (componente do Bootstrap) com um formulário deve ser aberta, permitindo que o usuário especifique informações adicionais para aquela tarefa, como prioridade, <i>tag</i> e descrição.	Funcional	Alta	Média

Fonte: Elaborado pelo o Autor

Quadro 12 - Requisitos da tela Projeto

(conclusão)

Requisitos	Tipo	Importância	Complexidade
[RF012] - Na aba “Equipe / Funções”, deve ser possível atribuir uma ou mais tarefas a um usuário da plataforma.	Funcional	Alta	Alta
[RF013] - Na aba “Equipe / Funções” deve ser possível filtrar as tarefas atribuídas a um usuário.	Funcional	Baixa	Média
[RF014] - Deve ser desenvolvido uma API que envie notificações diariamente aos usuários por meio de emails ou notificações na plataforma, respeitando as configurações definidas no projeto.	Funcional	Alta	Alta
<p>[RF015] - Deve ser configurado o sistema de notificações para os seguintes eventos de um projeto:</p> <ul style="list-style-type: none"> ● Notificar via email se o projeto atrasou. ● Notificar via portal se o projeto atrasou. ● Notificar via email tarefas atrasadas. ● Notificar via portal tarefas atrasadas ● Notificar aos participantes alteração de status do projeto. ● Notificar novas tarefas ou a retirada de tarefas aos envolvidos. 	Funcional	Média	Média

Fonte: Elaborado pelo o Autor

A tela “Projeto” possui as seguintes regras de negócio:

Quadro 13 - Regra de negócio da tela Projeto

Regra de negócio	Tipo
[RN001] - Criação do projeto	Para criar um projeto, é necessário incluir no mínimo uma atividade, e dentro de uma atividade, deve haver pelo menos uma tarefa.
[RN002] - Alterar o status de uma atividade	Para concluir uma atividade, todas as tarefas relacionadas a ela devem ter o status "Concluído".
[RN003] - Atribuição de tarefas a participantes	Se não for especificado um responsável para uma tarefa, a tarefa será automaticamente atribuída ao proprietário do projeto.
[RN004] - Criação de atividades	Não é permitido cadastrar atividades com nomes idênticos.
[RN005] - Criação de atividade com o projeto atrasado	Se um projeto estiver com status atrasado e uma nova atividade for cadastrada, essa atividade também terá status atrasado.

Fonte: Elaborado pelo o Autor

3.5.8 Cronograma

Os cronogramas desempenham um papel crítico na gestão de projetos, pois proporcionam uma representação visual das atividades do projeto em relação ao tempo. Essa representação clara e organizada é fundamental para o sucesso de qualquer empreendimento, pois ajuda a equipe de projeto e todas as partes interessadas a compreenderem o plano de execução e os prazos envolvidos.

De acordo com o Project Management Institute (PMI, 2017, p.175), um cronograma

"fornece um plano detalhado que representa como e quando o projeto vai entregar os produtos, serviços e resultados definidos no escopo do projeto, e serve como ferramenta de comunicação, gerenciamento de expectativas das partes interessadas e como base para a emissão de relatórios de desempenho".

Ricardo Vargas (2016), no livro "Gerenciamento de projetos: Estabelecendo Diferenciais Competitivos," ressalta a importância dos cronogramas como ferramentas cruciais para a gestão eficaz do tempo. Ele enfatiza que um cronograma bem elaborado é essencial para evitar conflitos de recursos e maximizar a utilização eficiente do tempo disponível. Com o objetivo de facilitar a gestão de projetos e a criação de cronogramas, foi desenvolvida a tela "Cronograma".

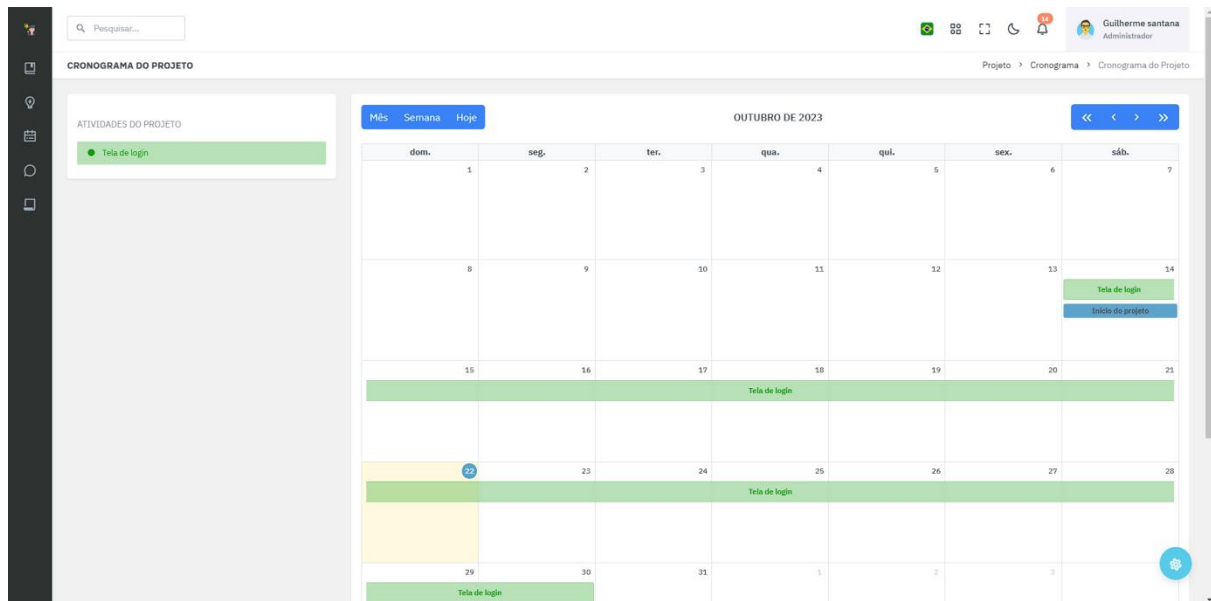
Figura 34 - Data grid da tela Cronograma

Ações	Cód	Título	Data de início	Data de conclusão	Status Projeto	Andamento do projeto
	47	Aplicativo mobile	03/11/2023	12/01/2024	Aberto	<div style="width: 100%; height: 10px; background-color: #007bff;"></div>
	46	Software e-commerce	15/10/2023	26/11/2023	Aberto	<div style="width: 100%; height: 10px; background-color: #007bff;"></div>
	44	Software e-commerce	04/09/2023	16/10/2023	Atrasado	<div style="width: 100%; height: 10px; background-color: #dc3545;"></div>

Fonte: Elaborado pelo o Autor

No *data grid* da tela "Cronograma" (Figura 34), são apresentadas informações essenciais do projeto para o qual o cronograma será configurado. Este *data grid* exibirá todos os projetos registrados no *software* (caso seja administrador) ou apenas aos seus projetos (caso seja um usuário comum).

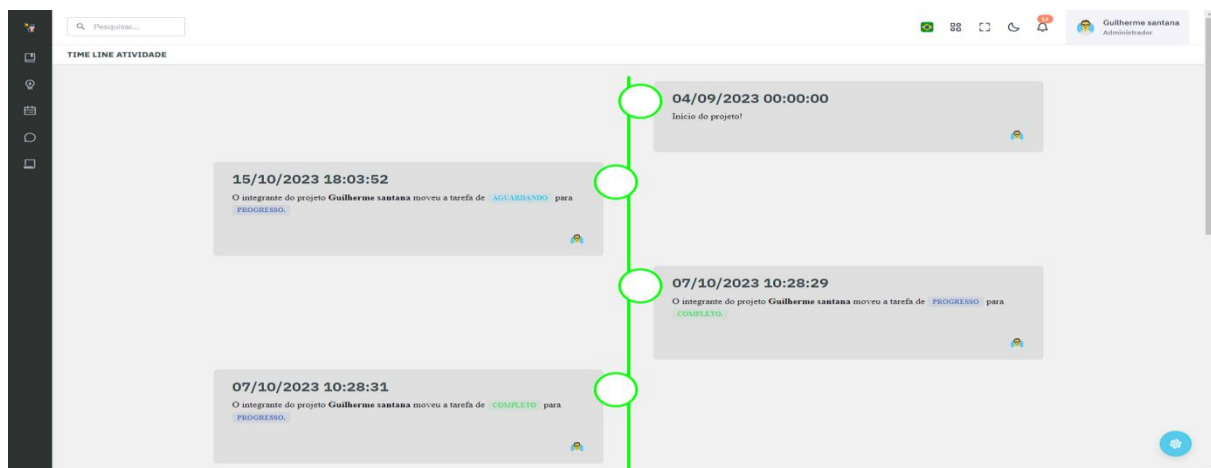
Figura 35 - Tela Cronograma



Fonte: Elaborado pelo o Autor

Na tela “Cronograma” (Figura 35), um calendário será exibido de maneira variada (Mês / Semana / Hoje), destacando as atividades atribuídas ao projeto dentro do período determinado. Ao clicar em uma atividade específica, os usuários poderão acessar uma linha do tempo como evidenciado na Figura 36 que detalha todas as atualizações relacionadas às suas tarefas, bem como quaisquer possíveis mudanças de status nesta atividade. Essa abordagem oferece uma visão mais detalhada e interativa das atividades do projeto e de seu progresso ao longo do tempo.

