**UNIVERSIDADE FEDERAL FLUMINENSE**

ENGENHARIA DE SOFTWARE II

**TICKET TO RIDE**

ADRIAN CODERQUE SANCHEZ

ANTÔNIO ROMANO FERREIRA

ERIC MOINHOS VIANNA

GERALDO FADA

GUILHERME TOMAZ RODRIGUES ALVES

JUAN PABLO MONTEIRO FERNANDES

NITERÓI, RIO DE JANEIRO

2025

**Escopo dos Requisitos**

**Requisitos Funcionais (RF):**

| **Código** | **Descrição** |
| --- | --- |
| **RF001** | O sistema deve renderizar o tabuleiro do jogo. |
| **RF002** | O sistema deve renderizar as cartas de trem. |
| **RF003** | O sistema deve renderizar as peças de trem. |
| **RF004** | O jogo deve permitir de 2 a 5 jogadores, podendo ser uma combinação de jogadores humanos e IA. |
| **RF005** | O jogo deve oferecer rotas duplas somente quando houver mais de três jogadores na partida. |
| **RF006** | O sistema deve ser capaz de determinar o último turno. |
| **RF007** | Se um jogador possui 2 ou menos peças de trem, o restante dos jogadores deve ter a chance de realizar um último turno. |
| **RF008** | O jogador pode comprar até 2 cartas de trem. |
| **RF009** | O jogador pode comprar duas cartas do deck. |
| **RF010** | O jogador pode escolher uma carta virada para cima. |
| **RF011** | Caso o jogador compre uma locomotiva, ele não pode realizar mais ações na rodada. |
| **RF012** | Se o jogador escolher uma carta virada para cima, ele deve substituir essa carta puxando uma nova do deck. |
| **RF013** | O jogador pode utilizar suas cartas de trem para conquistar uma rota. |
| **RF014** | O jogador deve possuir exatamente a mesma quantidade de cartas de trem da mesma cor que a rota. |
| **RF015** | Para rotas de cor cinza, pode-se usar qualquer cor de carta. |
| **RF016** | As cartas locomotivas podem ser usadas como substitutas para qualquer carta de trem. |
| **RF017** | O jogador deve comprar 3 cartas de destino. |
| **RF018** | O jogador deve escolher pelo menos uma carta e devolver as outras ao final do baralho. |
| **RF019** | O jogo deve embaralhar novamente as cartas viradas para cima se 3 locomotivas aparecem viradas para cima. |
| **RF020** | O sistema deve gerenciar automaticamente a sequência de turnos entre os jogadores. |
| **RF021** | O sistema deve ser capaz de calcular a pontuação de cada jogador ao final de cada rodada e ao final da partida. |
| **RF022** | O sistema deve registrar e mostrar as ações realizadas por cada jogador de forma clara e acessível. |
| **RF023** | A interface deve ser intuitiva e fornecer feedback adequado ao jogador após cada ação. |

**Requisitos Não Funcionais (RNF):**

| **Código** | **Categoria** | **Descrição** |
| --- | --- | --- |
| **CF01** | Confiabilidade | O sistema de jogo deve funcionar nas seguintes plataformas: Windows. |
| **CF02** | Confiabilidade | O sistema de jogo deve garantir que a ordem de jogadas seja registrada corretamente. |
| **EF01** | Eficiência | As ações de um jogador devem ser refletidas em até 5 segundos no sistema. |
| **US01** | Usabilidade | O sistema de jogo deve oferecer suporte aos idiomas português e inglês. |
| **US02** | Usabilidade | O sistema deve oferecer uma interface intuitiva para o jogador. |
| **US03** | Usabilidade | O sistema de jogo deve oferecer acesso às regras por meio de uma documentação. |
| **ES01** | Escalabilidade | O sistema deve permitir um aumento do número de jogadores até o máximo permitido. |
| **MN01** | Manutenibilidade | O sistema de jogo e seus componentes devem ser organizados de forma a permitir fácil manutenção. |

