



# Clean Code

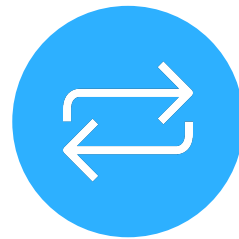
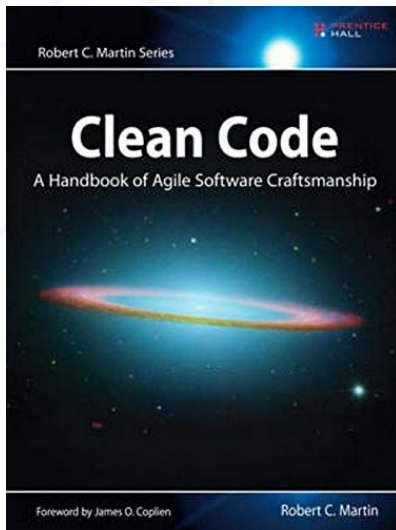
Padrões e Desenho de Software

Hugo Paiva de Almeida  
93915  
LEI

# Introdução

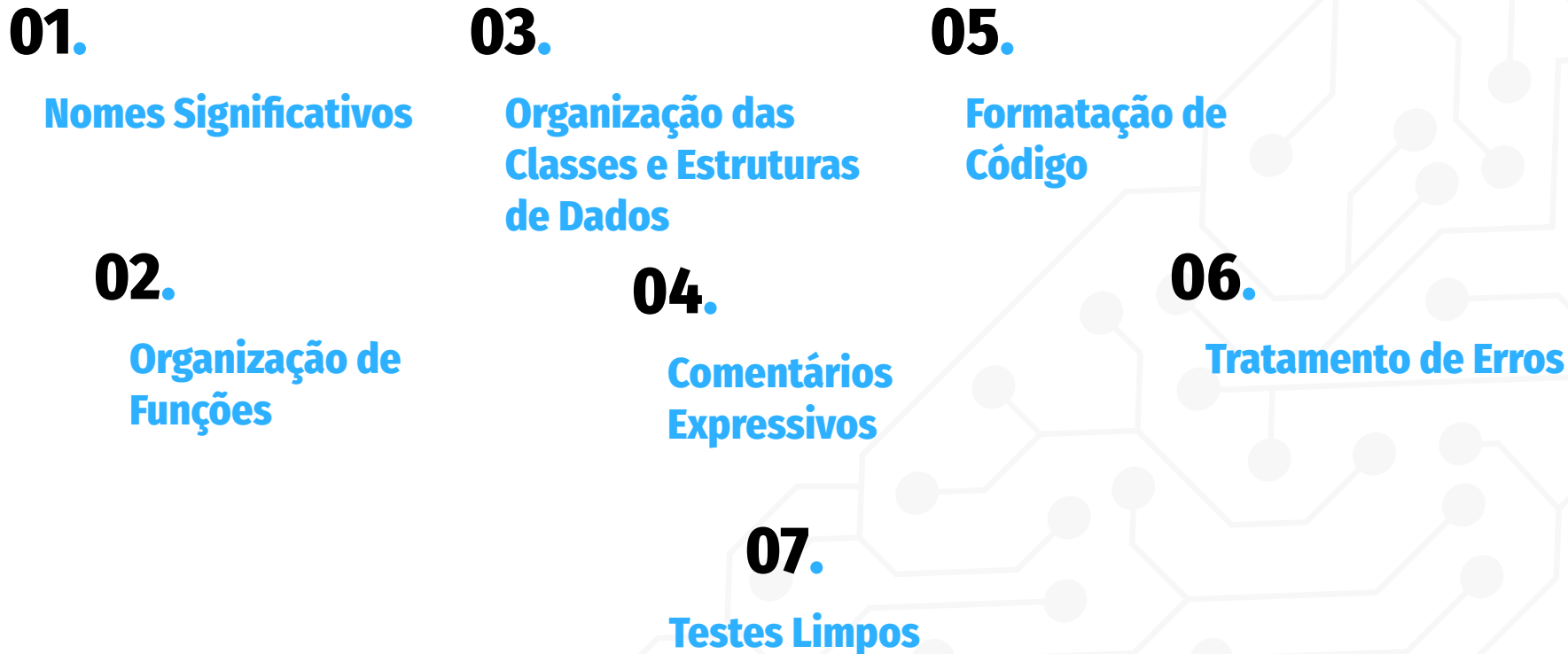


**Fácil de entender**



**Susceptível a  
mudanças**

# Princípios e Boas Práticas

- 
- 01.**  
Nomes Significativos
  - 02.**  
Organização de Funções
  - 03.**  
Organização das Classes e Estruturas de Dados
  - 04.**  
Comentários Expressivos
  - 05.**  
Formatação de Código
  - 06.**  
Tratamento de Erros
  - 07.**  
Testes Limpos

# Princípios e Boas Práticas

## Nomes Significativos

Devem passar a sua ideia central

```
1  int[] f; // frutas

1  for (int l=0; l<50; l++) {
    if (f[l] == 1) {
3      f[l] = 2;
    }
5 }
```

Em caso de necessidade, utilizar  
nomes grandes

```
1  int fruitsAtHome;

1  final int NUMBER_OF_FRUITS = 50;
   for (int l=0; l<NUMBER_OF_FRUITS; l++) {
3       if (fruitsAtHome[l] == ROTTEN) {
           fruitsAtHome[l] = TRASH;
5       }
   }
```

# Princípios e Boas Práticas

## Organização de Funções

As funções devem ser pequenas

Devem ser ainda mais pequenas

```
Circle makeCircle(double x, double y, double radius);    Circle makeCircle(Point center, double radius);
```

# Princípios e Boas Práticas

## Organização das Classes e Estruturas de Dados



### Abstração

```
1 public interface Phone {  
