

Primeiro seminário de ES2

Daniel Marcondes Bougleux Sodré
José Victor de Paiva e Silva
André Fernandes Gonçalves
Juan Müller Pereira Evangelista
Lucas Tavares Sousa

Escopo do produto





Requisitos Funcionais

Requisitos de UX e UI:

- **RF03** O sistema deve ter um chat de texto durante o jogo que permita aos participantes se comunicarem durante a partida.
- **RF04** O sistema deve permitir que o jogador mude as configurações de apresentação do jogo: volume do som, tela cheia ou modo janela.
- **RF05** O sistema deve ser capaz de apresentar uma interface distinta para o usuário que não tem permissão para jogar.
- **RF06** O sistema deve permitir que um jogador veja dados estatísticos como número de vitórias, derrotas e total de partidas jogadas para cada jogador que está na partida e na sala de Multiplayer.



Requisitos Funcionais

Requisitos Sobre as Regras do Jogo

Requisitos de Single Player:

- **RF29** Jogar no modo offline, contra Bots apenas.
- **RF30** jogadores no modo convidado tem acesso apenas a este modo de jogo.



Requisitos Funcionais

Requisitos do Multiplayer:

- **RF31** Caso o número de jogadores seja menor do que a quantidade mínima requerida (4 jogadores), o sistema deve utilizar de bots para suprir a quantidade necessária de jogadores.
- **RF32** O modo multiplayer deve ser online, com os jogadores se conectando uns aos outros através da internet, em uma sala.
- **RF33** Deve haver um saguão geral, onde todas as salas abertas ficam expostas.
- **RF34** O jogador deve ser capaz de abandonar a partida na qual está participando.
- **RF35** O sistema deve ser capaz de substituir o jogador que abandonou a partida por uma IA.



Requisitos Funcionais

Requisitos de Histórico de Partidas:

- **RF38** O sistema deve possuir um histórico de partidas de cada jogador com pelo menos as partidas jogadas por ele durante os últimos 30 dias.
- **RF39** O histórico de partida deve conter quem venceu o jogo, se o jogador foi vencedor ou perdedor, qual a duração da partida e qual foi o objetivo do vencedor.
- **RF40** O sistema deve ser capaz de armazenar dados estatísticos sobre os jogadores como número de vitórias, derrotas e partidas jogadas.



Requisitos não Funcionais

4.2.1. Disponibilidade:

- **DS01** O sistema online deve estar disponível em 98% do tempo nos dias úteis, das 18h às 23h, e 99% nos dias não úteis das 12h às 23h.

4.2.2. Portabilidade:

- **PR01** O jogo deve rodar no Windows.

4.2.3. Eficiência:

- **EF01** O jogo não deve ocupar mais de 1GB de RAM.

4.2.4. Segurança:

- **SE01** Alterações no banco de dados só devem ocorrer quando o usuário estiver logado.

