

# WAR I

Alisson Weiss

Carlos Eduardo Gaudie Ley Junior

Jorge Marinho Guimarães

Hiaggio Bezerra de Souza Machado

Leonardo Machado

Implementação do jogo de tabuleiro War I para o projeto de engenharia de software II

# Sumário

|   |           |
|---|-----------|
| <b>Sumário</b>                            | <b>2</b>  |
| <b>Escopo do Projeto</b>                  | <b>4</b>  |
| Descrição do Projeto                      | 4         |
| Descrição do Produto                      | 4         |
| <i>Stakeholders</i>                       | 4         |
| Patrocinador                              | 5         |
| Equipe e Gerência do Projeto              | 5         |
| Objetivos do Projeto                      | 6         |
| Para o Sponsor                            | 6         |
| Para a Equipe                             | 6         |
| Justificativa                             | 6         |
| Finalidade                                | 7         |
| Para o Sponsor                            | 7         |
| Para a Equipe                             | 7         |
| Fatores de Sucesso                        | 7         |
| Restrições                                | 7         |
| Premissas                                 | 7         |
| Exclusões                                 | 8         |
| Entregas                                  | 8         |
| Atividades                                | 8         |
| Descrição das Atividades                  | 9         |
| Estimativas                               | 12        |
| Estimativas de Custo e Orçamento Previsto | 12        |
| Estimativas de tempo                      | 14        |
| Marcos do Projeto e Entregas              | 15        |
| Critérios de Aceitação do Projeto         | 15        |
| <b>Cronograma</b>                         | <b>15</b> |
| Caminho Crítico e Folgas                  | 15        |
| Gráfico de Gantt                          | 17        |
| Data Mínima                               | 18        |
| Data Máxima                               | 20        |
| <b>Requisitos</b>                         | <b>21</b> |
| Atores                                    | 21        |
| Requisitos funcionais                     | 22        |
| Requisitos não funcionais                 | 22        |

|   |           |
|---|-----------|
| <b>Casos de Uso</b>                     | <b>27</b> |
| Escolher a cor do Exército              | 27        |
| Início do Jogo                          | 28        |
| Distribuir novas peças no mapa          | 29        |
| Realizar ataque                         | 30        |
| Remanejar exércitos pelo mapa           | 31        |
| Recebimento de cartas                   | 33        |
| Troca de cartas                         | 34        |
| <b>Monitoramento e Controle</b>         | <b>35</b> |
| Análise de Valor Agregado               | 35        |
| Iteração 1                              | 35        |
| Primeira Sprint                         | 35        |
| Segunda Sprint                          | 36        |
| Terceira Sprint                         | 37        |
| Iteração 2                              | 38        |
| Iteração 3                              | 38        |
| <b>Burndown Sprint</b>                  | <b>38</b> |
| Primeira sprint                         | 38        |
| Segunda sprint                          | 39        |
| Terceira sprint                         | 40        |
| <b>Plano de Monitoramento de Riscos</b> | <b>41</b> |
| Lista de Riscos                         | 41        |
| Externos                                | 41        |
| Internos                                | 41        |
| Técnicos                                | 41        |
| Estrutura Analítica de Riscos           | 42        |
| Análise de Riscos                       | 42        |
| Continuidade de Monitoramento           | 47        |
| <b>Relatório</b>                        | <b>47</b> |
| <b>Bibliografia</b>                     | <b>47</b> |

# Escopo do Projeto

## Descrição do Projeto

O jogo War I é um jogo de tabuleiro de guerra e estratégia. Este projeto consiste no desenvolvimento de uma versão digital do clássico de tabuleiro, voltada para a plataforma *web* e focada em partidas mais rápidas com um mapa inovador focado em uma versão da América do Sul. A elaboração desse jogo, assim como deste documento e demais outros com o propósito de acompanhamento, monitoramento e *log* do desenvolvimento contemplam a ementa da disciplina de Engenharia de Software II da Universidade Federal Fluminense.

## Descrição do Produto

O produto a ser desenvolvido é uma versão personalizada do jogo War, da Grow. Nesta versão, com foco em partidas mais rápidas, o mapa de atuação é o do Brasil e os jogadores terão a oportunidade de batalhar contra oponentes no mesmo dispositivo. O jogo será hospedado em um servidor *web*, e será acessado através do navegador.

A lógica do jogo rodará num motor Javascript de forma local, sem necessidade de chamadas a um servidor de *backend*. Os jogadores vão revezar sua vez de acordo com as rodadas utilizando o mesmo dispositivo, um computador com sistema operacional capaz de rodar os navegadores atuais, a saber, limitados inicialmente ao Firefox para fins de teste.

A interação se dará por mouse e teclado e toda comunicação com o jogador será visual, por escrita ou ilustrações.

## Stakeholders

Segundo CAMARGO (2018), "Partes interessadas são todas as pessoas ou organizações que possuem um envolvimento direto ou indireto no projeto e que podem afetá-lo positivamente ou negativamente. O PMBOK utiliza a tradução partes interessadas, porém em organizações que trabalham com projetos é comum manter o termo em inglês: *stakeholders*."

## Patrocinador

O professor desta disciplina, Troy Costa Kohwalter, para efeitos de construção do cenário da atuação será considerado o *sponsor* ou patrocinador do projeto, e assim será referenciado daqui em diante. Assim, certos aspectos deste documento tratarão o patrocinador como figura fictícia, como também serão fictícios seus anseios, objetivos e metas.

## Equipe e Gerência do Projeto

A equipe de desenvolvimento tem cinco membros que se revezam nos papéis de desenvolvedores, artistas gráficos e também fazem a manutenção e alimentação deste documento.

O membro Leonardo Machado é o gerente do projeto que, além de membro desenvolvedor, é a autoridade de representação da equipe, definição final dos custos e orçamentos e cumprimento das metas do cronograma.

Abaixo temos uma tabela síntese da equipe do projeto:

| Nome                             | Área                              |
|----------------------------------|-----------------------------------|
| Troy Kohwalter                   | Sponsor do projeto                |
| Leonardo Machado                 | Gerente do projeto, desenvolvedor |
| Alisson Weiss                    | Membro, desenvolvedor             |
| Carlos Eduardo Gaudie Ley Junior | Membro, desenvolvedor             |
| Hiaggo Bezerra de Souza Machado  | Membro, desenvolvedor             |
| Jorge Marinho Guimarães          | Membro, desenvolvedor             |

## Objetivos do Projeto

### Para o *Sponsor*

- Ter em seu portfólio um jogo *online* de fácil reconhecimento;
- Captação de novos jogadores com meta de pelo menos 100.000 partidas/dia;
- Expansão da marca;
- Firmar-se no mercado como *player* capaz de representar uma franquia tão conhecida com War;
- Avaliar a equipe de acordo com sua capacidade de produzir não somente um jogo de qualidade, mas de documentá-lo de forma correta, concisa, abrangente e em conformidade com as boas práticas acadêmicas.

### Para a Equipe

- Ser reconhecida como estúdio com capacidade de fornecer um entregável competitivo;
- Demonstrar capacidade de planejamento de requisitos, desenvolvimento, escopo;
- Ter uma documentação para referência futura disponível a todos os membros da equipe e que sirva de *template* para próximos projetos;

## Justificativa

- O *sponsor* tem interesse em adentrar o mercado de jogos, o qual pode atingir no Brasil a marca de US\$ 2.3 bilhões até 2023<sup>1</sup>.
- Jogos baseados em franquias de renome podem atrair mais jogadores;
- Com poucos jogos no mercado focados na temática brasileira e em seus jogadores, esta é uma boa oportunidade de lançar algo inovador e com carisma regional;
- A plataforma *web* apresenta vantagens frente aos *consoles* de mesa tradicionais, como menor burocracia para lançar o jogo, necessidade de menos recursos gráficos e liberdade de conteúdo.

---

1

[https://www.gamesindustry.biz/articles/2021-05-03-brazilian-games-market-to-hit-\\$2.3bn-in-2021-newzoo](https://www.gamesindustry.biz/articles/2021-05-03-brazilian-games-market-to-hit-$2.3bn-in-2021-newzoo)

## Finalidade

### Para o Sponsor

O patrocinador espera que com o *software* em produção sua marca se destaque no mercado de jogos casuais e atraia investidores para futuros projetos similares.

### Para a Equipe

A equipe tem como principal finalidade a oferta de um produto capaz de atender as expectativas de qualidade de *software*, controle de desenvolvimento, padrão de documentação de projeto, correta mensuração de recursos como tempo, pessoal e maquinário de apoio e desenvolvimento nos padrões acadêmicos estabelecidos.

## Fatores de Sucesso

São fatores preponderantes de sucesso a qualificação da equipe de desenvolvimento e o apoio do patrocinador ao longo do processo.

## Restrições

- O orçamento do projeto está restrito à R\$ 4371,90 ;
- O projeto deve ser realizado pela equipe designada. Não serão contratados recursos de outras equipes ou trabalhos terceirizados;
- A finalização do projeto, e por conseguinte implantação, deve ocorrer até o dia 17 de setembro de 2021.

## Premissas

São premissas do projeto:

- A apresentação regular do desenvolvimento ao patrocinador, antes da entrega final;
- A equipe alocada está disponível para o trabalho desde já;
- As entregas intermediárias e final serão realizadas pelo gerente do projeto;
- Haverá apoio do patrocinador.

## Exclusões

Estão excluídas do escopo deste projeto:

- Desenvolvimento para outras plataformas que não a *web*;
- Elaboração de mapas adicionais de outras regiões, como por exemplo, Europa ou América do Norte;
- Integração com outros *softwares* existentes, de posse do *sponsor*;
- Todo o equipamento (*hardware*) necessário para rodar o jogo em modo produção;

## Entregas

São de responsabilidade da equipe entregar ao patrocinador os seguintes itens:

- O jogo, produzido e rodando, além de seu código fonte e de testes;
- Documento de análise (este documento);
- Apresentações parciais e final, em formato de vídeo e resumo em *slides*, tipo *Powerpoint* ou similar.

Estas entregas serão realizadas de acordo com as estimativas abaixo, de forma progressiva. Assim, as versões vistas nas apresentações intermediárias sofrerão mudanças e correções ao longo do processo. O resultado final será apresentado na terceira entrega e validado pelo patrocinador.

