

BOAS PRÁTICAS E CLEAN CODE

CLEAN CODE

Óbvio para
outros
programadores

Não contém
código
duplicado

Quantidade
mínima de
elementos

Passa em
todos os
testes

CLEAN CODE

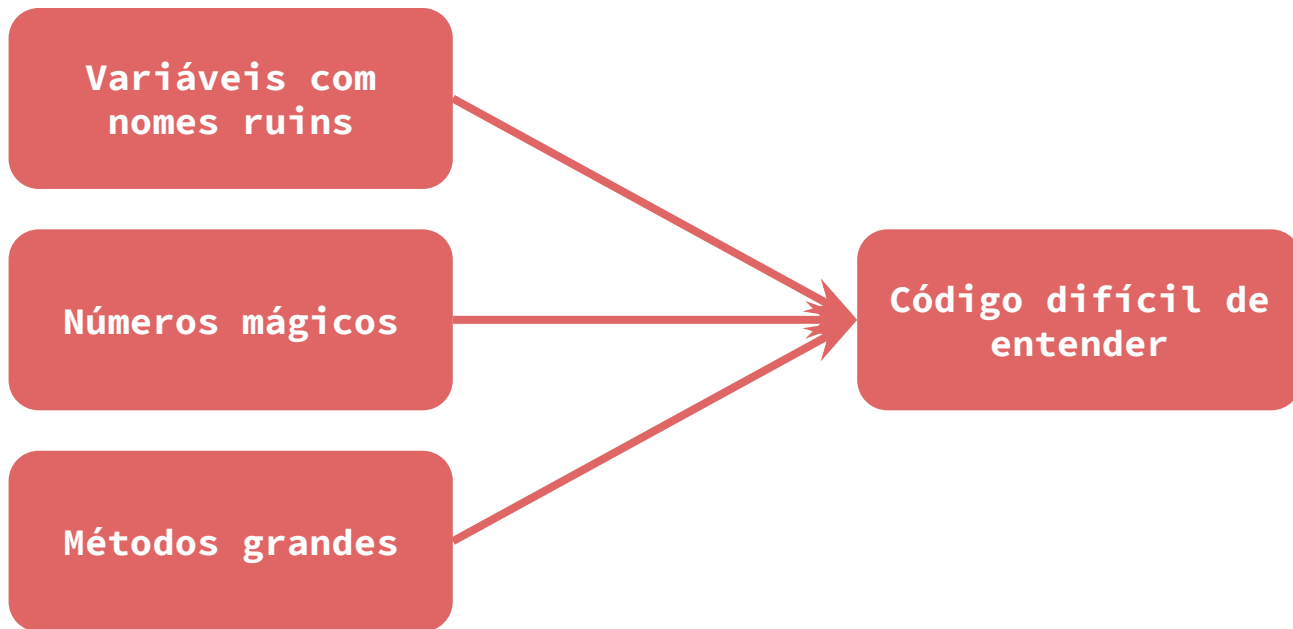
Óbvio para
outros
programadores

Não contém
código
duplicado

Quantidade
mínima de
elementos

Passa em
todos os
testes

ÓBVIO PARA OUTROS PROGRAMADORES



CLEAN CODE

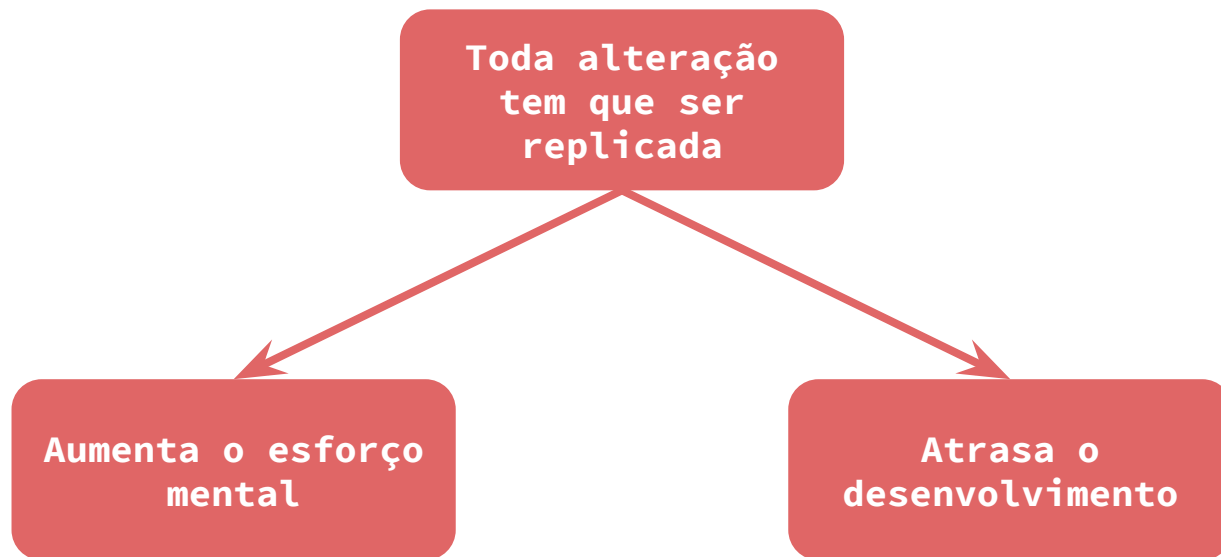
Óbvio para
outros
programadores

Não contém
código
duplicado

Quantidade
mínima de
elementos

Passa em
todos os
testes

NÃO CONTÉM CÓDIGO DUPLICADO



CLEAN CODE

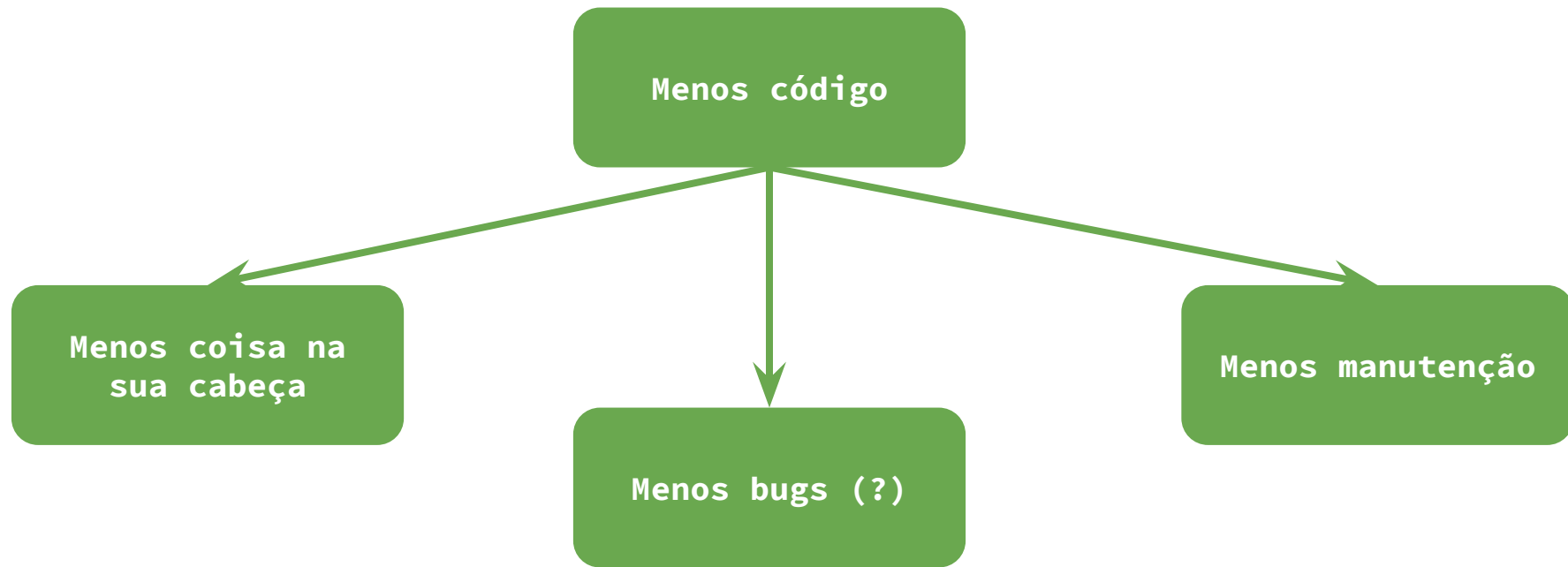
Óbvio para
outros
programadores

Não contém
código
duplicado

Quantidade
mínima de
elementos

Passa em
todos os
testes

QUANTIDADE MÍNIMA DE ELEMENTOS



QUANTIDADE MÍNIMA DE ELEMENTOS

KISS
Keep It Short and Simple

CLEAN CODE

