

Faculdade Engenheiro Salvador Arena

André Mendes Garcia – 081230012

Pedro Henrique Simões Reys – 081230022

Vinícius Yamaguti Augusto – 081220040

ARENINHA

Documento de Projeto de Software

São Bernardo do Campo

2025

Seção 1: Visão e Escopo do Projeto

1.1 Documento de Visão do Projeto

Descrição do Problema

A falta de plataformas de jogos educativos que combinem entretenimento com desenvolvimento de habilidades cognitivas, oferecendo uma experiência gamificada com ranking competitivo e progressão individual.

Público-Alvo

- **Estudantes:** Jovens e adultos em processo de aprendizagem
- **Profissionais:** Pessoas que buscam desenvolver habilidades mentais
- **Educadores:** Professores que desejam usar jogos como ferramenta pedagógica

Objetivos de Negócio

- **Engajamento:** Manter usuários ativos através de sistema de pontuação e ranking
- **Educação:** Desenvolver habilidades cognitivas de forma lúdica
- **Competição Saudável:** Promover competição através de rankings
- **Acessibilidade:** Oferecer plataforma web gratuita e fácil de usar

Levantamento de Requisitos

Requisitos Funcionais:

1. Sistema de autenticação de usuários
2. Cadastro e gerenciamento de perfil
3. Múltiplos jogos educativos
4. Sistema de pontuação e ranking
5. Histórico de partidas
6. Interface responsiva
7. Administração de usuários

Requisitos Não Funcionais:

1. Segurança: Autenticação e autorização
2. Performance: Tempo de resposta < 2 segundos
3. Usabilidade: Interface intuitiva

4. Confiabilidade: Disponibilidade 99%
5. Escalabilidade: Suporte a múltiplos usuários

1.2 Priorização Estratégica do Backlog

Técnica MoSCoW Aplicada

Must Have:

- Sistema de login/cadastro
- 3 jogos funcionais (Adivinhação, Acertar Palavra, Batalha Naval)
- Sistema de pontuação
- Ranking básico

Should Have:

- Histórico de partidas
- Edição de perfil
- Interface responsiva

Could Have:

- Mais jogos
- Sistema de conquistas
- Compartilhamento de resultados

Won't Have:

- Versão mobile nativa
- Integração com redes sociais
- Modo multiplayer em tempo real

Matriz de Impacto x Esforço

Alto Impacto/Baixo Esforço:

- Sistema de autenticação
- Jogo de adivinhação
- Ranking simples

Alto Impacto/Alto Esforço:

- Múltiplos jogos
- Sistema de pontuação integrado

Baixo Impacto/Baixo Esforço:

- Páginas estáticas
- Logout

Baixo Impacto/Alto Esforço:

- Modo multiplayer
- Integrações externas

MVP (Minimum Viable Product)

- Autenticação de usuários
- 3 jogos funcionais
- Sistema de pontuação
- Ranking básico
- Interface web responsiva

