



# Pandemic

Gerência de Projeto e Manutenção de  
Software



**START**    **MENU**

Guilherme Vieira, Isadora Duarte, Izabel Soares, Lylian Pacheco, Mateus  
Sacramento e Nayara Dornelas



# Sumário

## Tópicos

-  [Apresentando o Jogo](#)
-  [Escopo](#)
-  [Requisitos do jogo](#)
-  [EAP](#)

-  [Planning Poker](#)
-  [Gráfico de Gantt](#)
-  [Custo e Orçamento](#)
-  [Listagem de riscos](#)



# Sumário

## Tópicos

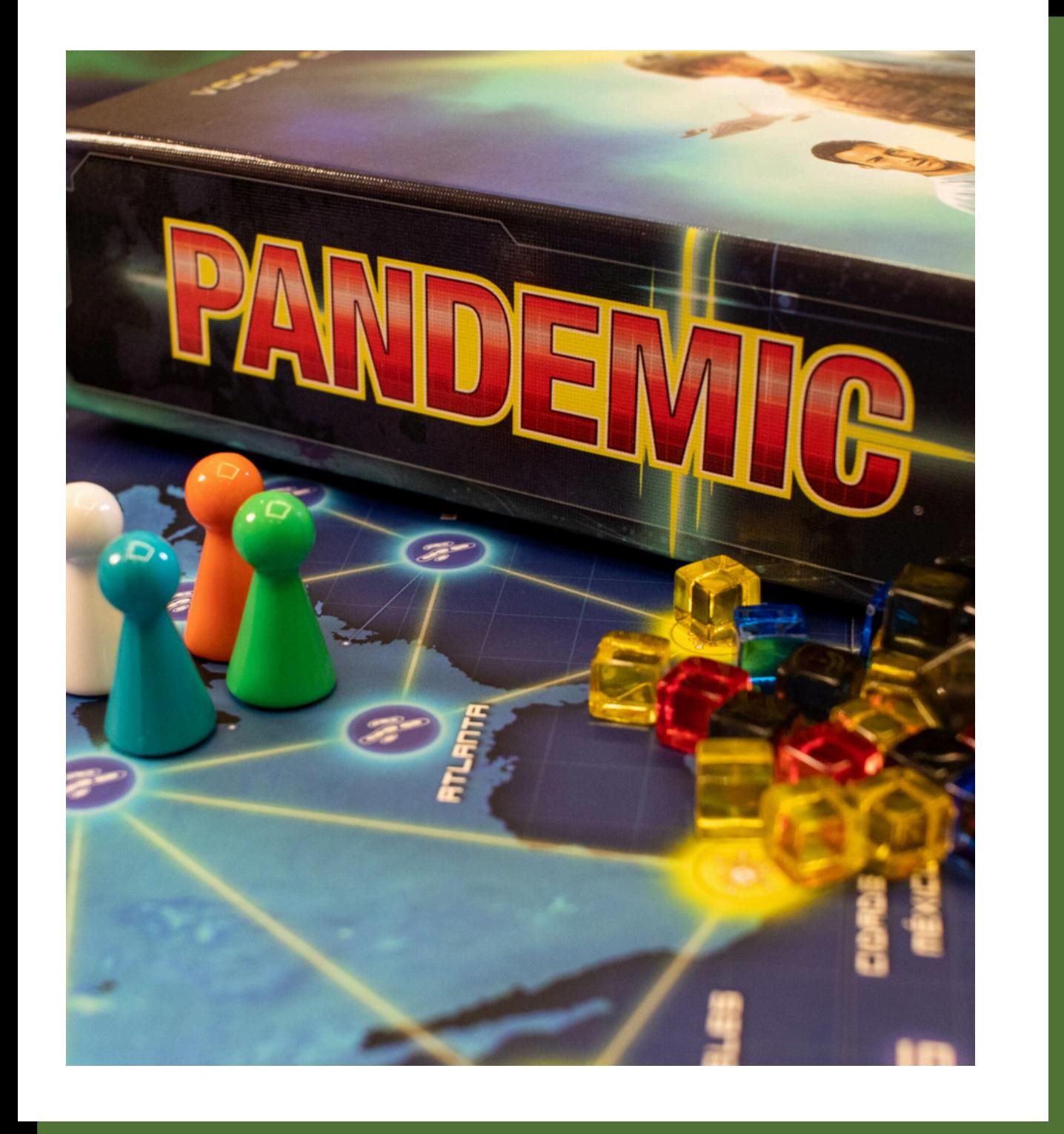
- [Prioridade x Impacto](#)
- [Exposição](#)
- [Priorização](#)
- [Planos de contenção e contingência](#)

