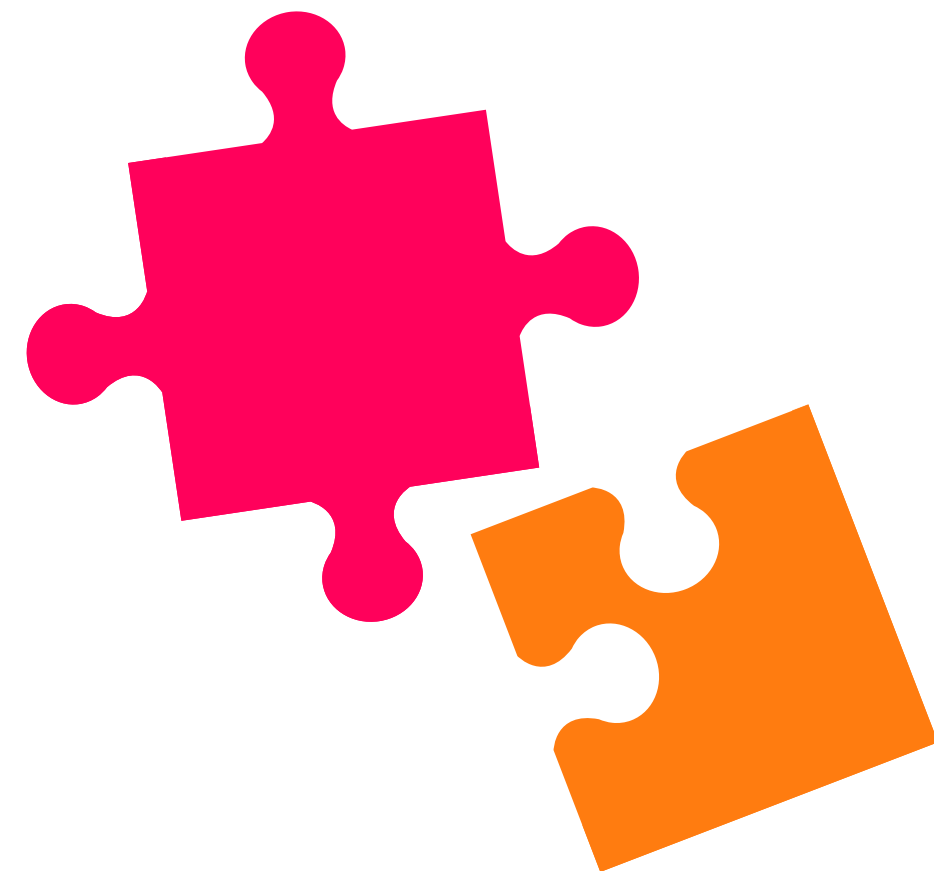


RoboLab:

Boas Práticas de Codificação



**Fique tranquilo(a),
não é necessário
conhecimento prévio!**



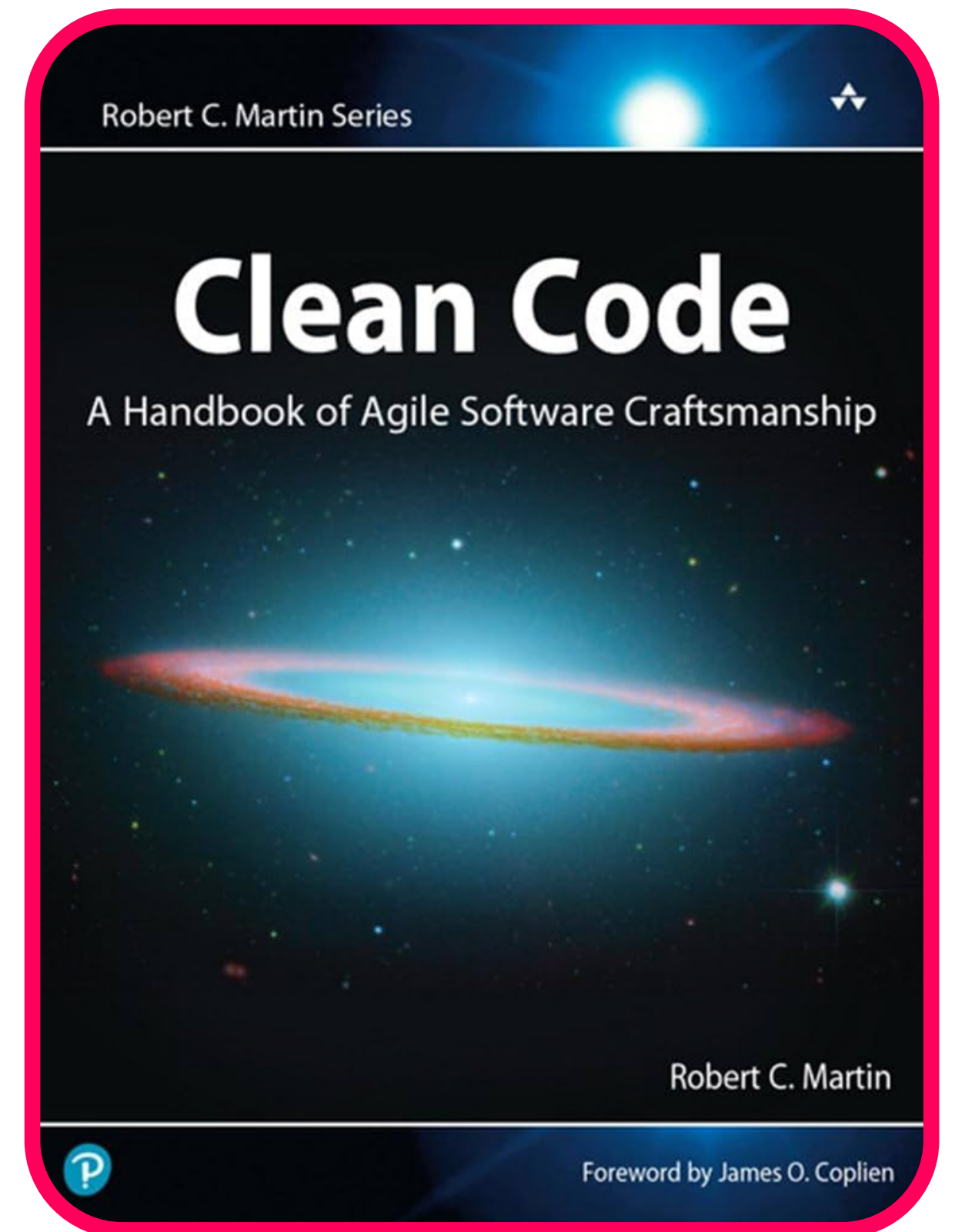
O que são boas práticas de codificação?

- São **diretrizes e padrões** que os desenvolvedores seguem ao escrever código de programação.
- Esses padrões **ajudam na compreensão do código**.
- **Agilizam** no processo de criação, alteração e manutenção do código.

```
b) return 1; return 0; } function  
e, j=i?m.cache:a, k=i?a[h]:a  
b) return k || (k=i?a[h]=c.pop  
eof b || "function"==typeof b)  
[k], e || (g.data || (g.data={}))  
b?(f=g[b], null==f&&(f=g[m  
d, e, f=a.nodeType, g=f?m.cac  
ta)) {m.isArray(b)?b=b.concat  
d?[b]:b.split(" ")), e=b.leng  
rn}(c || (delete g[h].data, P  
ete g[h]:g[h]=null))}} } func  
n y.activeElement} catch(a)  
); if(c.createElement) for(; b  
c, d, e=0, f=typeof a.getElem  
SelectorAll !== K?a.querySele  
(d=c[e]); e++) !b || m.nodeName  
ame(a, b)?m.merge([a], f):f}}  
ion wb(a, b){return m.nodeName  
.getElementsByTagName("tbody  
")):a} function xb(a){return  
}\n
```

Livro “Clean Code”

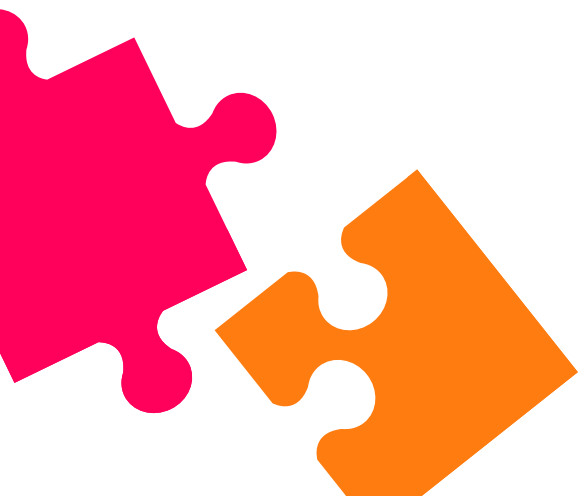
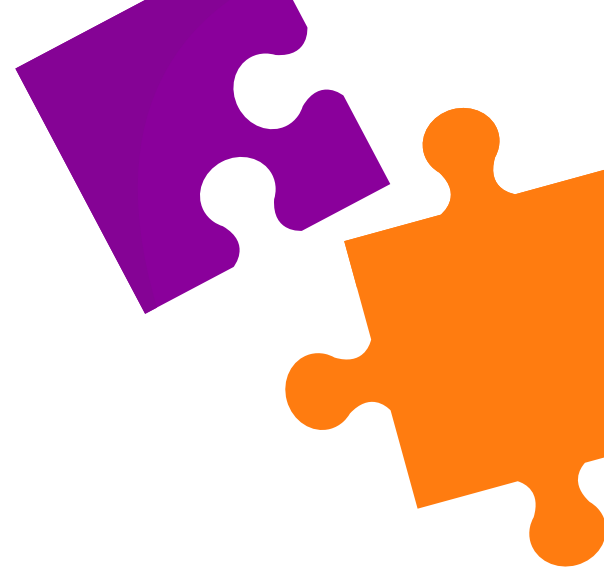
- Destaca importância da **clareza e da legibilidade do código** como fatores cruciais na programação.
- Guia valioso para **aprimorar as habilidades de programação** e escrever software mais eficiente e confiável.
- Amplamente adotado pela comunidade de programadores.