## Atividades

Nosso projeto teve a duração de 3 semanas até o momento, no qual desenvolvemos parte do jogo e fizemos a gestão do projeto. Nós decompomos o projeto em etapas e suas respectivas atividades, de acordo com a seguinte estrutura EAP de modelo *top-down*:



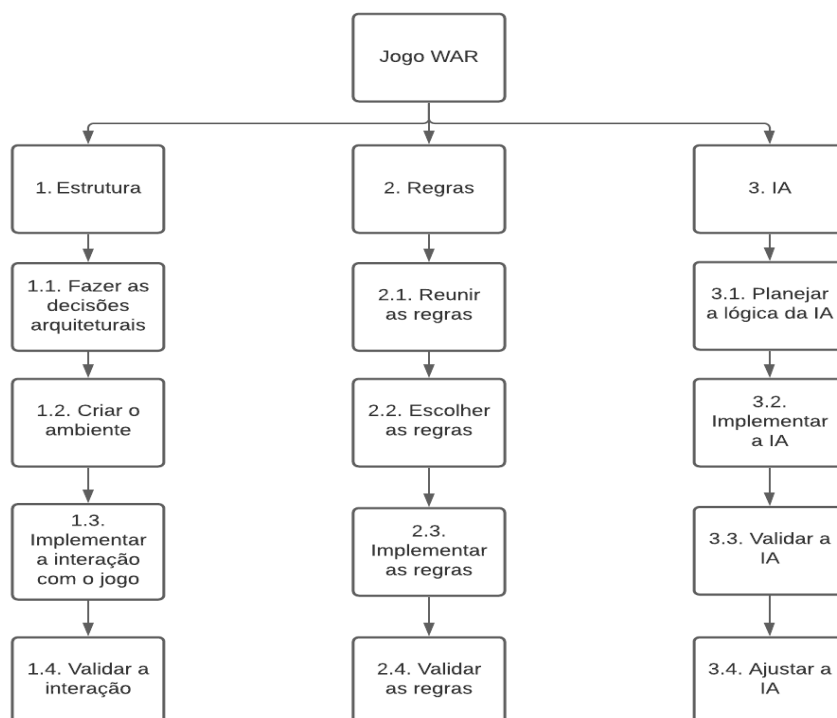


Figura 1. Estrutura EAP do projeto

## Descrição das Atividades

Detalhando as atividades expostas acima com seus respectivos esforços:

Obs.: O esforço é estimado baseado na quantidade de horas de trabalho.

| Pacote                               | Atividades  | Esforço |
|--------------------------------------|---|---------|
| 1.1. Fazer as decisões arquiteturais | <ul style="list-style-type: none"> <li>- Escolher linguagem de programação</li> <li>- Onde será feito o <i>deploy</i></li> <li>- Quais tecnologias auxiliares serão utilizadas</li> </ul> | 1       |
| 1.2. Criar o ambiente                | <ul style="list-style-type: none"> <li>- Montar o ambiente baseado nas tecnologias escolhidas no item acima</li> </ul>  | 2       |
| 1.3. Implementar a                   | <ul style="list-style-type: none"> <li>- Criar a interface gráfica</li> </ul>   | 12      |

|                              |   |    |
|------------------------------|---|----|
| interação com o jogo         | - Permitir que o usuário interaja com o jogo  |    |
| 1.4. Validar a interação     | - Testes das telas<br>- Testes dos cliques na tela  | 5  |
| 2.1. Reunir as regras        | - Entender as regras de negócio do War I  | 5  |
| 2.2. Escolher as regras      | - Selecionar as regras de negócio do War I que serão implementadas no nosso War I modificado.   | 2  |
| 2.3. Implementar as regras   | - Implementar as regras selecionadas no item acima no código.   | 20 |
| 2.4. Validar as regras       | - Verificar se as regras do jogo correspondem às regras de negócio estabelecidas neste documento.   | 3  |
| 3.1. Planejar a lógica da IA | - Definir a heurística<br>- Ajustar a heurística para ficar de acordo com as regras   | 5  |
| 3.2. Implementar a IA        | - Implementar o algoritmo escolhido   | 20 |
| 3.3. Validar a IA            | - Verificar o funcionamento da lógica da IA<br>- Verificar se o comportamento da IA está coerente<br>- Verificar se a dificuldade do jogo está adequada | 5  |
| 3.4. Ajustar a IA            | - A partir dos erros ocorridos nas validações, ajustar a IA   | 10 |

Com base na tabela acima, podemos perceber que há dependência entre as tarefas, de forma que uma precisa terminar para a outra começar. Abaixo temos a representação dessas dependências exibidas no formato de tabela:

| Pacote de atividades                    | Dependências |
|---|--------------|
| 1.1. Fazer as decisões arquiteturais    |              |
| 1.2. Criar o ambiente                   | 1.1          |
| 1.3. Implementar a interação com o jogo | 1.2          |
| 1.4. Validar a interação                | 1.3          |
| 2.1. Reunir as regras                   |              |
| 2.2. Escolher as regras                 | 2.1          |
| 2.3. Implementar as regras              | 1.3 e 2.2    |
| 2.4. Validar as regras                  | 2.3          |
| 3.1. Planejar a lógica da IA            | 2.2          |
| 3.2. Implementar a IA                   | 2.3 e 3.1    |
| 3.3. Validar a IA                       | 3.2          |
| 3.4. Ajustar a IA                       | 3.3          |

A partir disso, podemos construir o grafo de dependências das atividades:

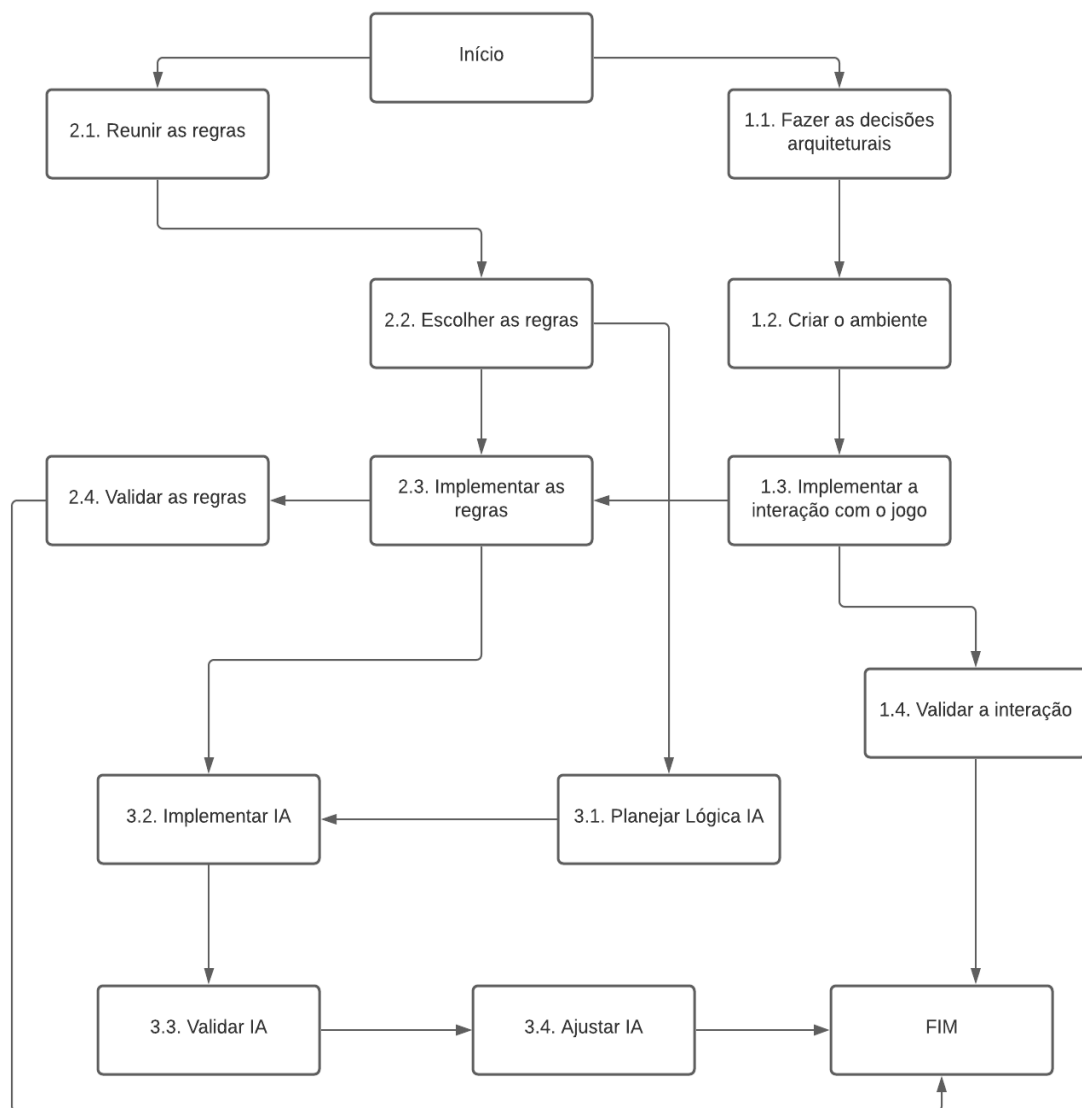


Figura 2. Gráfico de dependência

## Estimativas

### Estimativas de Custo e Orçamento Previsto

O orçamento foi apresentado ao patrocinador na data da primeira apresentação, e consta da quantia de R\$ 4371,90. Este valor é referente ao processo de análise e

desenvolvimento, e é passível de reajuste caso haja proposta de mudanças no escopo. Não estão incluídos neste valor a aquisição de artes, pagamento de franquias, processo de implantação ou outros custos atrelados restritamente ao produto, assim como sua posterior manutenção, seja *hardware* ou *software*.

Foi feita uma pesquisa de mercado para estimar o custo de um desenvolvedor sênior em ReactJS<sup>2</sup>. A remuneração mensal de um profissional com essa especialização é de, em média R\$ 8.549,50. Portanto, como estamos trabalhando com estimativa de hora / homem, o valor será de R\$ 8.549,50 / (8 horas \* 22 dias) = R\$ 48,58 / hora.

Para fins de demonstração, segue abaixo os cálculos de honorários:

| Pacote                                  | Custo      |
|---|------------|
| 1.1. Fazer as decisões arquiteturais    | R\$ 48,58  |
| 1.2. Criar o ambiente                   | R\$ 97,16  |
| 1.3. Implementar a interação com o jogo | R\$ 582,96 |
| 1.4. Validar a interação                | R\$ 242,90 |
| 2.1. Reunir as regras                   | R\$ 242,90 |
| 2.2. Escolher as regras                 | R\$ 97,16  |
| 2.3. Implementar as regras              | R\$ 971,60 |
| 2.4. Validar as regras                  | R\$ 145,74 |
| 3.1. Planejar a lógica da IA            | R\$ 242,90 |
| 3.2. Implementar a IA                   | R\$ 971,60 |
| 3.3. Validar a IA                       | R\$ 242,90 |

---

<sup>2</sup> Como visto em [https://www.glassdoor.com.br/Vaga/rio-de-janeiro-react-developer-vagas-SRCH\\_IL.0.14\\_IC2402386\\_KO15.30.htm](https://www.glassdoor.com.br/Vaga/rio-de-janeiro-react-developer-vagas-SRCH_IL.0.14_IC2402386_KO15.30.htm), em 9/7/2021

|                   |                    |
|-------------------|--------------------|
| 3.4. Ajustar a IA | R\$ 485,50         |
| <b>Total</b>      | <b>R\$ 4371,90</b> |

## Estimativas de tempo

As estimativas de tempo indicam quantas horas foram dedicadas a cada um dos pacotes de atividades do projeto.

| Pacote de atividades                    | Tempo (horas) |
|---|---------------|
| 1.1. Fazer as decisões arquiteturais    | 8             |
| 1.2. Criar o ambiente                   | 8             |
| 1.3. Implementar a interação com o jogo | 24            |
| 1.4. Validar a interação                | 16            |
| 2.1. Reunir as regras                   | 16            |
| 2.2. Escolher as regras                 | 8             |
| 2.3. Implementar as regras              | 40            |
| 2.4. Validar as regras                  | 8             |
| 3.1. Planejar a lógica da IA            | 16            |
| 3.2. Implementar a IA                   | 40            |
| 3.3. Validar a IA                       | 16            |
| 3.4. Ajustar a IA                       | 24            |

|       |     |
|-------|-----|
| Total | 224 |
|-------|-----|

## Marcos do Projeto e Entregas

Datas fixadas em acordo com o *sponsor*, que marcam o cronograma:

| Fase                      | Data       |
|---------------------------|------------|
| Primeira entrega          | 23/07/2021 |
| Segunda Entrega           | 13/08/2021 |
| Terceira entrega          | 17/09/2021 |
| Validação do patrocinador | 24/09/2021 |