**Regras de negócio:**

1. **Número de Jogadores**
   * O jogo deve permitir **no mínimo 2** e **no máximo 5** jogadores, podendo incluir uma combinação de jogadores humanos e IA.
2. **Rotas Duplas**
   * O jogo só deve oferecer **rotas duplas** quando houver **mais de três jogadores** na partida.
3. **Determinação do Último Turno**
   * O jogo deve ser capaz de determinar **quando é o último turno**.
     + Se um dos jogadores possui **2 ou menos peças de trem**, então o restante dos jogadores deve fazer um **último turno**.
4. **Ações do Jogador**
   * Durante o turno, o jogador pode realizar **apenas uma** das seguintes ações:

4.1. **Comprar Cartas de Trem**

* + O jogador pode comprar até **2 cartas de trem**.
    - O jogador pode:
      * Comprar **duas cartas do deck**.
      * Escolher **uma carta virada para cima**.
        + Se o jogador comprar uma **locomotiva (carta coringa)**, ele não pode realizar mais ações nesse turno.
        + Se o jogador escolher uma carta virada para cima, ele deve substituir essa carta puxando uma nova do deck.
        + Após substituir, ele deve comprar uma carta virada para cima ou do baralho.

4.2. **Conquistar uma Rota**

* + O jogador pode usar suas cartas de trem para **conquistar uma rota**.
    - O jogador deve possuir exatamente a **mesma quantidade de cartas de trem** e da **mesma cor** da rota que deseja conquistar.
      * Para rotas de **cor cinza**, pode-se usar **qualquer cor** de carta.
      * Uma **carta locomotiva (carta coringa)** pode substituir **qualquer carta de trem**.

4.3. **Comprar Cartas de Destino**

* + O jogador deve comprar **3 cartas de destino**.
    - O jogador deve escolher **pelo menos uma** das cartas compradas; as demais devem ser devolvidas ao final do baralho.

1. **Embaralhamento de Cartas**
   * O jogo deve **embaralhar novamente** as cartas viradas para cima caso apareçam **3 locomotivas (cartas coringas)** viradas para cima.

**EAP**

**Link:** [**https://miro.com/app/board/uXjVI\_THebk=/**](https://miro.com/app/board/uXjVI_THebk=/)

**Estimativas de esforço**

**Planning Poker**

Os valores finais se encontram no repositório do projeto

**APF**

**1. Entradas Externas (EE) – Total: 7**

1.1 Configuração inicial da partida (seleção de jogadores) - RF004

jogador seleciona o número de jogadores (2 a 5) e define se são humanos ou IA no menu principal. Isso é uma entrada única que configura a partida.

**Entidades referenciadas**:2 (jogadores, estado do jogo

**Campos de dados**: 2 (1 para número de jogadores, 1 para tipo)

**complexidade:** Baixa

1.2 Seleção de cartas de trem - RF008, RF010

O jogador insere a escolha de comprar até 2 cartas (viradas para cima ou do deck), representando uma entrada distinta.

**Entidades referenciadas**:3 (cartas de trem, Jogadores, estado do jogo)

**Campos de dados**: 3 (Escolha da carta, Tipo de carta, Quantidade de cartas)

**complexidade**: Média

1.3 Conquista de rotas - RF013, RF014

*O jogador seleciona uma rota e as cartas de trem correspondentes para conquistá-la, fornecendo uma entrada específica.*

**Entidades referenciadas**: 4(Rotas, Cartas de Trem, Jogadores, Estado do Jogo)

**Campos de dados**: 4 (Rota selecionada/ID da rota, cor das cartas, qtd de cartas, peças de trem usadas)

**complexidade**: Média

1.4 Compra de cartas de destino - RF017, RF018

O jogador insere a escolha de pelo menos uma carta de destino após comprar 3, representando uma entrada única.

**Entidades referenciadas:** 3 (Cartas de Destino, Jogadores, Estado do Jogo)

**Campos de dados**: 2( Cartas selecionadas, IDs das cartas de destino)

**complexidade:** Média

1.5 Ações do jogador humano no turno - EAP 4.1.1 jogada do jogador humano

Durante o turno, o jogador humano insere ações (comprar cartas, conquistar rotas ou comprar bilhetes), considerada uma entrada geral para o gerenciamento do turno.

