

05

O cálculo dos dígitos validadores

Para a validação do campo de CPF, existe uma API do governo que faz as validações, garantindo que o CPF é válido e que pertence à alguém. O problema é que essa API não é pública e também pode revelar informações pessoais. Então não podemos fazer a validação de que o CPF pertence à alguém, mas podemos verificar a integridade da estrutura desse CPF através de uma fórmula matemática.

Escolha a alternativa que contém a função que valida o primeiro dígito verificador:

Selezione uma alternativa

A

```
function checaDigitoVerificadorCPF(cpf, multiplicador)
    let soma = 0
    let contador = 0
    const cpfSemDigitos = cpf.substr(0, multiplicador)
    const digitoVerificador = cpf.charAt(multiplicador)
    for(; multiplicador > 1 ; multiplicador--) {
        soma = soma + cpfSemDigitos[contador] * multiplicador
        contador++
    }

    if(soma % 11 > 9) {
        return digitoVerificador == 0
    }

    return digitoVerificador == 11 - (soma % 11)
```

B

```
function checaDigitoVerificadorCPF(cpfs, multiplicado)
    let soma = 0
    let contador = 0
    const cpfSemDigitos = cpfs.substr(0, multiplicado)
    const digitoVerificador = cpfs.charAt(multiplicado)
    for(let multiplicador = 10 ; multiplicador > 1 ;
        soma = soma + cpfSemDigitos[contador] * multiplicador;
        contador++)
    }

    if(soma % 11 > 9) {
        return digitoVerificador == 0
    }

    return digitoVerificador == 11 - (soma % 11)
}
```

C

```
function checaDigitoVerificadorCPF(cpfs, multiplicado)
    let soma = 0
    let contador = 0
    const cpfSemDigitos = cpfs.substr(0, multiplicado)
    const digitoVerificador = cpfs.charAt(multiplicado)
    for(; multiplicador > 1 ; multiplicador--) {
        soma = soma + cpfSemDigitos[contador] * multiplicador;
        contador++
    }

    if(soma % 11 > 9) {
        return digitoVerificador == 0
    }
```

```
        return digitoVerificador == 11 - (soma % 11)
    }
```

D

```
function checaDigitoVerificadorCPF(cpf, multiplicado
    let soma = 0
    let contador = 0
    const cpfSemDigitos = cpf.substr(0, multiplicado
    const digitoVerificador = cpf.charAt(multiplicado
    for(; multiplicador > 1 ; multiplicador--) {
        soma = soma + cpfSemDigitos[contador] * multiplicador
        contador++
    }

    if(soma % 11 > 9) {
        return digitoVerificador == 0
    }

    return digitoVerificador == 11 - (soma % 11)
}
```