## Crerérios de Aceitação do Projeto

- Escopo de detalhado, bem organizado e entregue nas datas especificadas acima;
- Implantação do produto, ou seja, capacidade do produto de operar em modo produção;
- O produto deve estar em conformidade com o descrito neste documento.
- Os prazos devem ser cumpridos;

## Cronograma

## Caminho Crítico e Folgas

Considerando a data final da entrega do projeto, definimos que nosso projeto terá 3 iterações. Dadas as atividades mencionadas anteriormente e o fluxo das atividades, fizemos

a análise do caminho crítico e as possíveis folgas para a realização do projeto dentro do prazo, com o caminho crítico totalizando em 70 horas:

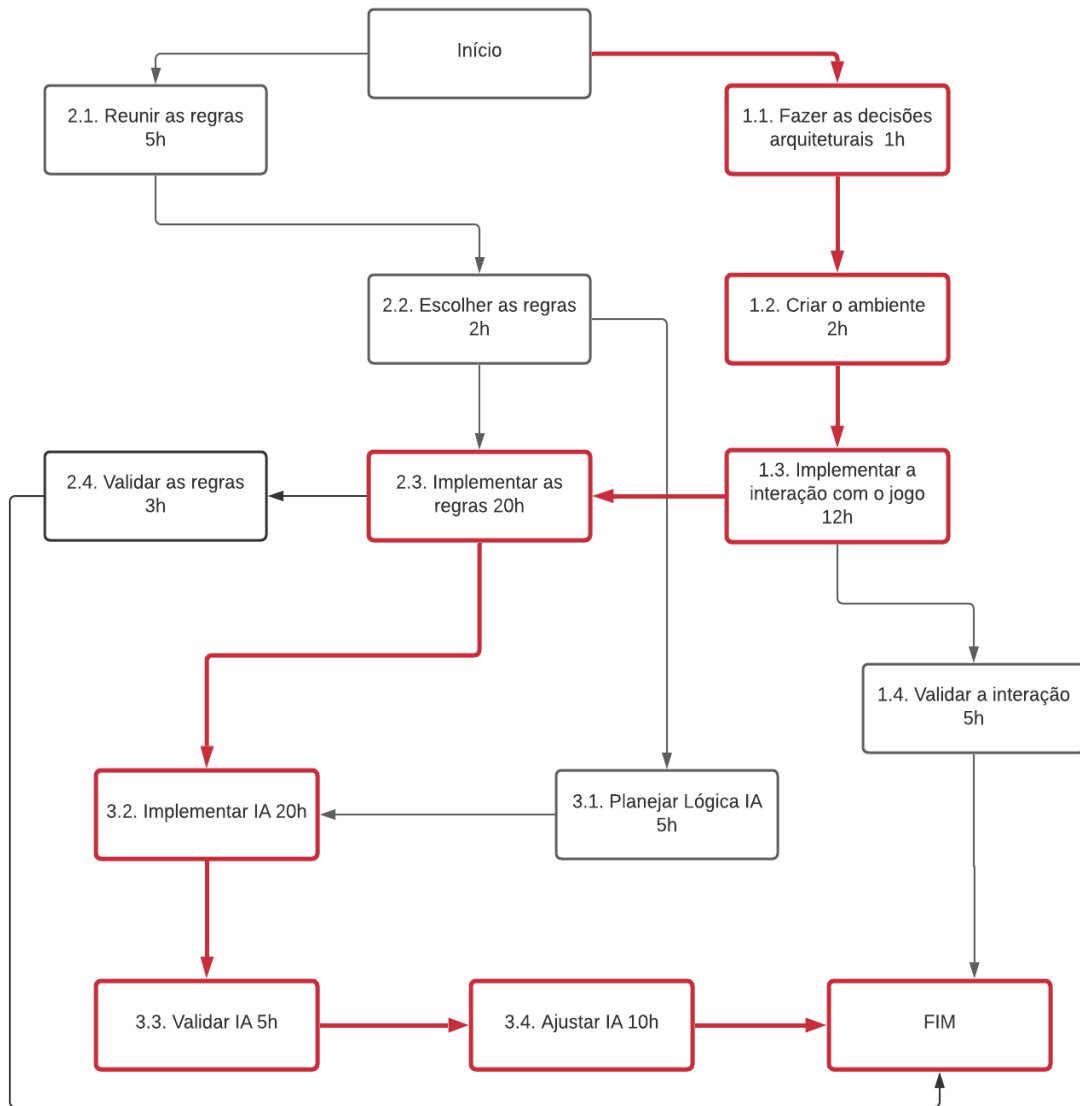


Figura 3. Caminho Crítico

Estabelecido o CPM, verificamos que a maior duração é de 20 dias. Com isso definimos as folgas da seguinte forma:



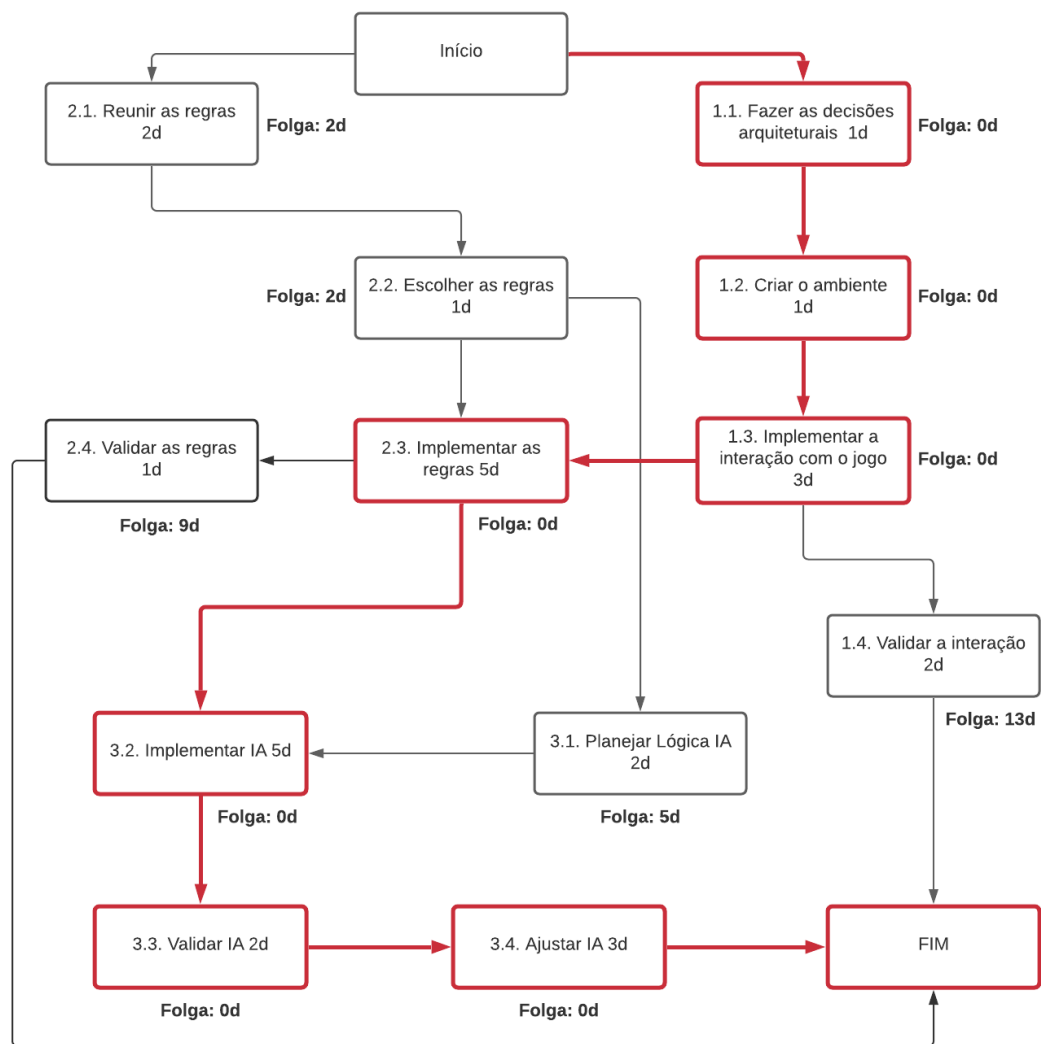


Figura 4. Caminho com folgas

## Gráfico de Gantt

Abaixo segue o gráfico de Gantt com data mínima e máxima e caminho crítico:

## Data Mínima

| Task name |   | Start date        | End date          | Duration    |
|-----------|---|-------------------|-------------------|-------------|
|           |   | <b>12/07/2021</b> | <b>06/08/2021</b> | <b>160h</b> |
| <b>1</b>  | <input type="checkbox"/> <b>Estrutura</b> | <b>12/07/2021</b> | <b>20/07/2021</b> | <b>56h</b>  |
| 1.1       | Fazer as decisões arquiteturais           | 12/07/2021        | 12/07/2021        | 8h          |
| 1.2       | Criar o ambiente                          | 13/07/2021        | 13/07/2021        | 8h          |
| 1.3       | Implementar a interação com o jogo        | 14/07/2021        | 16/07/2021        | 24h         |
| 1.4       | Validar a interação                       | 19/07/2021        | 20/07/2021        | 16h         |
| <b>2</b>  | <input type="checkbox"/> <b>Regras</b>    | <b>12/07/2021</b> | <b>26/07/2021</b> | <b>88h</b>  |
| 2.1       | Reunir as regras                          | 12/07/2021        | 13/07/2021        | 16h         |
| 2.2       | Escolher as regras                        | 14/07/2021        | 14/07/2021        | 8h          |
| 2.3       | Implementar as regras                     | 19/07/2021        | 23/07/2021        | 40h         |
| 2.4       | Validar as regras                         | 26/07/2021        | 26/07/2021        | 8h          |
| <b>3</b>  | <input type="checkbox"/> <b>IA</b>        | <b>12/07/2021</b> | <b>06/08/2021</b> | <b>160h</b> |
| 3.1       | Planejar a lógica da IA                   | 12/07/2021        | 13/07/2021        | 16h         |
| 3.2       | Implementar IA                            | 26/07/2021        | 30/07/2021        | 40h         |
| 3.3       | Validar a IA                              | 02/08/2021        | 03/08/2021        | 16h         |
| 3.4       | Ajustar a IA                              | 04/08/2021        | 06/08/2021        | 24h         |

