




Aula 10

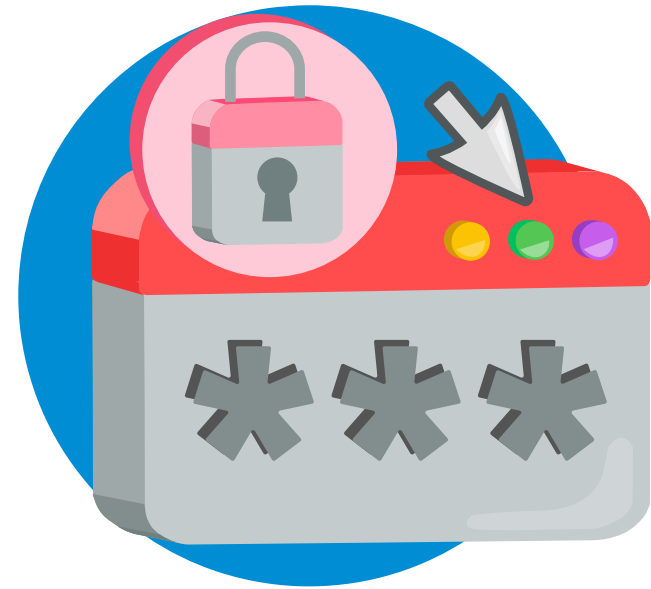
Medindo a força da senha

► **Unidade**

**Segurança digital: utilizando
matemática para programar
senhas seguras**

O que vamos aprender?

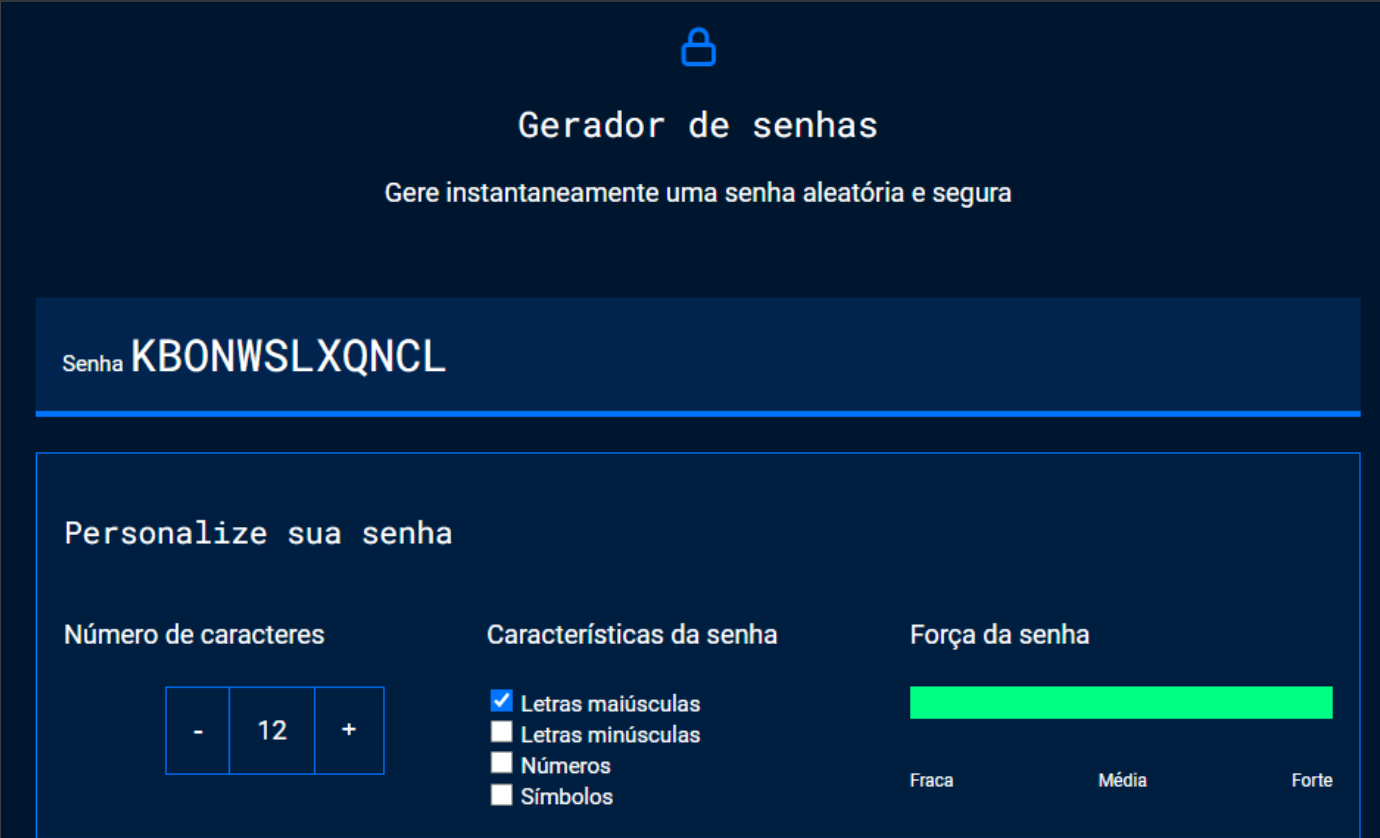
-  Gerar senhas com letras, números e símbolos aleatórios.
-  Testar retornos de uma função.
-  Usar condicionais para concatenar as opções marcadas nos checkboxes.