O que é o código limpo?

Pra que serve?

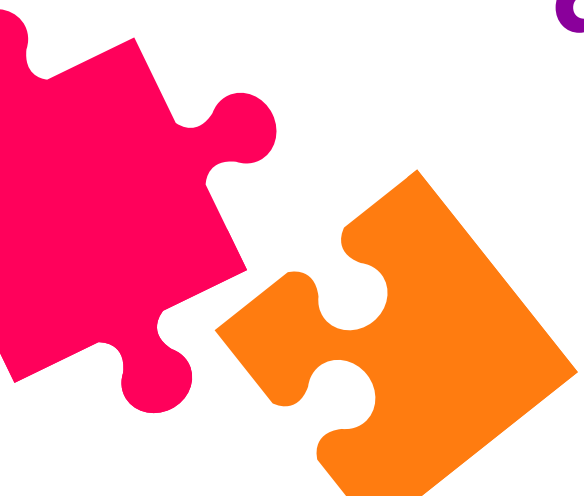
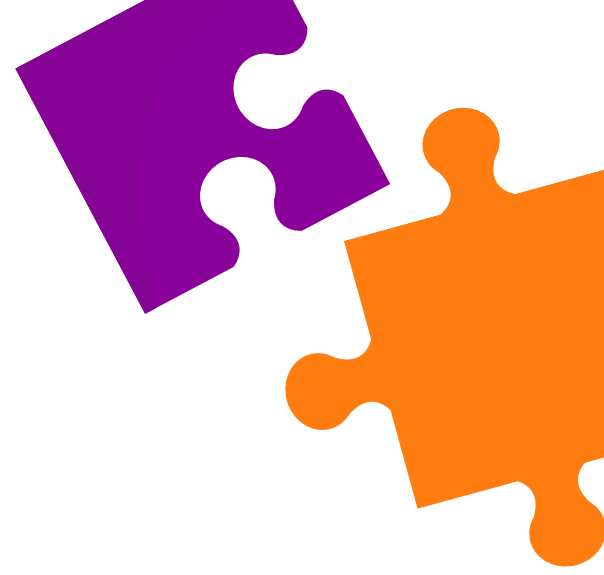
- Um código limpo é **mais fácil de ler e entender**, tornando-o acessível a todos, inclusive a outros desenvolvedores.
- Isso ajuda tanto você quanto outros programadores a **compreenderem o que o código faz e como ele opera**.



O que é o código limpo?

Vantagens

- **Detectar erros no código é mais simples** quando ele é limpo, agilizando o processo de correção.
- O código limpo é mais propenso a ser reutilizado em projetos futuros, **economizando tempo e esforço** no desenvolvimento subsequente.



Como o código feito de forma errada se parece:

```
function x(y, z) {  
  return y + z; }  
  
const a = 5;  
const b = 10;  
const c = x(a, b);  
console.Log('Resultado: ' + c)
```



