

## ✓ 1. Requisitos do Sistema

### 📌 Requisitos Funcionais (RF):

1. **RF01:** O sistema deve permitir que o usuário insira nome de usuário e senha.
  2. **RF02:** O sistema deve validar os dados de login com base em credenciais fixas.
  3. **RF03:** O sistema deve exibir mensagens de erro em caso de falha no login.
  4. **RF04:** O sistema deve redirecionar o usuário para a dashboard após login válido.
  5. **RF05:** O sistema deve manter o login ativo usando `localStorage`.
  6. **RF06:** O sistema deve permitir logout e limpar a sessão.
- 

### 📌 Requisitos Não Funcionais (RNF):

1. **RNF01:** A interface deve ser responsiva e amigável em dispositivos móveis.
  2. **RNF02:** O sistema deve ser desenvolvido utilizando HTML, CSS e JavaScript puro.
  3. **RNF03:** O tempo de resposta ao login não deve ultrapassar 2 segundos.
  4. **RNF04:** O código-fonte deve ser modular e separado em pastas organizadas.
  5. **RNF05:** O projeto deve possuir testes automatizados integrados com GitHub Actions.
- 

## ✓ 2. UML: Diagrama de Casos de Uso

### 🎯 Objetivo:

Mostrar as interações entre o **usuário** e as **funcionalidades principais** do sistema GoAuth.

### Atores:

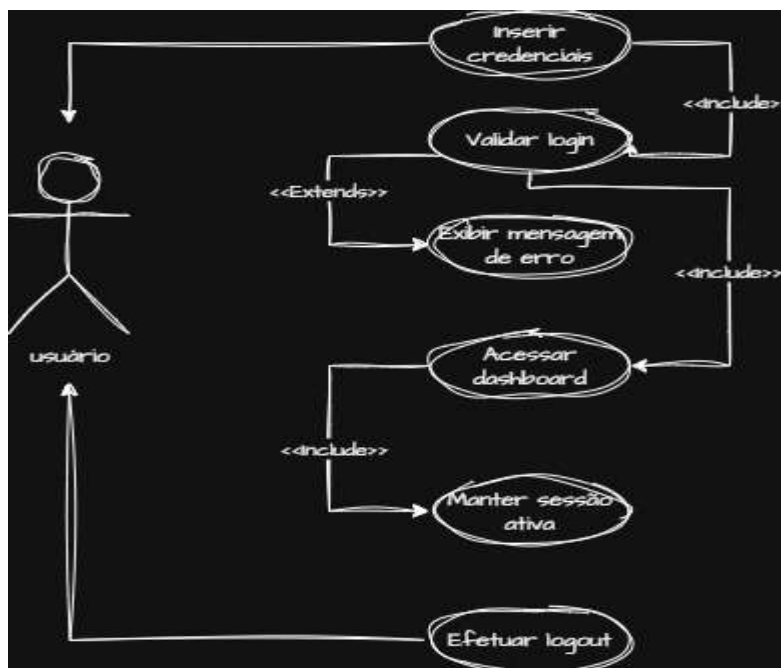
- Usuário

### Casos de Uso:

- Inserir credenciais
- Validar login
- Exibir mensagem de erro
- Acessar dashboard
- Manter sessão ativa
- Efetuar logout

### Relacionamentos:

- Usuário → (usa) Inserir credenciais
- Inserir credenciais → (include) Validar login
- Validar login → (extends) Exibir mensagem de erro
- Validar login → (success) Acessar dashboard
- Acessar dashboard → (include) Manter sessão ativa
- Usuário → (usa) Efetuar logout



### ✓ 3. UML: Diagrama de Classes

#### 🎯 Objetivo:

Mostrar a estrutura lógica do sistema, representando classes, atributos e responsabilidades.