▶ CLIQUE E ACOMPANHE A AULA NA ALURA

Classificando uma senha

Na aula anterior, implementamos as características da senha de acordo com os checkboxes selecionados. Agora, faremos a funcionalidade de medição da força da senha e refinaremos nosso código. Para isso, usaremos JavaScript e os arquivos já criados em HTML e CSS.



The image shows a web application interface for a password generator. At the top, there is a blue header with a lock icon and the text "Gerador de senhas" and "Gere instantaneamente uma senha aleatória e segura". Below this, a dark blue box displays the generated password "KBONWSLXQNCL". The main section, titled "Personalize sua senha", contains three columns: "Número de caracteres" with a numeric input set to 12, "Características da senha" with checkboxes for "Letras maiúsculas" (checked), "Letras minúsculas", "Números", and "Símbolos", and "Força da senha" with a green progress bar and labels "Frac", "Média", and "Forte".

Gerador de senhas

Gere instantaneamente uma senha aleatória e segura

Senha KBONWSLXQNCL

Personalize sua senha

Número de caracteres


- 12 +

Características da senha

- ☒ Letras maiúsculas
- ☐ Letras minúsculas
- ☐ Números
- ☐ Símbolos

Força da senha

Fraca Média Forte

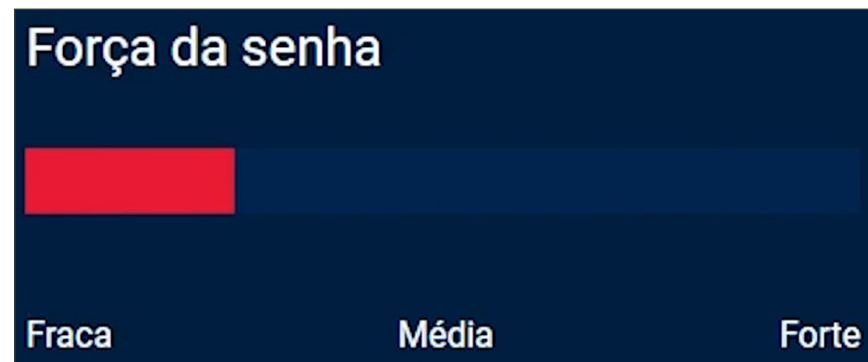


Nesta aula, vamos desenvolver a classificação da senha e refinaremos nosso código. Primeiramente, removeremos o **console.log()** e retomaremos à configuração anterior.

