



**UNIVERSIDADE FEDERAL DO MARANHÃO**  
**Bacharelado Interdisciplinar em Ciências e Tecnologia**  
**Projeto e Desenvolvimento de Software**

**Giordano Bruno de Araujo Mochel**  
**Vitor Ferreira Nunes**

**Script Inteligente**

## Sumário

1.0	Introdução .....	3
2.0	Descrição Geral do Produto .....	3
2.1	Situação Atual.....	4
2.2	Atores.....	4
2.3	Premissas .....	5
3.0	Requisitos .....	6
3.1	Requisitos Funcionais .....	6
3.2	Requisitos Não Funcionais.....	7
4.0	Diagramas .....	7
4.1	Identificação dos Casos de Uso.....	7
4.2	Modelagem de Casos de Uso .....	19
4.3	Modelagem de Classes .....	22
4.4	Modelagem de Sequências.....	25
4.5	Modelagem de Atividades .....	29

# 1.0 Introdução

A automação de tarefas é uma prática comum entre desenvolvedores de software, destinada a otimizar processos e aumentar a eficiência no ambiente de trabalho. No entanto, a ocorrência de erros nos scripts desenvolvidos pode causar frustração e desperdício de recursos. Pensando em solucionar esses problemas, apresentamos o "Script Inteligente".

O "Script Inteligente" é uma solução inovadora e eficaz projetada para analisar o código-fonte de scripts e corrigi-los. É implementado métodos que além de identificar a os erros de semântica e sintaxe do código, também detectam as lógicas ou finalidades para qual seu script está sendo desenvolvido. O projeto utiliza algoritmos avançados e inteligência artificial, o sistema fornece uma análise detalhada e sugestões de correção, permitindo ao usuário melhorar a qualidade de seus scripts de forma automatizada e eficiente.

Este sistema integra-se com APIs externas, oferecendo uma interface web intuitiva que facilita a interação do usuário com a plataforma, ao mesmo tempo que fornece uma aplicação modo offline da qual descompromete o usuário com a internet para usufruir do sistema. Além disso, o "Script Inteligente" garante segurança e privacidade, personalização, integração com outros sistemas, e suporte especializado.

As etapas de utilização do sistema incluem a submissão do script, análise detalhada, identificação de erros, fornecimento de feedback, e validação e testes automatizados. O objetivo final é fornecer ao usuário um código funcional e de alta qualidade, minimizando a ocorrência de erros e maximizando a eficiência do desenvolvimento de software, e executa-lo para testar sua validade.

Com a implementação do "Script Inteligente", desenvolvedores podem focar em tarefas mais complexas e criativas, enquanto o sistema cuida da identificação e correção de erros em scripts, resultando em um ambiente de trabalho mais produtivo, dinâmico e livre de frustrações.

# 2.0 Descrição Geral do Produto

O "Script Inteligente" é uma plataforma avançada de automação e correção de scripts, desenvolvida para otimizar o processo de desenvolvimento de software. Projetado para atender às necessidades de desenvolvedores que buscam eficiência

e precisão, o sistema oferece uma solução abrangente para a análise, correção e validação de scripts de código.

## 2.1 Situação Atual

A criação do projeto "Script Inteligente" foi motivada por desafios recorrentes enfrentados por desenvolvedores de software, tal como Tom, no seu dia a dia de trabalho. Tom, um desenvolvedor de software experiente, frequentemente cria scripts para automatizar tarefas em seu ambiente de trabalho. No entanto, ele constantemente enfrenta a frustração de lidar com erros em seus scripts, que muitas vezes só são identificados após a execução, resultando em desperdício de tempo e recursos.

## 2.2 Atores

A aplicação possui três atores, são eles:

Nº	Autor	Definição e Privilégio de Acesso
1	Administrador	<p>O Administrador é responsável pelo gerenciamento geral do sistema (herda funcionalidades de funcionário e usuário):</p> <ol style="list-style-type: none"><li>1. <b>Gerenciamento de Usuários:</b> Adicionar, editar e remover usuários (administradores, funcionários e usuários comuns).</li><li>2. <b>Configurações do Sistema:</b> Ajustar parâmetros do sistema, como integração com APIs e escolha de modelos de correção.</li><li>3. <b>Gerenciar Banco de Script:</b> Manipular a biblioteca de script do banco de dados</li><li>4. <b>Acesso Total:</b> Acesso irrestrito a todas as funcionalidades do sistema, incluindo submissão e análise de scripts, correção automática e execução de testes.</li></ol>
2	Funcionário	<p>O Funcionário auxilia no suporte aos usuários (também herda as funcionalidades do usuário):</p> <ol style="list-style-type: none"><li>1. <b>Visualização e Edição de Scripts:</b> Acesso para visualizar e revisar scripts submetidos pelos usuários.</li></ol>

		<ol style="list-style-type: none"> <li>2. <b>Gerenciamento de Dados:</b> Acesso para cadastrar e visualizar scripts de usuários. Gerenciar empréstimos e devoluções.</li> <li>3. <b>Fornecimento de Feedback:</b> Capacidade de fornecer feedback adicional sobre as correções realizadas.</li> </ol>
3	Usuário	<p>Responsável pelas funcionalidades padrões do sistema:</p> <ol style="list-style-type: none"> <li>1. <b>Cadastro e Login:</b> Registrar-se no sistema e realizar login.</li> <li>2. <b>Alterar Dados:</b> Alterar dados pessoais no sistema.</li> <li>3. <b>Submissão de Scripts:</b> Carregar arquivos de script ou inserir código diretamente para análise.</li> <li>4. <b>Visualização de Feedback:</b> Acessar relatórios detalhados dos erros identificados e sugestões de correção.</li> <li>5. <b>Aplicação de Correções:</b> Implementar as correções sugeridas pelo sistema</li> <li>6. <b>Execução de Testes:</b> Rodar testes automatizados para validar as correções aplicadas aos scripts.</li> </ol>

## 2.3 Premissas

**Banco de Dados:** O sistema requer dois bancos de dados, um embutido no modelo computacional, afim de garantir um modo offline da aplicação, e um não embutido via serviço web. A sincronização entre os bancos será feita quando o usuário entrar em modo online.

**Manutenção Contínua:** A aplicação deve ser mantida e atualizada continuamente para garantir a correção de bugs, implementação de novas funcionalidades e adaptação às necessidades emergentes dos usuários.

Essas premissas estabelecem as bases para o desenvolvimento e operação do "Script Inteligente", assegurando que o sistema atenda aos padrões de qualidade esperados e às necessidades dos usuários de maneira eficaz e eficiente.

## 3.0 Requisitos

### 3.1 Requisitos Funcionais

<i>Requisitos Funcionais</i>		
Código	Nome	Descrição
RF1	<b>Análise de Erros</b>	O sistema deve ser capaz de analisar scripts submetidos para identificar erros de sintaxe, semântica e lógica.
RF2	<b>Feedback de Erros</b>	Após a análise, o sistema deve fornecer feedback detalhado ao usuário, indicando os erros encontrados e sugerindo correções.
RF3	<b>Integração com API ou Modelo Interno</b>	O sistema deve integrar-se com APIs externas ou utilizar modelos internos para realizar a análise e correção de scripts.
RF4	<b>Corrigir Erros Identificados</b>	O sistema deve aplicar correções automáticas aos erros identificados nos scripts submetidos.
RF5	<b>Controle de Acesso</b>	O sistema deve implementar um controle de acesso que diferencie os níveis de permissão entre administradores, funcionários e usuários.
RF6	<b>Suporte à Inteligência Artificial</b>	Utilizar algoritmos de inteligência artificial para melhorar a precisão da análise e correção de scripts.
RF7	<b>Diferença de Acesso entre Usuários</b>	Garantir que os diferentes tipos de usuários (administradores, funcionários, usuários) tenham acesso apenas às funcionalidades pertinentes a seus papéis.
RF8	<b>Interface Web</b>	Fornecer uma interface web intuitiva e amigável para que os usuários possam interagir com o sistema de maneira eficiente
RF9	<b>Modelo Offiline</b>	Oferecer um modelo offline da aplicação

### 3.2 Requisitos Não Funcionais

<i>Requisitos Não-Funcionais</i>		
Código	Nome	Descrição
RNF1	<b>Eficiência do sistema</b>	O sistema deve ser capaz de realizar a análise e correção de scripts de forma rápida e eficiente, minimizando o tempo de espera do usuário.
RNF2	<b>Segurança e privacidade</b>	Implementar medidas robustas para garantir a segurança e privacidade dos dados dos usuários, especialmente ao utilizar APIs externas
RNF3	<b>Escalabilidade</b>	O sistema deve ser escalável para suportar um número crescente de usuários e submissões de scripts sem degradação de performance.
RNF4	<b>Recursividade:</b>	O sistema deve ser capaz de processar scripts que contenham chamadas recursivas sem comprometer a análise e correção
RNF5	<b>Perfomance</b>	Garantir que o sistema mantenha um alto nível de performance, mesmo sob cargas elevadas de trabalho
RNF6	<b>Segurança</b>	Implementar mecanismos de segurança para proteger o sistema contra acessos não autorizados e ataques cibernéticos.

