

GERADOR DE SENHAS AUTOMATICAS

◆ LINHAS 1 A 4 – Criação dos Arrays de Caracteres

```
js Copiar Editar

const letraMaiuscula = 'abcdefghijklmnopqrstuvwxyz'.split('');
const letraMinuscula = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'.split('');
const numeros = '0123456789'.split('');
const simbolos = '!@#$%^&*()_+={}:;><.,/?'.split('');
```

? O que é isso?

Cada linha está pegando uma **string de caracteres** e transformando ela em um **array (lista)** com cada caractere separado.

✅ Explicando cada um:

1. `split()`:

O método `.split()` quebra a string em **caracteres individuais**. Exemplo:

```
js Copiar Editar

'abc'.split('') // resultado: ['a', 'b', 'c']
```

2. O que temos em cada linha:

```
js Copiar Editar

const letraMinuscula = 'abcdefghijklmnopqrstuvwxyz'.split('');
const letraMaiuscula = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'.split('');
```

Cria um array com letras maiúsculas e minúsculas.

```
js Copiar Editar

const numeros = '0123456789'.split('');
```

Cria um array com os dígitos de 0 a 9.

```
js Copiar Editar

const simbolos = '!@#$%^&*()_+={}:;><.,/?'.split('');
```

Cria um array com vários **símbolos especiais**, comuns em senhas seguras.

◆ LINHA 5 – Juntando todos os caracteres

```
js Copiar Editar  
  
const todosCaracteres = letraMaiuscula.concat(letraMinuscula, numeros, simbolos);
```

? O que é .concat()?

O método .concat() **junta arrays**. Então:

```
js Copiar Editar  
  
['A', 'B'].concat(['a', 'b']) // retorna ['A', 'B', 'a', 'b']
```

✓ O que essa linha faz?

Ela pega todos os arrays de letras, números e símbolos e junta tudo em um único array chamado todosCaracteres.

Esse array será usado para **sortear os caracteres da senha**.

◆ LINHAS 6 A 7 – Tamanho da senha e variável inutilizada

```
js Copiar Editar  
  
const passwordlength = 12;  
const password = "";
```

- passwordlength: define o **tamanho da senha** que será gerada.
- password = "": esta linha é **desnecessária** porque a variável não é usada fora da função. Pode ser apagada.

◆ LINHAS 9 A 15 – A função que gera a senha

```
js Copiar Editar

function gerarSenha() {
  let password = "";
  for (let i = 0; i < passwordlength; i++) {
    const randomNumber = Math.floor(Math.random() * todosCaracteres.length);
    password += todosCaracteres[randomNumber];
  }
  return password;
}
```

Vamos por partes:

◆ Linha 1:

```
js Copiar Editar

function gerarSenha() {
```

Define uma função chamada gerarSenha que, ao ser chamada, vai gerar e **retornar** uma senha aleatória.

◆ Linha 2:

```
js Copiar Editar

let password = "";
```

Cria uma variável chamada password e inicia como string vazia (""), onde os caracteres sorteados serão colocados um por um.

◆ Linha 3:

```
js Copiar Editar

for (let i = 0; i < passwordlength; i++) {
```

Isso é um **loop for**. Ele repete o que está dentro dele 12 vezes (porque passwordlength = 12).

◆ Linha 4:

```
js                                                                    Copiar  Editar
const randomNumber = Math.floor(Math.random() * todosCaracteres.length);
```

Explicando parte por parte:

- **Math.random()** → gera um número decimal aleatório entre 0 e 1. Ex: 0.723
- **Math.random() * todosCaracteres.length** → transforma isso num número entre 0 e o tamanho do array.
- **Math.floor(...)** → arredonda para baixo, garantindo que o número final seja um índice válido.

Exemplo: se o **array todosCaracteres** tem 70 itens, essa linha gera um número entre 0 e 69, que é usado pra pegar um caractere aleatório.

◆ Linha 5:

```
js                                                                    Copiar  Editar
password += todosCaracteres[randomNumber];
```

Esse += é um **atalho para concatenar**.

- Ele pega o caractere aleatório e **acrescenta** na variável password.

◆ Linha 7:

```
js                                                                    Copiar  Editar
return password;
```

Depois que o loop termina, a função **devolve** a senha pronta com os 12 caracteres.

← **Resumo geral (bem mastigado):**

- Você cria listas com letras, números e símbolos.
- Junta tudo num único array.
- Sorteia caracteres aleatórios desse array 12 vezes.
- Vai juntando tudo na variável password.
- Quando terminar, a função retorna a senha pronta.

CODIGO COMPLETO

```
index.js > ...
1
2  /* Gerador de senha */
3  const letraMinuscula = 'abcdefghijklmnopqrstuvwxyz'.split('');
4  const letraMaiuscula = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'.split('');
5  const numeros = '0123456789'.split('');
6  const simbolos = '!@#%$%^&*()_+-={}[]:;><,./?'.split('');
7  const todosCaracteres = letraMaiuscula.concat(letraMinuscula, numeros, simbolos);
8  const passwordlength = 12;
9
10 /*Função para gerar senha*/
11 function gerarSenha() {
12     let password = "";
13     for (let i = 0; i < passwordlength; i++) {
14         const randomNumber = Math.floor(Math.random() * todosCaracteres.length);
15         password += todosCaracteres[randomNumber];
16     }
17     return password;
18 }
19
20 console.log(gerarSenha());
21
22
```

TERMINAL

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Vinicius Brandao@ViniciusBrandao MINGW64 /j/Meu Drive/04 - DIO.PRO/00. Estudos Gerais/03. Desafios/01. Gerador de Senha
• $ node index.js
Ml;T8X9vNS??

Vinicius Brandao@ViniciusBrandao MINGW64 /j/Meu Drive/04 - DIO.PRO/00. Estudos Gerais/03. Desafios/01. Gerador de Senha
• $ node index.js
TG.&rdQz*~W;

Vinicius Brandao@ViniciusBrandao MINGW64 /j/Meu Drive/04 - DIO.PRO/00. Estudos Gerais/03. Desafios/01. Gerador de Senha
• $ node index.js
YT0+;n]sp2Q2

Vinicius Brandao@ViniciusBrandao MINGW64 /j/Meu Drive/04 - DIO.PRO/00. Estudos Gerais/03. Desafios/01. Gerador de Senha
○ $
```