Figura 5. Data mínima para conclusão do projeto

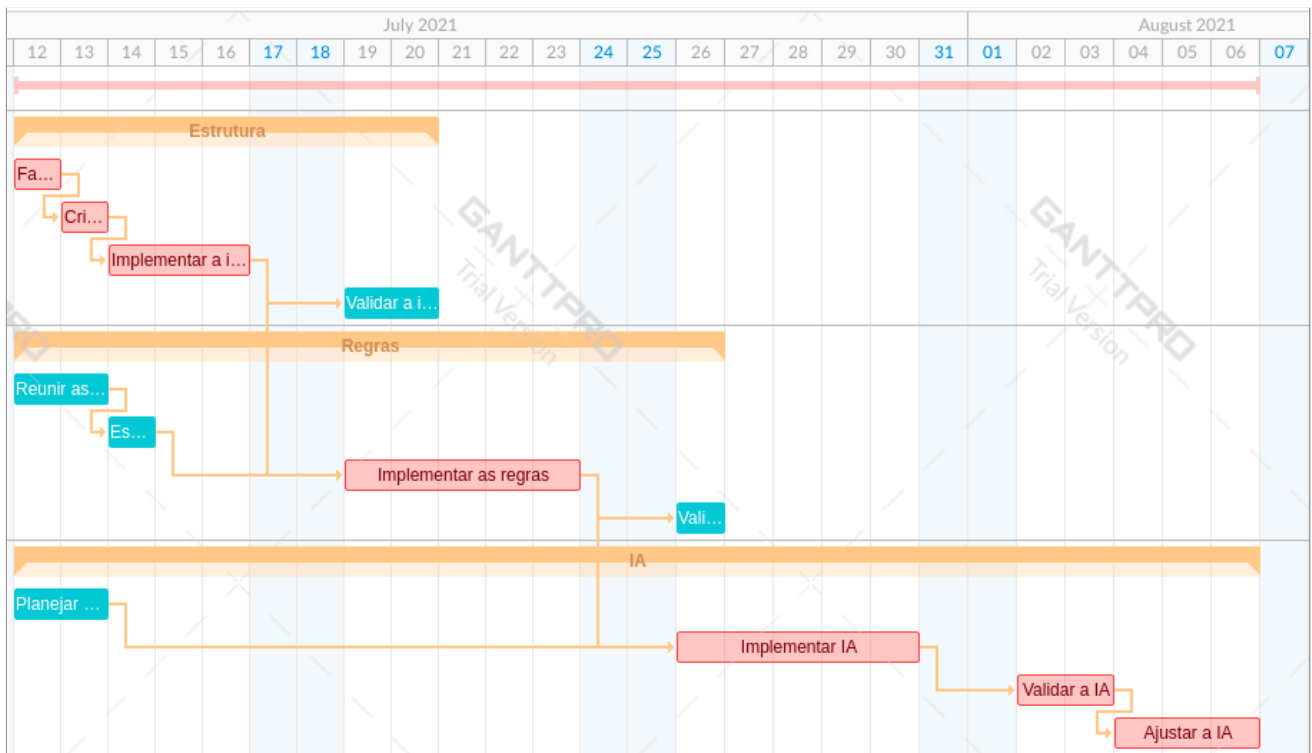


Figura 6. Data mínima para conclusão do projeto

## Data Máxima

| Task name |   | Start date        | End date          | Duration    |
|-----------|---|-------------------|-------------------|-------------|
|           |   | <b>23/08/2021</b> | <b>17/09/2021</b> | <b>160h</b> |
| <b>1</b>  | <input type="checkbox"/> <b>Estrutura</b> | <b>23/08/2021</b> | <b>31/08/2021</b> | <b>56h</b>  |
| 1.1       | Fazer as decisões arquiteturais           | 23/08/2021        | 23/08/2021        | 8h          |
| 1.2       | Criar o ambiente                          | 24/08/2021        | 24/08/2021        | 8h          |
| 1.3       | Implementar a interação com o jogo        | 25/08/2021        | 27/08/2021        | 24h         |
| 1.4       | Validar a interação                       | 30/08/2021        | 31/08/2021        | 16h         |
| <b>2</b>  | <input type="checkbox"/> <b>Regras</b>    | <b>23/08/2021</b> | <b>06/09/2021</b> | <b>88h</b>  |
| 2.1       | Reunir as regras                          | 23/08/2021        | 24/08/2021        | 16h         |
| 2.2       | Escolher as regras                        | 25/08/2021        | 25/08/2021        | 8h          |
| 2.3       | Implementar as regras                     | 30/08/2021        | 03/09/2021        | 40h         |
| 2.4       | Validar as regras                         | 06/09/2021        | 06/09/2021        | 8h          |
| <b>3</b>  | <input type="checkbox"/> <b>IA</b>        | <b>23/08/2021</b> | <b>17/09/2021</b> | <b>160h</b> |
| 3.1       | Planejar a lógica da IA                   | 23/08/2021        | 24/08/2021        | 16h         |
| 3.2       | Implementar IA                            | 06/09/2021        | 10/09/2021        | 40h         |
| 3.3       | Validar a IA                              | 13/09/2021        | 14/09/2021        | 16h         |
| 3.4       | Ajustar a IA                              | 15/09/2021        | 17/09/2021        | 24h         |

Figura 7. Data máxima para conclusão do projeto

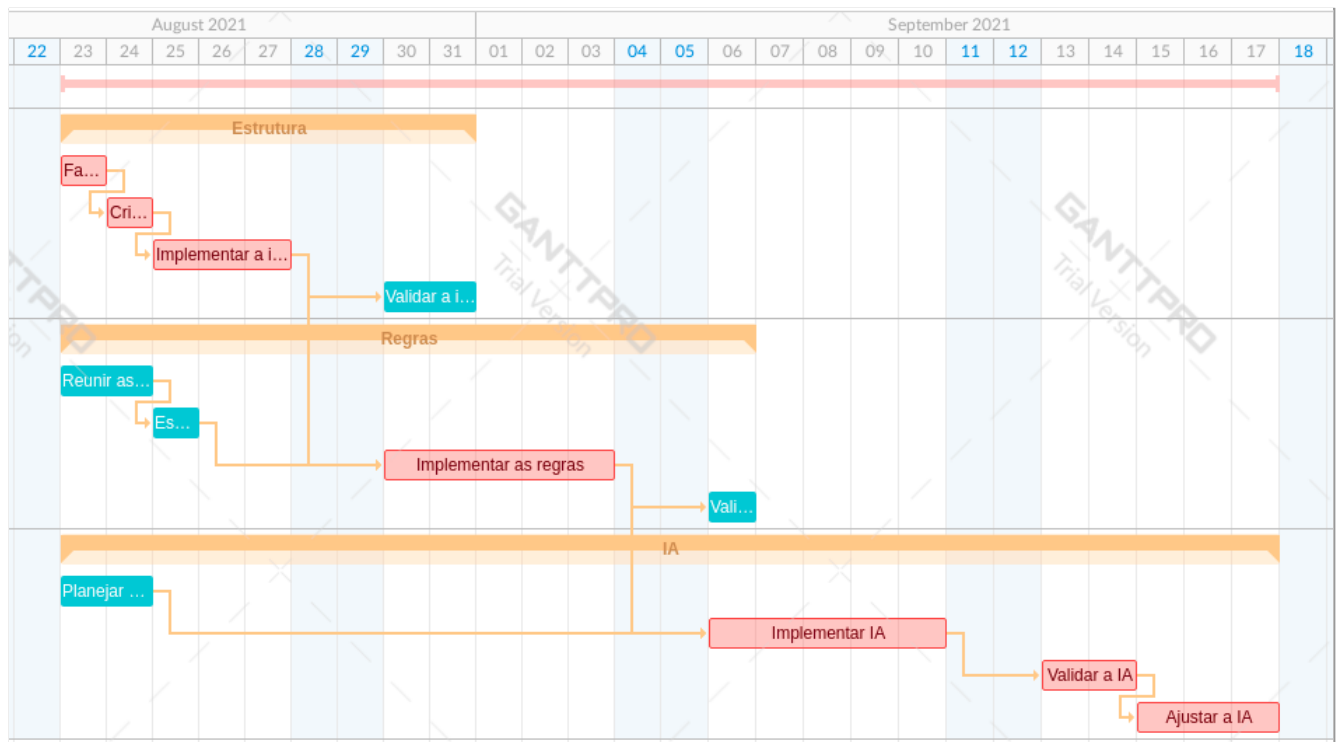


Figura 8. Data máxima para conclusão do projeto

## Requisitos

### Atores

Os atores do sistema são identificados como os papéis que um usuário pode assumir. No levantamento de requisitos junto com os *stakeholders* foram identificados os seguintes perfis:

- Jogador
- Sistema

### Requisitos funcionais

[RF001]

Escolher para cada cor de exército o tipo de participante.

**[RF002]**

Escolher o número de exércitos que serão enviados para reforçar um território.

**[RF003]**

Escolher o número de exércitos para enviar para o ataque de um território.

**[RF004]**

Escolher cartas para trocar por exércitos, a partir de 3 cartas.

**[RF005]**

Mostrar a cor no mapa do exército com o domínio de cada território.

**[RF006]**

Mostrar na tela para o jogador qual é o seu objetivo na partida.

**[RF007]**

Mostrar a quantidade de exércitos que estão em cada território do mapa.

## Requisitos não funcionais

**[RNF001]**

Os navegadores suportados são: Firefox *Browser* 89 e Chrome 91.

**[RNF002]**

O sistema operacional deve ser Windows 10 ou Ubuntu 20.04

**[RNF003]**

O sistema deve conter testes unitários.

**[RNF004]**

Usuário deve aceitar *cookies*.

**[RNF005]**

O *mouse* é necessário para jogar.

**[RNF006]**

O jogo deve ser jogado com resolução mínima de *HD* (720p) e máxima *full HD* (1080p).

**[RNF007]**

Linguagem utilizada no desenvolvimento do jogo: ReactJS v17.0.2.

**[RNF008]**

O resultado dos dados lançados em cada ataque/defesa é mostrado para os jogadores.

## Regras de negócio

Regra de negócio refere-se a premissas ou restrições de negócio que o sistema deverá atender, regras que poderão ou não estar associadas a um requisito funcional, mas que sempre serão realizadas por uma ou mais funcionalidades do sistema.

**[RN001]**

Para iniciar o jogo será necessário possuir ao menos três jogadores, dentre esses três jogadores ao menos um deve ser humano.

**[RN002]**

O sistema embaralha as cartas de território e as distribui entre os jogadores presentes na partida.

**[RN003]**

O jogador terá N peças a serem distribuídas pelo mapa e escolherá o território de destino em que deseja posicionar essas. O jogador pode posicionar quantas peças quiser em cada território, contanto que as peças distribuídas sejam iguais a N e o território de destino seja de sua propriedade.

**[RN004]**

Para realizar o ataque o jogador deverá selecionar um território de origem que possua ligação com o território que deseja atacar e este território de origem deve possuir no mínimo dois exércitos.

**[RN005]**

O jogador seleciona quantos exércitos deseja passar para o território que acabou de derrotar. O território de origem será, obrigatoriamente, o território que atacou o destino e venceu. O máximo de exércitos que podem ser passados será de três unidades e o local em que os exércitos serão removidos deverá possuir, no mínimo, um exército após a passagem de tropas.

**[RN006]**

Quando o jogador concluir seu objetivo, a partida deve seguir para o fluxo de encerramento.

**[RN007]**

O jogador acabou de atacar um território inimigo e venceu. O jogador deverá passar o seu exército para o local que acabou de vencer, o mínimo de peças que podem ser passadas é de uma unidade e o máximo de três.

**[RN008]**

Ao finalizar o ato de remanejamento, caso o jogador tenha vencido ao menos um território durante a rodada, este jogador será recompensado com uma carta de território presente na pilha de cartas.

**[RN009]**

Ao início da rodada do jogador, caso o jogador possua cinco cartas de território, o mesmo será obrigado a efetuar a troca. Caso o jogador possua três ou mais cartas o mesmo poderá, de forma totalmente opcional, efetuar a troca.

**[RN010]**



Para efetuar uma troca, o jogador deverá selecionar três cartas com os símbolos iguais ou então três cartas com símbolos diferentes.

#### **[RN011]**

As trocas seguirão conforme a tabela de trocas.

