



**Gerenciamento de Projeto de Software (11925/31)**  
**Prof. Dr. Renato Balancieri**

**Gerenciamento de projeto de software:**  
**Verde Gestão**

Gabriel Saraiva	RA 129145
Olga Maria	RA 130002
Rafael Tavares	RA 129969
Yasser Farid	RA 129706

## **1. Introdução**

O propósito do projeto em desenvolvimento trata-se de um sistema de gestão de documentos, criado para atender às necessidades do Grupo Escoteiro Verde Canção. O principal objetivo da aplicação é centralizar, organizar e facilitar o controle de solicitações de assinaturas de documentos internos, análise de documentos e acompanhamento do status de cada solicitação. Entre as funcionalidades principais, destacam-se a criação de solicitações, o controle de prazos, fluxo de assinaturas e a criação de avisos dentro do sistema.

A condução do projeto seguiu uma abordagem colaborativa e organizada. A equipe trabalhou em conjunto em algumas atividades como levantamento de requisitos, diagramas, em outras atividades a equipe foi dividida onde cada membro ficou responsável por tarefas específicas, como geração de relatório, arquitetura do software e desenvolvimento. Durante o processo, ocorreram encontros periódicos, tanto presenciais quanto virtuais, entre a equipe e com o cliente, para discutir o andamento das tarefas, validar decisões técnicas e garantir o alinhamento entre todos os envolvidos.

Para auxiliar na organização e comunicação, a equipe fez uso do Whatsapp para comunicação, Google Workspace como diretório e criação de documentos e Notion para gerenciamento de atividades que facilitou o acompanhamento das atividades e a atualização constante de todos os integrantes. Essa abordagem permitiu um desenvolvimento mais estruturado, com melhor controle de prazos e um fluxo de trabalho mais eficiente, contribuindo para o alcance dos objetivos do projeto.

## 2. Análise de Riscos

ID	RISCO	PROBABILIDADE	IMPACTO
001	Requisitos levantados não atendem às necessidades	Alta	4
002	Fluxo de documentos muito alto	Alta	3
003	O BD utilizado não processa tantas transações quanto o esperado.	Alta	3
004	Equívoco na compreensão e modelagem do banco de dados	Alta	3
005	Sistema com difícil adaptação por parte dos usuários (curva de nível muito alta)	Muito Alta	3
006	Maior esforço do que o previsto pela inexperiência técnica da equipe	Muito Alta	2
007	Erro na gestão de hierarquia, permissões e níveis de acesso dos diferentes usuários do sistema	Muito Alta	2
008	Tempo necessário para desenvolver é subestimado	Muito Alta	2
009	Sistema não atende às questões de acessibilidade esperadas pelo cliente	Muito Alta	1
010	Funcionamento do software muito distante da realidade do público alvo, baixa adesão e alta rejeição ao produto criado	Moderada	4
011	Dificuldade de contato e feedback com o cliente	Moderada	3
012	Testes insuficientes, com diversos erros lógicos e bugs no produto final	Moderada	3
013	Problemas para hospedar o sistema	Moderada	2
014	Problemas com os prazos	Moderada	1
015	Mudanças no escopo durante o andamento do projeto	Baixa	2
016	Problemas inesperados de integração e compatibilidade entre as tecnologias escolhidas para o desenvolvimento	Baixa	2

