

FCT - Faculdade de Ciências e Tecnologia  
DMC - Departamento de Matemática e Computação  
Bacharelado em Ciência da Computação

Engenharia de Software 2  
Projeto 3 - Sistema de Gestão de Churrascos Entre Amigos

## **Plano de Projeto de Software**

Título do Sistema:

**Sistema de Gestão de Churrascos Entre Amigos**  
**Pagamento Mensal e Controle de Despesas por Churrasco**

Gerentes de Projeto:	Julio Ricardo Igor
SQA (Garantia de Qualidade):	Luiz Sara
A&P (Análise e Projeto):	Augusto Pedro Coleta
COD (Codificação):	Guilherme Carrara Guilherme Digiori
Professor:	Prof. Dr. Rogério Garcia
Disciplina:	Engenharia de Software 2

Presidente Prudente, 1 de outubro de 2025

# Conteúdo

<b>1</b>	<b>Resumo</b>	<b>5</b>
1.1	Objetivos Principais	5
1.2	Escopo do Projeto	5
1.3	Cronograma Resumido	5
1.4	Equipe do Projeto	6
1.5	Principais Riscos Identificados	6
1.6	Tecnologias e Metodologias	6
<b>2</b>	<b>Introdução</b>	<b>6</b>
2.1	Escopo e Propósito deste Documento	6
2.2	Objetivos e Metas	6
2.2.1	Objetivos Específicos do Sistema	7
2.2.2	Funcionalidades Principais	7
2.2.3	Público-Alvo	7
2.2.4	Benefícios Esperados	7
<b>3</b>	<b>Avaliação de Riscos</b>	<b>8</b>
3.1	Análise de Riscos	8
3.2	Gerenciamento de Riscos	8
3.2.1	Estratégias de Mitigação	8
3.2.2	Indicadores de Alerta	9
3.2.3	Plano de Contingência	9
3.2.4	Monitoramento e Controle	10
<b>4</b>	<b>Estimativas, Cronogramas e Prazos</b>	<b>10</b>
4.1	Estimativas	10
4.1.1	Dados Históricos Utilizados	10
4.1.2	Técnicas de Estimativa Empregadas	10
4.1.3	Estimativas por Sprint	10
4.2	Cronograma	11
4.2.1	Estrutura de Trabalho	11
4.2.2	Legenda	11
4.2.3	Cronograma Detalhado	11
4.3	Gráfico de Gantt	12
4.3.1	Caminho Crítico	13
4.4	Marcos Principais	13
4.5	Entregáveis por Sprint	13
4.6	Critérios de Aceitação	13
<b>5</b>	<b>Organização da Equipe</b>	<b>13</b>
5.1	Estrutura da Disciplina	13
5.2	Estrutura da Equipe para Este Projeto	14
5.3	Matriz de Responsabilidades	15
5.4	Alocação por Sprint	15
5.4.1	Sprint 0 - Preparação	15
5.4.2	Sprint 1 - Desenvolvimento Inicial	15
5.4.3	Sprint 2 - Desenvolvimento Avançado	16

5.4.4	Sprint 3 - Finalização . . . . .	16
5.5	Comunicação e Coordenação . . . . .	16
5.5.1	Reuniões Regulares . . . . .	16
5.5.2	Ferramentas de Gestão . . . . .	16
5.6	Relatórios de Gestão . . . . .	17
5.6.1	Estrutura de Reportes . . . . .	17
5.6.2	Métricas de Acompanhamento . . . . .	17
5.7	Responsabilidades Específicas dos Gerentes . . . . .	17
5.8	CrITÉrios de Qualidade e Entrega . . . . .	18

# 1 Resumo

Este documento apresenta o plano completo para o desenvolvimento do Sistema de Gestão de Churrascos Entre Amigos, um software projetado para facilitar a organização, gestão financeira e coordenação de eventos sociais entre grupos de amigos.

O projeto será desenvolvido seguindo uma metodologia ágil com 4 sprints bem definidos, totalizando 86 dias úteis de desenvolvimento. O sistema contempla funcionalidades abrangentes desde o cadastro de usuários até a prestação de contas detalhada dos eventos.

