

**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 Atlético 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.

#### **2. ESCOPO DO SISTEMA**

##### **2.1 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.

### **3. 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. Microserviço atualize estatísticas automaticamente via Riot Games API.

### **4. Requisitos Não Funcionais**

1. 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**
2. 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

### 3. 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.

### 4. 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.

### 5. 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.

## 6. Regras de Negócio

### 1. 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.

## 2. 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.

## 3. 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.

## 4. Construção do sistema

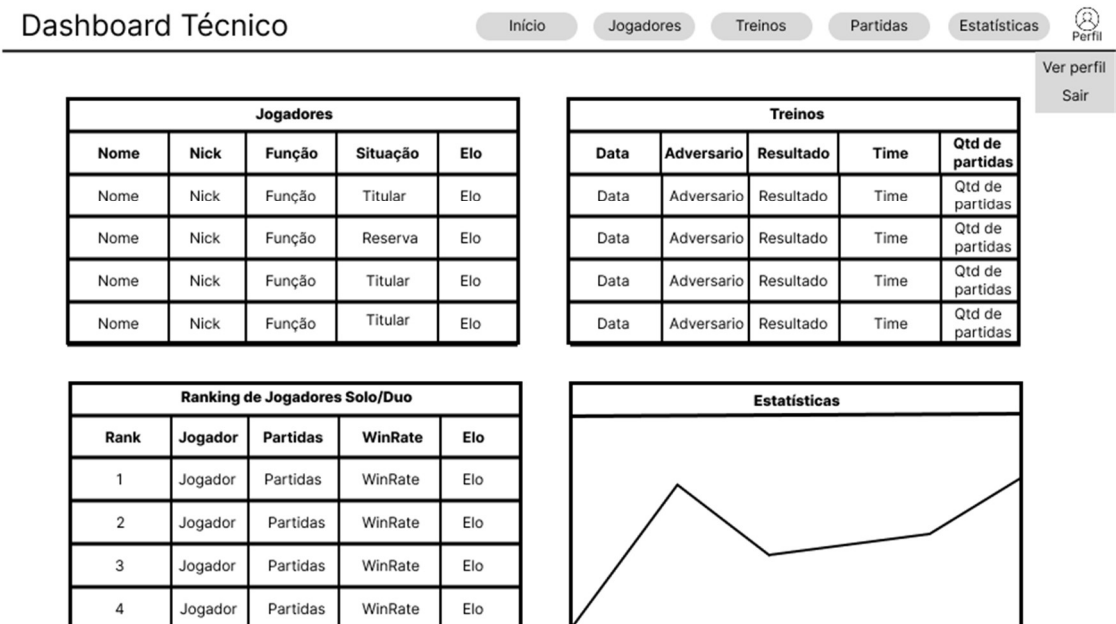
- O login deve ter uma função de esquecer senha
- Após o login ser efetuado com sucesso o usuário é direcionado até o dashboard principal
- O título principal da página deve seguir os parâmetros de rota
  1. Dashboard Técnico
  2. Gerenciamento de jogadores
  3. Treinos
  4. Estatísticas
  5. Treinos
- 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. Toda miniatura é também um botão para acessar a página do que está sendo mostrado
2. No icone perfil é também um botão para que mostra outros dois botões um para ver o perfil e o outro para Sair da conta redirecionando para o Login

Section 1



- **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

4. Deve existir um botão para adicionar um novo jogador no time

Section 2

Jogadores

Início
Jogadores
Treinos
Partidas
Estatísticas
 Perfil

Buscar Jogador

Jogadores				
Nome	Nick	Função	Situação	Elo
Nome	Nick	Função	Titular	Elo
Nome	Nick	Função	Reserva	Elo
Nome	Nick	Função	Titular	Elo
Nome	Nick	Função	Titular	Elo

Adicionar Jogador

### • 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 vitórias e derrotas do time ao longo dos meses
4. O histórico de treinos do time

## Estatísticas

Início

Jogadores

Treinos

Partidas

Estatísticas



Digite o nome do jogador

Buscar Jogador

Jogador : Nome do jogador



Historico de partidas				
Data	Campeão	KDA	Resultado	Adversário
Data	Campeão	KDA	Resultado	Adversário
Data	Campeão	KDA	Resultado	Adversário
Data	Campeão	KDA	Resultado	Adversário
Data	Campeão	KDA	Resultado	Adversário
Data	Campeão	KDA	Resultado	Adversário