**Entidades referenciadas**: 2 (Estado do Jogo, Jogadores)

**Campos de dados**: 1(Tipo de ação ex:comprar carta, conquistar rota)

**complexidade**: Baixa

1.6 Interação com interface (HUD) - RF023, EAP 4.4.1

O jogador interage com botões e elementos do HUD (ex.: clicar para confirmar ações), representando entradas específicas da interface.

prar bilhetes), considerada uma entrada geral para o gerenciamento do turno.

**Entidades referenciadas**: 2 (Estado do Jogo, Jogadores)

**Campos de dados**: 1(Ação do botão ex;confirmar, cancelar)

**complexidade:** Baixa

1.7 Seleção de idioma US001

o suporte a idiomas implica uma entrada onde o jogador seleciona português ou inglês, provavelmente no menu principal.

**Entidades referenciadas**: 1 (Estado do Jogo)

**Campos de dados**: 1(idioma escolhido)

**complexidade:** Baixa

**2. Saídas Externas (SE) – Total: 5**

2.1 Renderização do tabuleiro - EAP 4.4.2

O sistema gera a visualização do tabuleiro (cidades, rotas, peças de trem), que é uma saída única apresentada ao jogador.

**Entidades Referenciadas**: 2 (Rotas, Estado do Jogo)

**Campos de Dados**: 3 (cidades, rotas, peças de trem visíveis)

**Complexidade**: Baixa

2.2 Renderização das cartas de trem -RF002, EAP 4.4.1

As cartas de trem (8 cores + locomotivas) são exibidas na interface, representando uma saída distinta.

**Entidades Referenciadas**: 1 (Cartas de Trem)

**Campos de Dados**: 2 (cor, tipo de carta)

**Complexidade**: Baixa

2.3 Renderização das peças de trem EAP 4.4.1, RF003

As peças de trem colocadas nas rotas são exibidas, considerada uma saída separada.

**Entidades Referenciadas**: 2 (Jogadores, Rotas)

**Campos de Dados**: 2 (cor do jogador, posição na rota)

**Complexidade**: Baixa

2.4 Tela de fim de jogo (placar) RF021

O sistema gera a tela final com a pontuação de cada jogador, incluindo pontos de rotas, bilhetes e bônus, como uma saída única.

**Entidades Referenciadas**: 2 (Jogadores, Estado do Jogo)

**Campos de Dados**: 5 (pontuação por jogador, rotas, bilhetes, bônus, total)

**Complexidade**: Baixa

2.5 Registro de ações dos jogadores RF022

As saídas foram identificadas com base em elementos visuais gerados pelo sistema (telas, objetos renderizados) e informações apresentadas ao jogador. Cada saída é única e corresponde a um componente específico da EAP ou requisito.

**Entidades Referenciadas**: 3 (Jogadores, Estado do Jogo, Rotas)

**Campos de Dados**: 4 (jogador, ação, rota afetada, turno)

**Complexidade**: Baixa

**3. Consultas Externas (CE) – Total: 4**

3.1 Validação de compra de cartas - RF008, RF010, RF011

O jogador insere a escolha de uma carta (entrada), e o sistema valida a ação (ex.: se é uma locomotiva, bloqueia mais ações) e atualiza a interface (saída), como substituir a carta virada.

**Entidades Referenciadas**: 3 (Cartas de Trem, Jogadores, Estado do Jogo)

**Campos de Dados**: 3 (escolha da carta, tipo, quantidade)

**Complexidade**: Baixa

3.2 Validação de conquista de rotas - RF013, RF014, RF015, RF016 (Regras de rotas e cartas).

O jogador seleciona uma rota e cartas (entrada), e o sistema verifica se a ação é válida (ex.: cor e quantidade corretas) e atualiza o tabuleiro/pontuação (saída).

**Entidades Referenciadas**: 4 (Rotas, Cartas de Trem, Jogadores, Estado do Jogo)

**Campos de Dados**: 4 (rota, cor das cartas, quantidade, peças usadas)

**Complexidade**: Média

3.3 Verificação do fim de jogo - RF006 (O sistema deve ser capaz de determinar o último turno), RF007 (Último turno com 2 ou menos peças).

