
ESPECIFICAÇÃO DE REQUISITOS DE SOFTWARE

para

CARTAS DE AMOR

Versão 1.0

Preparado por:

1. Lucas Cardoso dos Santos (9865492)
2. Maya Monteiro Lima (13671942)
3. Miguel de Carvalho Oliveira (13672172)

Submetido a:

Prof. Dr. Clever Ricardo Guareis de Farias
Prof. Dr. Evandro Eduardo Seron Ruiz

23 de maio de 2025

Sumário

1. Introdução	3
1.1. Objetivo	3
1.2. Público-alvo e Sugestões de Leitura	3
1.3. Escopo do Projeto	3
2. Descrição Geral	4
2.1. Perspectiva do Produto	4
2.2. Classes e Características dos Usuários	4
2.3. Funções do Produto	4
2.4. Ambiente Operacional	5
2.5. Design	5
3. Funcionalidades do Sistema	6
3.1. Descrição e Prioridade	6
3.2. Requisitos Funcionais	6
4. Outros Requisitos Não Funcionais	8
4.1. Requisitos de Desempenho	8
4.2. Requisitos de Segurança	8
4.3. Atributos de Qualidade de Software	8
4.4. Regras de Negócio	8
5. Outros Requisitos	9
A. Regras do Jogo Love Letter	10
A.1. Visão Geral	10
A.2. Fluxo do Jogo	10
A.3. Lista de Cartas e Efeitos	10
A.4. Como Vencer o Jogo	11

1. Introdução

1.1. Objetivo

O objetivo deste documento é especificar os requisitos para o desenvolvimento de uma versão web multiplayer do jogo de cartas *Love Letter*. A aplicação permitirá que usuários se registrem, criem e entrem em salas de jogo, e joguem partidas em tempo real. Além disso, o sistema fornecerá estatísticas de usuários e rankings, promovendo engajamento e competição.

1.2. Público-alvo e Sugestões de Leitura

Este documento é destinado aos desenvolvedores do projeto, instrutores e testadores. Os desenvolvedores devem usar esta especificação para guiar a implementação do front-end, back-end e lógica de comunicação. Os instrutores a utilizarão para avaliar a conformidade do sistema com os requisitos do curso. Os testadores podem consultar este documento para validar as funcionalidades e aspectos não funcionais do sistema.

1.3. Escopo do Projeto

O sistema consistirá em uma interface gráfica baseada em navegador para o jogo de cartas *Love Letter*, um servidor back-end responsável pelo gerenciamento da lógica do jogo e armazenamento persistente de dados, e uma camada de comunicação usando WebSockets para funcionalidade multiplayer em tempo real. A API RESTful também dará suporte ao gerenciamento de contas, histórico de partidas e recursos de ranking.

2. Descrição Geral

2.1. Perspectiva do Produto

O sistema é uma aplicação web independente desenvolvida do zero. Segue uma arquitetura multicamadas com clara separação de responsabilidades:

- **Camada de Apresentação:** Interface web para interação com o jogo e a plataforma.
- **Camada de Lógica:** Servidor back-end responsável pelas regras do jogo, gerenciamento de usuários e controle de sessões.
- **Camada de Dados:** Armazenamento persistente de dados de usuários, estatísticas de jogos e rankings.

2.2. Classes e Características dos Usuários

O sistema prevê basicamente duas classes de usuários:

- **Jogadores**
 - **Jogador Registrado:** Pode acessar todas as funcionalidades do jogo, participar de partidas, consultar histórico e estatísticas, aparecer nos rankings e personalizar seu perfil.
 - **Visitante:** Não possui conta e, ao acessar o sistema sem estar logado, tem acesso apenas à tela de login. Não pode visualizar informações do jogo, rankings, nem navegar por outras áreas do sistema.

2.3. Funções do Produto

O sistema fornecerá as seguintes funcionalidades principais:

- Registro e login de contas de usuário.
- Criação e entrada em salas de jogo.
- Jogabilidade multiplayer em tempo real usando WebSockets.
- Aplicação das regras oficiais do jogo *Love Letter*.
- Registro do histórico de partidas.
- Visualização de estatísticas dos jogadores e ranking global.

2.4. Ambiente Operacional

- **Cliente:** Qualquer navegador web moderno (Chrome, Firefox, Edge, etc.).
- **Servidor:** Ambiente de execução .NET, API RESTful (ASP.NET Core), servidor WebSocket.
- **Banco de Dados:** PostgreSQL.

2.5. Design

A aplicação seguirá uma abordagem de design responsivo para acessibilidade em desktops e dispositivos móveis. O estado do jogo e atualizações serão gerenciados via WebSockets para interação em tempo real e baixa latência. A API RESTful gerenciará dados de usuários, autenticação e funcionalidades não em tempo real.

3. Funcionalidades do Sistema

3.1. Descrição e Prioridade

Todas as funcionalidades principais da aplicação são essenciais. A implementação da jogabilidade e comunicação em tempo real são prioridades máximas. Gerenciamento de contas, estatísticas e rankings são secundários, mas necessários para a completude.