Histórico do time			
Data	Adversário	Resultado	Partidas
Data	Adversário	Resultado	Partidas
Data	Adversário	Resultado	Partidas
Data	Adversário	Resultado	Partidas
Data	Adversário	Resultado	Partidas
Data	Adversário	Resultado	Partidas

- Treinos**

1. É necessário um input para a busca de treinos através do adversário
2. Um botão para buscar o treino referente ao input
3. Uma tabela mostra treinos do time com:
  - a. Data
  - b. Adversário
  - c. Resultado
  - d. Time (principal, reserva, mesclado, academy)
  - e. Quantidade de partidas
4. Tem um botão para adicionar um novo treino

## Treinos

Início

Jogadores

Treinos

Partidas

Estatísticas




Buscar treino

Treinos					Adicionar Treino
Data	Adversario	Resultado	Time	Qtd de partidas	
Data	Adversario	Resultado	Time	Qtd de partidas	
Data	Adversario	Resultado	Time	Qtd de partidas	
Data	Adversario	Resultado	Time	Qtd de partidas	
Data	Adversario	Resultado	Time	Qtd de partidas	

- Ranking de jogadores
  1. Nessa tela terá uma tabela de ranking dos jogadores que estão desempenhando melhor no modo ranqueado do jogo, essa tabela refere-se ao desempenho individual de cada jogador.
  2. Nessa tabela terá as seguintes colunas
    - a. Rank dentro do time
    - b. Jogador
    - c. Partidas
    - d. Winrate em porcentagem



e. Elo, rank dentro do jogo

Section 5

Partidas

Início

Jogadores

Treinos

Partidas

Estatísticas



Perfil

Ranking de Jogadores Solo/Duo				
Rank ▼	Jogador ▼	Partidas	WinRate	Elo
1	Jogador	Partidas	WinRate	Elo
2	Jogador	Partidas	WinRate	Elo
3	Jogador	Partidas	WinRate	Elo
4	Jogador	Partidas	WinRate	Elo

- **Login**

- **Email:**

- Deve estar cadastrado previamente no sistema.
    - Formato válido (ex: usuario@dominio.com).

- **Senha:**

- Deve corresponder à senha cadastrada para o email informado.
    - Em caso de 5 tentativas falhas consecutivas, o sistema pode bloquear temporariamente o acesso (segurança).

- **Botão "Entrar":**

- Autêntica o usuário e o redireciona à área principal do sistema em caso de sucesso.

- **Link "Registre-se":**

- Redireciona o usuário para o formulário de cadastro.

# Login

Email:

Senha:

[Registre-se](#)

Entrar

## 2. Cadastro

- **Nome completo:**
  - Obrigatório.
  - Deve conter pelo menos 3 caracteres.
- **Email:**
  - Obrigatório.
  - Formato válido.
  - Não pode estar cadastrado previamente no sistema.
- **Nickname + TAG:**
  - Obrigatório.
  - Deve seguir o padrão do jogo (ex: jogador#1234).
  - Não pode ser duplicado entre os usuários.
- **Função no jogo:**
  - Obrigatório.
  - Deve ser uma das funções válidas do jogo (por exemplo: Top, Jungle, Mid, ADC, Suporte).

- **Senha:**
  - Obrigatória.
  - Mínimo de 6 caracteres.
  - Recomenda-se o uso de letras maiúsculas, minúsculas, números e símbolos.
- **Botão "Registrar":**
  - Salva os dados no sistema, se todas as validações forem atendidas.
  - Redireciona o usuário para a tela de login.
- **Link "Login":**
  - Redireciona o usuário para a tela de login.

## Cadastro:

Nome completo:

Email:

Nickname +TAG:

Função no jogo:

Senha:

[Login](#)

**Registrar**

## 7. Diagrama de Casos de Uso

- **Atores do Sistema:** Administrador, Técnico, Jogador, Espectador, Microserviço.
- **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.

## 8. Descrição dos Casos de Uso

### 1. Caso de Uso 1 – Fazer Login

1. Atores: todos
2. Resumo: permite ao usuário acessar o sistema de acordo com o seu perfil.
3. Pré-condições: o usuário deve estar cadastrado
4. Pós-condições: o sistema libera acesso à interface do respectivo perfil

#### 5. Fluxo Principal:

1. O usuário acessa a tela inicial do sistema.
2. Informa o seu Email e senha.
3. Clica no botão “Entrar”.
4. O sistema verifica as credenciais
5. Se válidas, redireciona para o dashboard correspondente (Administrador, Técnico ou Jogador).