Roadmap de Releases

Release 1.0: MVP com funcionalidades básicas

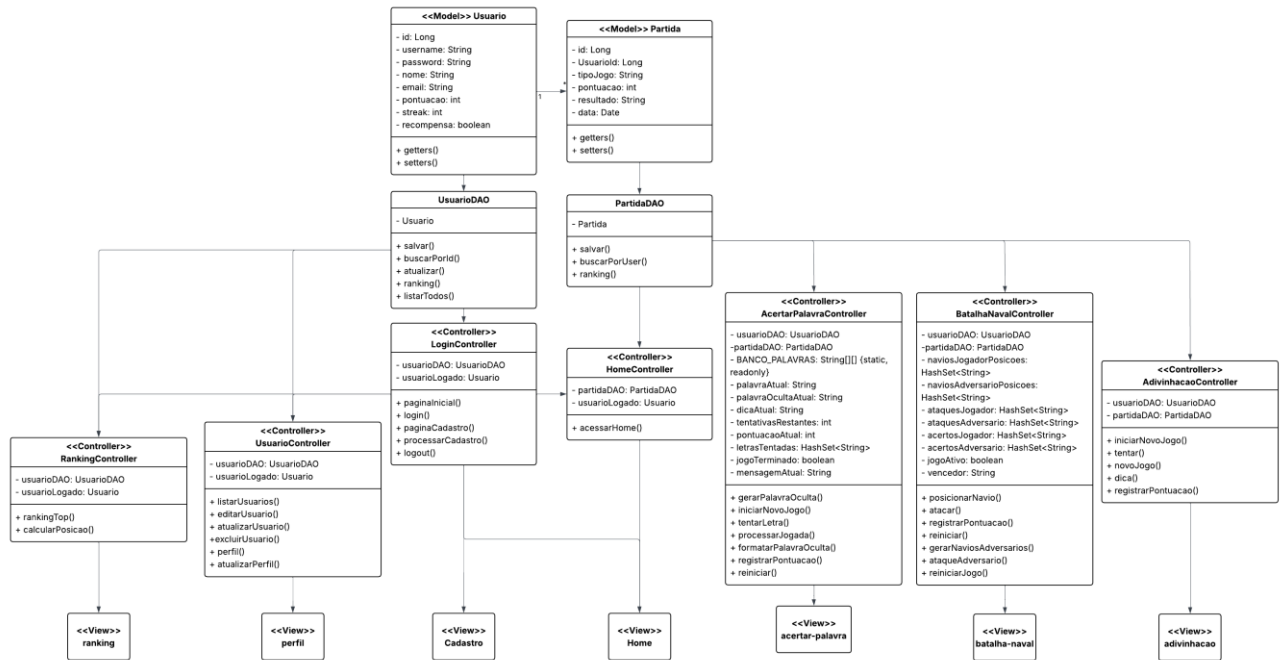
Release 1.1: Histórico de partidas e edição de perfil

Release 1.2: Novos jogos e sistema de conquistas

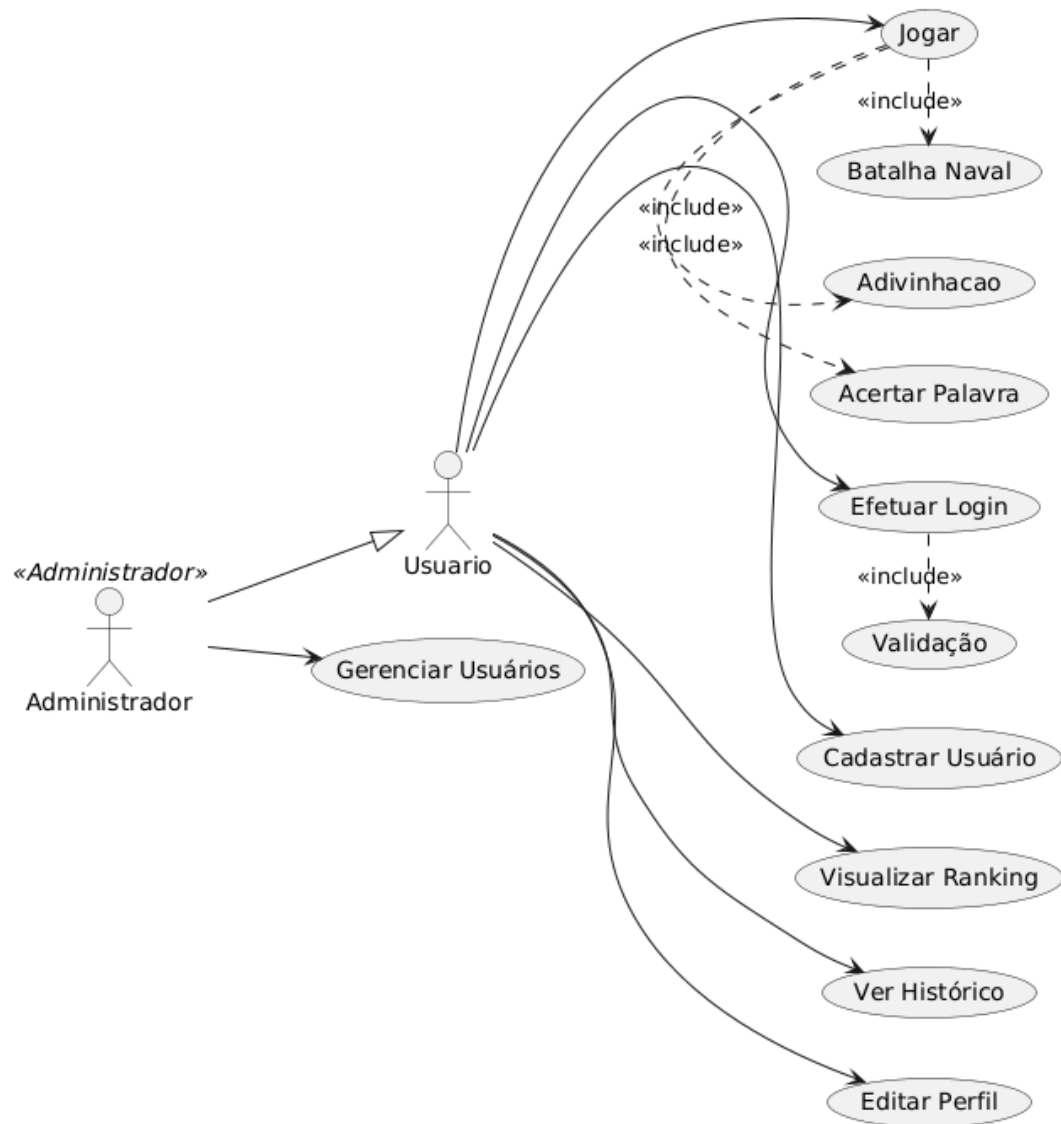
Release 2.0: Modo multiplayer e comunidades