**Probabilidade:** Baixa, Moderada, Alta, Muito Alta

**Impacto:**

- 1 - insignificante
- 2 - tolerável
- 3 - grave
- 4 - catastrófico

**Gerenciamento dos riscos mais significativos:**

ID DO RISCO	PROB: X %	IMPACTO: X
<b>Descrição:</b>	Equívoco na compreensão e modelagem do banco de dados	
<b>Estratégia de Prevenção</b>	<ol style="list-style-type: none"><li>1. Realizar periodicamente reuniões com o cliente, para levantar corretamente os requisitos e quais dados precisam ser armazenados</li><li>2. Criar um Modelo-ER e validar com a equipe antes de aplicar</li><li>3. Documentar decisões de modelagem</li></ol>	
<b>Estratégia de minimização</b>	<ol style="list-style-type: none"><li>1. Testar o banco com dados fictícios para garantir que está atendendo aos requisitos.</li></ol>	
<b>Plano de contingência</b>	<ol style="list-style-type: none"><li>1. Identificar a falha.</li><li>2. Reunir equipe e revisar o modelo.</li><li>3. Corrigir a estrutura e aplicar testes.</li><li>4. Restaurar backup se necessário.</li><li>5. Atualizar documentação.</li></ol>	
<b>Estado Atual</b>	Iniciada a etapa de prevenção	

ID DO RISCO	PROB: X %	IMPACTO: X
<b>Descrição:</b>	Fluxo de documentos muito alto	
<b>Estratégia de Prevenção</b>	<ol style="list-style-type: none"><li>1. Analisar detalhadamente os tipos de documentos e os fluxos previstos entre os níveis hierárquicos.</li><li>2. Definir e implementar regras claras de validação, triagem e delegação automática para otimizar o fluxo.</li></ol>	
<b>Estratégia de minimização</b>	<ol style="list-style-type: none"><li>1. Implementar mecanismos de controle de fila de e notificações para evitar gargalos, além de relatórios de acompanhamento em tempo real para o administrador monitorar gargalos e</li></ol>	

	atrasos.
<b>Plano de contingência</b>	1. Permitir o escalonamento automático de documentos parados por muito tempo e definir perfis de backup para assumir o fluxo quando um cargo (perfil) chave estiver inativo
<b>Estado Atual</b>	Iniciada a etapa de minimização

ID DO RISCO	PROB: X %	IMPACTO: X
<b>Descrição:</b>	Requisitos levantados não atendem às necessidades	
<b>Estratégia de Prevenção</b>	<ol style="list-style-type: none"> <li>1. Realizar encontros/reuniões frequentes com o cliente e os usuários finais, a fim de entender integralmente suas necessidades e opiniões a respeito do projeto.</li> <li>2. Garantir uma documentação clara e robusta acerca dos requisitos levantados (ao menos dos mais importantes), de modo a se ter uma base sólida com o que trabalhar ao longo do projeto.</li> </ol>	
<b>Estratégia de minimização</b>	<ol style="list-style-type: none"> <li>1. Priorização dos requisitos mais estimados pelo cliente, de que modo que, ao menos, as principais funções do sistema estejam de acordo com o esperado.</li> <li>2. Validação constante do software pelo cliente, garantindo ciclos, ainda que curtos, de aprovação da maneira que os requisitos estão sendo implementados</li> </ol>	
<b>Plano de contingência</b>	<ol style="list-style-type: none"> <li>1. Interromper o desenvolvimento dos requisitos entendidos erroneamente e iniciar-se um novo momento de elicitação e análise dos requisitos do cliente.</li> <li>2. Realizar ajustes no cronograma de desenvolvimento, de modo a evitar muitos atrasos e garantir mais tempo para essa nova fase de elicitação.</li> </ol>	
<b>Estado Atual</b>	Etapa de minimização	

ID DO RISCO	PROB: X %	IMPACTO: X
<b>Descrição:</b>	O BD utilizado não processa tantas transações quanto o esperado.	
<b>Estratégia de Prevenção</b>	<ol style="list-style-type: none"> <li>1. Modelar o banco de dados com estratégias de normalização e boas práticas relacionais.</li> <li>2. Optar pelo uso extensivo de ferramentas fornecidas pelo SGBD utilizado, como “views materializadas” que funcionam como uma espécie de cache para as transações.</li> <li>3. Realizar benchmarks e testes de estresse para avaliar constantemente o uso médio do processador e disco pelo sistema desenvolvido.</li> </ol>	
<b>Estratégia de minimização</b>	<ol style="list-style-type: none"> <li>1. Otimizar data queries para que materializem menos dados na memória.</li> <li>2. Reconfigurar o servidor para um uso mais dedicado de seus recursos para o banco de dados.</li> </ol>	
<b>Plano de contingência</b>	<ol style="list-style-type: none"> <li>1. Optar por transferir o sistema para a hospedagem de terceiros, como a Amazon AWS, assim permitindo acesso à máquinas mais robustas.</li> <li>2. Configurar a máquina virtual conforme o resultado dos benchmarks anteriormente realizados.</li> </ol>	
<b>Estado Atual</b>	Etapa de prevenção	