## 4.0 Diagramas

### 4.1 Identificação dos Casos de Uso

Identificação dos Casos de Uso	
<b>Caso de uso</b>	Registrar
<b>Ator</b>	Usuário
<b>Descrição</b>	O usuário irá acessar a interface de registro do sistema e inserir suas informações pessoais (nome, e-mail, senha, etc.). O sistema valida as informações e cria uma nova conta para o usuário.
<b>Pré-Condição</b>	O usuário precisa estar cadastrado no sistema.
<b>Pós-Condições</b>	O usuários está autenticado/logado no sistema
Ações	
Ator	Sistema
<ul style="list-style-type: none"> <li>- Inserir nome.</li> <li>- Inserir e-mail.</li> <li>- Inserir senha.</li> </ul>	<ul style="list-style-type: none"> <li>- Validar informações.</li> <li>- Exibir mensagem de erro se necessário.</li> <li>- Criar nova conta de usuário.</li> </ul>
Fluxo Principal	Fluxo Alternativo
<ol style="list-style-type: none"> <li>1. Usuário acessa a página de registro.</li> <li>2. Usuário preenche os campos de nome, e-mail e senha.</li> <li>3. O sistema valida as informações inseridas.</li> <li>4. O sistema cria a nova conta de usuário.</li> </ol>	<ol style="list-style-type: none"> <li>1. O sistema exibe uma mensagem de erro se as informações inseridas forem inválidas ou se o e-mail já estiver cadastrado.</li> </ol>



Identificação dos Casos de Uso	
<b>Caso de uso</b>	Logar
<b>Ator</b>	Usuário
<b>Descrição</b>	O usuário acessa a interface de login do sistema e insere seu e-mail e senha. O sistema valida as informações e concede o acesso.
<b>Pré-Condição</b>	O usuário precisa estar cadastrado no sistema.
<b>Pós-Condições</b>	O usuário está autenticado/logado no sistema.
Ações	
Ator	Sistema
<ul style="list-style-type: none"> <li>• Inserir e-mail.</li> <li>• Inserir senha.;</li> </ul>	<ul style="list-style-type: none"> <li>• Validar informações.</li> <li>• Exibir mensagem de erro se necessário.</li> <li>• Conceder acesso ao usuário.</li> </ul>
Fluxo Principal	Fluxo Alternativo
<ol style="list-style-type: none"> <li>1. Usuário acessa a página de login.</li> <li>2. Usuário preenche os campos de e-mail e senha.</li> <li>3. O sistema valida as informações inseridas.</li> <li>4. O sistema concede acesso ao usuário.</li> </ol>	<ol style="list-style-type: none"> <li>1. O sistema exibe uma mensagem de erro se as informações inseridas forem inválidas ou se o usuário não estiver cadastrado.</li> </ol>

Identificação dos Casos de Uso	
<b>Caso de uso</b>	Submeter Scripts
<b>Ator</b>	Usuário
<b>Descrição</b>	O usuário insere um script ou o importa. O sistema recebe o script para análise.
<b>Pré-Condição</b>	O usuário deve estar logado no sistema
<b>Pós-Condições</b>	O script do usuário é submetido e aguardando análise.
Ações	
Ator	Sistema
<ul style="list-style-type: none"> <li>• Selecionar questão.</li> <li>• Inserir ou fazer upload do script.</li> </ul>	<ul style="list-style-type: none"> <li>• Recebe script.</li> <li>• Confirma submissão.</li> </ul>
Fluxo Principal	Fluxo Alternativo
<ol style="list-style-type: none"> <li>1. Usuário insere ou faz upload do script.</li> <li>2. O sistema recebe o script.</li> <li>3. O sistema confirma a submissão do script.</li> </ol>	

Identificação dos Casos de Uso	
<b>Caso de uso</b>	Visualizar Feedback e correções
<b>Ator</b>	Usuário
<b>Descrição</b>	Após submeter o script, o usuário pode visualizar vários quadros de correções do script fornecidos pelo sistema, cada um detalhando diferentes aspectos dos erros encontrados e sugestões de melhoria.
<b>Pré-Condição</b>	O usuário deve estar logado no sistema e ter submetido um script.
<b>Pós-Condições</b>	O usuário visualiza um conjunto de quadros de correções fornecidos pelo sistema para testá-los ou alterá-los
Ações	
Ator	Sistema
<ul style="list-style-type: none"> <li>• Submeter script.</li> <li>• Visualizar quadros de correções.</li> </ul>	<ul style="list-style-type: none"> <li>• Analisar o script.</li> <li>• Gerar quadros de correções detalhados com feedback.</li> </ul>
Fluxo Principal	Fluxo Alternativo
<ol style="list-style-type: none"> <li>1. O sistema gera vários quadros de correções.</li> <li>2. O usuário visualiza os quadros de correções.</li> </ol>	<ol style="list-style-type: none"> <li>1. O sistema exibe uma mensagem de erro caso não for possível corrigir o script</li> </ol>

Identificação dos Casos de Uso	
<b>Caso de uso</b>	Testar Script Corrigido
<b>Ator</b>	Usuário
<b>Descrição</b>	Após submeter um script e receber as correções, o usuário pode executar testes no script corrigido para garantir que as correções aplicadas resolveram os problemas e que o script está funcionando corretamente
<b>Pré-Condição</b>	O usuário deve estar logado no sistema e ter recebido as correções para um script previamente submetido.
<b>Pós-Condições</b>	O script é testado e o usuário recebe um relatório detalhado sobre o funcionamento do script após as correções.
Ações	
Ator	Sistema
<ul style="list-style-type: none"> <li>• Selecionar o script corrigido.</li> <li>• Iniciar o teste do script corrigido.</li> </ul>	<ul style="list-style-type: none"> <li>• Gerar comprovante de empréstimo</li> <li>• Executar testes no script corrigido.</li> <li>• Gerar relatório detalhado dos resultados dos testes.</li> <li>• Exibir resultados para o usuário.</li> </ul>
Fluxo Principal	Fluxo Alternativo
<ol style="list-style-type: none"> <li>1. Usuário acessa a lista de scripts corrigidos.</li> <li>2. Usuário seleciona o script corrigido que deseja testar.</li> <li>3. Usuário inicia o teste do script corrigido.</li> <li>4. O sistema executa os testes automatizados no script corrigido.</li> <li>5. O sistema gera um relatório detalhado dos resultados dos testes.</li> <li>6. O sistema exibe os resultados dos testes para o usuário.</li> </ol>	O sistema exibe uma mensagem de erro se o teste não puder ser executado.

Identificação dos Casos de Uso	
<b>Caso de uso</b>	Gerenciar Banco de Scripts
<b>Ator</b>	Administrador
<b>Descrição</b>	O administrador pode visualizar, adicionar, editar ou remover scripts do banco de dados do sistema.
<b>Pré-Condição</b>	O administrador deve estar logado no sistema.
<b>Pós-Condições</b>	O banco de scripts é atualizado conforme as ações do administrador
Ações	
Ator	Sistema
<ul style="list-style-type: none"> <li>• Visualizar banco de scripts.</li> <li>• Adicionar/Editar/Remover scripts.</li> </ul>	<ul style="list-style-type: none"> <li>• Atualizar banco de scripts</li> </ul>
Fluxo Principal	Fluxo Alternativo
<ol style="list-style-type: none"> <li>1. Administrador acessa a página de gerenciamento de scripts.</li> <li>2. Administrador visualiza o banco de scripts.</li> <li>3. Administrador adiciona, edita ou remove scripts.</li> <li>4. Sistema atualiza o banco de scripts.</li> </ol>	

Identificação dos Casos de Uso	
<b>Caso de uso</b>	Configurar Integrações
<b>Ator</b>	Administrador
<b>Descrição</b>	O administrador pode configurar a integração do sistema com APIs externas ou modelos internos para a análise e correção de scripts.
<b>Pré-Condição</b>	O administrador deve estar logado no sistema.
<b>Pós-Condições</b>	As integrações com APIs ou modelos internos são configuradas e ativas.
Fluxo	
Ator	Sistema
<ul style="list-style-type: none"> <li>• Acessar configurações de integração.</li> <li>• Definir parâmetros de integração.</li> </ul>	<ul style="list-style-type: none"> <li>• Atualizar configurações de integração</li> </ul>
Fluxo Principal	Fluxo Alternativo
<ol style="list-style-type: none"> <li>1. Administrador acessa a página de configurações de integração.</li> <li>2. Administrador define os parâmetros de integração.</li> <li>3. Sistema atualiza as configurações de integração.</li> </ol>	

Identificação dos Casos de Uso	
<b>Caso de uso</b>	Gerenciar Usuários e Permissões
<b>Ator</b>	Administrador
<b>Descrição</b>	O administrador pode adicionar, editar ou remover usuários e definir suas permissões de acesso
<b>Pré-Condição</b>	O administrador deve estar logado no sistema.
<b>Pós-Condições</b>	Os usuários e suas permissões de acesso são atualizados conforme as ações do administrador.
Ações	
Ator	Sistema
<ul style="list-style-type: none"> <li>• Acessar gerenciamento de usuários.</li> <li>• Adicionar/Editar/Remover usuários.</li> <li>• Definir permissões de acesso.</li> </ul>	<ul style="list-style-type: none"> <li>• Atualizar banco de usuários e permissões.</li> </ul>
Fluxo Principal	Fluxo Alternativo
<ol style="list-style-type: none"> <li>1. Administrador acessa a página de gerenciamento de usuários.</li> <li>2. Administrador adiciona, edita ou remove usuários.</li> <li>3. Administrador define permissões de acesso.</li> <li>4. Sistema atualiza o banco de usuários e permissões.</li> </ol>	