Exemplo: 1ª troca o jogador ganhará cinco exércitos adicionais.

2ª troca o jogador ganhará 10 tropas adicionais e assim por diante.

#### **[RN012]**

As trocas são definidas pelo número de trocas da partida e não de forma individual de cada jogador. Exemplo: Se o jogador amarelo efetuar a primeira troca da partida, ele ganhará cinco exércitos adicionais. Se o jogador preto, posteriormente, efetuar a troca, será a segunda troca da partida e, portanto, o mesmo ganhará 10 exércitos adicionais.

#### **[RN013]**

O sistema embaralha as cartas de objetivo e distribui uma para cada jogador.

#### **[RN014]**

Caso o jogador pegue o objetivo de derrotar o próprio exército o seu objetivo passa a ser, automaticamente, obter 14 territórios.

#### **[RN015]**

Ao início de cada rodada o jogador receberá mais peças para serem posicionadas. A quantidade de peças será definida pela parte inteira da divisão dos seus territórios pela metade.

#### **[RN016]**

O jogo possui um total de 10 cartas objetivos, estes objetivos são distribuídos para cada jogador e, o primeiro que cumprir o seu objetivo ganhará a partida.

#### **[RN017]**

Dentre as cartas de objetivos, temos o seguinte:

- Derrotar o exército amarelo
- Derrotar o exército vermelho
- Derrotar o exército azul
- Derrotar o exército preto
- Derrotar o exército branco
- Derrotar o exército verde
- Conquistar a região sul, a região sudeste e mais 2 territórios a sua escolha
- Conquistar a região norte e mais 2 territórios a sua escolha
- Conquistar a região nordeste
- Conquistar 14 territórios

#### [RN018]

Quando o jogador elimina um exército do jogo, o jogador recebe aleatoriamente cartas que estavam em posse do oponente, até o jogador possuir 5 cartas em sua mão, devolvendo ao deck as demais (caso o oponente possuísse mais cartas).

## Casos de Uso

### Escolher a cor do Exército

| Caso de Uso   | Escolher a cor do exército   |
|---------------|--|
| Tipo          | Primário, expandido, essencial   |
| Ator          | Jogador  |
| Resumo        | A cada jogador é atribuído um exército de acordo com a ordem de entrada, subsequentemente sua cor, conforme <b>RN001</b> . |
| Pré-condições | Nenhuma  |

|                        |  |
|------------------------|--|
| Pós-condições          | Cada jogador tem um exército e uma posição na fila de jogadores                                      |
| <b>Fluxo Principal</b> |  |
| Ações do Atores        | (1) Seleciona num <i>combobox</i> o status de cada jogador (player humano, player IA, não participa) |

## Início do Jogo

|                    |   |
|--------------------|---|
| <b>Caso de Uso</b> | Início do Jogo  |
| Tipo               | Primário, expandido, essencial  |
| Atores             | Sistema   |
| Resumo             | <p>O sistema distribui para cada jogador dois tipos de cartas: a carta de objetivo e as cartas de território, conforme a <b>RN002</b> e <b>RN013</b>.</p> <p>Após isso, o sistema posiciona um exército em cada um dos territórios recebidos, para cada jogador. Por fim, o sistema recolhe as cartas e as embaralha.</p> |
| Pré-condições      | Os exércitos foram atribuídos.  |
| Pós-condições      | Os exércitos estão distribuídos pelo mapa. Cada território possuirá um  |

|                  |   |
|------------------|---|
|                  | exército. Há uma pilha de cartas embaralhadas no local determinado.                                       |
| Fluxo Principal  |   |
| Ações do Sistema | (1) Distribuir cartas<br>(2) Posiciona exércitos no mapa<br>(3) Recebe cartas, embaralha e repõe ao jogo. |

## Distribuir novas peças no mapa

|                    |   |
|--------------------|---|
| <b>Caso de Uso</b> | Distribuir novas peças no mapa  |
| Tipo               | Primário, expandido, essencial  |
| Ator               | Jogadores   |
| Resumo             | Cada jogador, na sua vez, distribui peças pelo mapa, de acordo com a <b>RN003</b> .                             |
| Pré-condições      | Início do jogo finalizado   |
| Pós-condições      | Peças distribuídas  |
| Fluxo Principal    |   |
| Ações do Atores    | (1) Após abrir um modal, o jogador digita um valor em cada caixa correspondente a um território que ele possui. |

|                     |  |
|---------------------|--|
|                     |  |
| Fluxo Alternativo 1 | (1) Se for a primeira rodada, ao clicar em OK, libera a vez para o próximo jogador |
| Fluxo Alternativo 2 | (1) Se não for a primeira rodada, o jogador pode atacar.                           |

## Realizar ataque

|                    |  |
|--------------------|--|
| <b>Caso de Uso</b> | Realizar ataque  |
| Tipo               | Primário, expandido, essencial   |
| Ator               | Jogador, Sistema   |
| Resumo             | O jogador seleciona seu território de origem, onde vai atacar e com quantos exércitos. |
| Pré-condições      | Há ligação entre as regiões de origem e destino, conforme a regra <b>RN004</b> .       |
| Pós-condições      | Foram alteradas as quantidades de peças ou do atacante ou do defensor.                 |
| Fluxo Principal    |  |
| Ações do Jogador   | (1) Em um modal o jogador escolhe um de seus territórios como origem do ataque         |

|                     |   |
|---------------------|---|
|                     | <p>(2) Escolhe um território contíguo de um oponente</p> <p>(3) Seleciona com quantos exércitos quer atacar</p> <p>(4) Rola o dado</p> <p>(5) Após ação do sistema (1), reiniciar o ataque</p>            |
| Ações do Sistema    | (1) Remover as peças que foram combatidas   |
| Fluxo Alternativo 1 | <p>(5.a) Se o jogador eliminou todo o exército do local de destino, este se apossa daquela região</p> <p>(5.b) O jogador define quantas peças colocará na região de destino, conforme a <b>RN005</b>.</p> |
| Fluxo Alternativo 2 | (5.a) O jogador concluiu um objetivo, e consequentemente o jogo, conforme a <b>RN006</b> .  |
| Fluxo Alternativo 3 | (5.a) O jogador decide não mais atacar, passando ao caso de remanejar seu exército.   |

## Remanejar exércitos pelo mapa

|             |                               |
|-------------|-------------------------------|
| Caso de Uso | Remanejar exércitos pelo mapa |
|-------------|-------------------------------|

|                     |   |
|---------------------|---|
| Tipo                | Primário, expandido, essencial  |
| Ator                | Jogador   |
| Resumo              | Reposiciona as peças pelo mapa, conforme a <b>RN005</b> e a <b>RN007</b> .  |
| Pré-condições       | O jogador tem peças que podem ser movidas   |
| Pós-condições       | As peças estão reposicionadas   |
| Fluxo Principal     |   |
| Ações do Atores     | (1) Em um modal o jogador seleciona um local de origem<br>(2) Seleciona a quantidade a mover<br>(3) Seleciona o destino<br>(4) Se reinicia a ação |
| Fluxo Alternativo 1 | (1.a) O jogador decide não remanejar mais peças<br>(1.b) A rodada termina e o jogador pode receber uma carta, conforme a <b>RN008</b> .           |

## Recebimento de cartas

|                    |                                |
|--------------------|--------------------------------|
| <b>Caso de Uso</b> | Recebimento de cartas          |
| Tipo               | Primário, expandido, essencial |

|                     |   |
|---------------------|---|
| Ator                | Sistema   |
| Resumo              | O Sistema retira uma carta da pilha e entrega ao jogador da rodada  |
| Pré-condições       | O jogador venceu ao menos um território nesta rodada  |
| Pós-condições       | O jogador tem uma carta a mais.   |
| Fluxo Principal     |   |
| Ações do Atores     | (1) O sistema retira uma carta da pilha<br>(2) Entrega esta carta para o jogador<br>(3) O jogador termina a ação com uma carta a mais<br>(4) Passa a vez para o próximo jogador |
| Fluxo Alternativo 1 |   |

## Troca de cartas

|                    |   |
|--------------------|---|
| <b>Caso de Uso</b> | Troca de cartas   |
| Tipo               | Primário, expandido, essencial  |
| Ator               | Jogador   |
| Resumo             | Ao colecionar ao menos três cartas, ou, ao máximo cinco, o jogador pode executar uma troca de cartas, |



|                     |   |
|---------------------|---|
|                     | conforme <b>RN009</b> .   |
| Pré-condições       | Ter o mínimo de cartas ou ter o máximo de cartas. Estar no início da rodada.  |
| Pós-condições       | O jogador terá três cartas a menos e mais exércitos   |
| Fluxo Principal     |   |
| Ações do Atores     | (1) Em um modal , o jogador seleciona as cartas a serem trocadas;<br>(2) Clica em "trocar cartas";<br>(3) Recebe as peças |
| Fluxo Alternativo 1 |   |

## Monitoramento e Controle

Com base nos custos do projeto expostos anteriormente, fizemos a análise de valor agregado, a fim de monitorar os gastos e o desenvolvimento do projeto frente ao cronograma pré-determinado:

### Análise de Valor Agregado

O projeto tem custo de R\$ 4371,90 e duração planejada de 11 semanas.

**BAC = R\$ 4371,90**

## Iteração 1

- Primeira Sprint

O valor gasto até o momento foi de **R\$ 388,64**, pois as atividades já realizadas até o momento foram “fazer as decisões arquiteturais do projeto”, “reunir as regras” e “escolher as regras”. O percentual planejado de conclusão para a sprint 1 da primeira iteração é de 6%.

$$\text{AC} = \text{R\$ } 388,64$$

$$\text{P\%C} = 6\%$$

Portanto, o valor planejado é de R\$ 262,31.

$$\text{PV} = \text{R\$ } 4371,90 \times 6\% = \text{R\$ } 262,31$$

O percentual real de conclusão para a sprint 1 é de 9%.

$$\text{A\%C} = 9\%$$

Portanto, o valor agregado é de R\$ 393,47.

$$\text{EV} = \text{R\$ } 4371,90 \times 9\% = \text{R\$ } 393,47$$

Segundo o Indicador de desempenho de cronograma o projeto está adiantado no que diz respeito à primeira iteração, no esperado da sprint 1, obtendo um valor de **1,50**.

$$\text{SPI} = \text{R\$ } 393,47 / \text{R\$ } 262,31 = 1,50 \rightarrow \text{adiantado}$$

$$\text{SV} = \text{R\$ } 393,47 - \text{R\$ } 262,31 = \text{R\$ } 131,16$$

Já para as métricas de desempenho de custo:

$$\text{CPI} = \text{R\$ } 393,47 / \text{R\$ } 388,64 = 1,01 \rightarrow \text{abaixo do custo}$$

$$\text{CV} = \text{R\$ } 393,47 - \text{R\$ } 388,64 = \text{R\$ } 4,83$$

- Segunda Sprint

O valor gasto até o momento foi de **R\$ 485,80**, pois a atividade de “criar o ambiente” foi realizada, a atividade que mais demandou tempo, até o

momento. O percentual planejado de conclusão para a sprint 2 da primeira iteração é de 18%.

$$\text{AC} = \text{R\$ } 485,80$$

$$\text{P\%C} = 18\%$$

Portanto, o valor planejado é de R\$ 786,94.

$$\text{PV} = \text{R\$ } 4371,90 \times 18\% = \text{R\$ } 786,94$$

O percentual real de conclusão para a sprint 2 é de 16%.