4.2.5. Usabilidade:

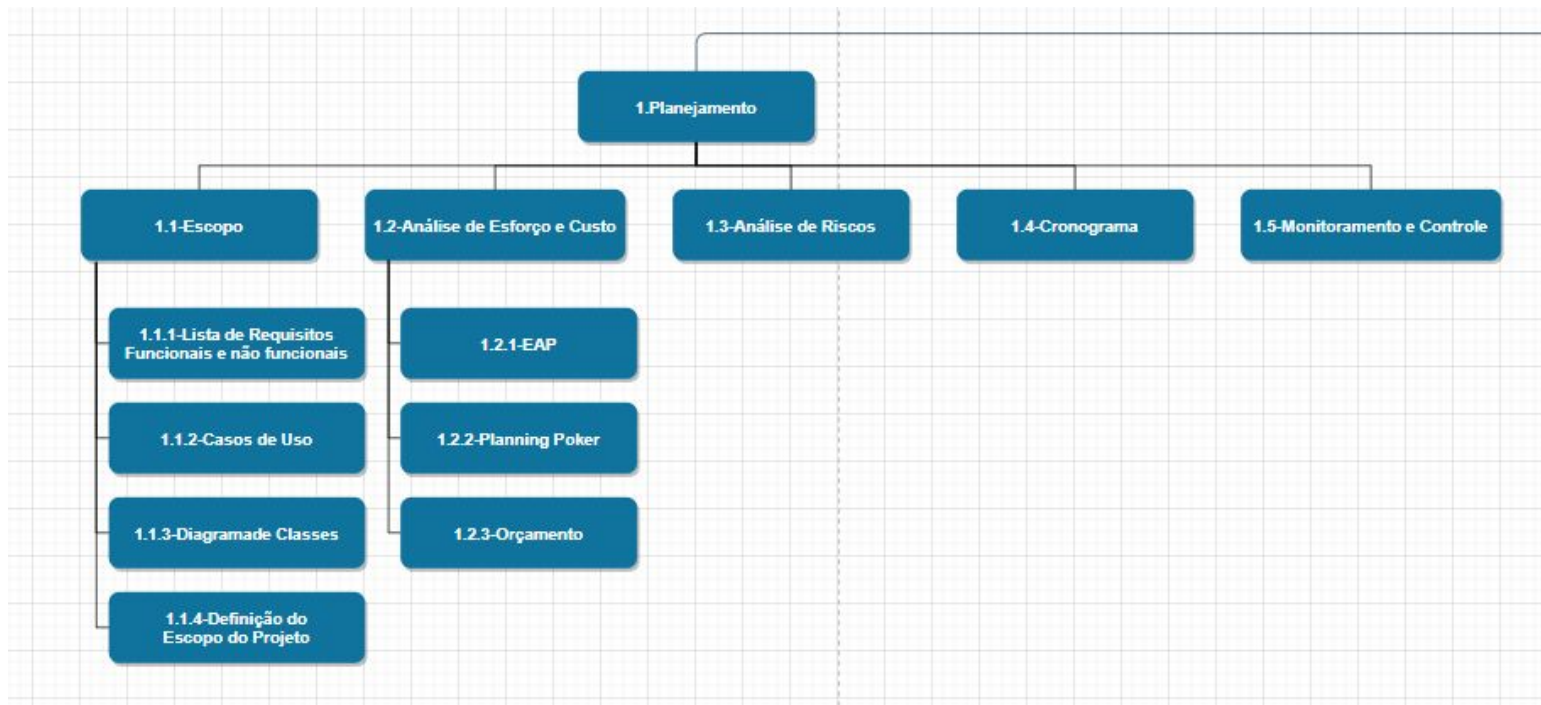
- **US01** Um jogador que não conhece o jogo deve ser capaz de criar ou entrar em uma partida em até 5 minutos.

Estimativas de esforço e custo



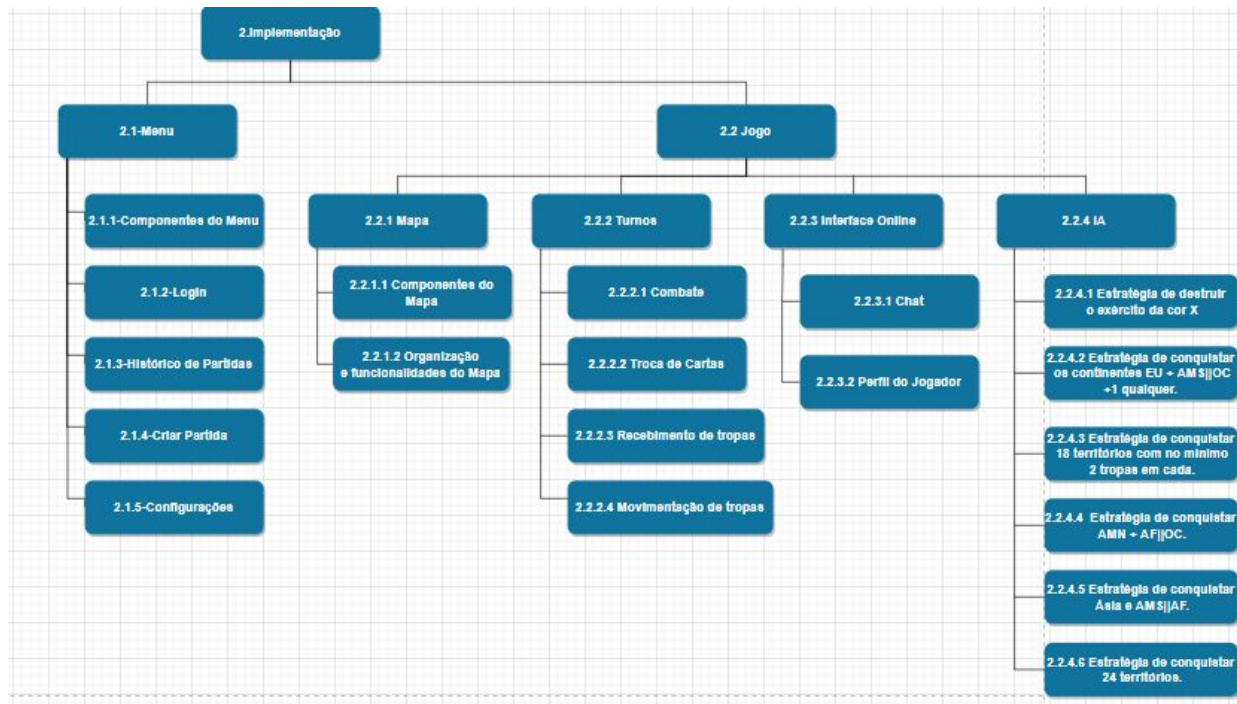


EAP





EAP





EAP





Planning Poker

Submit estimate

?

