# WAR GAME

## GRUPO 2

ALUNOS:

Allber Fellype Soares Ferreira
Ayrton Surica
Caio Mota Del Aguila
Carlos Alberto Santos da Anunciação Junior
Felipe Figueiredo Felisbino Branquinho
Flávio Matheus Pereira Francisco
Henrique Neves Braga
Pedro Henrique Ferreira Amaro

# Sumário:

# Relembrando

Adaptando o clássico de estratégia para o mundo digital

- **Objetivo:** Replicar as regras e a dinâmica do jogo de tabuleiro War.
- **Funcionalidade Principal:** Permitir que jogadores participem de partidas estratégicas online com amigos ou outros jogadores.
- **Proposta de Valor:** Uma experiência imersiva e competitiva a qualquer momento e em qualquer lugar.

# Controle de versões

# Controle de Versão do Projeto

Gerenciando o Código com Git e GitHub

O Git é a espinha dorsal do nosso processo de desenvolvimento, sendo o sistema de Controle de Versão Distribuído (DVCS) mais utilizado no mundo.

Ele foi escolhido por ser um padrão de mercado e por oferecer a segurança e rastreabilidade necessárias para um projeto como esse.

Optamos por separar o código em dois repositórios distintos:

- **war-backend:** Contém toda a lógica de negócio, API RESTful, banco de dados (Spring Boot/Java). `https://github.com/ES2-War-Game/war-backend.git`
- **war-frontend:** Contém a interface do usuário (UI) e a lógica de apresentação (React/JavaScript). `https://github.com/ES2-War-Game/war-frontend`

Existe também um repositório para documentação: https://github.com/ES2-War-Game/war-docs

# Controle de modificações

# Controle de Modificações

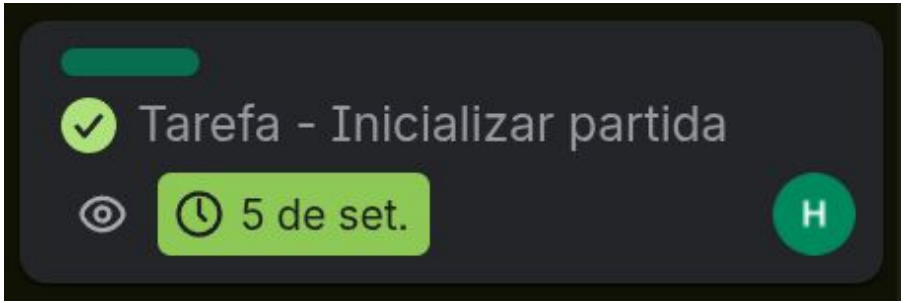
Utilizando Trello

Papel do controle de modificação: Conjunto de processos para gerenciar e documentar alterações em um sistema, produto ou projeto.

Objetivos:

- Garantir que toda alteração seja registrada.
- Exigir aprovação antes da implementação.
- Manter a rastreabilidade do porquê, quem e quando a mudança ocorreu.

No Trello: Cada Card representa uma Solicitação de Modificação.



# Mapeando o Workflow no Trello

1. Backlog: Onde as modificações são propostas e aguardam priorização.
2. A fazer: Modificações que devem ser implementadas na iteração atual.
3. Em Desenvolvimento: A mudança está sendo implementada.
4. Revisão da tarefa: A mudança está pronta para ser validada ou aprovada.
5. Concluído: A modificação foi aplicada ao ambiente de produção.

O Trello registra automaticamente cada movimentação do Card entre as Listas.

# Estratégia de ramificações

# Nossa Estratégia de Ramificação

Utilizamos uma variação do modelo Git Flow (ou Git Flow Simplificado), adaptado às necessidades do nosso projeto.

