|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | **Integrantes:** | 1272212363 | Edu Argôlo Guerreiro da Silva | |  | 1272212914 | Laís Magalhães Viana | |  | 1272224679 | Sinval Luz Souza Filho | |  | 1272313309 | Guilherme Almeida Guimarães | | **Turno:** | Noturno | | |  |  | |  |  |

**DEFINIÇÕES DO SISTEMA**

**SISTEMA DE GESTÃO PARA ATLÉTICA DE E-SPORTS (LOL)**

1. **INTRODUÇÃO**

Este documento descreve o projeto do **Sistema de Gestão para Atlética de E-sports**, focado no jogo **League of Legends (LoL)**. O objetivo é facilitar o gerenciamento de jogadores, treinos, partidas e estatísticas, proporcionando um ambiente organizado para a equipe.

1. **ESCOPO DO SISTEMA**
   1. **Objetivo**

O sistema permitirá o gerenciamento de **jogadores, categorias, treinos, partidas e estatísticas** de desempenho, automatizando registros e facilitando a análise de informações.

**2.2 Usuários e Níveis de Acesso**

* **Administrador:** Controle total do sistema.
* **Técnico:** Gerencia jogadores, treinos e partidas.
* **Jogador:** Acompanha suas próprias estatísticas e treinos.
* **Espectador:** Acessa apenas estatísticas e resultados públicos.

**2.3 Principais Funcionalidades**

* Cadastro e gerenciamento de jogadores.
* Registro e acompanhamento de treinos.
* Cadastro de partidas e torneios.
* Análise de estatísticas e relatórios de desempenho.
* Login e controle de acesso seguro.

**Fronteiras do Sistema**

* Não incluirá sistema de pagamentos.
* Não será uma rede social.
* Não realizará transmissão ao vivo de partidas.

1. **Requisitos Funcionais**

O sistema deve permitir:

1. O cadastro, edição e exclusão de jogadores.
2. Registro de treinos e presença dos jogadores.
3. A associação de jogadores a partidas e torneios.
4. Relatórios de desempenho individual e comparativo.
5. Login e permissões de acesso conforme o perfil.
6. Micro-serviço atualize estatísticas automaticamente via Riot Games API.
7. **Requisitos Não Funcionais**
8. O sistema deve ser desenvolvido em **Next.js + TypeScript**.
9. O banco de dados deve ser **MySQL**.
10. O sistema deve ser responsivo e acessível via navegador.
11. O micro-serviço deve rodar na nuvem (AWS Lambda ou alternativa gratuita).
12. As senhas dos usuários devem ser armazenadas com criptografia.
13. **Regras de Negócio**
14. Um jogador pode pertencer a apenas uma categoria por vez.
15. Apenas técnicos podem cadastrar e editar treinos.
16. Apenas administradores e treinadores podem cadastrar torneios.
17. Um jogador não pode estar em duas partidas simultâneas.
18. Somente administradores podem excluir registros do sistema.

**7. Protótipo Wireframe**

Diagrama

O conteúdo gerado por IA pode estar incorreto.Diagrama

O conteúdo gerado por IA pode estar incorreto.Tabela

O conteúdo gerado por IA pode estar incorreto.Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.Tabela

O conteúdo gerado por IA pode estar incorreto.Uma imagem contendo Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

Tabela

O conteúdo gerado por IA pode estar incorreto.

1. **Ferramentas**

* **Front-end & Back-end:** Next.js + TypeScript.
* **Banco de Dados:** MySQL.
* **Micro serviço para estatísticas:** Node.js com integração à RIOT API.
* **Hospedagem:** Front-end Vercel, Back-end Hostinger