- [Burndown por iteração](#)
- [Análise de Valor Agregado](#)
- [Demo Parcial](#)



# Apresentando o Jogo

Em Pandemic, jogadores assumem papéis de especialistas para combater quatro doenças que se espalham pelo mundo. Eles devem trabalhar juntos para tratar focos e pesquisar curas. A vitória acontece se curarem as quatro doenças, mas a derrota ocorre se as doenças saírem de controle ou as cartas se esgotarem.





# Escopo do Produto

## Objetivo

Criar um jogo cooperativo digital, inspirado no Pandemic: Hot Zone – Europa, em que três jogadores devem conter e curar três doenças fictícias que se espalham por 18 cidades do universo Hollow Knight, enfrentando surtos e desafios estratégicos.

## Características e Funcionalidades

- Modo cooperativo para 3 jogadores
- 18 cidades temáticas interligadas
- 3 doenças distintas
- 5 especialistas jogáveis
- Turnos alternados com 4 ações por jogador
- Mecânicas de surtos e epidemias



# Requisitos Funcionais



## Gerenciamento de Partida

- 3 jogadores
- Distribuição das classes dos jogadores
- Preparação do baralho



## Gerenciamento de Tabuleiro

- Representação das cidades no mapa
- Controle do estado de infecção da cidade
- Permissão para colocação/remoção de cubos de doenças



## Gerenciamento de Turno

- Cada jogador tem 4 ações por turno
- Ações possíveis: mover, tratar doenças, compartilhar carta, descobrir cura e usar evento



## Gerenciamento de Vitória/Derrota



São as funções ou comportamentos que o sistema deve realizar. Eles descrevem o que o software/jogo deve fazer.



# Requisitos Não Funcionais



## Usabilidade

- A interface do jogo deve ser intuitiva e fácil de usar, mesmo para novos jogadores



## Desempenho

- O sistema deve responder rapidamente às ações dos jogadores.



## Design

- Arte temática (mapa estilizado, cores distintas para doenças).