Óbvio para
outros
programadores

Não contém
código
duplicado

Quantidade
mínima de
elementos

Passa em
todos os
testes

PASSA EM TODOS OS TESTES

**Código sujo
quando 95% passa**

**Boa sorte se a
cobertura for 0%**

CLEAN CODE

Óbvio para
outros
programas

Não contém
código
repetido

**Mais fácil
e barato
de manter**

Quantidade
mínima de
elementos

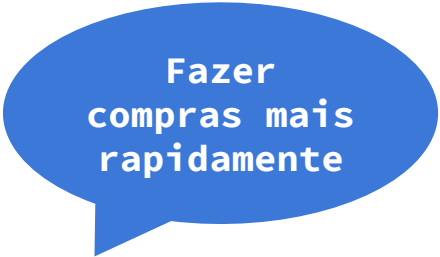
Passa em
todos os
testes

DÉBITO TÉCNICO





**Metáfora
do
empréstimo**



Fazer
compras mais
rapidamente



Metáfora
do
empréstimo

Fazer
compras mais
rapidamente

Metáfora
do
empréstimo

Paga juros
pela
“facilidade”

Fazer
compras mais
rapidamente

Pode ter
mais do que
pode pagar

**Metáfora
do
empréstimo**

Paga juros
pela
“facilidade”

Fazer
compras mais
rapidamente

Pode ter
mais do que
pode pagar

**Metáfora
do
empréstimo**

Paga juros
pela
“facilidade”