Identificação dos Casos de Uso	
<b>Caso de uso</b>	Cadastrar Scripts Submetidos por Usuários
<b>Ator</b>	Funcionário
<b>Descrição</b>	O funcionário cadastra scripts submetidos pelos usuários no banco de dados do sistema para análise posterior..
<b>Pré-Condição</b>	O funcionário deve estar logado no sistema.
<b>Pós-Condições</b>	Os scripts submetidos pelos usuários são cadastrados no banco de dados.
Ações	
Ator	Sistema
<ul style="list-style-type: none"> <li>• Visualizar scripts submetidos.</li> <li>• Cadastrar scripts no banco de dados</li> </ul>	<ul style="list-style-type: none"> <li>• Atualizar banco de dados com novos scripts.</li> </ul>
Fluxo Principal	Fluxo Alternativo
<ol style="list-style-type: none"> <li>1. Funcionário acessa a página de scripts submetidos.</li> <li>2. Funcionário visualiza os scripts submetidos.</li> <li>3. Funcionário cadastra scripts no banco de dados.</li> <li>4. Sistema atualiza o banco de dados com os novos scripts.</li> </ol>	

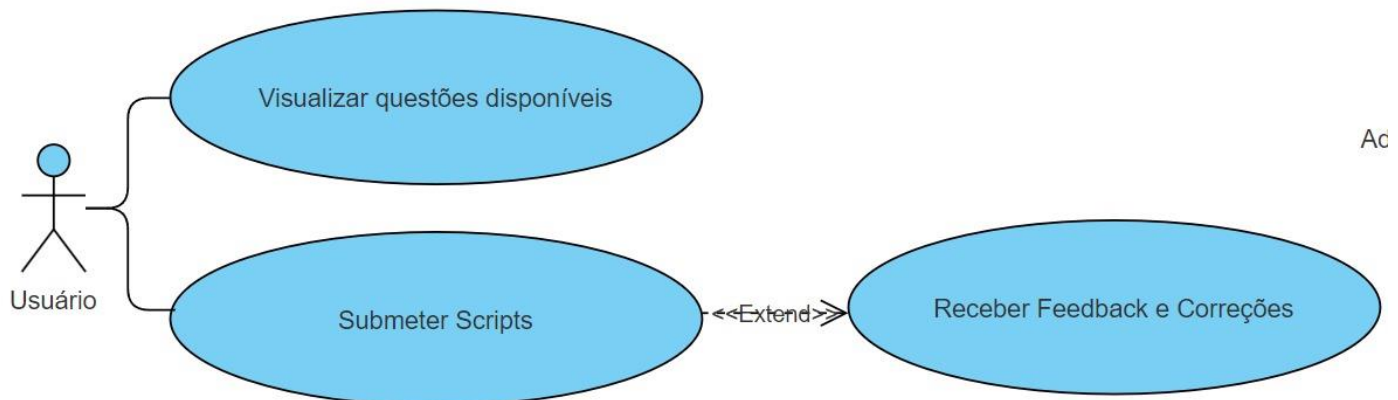
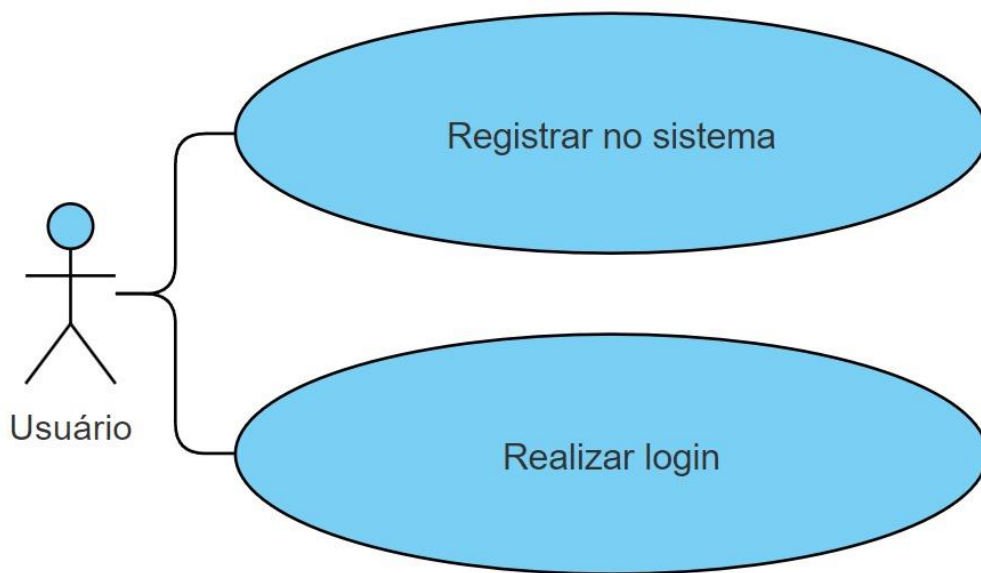
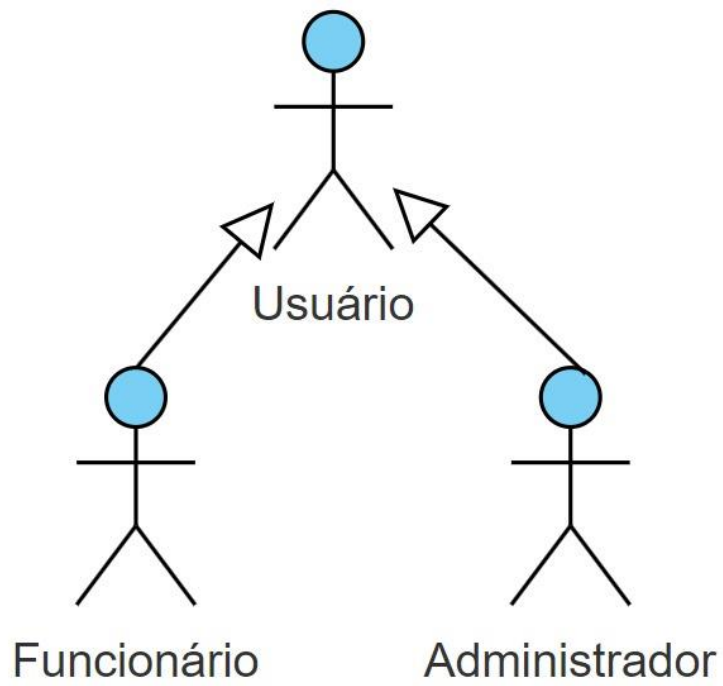