$$\text{A\%C} = 16\%$$

Portanto, o valor agregado é de R\$ 699,50.

$$\text{EV} = \text{R\$ } 4371,90 \times 16\% = \text{R\$ } 699,50$$

Segundo o Indicador de desempenho de cronograma o projeto está atrasado no que diz respeito à primeira iteração, no esperado da sprint 2, obtendo um valor de **0,89**.

$$\text{SPI} = \text{R\$ } 699,50 / \text{R\$ } 786,94 = 0,89 \rightarrow \text{atrasado}$$

$$\text{SV} = \text{R\$ } 699,50 - \text{R\$ } 786,94 = - \text{R\$ } 87,44$$

Já para as métricas de desempenho de custo:

$$\text{CPI} = \text{R\$ } 699,50 / \text{R\$ } 485,80 = 1,44 \rightarrow \text{abaixo do custo}$$

$$\text{CV} = \text{R\$ } 699,50 - \text{R\$ } 485,80 = \text{R\$ } 213,70$$

- Terceira Sprint

O valor gasto até o momento foi de **R\$ 485,80**, pois não foi possível realizar nenhuma nova tarefa. O percentual planejado de conclusão para a sprint 3 da primeira iteração é de 23%.

$$\text{AC} = \text{R\$ } 485,80$$

$$\text{P\%C} = 23\%$$

Portanto, o valor planejado é de R\$ 1005,54.

$$\text{PV} = \text{R\$ } 4371,90 \times 23\% = \text{R\$ } 1005,54$$

O percentual real de conclusão para a sprint 3 é de 25%, pois conseguimos terminar uma das tarefas atrasadas.

$$\mathbf{A\%C = 25\%}$$

Portanto, o valor agregado é de R\$ 1.092,98.

$$\mathbf{EV = R\$ 4371,90 \times 25\% = R\$ 1092,98}$$

Segundo o Indicador de desempenho de cronograma, o projeto está adiantado no que diz respeito à primeira iteração, no esperado da sprint 3, obtendo um valor de **1,09**.

$$\mathbf{SPI = R\$ 1092,98 / R\$ 1005,54 = 1,09 \rightarrow \text{adiantado}}$$

$$\mathbf{SV = R\$ 1092,98 - R\$ 1005,54 = R\$ 87,44}$$

Já para as métricas de desempenho de custo:

$$\mathbf{CPI = R\$ 1092,98 / R\$ 485,80 = 2,25 \rightarrow \text{abaixo do custo}}$$

$$\mathbf{CV = R\$ 1092,98 - R\$ 485,80 = R\$ 607,18}$$

## Iteração 2

- Sprint Única

O valor gasto até o momento foi de **R\$ 485,80**, pois não foi realizada nenhuma nova tarefa. O percentual planejado de conclusão para a sprint da segunda iteração é de 45%.

$$\mathbf{AC = R\$ 485,80}$$

$$\mathbf{P\%C = 45\%}$$

Portanto, o valor planejado é de R\$ 1967,36.

$$\mathbf{PV = R\$ 4371,90 \times 45\% = R\$ 1967,36}$$

O percentual real de conclusão para a sprint é de 31%, pois tivemos diversos problemas de atraso e não conseguimos efetivamente nos programar para a conclusão das tarefas.

$$A\%C = 31\%$$

Portanto, o valor agregado é de R\$ 1355,29.

$$EV = R\$ 4371,90 \times 31\% = R\$ 1355,29$$

Segundo o Indicador de desempenho de cronograma, o projeto está atrasado no que diz respeito à segunda iteração, obtendo um valor de **0,69**.

$$SPI = R\$ 1355,29 / R\$ 1967,36 = 0,69 \rightarrow \text{atrasado}$$

$$SV = R\$ 1355,29 - R\$ 1967,36 = - R\$ 612,07$$

Já para as métricas de desempenho de custo:

$$CPI = R\$ 1355,29 / R\$ 485,80 = 2,79 \rightarrow \text{abaixo do custo}$$

$$CV = R\$ 1355,29 - R\$ 485,80 = R\$ 869,49$$

## Burndown Sprint

O gráfico de *burndown* serve para dar uma representação gráfica ao que era esperado da *sprint* e o que de fato foi realizado. O eixo horizontal representa a quantidade de dias da *sprint* e o eixo vertical representa a quantidade de tarefas executadas na *sprint*. Abaixo temos os gráficos:

## Iteração 1

- Primeira *sprint*

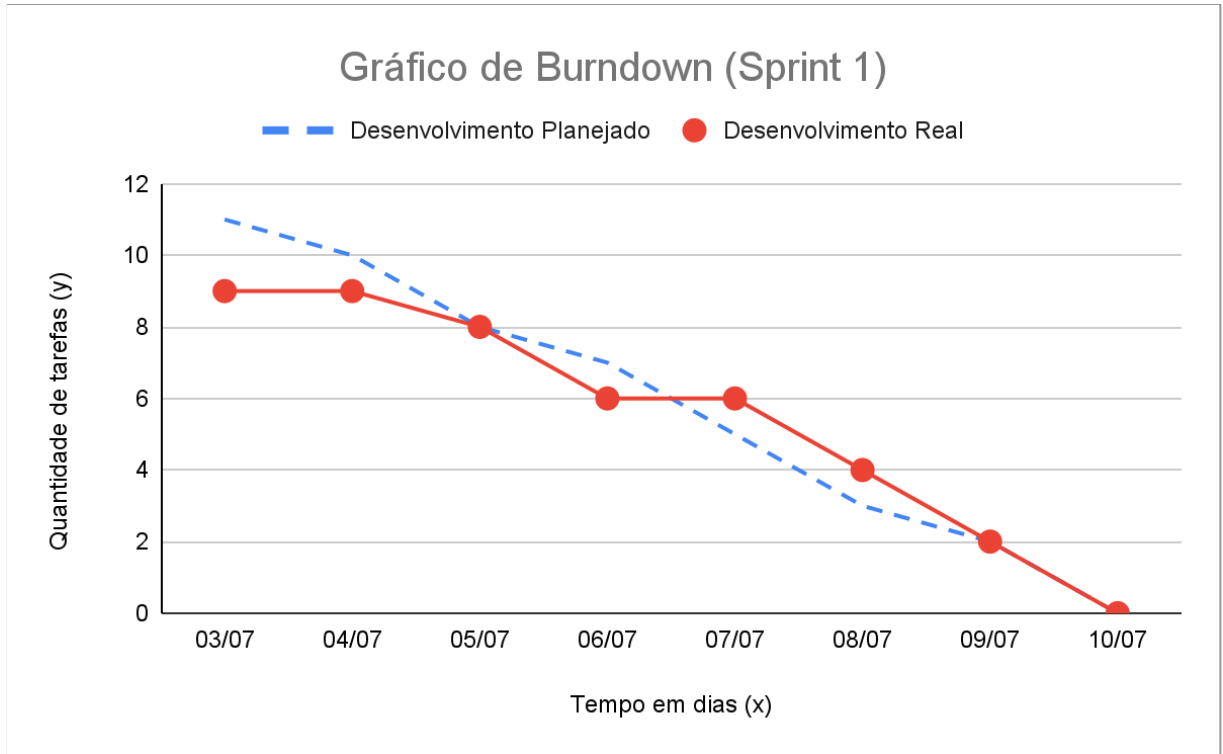


Figura 9. Gráfico de *Burndown* da primeira *sprint*

Percebe-se que no início e no fim da *sprint*, foi possível atingir o desenvolvimento planejado. Já no meio da *sprint*, acabamos perdendo um pouco o prazo e tivemos que acelerar o desenvolvimento para cumprir todos os pontos necessários para a iteração.

- Segunda *sprint*

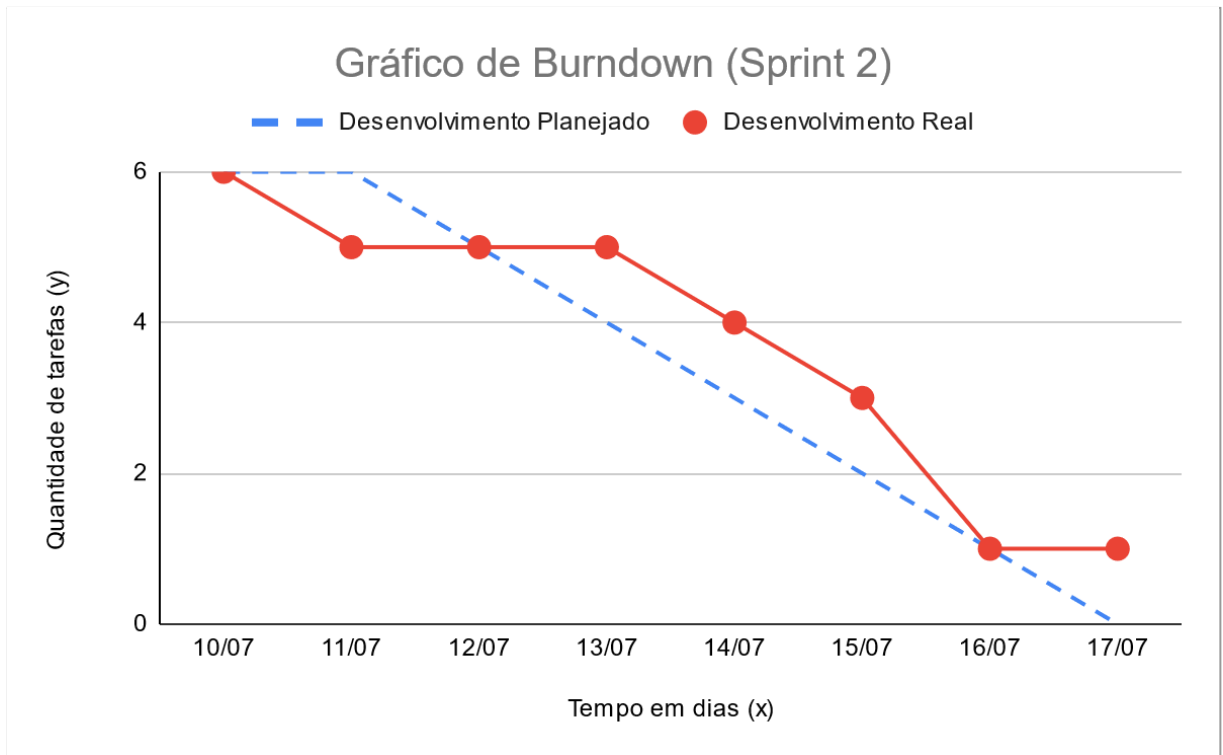


Figura 10. Gráfico de *Burndown* da segunda *sprint*

Nessa *sprint*, tivemos um começo relativamente bom, mas o progresso acabou estagnado e houve um grande atraso que não conseguimos corrigir, resultando em um percentual de conclusão menor do que o inicialmente planejado, por isso tivemos que reduzir em uma atividade a *sprint* 2, e ainda assim, uma das atividades não foi concluída.