Impossível

PRINCIPAIS CAUSAS

Pressão comercial

PRINCIPAIS CAUSAS

Pressão comercial

**Desconhecer as
consequências**

PRINCIPAIS CAUSAS

Pressão comercial

**Desconhecer as
consequências**

**Dificuldade de
modularizar**

PRINCIPAIS CAUSAS

Pressão comercial

**Desconhecer as
consequências**

**Dificuldade de
modularizar**

Falta de testes

PRINCIPAIS CAUSAS

Pressão comercial

**Desconhecer as
consequências**

**Dificuldade de
modularizar**

Falta de testes

**Falta de
documentação**

PRINCIPAIS CAUSAS

Pressão comercial

**Desconhecer as
consequências**

**Dificuldade de
modularizar**

Falta de testes

**Falta de
documentação**

**Falta de
interação da
equipe**

PRINCIPAIS CAUSAS

Pressão comercial

**Desconhecer as
consequências**

**Dificuldade de
modularizar**

Falta de testes

**Falta de
documentação**

**Falta de
interação da
equipe**

**Desenvolvimento
simultâneo
separado**

PRINCIPAIS CAUSAS

Pressão comercial

**Desconhecer as
consequências**

**Dificuldade de
modularizar**

Falta de testes

**Falta de
documentação**

**Falta de
interação da
equipe**

**Desenvolvimento
simultâneo
separado**

**Refatoração
atrasada**

PRINCIPAIS CAUSAS

Pressão comercial

**Desconhecer as
consequências**

**Dificuldade de
modularizar**

Falta de testes

**Falta de
documentação**

**Falta de
interação da
equipe**

**Desenvolvimento
simultâneo
separado**

**Refatoração
atrasada**

**Não conformidade
com padrões**

PRINCIPAIS CAUSAS

Pressão comercial

**Desconhecer as
consequências**

**Dificuldade de
modularizar**

Falta de testes

**Falta de
documentação**

**Falta de
interação da
equipe**

**Desenvolvimento
simultâneo
separado**

**Refatoração
atrasada**

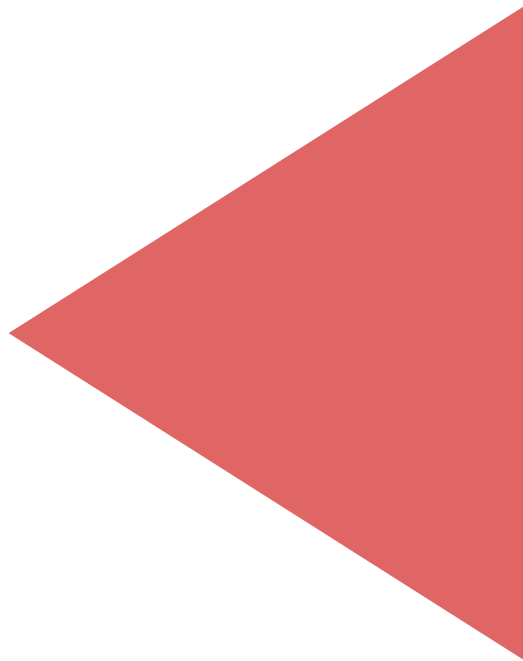
**Não conformidade
com padrões**

Incompetência

REFATORAR



QUANDO REFATORAR



QUANDO REFATORAR

3x Fez a mesma coisa três vezes?