3.2. Requisitos Funcionais

- **RF01:** O sistema deve permitir que usuários se registrem com nome de usuário, e-mail e senha.
- **RF02:** O sistema deve permitir que usuários façam login e mantenham sessões autenticadas.
- **RF03:** O sistema deve permitir que usuários criem novas salas de jogo.
- **RF04:** O sistema deve exibir salas de jogo disponíveis para entrada.
- **RF05:** Uma sala de jogo deve suportar até quatro jogadores.
- **RF06:** O jogo deve iniciar quando todos os jogadores de uma sala confirmarem prontidão.
- **RF07:** O estado do jogo deve ser sincronizado entre todos os jogadores via Web-Sockets.
- **RF08:** O sistema deve aplicar as regras oficiais do *Love Letter* durante o jogo. Um resumo das regras principais está abaixo, e a especificação completa dos efeitos das cartas e regras de interação está no Apêndice [A](#).
 - O jogo é jogado com 2 a 6 jogadores.
 - Cada jogador começa com uma carta na mão.
 - Em cada turno, um jogador compra uma carta e joga uma das duas cartas da mão.
 - Cada carta tem um efeito aplicado imediatamente (ex: adivinhar a carta de outro jogador, comparar mãos, proteção, etc.).
 - Jogadores são eliminados se forem alvo de efeitos específicos de cartas ou se tiverem a carta de menor valor em caso de empate.

- A rodada termina quando o baralho acaba ou resta apenas um jogador.
 - O jogador com a carta de maior valor ao final da rodada ganha um token de afeição.
 - O primeiro jogador a ganhar um número definido de tokens (ex: 3) vence o jogo.
- **RF09:** O sistema deve registrar os resultados das partidas concluídas no banco de dados.
- **RF10:** O sistema deve fornecer a cada usuário acesso ao seu histórico de partidas e estatísticas.
- **RF11:** O sistema deve fornecer um ranking global baseado em vitórias em partidas.

4. Outros Requisitos Não Funcionais

4.1. Requisitos de Desempenho

- A comunicação em tempo real durante as partidas deve ter latência inferior a 1s.
- O sistema deve suportar pelo menos 10 usuários simultâneos sem degradação.

4.2. Requisitos de Segurança

- Senhas de usuários devem ser armazenadas usando hash seguro.
- Todos os dados transmitidos pela rede devem ser criptografados (HTTPS/WSS).
- Sessões de usuários devem expirar após inatividade e suportar tokens de login seguros.

4.3. Atributos de Qualidade de Software

- **Usabilidade:** A interface deve ser intuitiva, responsiva e informativa.
- **Confiabilidade:** A sincronização do estado do jogo deve ser testada quanto à consistência.
- **Manutenibilidade:** O código seguirá princípios de design modular com comentários e documentação.
- **Portabilidade:** A aplicação deve funcionar nos principais navegadores e sistemas operacionais.

4.4. Regras de Negócio

- Apenas usuários registrados podem participar de jogos.
- Um jogador não pode entrar em múltiplas salas de jogo simultaneamente.
- O ranking será atualizado apenas com partidas concluídas.

5. Outros Requisitos

Nenhum requisito adicional foi identificado nesta etapa. Melhorias futuras podem incluir modo espectador ou suporte a outras variantes do jogo.

A. Regras do Jogo Love Letter

Este apêndice descreve o conjunto completo de regras do jogo de cartas *Love Letter*, conforme implementado pelo sistema.

A.1. Visão Geral

O jogo é jogado com 2 a 6 jogadores. O objetivo é vencer um certo número de rodadas (tokens de afeição) sendo o último jogador restante ou segurando a carta de maior valor ao final de uma rodada.

A.2. Fluxo do Jogo

- Cada jogador começa com uma carta.
- Em seu turno, o jogador compra uma carta e escolhe uma das duas cartas para jogar.
- O efeito da carta jogada é aplicado imediatamente.
- Jogadores podem ser eliminados durante a rodada com base nos efeitos das cartas.
- A rodada termina quando resta apenas um jogador ou quando o baralho acaba.
- O jogador com a carta de maior valor vence a rodada.

A.3. Lista de Cartas e Efeitos

- **Spy (0):** Ao final da rodada, se o jogador for o *único* restante que jogou ou descartou um Spy durante aquela rodada, recebe 1 token de afeição, mesmo que não tenha vencido a rodada.
- **Guard (1):** Adivinhe a carta na mão de outro jogador (exceto Guard). Se acertar, o jogador é eliminado.
- **Priest (2):** Veja a carta na mão de outro jogador.
- **Baron (3):** Compare a mão com outro jogador; quem tiver a carta de menor valor é eliminado.
- **Handmaid (4):** Proteção contra todos os efeitos até seu próximo turno.

- **Prince (5):** Escolha qualquer jogador (inclusive você) para descartar a mão (revelando a carta, mas sem aplicar seus efeitos) e comprar uma nova carta.
- **Chancellor (6):** Compre 2 cartas e escolha apenas uma das três cartas na mão para manter. As outras duas cartas retornam ao final do baralho.
- **King (7):** Troque de mão com outro jogador.
- **Countess (8):** Deve ser jogada se estiver na mão junto com Prince ou King.
- **Princess (9):** Se descartada ou jogada, o jogador é eliminado.

A.4. Como Vencer o Jogo

Um jogador ganha um token de afeição a cada rodada vencida. O número de tokens necessários para vencer depende do número de jogadores, conforme a tabela abaixo:

Jogadores	2	3	4	5	6
Tokens para Vencer	6	5	4	3	3

Por exemplo, em um jogo para 2 jogadores, o primeiro a conquistar 6 tokens vence a partida. Em jogos com 5 ou 6 jogadores, apenas 3 tokens são necessários.