

# Projeto Secure Password

**Segurança Digital para a Comunidade**

Lourenço Prudêncio

Janeiro, 2025

# Índice

## Conteúdos

<b>Índice</b> .....	2
<b>Introdução</b> .....	3
<b>Visão Geral do Projeto</b> .....	3
<b>Importância da Segurança de passwords</b> .....	3
<b>Objetivos do Projeto</b> .....	3
<b>Objetivo Geral:</b> .....	3
<b>Objetivos Específicos</b> .....	3
<b>Funcionalidades</b> .....	4
<b>Verificador de Qualidade/Nível de Proteção de Senhas</b> .....	4
<b>Feedback em Tempo Real</b> .....	4
<b>Suporte a Idiomas</b> .....	4
<b>Gerar Passwords com Base nos Critérios</b> .....	4
<b>Funcionalidade de Copiar para a Área de Transferência</b> .....	4
<b>Fun Facts sobre Senhas Seguras</b> .....	4
<b>Requisitos Técnicos</b> .....	5
<b>Requisitos de Sistema</b> .....	5
<b>Requisitos de Segurança</b> .....	5
<b>Hospedagem e Implantação</b> .....	5
<b>Estrutura</b> .....	6
<b>Estrutura do Código</b> .....	6
<b>Fluxo de Trabalho da Aplicação</b> .....	7
<b>Componentes e Funcionalidades</b> .....	8
<b>Design e Interface</b> .....	9
<b>Interface do utilizador (UI)</b> .....	9
<b>Experiência do utilizador (UX)</b> .....	9
<b>Fluxograma</b> .....	9
<b>Conclusão</b> .....	10
<b>Exceções e Considerações</b> .....	10
<b>Referências</b> .....	11
<b>Conteúdo Adicional</b> .....	12
<b>Função : escapeHtml</b> .....	12
<b>Função : generatePassword</b> .....	12
<b>Função : checkPasswordCriteria</b> .....	13
<b>Função : toggleCriteria</b> .....	13

# Introdução

**Visão Geral do Projeto:** O Secure Password é uma aplicação web criada com o objetivo de incentivar os colaboradores e estudantes instituição a criar passwords fortes que vão ao encontro dos requisitos implementados por nos. Com a crescente importância da cibersegurança, é fundamental que todos tenham passwords que protejam eficazmente as informações pessoais e contas de cada um.

**Importância da Segurança de passwords:** A segurança digital é uma prioridade crescente. Passwords fracas são um dos principais alvos de ataques nos dias de hoje. Este projeto visa aumentar a consciência dos colaboradores e estudantes sobre a importância de utilizarem passwords robustas, que sejam difíceis de adivinhar e, assim, proteger dados sensíveis.

## Objetivos do Projeto

**Objetivo Geral:** O objetivo principal deste projeto é proporcionar uma ferramenta simples e eficaz para que os integrantes da instituição possam criar e ter feedback em tempo real com base nos requisitos impostos para uma password segura, promovendo assim melhores práticas de cibersegurança.

### Objetivos Específicos:

- Incentivar a criação de passwords seguras e robustas.
- Garantir a privacidade e segurança de dados dos utilizadores, sem que estes sejam armazenados ou transmitidos para um servidor.
- Oferecer feedback em tempo real sobre a qualidade da password escolhida.
- Promover a educação sobre a importância de passwords fortes e o impacto na segurança digital.

# Funcionalidades

**Verificador de Qualidade/Nível de Proteção de Senhas:** A aplicação permite que o utilizador escolha uma password e receba uma avaliação da segurança da mesma com base em vários critérios, como comprimento, complexidade e presença de sequências comuns. Esta ferramenta ajuda o utilizador a perceber o nível de proteção de sua password.

**Feedback em Tempo Real:** O utilizador recebe imediatamente feedback visual sobre a validade da senha, incluindo sugestões de melhoria. Isso permite ajustes rápidos para criar uma password mais segura.

**Suporte a Idiomas:** A aplicação está disponível em dois idiomas: inglês e português, garantindo maior acessibilidade para a comunidade diversificada da instituição. (Poderá ser feita tradução para outros idiomas)

**Gerar Passwords com Base nos Critérios:** A aplicação gera passwords seguras utilizando um conjunto extenso de caracteres, incluindo letras maiúsculas (26), letras minúsculas (26), números (10) e caracteres especiais (30). Com isso, a aplicação oferece até 3.111.928.305.110.923.294.648.827.904 de combinações possíveis de passwords.

