



# Algoritmos e Introdução à Programação (EAGS SIN 2020)

AULAS 4.3, 4.4 - Qualidade de algoritmos

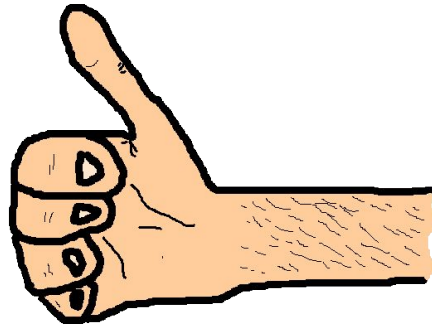
Apresentado por 2S SIN NETTO e 2S SIN MOURA



# Se liga !

Você vai aprender:


1. Refatorar algoritmos
2. Utilizar técnicas de Clean Code em algoritmos



# Qualidade de algoritmos



**FUI DORMIR ANTES DE  
TERMINAR O PROGRAMA**



**AGORA NÃO FAÇO A MÍNIMA IDEIA  
DO QUE EU TAVA FAZENDO**

**QUANDO ESCREVI ESSE CÓDIGO, SÓ  
EU E DEUS ENTENDÍAMOS O QUE ELE FAZIA.**



**AGORA, SÓ DEUS SABE**

# Mudança no requisito

Vamos supor que as regras de saque agora mudaram e para cada saque realizado partir de 50% do saldo, será cobrada uma taxa de 0.010% de imposto sobre o saldo inicial. Além disso, a conta corrente passará a possuir um limite de cheque especial no valor de R\$ 100,00. Ou seja, as solicitações agora poderão ser maiores que o saldo até o limite do cheque especial (saldo negativo de R\$ -100,00). Nem precisamos pensar muito que isso acrescenta mais dois problemas em nosso estudo de caso. Observe o algoritmo a seguir:

Preparado?



```
1 function permitirSaque(quantia) {
2
3     var saldo = 500.00;
4     const imposto = 0.010;
5     const chequeEspecial = 100.00
6
7     // permite sacar sempre que o saque for menor ou igual ao saldo + chequeEspecial E saque menor que a metade do saldo
8     if ((quantia <= (chequeEspecial + saldo)) && (quantia <= (saldo / 2))){
9         saldo = saldo - quantia
10        console.log('Saque da quantia de ' + quantia + ' autorizado! Novo saldo: R$ ' + saldo)
11
12        // permite sacar sempre que o saque for menor ou igual ao saldo + chequeEspecial E saque maior que a metade do saldo com imposto
13    } else if ((quantia <= (chequeEspecial + saldo)) && (quantia > (saldo / 2))) {
14        var novoSaldo = saldo - quantia
15        var saldoDescontadoOIposto = novoSaldo - saldo * imposto
16        console.log (saldoDescontadoOIposto)
17
18        console.log('Saque da quantia de ' + quantia + ' autorizado! Novo saldo: R$ ' + saldoDescontadoOIposto)
19    }else if ((quantia > (chequeEspecial + saldo))) {
20        console.log('Saque não autorizado!')
21    }
22
23 }
```

*Go Horse Detected ....*

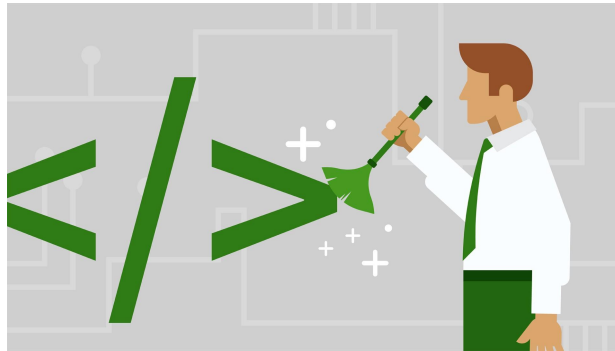


# Refatoração



```
1  var saldo = 500.00;
2
3  function permitirSaque(quantia) {
4
5      const chequeEspecial = 100;
6
7      if (quantia <= (chequeEspecial + saldo)){
8          console.log('Saque da quantia de ' + quantia + ' autorizado!') // Não sabe ainda o novo saldo
9          descontarSaqueDoSaldo(quantia)
10
11      }else{
12          console.log('Saque não autorizado!')
13      }
14  }
15
16
17  function descontarSaqueDoSaldo(quantia) {
18
19      var novoSaldo = saldo - quantia // O novo saldo deve ser descoberto aqui
20      var saldoFinal = descontarImpostoDoSaldo(novoSaldo, quantia)
21      console.log('Novo saldo: R$ ' + saldoFinal)
22  }
23
24
25  function descontarImpostoDoSaldo(novoSaldo, quantia){
26      const imposto = 0.010;
27      if (quantia > (saldo / 2)){
28
29          var saldoDescontadoOImposto = novoSaldo - (saldo * imposto)
30          return saldoDescontadoOImposto
31      }
32
33      return novoSaldo
34  }
35
36
```

# Clean Code



## Apostila!!

Tem muito mais sobre Clean Code para entender nos itens 4.4.1, 4.4.2 e 4.4.3 da apostila

Uma forma de chamar um conjunto de técnicas que podem ser utilizadas para um código de fácil entendimento.

- **Nomes de variáveis compatíveis com um contexto (domínio)**
- **Quanto menos Else .... Else-If melhor**
- **If ternários**
- **Código auto explicativo**

# AULA PRÁTICA

CARLOS





# Obrigado !

Você aprendeu:

1. Analisar algoritmos e compreendê-los
2. Observar a construção do princípio da responsabilidade única
3. Refatorar algoritmos