Figura 36 - Timeline na tela Cronograma



Fonte: Elaborado pelo o Autor

A tela “Cronograma” possui os seguintes requisitos:

Quadro 14 - Requisitos da tela Cronograma

Requisitos	Tipo	Importância	Complexidade
[RF001] - A tela “Cronograma” deve possibilitar que o usuário consulte o cronograma gerado após a criação do projeto.	Funcional	Média	Média
[RF002] - A tela “Cronograma” deve oferecer ao usuário a capacidade de visualizar o cronograma em três formatos diferentes: diário, semanal e mensal.	Funcional	Alta	Alta
[RF003] - Ao clicar em uma atividade específica no calendário, o usuário deve ser redirecionado para a <i>timeline</i> da atividade.	Funcional	Alta	Baixa
[RF004] - A <i>timeline</i> deve ser organizada por data, exibindo a data, o status e o responsável pelas atualizações das tarefas na atividade correspondente.	Funcional	Alta	Alta
[RF005] - O <i>data grid</i> deve exibir todos os projetos nos quais o usuário é o proprietário ou todos caso o usuário seja do perfil administrador.	Funcional	Alta	Média

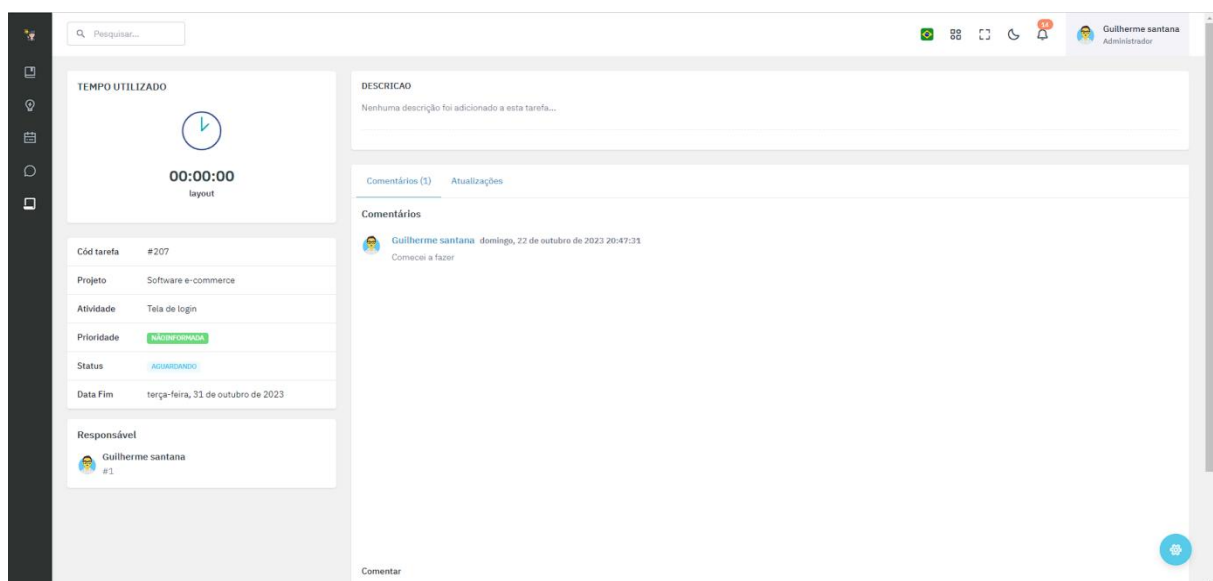
Fonte: Elaborado pelo o Autor

3.5.9 Detalhes da tarefa

Essa tela é acessada (Figura 37) quando você clica em "Ver mais detalhes" de uma tarefa na tela "Registro de Tarefas" ou após selecionar a opção "Visualizar" na tela "Administração de Tarefas". Seu propósito é fornecer uma visão mais abrangente de sobre a tarefa. No contexto dessa tela:

- Você pode verificar o status atual da tarefa para entender em que fase ela se encontra.
- Encontra a data prevista ou efetiva de conclusão da tarefa.
- A aba "Comentários", permite que os membros da equipe compartilhem informações e discussões relevantes sobre a tarefa.
- A aba "Atualizações" fornece um histórico das mudanças e atualizações feitas na tarefa durante o projeto.
- É possível ver o tempo que foi dedicado à execução da tarefa, especialmente quando ela está na coluna "Fazendo" na tela "Registro de Tarefas".
- A prioridade da tarefa é destacada, o que ajuda na gestão das prioridades em relação a outras tarefas.
- Uma descrição detalhada oferece informações adicionais sobre o que precisa ser realizado na tarefa.

Figura 37 - Tela Detalhes da tarefa



Fonte: Elaborado pelo o Autor

Essa tela é essencial para uma compreensão completa da tarefa, fornecendo acesso a dados e contexto importantes que facilitam a colaboração e o gerenciamento eficaz do trabalho em curso.

Esta tela possui os seguintes requisitos:

Quadro 15 - Requisitos da tela Detalhes da Tarefa

Requisitos	Tipo	Importância	Complexidade
[RF001] - A tela “Detalhes da tarefa”, deve possibilitar que o usuário consulte informações adicionais e específicas de uma tarefa.	Funcional	Média	Média
[RF002] - A tela “Detalhes da tarefa”, deve apresentar um contador que exiba o tempo gasto da tarefa na coluna "Fazendo" na tela “Registro de Tarefas”.	Funcional	Alta	Alta
[RF003] - A tela “Detalhes da tarefa”, deve permitir que o usuário faça e consulte comentários feitos pela equipe em relação àquela tarefa.	Funcional	Alta	Alta
[RF004] - A tela “Detalhes da tarefa” deve conter uma aba que exiba todas as atualizações de status da tarefa realizadas na tela “Registro de Tarefas”.	Funcional	Alta	Alta

Fonte: Elaborado pelo o Autor

3.5.10 Administração de tarefas

Essa tela tem a finalidade de gerenciar e administrar as tarefas relacionadas a um projeto específico. Ela apresenta um *dashboard* que oferece uma visão geral dos status das tarefas, fornecendo um totalizador que resume essas informações de

forma concisa. A tela possibilita a criação de novas tarefas dentro de uma atividade, bem como a opção de excluí-las ou editá-las conforme necessário.

Ela foi concebida como um recurso destinado principalmente ao responsável pelo projeto e aos usuários com perfil administrador. Sua funcionalidade central é fornecer uma visão abrangente e completa de todas as tarefas relacionadas ao projeto em questão, facilitando o acompanhamento do progresso e a tomada de decisões informadas.

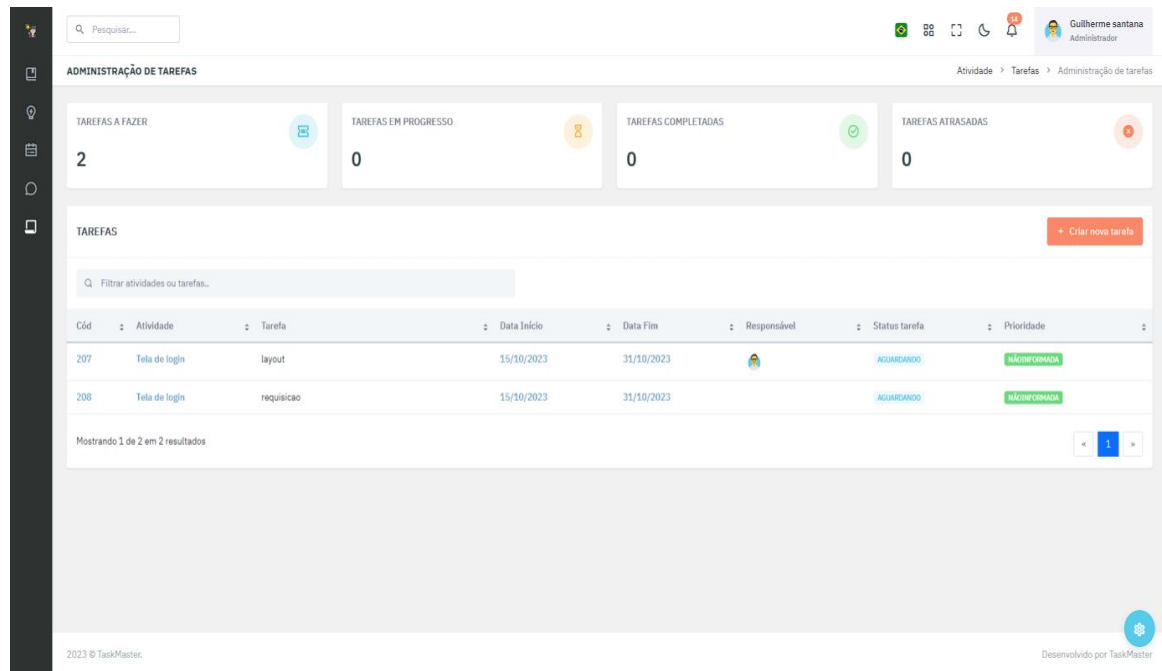
Figura 38 - Data grid da tela Administração de Tarefas

Ações	Cód	Título	Data de início	Data de conclusão	Status Projeto	Andamento do projeto
	47	Aplicativo mobile	03/11/2023	12/01/2024	Aberto	<div style="width: 100%; height: 10px; background-color: #ccc;"></div>
	46	Software e-commerce	15/10/2023	26/11/2023	Aberto	<div style="width: 80%; height: 10px; background-color: #ccc;"></div>
	44	Software e-commerce	04/09/2023	16/10/2023	Atrasado	<div style="width: 100%; height: 10px; background-color: #ccc;"></div>

Fonte: Elaborado pelo o Autor

No *data grid* da tela “Administração de Tarefas” (Figura 38), são apresentadas informações essenciais do projeto. Este *data grid* exibirá todos os projetos registrados no *software* (caso seja administrador) ou apenas aos seus projetos (caso seja um usuário comum). O *data grid* possui duas ações, caso o projeto esteja com status aberto ou atrasado, ele têm acesso a todas as funcionalidades da tela (botão azul habilitado), caso não esteja, ele têm acesso apenas como visualização servindo assim como histórico (botão verde habilitado).

Figura 39 - Tela Administração de Tarefas



Fonte: Elaborado pelo o Autor

A tela possui em sua composição, na parte superior um contador que contabiliza as tarefas e em que status estão, permitindo assim, uma visualização mais abrangente do estado geral do projeto. Na parte inferior, consta uma tabela onde são listados todas as tarefas do projeto, onde o usuário pode incluir uma nova tarefa, editar uma tarefa já existente ou acessar a tela “Detalhes da tarefa” para então poder visualizar mais informações da tarefa selecionada.