## 1.1 Objetivos Principais

- Automatizar o processo de organização de churrascos
- Facilitar o controle financeiro e divisão de despesas
- Implementar sistema de convites e confirmações automatizadas
- Proporcionar transparência na prestação de contas
- Ofertar cálculo automático de lista de compras

## 1.2 Escopo do Projeto

O sistema incluirá módulos para:

- Gestão de usuários e autenticação
- Criação e gerenciamento de eventos
- Sistema de convites automatizado
- Controle de pagamentos e check-in
- Cálculo automático de lista de compras
- Sistema de prestação de contas
- Relatórios financeiros detalhados

## 1.3 Cronograma Resumido

- **Sprint 0 (Preparação):** 27 dias - Documentação e planejamento
- **Sprint 1 (Desenvolvimento Inicial):** 26 dias - Cadastros e funcionalidades básicas
- **Sprint 2 (Desenvolvimento Avançado):** 23 dias - Sistema financeiro e cálculos
- **Sprint 3 (Finalização):** 10 dias - Testes finais e entrega

**Duração Total:** 86 dias úteis

## 1.4 Equipe do Projeto

O projeto será desenvolvido por uma equipe multidisciplinar composta por:

- **Gerentes:** Julio, Ricardo, Igor
- **SQA (Garantia de Qualidade):** Luiz e Sara
- **A&P (Análise e Projeto):** Augusto e Coleta
- **COD (Codificação):** Guilherme Carrara e Guilherme Digiori

## 1.5 Principais Riscos Identificados

- Atrasos no cronograma devido à complexidade técnica
- Indisponibilidade de recursos humanos
- Mudanças de escopo durante o desenvolvimento
- Dificuldades de integração entre módulos

## 1.6 Tecnologias e Metodologias

O projeto utilizará uma abordagem ágil com foco em entregas incrementais, permitindo validação contínua dos requisitos e adaptação às necessidades do cliente. A arquitetura será modular para facilitar manutenção e escalabilidade futura.

# 2 Introdução

## 2.1 Escopo e Propósito deste Documento

O propósito deste documento é fornecer uma visão geral do processo de desenvolvimento do Sistema de Gestão de Churrascos Entre Amigos. Ele visa esclarecer as atividades, recursos e pessoas envolvidas, bem como os prazos e cronogramas que vinculam esses elementos.

Este plano de projeto serve como base fundamental para o gerenciamento eficaz do desenvolvimento, estabelecendo marcos claros, estimativas precisas e definindo a alocação de recursos necessários para o sucesso do projeto.

## 2.2 Objetivos e Metas

O objetivo deste documento é ser uma base sólida sobre a qual os gerentes deste projeto possam confiar, fornecendo-lhes marcos claros, estimativas e disponibilidade de recursos, para que a equipe possa ser gerenciada com confiança e alocada para as atividades necessárias.

### 2.2.1 Objetivos Específicos do Sistema

1. **Automatização de Processos:** Eliminar processos manuais de organização de eventos sociais
2. **Transparência Financeira:** Proporcionar controle total sobre receitas e despesas
3. **Facilidade de Uso:** Interface intuitiva para usuários de diferentes perfis técnicos
4. **Escalabilidade:** Suporte para diferentes tamanhos de eventos e grupos
5. **Confiabilidade:** Sistema robusto com 99% de disponibilidade

### 2.2.2 Funcionalidades Principais

O Sistema de Gestão de Churrascos Entre Amigos contemplará as seguintes funcionalidades principais:

- **Gestão de Usuários:** Cadastro, autenticação e controle de perfis
- **Criação de Eventos:** Interface para criação e configuração de churrascos
- **Sistema de Convites:** Convites automatizados via e-mail
- **Controle de Participantes:** Gerenciamento de confirmações e status
- **Gestão Financeira:** Controle de pagamentos e divisão de custos
- **Lista de Compras Inteligente:** Cálculo automático baseado em parâmetros
- **Prestação de Contas:** Relatórios detalhados de receitas vs despesas
- **Check-in de Eventos:** Controle de presença vinculado ao pagamento