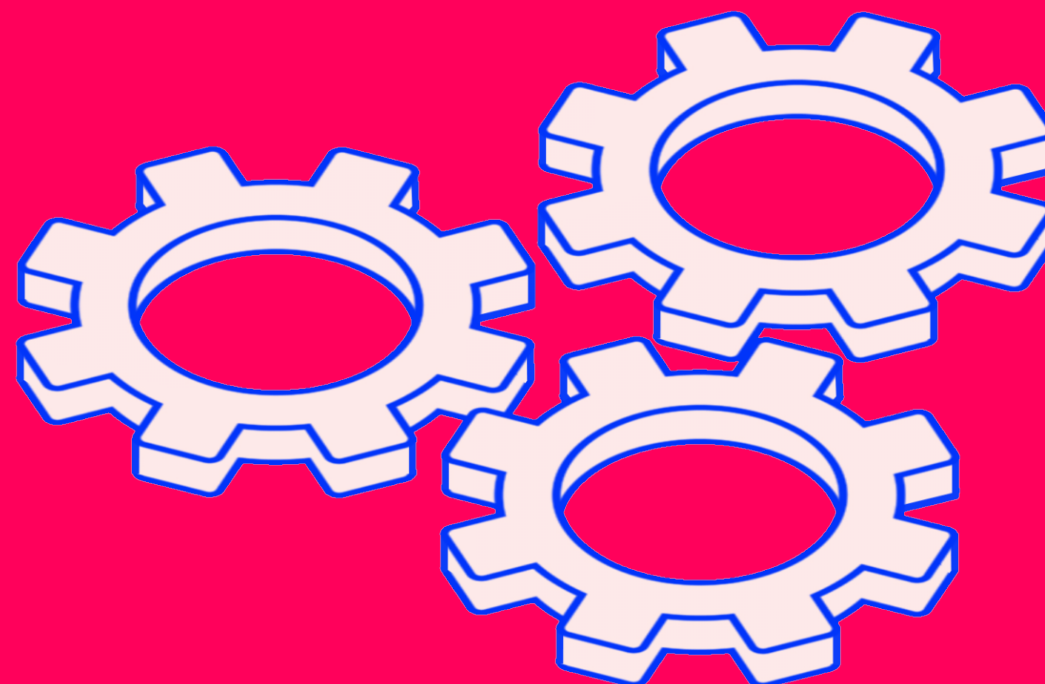
Como o código feito de forma correta se parece:



```
function somar(a, b) {  
    return a + b;  
}  
  
const numero1 = 5;  
const numero2 = 10;  
  
const resultado = somar(numero1, numero2);  
  
console.log(`Resultado: ` + resultado);
```

- **Recuo consistente;**
- **Nomenclatura descritiva;**
- **Espaços em branco adequados.**

**Vamos colocar a
mão na massa?**

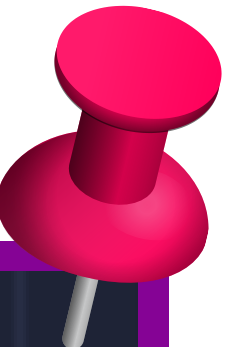


Desafio do dia:



Descrição

Você foi contratado para melhorar a legibilidade de um código que calcula o salário semanal de um funcionário. O código atual foi feito por um antigo desenvolvedor e usa variáveis pouco compreensíveis. Sua tarefa é refatorar o código, dando nomes mais descritivos às variáveis.



```
let h = parseFloat(prompt("Insira o número de horas trabalhadas:"));
let v = parseFloat(prompt("Insira o valor da hora:"));

let sb = h * v;
let p = 0.2 * sb;
let i = 0.1 * h * v;
let l = sb - p - i;
let df = 0.05 * l;
let s = l - df;

console.log("O salário semanal é: $" + s);
```

Falando com a equipe financeira, você descobriu algumas informações que podem ser úteis para que você consiga entender o que está sendo calculado no código:

O salário bruto é dado pela multiplicação entre o número de horas trabalhadas e o valor da hora;

O desconto da previdência social corresponde a 20% do salário bruto;

O desconto de imposto de renda corresponde a 10% do salário bruto;