### 3. Recursos Físicos

#### Hardware e Infraestrutura

Recurso	Sobre	Custo p/ Pessoa	Observação
<b>Serviço de Internet</b>	Conexão residencial de cada membro.	R\$ 0,00	Custo indireto (todos os membros da equipe já comissionam este custo independente do projeto)
<b>Computador</b> (máquina Linux ou Windows)	Computador pessoal do membro da equipe.	R\$ 0,00	Custo indireto (ferramentas já pertencentes aos membros da equipe)
<b>Periféricos</b> (mouse, teclado, etc.)	Periféricos pessoais para uso doméstico.	R\$ 0,00	Inclusos no setup de cada membro e de preferência pessoal.
<b>Dispositivo Móvel</b> (celular, tablet, etc.)	Dispositivo móvel pessoal do membro da equipe para testar e acessar remotamente o sistema.	R\$ 0,00	Todos da equipe possuem pelo menos um.

#### Recursos Virtuais

Recurso	Sobre	Custo p/ Pessoa	Observação
<b>Java</b>	Linguagem de programação para o desenvolvimento da API.	R\$ 0,00	Linguagem gratuita mantida pela Oracle.
<b>PostgreSQL</b>	Sistema de gerenciamento de banco de dados.	R\$ 0,00	Banco de dados relacional gratuito amplamente utilizado.
<b>Apache Tomcat</b>	Software para host de sistema web.	R\$ 0,00	Gratuito, open-source e disponibilizado pela Apache Foundation.
<b>IntelliJ</b>	IDE para desenvolvimento em Java.	R\$ 0,00	IDE paga, porém estudantes da Universidade Estadual de Maringá podem receber uma licença estudantil para realização de projetos acadêmicos.
<b>Java Hibernate</b>	Framework ORM que integra o Java com	R\$ 0,00	Framework gratuito que consegue integrar o

	diversos bancos de dados.		PostgreSQL com o JavaDBC.
<b>Java Spring Boot</b>	Framework do Java que facilita o desenvolvimento de APIs RESTful e evita o código “boiler-plate”	R\$ 0,00	Framework gratuito.
<b>Flutter</b>	Framework baseado na linguagem Dart.	R\$ 0,00	Desenvolvido e disponibilizado gratuitamente pela Google para criação de softwares responsivos.
<b>VSCode</b>	IDE para desenvolvimentos gerais.	R\$ 0,00	Ideal para trabalhar com Flutter.
<b>Figma</b>		R\$ 0,00	Utilizamos o plano gratuito.
<b>Canva</b>		R\$ 0,00	Utilizado para apresentações e protótipos apresentados aos stakeholders.
<b>Google Drive</b>	Serviço em nuvem para armazenamento.	R\$ 0,00	Utilizamos o plano estudantil oferecido pela parceria da Universidade Estadual de Maringá com a Google.
<b>Anydesk</b>	Serviço de acesso remoto para facilitar implantação.	R\$ 0,00	Possui licença gratuita somente para acesso de máquina a distância.



#### 4. Estimativa de Custo e de Esforço

A técnica de estimativa de custo e esforço por ponto de casos de uso foi utilizada para prever o trabalho, tempo e custos necessários com base na complexidade dos casos de uso. O diagrama de casos de uso utilizado para o cálculo consta no apêndice A.