O sistema detecta que um jogador tem 2 ou menos peças (entrada interna) e gera a notificação do último turno e a tela final (saída).

**Entidades Referenciadas**: 2 (Jogadores, Estado do Jogo)

**Campos de Dados**: 2 (peças restantes, turno atual)

**Complexidade**: Baixa

3.4 Cálculo de pontuação em tempo real -RF021 (Calcular pontuação ao final de cada rodada).

Após uma ação (entrada, ex.: conquistar rota), o sistema calcula e exibe a pontuação atualizada do jogador (saída), visível na interface.

**Entidades Referenciadas**: 3 (Jogadores, Rotas, Estado do Jogo)

**Campos de Dados**: 3 (pontos por rota, total atual, jogador)

**Complexidade**: Baixa

**4. Arquivos Lógicos Internos (ALI) – Total: 5**

4.1 Jogadores -RF004 (2 a 5 jogadores, humanos ou IA).

Entidade que armazena informações de cada jogador (nome, cartas, peças, pontos, bilhetes), manipulada durante o jogo.

**Agrupamentos**: 1 (um grupo por jogador)

**Campos de Dados**: 5 (nome, cor, cartas, peças, pontos)

**Complexidade**: Baixa

4.2 Cartas de trem -RF002, RF008–RF012 (Regras de cartas).

Entidade que gerencia o baralho de cartas de trem (cores, locomotivas, viradas, descarte), essencial para as mecânicas.

**Agrupamentos**: 2 (baralho, descarte)

**Campos de Dados**: 3 (cor, tipo, quantidade)

**Complexidade**: Baixa

4.3 Cartas de destino -RF017, RF018 (Regras de bilhetes).

Entidade que armazena bilhetes de destino (rotas, pontos), gerenciada separadamente das cartas de trem.

**Agrupamentos**: 1 (baralho de bilhetes)

**Campos de Dados**: 4 (cidade inicial, cidade final, pontos, status)

**Complexidade**: Baixa

4.4 Rotas RF013–RF016 (Regras de rotas).

Entidade que representa as rotas do tabuleiro (cidades conectadas, cor, tamanho, dono), manipulada ao conquistar rotas.

**Agrupamentos**: 1 (rotas do tabuleiro)

**Campos de Dados**: 5 (cidade inicial, cidade final, cor, tamanho, dono)

**Complexidade**: Baixa

4.5 Estado do jogo RF020 (Gerenciar sequência de turnos).

Entidade que armazena o estado global do jogo (turno atual, pontuações, cartas disponíveis), usada para sincronizar a partida.

**Agrupamentos**: 1 (estado global)

**Campos de Dados**: 4 (turno atual, jogadores ativos, cartas disponíveis, status)

**Complexidade**: Baixa

**5. Arquivos de Interface Externos (AIE) – Total: 0**

Arquivos de interface externos são entidades compartilhadas com sistemas externos. Não identifiquei nenhum AIE com base na EAP ou nos requisitos.

* **Justificativa**: Se trata de um jogo standalone para Windows (RF001), sem a integração com sistemas externos. Não há compartilhamento de dados com outros sistemas), então o total de AIE é zero.