- Terceira *sprint*

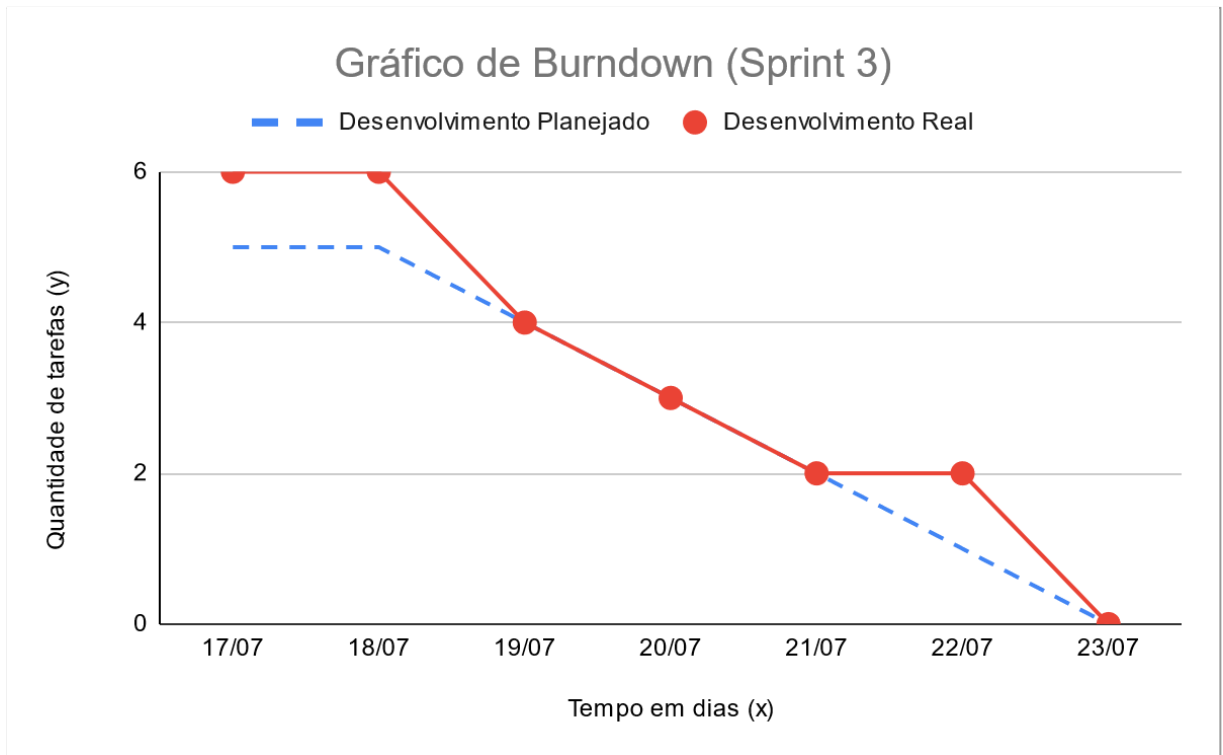


Figura 11. Gráfico de *Burndown* da terceira *sprint*

Como tivemos um atraso na *sprint* anterior e tivemos que passar uma das atividades para a terceira *sprint*, começamos com uma atividade a mais do que o planejado, mas conseguimos nos manter dentro do previsto, completando todas as atividades da *sprint* 3 dessa primeira iteração.



## Iteração 2

- Sprint Única

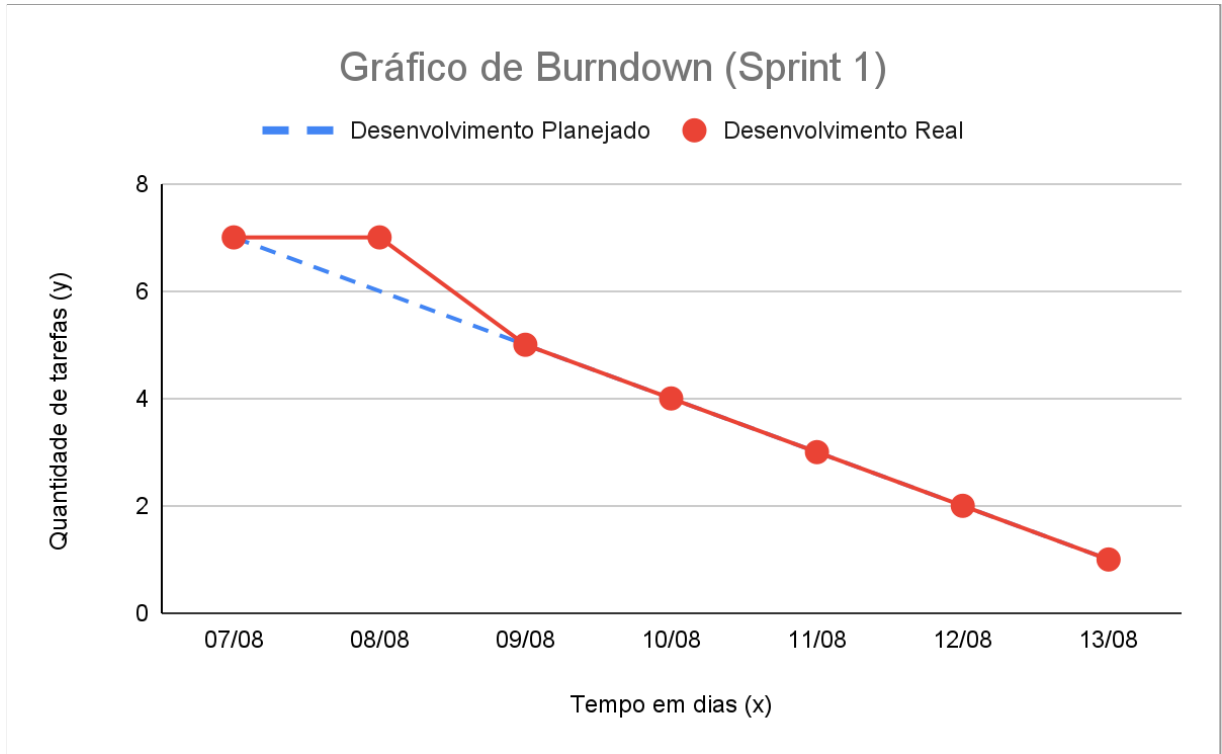


Figura 12. Gráfico de *Burndown* da primeira *sprint*

No começo da *sprint*, tivemos um atraso e percebemos que não seria possível realizarmos o que estava planejado. A equipe cortou o número de atividades, colocando-as para a próxima e última iteração do projeto. Após o corte, foi possível atingir o novo quadro de atividades.

## Plano de Monitoramento de Riscos

Segundo o PMBOK (2004), "o risco é um evento ou condição incerta que, se ocorrer, terá um efeito positivo ou negativo sobre pelo menos um objetivo do projeto, como tempo, custo, âmbito ou qualidade". Listam-se aqui os possíveis riscos detectados, e de acordo com as melhores práticas, seus remédios, contingências e monitoramento. Para tal, adotaremos

a abordagem de análise qualitativa de riscos, de forma que os riscos serão analisados "considerando suas implicações ao cumprimento dos objetivos do projeto" (CAMARGO). Em seguida, serão expostos conforme diagrama de estrutura analítica de riscos e por fim a análise de riscos estruturada, com avaliação de cada um dos riscos e suas possíveis limitações.

## Lista de Riscos

### Externos

- **[01RE]** Adequação às leis de proteção a dados, como a Lei nº 12.965 LGPD (variável de acordo com o alcance de países da plataforma)
- **[02RE]** Variação do dólar, que pode acarretar consequente flutuação nos custos
- **[03RE]** Aceitação do jogo

### Internos

- **[04RI]** Tempo para conclusão do projeto
- **[05RI]** Custo de implantação em produção
- **[06RI]** Recursos Humanos (quantidade de membros da equipe é enxuta)

### Técnicos

- **[07RT]** Variações de versão do framework
- **[08RT]** Produto apresenta complexa relação de regras de negócio

## Estrutura Analítica de Riscos

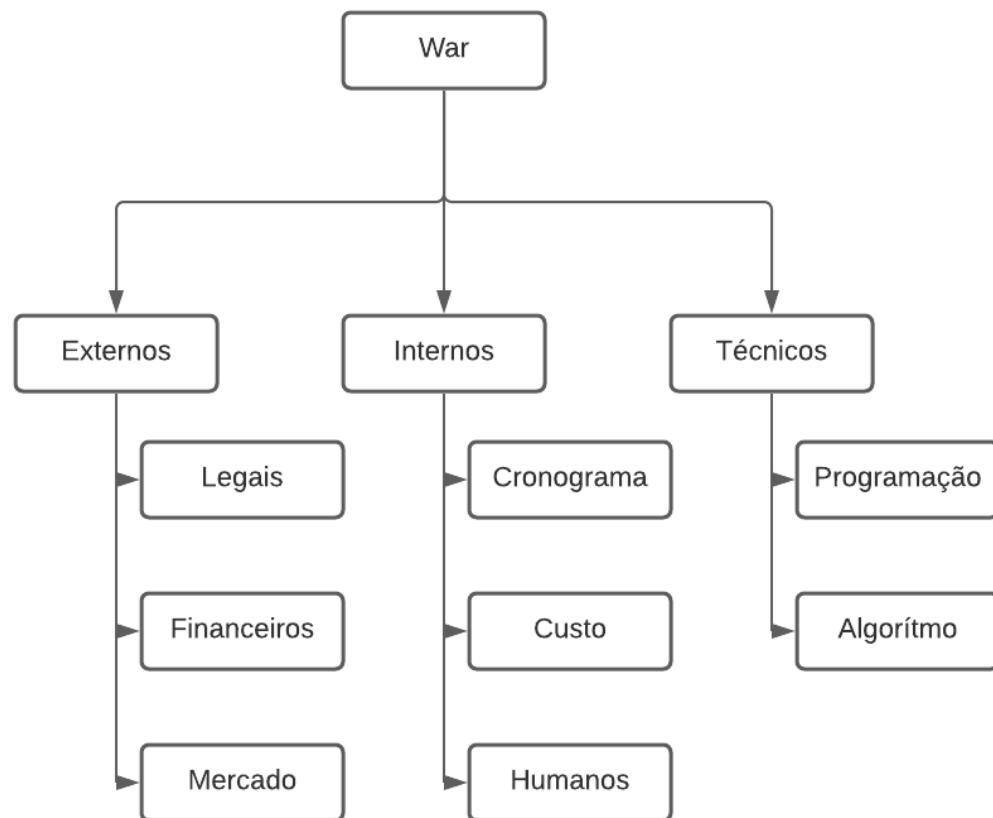


Figura 13. Estrutura de riscos

## Análise de Riscos

Cada risco está disposto em uma observação da tabela abaixo com os seguintes atributos:

- Identificação (Id) - identificação do risco, conforme descrito na lista de riscos, acima
- Categoria (Cat) - cada risco pode ser categorizado como:
  - FIN - Financeiro
  - HMN - Humano

- CRO - Cronograma
- TEC - Técnico
- LEG - Legal
- GST - Gestão
- OPS - Operacional
- LOG - Tecnológico
- MEC - Mercado
- Probabilidade (Prob) - chances de acontecer:
  - 1 - Muito improvável
  - 2 - De certa forma improvável
  - 3 - Chance de 50% de acontecer
  - 4 - Altamente provável
  - 5 - Certeza de acontecer
- Severidade (Sev) - Gravidade do risco, classificados como impacto:
  - 1 - pequeno, na base do projeto
  - 2 - Moderado, na base do projeto
  - 3 - Significativo, na base do projeto
  - 4 - Muito significativo
  - 5 - Desastroso
- Nível de Controle (Con) - Controle que a equipe exerce sobre o projeto:
  - 1 - Evitável, por meio de mitigação
  - 2 - Altamente controlável
  - 3 - Moderadamente controlável
  - 4 - Altamente incontrolável
  - 5 - Totalmente incontrolável
- Significância (Sig) - Soma da probabilidade, severidade e nível de controle:
  - 3-5 - Baixa, pouco impacto sobre o projeto
  - 6-10 - Impacta algumas áreas
  - Acima de 10 - Alta, impacta os objetivos do projeto
- Estratégias para gerenciar o risco (Est) - Ações padronizadas, dadas de acordo com a condição do risco:
  - AC - Aceitar