##### Complexidade dos atores (CA):

Tipo de interação	Pontos	Nº de atores	Resultado
Atores humanos	3	4	12
Sistemas TCP/CMD	2	0	0
APIs	1	0	0

CA = 12

##### Complexidade de Casos de Uso (CCU)

Complexidade	Pontos	Nº de caso de uso	Resultado
Simples	5	7	35
Médio	10	3	30
Complexo	15	0	0

CCU = 65

##### Pontos por Caso de Uso Não Ajustados:

<b>PCUNA = 12 + 65 = 77</b>
-----------------------------

Ajustes técnicos e ambientais

##### Fatores técnicos:

Sigla	Peso	Influência	Resultado
T1	2	0	0
T2	2	3	6
T3	1	5	5
T4	1	1	1
T5	1	0	0

T6	0,5	4	2
T7	0,5	2	1
T8	2	1	2
T9	1	5	5
T10	1	0	0
T11	1	2	2
T12	1	0	0
T13	1	3	3

TFactor = 27

**Technical Complexity Factor:**

$TCF = 0.6 + (0.01 * 27) = 0,87$
----------------------------------

**Fatores Ambientais:**

Sigla	Peso	Influência	Resultado
E1	1,5	3	4,5
E2	0,5	4	2
E3	1	3	3
E4	0,5	4	2
E5	1	2	2
E6	2	3	6
E7	-1	4	-4
E8	-1	4	-4

EFactor = 11,5

**Environment Factor:**

$EF = 1.4 - (0,03 * 11.5) = 1,055$
------------------------------------

**Pontos de Casos de Uso Ajustado:**

$PCUA = PCUNA * TCF * EF$
$PCUA = 77 * 0,87 * 1,055 = 70,69$

Foi adotado o índice de produtividade de 20 horas, esse valor foi escolhido com base em convenções da literatura da engenharia de software, sendo considerado um valor médio ideal para equipes.

O sistema possui um total estimado de 71 pontos de caso de uso, o que resulta em um esforço de tempo de 1.420 horas ( $\text{Esforço} = \text{PCUA} \times \text{Índice de Produtividade} = 71 \times 20$ ). Considerando uma equipe de 4 pessoas, o esforço total será dividido entre os integrantes, temos 355 horas para cada integrante e o tempo estimado de desenvolvimento é de 4 meses, assim são cerca de 3 horas de trabalho por dia para cada integrante para que entreguemos o projeto no prazo com os requisitos levantados.

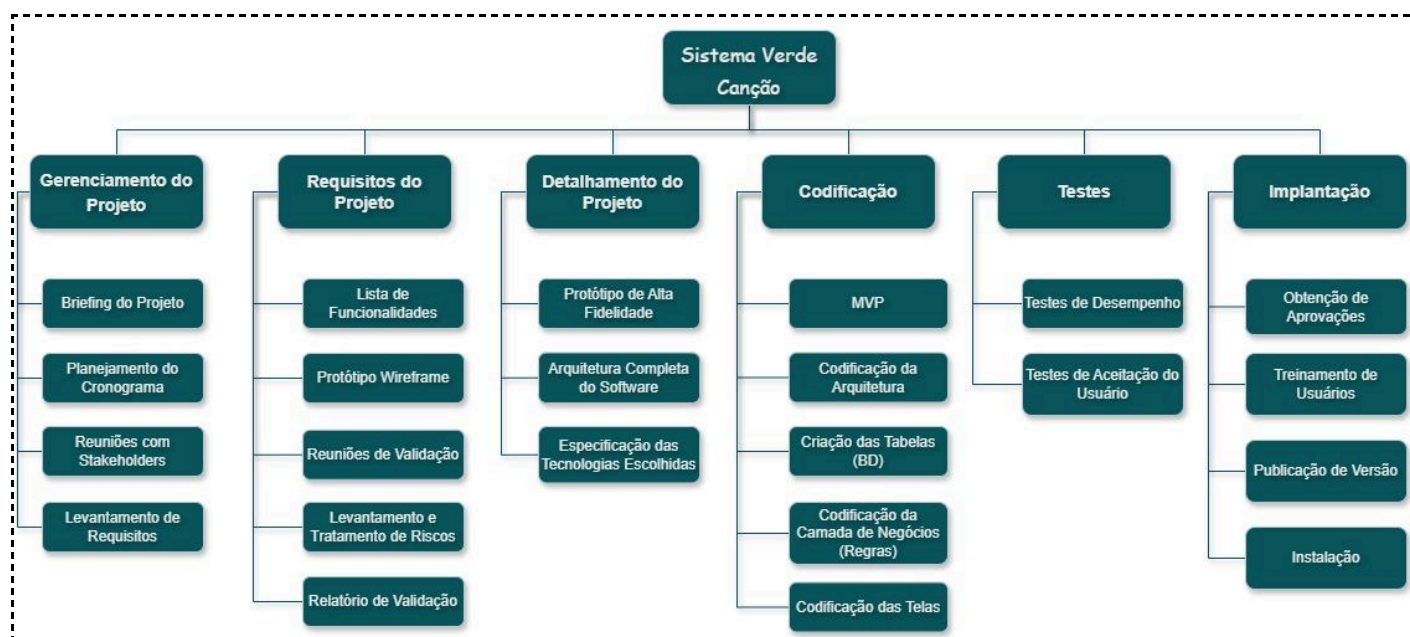
Para calcular os custos de recursos humanos, foi considerado um valor de R\$ 35,00 por hora de trabalho. Assim, o custo total estimado de mão de obra é:

- Total de horas: 1.420 h
- Custo por hora: R\$ 35,00
- Custo total:  $1.420 \times 35 = \text{R\$ } 49.700,00$

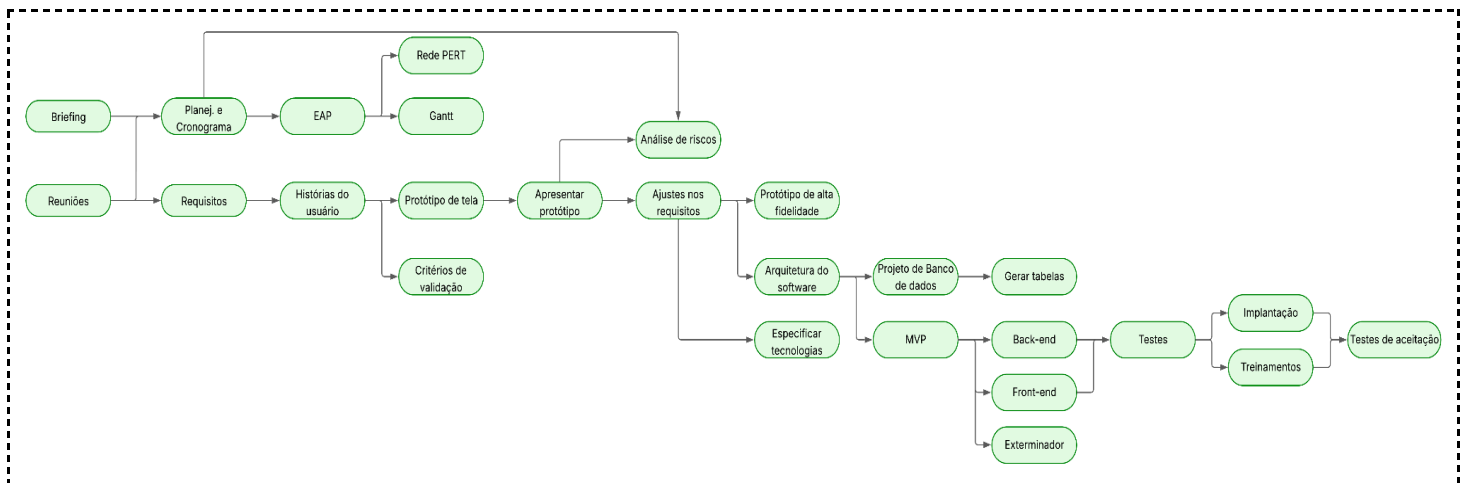
Portanto, o custo estimado com recursos humanos para todo o projeto é de R\$49.700,00, considerando o esforço total necessário para concluir o sistema dentro do prazo previsto.