A tela “Administração de Tarefas” possui os seguintes requisitos:

Quadro 16 - Requisitos da tela Administração de Tarefas

(continua)

Requisitos	Tipo	Importância	Complexidade
[RF001] - A tela “Administração de tarefas” deve permitir o usuário acrescentar, editar ou remover novas tarefas de um projeto.	Funcional	Alta	Alta

Fonte: Elaborado pelo o Autor

Quadro 16 - Requisitos da tela Administração de Tarefas
(continuação)

Requisitos	Tipo	Importância	Complexidade
[RF002] - A tela “Administração de tarefas” deve fornecer uma visão geral ao usuário sobre o andamento das tarefas no projeto por meio de contadores que exibam o status das tarefas.	Funcional	Alta	Média
[RF003] - O <i>data grid</i> desta tela deve exibir apenas os projetos nos quais o usuário é o proprietário ou todos da plataforma caso o usuário seja um administrador.	Funcional	Alta	Alta
[RF004] - A tela deve apresentar um botão de ação em cada linha da tabela, permitindo assim, redirecionar o usuário para a tela “Detalhes da tarefa”.	Funcional	Média	Média
[RF005] - Quando uma tarefa for incluída, excluída ou modificada, todos os envolvidos devem ser notificados sobre a ação realizada.	Funcional	Alta	Alta
[RF006] - Em projetos com status diferente de "aberto" e "atrasado", apenas a opção de visualização deve estar habilitada no <i>data grid</i> . Portanto, não será possível criar, editar ou excluir tarefas nesses projetos, e a tela se comportará apenas como um histórico.	Funcional	Alta	Média

Fonte: Elaborado pelo o Autor

Quadro 16 - Requisitos da tela Administração de Tarefas

(conclusão)

Requisitos	Tipo	Importância	Complexidade
[RF007] - Em projetos com status "aberto" e "atrasado", apenas a opção "gerenciar" deve estar habilitada no <i>data grid</i> , proporcionando acesso a todas as funcionalidades que a tela contém.	Funcional	Alta	Média

Fonte: Elaborado pelo o Autor

A tela "Administração de projetos" possui as seguintes regras de negócio:

Quadro 17 - Regra de negócio da tela Administração de Tarefas

Regra de negócio	Tipo
[RN001] - Exclusão de tarefas	Se todas as tarefas de uma atividade forem excluídas, a atividade também deve ser automaticamente excluída.
[RN002] - Exclusão de atividades	Não deve ser permitido que o usuário exclua todas as atividades de um projeto. Se o usuário estiver excluindo a última tarefa de uma atividade, esta mensagem deve ser exibida: "Não é possível deletar esta tarefa pois ela é a última de sua atividade, em um projeto é necessário pelo menos uma atividade com uma tarefa!"

Fonte: Elaborado pelo o Autor

3.5.11 Registro de tarefas

O quadro Kanban é uma abordagem poderosa na gestão de projetos, com raízes profundas no Sistema Toyota de Produção. Originário do Japão, o termo "Kanban" significa "cartão visual" e representa uma maneira altamente eficaz de visualizar o trabalho, controlar o trabalho em progresso e promover a melhoria contínua do processo.

O quadro Kanban é composto por colunas que representam diferentes estágios do fluxo de trabalho, como "A fazer", "Em andamento" e "Concluído". Cada tarefa ou trabalho é representado por cartões ou *post-its*, que são movidos pelas colunas à medida que progredem no processo. Essa visualização oferece transparência e controle, permitindo que a equipe acompanhe o progresso e tome decisões informadas sobre o trabalho em andamento.

A limitação do trabalho em progresso (WIP) é uma característica fundamental do Kanban. Essa limitação é aplicada definindo-se um limite para o número de cartões que podem estar em cada coluna. Quando uma coluna atinge seu limite, não é permitido iniciar novas tarefas até que espaço esteja disponível. Essa prática evita a sobrecarga da equipe, foca na conclusão de tarefas antes de iniciar novas e mantém um fluxo de trabalho equilibrado.

Henry Kniberg e Mattias Skarin (KNIBERG; SKARIN, 2010), autores de "Kanban and Scrum making the Most of Both" destacam o valor do quadro Kanban como uma ferramenta visual que ajuda as equipes a controlar o trabalho em progresso e a melhorar o fluxo. Eles enfatizam a importância da gestão do fluxo de trabalho e como o quadro Kanban ajuda a identificar gargalos e ineficiências, permitindo que a equipe tome medidas para otimizar o processo.

James K. Liker, autor de "O modelo Toyota: 14 Princípios de gestão do maior fabricante do mundo", destaca que o Kanban é uma das ferramentas-chave do Sistema Toyota de Produção, promovendo a evolução constante do processo (LIKER, 2021). Ao visualizar o trabalho e os resultados de forma transparente, a equipe pode identificar áreas que precisam de aprimoramento e implementar ajustes de maneira iterativa.

Para seguir os princípios da metodologia ágil, optou-se pelo uso do quadro Kanban para o controle da execução das tarefas.

Figura 40 - Data grid da tela Registro de Tarefas

Ações	Cód	Título	Data de início	Data de conclusão	Gerente do projeto	Andamento do projeto
	47	Aplicativo mobile	03/11/2023	12/01/2024	Guilherme santana	<div style="width: 100%;"></div>
	46	Software e-commerce	15/10/2023	26/11/2023	Usuario	<div style="width: 100%;"></div>
	44	Software e-commerce	04/09/2023	16/10/2023	Guilherme santana	<div style="width: 100%;"></div>

Fonte: Elaborado pelo o Autor

No *data grid* da tela “Kanban” (Figura 40), são apresentadas informações essenciais do projeto para o qual o Kanban será exibido. Este *data grid* exibirá todos os projetos registrados no *software* (caso seja administrador) ou apenas aos seus projetos ou a projetos que você tenha tarefas a cumprir (caso seja um usuário comum).

Figura 41 - Tela Registro de Tarefas

Fonte: Elaborado pelo o Autor

Na tela "Registro de Tarefas" (Figura 41), você encontrará um sistema organizado em três colunas distintas: "A Fazer", "Fazendo" e "Feito". Cada uma dessas colunas representa o status atual das tarefas designadas a você no projeto. A atualização do status de uma tarefa é uma operação simples e eficiente. Basta mover o cartão correspondente de uma coluna para outra, refletindo assim o estado atual da tarefa. No entanto, é importante destacar que essa movimentação só é possível quando a data de início da tarefa corresponde à data atual ou já tenha passado da mesma. Isso garante que as tarefas sejam movidas de acordo com o cronograma estabelecido, evitando ações prematuras ou desalinhadas com o planejamento.

Cada cartão de tarefa no quadro Kanban é rico em informações relevantes. Ele apresenta detalhes como a atividade à qual a tarefa está associada, os responsáveis pela sua execução, as datas de início e término estabelecidas, *tags* que categorizam a tarefa, a sua prioridade se for especificada, uma descrição que oferece contexto adicional e, é claro, seu status atual. Esses detalhes são essenciais para uma compreensão completa de cada tarefa e ajudam a equipe a priorizar e realizar o trabalho de maneira eficaz.

A tela "Registro de Tarefas" possui os seguintes requisitos:

Quadro 18 - Requisitos da tela Registro de Tarefas

(continua)

Requisitos	Tipo	Importância	Complexidade
[RF001] - A tela "Registro de Tarefas" deve ser apresentada em formato de um quadro Kanban, com três colunas distintas: "A Fazer", "Fazendo" e "Feito".	Funcional	Alta	Alta
[RF002] - Qualquer movimentação ou ação realizada deve gerar uma notificação para o proprietário do projeto.	Funcional	Alta	Alta
[RF003] - Deve ser possível realizar uma busca que filtre as tarefas no quadro Kanban	Funcional	Média	Açta

Fonte: Elaborado pelo o Autor

Quadro 18 - Requisitos da tela Registro de Tarefas

(conclusão)

Requisitos	Tipo	Importância	Complexidade
[RF004] -. No cartão relativo a tarefa, deve haver uma opção chamada "Ver detalhes" que, quando clicada, redireciona o usuário para a tela "Detalhes da tarefa".	Funcional	Média	Baixa
[RF005] - As tarefas cuja data de início seja posterior à data atual o cartão devem ser exibidas com uma cor mais acinzentada.	Funcional	Baixa	Baixa
[RF006] - O <i>data grid</i> da tela "Registro de Tarefas" deve listar apenas projetos com status "aberto" nos quais o usuário é o proprietário do projeto ou em projetos nos quais ele tem tarefas a serem feitas.	Funcional	Alta	Média
[RF007] - A tela "Registro de Tarefas" deve listar todas as tarefas se o usuário for o proprietário do projeto, caso contrário deve listar apenas as tarefas atribuídas a ele.	Funcional	Alta	Média
[RF008] - Cada tarefa deve apresentar as seguintes informações: data de início, data de término, porcentagem de conclusão, título da atividade, título da tarefa, <i>tags</i> relacionadas à tarefa e os usuários responsáveis por essa tarefa.	Funcional	Média	Média

Fonte: Elaborado pelo o Autor

A tela “Registro de Tarefas” possui as seguintes regras de negócio:

Quadro 19 - Regra de negócio da tela Registro de Tarefas

Regra de negócio	Tipo
[RN001] - Movimentação de cartões	Ao tentar mover um cartão (tarefa) cuja data de início seja posterior à data atual, a movimentação não deve ser permitida e uma mensagem de erro deve ser exibida: “Aguarde a data [“Data inicial da atividade”] para o início desta atividade!”

Fonte: Elaborado pelo o Autor

3.5.12 Chat

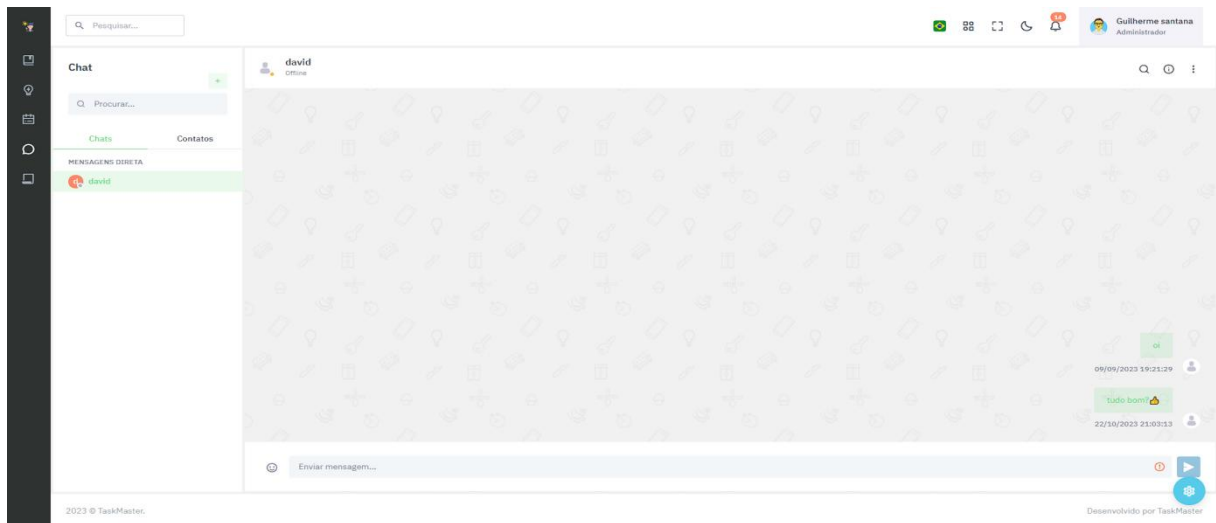
A comunicação eficaz é um pilar fundamental para o sucesso de qualquer projeto. Desde a fase inicial de planejamento até a conclusão e análise pós-implantação, a capacidade de transmitir informações de maneira clara e consistente é essencial para garantir que os objetivos sejam alcançados e que os stakeholders estejam alinhados.

