



CLEAN CODE

O USO DE BOAS PRÁTICAS DE
CÓDIGO LIMPO COM NODEJS

Oficina de NodeJS com Typescript que apresentará a importância do Código Limpo no desenvolvimento de software e utilizará de boas práticas para construir uma simples aplicação back-end de criação de usuários.

TS



Clean Code

A Handbook of Agile Software Craftsmanship

O QUE É?

Clean Code é uma filosofia de desenvolvimento cuja o principal objetivo é aplicar técnicas simples que visam facilitar a escrita e leitura de um código, tornando-o de fácil compreensão e revelando a sua real intenção.

<https://medium.com/desenvolvendo-com-paixao/1-clean-code-o-que-%C3%A9-porque-usar-1e4f9f4454c6>



POR QUE USAR?

- 01. Legibilidade
- 02. Manutenibilidade
- 03. Escalabilidade
- 04. Eficiência
- 05. Reutilização
- 06. Depuração
- 07. Colaboração



ROBERT C. MARTIN

APRESENTAÇÃO

Robert Cecil Martin, mais conhecido como "Uncle Bob," é um renomado autor, consultor e palestrante na área de desenvolvimento de software. Ele é amplamente reconhecido por seu papel na promoção das melhores práticas de engenharia de software e é um dos principais defensores da metodologia ágil e do desenvolvimento orientado a objetos.

Martin é autor de vários livros influentes sobre programação e design de software, incluindo "Clean Code: A Handbook of Agile Software Craftsmanship." Ele é um dos fundadores do Manifesto Ágil e tem desempenhado um papel fundamental na disseminação dos princípios ágeis na indústria de desenvolvimento de software.

Além disso, Robert C. Martin é co-autor do Princípio SOLID, um conjunto de diretrizes de design de software que promovem a criação de código limpo e de alta qualidade. Ele é um proponente entusiástico da importância da manutenção e melhoria contínuas do código-fonte para criar sistemas robustos e escaláveis.



***Não basta que o código
funcione.***



PRINCÍPIO KISS

Keep It Simple, Stupid é um princípio de design que afirma que projetos e/ou sistemas devem ser tão simples quanto possível. Sempre que possível, a complexidade deve ser evitada, pois a simplicidade garante maiores níveis de aceitação e interação do usuário. O *KISS* é usado em uma variedade de disciplinas, como design de interface, design de produto e desenvolvimento de software.



PRINCÍPIO DRY

Don't Repeat Yourself é um princípio de desenvolvimento de software que visa reduzir a repetição de informações de todos os tipos, especialmente no código-fonte. Isso é alcançado pela divisão do código em partes menores, chamadas funções ou métodos, que podem ser reutilizadas em vez de duplicadas.



PRINCÍPIO YAGNI

You aren't gonna need it é um princípio de desenvolvimento de software extremamente útil, que visa combater a sobrecarga desnecessária no desenvolvimento de código. Ao aderir a este princípio, podemos assegurar que estamos investindo nosso tempo e recursos de forma mais eficiente, aprimorando o valor que nosso software traz para os usuários.



REGRAS DE CÓDIGO

Existem algumas regras que o Clean Code dita, importantes para manter o propósito da filosofia, são regras que se seguidas, deixam o código mais limpo e entendível.

MANTENHA CONSTÂNCIA



```
1  class CustomerRepository { }  
2  
3  class ProdutoRepository { }  
4  
5  class RepositorioUnidadeMedida { }
```

ABUSE DE VARIÁVEIS DESCRITIVAS



```
1  const total = 0
```

```
2
```

```
3  const valorTotalDoCarrinho = 0
```

FOQUE EM DISTINÇÕES SIGNIFICANTES



```
1  const cron = '*/* * * * *'  
2  
3  const temporizador = '*/* * * * *'  
4  
5  const aCadaCincoSegundos = '*/* * * * *'
```

- # TODOS OS NOMES PRECISAM SER BUSCÁVEIS



```
1  let nmVgs = 5
```

```
2
```

```
3  let numeroDeVagas = 5
```

NÃO USE PREFIXOS



```
1  class clsCustomerRepository { }  
2  
3  function funcAddsNumber () { }  
4  
5  const strName = 'Otávio'
```

NÃO USE CARACTERES ESPECIAIS



```
1  const tipoCartão = 'CRÉDITO'  
2  
3  const função = 'PROGRAMADOR'
```

APENAS UM OBJETIVO



```
1  async cadastrarUsuario() {  
2    // Valida CPF  
3    // Cadastra no banco  
4  }  
5  
6  async validaCPF() { }  
7  async persiste() { }
```



```

function getElementsByName(name) {
    var elements = [];
    for (var i = 0; i < document.all.length; i++) {
        if (document.all[i].name == name) {
            elements.push(document.all[i]);
        }
    }
    return elements;
}

getElementsByName("id");

```