**Funcionalidade de Copiar para a Área de Transferência:** Após ser gerada uma password ou ser feita validação de uma password, o utilizador pode copiar facilmente a password para a área de transferência, facilitando sua utilização para outros sistemas de login.

**Fun Facts sobre Senhas Seguras:** Após gerar uma password segura, a aplicação apresenta curiosidades sobre segurança de passwords, como a importância de ter complexidade nas passwords para prevenir ataques. Estes "fun facts" ajudam o utilizador a perceber sobre a relevância de manter boas práticas de segurança digital.

# Requisitos Técnicos

**Tecnologias Utilizadas:** O Secure Password foi desenvolvido utilizando as seguintes tecnologias:

Frontend:



HTML    CSS    JavaScript

**Requisitos de Sistema:** A aplicação é totalmente responsiva, funcionando de forma eficaz em qualquer dispositivo desktop e móveis.

**Requisitos de Segurança:** A aplicação não utiliza servidores e não envia dados para nenhum lugar. As informações de password e nome são processadas localmente no browser do utilizador e apagadas permanentemente quando a página é fechada/atualizada. Não é feita nenhuma coleta de dados pessoais, o que garante que os utilizadores tenham total controlo sobre suas informações.

**Hospedagem e Implantação:** A aplicação está atualmente hospedada no [Netlify](#), uma plataforma gratuita, que oferece deploy contínuo e fácil. A plataforma foi escolhida devido à sua simplicidade e integração com repositórios Git, tornando a implementação rápida e eficiente.

Caso seja necessário, a aplicação pode ser vista uma opção de implementar a aplicação diretamente numa página do [SharePoint](#).

# Estrutura

## Estrutura do Código

O código da aplicação Secure Password é dividido em três ficheiros principais, cada um com uma responsabilidade clara:

### **index.html:**

Este ficheiro contém o HTML da aplicação e é o ponto de entrada da interface de utilizador. Ele define a estrutura básica da página, incluindo o título, os campos de entrada (nome e password), botões de interação, e os elementos de feedback, como a lista de critérios e as mensagens de validação.

O ficheiro também inclui os links para o arquivo **CSS (Password.css)** e o **JavaScript (Password.js)**, além de recursos como o favicon da página e o logotipo da instituição.

### **Password.css:**

Este ficheiro contém a estilização da interface. Ele define a aparência visual de todos os elementos, como a cor de fundo, os campos de entrada, os botões, os critérios de password, e os efeitos de animação.

A utilização de animações CSS, como **fadeIn**, **slideIn**, e **showFact**, torna a interação mais agradável para o utilizador.

### **Password.js:**

O JavaScript é responsável por toda a interatividade da aplicação, incluindo a lógica de validação das passwords, a geração de passwords seguras, a mudança de idiomas, e todo o feedback visual.

Este ficheiro lida com eventos como os clique no botão de gerar as passwords, a atualização do contador de caracteres, a verificação dos critérios da password e a cópia da password para a área de transferência.

## **Fluxo de Trabalho da Aplicação**

O fluxo de trabalho da aplicação é simples e intuitivo, proporcionando uma experiência ao utilizador clara e eficiente. Pode ser descrito da seguinte forma:

### **Acesso à Página:**

- O utilizador acessa a página [Secure Password](#) no browser, onde é apresentado com os campos de entrada e a interface de verificação da password.

### **Escolha ou Geração da password:**

- O utilizador pode optar por escolher sua própria password ou gerar uma password segura ao clicar no botão "Gerar Password". A password gerada é automaticamente preenchida no campo de senha.

### **Feedback sobre a Segurança da Senha:**

- À medida que o utilizador escreve ou gera uma password, a aplicação fornece feedback em tempo real, indicando se a password atende aos critérios de segurança definidos, como comprimento, uso de caracteres especiais, números, e letras maiúsculas.
- Esse feedback é apresentado na lista de critérios, onde cada critério é marcado com uma cor ( ou ) dependendo se está atendido ou não.

### **Ajuste da password:**

- Se a password não atender a todos os critérios, o utilizador pode ajustá-la conforme as sugestões indicadas na interface. O contador de caracteres também é atualizado enquanto o utilizador escreve a password.

### **Cópia da password:**

- Quando todos os critérios de segurança são atendidos, o botão "Copiar password" aparece, permitindo que o utilizador copie a password gerada ou validada diretamente para a área de transferência.