QUANDO REFATORAR

3x Fez a mesma coisa três vezes?

Q É hora do code review?

QUANDO REFATORAR

 Fez a mesma coisa três vezes?

 É hora do code review?

 Vai adicionar funcionalidade?

QUANDO REFATORAR

 Fez a mesma coisa três vezes?

 É hora do code review?

 Vai adicionar funcionalidade?

 Vai consertar um bug?

QUANDO REFATORAR

 Fez a mesma coisa três vezes?

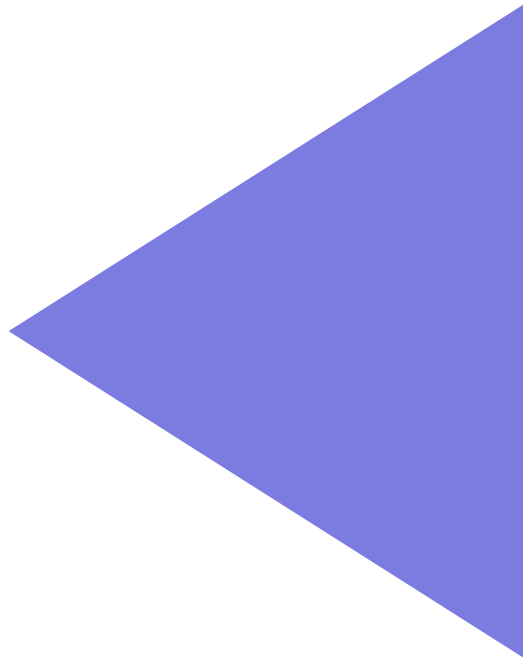
 É hora do code review?

 Vai adicionar funcionalidade?

 Vai consertar um bug?

**Hora de
refatorar**

COMO REFATORAR



COMO REFATORAR

O código deve ficar mais limpo.

COMO REFATORAR

O código deve ficar mais limpo.

Por que não ficou?

COMO REFATORAR

O código deve ficar mais limpo.

Por que não ficou?

O código estava muito muito ruim

COMO REFATORAR

O código deve ficar mais limpo.

Por que não ficou?

O código estava muito muito ruim

A refatoração virou algo muito grande

COMO REFATORAR

O código deve ficar mais limpo.

Por que não ficou?

O código estava muito muito ruim

A refatoração virou algo muito grande

Se pergunte:
O código está, de fato, melhor?

COMO REFATORAR

COMO REFATORAR

Não misture desenvolvimento de funcionalidades com refatoração

COMO REFATORAR

Não misture desenvolvimento de funcionalidades com refatoração

Separe as duas coisas

COMO REFATORAR

Não misture desenvolvimento de funcionalidades com refatoração

Separe as duas coisas

Pelo menos a nível de commit

COMO REFATORAR

Não misture desenvolvimento de funcionalidades com refatoração

Separe as duas coisas

Pelo menos a nível de commit

Refatoração != Desenvolvimento

COMO REFATORAR

COMO REFATORAR

Todos os testes anteriores devem passar

COMO REFATORAR

Todos os testes anteriores devem passar

Se você bagunçou algo, conserte

COMO REFATORAR

Todos os testes anteriores devem passar

Se você bagunçou algo, conserte

Se os testes estavam muito baixo nível...