Isso é necessário, pois, à medida que progredimos, sempre precisamos pensar em boas práticas para facilitar o entendimento do código por outras pessoas e evitar bugs no futuro. Assim, vamos organizar o código antes de continuarmos. Deixaremos todas as variáveis no início do código. Observe:

```
const numeroSenha = document.querySelector('.parametro-senha__texto');
let tamanhoSenha = 12
numeroSenha.textContent = tamanhoSenha;
const letrasMaiusculas = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';
const letrasMinusculas = 'abcdefghijklmnopqrstuvwxyz';
const numeros = '0123456789';
const simbolos = '!@%*?';
const botoes = document.querySelectorAll('.parametro-senha__botao');
const campoSenha = document.querySelector('#campo-senha');
const checkbox = document.querySelectorAll('.checkbox');
```

Agora, nosso código começa com declarações de variáveis e, em seguida, colocaremos as funções, como o evento **onclick**. Após isso, basta chamar as funções. Uma observação importante é que a função **geraSenha** deve estar sempre no final do código. Portanto, ao chamar **geraSenha**, o programa buscará essa função no final do código. Assim, é importante também pensarmos um pouco na ordem em que os elementos estão organizados no código. Após organizar o código, vamos acessar a página do gerador de senhas. Observe que conseguimos gerar uma senha aplicando as diferentes características dos checkboxes, contudo, ainda precisamos adicionar a funcionalidade de classificação da força da senha (fraca, média ou forte). Essa implementação será feita por meio de uma barra vermelha, uma amarela e uma verde, que já implementamos parcialmente em aulas anteriores.





Desse modo, precisamos localizar a divisão de conteúdo que mostra essa classificação de força. Observe:

```
<div class="parametro-senha">  
  <h4 class="parametro-senha__titulo">Força da senha</h4>  
  <div class="barra"></div>  
  <div class="forca fraca"></div>  
  <div class="parametro-senha-textos">
```

Se usarmos um **querySelector** para selecionar esse elemento no arquivo *main.js*, podemos começar a manipular esse elemento, declarando uma variável, como mostrado a seguir:


```
const forcaSenha = document.querySelector('.forca');
```



Sabemos, então, que **forçaSenha** é um elemento manipulável. Observe que existe um conjunto de regras a serem definidas, como estipular o número de caracteres para a força da senha. Observe:

- Se a senha tiver mais que 12 caracteres ou mais, é forte;
- Se a senha tiver entre 11 e 6 caracteres, é média;
- Se a senha tiver 5 caracteres ou menos, é fraca.

Ou seja, podemos pensar em várias hipóteses. Desse modo, teremos uma série de códigos para escrever e, por isso, criaremos uma função para fazer a classificação da senha: **classificaSenha()**.