0

0.5

1

2

3

Submit

Submit

Submit

Submit

Submit

Submit

5

8

13

20

40

100

Submit

Submit

Submit

Submit

Submit

Submit

Results

SHOW

CLEAR ROOM

DELETE ESTIMATES

Name	Story Points
● Daniel	-



Dicionário do EAP

	Pacote de Trabalho	Descrição	Estimativa de esforço (Homem-Hora)	Dependências	Custo Estimado
1.1.1	Lista de Requisitos funcionais e não funcionais	Criar um documento e colocar os requisitos funcionais e não funcionais do sistema.	5	Não Possui	R\$ 83,35
1.1.2	Casos de Uso	Gerar um documento contendo os Casos de Uso (interação do sistema com o usuário).	6	Não Possui	R\$ 100,02
1.1.3	Diagrama de Classes	Baseado nos requisitos e casos de uso, gerar um diagrama de classes em nível de análise.	10	1.1.1, 1.1.2	R\$ 166,70

Escopo do Projeto





Escopo do Projeto

- Justificativa do Projeto
- Finalidade do Projeto
- Objetivo(s) do Projeto
- Stakeholders do Projeto
- Descrição do Produto



Escopo do Projeto

- Entregas do Projeto
- Estimativas de Tempo e Custo
- Exclusões do Projeto
- Critérios de Aceitação
- Premissas
- Restrições

Orçamento





Orçamento

Total de Esforço em Atividades:	253
Custo por Devs/Hora	R\$ 16,67
Custo total das Atividades:	R\$ 4.217,51
Orçamento:	R\$ 4.639,26



Orçamento

Pacote de Trabalho	Estimativa de esforço (Homem-Hora)	Dependências	Custo Estimado
Lista de Requisitos funcionais e não funcionais	5	Não Possui	R\$ 83,35
Casos de Uso	6	Não Possui	R\$ 100,02
Diagrama de Classes	10	1.1.1, 1.1.2	R\$ 166,70
Descrição do escopo do projeto	5	1.4, 1.2.3	R\$ 83,35
EAP	6	1.1.1, 1.1.2	R\$ 100,02
Planning Poker	5	1.2.1	R\$ 83,35