- EV - Evitar
- MT - Mitigar
- TF - Transferir
- Ações preventivas - Visam reduzir o risco e estão de acordo com a estratégia de gerenciamento de risco. As ações são analisadas e propostas caso a caso.

| Id   | Cat | Descrição                             | Prob | Sev | Con | Sig | Est | Plano de ações preventivas   | Plano de Contenção  |
|------|-----|---------------------------------------|------|-----|-----|-----|-----|--|---|
| 01RE | LEG | Adequação às leis de proteção a dados | 3    | 4   | 1   | 8   | MT  | Nas próximas fases serão analisados modelos de notificação ao usuário sobre uso de dados, e se necessário, contatar consultoria.   | Cumprir os requisitos para a adequação à lei, avisando aos usuários das mudanças, pedindo autorização e aceitação do novo contrato. |
| 02RE | FIN | Variação do dólar                     | 3    | 2   | 3   | 8   | EV  | O máximo de serviços possíveis serão contratados em moeda local, de preferência com previsão de manutenção de preço a longo prazo. |   |
| 03RE | MEC | Aceitação                             | 1    | 3   | 3   | 7   | MT  | Analisar, por meio de <i>beta testers</i> , como o jogo se comporta entre jogadores  | Mudar as características do jogo para aumentar a qualidade  |

|      |     |                        |   |   |   |   |    |  |   |
|------|-----|------------------------|---|---|---|---|----|--|---|
|      |     |                        |   |   |   |   |    | reais. Mudar de acordo com o nível de aceitação.   | seguindo o nível de aceitação, reclamações e pedidos dos jogadores.   |
| 04RI | CRO | Cronograma             | 2 | 3 | 1 | 6 | MT | Realizar reuniões semanais para avaliação do cronograma. Adiantar sempre o que for possível. Evitar ociosidade na equipe.  | Caso haja ociosidade na equipe, alterar a quantidade de tarefas atribuídas para cada um, com o intuito de não atrasar o projeto sem sobrecarregar a equipe. |
| 05RI | FIN | Custo em modo produção | 3 | 4 | 2 | 9 | MT | Propor comparação de custos on-premise versus nuvem, por estratificação de total de jogadores on-line. Mover para solução na nuvem com alta escalabilidade apenas após o jogo apresentar grande quantidade de jogadores (a definir). | Buscar soluções alternativas que não prejudiquem a qualidade, como um todo, do jogo.  |

|      |     |                               |   |   |   |   |    |  |  |
|------|-----|-------------------------------|---|---|---|---|----|--|--|
| 06RI | HMN | Equipe pequena                | 4 | 2 | 3 | 9 | AC | A quantidade de membros na equipe é conhecida desde a aceitação do projeto. Caso um membro saia da equipe, haverá reunião com o sponsor para possibilidade de definição de novo membro.                      |  |
| 07RT | LOG | Mudanças no Framework (React) | 2 | 4 | 3 | 9 | EV | Após a escolha da versão do framework, a equipe seguirá com a mesma até a data de entrega.   | Caso a plataforma de hospedagem mude a versão suportada, trocar-se-á primeiramente a plataforma. |
| 08RT | TEC | Complexidade do produto       | 4 | 4 | 1 | 9 | MT | Caso alguma funcionalidade se faça demasiadamente complexa para implementação, a equipe juntamente com o sponsor, poderá mudá-la de forma a simplificar a regra, e posteriormente (2.0) aplicar a regra mais |  |

|  |  |  |  |  |  |  |  |           |  |
|--|--|--|--|--|--|--|--|-----------|--|
|  |  |  |  |  |  |  |  | complexa. |  |
|--|--|--|--|--|--|--|--|-----------|--|

## Continuidade de Monitoramento

Este plano é reavaliado a cada quinzena, e caso haja alguma alteração nos riscos do projeto, este plano será atualizado para refletir nossa visão de ataque ao problema.

## Relatório

A duração das iterações está sendo medida através das entregas. A primeira entrega foi dividida em 3 sprints de 1 semana, então a iteração teve 3 sprints de duração. A segunda entrega contou somente com uma sprint única. A primeira iteração entregou o documento de análise parcialmente completo, o jogo prototipado, o vídeo de apresentação e os slides. Enquanto a segunda iteração entregou a segunda iteração do documento de análise, o jogo prototipado, o vídeo de apresentação e os slides.

A equipe tem se reunido uma vez por semana para fazer o *poker planning* e *sprint review* e se encontra durante a semana através de sessões virtuais no [Jitsi](#), além da comunicação constante através do WhatsApp.

A equipe de desenvolvimento tem cinco membros que se revezam nos papéis de desenvolvedores, artistas gráficos e também fazem a manutenção e alimentação deste documento.

A arquitetura escolhida foi a *web* para não prendermos a aplicação às restrições dos sistemas operacionais. O jogo está sendo implementado somente com *frontend*, sem a lógica de *backend*, pois para o nosso caso de uso, não é necessário.

## Estratégia de ramificação e controle de modificação

A estratégia de ramificação foi feita da seguinte forma: A *branch* principal é a *Main*. A *branch* de homologação se chama *Homolog* e a de desenvolvimento se chama *DEV*. Toda modificação no código deve ser justificada. Desta forma, para um código novo ser criado, é



necessário criar uma *issue* com as especificações da tarefa. Cada *issue* ganha uma id única (número natural incremental). A partir disso, criamos uma *branch* com o nome *issue#<id\_issue>* e nesta *branch* que será feito o desenvolvimento relativo a essa *issue*. Na hora de fazer um *commit*, é necessário incluir a id da *issue* na mensagem para que ela seja referenciada no histórico da *issue*. Exemplo:

## Colocar background do jogo #5

The screenshot shows a GitHub issue page for the title "Colocar background do jogo #5". At the top, a red "Closed" badge is visible, followed by the text "machadodev opened this issue 3 days ago · 0 comments". The issue history is shown as a vertical timeline. The first entry is a comment by "machadodev" from 3 days ago, with the text "Colocar o background na tela principal do jogo". Below this, "machadodev" added an "enhancement" label. Then, "machadodev" assigned the issue to "machadodev", "Hiaggo", "CeMenezesJunior", and "AlissonWeiss". The next entry, highlighted by a blue rectangular box, shows "machadodev" adding a commit that referenced the issue 3 days ago. The commit message is "background adicionado #5" and the commit hash is "e0a8b40". The final entry shows "machadodev" closing the issue 3 days ago.

Figura 14. Referência da *issue* no *commit*

No retângulo azul, podemos ver como fica na aba *Issues* do Github a referência ao *commit*. Desta forma, podemos manter histórico de quais alterações resolveram aquela *issue*. Após a *issue* ter sido concluída, é feito o *merge* com a *branch* DEV. Desta forma colocamos esta *issue* na fila de entrega desta iteração. Após todo desenvolvimento estar na *branch* DEV, é feito *merge* entre a *branch* DEV e a Homolog. Na *branch* de homologação faremos os testes para verificar se tudo que foi desenvolvido está correto. Após os testes serem realizados e todos estarem corretos, é feito um *Pull Request* da *branch* Homolog para a *branch* Main. Este *Pull Request* deve obrigatoriamente ser aprovado por outro membro da equipe, de forma a existir um *code review* e ter mais um nível de validação e segurança. Após alguém da equipe autorizar o *Pull Request*, o código vai para a *branch* Main. Existe um *webhook* na

*branch Main*, de forma que sempre que um novo *Pull Request* é aprovado, é feita uma comunicação com o [Vercel](https://vercel.com) para avisar que existe novo código para *deploy*. Automaticamente o Vercel faz o *git clone* do projeto na sua última versão da *branch Main* e coloca no ar a versão mais recente da aplicação, que fica disponível em <http://war-pi.vercel.app/>.

Temos abaixo uma representação visual deste *branching workflow*:

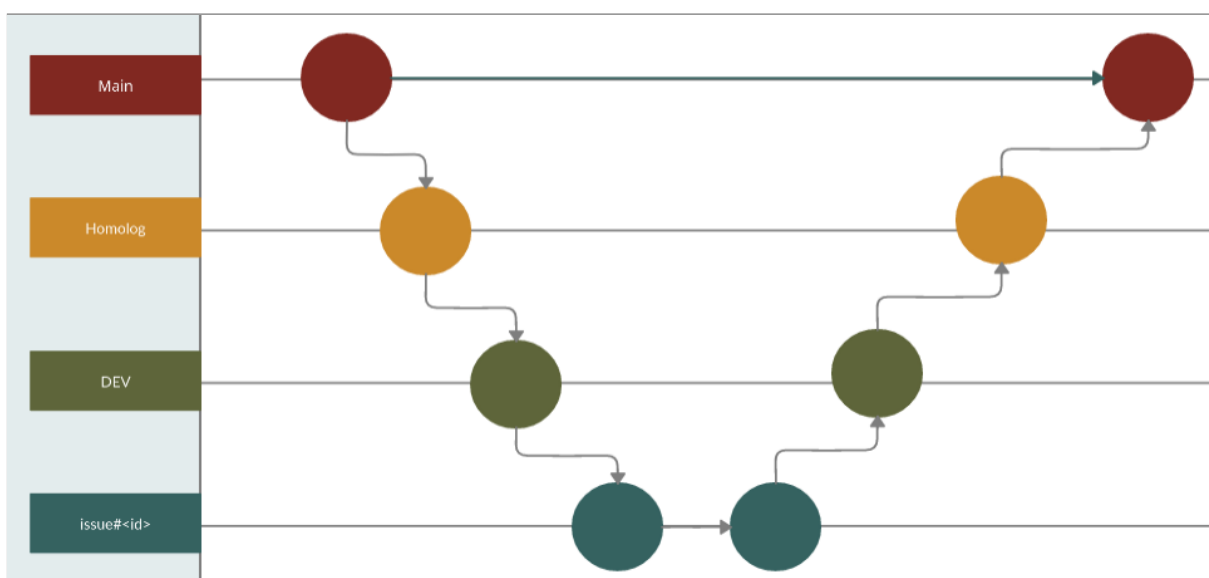


Figura 15. *Branching workflow*

## Contribuições no repositório

O desenvolvimento foi feito na maior parte do tempo em programação em pares, então os *commits* carregam a assinatura de todos que estavam presentes. O computador host do pair programming foi o do Leonardo Machado, então ele ficou como autor principal do commit e os outros membros ficaram como co-autores, porém todos estavam trabalhando ao mesmo tempo. O Github faz o mapeamento deste trabalho em grupo desta forma:



Figura 16. *Commit* em grupo

Trabalhando desta forma, foi possível tirarmos dúvidas uns com os outros em tempo real, pois estávamos também em chamada de áudio e pudemos nos concentrar em cada tarefa com foco total do grupo.

## Bibliografia

**CAMARGO**, Marta Rocha. Gerenciamento de projetos : fundamentos e prática integrada. 2. ed. Rio de Janeiro: Elsevier, 2018.

**PMI**. A Guide to the Project Management Body of Knowledge (PMBOK Guide). Pennsylvania: PMI, 2008