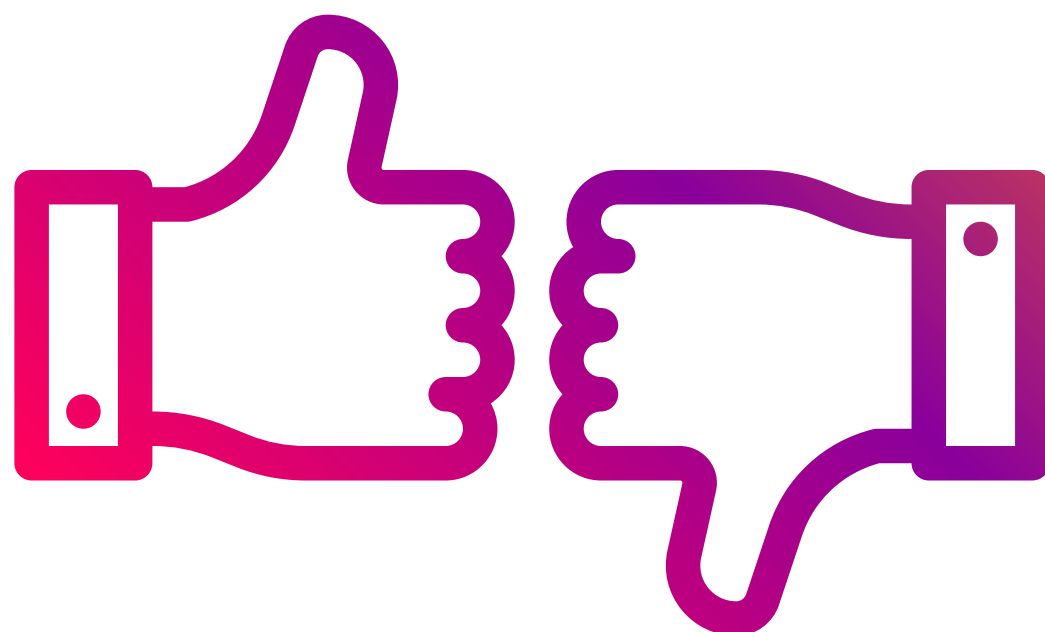
O salário líquido é o salário bruto menos os descontos de previdência social e de imposto de renda;

O salário líquido possui um desconto de 5%;

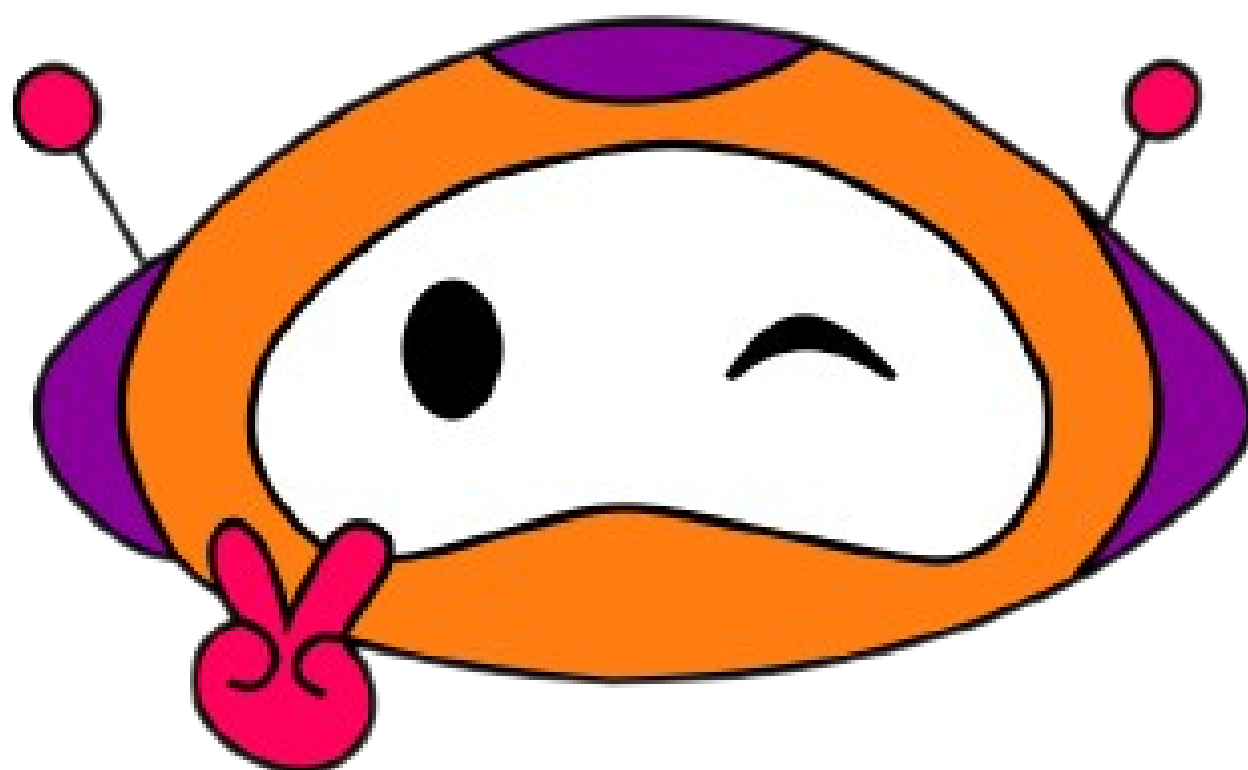
O salário final é dado pelo salário líquido menos o desconto de 5%.



Feedback



**Seu feedback
é muito
importante
para nós!**



Obrigado!



Obrigada por participar!
Te esperamos na próxima aula.