|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | **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. Microsserviço atualize estatísticas automaticamente via Riot Games API.
7. **Requisitos Não Funcionais**
8. Tecnologias que devem ser utilizadas:

* Linguagem de programação **TypeScript v4.5.2**
* Framework fullstack **Next.JS v15.3.3**
* Framework de estilização **Tailwind CSS v4**
* Biblioteca de ícones **lucide v0.511.0**
* Mapeamento objeto-relacional **Prisma v6.8.2**
* Sistema gerenciador de banco de dados **mySQL v8.4**

1. Desempenho:

* O site deve seguir as funcionabilidades de uma Single Page Applications (SPA)
* O tempo de resposta da api deve ser do no máximo 2segundos
* O banco de dados deve comportar todos os campeões do League of Legends

1. Segurança:

* Todas as comunicações devem ser criptografadas via HTTPS.
* O site deve implementar autenticação de dois fatores para usuários administrativos.
* Dados sensíveis como senha devem estar hashadas no banco de dados
* Deve haver proteção contra-ataques de injeção SQL, XSS e CSRF.

1. Usabilidade:

* A navegação deve ser intuitiva e acessível para novos usuários sem treinamento.
* O site deve estar em conformidade com o padrão de acessibilidade WCAG 2.1 nível AA.

1. Compatibilidade:

* O site deve ser responsivo e funcionar corretamente em dispositivos móveis, tablets e desktops.
* O sistema deve ser compatível com as versões mais recentes dos principais navegadores: Chrome, Firefox, Edge e Safari.

1. **Regras de Negócio**
2. Estatísticas e Desempenho:

* Estatísticas de partidas só podem ser registradas por usuários com permissão de capitão ou técnico.
* Cada partida registrada deve conter, no mínimo, data, nome dos jogadores, time adversário e resultado
* Estatísticas de scrims (treinos) são separadas das estatísticas de campeonatos
* Histórico de estatísticas é mantido por no mínimo 1 ano e não pode ser deletado.

1. Agendamento de Treinos:

* Treinos só podem ser agendados por líderes ou técnicos do time.
* Um treino deve ter no mínimo 2 horas de antecedência para ser agendado.
* É possível marcar treinos apenas nos horários disponíveis definidos no calendário da equipe.
* Um jogador pode confirmar ou recusar presença no treino até 1 hora antes do início.

1. Controle de Acesso e Usuários:

* Existem três níveis de usuário: Jogador, Técnico/Coach, capitão.
* Apenas o capitão e técnico tem nível de administrador
* Apenas administradores podem cadastras ou remover membros do time, agendar treinos e editar estatísticas.
* Jogadores só podem visualizar estatísticas próprias e do time, sem editar.

1. Construção do sistema

* O login deve ter uma função de esquecer senha
* Após o login ser efetuado com sucesso o usuário é direcionado até o dashboard principal
* O título principal da página deve seguir os parâmetros de rota

1. Dashboard Técnico
2. Gerenciamento de jogadores
3. Treinos
4. Estatísticas
5. Treinos

* Toda página do site deve conter uma lista de navegação com Links para cada página do site

1. Início
2. Jogadores
3. Treinos
4. Partidas
5. Estatísticas

* **Dashboard técnico**
  1. Tabela

     O conteúdo gerado por IA pode estar incorreto.Toda miniatura é também um botão para acessar a página do que está sendo mostrado
* **Jogadores** 
  1. Deve ter um input com um Botão para buscar Jogadores, para facilitar a filtragem de jogadores
  2. Uma barra de rolagem é necessária para a quantidade de jogadores elevada
  3. A tabela de jogadores deve mostrar os seguintes parâmetros
     + Nome
     + Nick
     + Função
     + Situação
       - Titular
       - Reserva
* Elo
  1. Deve existir um botão para adicionar um novo jogador no time

Tabela

O conteúdo gerado por IA pode estar incorreto.