### 2.2.3 Público-Alvo

O sistema é direcionado para:

- **Organizadores de Eventos:** Pessoas responsáveis por coordenar churrascos entre amigos
- **Participantes:** Amigos que participam dos eventos e precisam de transparência financeira
- **Administradores:** Usuários com privilégios para configuração e moderação do sistema

### 2.2.4 Benefícios Esperados

1. **Redução de Conflitos:** Transparência na divisão de custos
2. **Economia de Tempo:** Automatização de tarefas repetitivas
3. **Maior Participação:** Facilidade para convidar e confirmar presença
4. **Controle Financeiro:** Visibilidade completa de receitas e despesas
5. **Histórico de Eventos:** Registros para consultas futuras

## 3 Avaliação de Riscos

### 3.1 Análise de Riscos

O principal risco ao qual este projeto está sujeito é o atraso no seu desenvolvimento, fazendo com que não seja entregue nos prazos estabelecidos.

Riscos específicos identificados:

1. **Risco Temporal:** Atrasos no cronograma devido à complexidade técnica
2. **Risco de Recursos:** Indisponibilidade de membros da equipe
3. **Risco Técnico:** Dificuldades na integração com sistemas externos
4. **Risco de Escopo:** Mudanças de requisitos durante o desenvolvimento
5. **Risco de Qualidade:** Falhas na validação e testes do sistema
6. **Risco de Comunicação:** Falhas na coordenação entre equipes multidisciplinares
7. **Risco Tecnológico:** Obsolescência ou incompatibilidade de ferramentas

### 3.2 Gerenciamento de Riscos

Para mitigar os riscos mencionados, os gerentes devem fazer uma estimativa precisa de cronograma que englobe todas as atividades necessárias que farão o software ser desenvolvido como um produto completo e de alta qualidade.

#### 3.2.1 Estratégias de Mitigação

- **Controle Rigoroso do Cronograma:** Como todas as tarefas estão no caminho crítico, qualquer atraso impacta diretamente a entrega final
- **Acompanhamento de Marcos:** Monitoramento semanal do progresso de cada sprint
- **Recursos de Contingência:** Identificação de recursos alternativos para situações críticas
- **Comunicação Contínua:** Reuniões regulares para identificar problemas precocemente
- **Testes Incrementais:** Validação contínua para detectar problemas de qualidade
- **Documentação Abrangente:** Registro detalhado de decisões e mudanças
- **Treinamento da Equipe:** Capacitação em tecnologias e metodologias utilizadas

Para isso, este documento será usado para registrar métricas, estimar o tempo associado a cada atividade particular e organizar a interdependência entre atividades e a ordem em que elas serão realizadas pela equipe.



Dada essa informação, o tempo necessário para marcos particulares serem alcançados pode ser usado como sinal sobre se a equipe está mais lenta, mais rápida, ou na velocidade certa de desenvolvimento.

Esses sinais podem ser usados para refinar o conteúdo deste documento e instruir a equipe sobre como agir para entregar o produto final no prazo.

### **3.2.2 Indicadores de Alerta**

- Atraso superior a 2 dias em qualquer tarefa crítica
- Taxa de defeitos superior a 5% nos testes de sprint
- Indisponibilidade de recursos por mais de 3 dias consecutivos
- Mudanças de escopo que impactem mais de 10% do cronograma
- Problemas de comunicação entre equipes por mais de 24 horas
- Falhas recorrentes em builds ou deploys

### **3.2.3 Plano de Contingência**

Em caso de materialização dos riscos principais:

#### **1. Atraso no Cronograma:**

- Reavaliação de prioridades das funcionalidades
- Redistribuição de recursos entre equipes
- Extensão de jornada de trabalho (limitada)
- Simplificação de funcionalidades não críticas

#### **2. Problemas de Qualidade:**

- Intensificação dos testes pela equipe SQA
- Revisão de código por pares
- Implementação de ferramentas de análise estática
- Criação de ambiente de testes dedicado