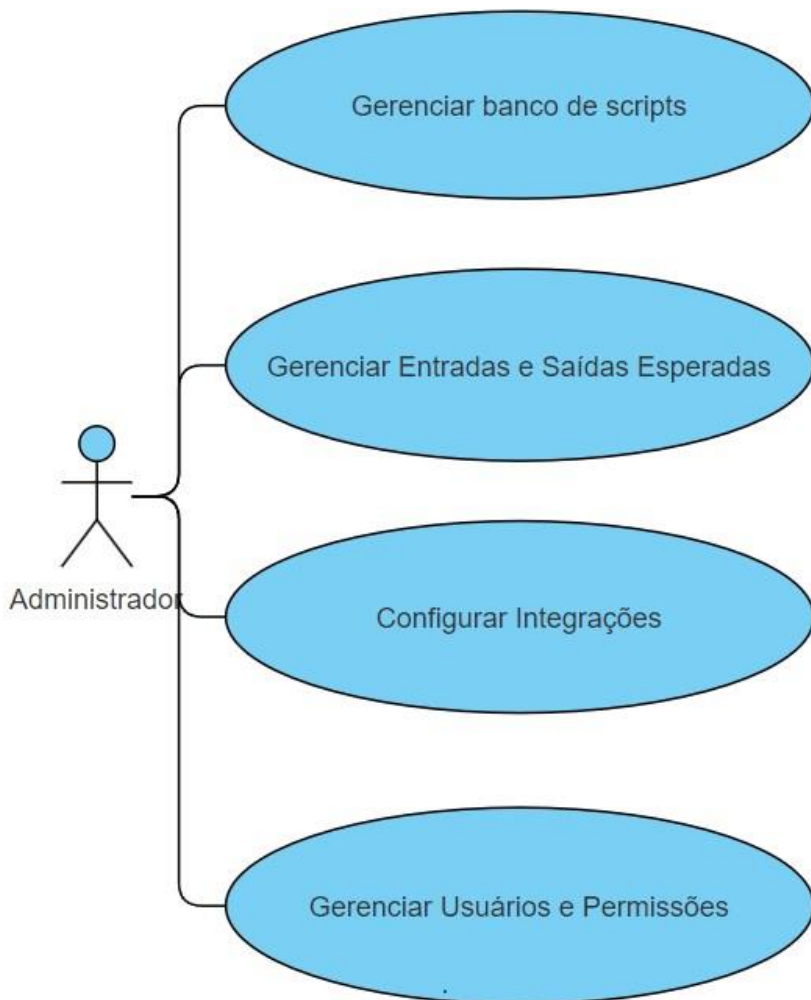
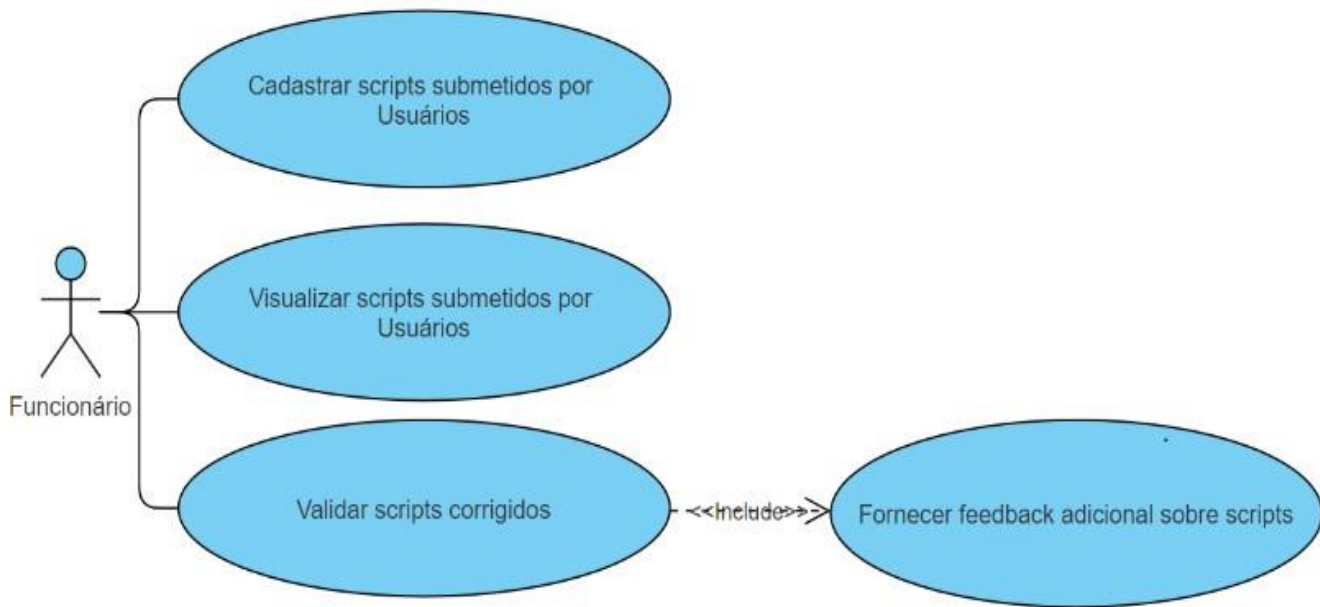
Identificação dos Casos de Uso	
<b>Caso de uso</b>	Visualizar Scripts Submetidos por Usuários
<b>Ator</b>	Funcionário
<b>Descrição</b>	O funcionário pode visualizar os scripts submetidos pelos usuários para revisão e análise.
<b>Pré-Condição</b>	O funcionário deve estar logado no sistema.
<b>Pós-Condições</b>	Os funcionários têm o poder de validar os scripts
Ações	
Ator	Sistema
<ul style="list-style-type: none"> <li>• Acessar scripts submetidos.</li> <li>• Visualizar detalhes de scripts.</li> </ul>	<ul style="list-style-type: none"> <li>• Exibir scripts submetidos</li> </ul>
Fluxo Principal	Fluxo Alternativo
<ol style="list-style-type: none"> <li>1. Funcionário acessa a página de scripts submetidos.</li> <li>2. Funcionário visualiza os detalhes dos scripts submetidos.</li> </ol>	

Identificação dos Casos de Uso	
<b>Caso de uso</b>	Validar Scripts Corrigidos
<b>Ator</b>	Funcionário
<b>Descrição</b>	O funcionário valida os scripts corrigidos para garantir que as correções estão corretas e o script está funcional
<b>Pré-Condição</b>	O funcionário deve estar logado no sistema e ter acesso aos scripts submetidos.
<b>Pós-Condições</b>	Os scripts corrigidos são validados e prontos para uso.
Fluxo	
Ator	Sistema
<ul style="list-style-type: none"> <li>• Visualizar scripts corrigidos.</li> <li>• Validar correções aplicadas.</li> </ul>	<ul style="list-style-type: none"> <li>• Atualizar status dos scripts validados.</li> </ul>
Fluxo Principal	Fluxo Alternativo
<ol style="list-style-type: none"> <li>1. Funcionário acessa a página de scripts corrigidos.</li> <li>2. Funcionário visualiza os scripts corrigidos.</li> <li>3. Funcionário valida as correções aplicadas.</li> <li>4. Sistema atualiza o status dos scripts validados.</li> </ol>	

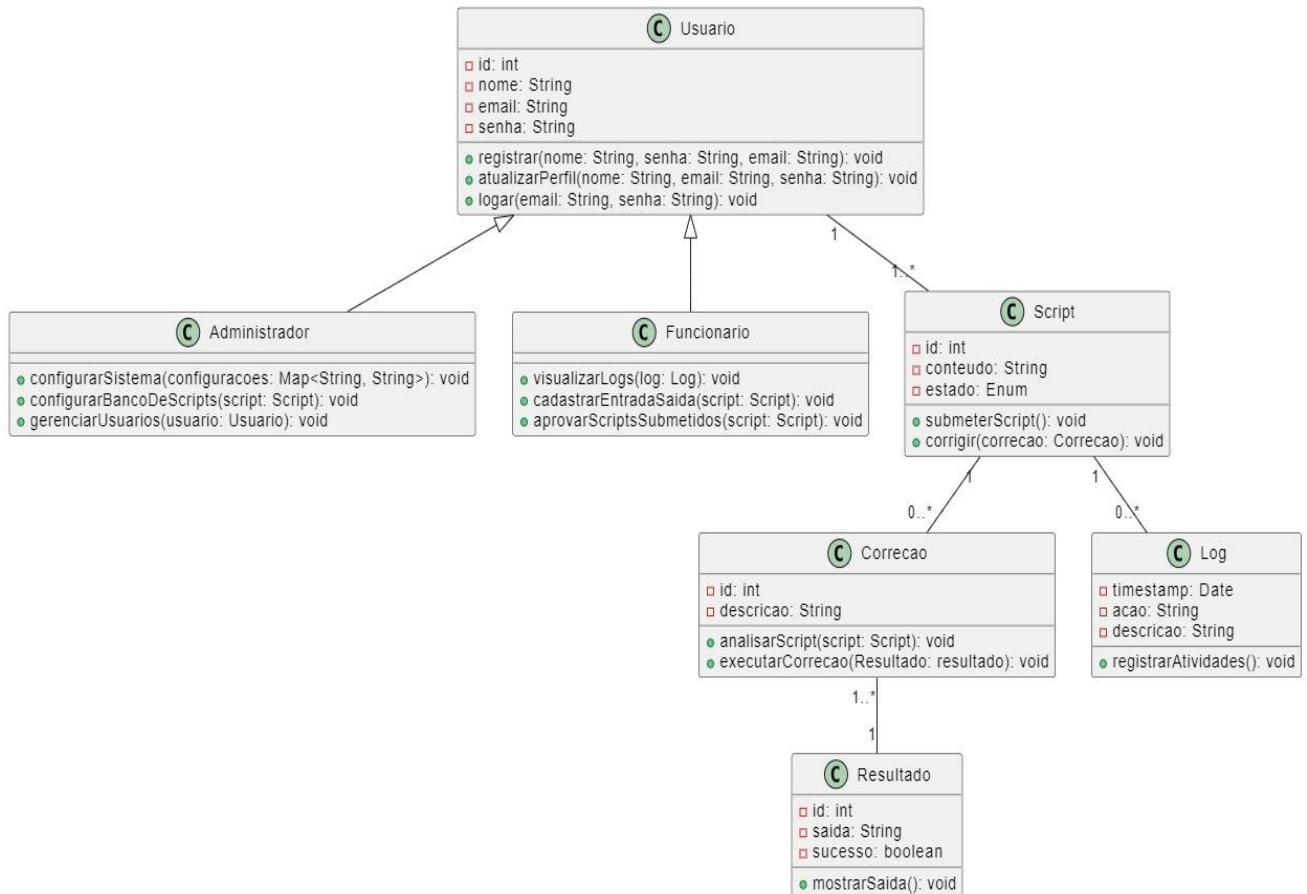
Identificação dos Casos de Uso	
<b>Caso de uso</b>	Fornecer Feedback Adicional sobre Scripts
<b>Ator</b>	Funcionário
<b>Descrição</b>	O funcionário pode fornecer feedback adicional sobre os scripts submetidos, complementando as correções automáticas.
<b>Pré-Condição</b>	O funcionário deve estar logado no sistema e ter visualizado os scripts submetidos.
<b>Pós-Condições</b>	Feedback adicional é fornecido aos usuários para aprimoramento dos scripts.
Fluxo	
Ator	Sistema
<ul style="list-style-type: none"> <li>• Visualizar scripts submetidos.</li> <li>• Fornecer feedback adicional.</li> </ul>	<ul style="list-style-type: none"> <li>• Atualizar scripts com feedback adicional</li> </ul>
Fluxo Principal	Fluxo Alternativo
<ol style="list-style-type: none"> <li>1. Funcionário acessa a página de scripts submetidos.</li> <li>2. Funcionário visualiza os scripts submetidos.</li> <li>3. Funcionário fornece feedback adicional.</li> <li>4. Sistema atualiza os scripts com o feedback adicional.</li> </ol>	

## 4.2 Modelagem de Casos de Uso





## 4.3 Modelagem de Classes



O diagrama de classe apresentado mostra a estrutura do sistema "Script Inteligente" e as interações entre as diferentes classes. Aqui está uma descrição detalhada dos atributos, métodos e suas interações:

### Classes:

#### Usuário

##### ○ Atributos:

- id: int: Identificador único do usuário.
- nome: String: Nome do usuário.
- email: String: E-mail do usuário.
- senha: String: Senha do usuário.