Seção 2: Modelagem de Software (UML)

2.1 Diagrama de Classes

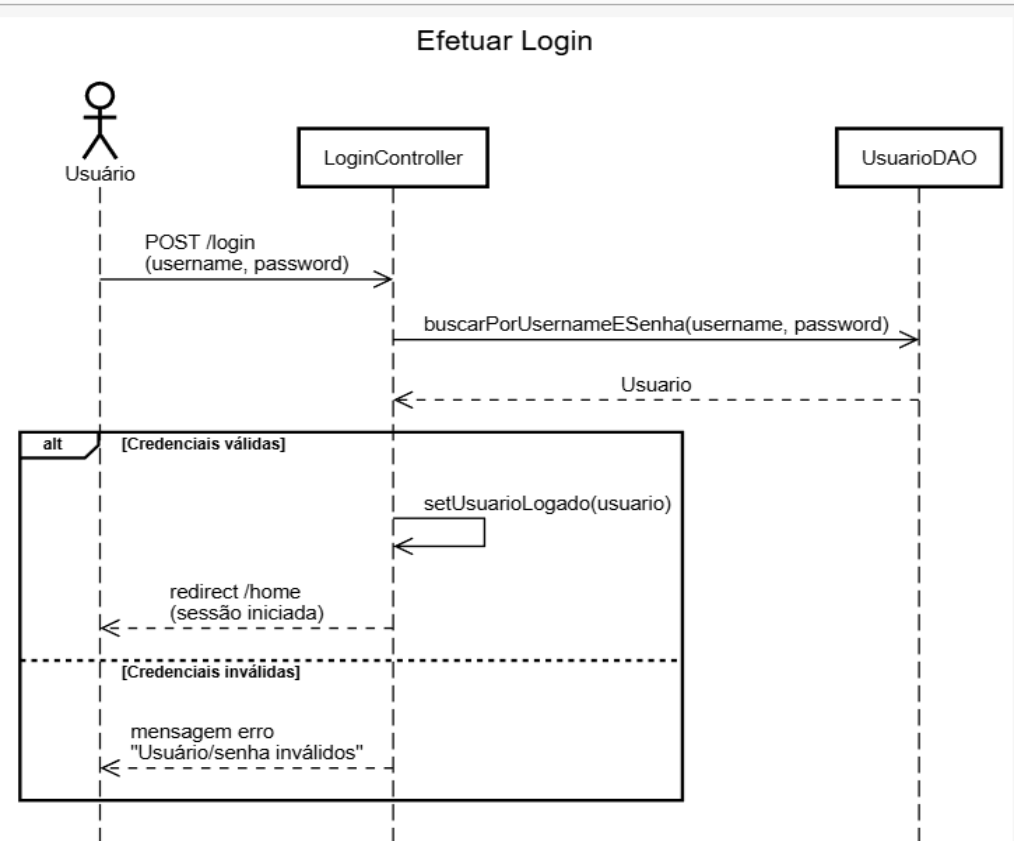


2.2 Diagrama de Casos de Uso