Cronograma



Diagrama CPM

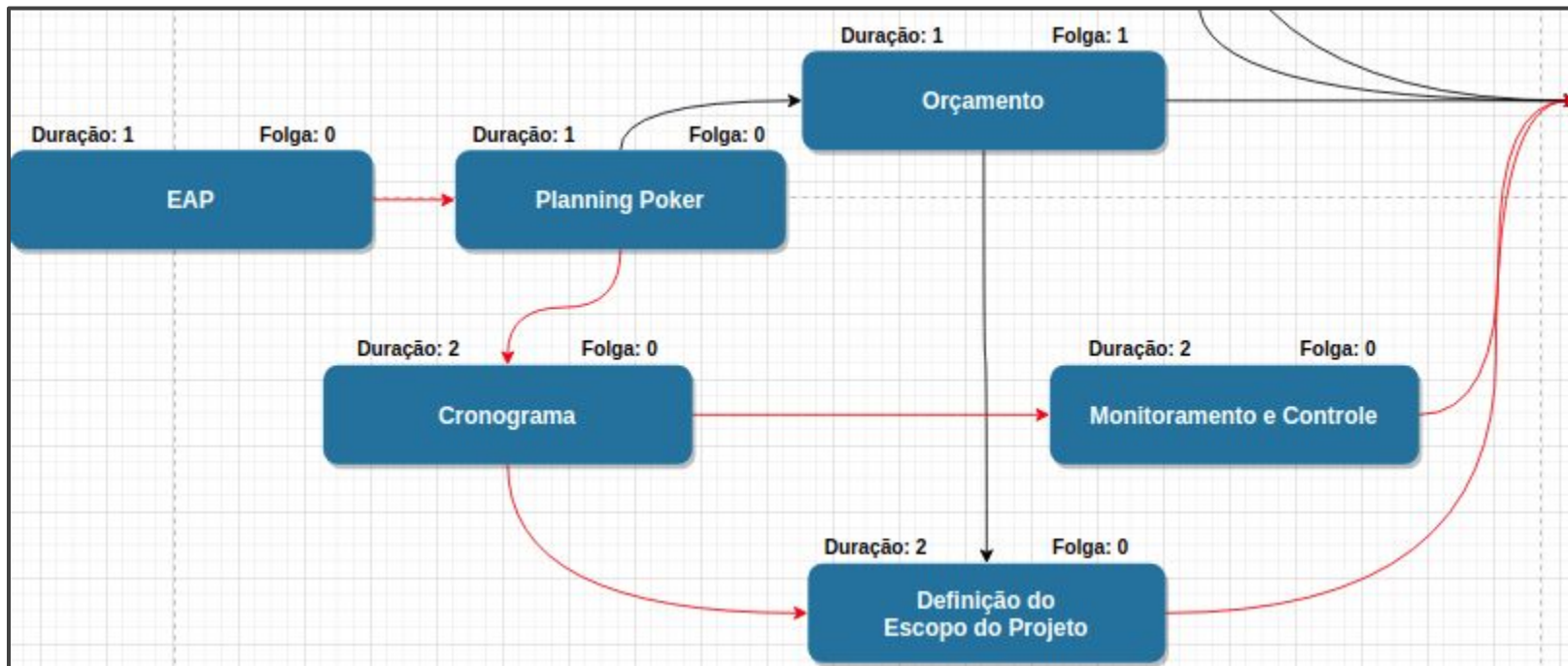
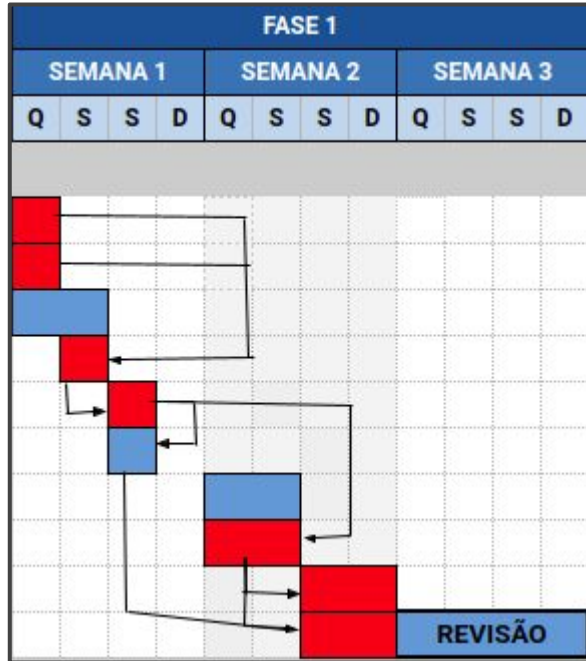




Gráfico de Gantt

TÍTULO DA TAREFA	PROPRIETÁRIO DA TAREFA	DATA DE INÍCIO MÍN	DATA DE INÍCIO MÁX	DATA DE FIM MÍN	DATA DE FIM MÁX	FOLGAS	DURAÇÃO	DEPENDÊNCIAS
Planejamento								
Lista de Requisitos funcionais e não funcionais	José/Juan	01/07/21		01/07/21		0	1	
Casos de Uso	Daniel/André	01/07/21		01/07/21		0	1	
Análise de Riscos	Lucas	01/07/21	10/07/21	02/07/21	11/07/21	5	2	
EAP	Daniel/Juan	02/07/21		02/07/21		0	1	1.1.1 - 1.1.2
Planning Poker	Todos	03/07/21		03/07/21		0	1	1.2.1
Orçamento	José/André	03/07/21	04/07/21	03/07/21	04/07/21	1	1	1.2.2
Diagrama de Classes	Lucas/Juan/Daniel	08/07/21	10/07/21	09/07/21	11/07/21	4	2	
Cronograma	José/André	08/07/21		09/07/21		0	2	1.2.2
Monitoramento e Controle	Lucas/Daniel	10/07/21		11/07/21		0	2	1.4
Descrição do escopo do projeto	José/Juan/André	10/07/21		11/07/21		0	2	1.4 - 1.2.3

Gráfico de Gantt



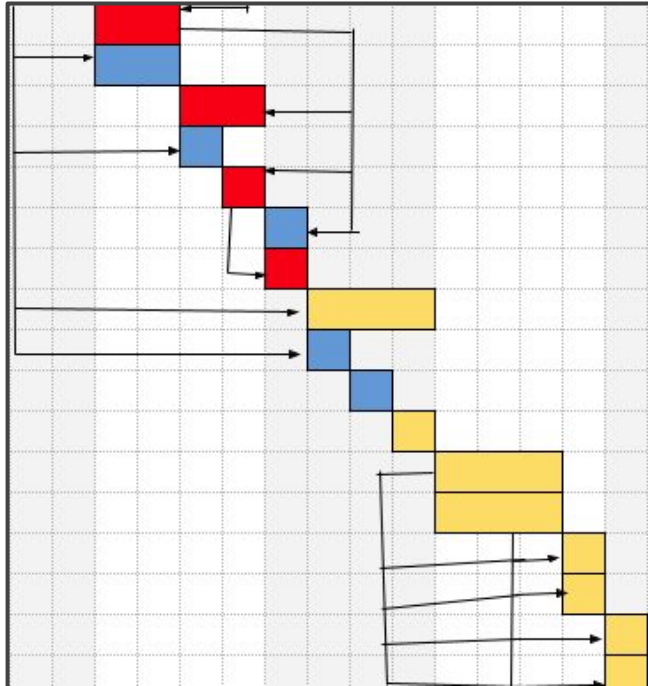
LEGENDA:

AZUL = NÃO CRÍTICO

AMARELO = PSEUDO-CRÍTICO

VERMELHO = CRÍTICO

Gráfico de Gantt



LEGENDA:

AZUL = NÃO CRÍTICO

AMARELO = PSEUDO-CRÍTICO

VERMELHO = CRÍTICO

Análise de riscos





Riscos identificados

- de Processo
 - Cronograma incorreto
 - Ignorância sobre *multiplayer online*.
 - Conflito entre *API* 's.
 - Quantidade insuficiente de desenvolvedores.
- de Produto
 - Manipulação da partida.
 - Problemas de desempenho.



1. Manipulação da partida

- a. Descrição: Possibilidade de manipulação da partida por parte do *host*.
- b. Probabilidade: Média (0,5).
- c. Impacto: Alto (0,9).
- d. Exposição: Alta (0,45).
- e. Plano(s) de contenção:
 - i. Tomar cuidados no desenvolvimento do código, para evitar que usuários mal intencionados possam corrompê-lo.
- f. Plano(s) de contingência:
 - i. Banir a conta do usuário identificado através de um botão de *report*.
- g. Monitoramento:
 - i. Duração: Durante todo o ciclo de vida do jogo, desde o início do projeto até o fim do suporte ao jogo.
 - ii. Revisar possíveis falhas do código durante sua implementação.
 - iii. Analisar jogadores reportados.



2. Cronograma incorreto.

- a. Descrição: Estipulação incorreta do tempo necessário para realização de pelo menos uma das iterações.
- b. Probabilidade: Média (0,5).
- c. Impacto: Alto (0,7). Possibilidade de não entregar todas as funcionalidades prometidas da iteração.
- d. Exposição: Alta (0,35).
- e. Plano(s) de contenção:
 - i. No caso de uma iteração levar menos que o esperado, adiantar o início da próxima iteração.
 - ii. Começar cada iteração o mais cedo possível, para identificar possíveis problemas o quanto antes.
 - iii. Planejamento bem detalhado de todos os requisitos de cada funcionalidade, assim como o passo a passo de sua execução.
- f. Plano(s) de contingência:
 - i. Utilizar de mais horário que o antecipado para concluir corretamente a iteração.
- g. Monitoramento:
 - i. Duração: Durante todo o planejamento e implementação do jogo.
 - ii. Comunicação ativa entre os desenvolvedores para possíveis suspeitas de atraso.



3. Ignorância sobre multiplayer online.

- a. Descrição: Dificuldade na implementação do *multiplayer online* com Firebase.
- b. Probabilidade: Média (0,7).
- c. Impacto: Baixo (0,3). Entrega do projeto sem o *multiplayer online*, apenas o *multiplayer offline*.
- d. Exposição: Média (0,21).
- e. Plano(s) de contenção:
 - i. Aprender o mais cedo possível sobre implementação do *multiplayer online*, antes mesmo de sua implementação ser definida no cronograma.
- f. Plano(s) de contingência:
 - i. Não entregar o programa com *multiplayer online*, apenas o *multiplayer offline*.
- g. Monitoramento:
 - i. Duração: Durante a iteração reservada para implementação do *multiplayer online*.
 - ii. Testes frequentes durante a respectiva iteração para avaliar o funcionamento correto do *multiplayer online*.



4. Quantidade insuficiente de desenvolvedores.

- a. Descrição: Por qualquer motivo, pelo menos um dos desenvolvedores não conseguiu produzir o que foi planejado.
- b. Probabilidade: Baixa (0,2).
- c. Impacto: Alto (0,8).
- d. Exposição: Média (0,16).
- e. Plano(s) de contenção:
 - i. Estabelecimento de uma boa rede de comunicação entre os desenvolvedores.
- f. Plano(s) de contingência:
 - i. Distribuição do trabalho não realizado entre os outros desenvolvedores.
- g. Monitoramento:
 - i. Duração: Durante toda a implementação do jogo.
 - ii. Discussão ativa entre os desenvolvedores para possíveis problemas individuais.