**RESUMO**

**Contagem sem pesos:**

**Entradas Externas (EE)**: 7 (4x Baixas + 3x Médias)

**Saídas Externas (SE)**: 5 (5x Baixas)

**Consultas Externas (CE)**: **4** (3x Baixas + 1x Média)

**Arquivos Lógicos Internos (ALI)**: 5 (5x Baixas)

**Arquivos de Interface Externos (AIE)**: 0

**Cálculo do PFNA:**

**Entradas Externas (EE)**: 4x Baixas \* 3 + 3x Médias \* 4 = 12 + 12 = 24

**Saídas Externas (SE)**: 5x Baixas \* 4 = 20

**Consultas Externas (CE)**: 3x Baixas \* 3 + 1x Média \* 4 = 13

**Arquivos Lógicos Internos (ALI)**: 5x Baixas \* 7 = 35

**Arquivos de Interface Externos (AIE)**: 0

**TOTAL PFNA: 92**

**Ajuste dos pontos de função (Menor resposta: 0 Maior resposta:5)**

1. Necessita de backup? 1

2. Necessita de mecanismos especializados de comunicação? 0

3. Tem processamento distribuído? 0

4. Precisa de alto desempenho? 2

5. Terá grande número de usuários em paralelo? 0

6. Precisará de entrada de dados on-line? 0

7. No caso de entradas on-line, existirão múltiplas telas? 0

8. A atualização das entidades será feita on-line? 0

9. As entradas e saídas de dados serão complexas? 2

10. O processamento interno será complexo? 2

11. O código será projetado para ser reutilizado? 1

12. Migração e instalação estarão incluídos? 1

13. O sistema será instalado em diversas organizações? 0

14. O projeto pretende facilitar mudanças e operação do usuário? 2

**Somatório das 14 respostas**:11

**PF: 92 \* (0,65 + 0,01 \* 11) = 69,92 ≅ 70**

***Consideramos 1 homem-mês = 6 PF, então o esforço será de 11 homem-mês.***

**Definição do Cronograma**

**Diagrama de Gantt**

**Link:** [**https://github.com/users/Romanoferr/projects/2/views/4**](https://github.com/users/Romanoferr/projects/2/views/4)

**Caminho Crítico**

**Link (Miro):** [**https://miro.com/app/board/uXjVI\_THebk=/**](https://miro.com/app/board/uXjVI_THebk=/)

**Link (Planilha de Excel):**

[**https://docs.google.com/spreadsheets/d/1dLeDDBL6s\_6OLInnpqFOQh3ocvnWx-5Xke2CS3oikxU/edit?gid=1404283177#gid=1404283177**](https://docs.google.com/spreadsheets/d/1dLeDDBL6s_6OLInnpqFOQh3ocvnWx-5Xke2CS3oikxU/edit?gid=1404283177#gid=1404283177)

**Custo e Orçamento**

**Recursos Humanos**

* Salário de por hora do gerente
  + Baseado no salário médio de um gerente engenharia de software, com uma carga de 40 horas semanais: R$ 16000,00
  + R$ 100,00 por hora
* Salário por hora do desenvolvedor
  + Baseado no salário médio de um engenheiro de software pleno, com uma carga horária de 40 horas semanais : R$ 5000,00
  + R$ 31,25 por hora
* Custo total do projeto:
  + 164,8 homem-hora \* (R$ 31,25 + R$ 100,00) = R$ 21630,00

**Recursos de Capital**

* Notebook:
  + Custo unitário: R$ 5000,00
  + Custo por utilização: R$ 5000,00 \* (1680 duração do projeto em horas /43800 vida útil em horas) = R$ 191,78

**Recursos de Consumo**

* Nenhum presente

**Lucro**

* Margem de lucro: 20%
* Valor do lucro: (R$ 21630,00 + R$ 191,78 \* 5)\*0,20 = R$ 4517,78