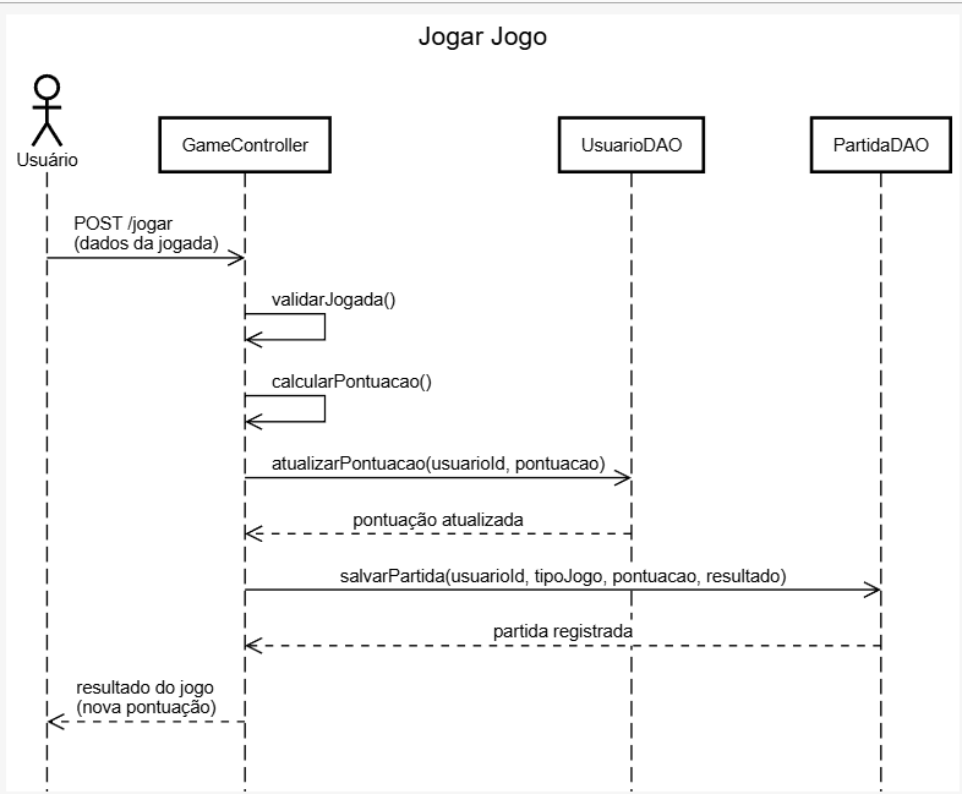


2.3 Diagrama de Sequência

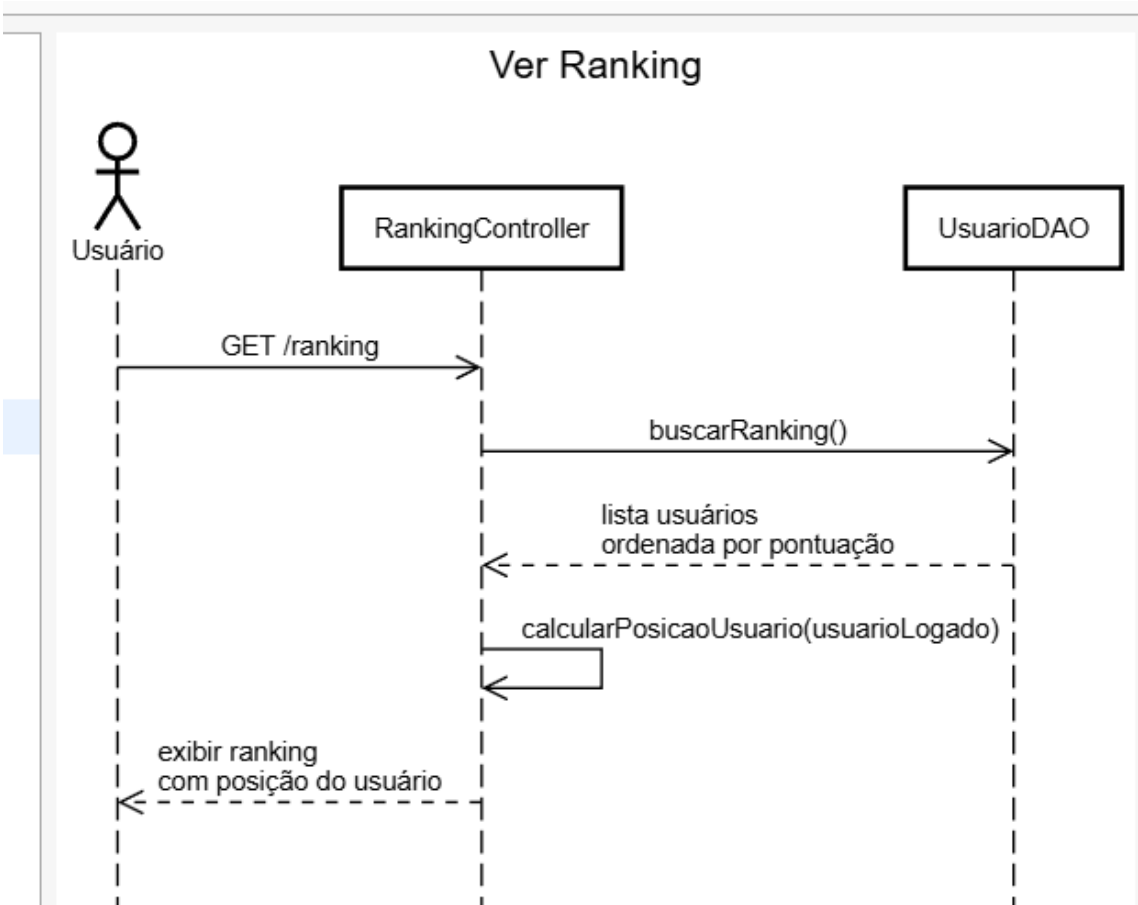
2.3.1. Efetuar Login