## Segurança

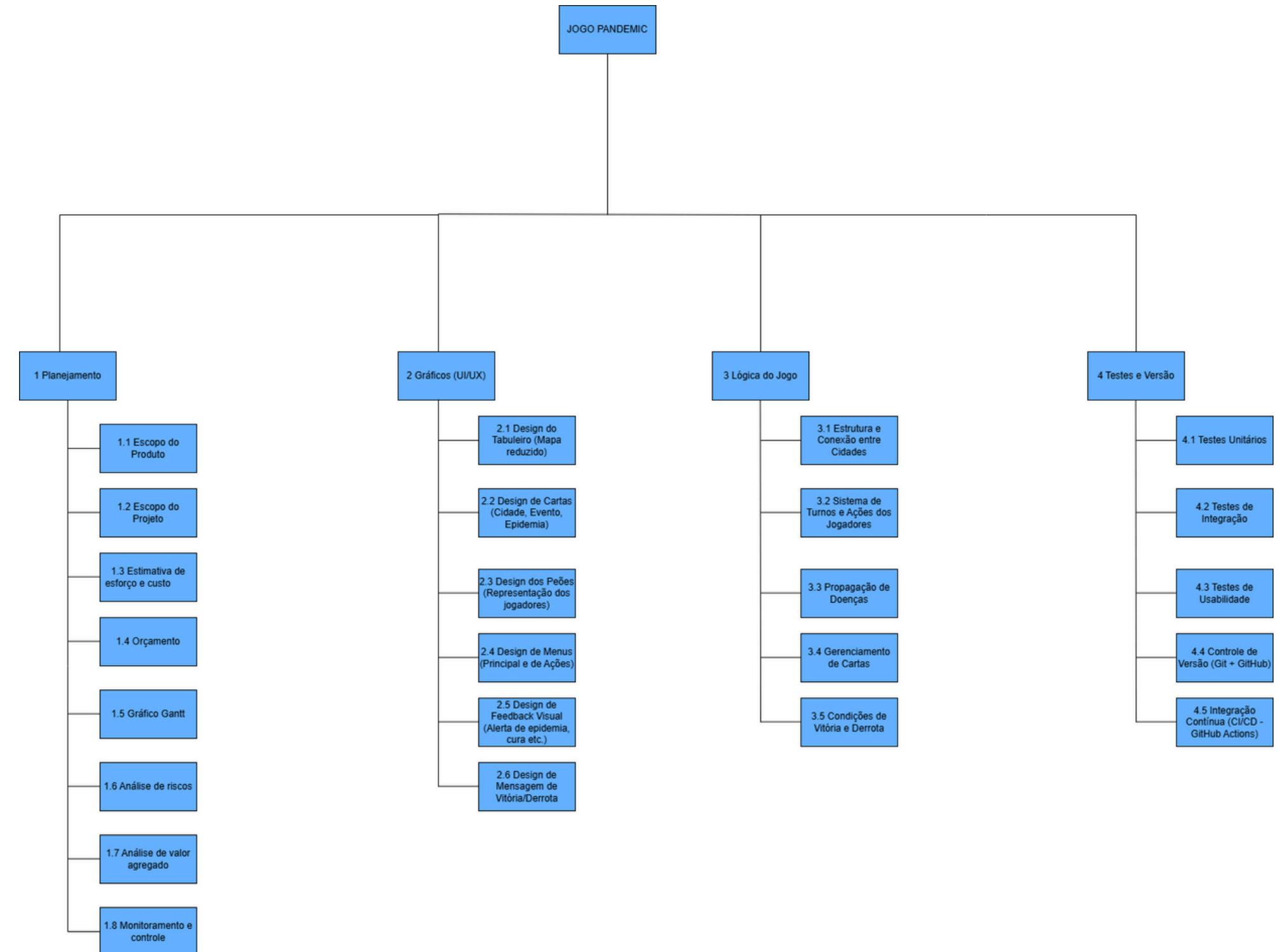
- Os dados dos jogadores devem estar protegidos e acessíveis apenas a eles.
- Proteção contra trapaças.



São relacionados a como o sistema deve se comportar, e não o que ele faz. Estão ligados à qualidade, desempenho, usabilidade, segurança, etc.