##### ○ Métodos:

- registrar(nome: String, senha: String, email: String): void: Método para registrar um novo usuário no sistema.
- atualizarPerfil(nome: String, email: String, senha: String): void: Método para atualizar as informações do perfil do usuário.
- login(email: String, senha: String): void: Método para o usuário efetuar login no sistema.

## Administrador

- **Métodos:**

- configurarSistema(configuracoes: Map<String, String>): void: Método para configurar as definições do sistema.
- configurarBancoDeScripts(script: Script): void: Método para gerenciar o banco de scripts.
- gerenciarUsuarios(usuario: Usuario): void: Método para gerenciar os usuários do sistema.

## Funcionário

- **Métodos:**

- visualizarLogs(log: Log): void: Método para visualizar logs de atividades do sistema.
- cadastrarEntradaSaida(script: Script): void: Método para cadastrar entradas e saídas esperadas para um script.
- aprovarScriptsSubmetidos(script: Script): void: Método para aprovar scripts submetidos pelos usuários.

## Script

- **Atributos:**

- id: int: Identificador único do script.
- conteudo: String: Conteúdo do script.
- estado: Enum: Estado atual do script (submetido, corrigido, etc.).

- **Métodos:**

- submeterScript(): void: Método para submeter um script para análise.
- corrigir(correcao: Correcao): void: Método para aplicar correções ao script.

## Correção

- **Atributos:**

- id: int: Identificador único da correção.
- descricao: String: Descrição da correção.

- **Métodos:**

- analisarScript(script: Script): void: Método para analisar um script.
- executarCorrecao(resultado: Resultado): void: Método para executar a correção de um script.

## Log

- **Atributos:**

- timestamp: Date: Data e hora da atividade registrada.
- acao: String: Descrição da ação realizada.
- descricao: String: Descrição detalhada da atividade.

- **Métodos:**

- registrarAtividades(): void: Método para registrar atividades no sistema.

## Resultado

- **Atributos:**

- id: int: Identificador único do resultado.
- saida: String: Saída gerada após a execução do script.
- sucesso: boolean: Indica se a execução foi bem-sucedida.

- **Métodos:**

- mostrarSaida(): void: Método para exibir a saída gerada após a execução do script.

### *Interações Entre as Classes:*

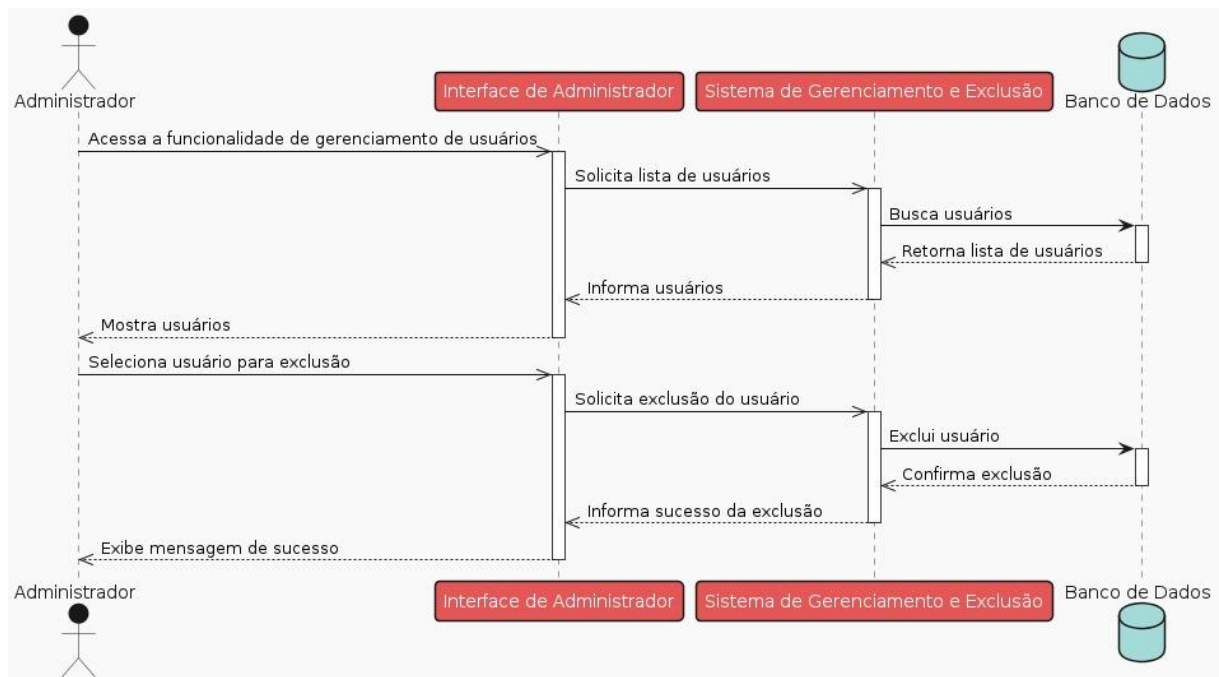
- **Usuário:**
  - Pode registrar-se, atualizar seu perfil e efetuar login no sistema.
  - Relaciona-se com a classe Script ao submeter scripts para análise.
- **Administrador:**
  - Configura o sistema, gerencia o banco de scripts e gerencia os usuários.
  - Tem acesso total ao sistema, podendo interagir com todas as classes principais.
- **Funcionário:**
  - Visualiza logs, cadastra entradas e saídas e aprova scripts submetidos.
  - Atua como um intermediário entre os usuários e o sistema, garantindo a qualidade das submissões.
- **Script:**
  - Representa o código submetido pelo usuário.
  - Interage com a correção para receber análise e aplicar correções.
- **Correção:**
  - Analisa os scripts e aplica correções.
  - Gera resultados que indicam o sucesso ou falha das correções aplicadas.
- **Log:**
  - Registra atividades no sistema, como submissões de scripts e correções aplicadas.
  - Pode ser visualizado por funcionários para monitorar o histórico de atividades.
- **Resultado:**
  - Armazena a saída gerada após a execução de um script corrigido.
  - Indica se a execução do script foi bem-sucedida e mostra a saída correspondente.

### *Resumo das Interações:*

5. **Usuário** submete um **Script**.
6. O **Script** é analisado e corrigido pela **Correção**.
7. A **Correção** gera um **Resultado**, indicando o sucesso ou falha da execução do script corrigido.
8. As atividades são registradas no **Log**.
9. **Funcionário** pode visualizar os **Logs** e cadastrar entradas e saídas esperadas.
10. **Administrador** gerencia configurações do sistema, scripts e usuários



## 4.4 Modelagem de Sequências



Os diagramas de sequência fornecem uma visão detalhada da interação entre diferentes componentes do sistema ao longo do tempo. A seguir, a explicação de cada diagrama:

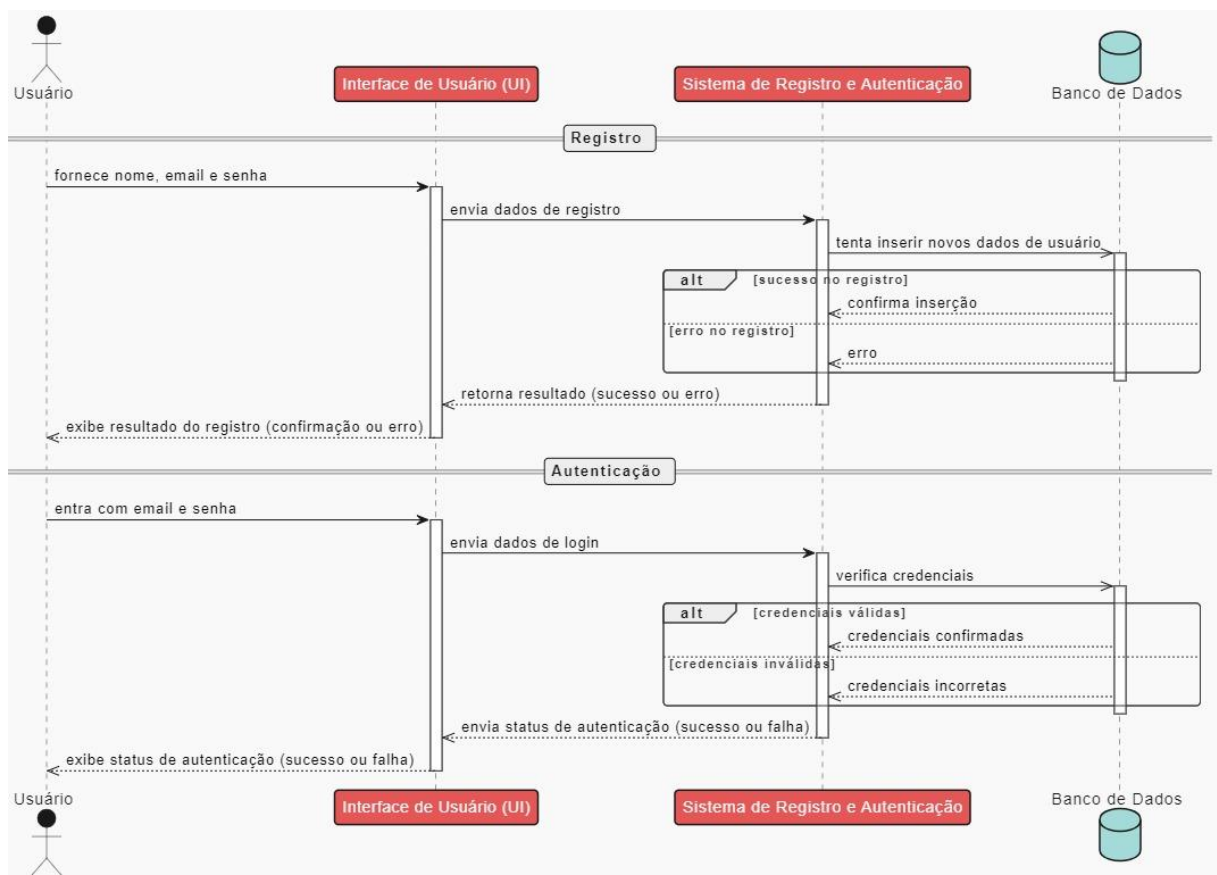
### *Diagrama 1: Exclusão de Usuário pelo Administrador*

**Descrição Geral:** Este diagrama mostra o processo pelo qual um administrador gerencia e exclui usuários do sistema.

#### **Passos:**

1. **Administrador** acessa a funcionalidade de gerenciamento de usuários através da **Interface de Administrador**.
2. A **Interface de Administrador** solicita a lista de usuários ao **Sistema de Gerenciamento e Exclusão**.
3. O **Sistema de Gerenciamento e Exclusão** consulta o **Banco de Dados** para buscar a lista de usuários.
4. O **Banco de Dados** retorna a lista de usuários ao **Sistema de Gerenciamento e Exclusão**.
5. O **Sistema de Gerenciamento e Exclusão** informa a lista de usuários à **Interface de Administrador**.
6. A **Interface de Administrador** mostra os usuários ao **Administrador**.
7. O **Administrador** seleciona o usuário a ser excluído.
8. A **Interface de Administrador** solicita a exclusão do usuário ao **Sistema de Gerenciamento e Exclusão**.

9. O **Sistema de Gerenciamento e Exclusão** exclui o usuário do **Banco de Dados**.
10. O **Banco de Dados** confirma a exclusão ao **Sistema de Gerenciamento e Exclusão**.
11. O **Sistema de Gerenciamento e Exclusão** informa o sucesso da exclusão à **Interface de Administrador**.
12. A **Interface de Administrador** exibe uma mensagem de sucesso ao **Administrador**.



**Diagrama: Registro e Autenticação de Usuário**

## Parte 1: Registro

**Descrição Geral:** Este diagrama mostra o processo de registro de um novo usuário no sistema.

**Passos:**

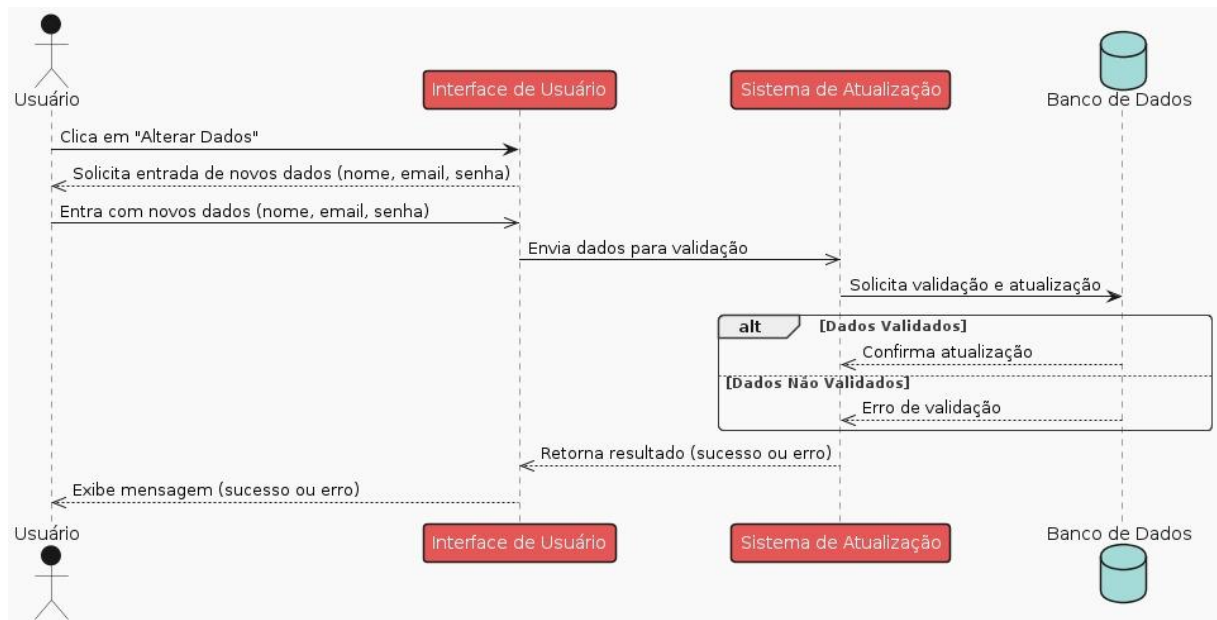
1. **Usuário** fornece nome, e-mail e senha através da **Interface de Usuário (UI)**.
2. A **Interface de Usuário (UI)** envia os dados de registro ao **Sistema de Registro e Autenticação**.
3. O **Sistema de Registro e Autenticação** tenta inserir novos dados de usuário no **Banco de Dados**.
4. O **Banco de Dados**:
  - **Alternativa 1 (Sucesso)**: Confirma a inserção.
  - **Alternativa 2 (Erro)**: Informa erro no registro.
5. O **Sistema de Registro e Autenticação** retorna o resultado (sucesso ou erro) à **Interface de Usuário (UI)**.
6. A **Interface de Usuário (UI)** exibe o resultado do registro ao **Usuário**.

## **Parte 2: Autenticação**

**Descrição Geral:** Este diagrama mostra o processo de autenticação de um usuário no sistema.

### **Passos:**

1. **Usuário** entra com e-mail e senha na **Interface de Usuário (UI)**.
2. A **Interface de Usuário (UI)** envia os dados de login ao **Sistema de Registro e Autenticação**.
3. O **Sistema de Registro e Autenticação** verifica as credenciais no **Banco de Dados**.
4. O **Banco de Dados**:
  - **Alternativa 1 (Credenciais Válidas)**: Confirma as credenciais.
  - **Alternativa 2 (Credenciais Inválidas)**: Informa erro de credenciais.
5. O **Sistema de Registro e Autenticação** envia o status de autenticação (sucesso ou falha) à **Interface de Usuário (UI)**.
6. A **Interface de Usuário (UI)** exibe o status de autenticação ao **Usuário**.