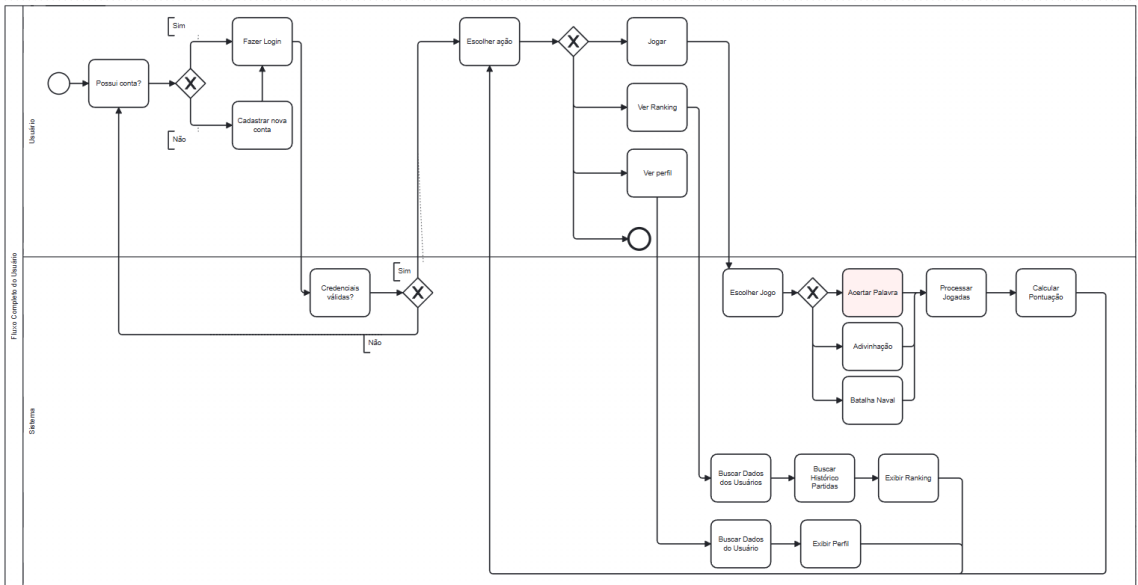
2.3.2. Jogar jogo



2.3.3. Ver Ranking



Seção 3: Modelagem de Processos (BPMN)