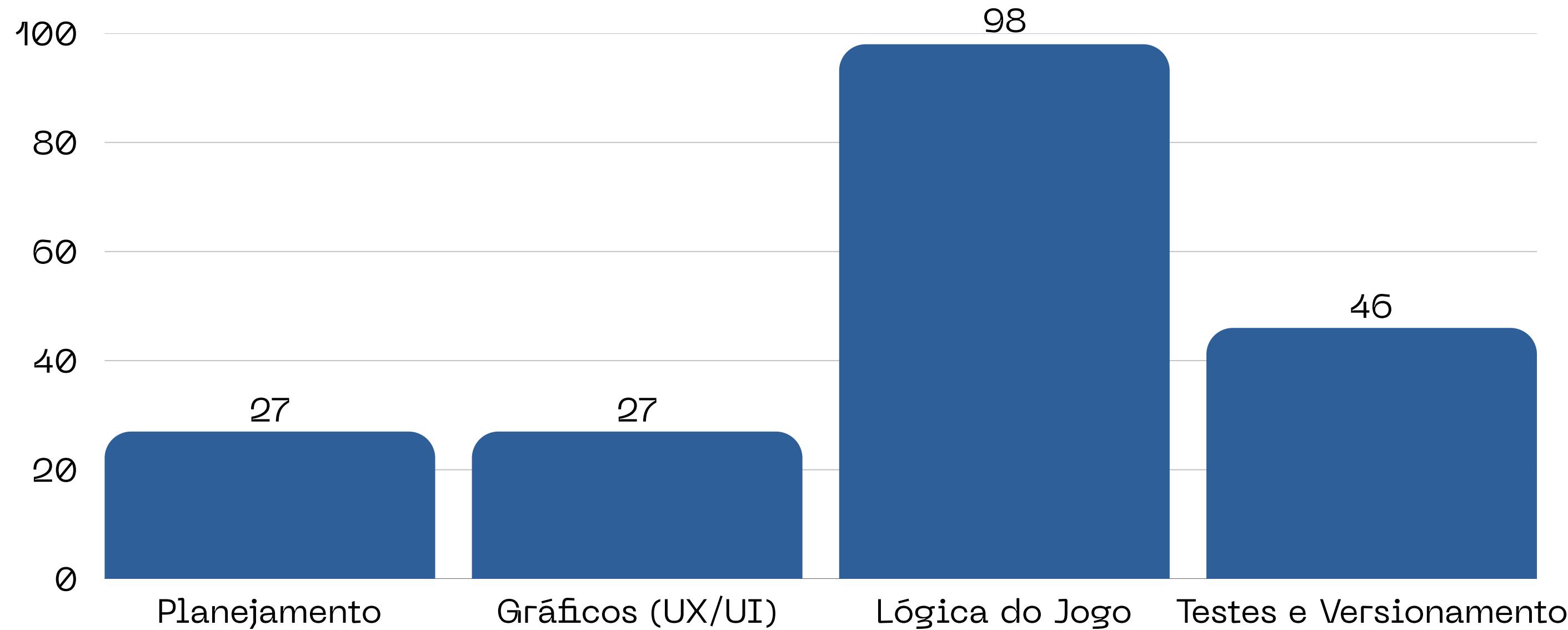
# EAP





# Planning Poker

● Homem-Noga





# Gráfico de Gantt

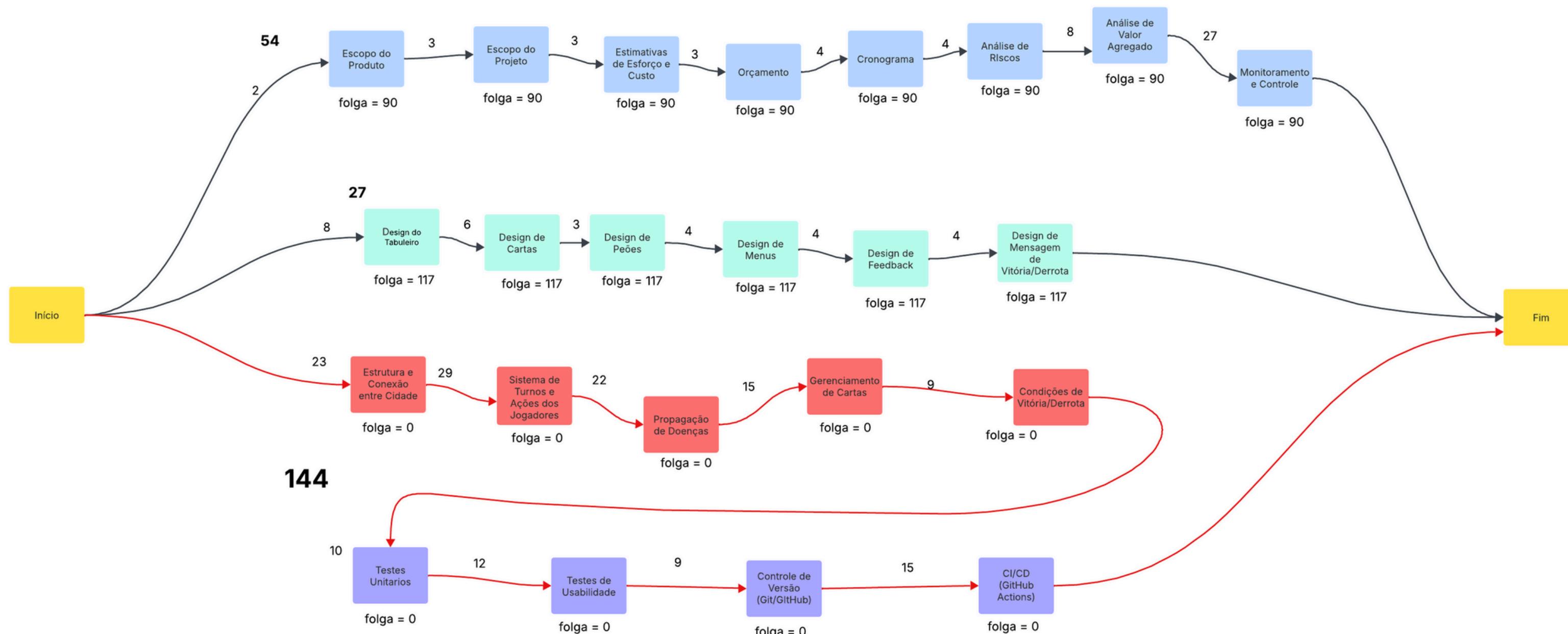
# PANDEMIC - GRÁFICO DE GANTT

Dica do  
Smartsheet →

O cronograma visual do gráfico de Gantt mostra os detalhes de cada tarefa e as necessidades do projeto.



# Caminho Crítico





# Custo e orçamento

Soma total de Homem-Horas (Planning Poker)

- Planejamento = 54
- Gráficos (UI/UX) = 27
- Lógica do Jogo = 98
- Testes e Versão = 46

Total estimado = 225 homem-horas

Custo estimado do projeto

- $225 \text{ horas} \times \text{R\$ } 20/\text{hora} = \text{R\$ } 4.500,00$

Orçamento sugerido tendo uma margem de segurança de 10% a 20% sobre o custo estimado para imprevistos, revisões ou retrabalho.

Margem de 15%:

$$\text{R\$ } 4.500 \times 1,15 = \text{R\$ } 5.175$$

## Resultado Final

Item	Valor
Custo Estimado	R\$ 4.500
Orçamento Sugerido (15% margem)	R\$ 5.175



# Mitigação

## Risco 1: atraso no cronograma do projeto

### Contenção:

Estabelecer um cronograma detalhado e realista, revisando e atualizando semanalmente

Realizar reuniões de alinhamento para avaliar o progresso e resolver impedimentos  
Dividir o projeto em sprints com entregas pequenas.

Utilizar uma ferramenta de controle de tarefas

### Contingência:

Negociar ajustes de escopo ou prazos  
Redistribuir tarefas e alocar recursos extras em áreas críticas  
Priorizar funcionalidades essenciais.