## 5. Gerenciamento e Cronograma de Projeto

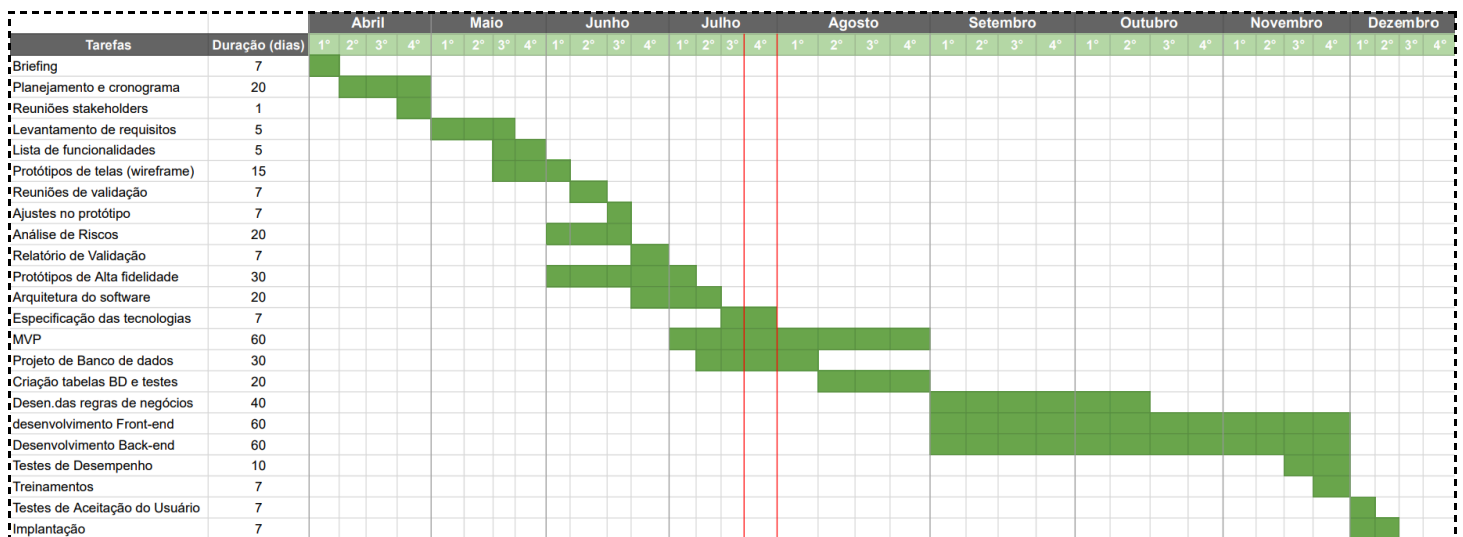
### 5.1. Estrutura Analítica de Projeto (EAP)



## 5.2. Rede PERT



## 5.3. Diagrama de Gantt



## 6. Política de Gerenciamento do Trabalho

A fim de manter a organização, controle e rastreabilidade dos artefatos a equipe fez o uso de diversas ferramentas que serão citadas abaixo conforme seu propósito de uso.

### 6.1. Gerenciamento de Configuração de Artefatos:

- Notion: utilizado para criação do cronograma, onde os principais artefatos foram estruturados com prazos, filtros, estados entre outras especificações.
- Google Drive: utilizado para armazenamento dos principais arquivos criados, por exemplo, briefing do projeto, documentação contendo as principais

funcionalidades, registros das reuniões com os stakeholders, relatórios de validação, principais links de outros arquivos do projeto.

- Draw.io e LucidChart: plataformas utilizadas para elaboração dos principais diagramas e estruturas do projeto.
- Figma: tecnologia utilizada para versionamento dos protótipos de tela de alta fidelidade.