A importância da comunicação eficaz em projetos é destacada por muitos especialistas e pesquisadores. Segundo o PMI (PMI, 2017, p.53) "O gerente de projetos executa funções de comunicação entre o patrocinador, os membros da equipe outras partes interessadas. Isso inclui fornecer diretrizes e apresentar a visão de sucesso para o projeto". Isso reflete o papel central que a comunicação desempenha na gestão de projetos, onde a transmissão de informações, expectativas e *feedback* é crucial para a coordenação de equipes e a tomada de decisões informadas.

Apesar de seu papel crítico, a comunicação ainda é uma área problemática em muitos projetos. Muitos estudos têm investigado as principais causas de falha em projetos, e a comunicação inadequada é consistentemente identificada como um dos principais fatores. De acordo com pesquisa da PwC em 2018 ("Pulse of the Profession"), a falta de comunicação clara é responsável por 29% dos projetos que falham (PMI, 2018).

Levando em consideração a importância da comunicação eficaz em projetos, foi desenvolvido um chat dentro da plataforma, proporcionando aos usuários a capacidade de se comunicarem sobre assuntos e temas diretamente relacionados ao projeto. Essa implementação visa atender à necessidade crítica de uma comunicação eficiente, permitindo que todos os envolvidos interajam de maneira conveniente e produtiva.

Figura 42 - Tela Chat



Fonte: Elaborado pelo o Autor

A tela "Chat" (Figura 42) é o local central para interações com os usuários da plataforma. Além de possibilitar a comunicação com todos os usuários, ela oferece a funcionalidade de adicionar novos contatos e enviar mensagens em tempo real. Essa interface dinâmica e interativa promove uma experiência de comunicação fluida e eficaz entre os membros da comunidade. Esta tela possui os requisitos abaixo:

Quadro 20 - Requisitos da tela Chat

(continua)

Requisitos	Tipo	Importância	Complexidade
[RF001] - A tela "Chat" deve possibilitar a troca de mensagens simultâneas em tempo real.	Funcional	Alta	Alta

Fonte: Elaborado pelo o Autor

Quadro 20 - Requisitos da tela Chat

(conclusão)

[RF002] - A tela “Chat” deve permitir o cadastro e a exclusão de contatos.	Funcional	Alta	Média
[RF003] - A tela “Chat” deve permitir o bloqueio e o desbloqueio de um contato.	Funcional	Média	Média
[RF004] - A tela “Chat” deve oferecer a funcionalidade de silenciar e desabilitar o silenciamento de um contato.	Funcional	Baixa	Média
[RF005] - A tela “Chat” deve possibilitar que o usuário exclua mensagens.	Funcional	Baixa	Média
[RF006] - A tela “Chat” deve permitir que o usuário visualize informações adicionais sobre o contato.	Funcional	Média	Baixa
[RF007] - O <i>chat</i> deve permitir que o usuário pesquise mensagens enviadas ou recebidas.	Funcional	Média	Baixa
[RF008] - A troca de mensagens em tempo real deve ser implementada usando a biblioteca “SignalR”.	Não Funcional	Alta	Alta

Fonte: Elaborado pelo o Autor

Esta tela possui as regras de negócio abaixo:

Quadro 21 - Regra de negócio da tela Chat

Regra de negócio	Tipo
[RN001] - Envio de mensagem	<p>Se o usuário tiver bloqueado outro contato ou tenha sido bloqueado por esse contato, o botão de envio de mensagem será desabilitado, e uma aviso será exibida:</p> <ul style="list-style-type: none"> ● Se você bloqueou o contato: "Contato bloqueado". ● Se você foi bloqueado pelo contato: "Não é possível enviar mensagens para este contato".

Fonte: Elaborado pelo o Autor

4 FUNDAMENTAÇÃO TÉCNICA

A fundamentação técnica de código desempenha um papel crucial na compreensão e avaliação de qualquer *software* desenvolvido. É por meio do código-fonte que as ideias e os conceitos teóricos se transformam em aplicações funcionais, prontas para atender às necessidades dos usuários.

Neste capítulo do TCC, será explorado a fundação técnica por trás do *software*, analisando os aspectos mais importantes do código, sua estrutura, arquitetura e as tecnologias utilizadas para implementá-lo. Será aprofundado a compreensão das decisões de *design*, da eficiência da programação e do rigor técnico que fundamentam a qualidade e a eficácia do *software*.

Esta parte da pesquisa não apenas destaca a técnica envolvida no desenvolvimento do *software*, mas também fornece uma base sólida para a avaliação crítica e a melhoria contínua do produto.

4.1 IDE (INTEGRATED DEVELOPMENT ENVIRONMENT)

Neste capítulo, será abordado as Integrated Development Environment (IDEs) utilizadas no desenvolvimento do "TaskMaster".

4.1.1 Rider

O Rider foi desenvolvido pela JetBrains, uma empresa de *software* com sede em Praga, República Tcheca. A JetBrains é conhecida por criar ferramentas de desenvolvimento de *software* de alta qualidade, incluindo IDEs populares como o IntelliJ IDEA e o PhpStorm. Segundo a JetBrains (JETBRAINS, 2023):

"o Rider é um IDE multiplataformas rápido e poderoso baseado na plataforma IntelliJ e no ReSharper. Ele combina ferramentas para desenvolvimento Web e suporte a banco de dados, para que você tenha tudo do que precisa para desenvolver aplicações ASP.NET Core e ASP.NET clássicas em um só lugar."

O Rider oferece uma série de funcionalidades poderosas para desenvolvedores de .NET, tornando-o uma escolha atraente. Algumas de suas funcionalidades notáveis incluem:

- **Suporte Multiplataforma:** O Rider é uma IDE multiplataforma, compatível com Windows, macOS e Linux.
- **Integração com .NET Core:** Ele oferece suporte nativo para .NET Core, tornando-o ideal para o desenvolvimento de aplicativos .NET modernos e multiplataforma.
- **ReSharper Integrado:** O Rider integra o “ReSharper”, uma ferramenta da JetBrains conhecida por suas capacidades de análise de código e refatoração, tornando o desenvolvimento mais eficiente e seguro.
- **Depuração e Testes:** Possui uma poderosa suíte de ferramentas de depuração e testes para identificar e corrigir problemas no código.
- **Refatorações Inteligentes:** Oferece refatorações inteligentes que ajudam os desenvolvedores a melhorarem a qualidade do código e a manutenção.

O Rider ganhou destaque na comunidade de desenvolvedores de .NET e C#. Sua combinação de funcionalidades avançadas e a sólida reputação da JetBrains no desenvolvimento de *software* o tornaram uma opção respeitável e confiável para projetos de .NET.

4.1.2 Visual Studio Code

O Visual Studio Code, foi desenvolvido pela Microsoft, uma das gigantes da indústria de tecnologia. Sua origem remonta ao anúncio oficial em abril de 2015, quando a Microsoft apresentou o editor de código pela primeira vez.

O VS Code foi projetado para atender às necessidades dos desenvolvedores em um ambiente de código aberto. Segundo a Microsoft o "Visual Studio Code é um editor de código-fonte leve, mas poderoso, executado em sua área de trabalho e disponível para Windows, macOS e Linux". (MICROSOFTa, 2023).

O Visual Studio Code conquistou uma base de usuários considerável desde seu lançamento, tornando-se uma das ferramentas mais amplamente adotadas no desenvolvimento de *software*. É reconhecido pela comunidade de desenvolvedores como uma opção eficaz e versátil para suas necessidades de codificação. Isso se deve há segundo a revista "InfoWorld" no VS Code "os desenvolvedores gostam da sensação leve do Visual Studio Code como um editor combinado com sua

capacidade de verificar sintaxe, completar código, refatorar código, depurar e fazer check-in em um repositório”. (HELLER, 2022).

Levando isso em conta, o Visual Studio Code se torna uma escolha popular para desenvolvedores que buscam uma experiência de desenvolvimento eficiente e altamente personalizável.

4.2 BACKEND

Nesta seção, será abordado os *frameworks* utilizados para criação das APIs que alimentam o “TaskMaster”. Será apresentado a infraestrutura e os conceitos utilizados no lado do servidor para fornecer uma compreensão mais ampla das ferramentas e técnicas empregadas durante o desenvolvimento.

4.2.1 .NET

O .NET é uma plataforma de desenvolvimento de *software* que proporciona uma estrutura unificada para criar aplicativos em diversos sistemas operacionais. A Microsoft (MICROSOFT, 2023b) descreve o .NET da seguinte forma “o .NET é a estrutura gratuita, de *software* livre e multiplataforma para compilar aplicativos modernos e serviços de nuvem poderosos”. Essa versatilidade permite que os desenvolvedores alcancem uma audiência mais ampla e desenvolvam aplicativos que podem ser executados em diferentes plataformas.

Uma das vantagens proeminentes do .NET é sua capacidade de ser executado em várias plataformas. Scott Hunter, diretor de programas do .NET na Microsoft, enfatiza que “você pode construir qualquer coisa com .NET. É uma das principais razões pelas quais milhões de desenvolvedores escolhem o .NET como plataforma para suas carreiras e as empresas investem em seus negócios”. (HUNTER, 2020). Essa flexibilidade permite que os desenvolvedores atendam a diversas necessidades e públicos.