#### **3. Indisponibilidade de Recursos:**

- Realocação temporária de membros entre equipes
- Contratação de consultor externo (se necessário)
- Redistribuição de tarefas entre membros disponíveis
- Priorização de atividades críticas

### 3.2.4 Monitoramento e Controle

- **Reuniões Diárias:** Status de progresso e identificação rápida de problemas
- **Relatórios Semanais:** Análise de métricas e tendências
- **Revisões de Sprint:** Avaliação abrangente de qualidade e progresso
- **Escalação de Problemas:** Procedimentos claros para comunicação de riscos críticos

## 4 Estimativas, Cronogramas e Prazos

### 4.1 Estimativas

#### 4.1.1 Dados Históricos Utilizados

Não foram coletados dados históricos específicos para este tipo de projeto até o momento. As estimativas foram baseadas em:

- Análise de complexidade de cada módulo
- Experiência da equipe em projetos similares
- Benchmarks da indústria para desenvolvimento web
- Consideração de integrações externas necessárias
- Metodologias ágeis de estimativa (Planning Poker)

#### 4.1.2 Técnicas de Estimativa Empregadas

- **Decomposição por Funcionalidades:** Divisão em módulos menores e estimativa individual
- **Análise de Pontos de Função:** Avaliação da complexidade baseada em entradas, saídas e processos
- **Estimativa por Analogia:** Comparação com sistemas similares já desenvolvidos
- **Consenso de Especialistas:** Validação das estimativas pela equipe técnica
- **Três Pontos:** Estimativa otimista, pessimista e mais provável para cálculo da média ponderada

#### 4.1.3 Estimativas por Sprint

- **Sprint 0 (Preparação):** 27 dias úteis - Documentação, análise e design
- **Sprint 1 (Desenvolvimento Inicial):** 26 dias úteis - Cadastros e funcionalidades básicas
- **Sprint 2 (Desenvolvimento Avançado):** 23 dias úteis - Sistema financeiro e cálculos
- **Sprint 3 (Finalização):** 10 dias úteis - Testes finais, correções e entrega

**Total do Projeto:** 86 dias úteis

## 4.2 Cronograma

### 4.2.1 Estrutura de Trabalho

Todo o trabalho será realizado em várias sprints de uma semana, organizadas em 4 ciclos principais, onde cada um desses ciclos contém diferentes números de sprints, respectivamente.

A organização das atividades e seu cronograma podem ser encontrados na tabela a seguir:

### 4.2.2 Legenda

- **G** - Gerentes (Gestão e Planejamento)
- **A&P** - Analistas e Programadores
- **COD** - Programadores (Codificação)
- **SQA** - Garantia da Qualidade (Testes)

### 4.2.3 Cronograma Detalhado

Atividade	Sprint 0	Sprint 1	Sprint 2	Sprint 3
<b>SPRINT 0 - PREPARAÇÃO (27 dias)</b>				
1.1 - Planejamento 0 (3d) - G	X			
1.2 - Criar Doc. Requisitos (10d) - A&P	X			
1.2.1 - Revisar Doc. Requisitos (8d) - SQA	X			
1.3 - Doc. Casos de Uso (6d) - A&P	X			
1.3.1 - Revisar Casos de Uso (2d) - SQA	X			
1.4 - Criar Modelo Conceitual (6d) - A&P	X			
1.4.1 - Revisar Modelo Conceitual (3d) - SQA	X			
1.5 - Criar Diagrama Classes (4d) - A&P	X			
1.5.1 - Revisar Diagrama Classes (2d) - SQA	X			
1.6 - Diagramas Sequência (5d) - A&P	X			
1.6.1 - Revisar Diagramas Sequência (3d) - SQA	X			
<b>SPRINT 1 - DESENVOLVIMENTO INICIAL (26 dias)</b>				
2.1 - Planejamento 1 (1d) - G		X		
2.1.1 - Criar Diag. Colaboração (3d) - A&P		X		