5. Conflito entre API 's.

- a. Descrição: Mais especificamente, durante a implementação do chat, há a possibilidade de conflitos entre PPlay e tkinter.
- b. Probabilidade: Média (0,4).
- c. Impacto: Baixo (0,2).
- d. Exposição: Baixa (0,08).
- e. Plano(s) de contenção:
 - i. Estudar as limitações do PPlay.
- f. Plano(s) de contingência:
 - i. Não implementação do chat.
- g. Monitoramento:
 - i. Duração: Durante a iteração reservada para implementação do *multiplayer online*.
 - ii. Testes frequentes durante a respectiva iteração para avaliar o funcionamento correto do chat.



6. Problemas de desempenho

- a. Descrição: Possibilidade de baixo desempenho do produto final devido a implementação de algumas bibliotecas utilizadas.
- b. Probabilidade: Média (0,4).
- c. Impacto: Médio (0,4).
- d. Exposição: Baixa (0,08).
- e. Plano(s) de contenção:
 - i. Melhor compreender as limitações das bibliotecas utilizadas.
- f. Plano(s) de contingência:
 - i. Atualização com implementação de funções análogas, mas com foco no desempenho.
- g. Monitoramento:
 - i. Duração: Durante toda a implementação do jogo.
 - ii. Realização de testes para averiguar o desempenho.

Detalhamento de uma Sprint





Primeiro Product Backlog

Lista de Requisitos Funcionais e não funcionais

caso de uso

Análise de Risco

EAP

Planning Poker

Orçamento

Cronograma

Diagrama de Classes

Monitoramento Controle

Descrição de escopo do projeto



Fim da Sprint

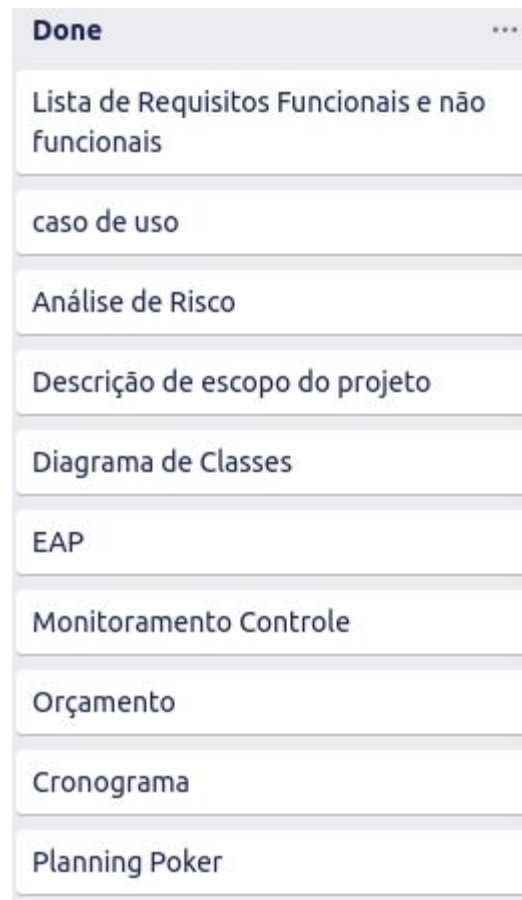


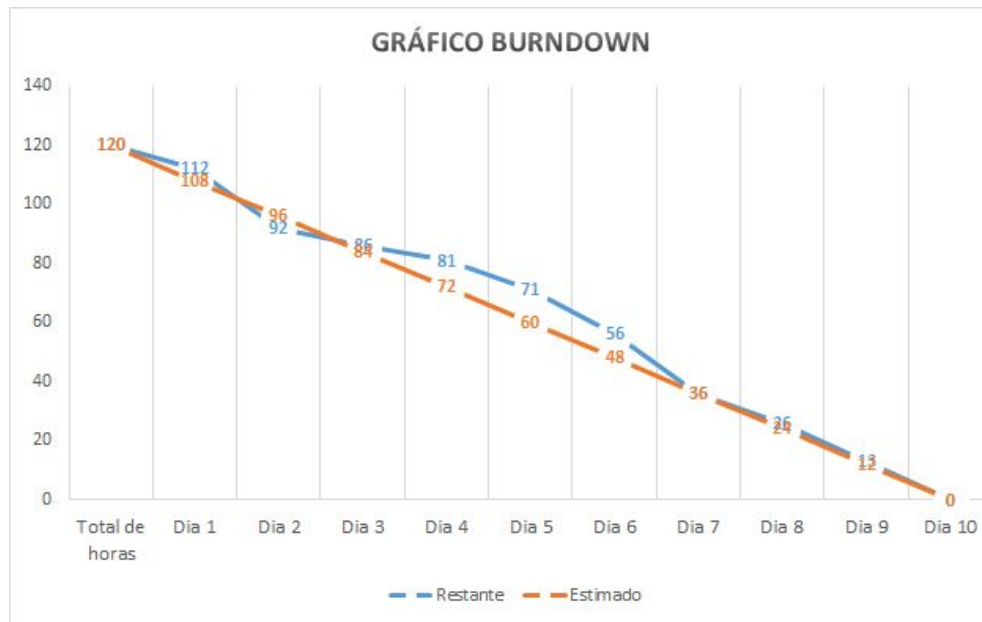
Gráfico de BurnDown





O que é?