O .NET é respaldado por uma comunidade de desenvolvedores ativa e uma rica biblioteca de classes, ressaltado pela própria Microsoft (MICROSOFT, 2023b) que diz que “o ecossistema de pacotes mais seguro, confiável por milhões de desenvolvedores. Com mais de 320,000 pacotes exclusivos até hoje e mais 50,000

a cada ano". O .NET é uma plataforma de desenvolvimento de *software* versátil que oferece vantagens notáveis, como suporte multiplataforma, um ecossistema robusto e uma variedade de recursos para desenvolvedores. Suas capacidades englobam o desenvolvimento web, móvel e muito mais. Com base em referências confiáveis da Microsoft e outras fontes, o .NET é uma escolha sólida para o desenvolvimento de aplicativos em diversos cenários.

Devido a todas a essas vantagens a flexibilidade ela foi escolhida para o desenvolvimento back-end do "TaskMaster". Com isso, será detalhado e conceituado as principais bibliotecas .NET utilizadas no desenvolvimento do "TaskMaster" nos próximos capítulos.

4.2.2 Jwt Token

A segurança é um aspecto crítico no desenvolvimento de aplicações .NET, especialmente quando se trata de proteger informações confidenciais dos usuários. Neste contexto, os *tokens* JWT emergem como uma solução eficaz para garantir a autenticação segura e a troca de informações entre aplicações e usuários.

Os *tokens* JWT, ou JSON Web Tokens, são uma forma eficaz de autenticação e autorização em aplicações .NET. Segundo a documentação o

"JSON Web Token (JWT) é um padrão aberto (RFC 7519) que define uma maneira compacta e independente de transmitir informações com segurança entre as partes como um objeto JSON. Essas informações podem ser verificadas e confiáveis porque são assinadas digitalmente." (AUTH0, 2023).

A utilização de *tokens* JWT oferece vários benefícios que fortalecem a segurança. Os *tokens* JWT permitem que os usuários se autenticem com segurança, evitando a exposição de credenciais sensíveis. Com isso, ajuda a reduzir vulnerabilidades comuns, como ataques de Structured Query Language Injection (SQLi) e Cross-Site Request Forgery (CSRF). Ao verificar a assinatura digital dos *tokens*, as aplicações .NET podem garantir a integridade dos dados.

4.2.3 Swagger

O Swagger é uma estrutura de código aberto que auxilia no *design*, documentação e teste de APIs de maneira eficiente. De acordo com a definição oficial o Swagger "é um conjunto poderoso e fácil de usar de ferramentas de desenvolvedor de API para equipes e indivíduos, permitindo o desenvolvimento em todo o ciclo de vida da API, desde o design e documentação até o teste e a implantação". (SMARTBEAR, 2023).

O Swagger gera automaticamente uma documentação interativa para a API, detalhando todos os *endpoints*, métodos, parâmetros e respostas. Isso elimina a necessidade de criar documentação manualmente, economizando tempo e reduzindo erros. Além de documentar a API, o Swagger permite que os desenvolvedores testem a API diretamente por meio da interface Swagger UI. Isso agiliza o processo de depuração e verifica se a API está funcionando conforme o esperado.

O Swagger pode gerar automaticamente código cliente em várias linguagens de programação com base na especificação da API. Isso facilita a integração de aplicativos com a API, garantindo que os clientes a utilizem corretamente.

4.2.4 Automapper

No desenvolvimento de *software*, frequentemente é necessário mapear dados entre objetos de diferentes classes ou estruturas. O AutoMapper é uma ferramenta que visa simplificar esse processo, economizando tempo e esforço. O AutoMapper é uma biblioteca de mapeamento de objetos para .NET que permite mapear automaticamente os valores de propriedades de uma classe para outra. De acordo com a definição oficial o AutoMapper "é um mapeador objeto-objeto. O mapeamento objeto-objeto funciona transformando um objeto de entrada de um tipo em um objeto de saída de um tipo diferente". (BOGARD, 2023).

O AutoMapper traz agilidade para as aplicações, ao eliminar a necessidade de escrever código de mapeamento manual, os desenvolvedores economizam tempo e reduzem a chance de erros. Conforme Tiago Tartari (2023) ressalta a "facilidade de uso e a redução do tempo de desenvolvimento oferecidos pelo

AutoMapper podem ser mais valiosos”. À medida que as classes evoluem, o AutoMapper simplifica a manutenção, pois as alterações nas propriedades são refletidas automaticamente no mapeamento. Isso reduz o risco de “bugs” de mapeamento ao longo do tempo.

O AutoMapper funciona mapeando automaticamente propriedades com nomes correspondentes de uma classe de origem para uma classe de destino. A configuração é feita de forma declarativa, associando as propriedades correspondentes. Isso elimina a necessidade de escrever código repetitivo para mapear manualmente cada propriedade.

4.2.5 SignalR

A comunicação em tempo real desempenha um papel crucial em muitas aplicações web, desde bate-papo em tempo real até notificações em tempo real. O SignalR é uma tecnologia que simplifica a implementação desse tipo de funcionalidade.

O SignalR é uma biblioteca e um conjunto de tecnologias que permitem a comunicação bidirecional em tempo real em aplicações web. Ele é projetado para simplificar a criação de recursos de tempo real e facilitar a comunicação entre clientes e servidores. Segundo a Microsoft

"o ASP.NET Core SignalR é uma biblioteca de código aberto que simplifica a adição de funcionalidade da Web em tempo real aos aplicativos. A funcionalidade da Web em tempo real permite que o código do lado do servidor envie conteúdo aos clientes instantaneamente". (GASTER, 2023).

O SignalR funciona estabelecendo uma conexão persistente entre o cliente e o servidor, permitindo que os dados sejam transmitidos em ambas as direções. Quando os dados são atualizados no servidor, eles são propagados instantaneamente para todos os clientes conectados. Isso é ideal para cenários como chats em tempo real e notificações instantâneas.

4.2.6 Entity Framework

O acesso a dados é uma parte essencial de muitas aplicações de *software*. O Entity Framework é uma tecnologia que oferece uma abordagem simplificada para lidar com o acesso a bancos de dados, facilitando a interação com bancos de dados relacionais. A Microsoft o define como “um mapeador de relação de objetos moderno que permite construir uma camada de acesso a dados limpa, portátil e de alto nível com .NET (C#) em uma variedade de bancos de dados”.(MICROSOFT, 2023c).

O Entity Framework fornece uma camada de abstração sobre o banco de dados subjacente. Isso permite que os desenvolvedores trabalhem com objetos de domínio em vez de se preocuparem com tabelas e colunas de banco de dados. Como resultado, a complexidade do acesso a dados é reduzida significativamente.

Uma das funcionalidades mais poderosas do Entity Framework é o suporte a consultas Language Integrated Query (LINQ), como afirmado pela equipe da Microsoft “o LINQ oferece um modelo mais simples e consistente para trabalhar com dados em vários tipos de fontes de dados e formatos. Em uma consulta LINQ, você sempre trabalha com objetos de programação”. (LEBLANC, 2023). Isso significa que os desenvolvedores podem interagir com os dados usando objetos e consultas LINQ em vez de escrever SQL (Structured Query Language) puro. Essa abstração simplifica o código e torna o desenvolvimento mais produtivo.

Além disso, ele facilita a migração do esquema do banco de dados, nele você pode criar migrações para aplicar alterações de esquema de forma controlada e rastreável. Isso é fundamental para manter a consistência e a integridade dos dados. Além de oferecer também o controle de transações, garantindo que as operações de banco de dados sejam executadas com consistência. Isso é vital em aplicações onde a integridade dos dados é crítica. O Entity Framework possui recursos de cache de entidades que melhoram o desempenho ao minimizar consultas repetidas ao banco de dados. Isso significa que dados frequentemente acessados são armazenados em memória para acesso mais rápido.

Esta biblioteca, é uma ferramenta essencial no desenvolvimento de aplicações .NET, proporcionando benefícios significativos, como acesso a dados simplificado, abstração de banco de dados, consultas LINQ, migrações de banco de dados controladas, gerenciamento de transações e cache de entidades. Essas

funcionalidades tornam o desenvolvimento mais eficiente, facilitando a criação de aplicações robustas e de alto desempenho.

4.2.7 DDD (Domain driven design)

O Domain-Driven Design (DDD) é uma abordagem de *design* de *software* que se concentra em compreender, modelar e resolver os problemas do domínio da aplicação. De acordo com Eric Evans (EVANS, 2010, p.xxii), autor do livro "Domain-Driven Design: Atacando as complexidades no coração do *software*" o DDD "compacta uma grande quantidade de conhecimento em um modelo que reflete uma visão profunda do domínio e um enfoque nos conceitos principais. É uma colaboração entre quem conhece o domínio e quem sabe como construir softwares". Com base nessa filosofia, o DDD propõe a estruturação das camadas de um sistema de *software* de forma a refletir a sua lógica e funcionalidade essenciais.

A camada de Interface do Usuário representa o ponto de interação direta entre os usuários e o sistema, facilitando a exibição de informações e a coleta de entradas. Por outro lado, a camada de Aplicação gerencia a execução das operações e abriga a lógica de alto nível do aplicativo. É na camada de Domínio que reside o coração do DDD, contendo os modelos de domínio, regras de negócios e lógica central. Esta camada é o epicentro do sistema, onde a essência do negócio é expressa e aplicada.

Por fim, a camada de Infraestrutura atua como suporte técnico, fornecendo acesso a bancos de dados, serviços de terceiros e a gestão de recursos. É importante ressaltar que, apesar de oferecer suporte técnico, essa camada não contém regras de negócios. A divisão clara entre essas camadas no DDD permite a criação de sistemas robustos e de fácil manutenção, onde cada camada tem responsabilidades bem definidas e interações bem estruturadas.

A utilização do Domain-Driven Design (DDD) oferece diversos benefícios no desenvolvimento de *software* como o:

- **Alinhamento com o Domínio de Negócios:** O DDD incentiva uma compreensão profunda do domínio de negócios, resultando em *software* que reflete com precisão os requisitos do negócio.

- **Modelos de Domínio Claros:** A camada de domínio permite a criação de modelos de domínio ricos e expressivos, tornando o código mais legível e fácil de manter.
- **Separação de Responsabilidades:** O DDD promove a separação de responsabilidades entre as camadas, melhorando a modularidade do *software*.
- **Flexibilidade e Adaptação:** O foco no domínio permite que o *software* seja adaptado facilmente às mudanças nos requisitos de negócios.
- **Colaboração Eficaz:** O DDD promove uma linguagem ubíqua compartilhada entre desenvolvedores e especialistas de domínio, facilitando a colaboração.