## Risco 2: definição inadequada dos requisitos e escopo

### Contenção:

Realizar uma coleta detalhada e validação dos requisitos em conjunto com toda a equipe  
Analizar projetos similares para ajustar o escopo do jogo de forma realista.

### Contingência:

Reservar parte do cronograma para revisões e ajustes dos requisitos  
Reavaliar o escopo e, se necessário, realizar uma redefinição das funcionalidades

Monitoramento: revisões regulares a cada fase do projeto para garantir que os requisitos estejam sendo cumpridos e atendidos



# Mitigação

## Risco 3: Falta de alinhamento e comunicação entre os membros

### Contenção:

Definir claramente as funções e entregáveis de cada membro logo no início.

Criar um canal de comunicação direta entre os membros do grupo.

### Contingência:

Nomear um responsável por centralizar a comunicação  
Convocar reuniões extraordinárias para realinear objetivos e prazos.

## Risco 4: Problemas técnicos no desenvolvimento do software

### Contenção:

Escolher um escopo e uma arquitetura simples e documentada desde o começo.

Fazer revisões técnicas semanais para identificar e corrigir problemas cedo.

Garantir que pelo menos 2 membros tenham domínio das tecnologias que serão utilizadas.

### Contingência:

Buscar suporte externo.  
Adaptar a solução para alternativas técnicas viáveis (por exemplo, simplificar algumas funções ou regras, se necessário).

Monitoramento: Fazer testes contínuos.