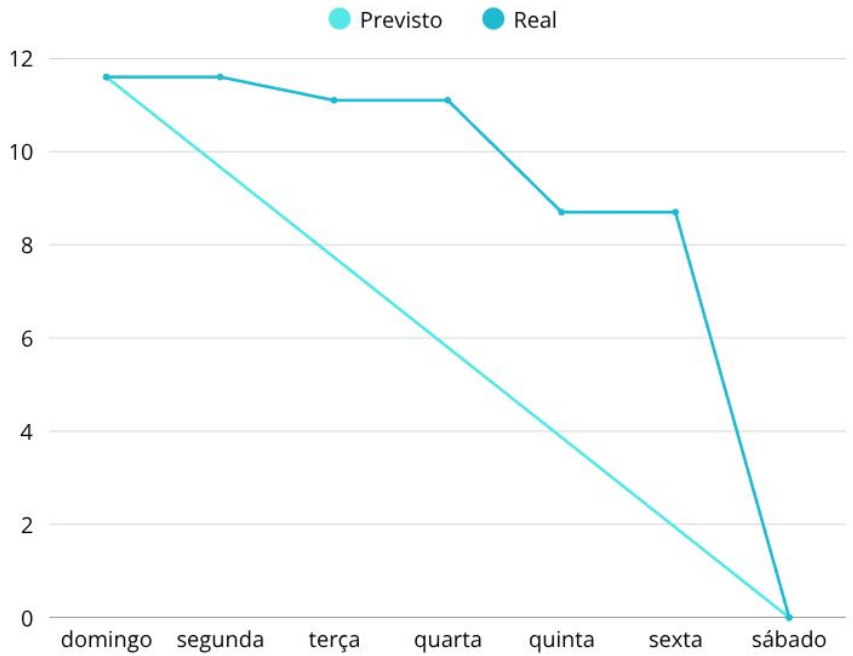
- **main:** Contém o código estável e pronto para produção.
- **feature/<nome-da-feature>:** Para o desenvolvimento de novas funcionalidades.
- **fix/<nome-da-fix>:** Para conserto de bugs.

Fluxo Típico:

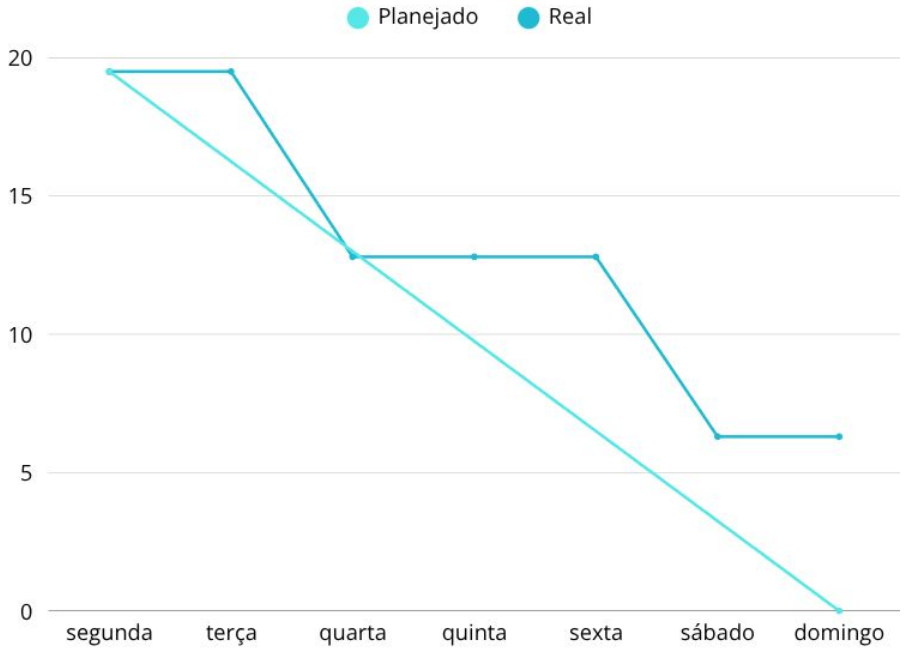
1. Criar feature/X a partir da main.
2. Desenvolver e fazer commits na feature/X.
3. Abrir um Pull Request de feature/X para main.
4. Após revisões e aprovação, mesclar em main.