O DDD proporciona benefícios substanciais, como alinhamento com o domínio de negócios, modelos de domínio claros, separação de responsabilidades e flexibilidade para adaptação. É uma técnica valiosa para desenvolvedores que desejam criar *software* de alta qualidade que atenda aos requisitos de negócios de forma eficaz.

4.3 FRONTEND

Neste seção, será apresentado as tecnologias utilizados para criar os *layouts* das páginas do "TaskMaster". Será demonstrado a infraestrutura e os conceitos utilizados no lado do cliente para assim fornecer uma compreensão mais ampla das ferramentas e técnicas empregadas durante o desenvolvimento.

4.3.1 HTML

O *HyperText Markup Language* (HTML) é uma linguagem de marcação usada para criar a estrutura e o conteúdo de páginas da web. Ela desempenha um papel crucial na representação de informações na web. De acordo com a definição oficial (MDN, 2023) o "HTML é o bloco de construção mais básico da web. Define o significado e a estrutura do conteúdo da web".

Além de possibilitar a marcação de elementos como títulos, parágrafos, listas e imagens, o HTML também define a estrutura básica das páginas, incluindo

cabeçalhos, rodapés e áreas de conteúdo. Essa estruturação é essencial para garantir uma experiência de usuário consistente e intuitiva.

Além disso, o HTML viabiliza a criação de *links* entre diferentes páginas da web, facilitando a navegação e a interconexão de informações. Com o suporte a multimídia, é possível incorporar facilmente elementos como áudio e vídeo diretamente nas páginas da web, enriquecendo a experiência do usuário. Por fim, o HTML permite a criação de formulários interativos, o que possibilita aos usuários enviar informações para servidores web de maneira eficiente e conveniente.

Com todas essas funcionalidades integradas, o HTML continua a ser uma ferramenta fundamental para a criação de conteúdo e aprimoramento da experiência de navegação na internet.

4.3.2 CSS

CSS, ou *Cascading Style Sheets*, desempenha um papel fundamental na criação de páginas da web atraentes e funcionais, e é uma parte essencial da web moderna. Como mencionado por Felipe Alex "o CSS é a principal tecnologia utilizada para a definição da aparência, ou seja, aquilo que o usuário visualiza". (SCHEIDT, 2015, p.7). Isso ilustra a importância do CSS como uma linguagem que controla o visual e o *layout* das páginas da web.

Uma das principais vantagens do CSS é sua capacidade de separar a estrutura do conteúdo HTML da apresentação visual, como destacado por Jeffrey Zeldman, um pioneiro do *design* web responsivo, em seu livro "Designing with Web", ressalta que o "layout CSS puro, combinado com XHTML estrutural, permite que você separe o conteúdo do design, tornando seus sites mais flexíveis, acessíveis e fáceis de manter". (ZELDMAN; MARCOTTE, 2010, p.166, tradução nossa). Essa abordagem é fundamental para criar sites que atendam às demandas variadas dos usuários modernos.

4.3.3 Bootstrap

O Bootstrap é um *framework* de código aberto amplamente adotado para o desenvolvimento de interfaces de usuário responsivas e com *design* elegante na

web. Inicialmente criado pela equipe do Twitter, o Bootstrap se tornou uma ferramenta valiosa no kit de ferramentas de qualquer desenvolvedor web. Segundo Mauricio Samy o "Bootstrap é o mais popular framework JavaScript, HTML e CSS para desenvolvimento de sites e aplicações web responsivas e alinhadas com a filosofia *mobile first*". (SILVA, 2023, p.20). Essa afirmação ressalta a sua popularidade e adoção generalizada em todo o mundo.

O Bootstrap é reconhecido por sua capacidade de economizar tempo e esforço no desenvolvimento web. Como observou Heitor Jugrand (JUGRAND, 2023), um experiente desenvolvedor web e autor, "O Bootstrap é uma escolha popular entre os desenvolvedores front-end devido à sua facilidade de uso e ampla documentação."

Uma das características mais notáveis do Bootstrap é sua capacidade de criar *designs* responsivos de forma eficaz. Heitor Jugrand (2023), destaca ainda que "ele oferece uma série de componentes e estilos pré-definidos, facilitando a criação de layouts atraentes e funcionais".

Além disso, o Bootstrap desfruta de uma comunidade ativa e crescente, o que representa uma vantagem significativa para os desenvolvedores. Segundo o site "Built with" (2023) o Bootstrap é utilizado em mais de 12 milhões de sites espalhado pela Internet. Essa colaboração e apoio mútuo na comunidade tornam o Bootstrap uma escolha sólida para aqueles que desejam criar interfaces web eficientes e visualmente atraentes.

4.3.4 Angular

O Angular, é um *framework* de código aberto desenvolvido pelo Google para criar aplicativos web dinâmicos e interativos. Ele usa TypeScript, uma linguagem *superset* do JavaScript, para construir aplicações robustas do lado do cliente. De acordo com a definição oficial o Angular "foi projetado para tornar a atualização o mais simples possível, portanto, aproveite os desenvolvimentos mais recentes com o mínimo de esforço". (ANGULAR, 2022).

O Angular, um dos principais *frameworks* de desenvolvimento frontend, oferece uma série de funcionalidades e benefícios comprovados que o destacam no cenário tecnológico atual. A estrutura do Angular se baseia na organização da

interface do usuário em componentes reutilizáveis, os quais são combinados em *templates* para criar a interface do aplicativo. Além disso, o *framework* fornece um sistema de injeção de dependência que simplifica a criação e o gerenciamento de objetos e serviços, aprimorando a estrutura do código e a modularidade do projeto.

Em termos de ferramentas de desenvolvimento, o Angular se destaca pela sua Command Line Interface (CLI), que simplifica consideravelmente a configuração e a criação de projetos, proporcionando aos desenvolvedores uma experiência de desenvolvimento eficiente e simplificada. Sua relevância no mercado é evidenciada pelo fato de ser uma das tecnologias mais populares, de acordo com a pesquisa do Stack Overflow Developer Survey de 2021, sendo a escolha de 22,96% dos desenvolvedores. (STACK OVERFLOW, 2021).

A produtividade é aprimorada devido ao uso de componentes reutilizáveis e à estrutura bem organizada do Angular, enquanto o desempenho é otimizado com detecção de mudanças eficiente e compilação just-in-time (JIT), garantindo uma experiência de usuário rápida e responsiva. A estrutura modular do Angular também simplifica a manutenção e a escalabilidade de grandes projetos, facilitando o gerenciamento de códigos complexos.

Por fim, o suporte ativo e a comunidade engajada do Angular fornecem uma abundância de recursos, documentação e suporte para os desenvolvedores. O ecossistema Angular é sustentado por um grupo diversificado de mais de 1,7 milhão de desenvolvedores, autores de bibliotecas e criadores de conteúdo, o que confirma a solidez e a abrangência da comunidade em torno deste *framework*. (ANGULAR, 2022).

O Angular é um *framework* poderoso para o desenvolvimento frontend, com funcionalidades avançadas, uma forte presença no mercado e benefícios comprovados, como produtividade aprimorada, desempenho otimizado e manutenção simplificada. Por isso, o Angular se torna um *framework* sólido para *softwares* que buscam uma combinação de desenvolvimento eficiente, alto desempenho, segurança e uma comunidade ativa de suporte.

4.3.5 Typescript

O TypeScript, uma linguagem de programação de código aberto, tem experimentado um aumento notável em sua importância no desenvolvimento de *software* nos últimos anos.

Desenvolvido pela Microsoft, o TypeScript se destaca por sua habilidade de incorporar tipagem estática e melhorias na manutenção de código ao JavaScript. Uma de suas características mais distintas é a tipagem estática, como ressaltado pela documentação oficial "o TypeScript verifica se há erros em um programa antes da execução e o faz com base nos tipos de valores , tornando-o um verificador de tipo estático". (MICROSOFT, 2023d).

A produtividade do desenvolvedor é outra área em que o TypeScript brilha, conforme destacado por Ugo Roveda, Co-Founder na Kenzie, "o TypeScript eleva o nível de produtividade e ainda garante o desenvolvimento de aplicações complexas, eficazes e seguras". (ROVEDA,2023).

O TypeScript tem experimentado um aumento considerável em sua adoção pela comunidade de desenvolvedores. De acordo com dados da pesquisa Stack Overflow Developer Survey de 2021, o TypeScript foi classificado como a sétima linguagem mais utilizada pelos desenvolvedores. (STACK OVERFLOW, 2021).

Além disso, o TypeScript possui um ecossistema amplo e dinâmico, com suporte de várias bibliotecas e *frameworks* populares, como Angular, React e Vue.js. Essa combinação de recursos poderosos e suporte robusto tem consolidado o TypeScript como uma escolha proeminente para o desenvolvimento de *software* atualmente.

4.4 BANCO DE DADOS

Nesta seção, irá ser feito a conceituação e justificção da escolha do sistema de gerenciamento de dados, além de demonstrar o modelo de dados construído para o desenvolvimento do "TaskMaster".

4.4.1 Microsoft SQL Server

O Microsoft SQL Server é um sistema de gerenciamento de banco de dados relacional desenvolvido pela Microsoft, projetado para armazenar, recuperar, gerenciar e analisar dados em diversos tipos de aplicativos, desde aplicações web até sistemas empresariais críticos. Segundo Paulo Lisboa o SQL Server é “projetado para fornecer um ambiente seguro, confiável e escalável para gerenciar grandes volumes de informações” (LISBOA, 2023).

O Microsoft SQL Server é reconhecido por sua robustez e oferece uma série de recursos notáveis que o destacam como uma escolha popular para sistemas de banco de dados. Sua capacidade de proporcionar um desempenho excepcional, especialmente em ambientes de alta demanda, é uma característica fundamental destacada pela Microsoft. Além disso, o SQL Server oferece recursos de segurança avançados, incluindo autenticação, autorização, criptografia de dados e auditoria, o que o torna uma opção confiável para proteger dados sensíveis.