# Mitigação



## Risco 5: Baixa produtividade dos membros da equipe

### **Contenção:**

Trabalhar com a divisão de tarefas menores para ser mais fácil de concluir e dar maior sensação de progresso

### **Contingência:**

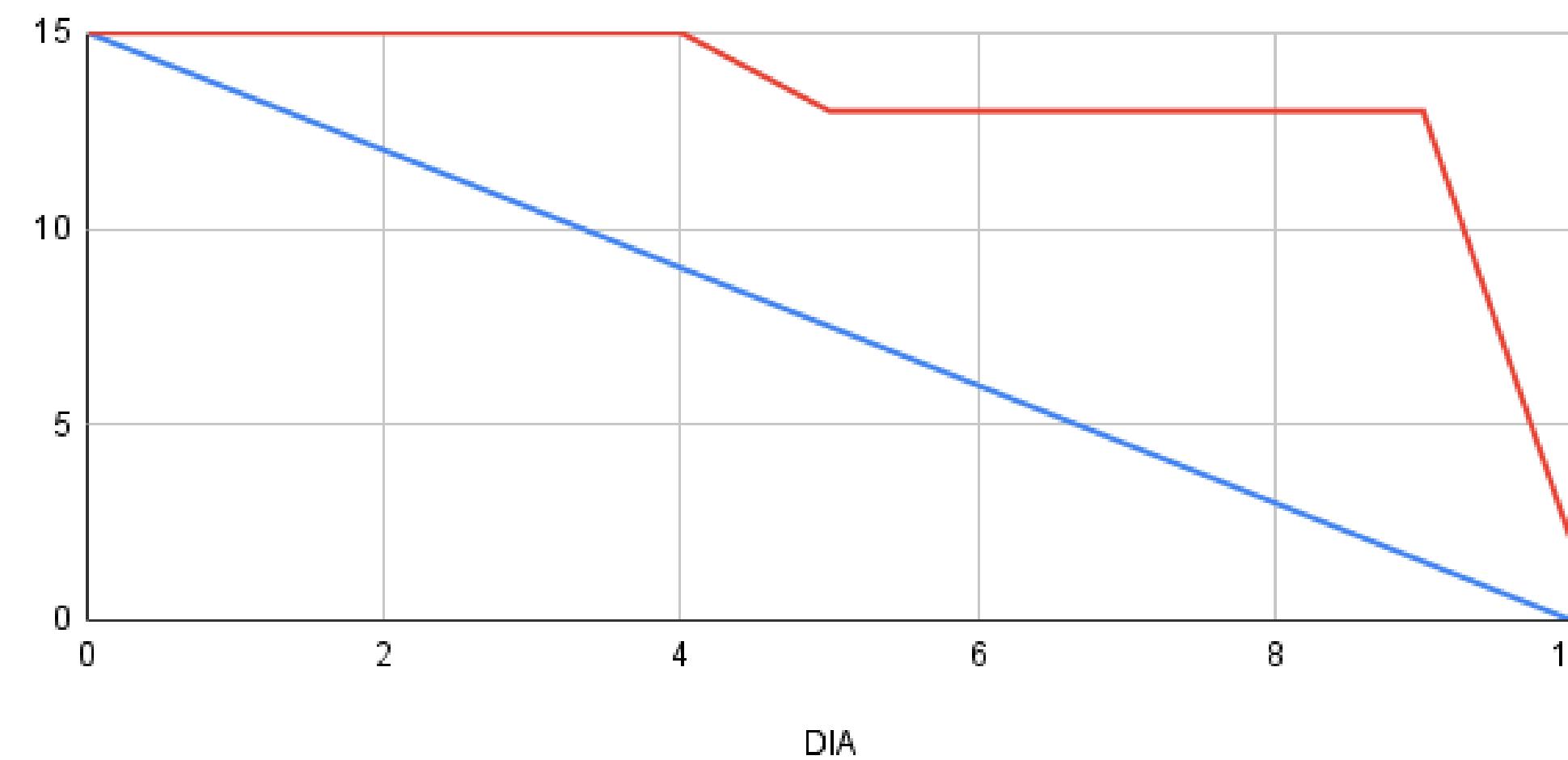
Redistribuir tarefas críticas para membros mais produtivos, priorizando funcionalidades essenciais