* **Estatísticas**

1. É necessário um indicador de qual jogador está sendo mostrado nas estatísticas
2. Um gráfico mostrando o desempenho do jogador ao longo dos meses
3. Um gráfico com as vitorias e derrotas do time ao longo dos meses
4. O histórico de treinos do time

* Uma imagem contendo Interface gráfica do usuário

  O conteúdo gerado por IA pode estar incorreto.**Treinos**

1. É necessário um input para a busca de treinos através do adversário
2. Uma tabela mostra treinos do time

**Tabela

O conteúdo gerado por IA pode estar incorreto.**

Interface gráfica do usuário, Aplicativo

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. **Diagrama de Casos de Uso**

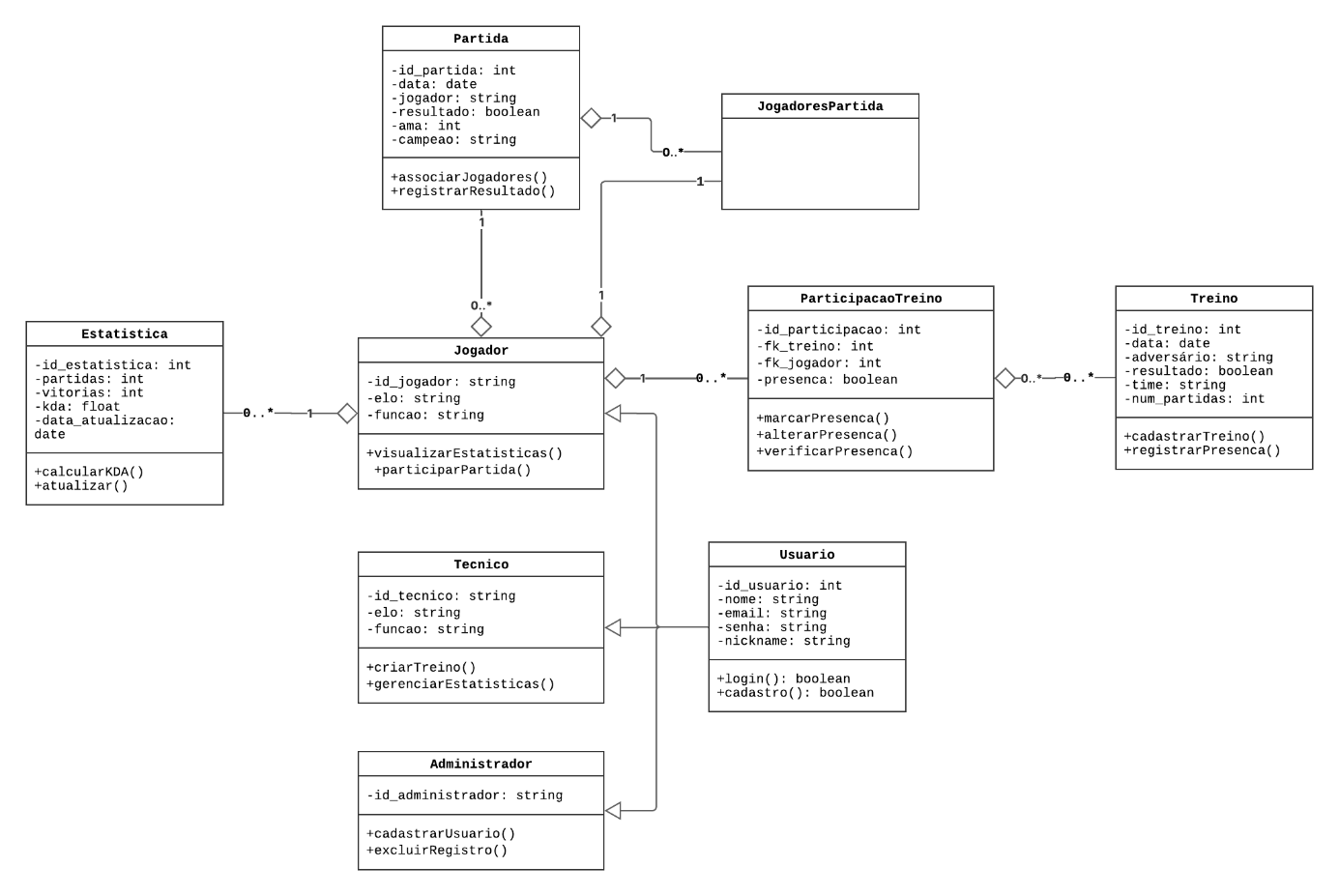
* **Atores do Sistema:** Administrador, Técnico, Jogador, Espectador, Microsserviço.
* **Casos de Uso Administrador:** Fazer login; gerenciar usuários (cadastrar, editar, excluir); cadastrar partidas; excluir registros do sistema, visualizar estatísticas (treinos, estatísticas públicas).
* **Casos de 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.

****