## 6.2 Controle de Versões:

- Para controle dos códigos foi utilizado a ferramenta Git/Github

Para construção do mínimo produto viável (MVP) foi utilizado somente uma branch principal, onde os contribuintes integravam seus commits (cada adição, alteração ou remoção de funcionalidades).

**Link do Repositório do código-fonte:** [github.com/SGD\\_verde\\_gestao](https://github.com/SGD_verde_gestao)

Para o futuro desenvolvimento, será aplicado boas práticas de versionamento, onde a partir da branch principal cada contribuinte deverá criar uma nova branch para cada ciclo de iteração ou funcionalidade adicionada, e essa branch só será mergeada (integrada na principal) a partir da revisão e aprovação de dois desenvolvedores. Para melhor controle, os merge requests deverão seguir um padrão, com ID da tarefa, nome da funcionalidade adicionada e descrição do que foi feito.

## 7. Considerações Finais

Apresentamos o gerenciamento do projeto "Verde Gestão", um sistema de gerenciamento de documentos desenvolvido para o grupo de escoteiros. Este sistema tem como finalidade centralizar, organizar e otimizar o controle de solicitações de assinaturas de documentos internos. Para atingir esses objetivos, adotamos uma abordagem colaborativa, com reuniões regulares e utilizando diversas ferramentas para gerenciar os artefatos. A aplicação da metodologia de gerenciamento de projetos nos guiou para uma forte estruturação do desenvolvimento, assegurando um controle mais eficaz dos prazos e do fluxo de trabalho, contribuindo diretamente para o cumprimento das metas.

A análise de riscos nós ajudou a identificar dezesseis possíveis problemas, assim conseguimos tratar estratégias para prevenir, minimizar e tratar caso ocorram, isso nos ensinou a um olhar analítico sobre o projeto, conseguimos identificar o que era e que não era um risco de fato. A estimativa de custo e esforço, baseada na técnica de pontos por caso de uso, revelou um esforço total de 1.420 horas, com um custo de mão de obra de R\$ 49.700,00, tais estimativas são essenciais para entendermos a dimensão do projeto, bem como a criação de um cronograma factível, que vai de abril a dezembro, conforme mostrado pelo EAP, Rede PERT e Diagrama de Gantt.

A importância da aplicação do gerenciamento de projetos é inegável para o sucesso do desenvolvimento de software. A antecipação e mitigação de riscos, o cálculo de recursos e prazos, e a organização em ferramentas de gerenciamento de tarefas e controle de versões, como Git para o código-fonte, garantem um processo fortemente estruturado, eficiente e transparente. A gestão sistemática de todas as fases do projeto não só minimiza desvios e otimiza os recursos, mas também eleva a probabilidade de entregar um produto final de alta qualidade que atenda às expectativas do cliente.