**Orçamento**

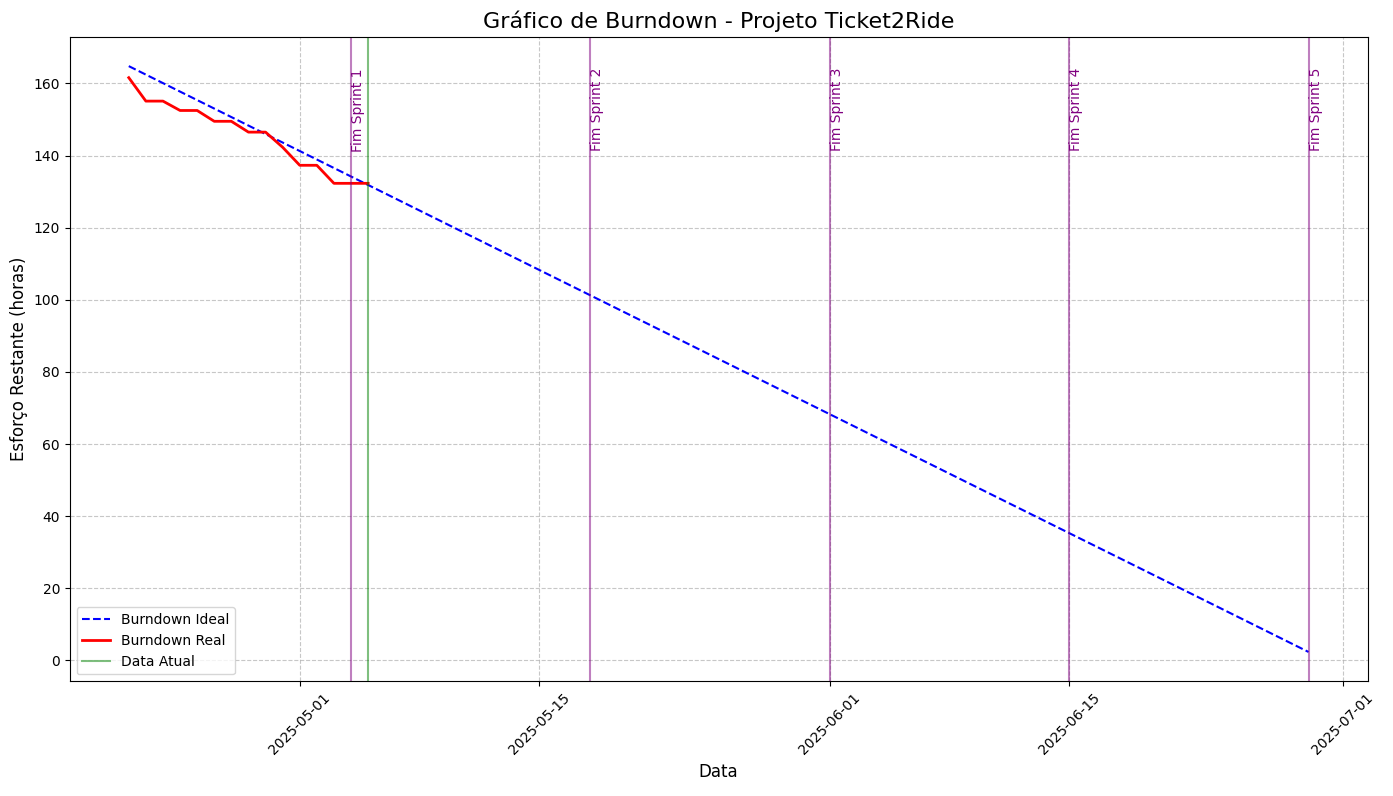
* Custo + lucro: R$ 21630,00 + R$ 4517,78 = R$ 26147,78
* Divisão em três parcelas, baseado no número de apresentações:
  + R$ 26147,78 / 3 = R$ 8715,93

**Monitoramento e Controle**

**Organização e planos:**

[**Documento com o planejamento das nossas sprints**](https://docs.google.com/document/d/1JzLiFPppFAzL5QwUtR_M1njFtSY4Es1ud0Z8mZJys04/edit?usp=sharing)

**Burndown**

****

**Análise de Valor Agregado**

[**Analise de Valor Agregado**](https://docs.google.com/spreadsheets/d/1NWqUsq1A_f3JYcfT3wMXDyEI6puRi4AYg2hsbnfPh9g/edit?usp=sharing)

**SPI < 1** indica que estamos atrasados em relação ao nosso cronograma.

**CPI < 1** mostra que estamos acima do custo planejado.

**Análise de Risco**

**Link:**

[**https://docs.google.com/spreadsheets/d/1KxTKiT1lv2ERuZzOJl3jvRo-hdHa32VW79tUrVFTTEE/edit?gid=1908939778#gid=1908939778**](https://docs.google.com/spreadsheets/d/1KxTKiT1lv2ERuZzOJl3jvRo-hdHa32VW79tUrVFTTEE/edit?gid=1908939778#gid=1908939778)

**Esboços**

**Menu principal**

[**https://excalidraw.com/#json=1YoNzftHNrSsAS1-gY1QZ,nKPzY2d0J6orR\_kXBf89Lw**](https://excalidraw.com/#json=1YoNzftHNrSsAS1-gY1QZ,nKPzY2d0J6orR_kXBf89Lw)