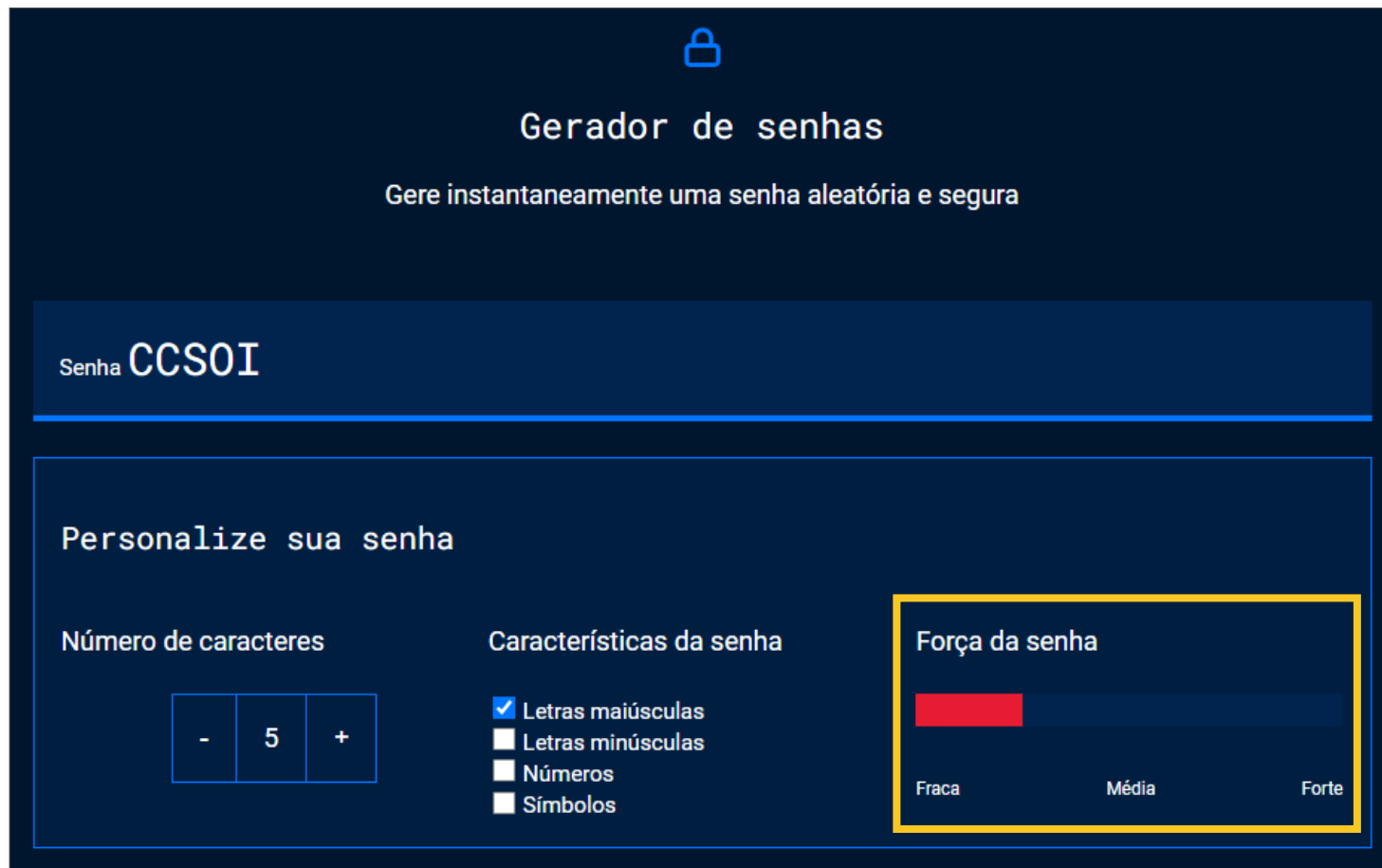
Dessa forma, queremos adicionar a classificação '**forte**', porque uma senha com 12 caracteres ou mais pode ser considerada forte. Acessaremos o elemento **forcaSenha** e utilizaremos o **classList**, que nos permite acessar todas as listas de classe. Por fim, vamos inserir a classificação de senha forte, observe:

```
function classificaSenha(){  
  forcaSenha.classList.add('forte')  
}
```

Note que a função foi criada, porém, ainda não foi chamada. Podemos chamá-la abaixo da definição da senha. Ou seja, dentro da função **geraSenha()**, antes de fecharmos a chave, podemos chamar a função **classificaSenha()**, como mostrado a seguir:

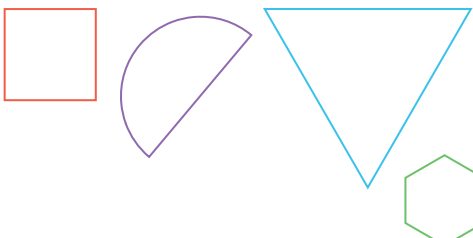
```
campoSenha.value = senha;  
classificaSenha();  
  
function classificaSenha(){  
  forcaSenha.classList.add('forte')  
}
```


Quando executamos a função **classificaSenha**, podemos visualizar o resultado em nossa página:



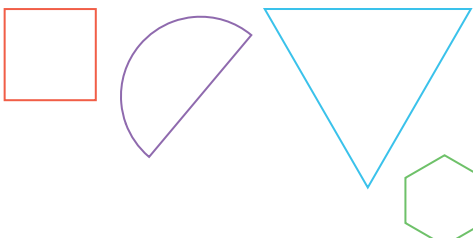
The screenshot shows a web application titled "Gerador de senhas" (Password Generator) with the subtitle "Gere instantaneamente uma senha aleatória e segura" (Generate instantly a random and secure password). The generated password "CCSOI" is displayed in a dark blue box. Below this, the "Personalize sua senha" (Customize your password) section allows users to adjust settings. The "Número de caracteres" (Number of characters) is set to 5, with minus and plus buttons for adjustment. The "Características da senha" (Password characteristics) section includes checkboxes for "Letras maiúsculas" (checked), "Letras minúsculas", "Números", and "Símbolos". The "Força da senha" (Password strength) section features a progress bar that is approximately 20% red, indicating a weak password, with labels for "Fraca", "Média", and "Forte".

Note que a aplicação mostra uma pequena barra vermelha. No entanto, ainda não definimos nenhuma condição para isso. Apenas adicionamos uma classificação e testamos se a função está correta.




Ao inspecionarmos os elementos da página, verificaremos que existe uma **classe forca** e as classificações de senha **fraca** e **forte**. Ou seja, há duas classificações aplicadas e queremos remover uma delas, pois não podemos ter as senhas fracas, médias e fortes em uma mesma divisão de conteúdo. Portanto, o objetivo é remover a classificação fraca, observe:

```
function classificaSenha(){  
    forcaSenha.classList.remove('fraca');  
    forcaSenha.classList.add('forte');  
}
```



Ainda não definimos a regra para adicionar a classificação, a **classificaSenha()**, e podemos especificar que uma senha forte só será adicionada sob determinada condição. A condição, nesse caso, é que o número de caracteres seja igual a 12 ou mais, ou seja, maior que 11. Então, adicionaremos a classificação de senha forte, observe:

```
function classificaSenha(){  
  forcaSenha.classList.remove('fraca');  
  if (tamanhoSenha > 11){  
    forcaSenha.classList.add('forte');  
  }  
}
```



Se a senha tiver menos que 12 caracteres, a ideia é que a classe forte seja apagada, pois, para 11 caracteres ou menos, outra barra deve aparecer. Porém, nosso marcador continua mostrando que a senha é forte. Ou seja, não removemos essa classe, já que a única classe removida foi a fraca. Por isso, precisamos pensar em uma forma de remover não somente a classe fraca, mas também as classes média e forte. Observe:

```
function classificaSenha(){  
  forcaSenha.classList.remove('fraca', 'media', 'forte');  
  if (tamanhoSenha > 11){  
    forcaSenha.classList.add('forte');  
  }  
}
```

Portanto, resolveremos essa questão removendo todas as classes possíveis e adicionando uma classe por vez, a depender do número de caracteres da senha. Se ela for maior que 11, ou seja, 12 ou mais caracteres, será definida como senha forte.

Analizando as funcionalidades, se uma senha com mais de 11 caracteres é classificada como forte, podemos usar a condição **else** para definir outra força de senha. Observe:

```
function classificaSenha(){
  forcaSenha.classList.remove('fraca', 'media', 'forte');
  if (tamanhoSenha > 11){
    forcaSenha.classList.add('forte');
  } else {
    forcaSenha.classList.add('media');
  }
}
```

Até o momento, a senha é forte ou média, a depender do número de caracteres. Ao testar nossa aplicação, notaremos que a barra está funcionando adequadamente para essas duas classificações:

Personalize sua senha


Número de caracteres: - 10 +

Características da senha:

- ☒ Letras maiúsculas
- ☐ Letras minúsculas
- ☐ Números
- ☐ Símbolos

Força da senha:

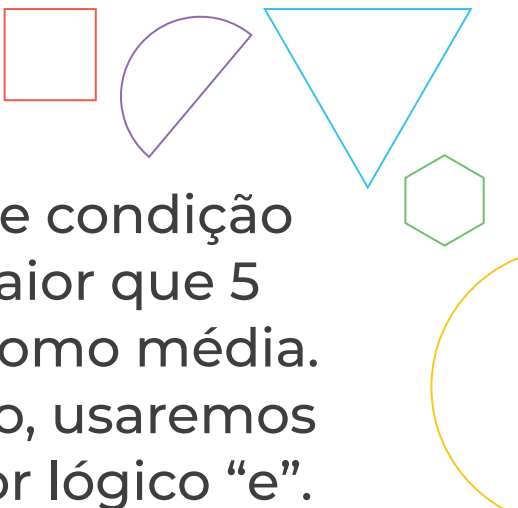
Fraca Média Forte



Contudo, ainda precisamos programar a classificação das senhas fracas, pois todas as senhas com 11 caracteres ou menos estão sendo sinalizadas como médias.

Logo, já especificamos uma condição para as senhas fortes (ter 12 ou mais caracteres). Agora, precisamos especificar a condição de uma senha média e a condição de uma senha fraca. Para fazer isso, adicionaremos outro **if**, usando a sintaxe **else if**. Observe:

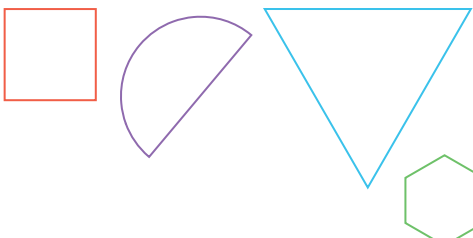
```
function classificaSenha(){
  forcaSenha.classList.remove('fraca', 'media', 'forte');
  if (tamanhoSenha > 11){
    forcaSenha.classList.add('forte');
  } else if {
    forcaSenha.classList.add('media');
  }
}
```



Após **else if**, abriremos parênteses e incluiremos a seguinte condição para definir as senhas médias: se o tamanho da senha for maior que 5 caracteres e menor que 12, então, a senha será classificada como média. Ou seja, temos dois limites: mínimo e máximo. Para fazer isso, usaremos dois operadores **&&** (“e” comercial), representando o operador lógico “e”. Observe:

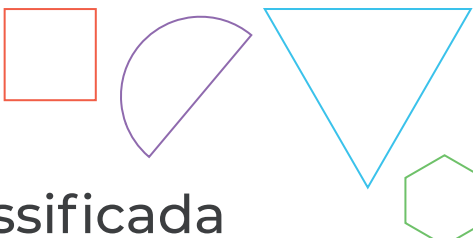
```
function classificaSenha(){  
    forcaSenha.classList.remove('fraca', 'media', 'forte');  
    if (tamanhoSenha > 11){  
        forcaSenha.classList.add('forte');  
    } else if (tamanhoSenha > 5 && tamanhoSenha < 12) {  
        forcaSenha.classList.add('media');  
    }  
}
```

Testando nossa aplicação, observaremos que conseguimos classificar de forma adequada as senhas médias também:



Personalize sua senha

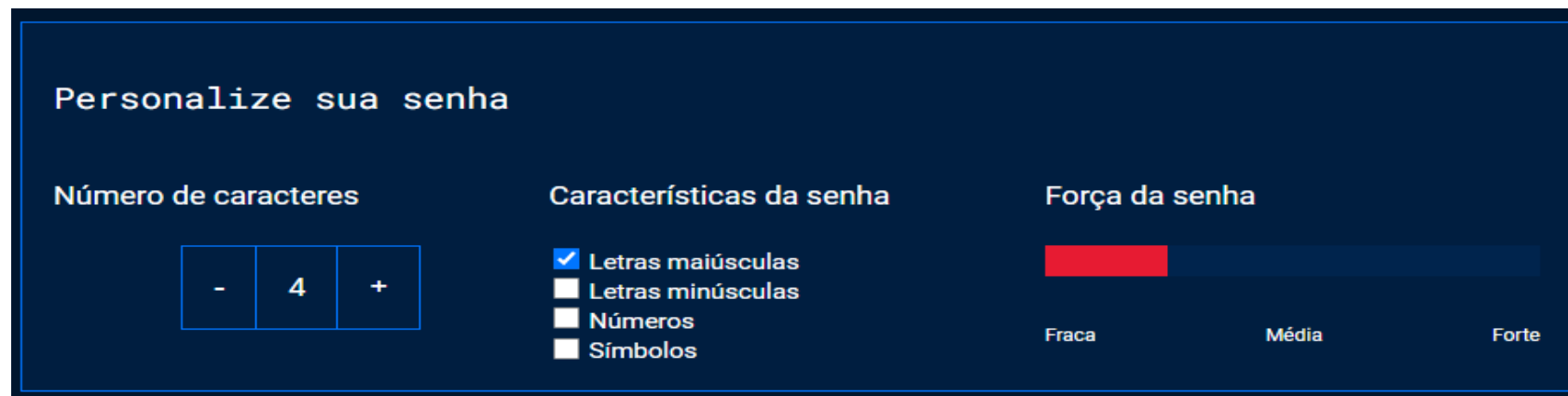
Número de caracteres	Características da senha	Força da senha
<div>-7+</div>	<div><input checked="" type="checkbox"/> Letras maiúsculas</div> <div><input type="checkbox"/> Letras minúsculas</div> <div><input type="checkbox"/> Números</div> <div><input type="checkbox"/> Símbolos</div>	<div><div></div></div> <div>FracaMédiaForte</div>



Ao testarmos, veremos que, com 12 caracteres, a senha é classificada como forte; com 11, a senha é média. Ao diminuirmos os caracteres, quando chegamos a 5, já não há mais nenhum critério definido e as barras coloridas desaparecem. Portanto, precisamos adicionar uma última classe para classificar as senhas fracas, observe:

```
function classificaSenha(){
    forcaSenha.classList.remove('fraca', 'media', 'forte');
    if (tamanhoSenha > 11){
        forcaSenha.classList.add('forte');
    } else if (tamanhoSenha > 5 && tamanhoSenha < 12) {
        forcaSenha.classList.add('media');
    } else if (tamanhoSenha < 6){
        forcaSenha.classList.add('fraca');
    }
}
```

Ao atualizarmos a página, observaremos que a classificação fraca é aplicada a todas as senhas que apresentam 5 caracteres ou menos:



The screenshot shows a dark-themed interface titled "Personalize sua senha". It is divided into three sections:

- Número de caracteres:** A numeric input field with a minus sign, the number "4", and a plus sign.
- Características da senha:** A list of checkboxes:
 - ☒ Letras maiúsculas
 - ☐ Letras minúsculas
 - ☐ Números
 - ☐ Símbolos
- Força da senha:** A horizontal progress bar with a red segment on the left. Below the bar are three labels: "Fraca", "Média", and "Forte".

Observe que, ao testamos novamente nossa aplicação, verificamos que a senha com 12 caracteres é forte e, com 11, média. Se deixarmos o valor da senha fraca como menor que 5, o valor 5 não funcionará, e a senha será classificada como fraca apenas para 4, 3, 2 e 1. Portanto, a condição correta para a senha média deve ser “maior que 6”. Podemos também utilizar o sinal de menor igual (\leq) a 5, por exemplo.

Na próxima aula, iremos pensar mais a respeito da segurança da senha, mesclando números, símbolos, caracteres maiúsculos e minúsculos. Até breve!

► Desafio

Nesta aula, aprendemos como definir a segurança de uma senha por meio de intervalos. Seu desafio será criar uma condição para descobrir quando uma pessoa é considerada criança (menor que 12 anos), adolescente (maior que 12 anos e menor que 18) e adulta (maior e igual a 18 anos), trazendo na tela a classificação de acordo com a idade em barras coloridas, como fizemos com nossa senha.



CLIQUE **AQUI** PARA AVALIAR ESTE MATERIAL