2.1.2 - Revisar Diag. Colaboração (2d) - SQA		X		
2.1.3 - Integrar Diag. Colaboração (1d) - A&P		X		
2.2 - Codificar Cadastro (13d) - COD		X		
2.3 - Testar Cadastro (5d) - SQA		X		
2.4 - Review (1d) - G		X		
<b>SPRINT 2 - DESENVOLVIMENTO AVANÇADO (23 dias)</b>				
3.1 - Planejamento 2 (1d) - G			X	
3.2 - Codificar Despesas (14d) - COD			X	
3.3 - Testar Despesas (8d) - SQA			X	
<b>SPRINT 3 - FINALIZAÇÃO (10 dias)</b>				
4.1 - Planning 3 (1d) - G				X
4.2 - Corrigir Bugs (5d) - COD				X
4.3 - Testes Finais (3d) - SQA				X
4.4 - Preparar Entrega (1d) - G				X

### 4.3 Gráfico de Gantt

O gráfico de Gantt completo do projeto está apresentado na figura abaixo, mostrando todas as atividades, suas dependências e alocação de recursos ao longo do tempo:

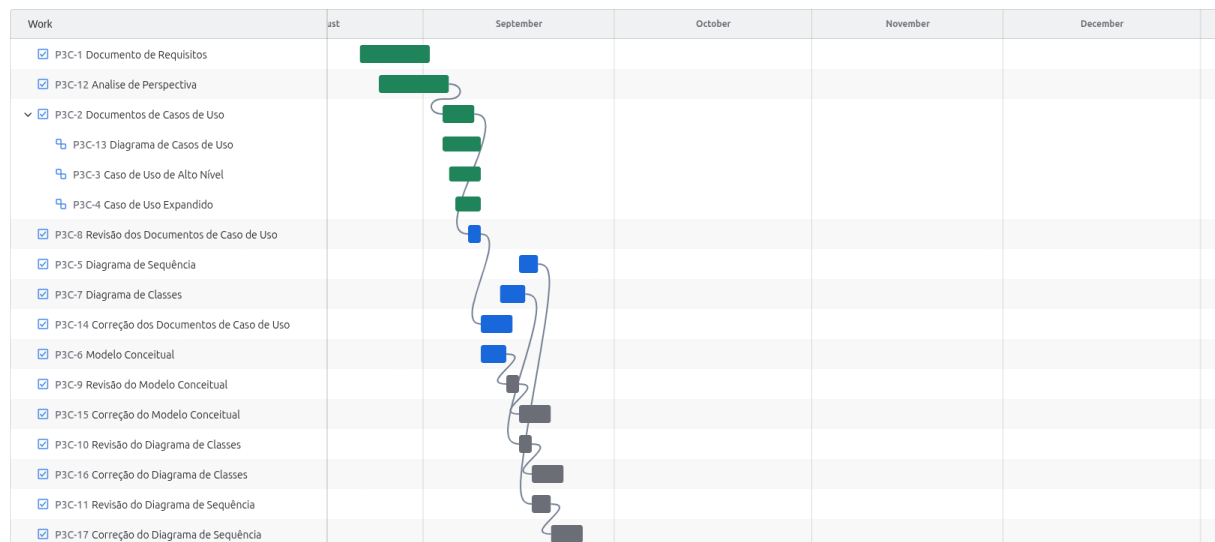


Figura 1: Gráfico de Gantt do Projeto - Sistema de Gestão de Churrascos Entre Amigos

O gráfico de Gantt ilustra claramente:

- A sequência temporal de todas as atividades
- As dependências críticas entre tarefas
- A alocação de recursos por equipe ao longo do tempo

- Os marcos principais de cada sprint
- O caminho crítico do projeto

#### 4.3.1 Caminho Crítico

**Sequência do Caminho Crítico:** 1.1 → 1.2 → 1.3 → 1.4 → 2.1 → 2.1.1 → 2.1.2 → 2.1.3 → 2.2 → 2.3 → 2.4 → 3.1 → 3.2 → 3.3 → 4.1 → 4.2 → 4.3 → 4.4

Como o cronograma é totalmente sequencial, **todas as tarefas fazem parte do caminho crítico**. Qualquer atraso em uma tarefa atrasa o projeto inteiro.