# BURNDOWN

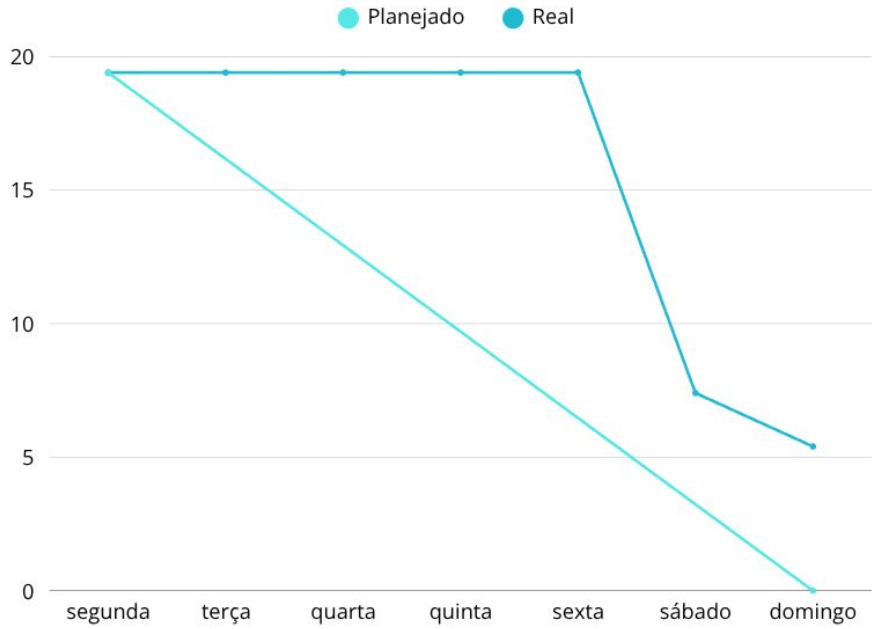
# Quarta iteração



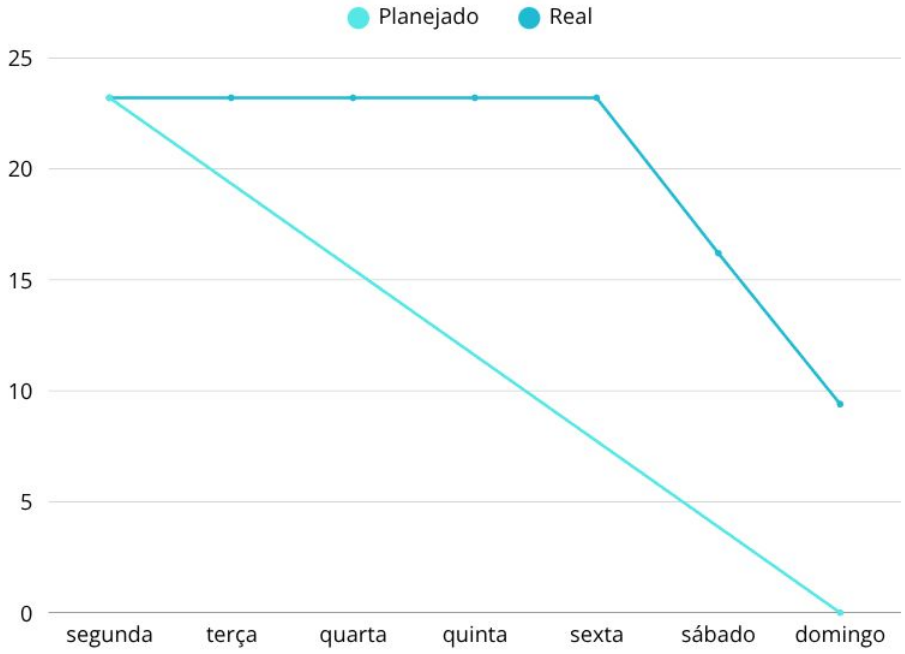

# Quinta iteração



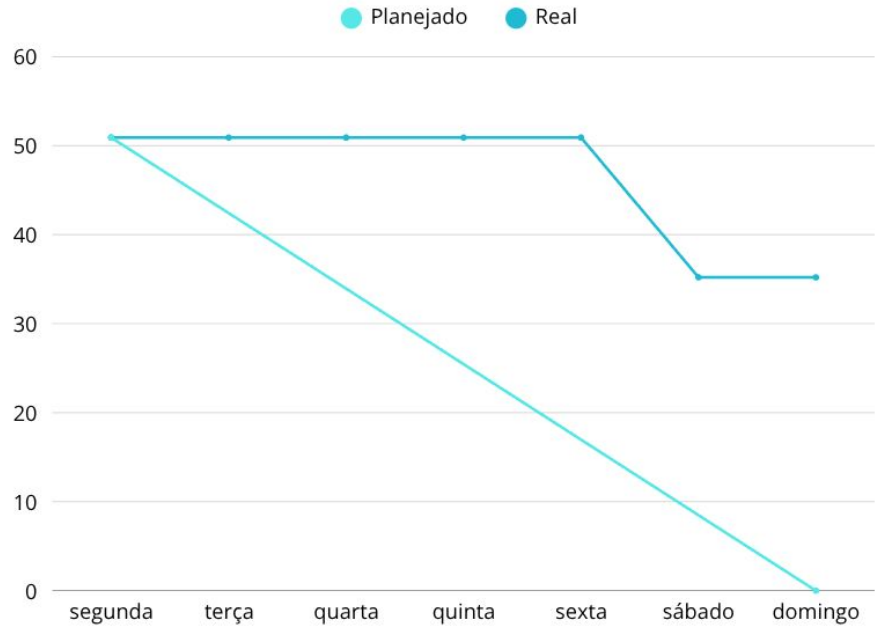
# Sexta iteração



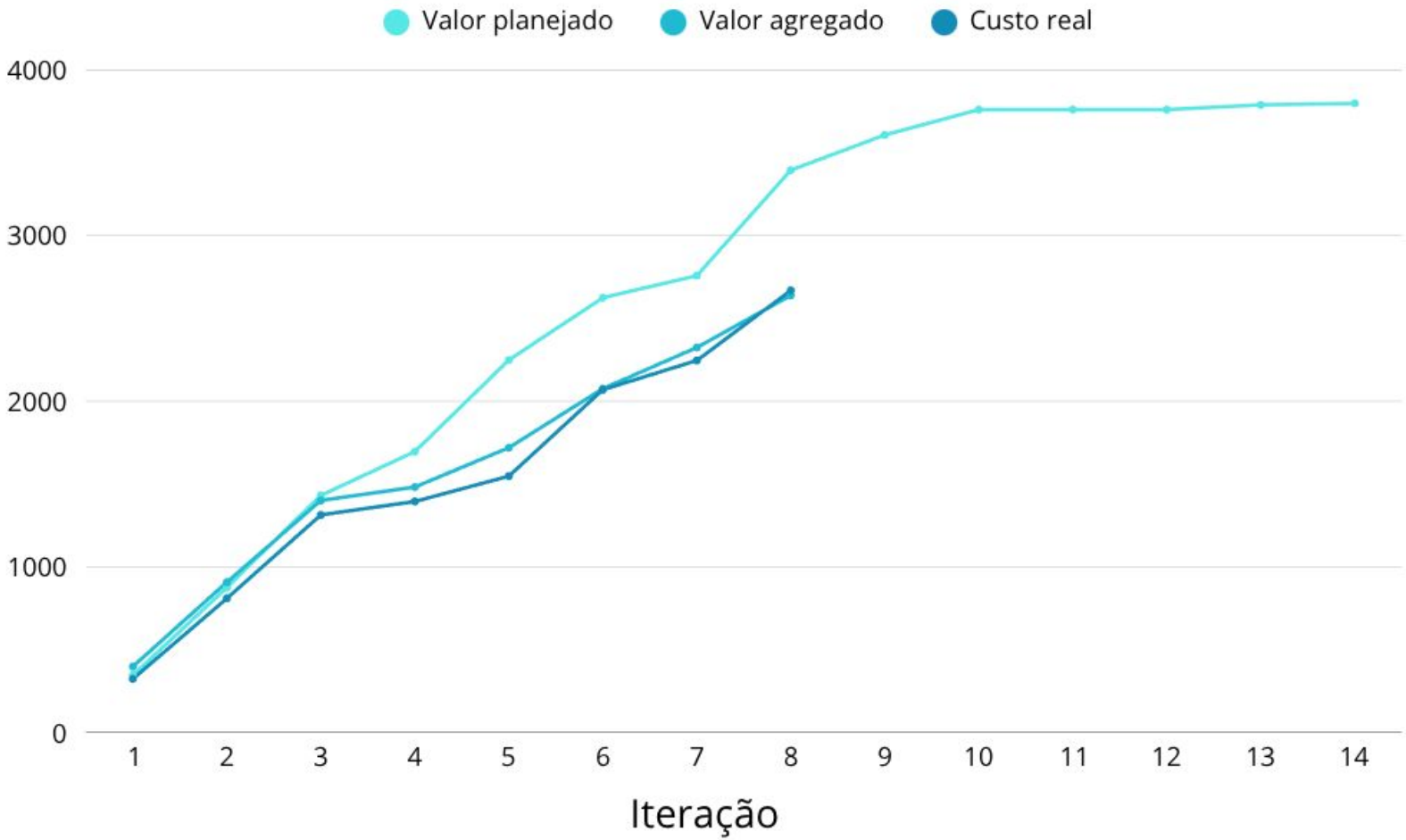

# Sétima iteração



# Oitava iteração



# ANÁLISE DE VALOR AGREGADO


Valor planejado
Valor agregado
Custo real
4000
3000
2000
1000
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Iteração

# ESTADO ATUAL DO PROJETO