## **Componentes e Funcionalidades**

### **Verificador de password:**

O verificador de password é uma funcionalidade central da aplicação. Ele avalia a password escolhida pelo utilizador em tempo real e verifica se ela atende a todos os critérios de segurança, incluindo:

- Comprimento mínimo de 14 caracteres.
- Inclusão de pelo menos um caractere especial.
- Presença de números, letras maiúsculas e minúsculas.
- Exclusão de partes do nome do utilizador e sequências comuns de caracteres.
- O feedback visual é mostrado imediatamente, e a interface é atualizada conforme os critérios são atendidos.

### **Gerador de passwords:**

- A aplicação gera passwords complexas a partir de um conjunto de caracteres pré-definido (letras maiúsculas e minúsculas, números e caracteres especiais). A password gerada possui pelo menos 14 caracteres e é misturada aleatoriamente para garantir que atenda os requisitos.
- O gerador é acionado ao clicar no botão "Gerar Password", e a password gerada é preenchida automaticamente no campo de entrada.

### **Área de Transferência:**

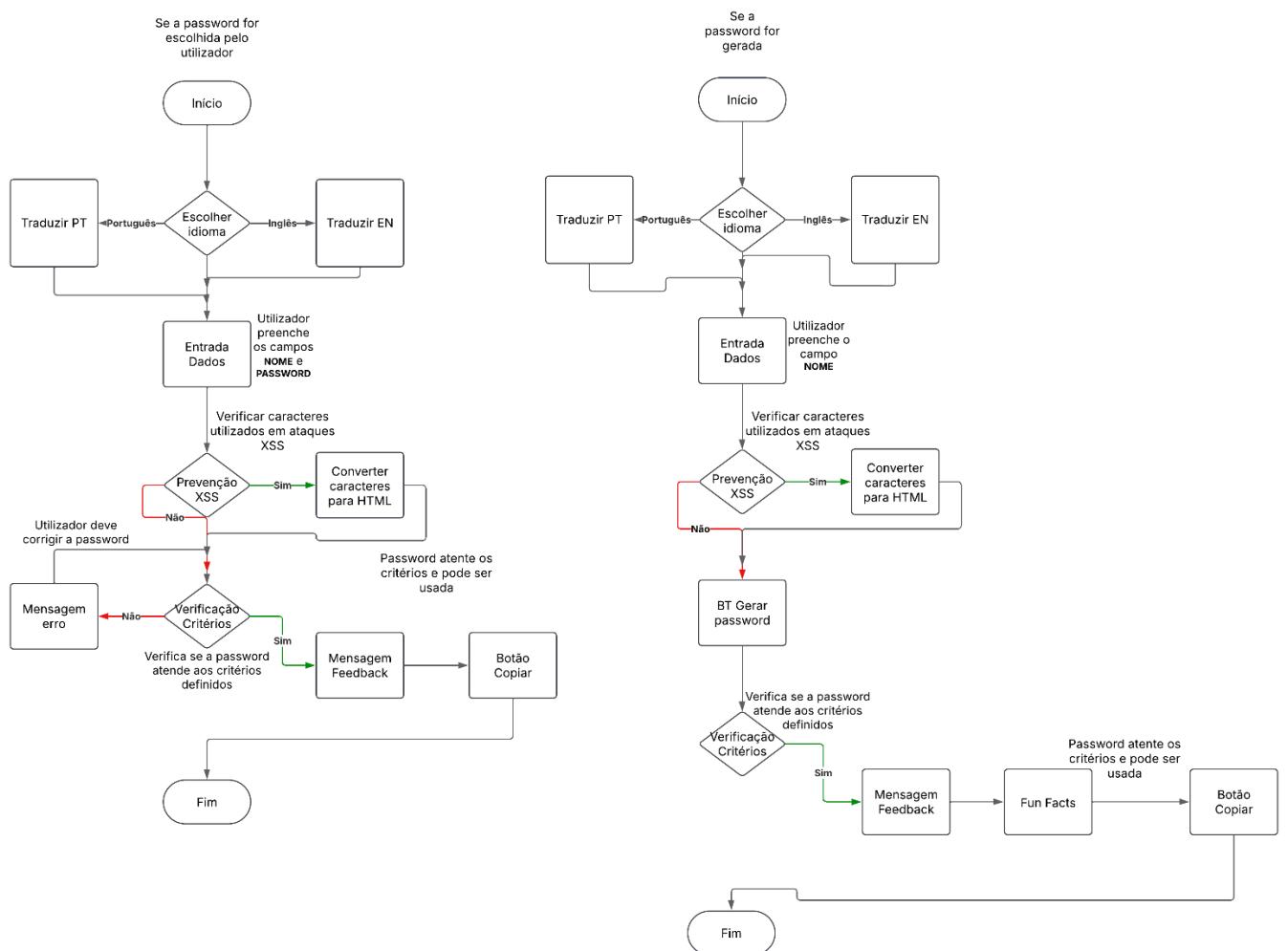
- A funcionalidade de cópia para a área de transferência permite que o utilizador copie facilmente a password gerada ou validada para usar em outros sites ou sistemas. Ao clicar no botão "Copiar password", a password é copiada para o clipboard do utilizador.
- A aplicação fornece uma mensagem de sucesso indicando que a password foi copiada, ou um erro caso a cópia falhe.