    double getBatteryCapacityInMah();  
3    double getMahOfEnergyRemaing();  
}
```

```
2 public interface Phone {  
    double getPercentBatteryRemaining();  
}
```

# Princípios e Boas Práticas

## Comentários Expressivos



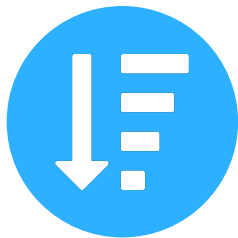
São mesmo  
necessários?

```
// Verificar se o carro ja deixou de pagar IUC em Portugal  
if (car.age>25 || (car.eletric == true))...
```

```
if (!car.paysIuc())...
```

# Princípios e Boas Práticas

## Formatação de Código



**Ordem das  
funções**

```
public class Student{  
    private    int        number;  
    private    String     name;  
  
    public void    insertData(int    n,        String    na){  
        number = n;  
        name =    na;  
    }  
}
```



**120 caracteres por  
linha**

```
public class Student {  
    private int number;  
    private String name;  
  
    public void insertData(int r, String na) {  
        number = n;  
        name = na;  
    }  
}
```



**Comentários são  
necessários?**



# Princípios e Boas Práticas

## Tratamento de Erros



**Utilizar exceções**



**Evitar return *null***

# Princípios e Boas Práticas

## Testes Limpos

**F.I.R.S.T**

**Fast.**

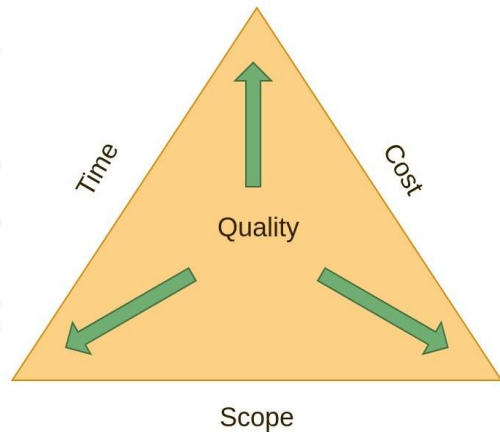
**Repeatable.**

**Independent.**

**Self-Validating.**

**Timely**

# Clean Code aplicado no Mundo



**Triângulo da gestão de um projeto**

**Qualidade = Código Limpo**