#### 4.4 Marcos Principais

- **M0 (Fim do Sprint 0):** Documentação completa e aprovada - Dia 27
- **M1 (Fim do Sprint 1):** Módulo de cadastros funcionando - Dia 53
- **M2 (Fim do Sprint 2):** Sistema financeiro implementado - Dia 76
- **M3 (Fim do Sprint 3):** Sistema completo e entregue - Dia 86

#### 4.5 Entregáveis por Sprint

- **Sprint 0:** Documento de Requisitos, Casos de Uso, Diagramas UML
- **Sprint 1:** Módulo de cadastros funcionando, Diagramas de Colaboração
- **Sprint 2:** Sistema financeiro completo, Módulo de despesas
- **Sprint 3:** Sistema integrado, testado e documentado

#### 4.6 Critérios de Aceitação

Cada marco deve atender aos seguintes critérios:

- Todos os entregáveis foram revisados e aprovados pela equipe SQA
- Cobertura de testes mínima de 80% para código
- Documentação atualizada e revisada
- Aprovação formal dos gerentes de projeto

### 5 Organização da Equipe

#### 5.1 Estrutura da Disciplina

O projeto está inserido no contexto da disciplina de Engenharia de Software 2, onde cada dupla de alunos assume diferentes papéis em múltiplos projetos, proporcionando experiência completa no ciclo de desenvolvimento de software.

## 5.2 Estrutura da Equipe para Este Projeto

Para o **Projeto 3 - Sistema de Gestão de Churrascos Entre Amigos**, a estrutura da equipe segue o modelo de papéis rotacionados conforme especificado:

- **Gerentes (G) - Julio, Ricardo, Igor:**
  - Planejamento estratégico e operacional
  - Controle de cronograma e recursos
  - Coordenação entre equipes das outras duplas
  - Tomada de decisões de alto nível
  - Comunicação com stakeholders e professor
  - Criação e manutenção deste Plano de Projeto
  - Monitoramento de riscos e qualidade
  - Aprovação de entregáveis e marcos
- **SQA - Garantia da Qualidade (Luiz e Sara):**
  - Elaboração de planos de teste
  - Execução de testes funcionais e não-funcionais
  - Identificação e documentação de defeitos
  - Revisão de artefatos de projeto
  - Validação da qualidade final do produto
  - Definição de critérios de aceitação
  - Auditoria de processos de desenvolvimento
  - Relatórios de qualidade e métricas
- **A&P - Analistas e Programadores (Augusto e Coleta):**
  - Análise e especificação de requisitos
  - Modelagem conceitual e de dados
  - Design de arquitetura do sistema
  - Criação de diagramas UML
  - Validação de especificações
  - Prototipação de interfaces
  - Documentação técnica de análise
  - Suporte ao design de banco de dados
- **COD - Programadores (Guilherme Carrara e Guilherme Digiorgi):**
  - Implementação das funcionalidades
  - Codificação seguindo padrões estabelecidos
  - Integração de módulos

- Correção de bugs e melhorias
- Documentação técnica do código
- Testes unitários básicos
- Configuração de ambiente de desenvolvimento
- Versionamento e controle de código

### 5.3 Matriz de Responsabilidades

Atividade	Gerentes	SQA	A&P	COD
Planejamento	R	C	C	C
Análise de Requisitos	A	R	R	C
Design do Sistema	A	R	R	C
Codificação	A	C	C	R
Testes	A	R	C	C
Documentação	R	R	R	R
Integração	A	R	C	R
Entrega	R	A	C	C

**Legenda:** R = Responsável, A = Aprovador, C = Consultado

### 5.4 Alocação por Sprint

#### 5.4.1 Sprint 0 - Preparação

- **Gerentes (Julio, Ricardo e Igor):** Planejamento inicial e definição de diretrizes
- **A&P (Augusto e Coleta):** Foco principal - documentação e modelagem
- **COD (Guilherme Carrara e Guilherme Digiorgi):** Preparação do ambiente de desenvolvimento
- **SQA (Luiz e Sara):** Planejamento de estratégias de teste e qualidade