- Controlar o andamento da sprint.
- Ter uma comparação com a sprint ideal





Como foi feito?

- Primeiro calculamos o tempo planejado total da sprint e a quantidade de dias
- A medida que as tarefas foram concluídas colocamos o tempo esperado na planilha.
- Além disso criamos outro gráfico que representa a quantidade de tempo sendo reduzida de forma constante



Resultado: Primeira Sprint

Gráfico BurnDown: Primeira Sprint

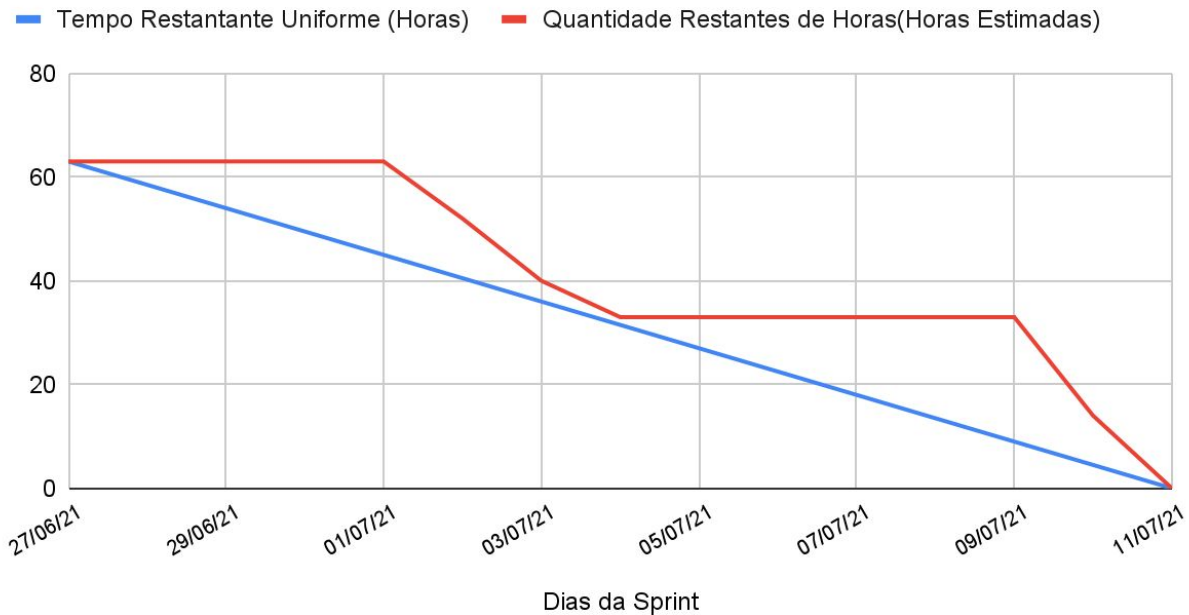


Gráfico de Valor Agregado



O que é?

-Verificar o valor do projeto considerando gastos

-Conseguir uma comparação entre os valores: PV ,EV,AC





Como foi feito:

- Primeiramente, colocamos a distribuição das horas planejadas com o decorrer do tempo, e com essa informação calculamos o preço planejado(PV)
- Conforme terminamos as tarefas, colocamos reduzimos o valor planejados dessa tarefa do total de horas naquele dia, e dessa forma calculamos o valor agregado(EV)
- Por fim, quando terminamos a tarefa, adicionamos no dia a quantidade de horas que realmente foi necessária para ser realizada, dessa forma calculamos o preço real(AC)