#### 6. Fluxo Alternativo A – Senha incorreta:

1. O sistema exibe uma mensagem de erro “Senha ou email inválido”.
2. O usuário pode tentar novamente ou solicitar recuperação de senha.

**7. Fluxo Alternativo B – Conta inexistente:**

1. O sistema informa que o usuário não foi encontrado

**2. Caso de Uso 02 – Cadastrar Jogador**

1. Ator principal: Técnico
2. Resumo: Permite ao técnico cadastrar um novo jogador.
3. Pré-condição: Usuário deve estar autenticado como técnico.
4. Pós-condição: Novo jogador é adicionado ao sistema com vínculo à categoria.

**5. Fluxo Principal:**

1. O técnico acessa a aba “Jogadores”.
2. Clica em “+ Novo Jogador”.
3. Preenche os campos: nome, nickname, email, categoria.
4. Define uma senha temporária para o jogador.
5. Clica em “Salvar”.
6. O sistema valida e armazena os dados.
7. Exibe mensagem de sucesso.

**6. Fluxo Alternativo A – Campos obrigatórios em branco:**

1. O sistema impede o envio e destaca os campos ausentes.

**7. Fluxo Alternativo B – Email já cadastrado:**

1. O sistema exibe erro: “Email já está em uso”.

**3. Caso de Uso 3 – Registrar Treino**

1. Ator principal: Técnico e Administrador
2. Resumo: Permite ao técnico registrar um novo treino.
3. Pré-condição: Técnico autenticado.
4. Pós-condição: Treino é salvo e jogadores podem ter presença marcada.

**5. Fluxo Principal:**

1. Técnico acessa a aba “Treinos”.
2. Clica em “+ Novo Treino”.
3. Preenche data, horário, descrição e categoria.
4. Clica em “Salvar”.
5. O sistema salva o treino e o adiciona à agenda.

**6. Fluxo Alternativo A – Data em conflito com outro treino:**

1. O sistema exibe um alerta: “Treino já agendado para este horário e categoria”.

**7. Fluxo Alternativo B – Categoria sem jogadores vinculados:**

1. O sistema alerta: “Não há jogadores cadastrados nessa categoria”.

**4. Caso de Uso 4 – Registrar Presença**

1. Ator principal: Técnico
2. Resumo: Permite marcar a presença dos jogadores em um treino.
3. Pré-condição: Treino deve existir e jogadores vinculados.
4. Pós-condição: Presença registrada por jogador.

**5. Fluxo Principal:**

1. Técnico acessa um treino específico.
2. Clica em “Registrar Presença”.
3. Visualiza lista de jogadores da categoria.
4. Marca os jogadores presentes.
5. Clica em “Salvar”.
6. Sistema salva a lista de presença.

**6. Fluxo Alternativo A – Nenhum jogador marcado:**

1. O sistema alerta: “Nenhum jogador foi selecionado”.

**5. Caso de Uso 5 – Visualizar Estatísticas:**

1. Ator principal: Jogador, Técnico, Administrador, Espectador
2. Resumo: Permite consultar as estatísticas de performance.
3. Pré-condição: Estatísticas devem estar registradas no banco.
4. Pós-condição: Dados apresentados ao usuário.

**5. Fluxo Principal:**

1. Usuário acessa a aba “Estatísticas”.
2. Visualiza dados como: KDA, vitórias, derrotas, últimas atualizações.
3. Jogador vê apenas seus dados; técnico/admin veem todos.