Sua alta escalabilidade é outro ponto forte, permitindo que ele se adapte facilmente às demandas de cargas de trabalho de todos os tamanhos, desde aplicações de pequena escala até soluções corporativas de grande porte. A capacidade de integração perfeita com outras tecnologias da Microsoft, como o sistema operacional Windows e o ambiente de desenvolvimento .NET, é um diferencial significativo, tornando-o uma escolha conveniente para ambientes de desenvolvimento que dependem de várias soluções da Microsoft.

O Microsoft SQL Server é amplamente reconhecido como um sistema de gerenciamento de banco de dados confiável e poderoso, usado globalmente. Suas características notáveis, incluindo desempenho de alto nível, segurança avançada, escalabilidade e integração com a plataforma Microsoft, o tornam uma escolha sólida para organizações que buscam um sistema de gerenciamento de banco de dados eficaz e confiável.

4.4.2 Modelo de banco de dados

Segundo a IBM os modelos de banco de dados podem ser definidos como "o processo de criação de uma representação visual de um sistema de informação

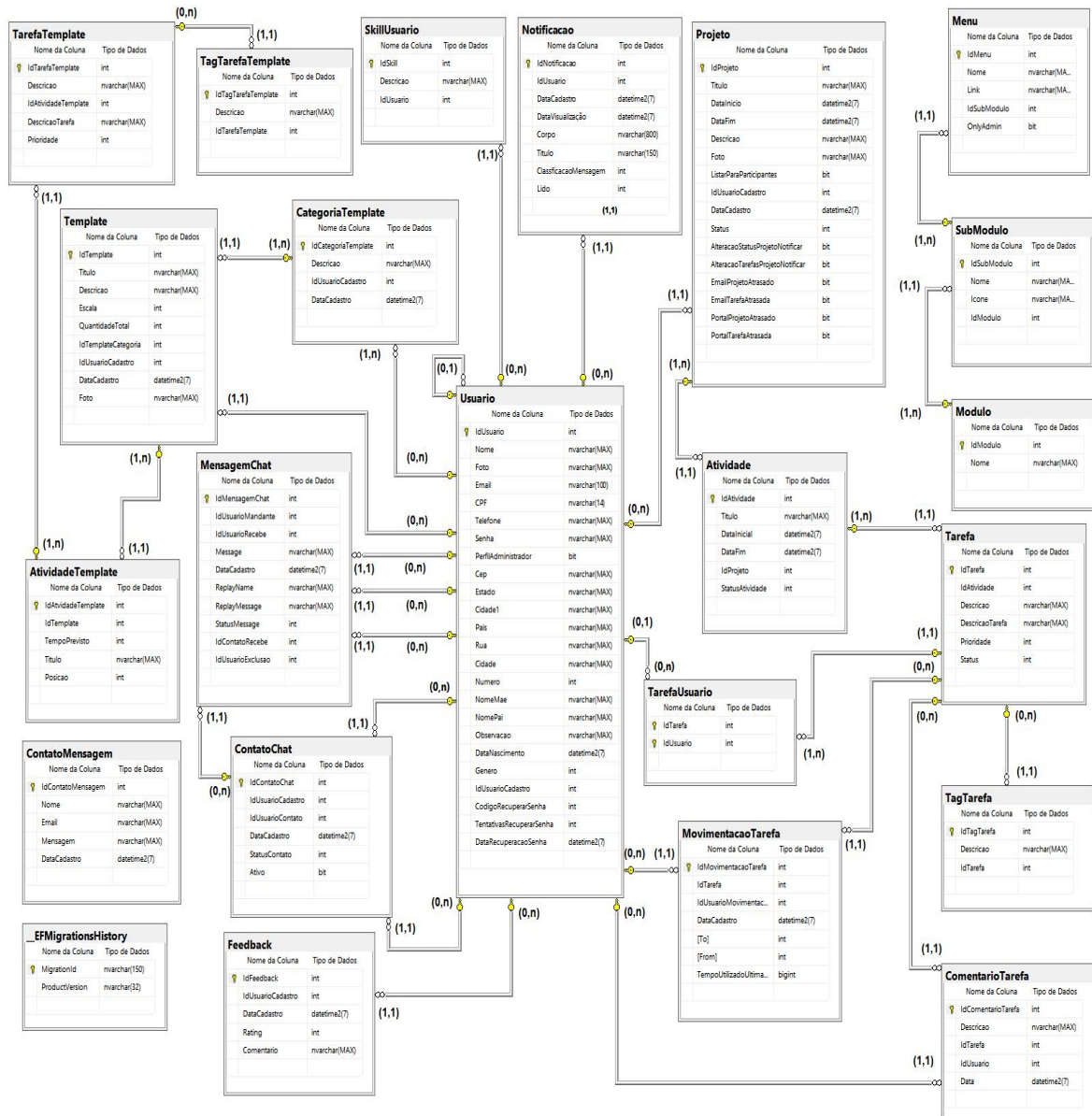
inteiro ou de partes dele para comunicar conexões entre pontos de dados e estruturas” (IBM, 2023). Esses modelos fornecem uma estrutura para a criação e manutenção de bancos de dados, permitindo que os dados sejam armazenados de forma lógica e coerente.

A necessidade de modelos de banco de dados surge da crescente quantidade de dados gerados e utilizados pelas organizações. Sem uma estrutura adequada para armazenar e recuperar informações, as organizações enfrentariam desafios na organização e no acesso aos dados.

Os modelos de banco de dados fornecem uma maneira sistemática de abordar essa complexidade, permitindo que as organizações gerenciem grandes volumes de informações de maneira eficiente. A utilização dos modelos de banco de dados é abrangente em toda a engenharia de *software* e na gestão de sistemas de informação. Durante a fase de projeto de um sistema de *software*, os modelos de banco de dados são fundamentais para definir como os dados serão estruturados e como as entidades e relacionamentos serão representados (Connolly; Begg, 2015). Eles ajudam a criar um plano sólido para a implementação de um sistema de gerenciamento de banco de dados (SGBD), que é essencial para a manipulação e consulta eficaz dos dados.

Além disso, os modelos de banco de dados são usados em toda a vida útil de um sistema para manter a consistência dos dados e garantir que as informações sejam acessadas de maneira precisa e segura. Eles também desempenham um papel importante na otimização de consultas e na garantia da integridade dos dados, que são considerações críticas em ambientes de negócios e científicos.

Figura 43 - Modelo de banco de dados utilizado no Taskmaster



Fonte: Elaborado pelo o Autor

5 CONCLUSÃO

A gestão de projetos é uma área crucial em qualquer organização, independentemente de seu tamanho ou setor de atuação. A eficácia na gestão de projetos pode resultar em economia de tempo, recursos e, em última instância, no sucesso do projeto. Nos últimos anos, tem havido um crescimento notável no uso de *software* para auxiliar na gestão de projetos, oferecendo uma variedade de

ferramentas e recursos que facilitam o planejamento, execução e monitoramento de projetos complexos.

Essas ferramentas são valiosas para uma ampla gama de organizações, proporcionando uma abordagem mais estruturada e eficiente na gestão de projetos. No entanto, apesar da diversidade de opções já disponíveis, foi identificado um nicho que ainda carece de exploração. No contexto atual do mercado, é evidente uma demanda crescente por soluções que tornem a gestão de projetos mais simples e acessível, eliminando a necessidade de preenchimento complexo de diversos campos.

Além disso, há uma necessidade latente de fornecer *templates* prontos para usuários que possam não ter familiaridade com o processo inicial de criação de um projeto. O *software* desenvolvido neste projeto aborda precisamente esse nicho, fornecendo uma abordagem ágil para o planejamento e execução de projetos. Ele capacita os usuários a selecionar e personalizar *templates* de acordo com suas necessidades específicas, preenchendo assim uma lacuna importante no mercado de gerenciamento de projetos.

O desenvolvimento do sistema “TaskMaster” enfrentou diversos desafios ao longo de sua jornada. A necessidade de lidar com novas tecnologias utilizadas no desenvolvimento do sistema resultou em uma curva de aprendizado fascinante. Essa curva de aprendizado foi essencial para compreender melhor o funcionamento dos *frameworks* e bibliotecas utilizadas no desenvolvimento e conseqüentemente a implementação de novas funcionalidades.

Essa experiência enriquecedora proporcionou a aquisição de novos conhecimentos e, sem dúvida, contribuiu significativamente para o crescimento profissional e pessoal. Além do aprimoramento dos conhecimentos técnicos, houve um aprofundamento em metodologias de desenvolvimento e na aplicação de métodos ágeis de *software*. O objetivo sempre foi garantir que o sistema fosse projetado e entregue de maneira eficaz e dinâmica, seguindo boas práticas de desenvolvimento.

Os desafios enfrentados ao longo do desenvolvimento ajudaram a desenvolver soluções criativas para os obstáculos encontrados. Esse processo de aprendizado contínuo e adaptação às demandas do projeto contribuiu para o desenvolvimento de um sistema robusto e eficiente.

O mundo da gestão de projetos está em constante evolução, e a inovação é fundamental para acompanhar as demandas em mudança. O *software* desenvolvido como parte deste trabalho de conclusão de curso é um exemplo de como a tecnologia pode ser aplicada para melhorar a gestão de projetos. No entanto, ainda há muito a ser feito para aprimorar e aprofundar as funcionalidades deste *software*. A jornada de melhoria contínua é essencial na busca pela excelência na gestão de projetos, e a inovação desempenha um papel fundamental nesse processo.

5.1 TRABALHOS FUTUROS

Após a criação e elaboração da documentação do *software*, o estágio subsequente envolve a implantação e a fase de testes. Durante essa fase crucial, a ênfase recai sobre a identificação de eventuais erros ou falhas na aplicação. Os testes não são apenas um meio de validar a integridade do *software*, mas também uma oportunidade valiosa para coletar *feedbacks* significativos dos usuários. Essa interação direta com os usuários é um elemento-chave na compreensão das reais necessidades e expectativas do público.

Através desses *feedbacks*, a intenção é obter *insights* detalhados sobre a experiência do usuário, identificar áreas de melhoria e compreender os desafios enfrentados durante a utilização. Essa abordagem centrada no usuário é essencial para ajustar o *software* de maneira a atender de forma mais precisa e eficiente às demandas do público em geral. Com isso, os resultados coletados irá orientar as decisões subsequentes sobre as próximas funcionalidades a serem implementadas, garantindo que o *software* evolua de maneira coesa e em sintonia com as demandas do público-alvo.