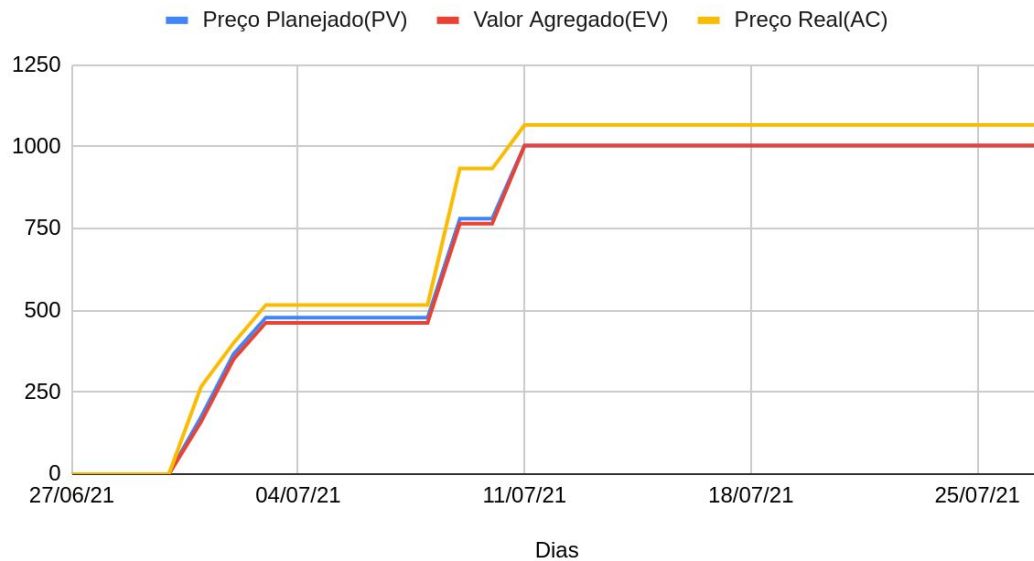
Problemas

- No momento que foi feito o cronograma, muitas tarefas já tinha sido realizadas, então baseamos em fatos reais para incluí-las no cronograma
- Dessa forma, os valores de PV e EV são basicamente os mesmos
- Esse problema deve somente persistir nessa entrega



Resultado

Gráfico de Valor Agregado





Conclusões

O SPI, Schedule Performance Index (indicador de desempenho de cronograma), possui valor de 1, ou seja, estamos no cronograma.

O CPI, Cost Performance Index (indicador de desempenho de custo), possui valor de 0,94, ou seja, o projeto está acima do custo planejado

Casos de uso





Lista de Casos de Uso

- UC01: Configurar jogo.
- UC02: Log In.
- **UC03: Criar saguão.**
- **UC04: Conectar a um saguão.**
- UC05: Recebimento de exército.
- **UC06: Combate.**
- UC07: Movimentação de tropas.
- UC08: Troca de cartas.
- UC09: Finalizar turno.
- UC10: Finalizar jogo.
- UC11: Sair do jogo.
- UC12: Enviar mensagem no chat.
- UC13: Acessar histórico.
- UC14: Ver informações dos jogadores.
- **UC15: Iniciar partida.**



UC03: Criar saguão

Cenário típico:

1. O jogador seleciona a opção de criar um novo saguão.
2. O sistema pede para informar o nome do saguão e uma senha.
3. O jogador fornece o nome do saguão, a senha e o número de jogadores.
4. O sistema valida as informações e cria o saguão fechado, enviando o jogador para a tela do saguão. O jogador que criou o saguão é o host do saguão.

Cenário(s) Alternativo(s):

4.a O jogador não fornece senha.

1. O sistema define a sala como sala aberta.

4.b O jogador fornece um número de jogadores <4 ou >6 .

1. O sistema informa que o número de jogadores é inválido.
2. Retorno ao passo 3 do cenário típico.



UC04: Conectar a um saguão

Cenário típico:

1. O jogador clicar na opção de conectar a um jogo.
2. O sistema pede ao usuário as informações de nome do saguão e senha.
3. O jogador insere as informações.
4. O sistema procura no banco de dados o saguão informado.
5. Se as informações estiverem corretas, o sistema conecta o jogador ao saguão informado.

Cenário(s) alternativo(s):

4.a As informações fornecidas pelo usuário estão incorretas.

1. O sistema informa ao jogador que alguma das informações está incorreta.



UC06: Combate

Cenário típico:

1. O jogador do turno (atacante) escolhe um território adjacente ao seu para atacar.
2. O atacante escolhe uma quantidade de exércitos do seu território atacante para utilizar no combate, sendo no máximo 3 e deixando pelo menos um exército no território atacante.
3. O sistema joga uma quantidade de dados atacantes igual ao número de exércitos atacantes.
4. O sistema joga uma quantidade de dados defensores igual a quantidade de exércitos do jogador (defensor) que possui o território, sendo no máximo 3.
5. O sistema compara os dados jogados em ordem decrescente.
6. Em cada comparação, o sistema remove um exército do jogador que possuir o menor número.
7. O sistema salva a quantidade de exércitos nos territórios atacante e defensor.



UC15: Iniciar a partida

Cenário típico:

1. O jogador clicar no botão de começar.
2. O sistema completará a partida com IAs de acordo com o número de jogadores.
3. O sistema sorteia as cores dos jogadores.
4. O sistema gera o mapa do jogo.
5. O sistema distribui os territórios e exércitos aos jogadores.
6. O sistema distribui os objetivos aos jogadores.
7. O sistema sorteia o jogador inicial.