# Gráfico de Burndown

Burndown Sprint 1

— HORAS PLANEJADAS — HORAS FALTANTES

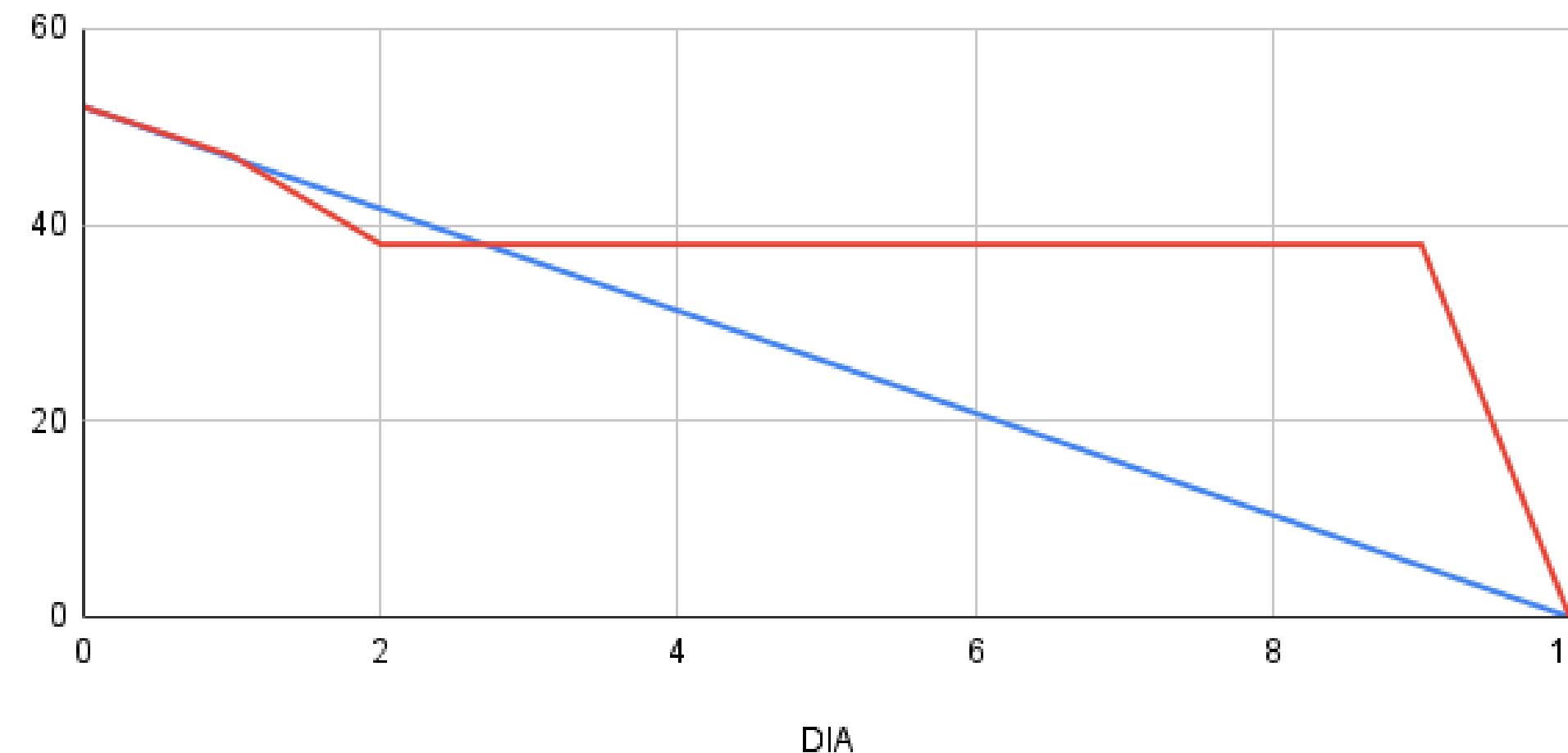




# Gráfico de Burndown

Burndown Sprint 2

— HORAS PLANEJADAS — HORAS FALTANTES





# Análise de Valor Agregado

## Até a Sprint 2

Indicador	Valor
PV (Planejado)	67h
EV (Executado)	55h
Variação de Cronograma (SV = EV – PV)	-12h
Índice de Desempenho de Prazo (SPI = EV / PV)	55 / 67 ≈ 0,8209

# Demo parcial



## ESPECIALISTAS

~~

**KNIGHT**

Impede infecção na sua cidade e nas cidades conectadas.