**6. Fluxo Alternativo – Dados não atualizados:**

1. O sistema exibe a data da última atualização.
2. Se passar de 7 dias, sistema alerta: “Atualização necessária”.

**6. Caso de Uso 6 – Atualizar Estatísticas (Automático)**

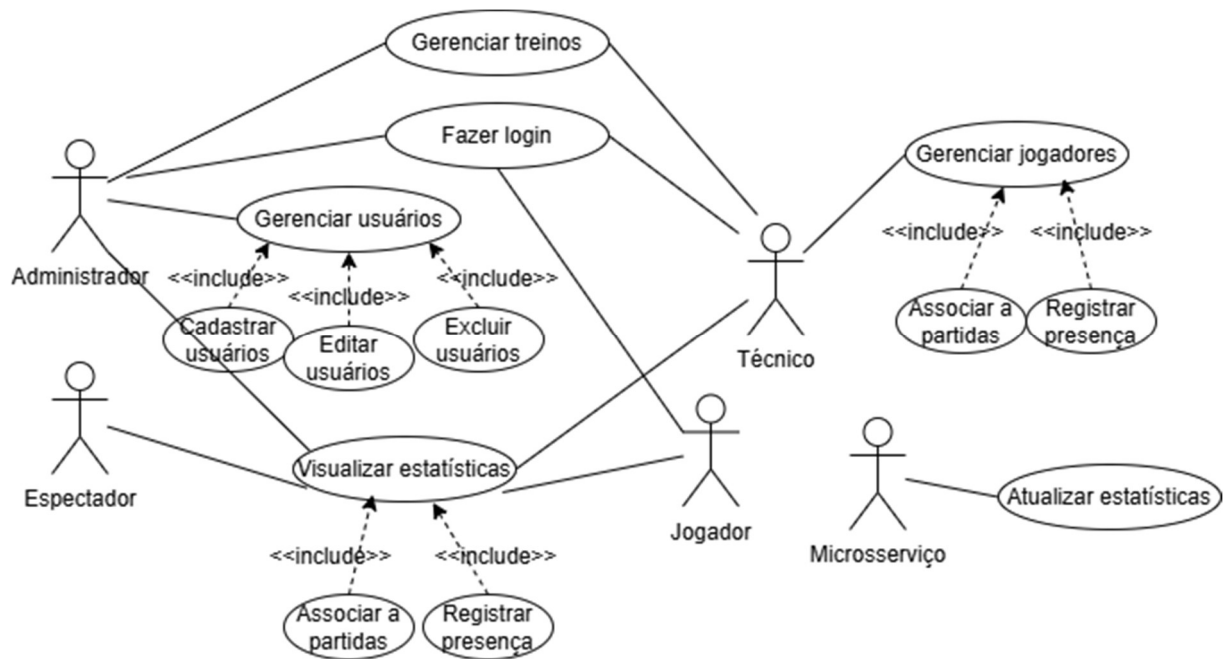
1. Ator: Microserviço do sistema
2. Resumo: Coleta estatísticas atualizadas da API da Riot.
3. Pré-condição: Jogador deve ter link válido de perfil na OP.GG.
4. Pós-condição: Banco de dados atualizado.

**5. Fluxo Principal:**

1. Microserviço roda diariamente às 03h.
2. Coleta dados dos jogadores ativos na última semana.
3. Atualiza registros no banco de dados.

**6. Fluxo Alternativo – Falha de API externa:**

1. O sistema tenta novamente em 10 minutos.
2. Caso não consiga após 3 tentativas, envia log de erro ao administrador.



## 9. Diagrama de Classes

O diagrama de classes apresentado representa a estrutura estática do sistema de gerenciamento de uma atlética universitária de 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. Segue descrição detalhada abaixo:

1. **Classe Usuario e Herança:** A classe Usuario é uma classe base (abstrata) que define os atributos e métodos comuns a todos os usuários do sistema. Dela derivam as subclasses Administrador, Técnico e Jogador, cada uma com suas responsabilidades específicas.
  1. Administrador pode cadastrar e excluir usuários.
  2. Técnico pode criar treinos e gerenciar estatísticas.
  3. Jogador pode visualizar suas estatísticas e participar de partidas.
2. **Classe Estatística:** Cada jogador possui uma estatística individual vinculada a ele por um relacionamento 1:1. A classe Estatística armazena dados de desempenho como KDA, vitórias, derrotas e número de partidas. Ela também possui métodos como calcularKDA() e atualizar (), representando operações automáticas ou manuais de atualização dos dados.
3. **Classe Treino e Participacao:** Treinos são agendamentos criados pelo técnico para os jogadores. Como um jogador pode participar de vários treinos e um treino pode ter vários jogadores, o relacionamento é N:N, implementado através da classe intermediária ParticipacaoTreino.



4. **Classe Partida e Participação:** Similar aos treinos, a participação dos jogadores em partidas é representada por um relacionamento N:N através da classe ParticipacaoPartida. Essa classe permite registrar

1. Qual jogador participou de qual partida
2. Qual foi sua função (ex: top, jungle, suporte)
3. Se foi titular ou reserva