# Design e Interface

**Interface do utilizador (UI):** O design da interface é simples e intuitivo, permitindo que qualquer utilizador, independentemente da experiência com tecnologia, consiga utilizar a aplicação sem dificuldades. A paleta de cores e elementos gráficos foi escolhida para garantir boa visibilidade e acessibilidade.

**Experiência do utilizador (UX):** O site foi projetado para garantir uma navegação fluida e eficiente. Cada interação é clara, e o utilizador é orientado por meio de feedbacks visuais em tempo real.

## Fluxograma



# Conclusão

- **Resumo do Projeto:** O **Secure Password** oferece uma solução simples, mas poderosa, para promover a criação de passwords seguras dentro da comunidade da instituição. A aplicação proporciona feedback instantâneo sobre a qualidade da password e garante que a privacidade do utilizador seja respeitada.
- **Benefícios Esperados:** Espera-se que os colaboradores e estudantes utilizem a ferramenta para melhorar a segurança das suas passwords e, consequentemente, a proteção das suas contas online.
- **Diminuição de carga de trabalho:** Com a implementação desta ferramenta, irá haver uma diminuição significativa em questões de alteração de password, pois os critérios necessários para criar uma password segura para as contas da instituição estão já definidos.
- **Manutenção:** A aplicação Secure Password foi projetada para ser de fácil manutenção, com base na simplicidade e na escolha de tecnologias como HTML, CSS e JavaScript puros, sem a necessidade de servidores ou base dados. Dado que a aplicação não depende de nenhum tipo de infraestrutura de backend. Como a aplicação depende de funcionalidades estáticas (como a geração e validação de passwords) e não de componentes dinâmicos ou complexos, as atualizações são limitadas e raras. Caso surjam novos requisitos ou melhorias, o código pode ser facilmente atualizado e implantado sem afetar a estabilidade da aplicação. Por não haver comunicação com servidores externos, os dados inseridos pelo utilizadores são processados localmente, o que reduz o risco de ataques externos ou leaks de dados. A aplicação não exige manutenção contínua de segurança, exceto por atualizações para corrigir vulnerabilidades no código do frontend.

## Exceções e Considerações:

Embora a necessidade de manutenção seja mínima, pequenas atualizações podem ser necessárias, como:

**Correções de bugs:** Caso sejam identificados problemas no código ou funcionalidades.

**Melhorias na interface:** Para optimizar a experiência do usuário conforme o feedback.

**Ajustes para novos navegadores:** Caso surjam novas versões de navegadores que requeiram ajustes no código.

No entanto, essas atualizações podem ser feitas de maneira rápida e sem a necessidade de infraestrutura de servidor, garantindo que a manutenção seja leve e sem grandes custos.

# **Referências**

Website Secure Password - [Password Segura](#)

Projeto GitHub - [lourencoprudencio/Secure-Password: Try it out at: Secure Password \(Finalizado\)](#)

Hospedagem Netlify - <https://app.netlify.com/>

# Conteúdo Adicional

## Função : escapeHtml

Esta função foi implementada para evitar ataques XSS (Cross-Site Scripting), converte caracteres especiais, como <, >, e &, nas representações deles em HTML seguras (por exemplo, &lt;, &gt;, &amp;), garantindo que qualquer entrada de texto não seja interpretada como código HTML ou JavaScript.

```
// Função para tratar tentativas de utilizar caracteres comuns em ataques XSS e evitar ataques
function escapeHtml(input) {
  console.log("Texto a ser tratado escapeHtml:", input); // Console log do texto a ser tratado
  var element = document.createElement('div');
  if (input) {
    element.innerText = input;
    element.textContent = input;
  }
  const escapedText = element.innerHTML; // Retorna o texto tratado
  console.log("Texto tratado:", escapedText); // Console log do texto depois de tratado
  return escapedText;
}
```