#### 5.4.2 Sprint 1 - Desenvolvimento Inicial

- **Gerentes (Julio, Ricardo e Igor):** Acompanhamento e controle do cronograma
- **A&P (Augusto e Coleta):** Finalização de diagramas de colaboração
- **COD (Guilherme Carrara e Guilherme Digiorgi):** Foco principal - codificação de cadastros
- **SQA (Luiz e Sara):** Testes e revisão do módulo de cadastro

#### 5.4.3 Sprint 2 - Desenvolvimento Avançado

- **Gerentes (Julio, Ricardo e Igor):** Monitoramento de riscos e ajustes no plano
- **A&P (Augusto e Coleta):** Suporte à codificação complexa
- **COD (Guilherme Carrara e Guilherme Digiori):** Foco principal - sistema financeiro e cálculos
- **SQA (Luiz e Sara):** Testes extensivos do sistema financeiro

#### 5.4.4 Sprint 3 - Finalização

- **Gerentes (Julio, Ricardo e Igor):** Preparação da entrega e documentação final
- **A&P (Augusto e Coleta):** Revisão final e documentação
- **COD (Guilherme Carrara e Guilherme Digiori):** Correção de bugs e refinamentos
- **SQA (Luiz e Sara):** Testes finais e validação completa do sistema

### 5.5 Comunicação e Coordenação

#### 5.5.1 Reuniões Regulares

- **Daily Standups:** Reuniões diárias de 15 minutos entre as duplas
- **Sprint Planning:** Planejamento detalhado no início de cada sprint
- **Sprint Review:** Avaliação dos resultados ao final de cada sprint
- **Retrospectivas:** Identificação de melhorias no processo
- **Reuniões Interdisciplinares:** Coordenação com outros projetos da disciplina

#### 5.5.2 Ferramentas de Gestão

- Sistema de controle de versão (Git)
- Ferramenta de gestão de projetos (ex: Jira, Trello)
- Plataforma de comunicação (ex: Slack, Teams, WhatsApp)
- Repositório de documentação centralizado
- Ambiente de desenvolvimento compartilhado



## 5.6 Relatórios de Gestão

### 5.6.1 Estrutura de Reportes

O projeto seguirá uma estrutura hierárquica de reportes para garantir comunicação eficaz e controle adequado:

- **Relatórios Diários:** Status updates das tarefas em andamento
- **Relatórios Semanais:** Progresso detalhado por sprint e identificação de impedimentos
- **Relatórios de Marco:** Avaliação completa ao final de cada sprint
- **Relatórios de Exceção:** Comunicação imediata de problemas críticos

### 5.6.2 Métricas de Acompanhamento

#### 1. Progresso de Cronograma:

- Percentual de tarefas concluídas no prazo
- Variação entre estimado vs realizado
- Identificação de gargalos

#### 2. Qualidade:

- Número de defeitos por módulo
- Taxa de retrabalho
- Cobertura de testes

#### 3. Recursos:

- Utilização da equipe por perfil
- Disponibilidade vs demanda
- Identificação de sobrecarga

## 5.7 Responsabilidades Específicas dos Gerentes

Como líderes do Projeto 3, Julio, Ricardo e Igor têm as seguintes responsabilidades adicionais:

- Coordenar atividades com as outras duplas envolvidas no projeto
- Reportar progresso para o professor da disciplina
- Gerenciar dependências entre projetos da disciplina
- Assegurar qualidade e cumprimento de prazos
- Facilitar comunicação entre diferentes perfis
- Tomar decisões estratégicas quando necessário

- Manter documentação atualizada e acessível
- Resolver conflitos e impedimentos
- Aprovar mudanças de escopo ou cronograma

## 5.8 Critérios de Qualidade e Entrega

- **Código:** Seguir padrões de codificação definidos, comentários adequados, versionamento correto
- **Documentação:** Completa, atualizada e revisada por pelo menos dois membros da equipe
- **Testes:** Cobertura mínima de 80%, testes automatizados onde possível
- **Integração:** Builds automatizados sem falhas, integração contínua funcional