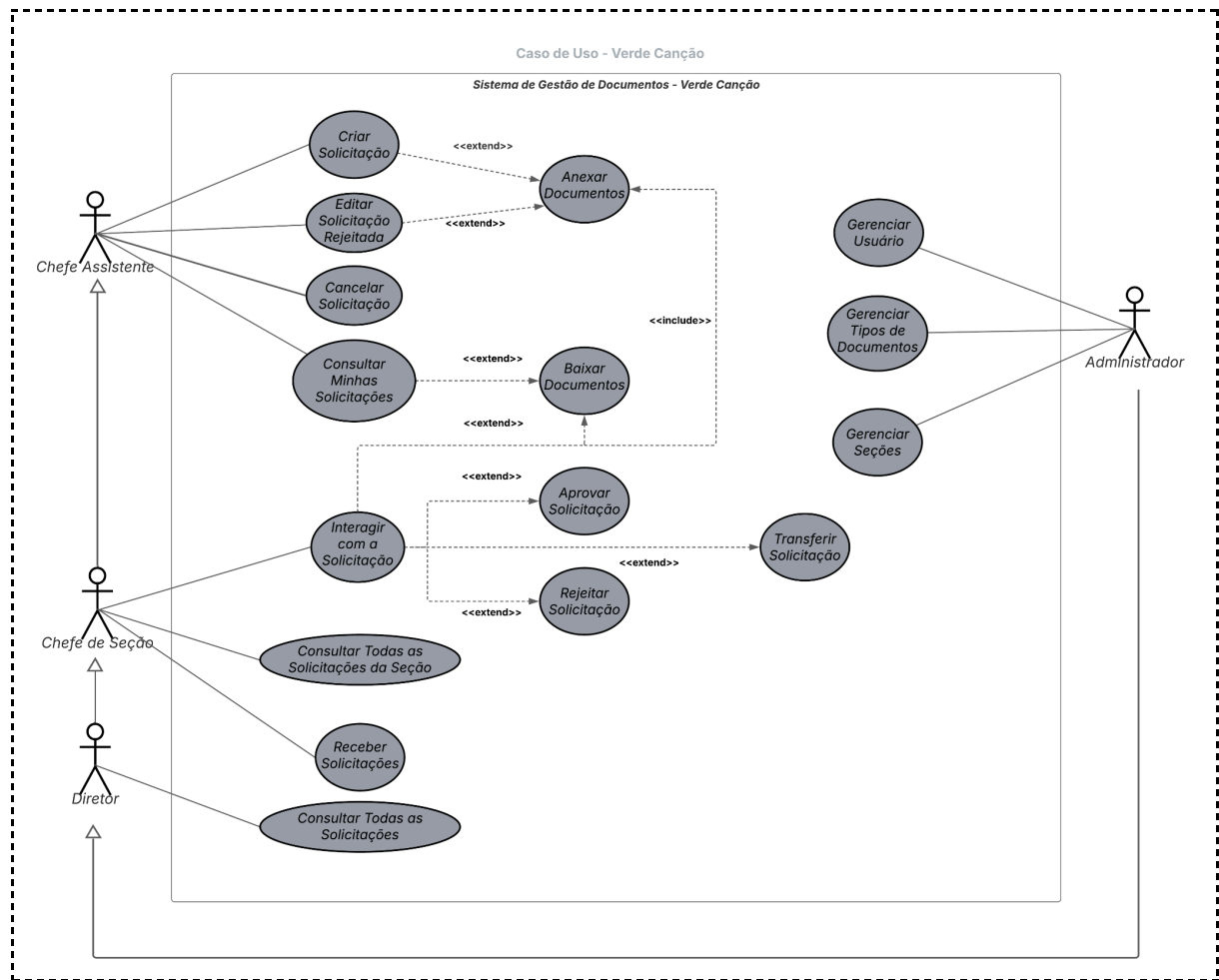
## **8. Referências Bibliográficas:**

CASTELÃO, E.; QUEIROZ DA SILVA, J.; DE MESQUITA, M. Processo de estimativa de software com a métrica de use case point PMBOK e RUP. [s.l: s.n.].Disponível em: <[https://www.unip.br/eceeic/admin/Anexos/Conteudo/C2011/C8/file\\_1682011170401406.pdf](https://www.unip.br/eceeic/admin/Anexos/Conteudo/C2011/C8/file_1682011170401406.pdf)>. Acesso em: 20 jul. 2025.

TATHIANE. Quanto ganha um desenvolvedor de software no Brasil. Disponível em: <<https://www.rj.senac.br/noticias/tecnologia-da-informacao/quanto-ganha-um-desenvolvedor-de-software>>. Acesso em: 20 jul. 2025.

Microlins Cursos Profissionalizantes. Disponível em: <<https://www.microlins.com.br/blog/mercado-de-trabalho/quanto-ganha-um-programador-no-brasil/>>. Acesso em: 30 jul. 2025.

## APÊNDICE A – Diagrama de casos de usos



## APÊNDICE B – Projeto de Banco de Dados