## Função : generatePassword

Esta função gera uma password segura com critérios específicos, como comprimento mínimo e inclusão de caracteres especiais, números e letras maiúsculas/minúsculas. Ela também embaralha a senha gerada para garantir sua aleatoriedade.

```
// Função para gerar uma password segura
function generatePassword() {
  const length = 14; // Comprimento da password

  // Diferentes tipos de caracteres a serem usados
  const specialCharacters = "!\"#$%&'^*,.-/:;<>?{[\]^_`{|}~";
  const numbers = "0123456789";
  const lowercaseLetters = "abcdefghijklmnopqrstuvwxyz";
  const uppercaseLetters = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

  let password = "";

  // Adiciona pelo menos um caractere de cada tipo
  password += specialCharacters[Math.floor(Math.random() * specialCharacters.length)];
  password += numbers[Math.floor(Math.random() * numbers.length)];
  password += lowercaseLetters[Math.floor(Math.random() * lowercaseLetters.length)];
  password += uppercaseLetters[Math.floor(Math.random() * uppercaseLetters.length)];

  // Preenche o resto da password com caracteres aleatórios
  const allCharacters = specialCharacters + numbers + lowercaseLetters + uppercaseLetters;
  while (password.length < length) {
    password += allCharacters[Math.floor(Math.random() * allCharacters.length)];
  }

  // Baralha a password gerada
  password = password.split('').sort(() => Math.random()).join('');

  // Atualiza o campo da password com a nova password
  passwordInput.value = password;
  updateCharCounter(); // Atualiza o contador de caracteres
  checkPasswordCriteria(); // Verifica se a password cumpre os critérios
}

// Atualiza o contador de caracteres quando o utilizador escrever a password
passwordInput.addEventListener("input", () => {
  charCounter.textContent = `${passwordInput.value.length} caracteres`;
  checkPasswordCriteria();
});
```

## Função : checkPasswordCriteria

Esta função valida a password inserida pelo utilizador de acordo com os critérios predefinidos (comprimento, presença de caracteres especiais, números, etc.).

```
// Verifica se a password atende a todos os critérios
function checkPasswordCriteria() {
    const password = passwordInput.value;
    const username = usernameInput.value;

    // Escapa o nome do usuário para evitar XSS
    const sanitizedUsername = escapeHtml(username);

    // Verifica se a password tem pelo menos 14 caracteres
    const lengthValid = password.length >= 14;
    toggleCriteria("length", lengthValid);

    // Verifica se a password contém pelo menos um caractere especial
    const specialValid = /[!#$%&()^+,-./;:<=>?@[\]^_`{|}~]/s.test(password);
    toggleCriteria("special", specialValid);

    // Verifica se a password contém pelo menos uma letra
    const letterValid = /[a-zA-Z]/.test(password);
    toggleCriteria("letter", letterValid);

    // Verifica se a password contém pelo menos um número
    const numberValid = /\d/.test(password);
    toggleCriteria("number", numberValid);

    // Verifica se a password contém letras maiúsculas e minúsculas
    const upperLowerValid = /[A-Z]/.test(password) && /[a-z]/.test(password);
    toggleCriteria("uppercase", upperLowerValid);

    // Verifica se a password não contém partes do nome
    const noNameIncluded = !containsSequentialLetters(sanitizedUsername, password);
    toggleCriteria("noName", noNameIncluded);

    // Verifica se a password não contém sequências comuns
    const noCommonSeq = !/(123|abc|password|qwerty|asdf)/i.test(password);
    toggleCriteria("noCommonSequences", noCommonSeq);
}

// Verifica se o nome do está contido na password
function containsSequentialLetters(input, password) {
    const inputLower = input.toLowerCase().replace(/\s+/g, '').replace(/-/g, '');
    const passwordLower = password.toLowerCase();
}
```

## Função : toggleCriteria

Esta função altera o estado visual dos critérios, indicando se a password cumpre ou não com cada critério.

```
// Alternar o estado visual dos critérios
function toggleCriteria(id, isValid) {
    const criteriaItem = document.getElementById(id);
    if (isValid) {
        criteriaItem.classList.add("completed");
        criteriaItem.textContent = criteriaItem.textContent.replace("●", "●");
    } else {
        criteriaItem.classList.remove("completed");
        criteriaItem.textContent = criteriaItem.textContent.replace("●", "●");
    }
}
```