REFERÊNCIAS

ANGULAR. **O que é angular?**. *In* Documentação. [S.l.], 2021 Disponível em: <<https://angular.io/guide/what-is-angular>>. Acesso em: 02 de nov. 2023.

AUTH0. **Introdução aos tokens JSON Web**. *In* JWT. [S.l.], 2023. Disponível em: <<https://jwt.io/introduction>>. Acesso em: 02 de nov. 2023.

BOGARD, J. **Guia de Introdução**. *In* AutoMapper Documentação. [S.l.], 2023 Disponível em: <<https://docs.automapper.org/en/stable/Getting-started.html>>. Acesso em: 02 de nov. 2023.

BUILTWITH. **Twitter Bootstrap Usage Statistics**. *In* BuiltWith. [S.l.], 2023 Disponível em: <<https://trends.builtwith.com/docinfo/Twitter-Bootstrap>>. Acesso em: 02 de nov. 2023.

CONNOLLY, T. M.; BEGG, C. E. **Database systems: a practical approach to design, implementation, and management**. 6. ed. Harlow, Essex, England: Pearson Education Limited, 2015. Citado na página 97.

EVANS, E. **Domain-Driven Design: Atacando as complexidades no coração do software**. 3. ed. Rio de Janeiro: Alta Books, 2010. Citado na página 90.

GASTER, B. *et al.* **Visão geral do ASP.NET Core SignalR**. *In* Microsoft Asp.net Core. [S.l.], 2023. Disponível em: <<https://learn.microsoft.com/pt-br/aspnet/core/signalr/introduction?view=aspnetcore-7.0>>. Acesso em: 02 de nov. 2023.

HELLER, M. **What is Visual Studio Code? Microsoft's extensible code editor**. *In* InfoWorld. [S.l.], 2022. Disponível em: <<https://www.infoworld.com/article/3666488/what-is-visual-studio-code-microsofts-extensible-code-editor.html>>. Acesso em: 01 de nov. 2023.

HUNTER, S. **Apresentando a UI do aplicativo multiplataforma .NET**. *In* .NET Blog. [S.l.], 2020. Disponível em: <<https://devblogs.microsoft.com/dotnet/introducing-net-multi-platform-app-ui/>>. Acesso em: 02 de nov. 2023.

IBM. **O que é modelagem de dados?**. *In* Tópicos IBM. [S.l.], 2023. Disponível em: <<https://www.ibm.com/br-pt/topics/data-modeling>>. Acesso em: 02 de nov. 2023.

JETBRAINS. **Rider para desenvolvimento Web full-stack**. *In* JetBrains. [S.l.], 2023. Disponível em: <https://www.jetbrains.com/pt-br/lp/rider-web/>. Acesso em: 01 de nov. 2023.

JUGRAND, H. **Aprenda O Melhor Framework De Front-End Para Impulsionar Sua Carreira Tech.** *In* Blog Awari. [S.l.], 2023. Disponível em: <https://awari.com.br/aprenda-o-melhor-framework-de-front-end-para-impulsionar-sua-carreira-tech-2/?utm_source=blog&utm_campaign=projeto+blog&utm_medium=Aprenda%20O%20Melhor%20Framework%20De%20Front-End%20Para%20Impulsionar%20Sua%20Carreira%20Tech#:~:text=O%20Bootstrap%20%C3%A9%20uma%20escolha>. Acesso em: 02 de nov. 2023.

KERZNER, H. **Project management: a Systems Approach to planning, scheduling, and Controlling.** 12. ed. Hoboken, New Jersey: John Wiley & Sons, Inc, 2017. Citado na página 25.

KNIBERG, H.; SKARIN, M. **Kanban and Scrum making the most of both.** [S.l.], C4media Inc, 2010. Citado na página 75.

LEBLANC, M. *et al.* **Consultas no LINQ to Entities.** *In* Documentação .Microsoft NET. [S.l.], 2023. Disponível em: <<https://learn.microsoft.com/pt-br/dotnet/framework/data/adonet/ef/language-reference/queries-in-linq-to-entities>>. Acesso em: 02 de nov. 2023.

LEWIS, J. P. **Project planning, scheduling & control : the ultimate hands-on guide to bringing projects in on time and on budget.** New York: Mcgraw-Hill, 2011. Citado na página 15.

LISBOA, P. **Curso de Microsoft Sql Server: Aprenda Tudo sobre Essa Poderosa Ferramenta de Banco de Dados.** *In* Blog Ciência de Dados. [S.l.], 2023 Disponível em: <https://awari.com.br/curso-de-microsoft-sql-server-aprenda-tudo-sobre-essa-poderosa-ferramenta-de-banco-de-dados/?utm_source=blog&utm_campaign=projeto+blog&utm_medium=Curso%20de%20Microsoft%20Sql%20Server:%20Aprenda%20Tudo%20sobre%20Essa%20Poderosa%20Ferramenta%20de%20Banco%20de%20Dados>. Acesso em: 02 de nov. 2023.

LIKER, J. K. **O modelo Toyota 14 Princípios de gestão do maior fabricante do mundo.** Portol Alegre: Bookman, 2021. Citado na página 75.

MATHIS, B. **Gerenciamento Ágil de Projetos Para Iniciantes: Dominando o Básico com o Scrum.** [S.l.] Babelcube Inc., 2018. Citado na página 22.

MATTOS, A. N. **Informação é Prata, Compreensão é Ouro.** [S.l.:s.n], 2009. Citado na página 35.

TINNIRELLO, P. C. **New directions in project management.** Boca Raton, Fla.: CRC Press LLC, 2001. Citado na página 52.

MDN. **HTML: Linguagem de Marcação de Hipertexto.** *In* MDN Documentação para Desenvolvedores. [S.l.], 2023. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTML>>. Acesso em: 02 de nov. 2023.

MICROSOFT. **Documentação do Visual Studio Code**. *In* Visual Studio Code. [S.l.], 2023a. Disponível em: <<https://code.visualstudio.com/docs>>. Acesso em: 01 de nov. 2023.

MICROSOFT. **.NET | Crie. Teste. Implante**. *In* .NET. [S.l.], 2023b. Disponível em: <<https://dotnet.microsoft.com/pt-br/>>. Acesso em: 02 de nov. 2023.

MICROSOFT. **Hub de documentação do Entity Framework**. *In* Microsoft .NET. [S.l.], 2023c. Disponível em: <<https://learn.microsoft.com/pt-br/ef/>>. Acesso em: 02 de nov. 2023.

MICROSOFT. **TypeScript para o novo programador**. *In* Documentação. [S.l.], 2023d. Disponível em: <<https://www.typescriptlang.org/docs/handbook/typescript-from-scratch.html>>. Acesso em: 02 de nov. 2023.

MIRANDA, G. **Como desenvolver um Aplicativo com Metodologia Ágil**. [S.l.:s.n] , 2023. Citado na página 21.

PMI. **Um guia do conhecimento em gerenciamento de projetos (GUIA PMBOK)**. 6.ed. Newtown Square, Pennsylvania: Project Management Institute, Inc, 2017. Citado na página 14,15,19,20,23,24,66,79.

PMI. **PMI Pulse of the Profession® 2018**. *In* Project Management Institute. [S.l.], 2018. Disponível em: <https://www.pmi.org/learning/thought-leadership/pulse/pulse-of-the-profession-2018>. Acesso em: 28 de out. 2023.

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software**. 9.ed. Rio Grande do Sul: Porto Alegre, McGraw Hill Brasil, 2021. Citado na página 23,27,28.

ROVEDA, U. **TYPESCRIPT: O QUE É, COMO COMEÇAR E QUAIS SÃO AS VANTAGENS?**. *In* Blog Kenzie. [S.l.], 2023. Disponível em: <<https://kenzie.com.br/blog/typescript/>>. Acesso em: 02 de nov. 2023.

SCHEIDT, F. A. **Fundamentos de CSS: criando design para sistemas web**. Paraná: Foz do Iguaçu: Outbox Livros Digitais, 2015. Citado na página 92.

SILVA, M. S. **Bootstrap 3.3.5 - Aprenda a usar o framework Bootstrap para criar layouts css complexos e responsivos**. São Paulo: Novatec Editora, 2023. Citado na página 93

SMARTBEAR. **Sobre Swagger**. *In* Swagger. [S.l.], 2023. Disponível em: <<https://swagger.io/solutions/api-documentation/>>. Acesso em: 02 de nov. 2023.

SOMMERVILLE, I. **Engenharia de software**. São Paulo: Pearson Prentice Hall, 2011. Citado na página 27,28.

SOUZA, B. **Métodos Ágeis e Gestão de Serviços de TI**. Rio de Janeiro: Brasport, 2018. Citado na página 23.

STACK OVERFLOW. **Stack Overflow Developer Survey 2021**. In STACK OVERFLOW. [S.l.], 2021 Disponível em: <<https://insights.stackoverflow.com/survey/2021>>. Acesso em: 02 de nov. 2023.

SUTHERLAND, J. **Scrum: a arte de fazer o dobro do trabalho na metade do tempo**. 2.ed. São Paulo, Leya, 2016. Citado na página 23.

TARTARI, T. **AutoMapper no .NET: Você deve Priorizar Performance ou Facilidade?**. In Tiago Tartari Consultoria. [S.l.], 2023 Disponível em: <<https://tiagotartari.net/automapper-no-net-voce-deve-priorizar-performance-ou-facilidade.html>>. Acesso em: 02 de nov. 2023.

TORTELLI, A. **Digitalização empresarial: o futuro dos negócios está aqui**. In Consultoria Digital Conex Negócios. [S.l.], 2018. Disponível em: <https://conexnegocios.com.br/digitalizacao-empresarial-o-futuro-dos-negocios-esta-aqui/>. Acesso em: 22 out. 2023.

VARGAS, R. V. **Gerenciamento de Projetos**. 9.ed. Rio de Janeiro Brasport, 2018. Citado na página 14.

VAZQUEZ, C. E.; SIMÕES, G. S. **Engenharia de Requisitos**. Rio de Janeiro: Brasport, 2016. Citado na página 29.

WAZLAWICK, R. **Engenharia de Software - Conceitos e Práticas**. 2.ed. Rio de Janeiro: Gen Ltc, 2019. Citado na página 23.

ZELDMAN, J.; MARCOTTE, E. **Designing with Web standards**. Berkeley, Calif.: New Riders, 2010. Citado na página 92.