Análise de Riscos - Jogo WAR

# Mitigação dos Riscos: Contenção e Contingência

## Riscos Relacionados ao Escopo

### Risco 1: Mudanças no escopo durante o desenvolvimento

**Contenção**: Definir claramente o escopo desde o início. Utilizar a Estrutura Analítica do Projeto (EAP) para gerenciar e priorizar funcionalidades.

**Contingência**: Implementar um processo rigoroso de controle de mudanças para avaliar e aprovar novas solicitações de forma estruturada.

## Riscos Relacionados ao Cronograma

### Risco 2: Atrasos no desenvolvimento devido à complexidade do sistema

**Contenção**: Quebrar o desenvolvimento em sprints curtos e monitorar o progresso de perto.

**Contingência**: Ajustar o cronograma conforme a complexidade das funcionalidades e o progresso da equipe.

## Riscos Relacionados à Equipe

### Risco 3: Mão de obra insuficiente ou inexperiente

**Contenção**: Investir tempo na capacitação da equipe antes do início do projeto.

**Contingência**: Utilizar a metodologia Ágil para permitir que a equipe trabalhe de maneira colaborativa e se ajuste ao longo do projeto, aprendendo com os desafios.

### Risco 4: Rotatividade de membros da equipe

**Contenção**: Garantir uma boa integração entre os membros e realizar reuniões regulares para mantê-los atualizados.

**Contingência**: Manter uma documentação clara para facilitar a transição de novos membros caso a rotatividade ocorra.

## Riscos Técnicos

### Risco 5: Problemas de integração com a API

**Contenção**: Definir claramente os endpoints e realizar testes de integração contínuos. Garantir que a API seja bem documentada e tenha um ambiente de testes automatizados.

**Contingência**: Ter planos alternativos para lidar com falhas de API, como o uso de dados de mock para continuar o desenvolvimento do jogo.

### Risco 6: Desempenho do jogo (lag, travamentos)

**Contenção**: Realizar testes de desempenho e otimizar o código que possa causar lentidão.

**Contingência**: Se os problemas de desempenho persistirem, ajustar a interface gráfica e a lógica do backend para garantir uma experiência mais suave para o usuário.

## Riscos de Qualidade

### Risco 7: Testes insuficientes e falhas não detectadas

**Contenção**: Implementar testes automatizados (unitários, de integração e de usabilidade) e realizar revisões contínuas de código.

**Contingência**: Realizar sessões de teste alpha/beta para identificar falhas previamente, antes do lançamento.

### Risco 8: Desvio da qualidade do produto final

**Contenção**: Estabelecer um padrão de qualidade claro desde o início e realizar revisões constantes para manter esse padrão.

**Contingência**: Priorizar a qualidade sobre a velocidade de entrega, mesmo que isso resulte em um prazo mais longo.

## Riscos Relacionados ao Orçamento

### Risco 9: Excesso de custos

**Contenção**: Estabelecer um orçamento detalhado no início do projeto e fazer um controle de custos rigoroso.

**Contingência**: Utilizar ferramentas de monitoramento de custos para fazer ajustes rápidos no orçamento caso ele comece a ser ultrapassado.

# Probabilidade, Impacto e Exposição

| **Risco** | **Probabilidade** | **Impacto** | **Exposição** |
| --- | --- | --- | --- |
| **Subestimativa de tempo devido a complexidade técnica** | 0.6 | 0.8 | 0.48 |
| **Desvio da qualidade do produto final** | 0.6 | 0.6 | 0.36 |
| **Mão de obra insuficiente ou inexperiente** | 0.5 | 0.7 | 0.35 |
| **Mudanças no escopo durante o desenvolvimento** | 0.8 | 0.4 | 0.32 |
| **Problemas de integração com a API** | 0.3 | 0.8 | 0.24 |
| **Testes insuficientes e falhas não detectadas** | 0.5 | 0.3 | 0.15 |
| **Rotatividade de membros da equipe** | 0.2 | 0.7 | 0.14 |
| **Excesso de custos** | 0.1 | 0.9 | 0.09 |
| **Desempenho do jogo (lag, travamentos)** | 0.2 | 0.5 | 0.1 |