**Diagrama: Atualização de Dados do Usuário**

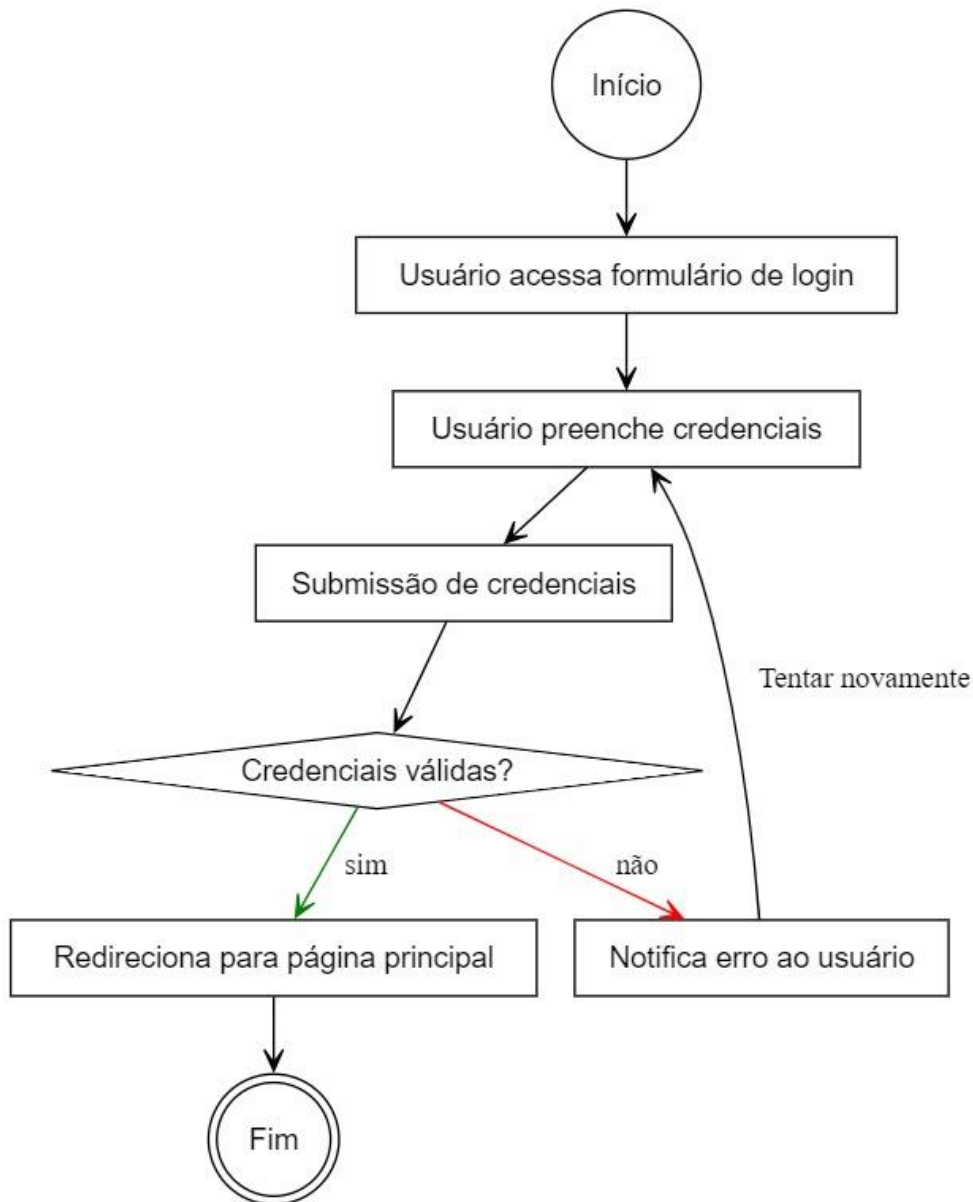
**Descrição Geral:** Este diagrama mostra o processo de atualização dos dados do usuário no sistema.

**Passos:**

1. **Usuário** clica em "Alterar Dados" na **Interface de Usuário**.
2. A **Interface de Usuário** solicita a entrada de novos dados (nome, e-mail, senha).
3. **Usuário** entra com os novos dados (nome, e-mail, senha) na **Interface de Usuário**.
4. A **Interface de Usuário** envia os dados para validação ao **Sistema de Atualização**.
5. O **Sistema de Atualização** solicita a validação e atualização no **Banco de Dados**.
6. O **Banco de Dados**:
  - **Alt1 (Dados Validados)**: Confirma a atualização.
  - **Alt2 (Dados Não Validados)**: Informa erro de validação.
7. O **Sistema de Atualização** retorna o resultado (sucesso ou erro) à **Interface de Usuário**.
8. A **Interface de Usuário** exibe uma mensagem de sucesso ou erro ao **Usuário**.

## 4.5 Modelagem de Atividades

Os diagramas de atividades fornecem uma visão sobre o fluxo de ações e decisões em diferentes processos do sistema. A seguir, a explicação de cada diagrama:



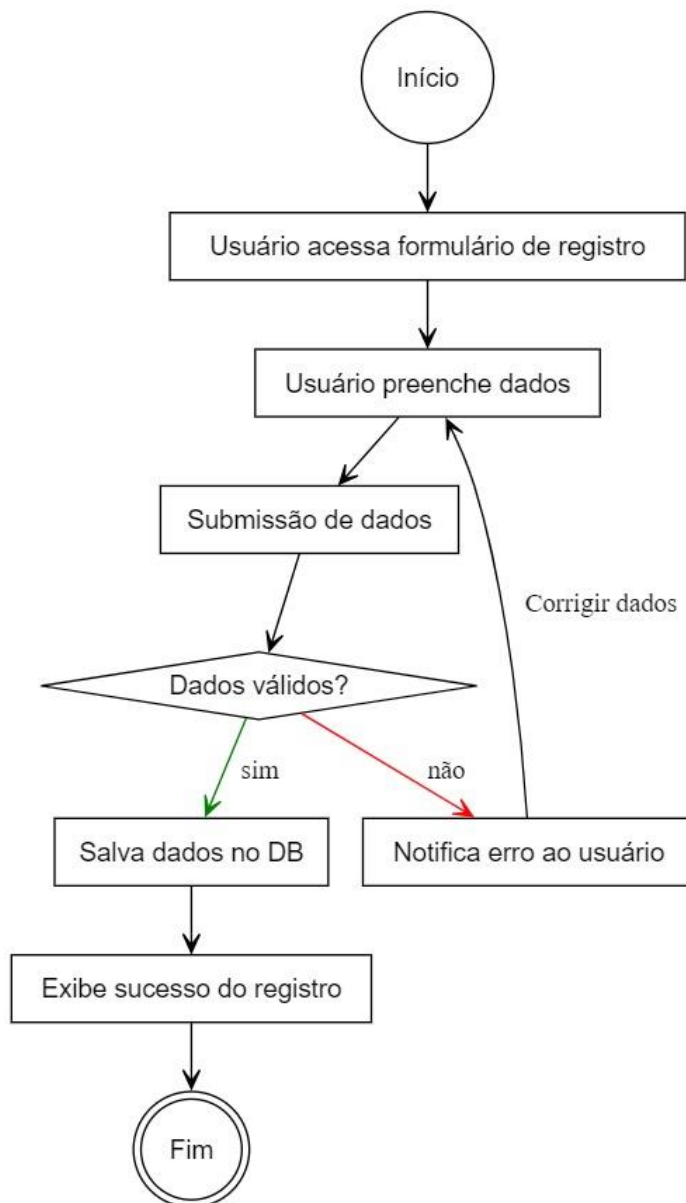
### *Diagrama: Processo de Login*

**Descrição Geral:** Este diagrama ilustra o fluxo de atividades para o processo de login de um usuário no sistema.

#### **Passos:**

1. **Início:** O processo começa.
2. **Usuário acessa formulário de login:** O usuário navega até a página de login.

3. **Usuário preenche credenciais:** O usuário insere seu e-mail e senha.
4. **Submissão de credenciais:** O usuário envia as credenciais para validação.
5. **Credenciais válidas?:** O sistema verifica se as credenciais são corretas.
  - **Sim:** Se as credenciais são válidas, o sistema redireciona para a página principal.
  - **Não:** Se as credenciais são inválidas, o sistema notifica o erro ao usuário e o usuário tenta novamente.
6. **Fim:** O processo termina.