Seção 4: Especificação de Funcionalidade para Desenvolvimento

4.1 História de Usuário Crítica

Como: Usuário não cadastrado

Quero: Me cadastrar na plataforma

Para: Poder jogar e competir no ranking

Critérios de Aceitação:

- Deve validar username único
- Deve validar formato de email
- Deve criptografar a senha
- Deve redirecionar para login após cadastro
- Deve mostrar mensagens de erro claras

4.2 Design da Experiência do Usuário (UX)

Tela de Cadastro:



O mockup da tela de cadastro do Areninha é apresentado em um fundo roxo. No topo, há um ícone de controle de videogame roxo. Abaixo dele, o nome "Areninha" é exibido em uma fonte azul. O formulário contém os seguintes elementos: um rótulo "Usuário:" seguido por um campo de entrada com o placeholder "Digite seu usuário"; um rótulo "Senha:" seguido por um campo de entrada com o placeholder "Digite sua senha"; um botão azul com o texto "Entrar"; e um botão branco com o texto "Cadastrar". Na base do formulário, há o texto "Teste rápido:" seguido por "Use: admin / admin123".

Fluxo de Cadastro (Storyboard):

1. **Tela Inicial** → Botão "Cadastrar"
2. **Formulário** → Preencher dados
3. **Validação** → Processamento

4. **Sucesso** → Redireciona para Login
5. **Erro** → Mostra mensagem específica

4.3 Contrato de Qualidade (Casos de Teste)

Caso de Teste 1: Cadastro Bem-Sucedido (Caminho Feliz)

Pré-condições:

- Usuário não está logado
- Dados são válidos e únicos

Passos:

1. Acessar /cadastro
2. Preencher:
 - Username: "novousuario"
 - Password: "senha123"
 - Nome: "Novo Usuário"
 - Email: "novo@email.com"
3. Clicar em "Cadastrar"

Resultado Esperado:

- Sistema redireciona para /login
- Mensagem de sucesso é exibida
- Usuário é criado no banco de dados
- Senha está criptografada

Caso de Teste 2: Cadastro com Erro (Caminho de Erro)

Pré-condições:

- Username "teste" já existe no sistema

Passos:

1. Acessar /cadastro
2. Preencher:
 - Username: "teste" (já existente)
 - Password: "senha123"
 - Nome: "Usuário Teste"

- Email: "teste@email.com"

3. Clicar em "Cadastrar"

Resultado Esperado:

- Sistema permanece na página de cadastro
- Mensagem: "Usuário já existe!"
- Campos mantêm os valores preenchidos
- Nenhum usuário duplicado é criado

Casos de Teste Adicionais:

- Cadastro com email inválido
- Cadastro com campos vazios
- Cadastro com senha muito curta
- Teste de responsividade da interface

5. Apêndice

Repositório GitHub: <https://github.com/PedroReyss/Areninha-Project/tree/main>

Tecnologias Utilizadas:

- Backend: Java Spring Boot
- Frontend: Thymeleaf + HTML/CSS/JavaScript
- Banco de Dados: H2 (desenvolvimento)
- Build: Maven
- Versionamento: Git

Estrutura do Projeto:

text

src/

```
|— main/
|  |— java/com/cefsa/areninha/
|  |  |— controller/    # Controladores
|  |  |— model/         # Entidades
|  |  |— dao/           # Data Access Objects
|  |  |— config/        # Configurações
|  |  |— resources/
|  |  |— templates/     # Views Thymeleaf
|  |  |— static/        # CSS, JS, imagens
|  |  |— application.properties
|  |— test/             # Testes unitários
```

Instruções de Execução:

1. Clone o repositório
2. Crie o banco de dados
3. Ajuste o login e a senha do banco de dados em application.properties
4. Execute mvn spring-boot:run
5. Acesse <http://localhost:8080>
6. Use credenciais de teste ou crie novo usuário