1. **Diagrama de Casos de Uso**

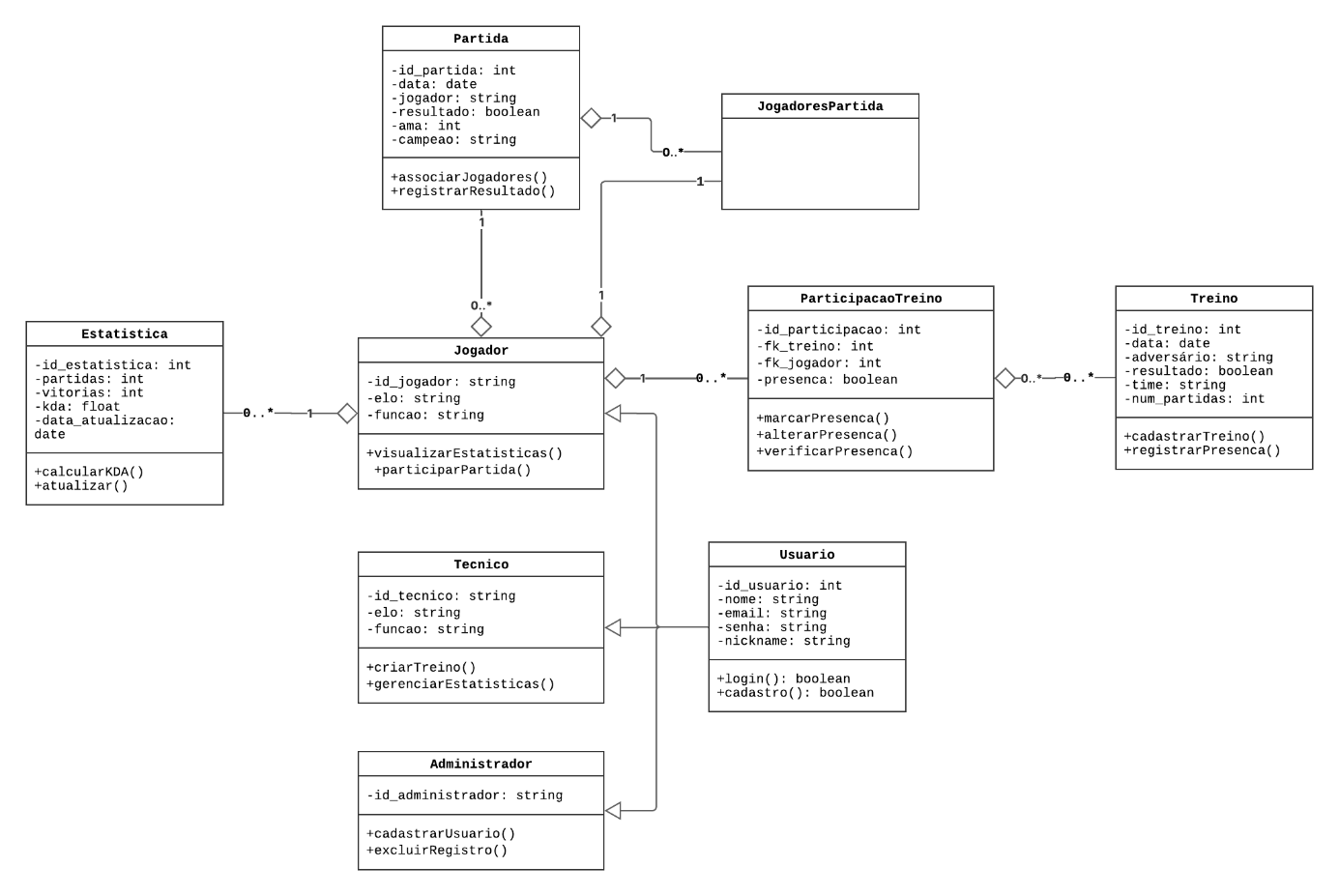
* **Atores do Sistema:** Administrador, Técnico, Jogador, Espectador, Microsserviço.
* **Casos de Uso Administrador:** Fazer login; gerenciar usuários (cadastrar, editar, excluir); cadastrar partidas; excluir registros do sistema, visualizar estatísticas (treinos, estatísticas públicas).
* **Casos de Uso Técnico:** Fazer login; cadastrar partidas/treinos; registrar presença; associar jogadores a partidas; visualizar estatísticas (treinos, estatísticas públicas).
* **Casos de Uso Jogador:** Fazer login; visualizar estatísticas (treinos, estatísticas públicas).
* **Casos de Uso Espectador:** Visualizar estatísticas (treinos, estatísticas públicas) – sem login.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

1. **Diagrama de Classes**

O diagrama de classes apresentado representa a estrutura estática do sistema de gerenciamento de uma atlética universitária de E-Sports, com foco na modalidade League of Legends. Este sistema visa organizar a administração dos usuários, o controle de treinos e partidas, além de disponibilizar estatísticas relacionadas ao desempenho dos jogadores.

****