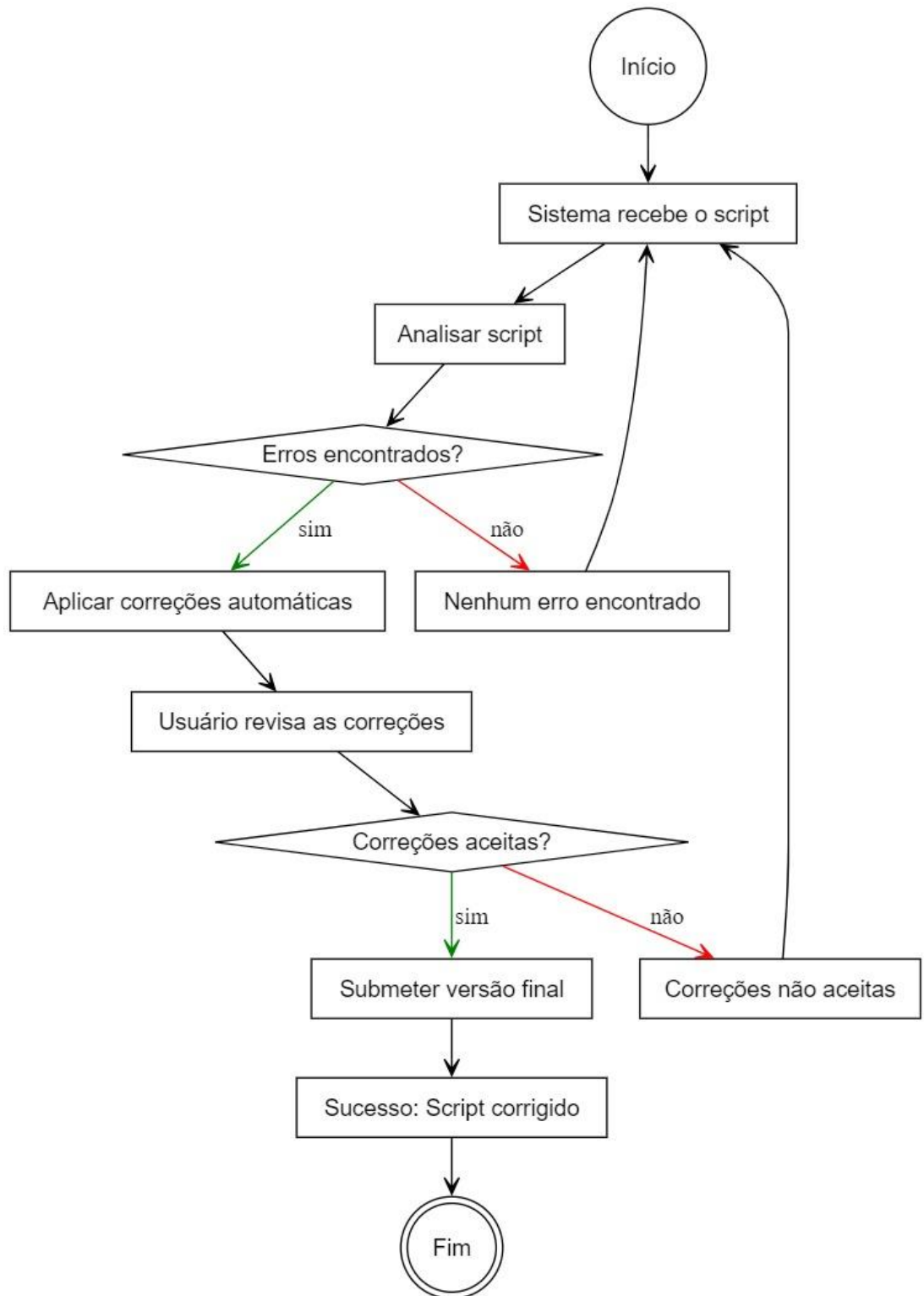


### *Diagrama: Processo de Registro*

**Descrição Geral:** Este diagrama mostra o fluxo de atividades para o registro de um novo usuário no sistema.

**Passos:**

1. **Início:** O processo começa.
2. **Usuário acessa formulário de registro:** O usuário navega até a página de registro.
3. **Usuário preenche dados:** O usuário insere seu nome, e-mail e senha.
4. **Submissão de dados:** O usuário envia os dados para validação.
5. **Dados válidos?:** O sistema verifica se os dados são válidos.
  - **Sim:** Se os dados são válidos, o sistema salva os dados no banco de dados e exibe o sucesso do registro.
  - **Não:** Se os dados são inválidos, o sistema notifica o erro ao usuário e o usuário corrige os dados.
6. **Fim:** O processo termina.



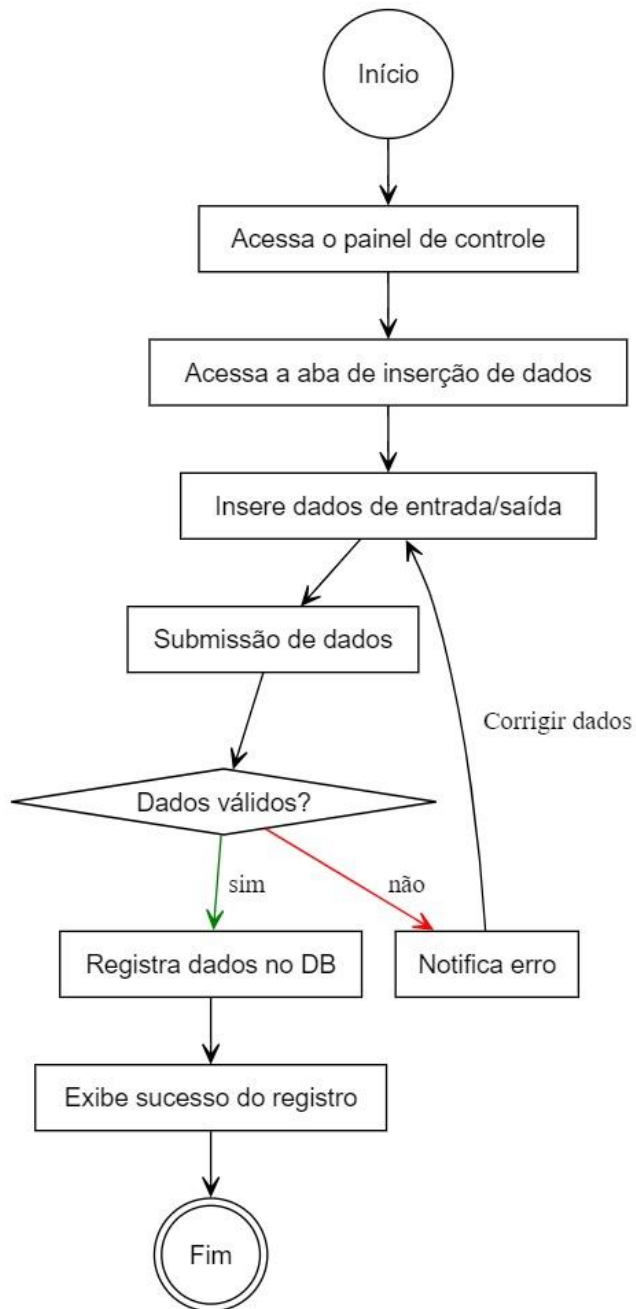


### ***Diagrama: Processo de Correção de Script***

**Descrição Geral:** Este diagrama ilustra o fluxo de atividades para a correção de um script submetido pelo usuário.

**Passos:**

1. **Início:** O processo começa.
2. **Sistema recebe o script:** O script é submetido pelo usuário e recebido pelo sistema.
3. **Analisar script:** O sistema analisa o script.
4. **Erros encontrados?:** O sistema verifica se há erros no script.
  - **Sim:** Se erros são encontrados, o sistema aplica correções automáticas.
  - **Não:** Se nenhum erro é encontrado, o processo termina.
5. **Usuário revisa as correções:** O usuário revisa as correções aplicadas pelo sistema.
6. **Correções aceitas?:** O usuário decide se aceita ou não as correções.
  - **Sim:** Se as correções são aceitas, o usuário submete a versão final e o script corrigido é considerado um sucesso.
  - **Não:** Se as correções não são aceitas, o processo termina.
7. **Fim:** O processo termina.



### **Diagrama: Processo de Inserção de Dados**

**Descrição Geral:** Este diagrama mostra o fluxo de atividades para a inserção de dados de entrada e saída no sistema pelo administrador.

#### **Passos:**

1. **Início:** O processo começa.
2. **Acessa o painel de controle:** O administrador navega até o painel de controle.

3. **Acessa a aba de inserção de dados:** O administrador acessa a seção para inserir dados de entrada e saída.
4. **Insere dados de entrada/saída:** O administrador insere os dados necessários.
5. **Submissão de dados:** O administrador envia os dados para validação.
6. **Dados válidos?:** O sistema verifica se os dados são válidos.
  - **Sim:** Se os dados são válidos, o sistema registra os dados no banco de dados e exibe o sucesso do registro.
  - **Não:** Se os dados são inválidos, o sistema notifica o erro ao administrador e o administrador corrige os dados.
7. **Fim:** O processo termina.

@autor: Giordano Bruno Mochel e Vitor Ferreira Nunes

@contato:

@data última versão: 21 / 06 / 2024

@versão: 1.1

@outros repositórios:

@Agradecimentos: Universidade Federal do Maranhão (UFMA), Professor Doutor Thales Levi Azevedo Valente, e colegas de curso.

@Copyright/License

Este material é resultado de um trabalho acadêmico para a disciplina PROJETO E DESENVOLVIMENTO DE SOFTWARE, sobre a orientação do professor Dr. THALES LEVI AZEVEDO VALENTE, semestre letivo 2024.1, curso Engenharia da Computação, na Universidade Federal do Maranhão (UFMA). Todo o material sob esta licença é software livre: pode ser usado para fins acadêmicos e comerciais sem nenhum custo. Não há papelada, nem royalties, nem restrições de "copyleft" do tipo GNU. Ele é licenciado sob os termos da licença MIT reproduzida abaixo e, portanto, é compatível com GPL e também se qualifica como software de código aberto. É de domínio público. Os detalhes legais estão abaixo. O espírito desta licença é que você é livre para usar este material para qualquer finalidade, sem nenhum custo. O único requisito é que, se você usá-los, nos dê crédito.

Copyright © 2024 Educational Material

Este material está licenciado sob a Licença MIT. É permitido o uso, cópia, modificação, e distribuição deste material para qualquer fim, desde que acompanhado deste aviso de direitos autorais.

O MATERIAL É FORNECIDO "COMO ESTÁ", SEM GARANTIA DE QUALQUER TIPO, EXPRESSA OU IMPLÍCITA, INCLUINDO, MAS NÃO SE LIMITANDO ÀS GARANTIAS DE COMERCIALIZAÇÃO, ADEQUAÇÃO A UM DETERMINADO FIM E NÃO VIOLAÇÃO. EM HIPÓTESE ALGUMA OS AUTORES OU DETENTORES DE DIREITOS AUTORAIS SERÃO RESPONSÁVEIS POR QUALQUER RECLAMAÇÃO, DANOS OU OUTRA RESPONSABILIDADE, SEJA EM UMA AÇÃO DE CONTRATO, ATO ILÍCITO OU DE OUTRA FORMA, DECORRENTE DE, OU EM CONEXÃO COM O MATERIAL OU O USO OU OUTRAS NEGOCIAÇÕES NO MATERIAL.

Para mais informações sobre a Licença MIT: <https://opensource.org/licenses/MIT>.