~~

~~

**CORNIFER**

Pode dar qualquer carta para outro jogador, independentemente da cidade.

~~

~~

**HORNET**

Remove todos os cubos de uma doença com apenas 1 ação.

~~

~~

**GRIMM**

Pode mover outros jogadores como se fossem o próprio especialista, gastando cartas dele.

~~

~~

**MONOMON**

Descobre cura com apenas 3 cartas da mesma doença.

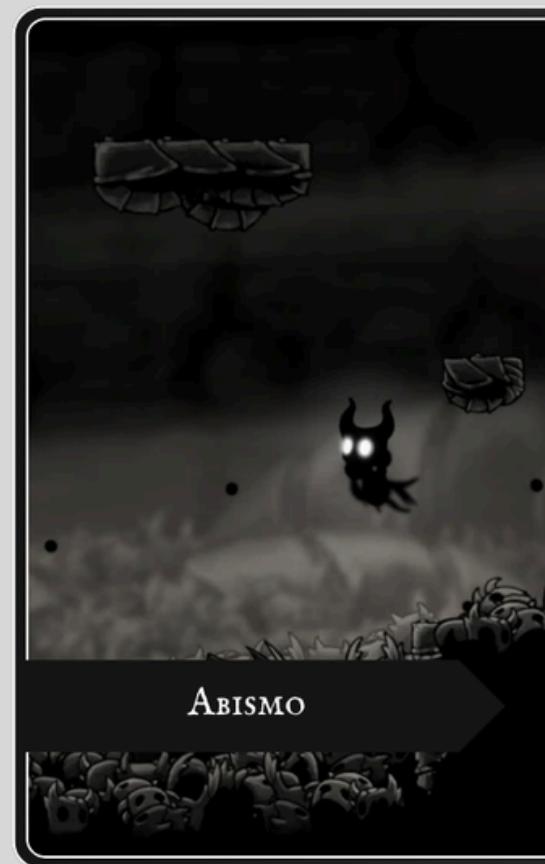
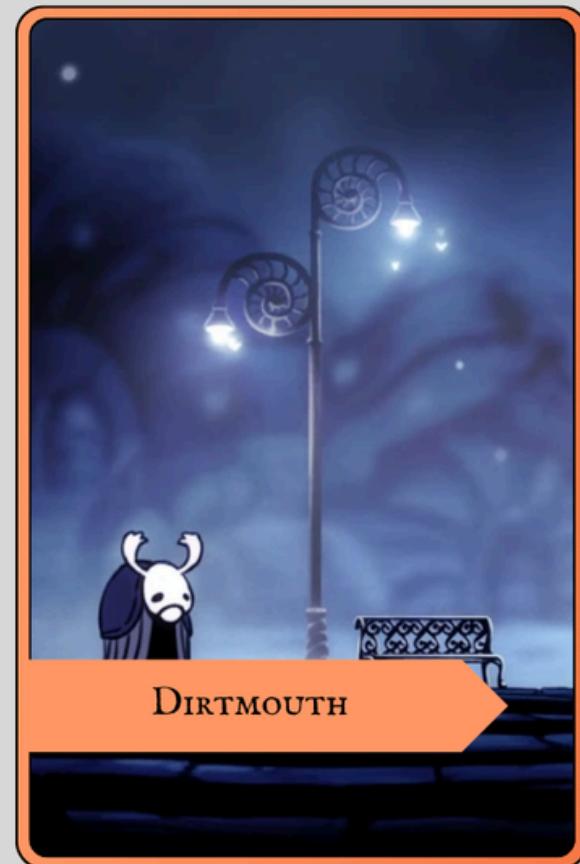
~~



# Demo parcial



## CIDADES





# Obrigado!





# GitHub

[Link do repositório](#)