COMO REFATORAR

Todos os testes anteriores devem passar

Se você bagunçou algo, conserte

Se os testes estavam muito baixo nível...

refatore os testes

COMO REFATORAR

Todos os testes anteriores devem passar

Se você bagunçou algo, conserte

Se os testes estavam muito baixo nível...

refatore os testes

Os testes devem passar

COMO REFATORAR

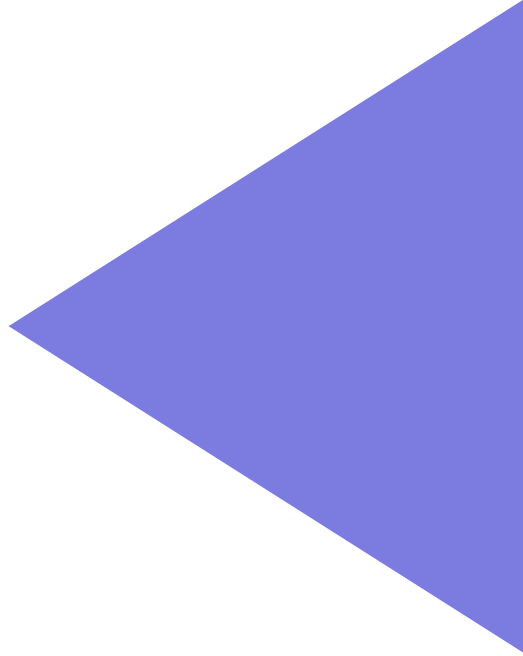
Se pergunte:
O código está, de fato, melhor?

Refatoração != Desenvolvimento

Os testes devem passar

BOAS PRÁTICAS

NOMES SIGNIFICATIVOS



NOMES SIGNIFICATIVOS

Nomes com intenção

Evite desinformação

Nomes pronunciáveis

Evite trocadilhos

Evite prefixos de métodos

Métodos devem ter verbos

Classes não devem conter verbos

Use o contexto

Uma palavra por conceito

NOMES SIGNIFICATIVOS

Nomes com intenção

Evite desinformação

Nomes pronunciáveis

Evite trocadilhos

Evite prefixos de métodos

Métodos devem ter verbos

Classes não devem conter verbos

Use o contexto

Uma palavra por conceito

```
int d;
```

NOMES SIGNIFICATIVOS

Nomes com intenção

Evite desinformação

Nomes pronunciáveis

Evite trocadilhos

Evite prefixos de métodos

Métodos devem ter verbos

Classes não devem conter verbos

Use o contexto

Uma palavra por conceito

```
int d;
```



Q?

NOMES SIGNIFICATIVOS

Nomes com intenção

Evite desinformação

Nomes pronunciáveis

Evite trocadilhos

Evite prefixos de métodos

Métodos devem ter verbos

Classes não devem conter verbos

Use o contexto

Uma palavra por conceito

```
int d;
```



Q?



Opções

NOMES SIGNIFICATIVOS

Nomes com intenção

Evite desinformação

Nomes pronunciáveis

Evite trocadilhos

Evite prefixos de métodos

Métodos devem ter verbos

Classes não devem conter verbos

Use o contexto

Uma palavra por conceito



Q?

```
int d;
```

```
int elapsedTimeInDays;
```

```
int daysSinceCreation;
```

```
int daysSinceModification;
```

```
int fileAgeInDays;
```



Opções

NOMES SIGNIFICATIVOS

Nomes com intenção

Evite desinformação

Nomes pronunciáveis

Evite trocadilhos

Evite prefixos de métodos

Métodos devem ter verbos

Classes não devem conter verbos

Use o contexto

Uma palavra por conceito

NOMES SIGNIFICATIVOS

Nomes com intenção

Evite desinformação

Nomes pronunciáveis

Evite trocadilhos

Evite prefixos de métodos

Métodos devem ter verbos

Classes não devem conter verbos

Use o contexto

Uma palavra por conceito

NOMES SIGNIFICATIVOS

Nomes com intenção

Evite desinformação

Nomes pronunciáveis

Evite trocadilhos

Evite prefixos de métodos

Métodos devem ter verbos

Classes não devem conter verbos

Use o contexto

Uma palavra por conceito

```
int a = l;  
if ( 0 == l )  
a = 01;  
else  
l = 01;
```

NOMES SIGNIFICATIVOS

Nomes com intenção

Evite desinformação

Nomes pronunciáveis

Evite trocadilhos

Evite prefixos de métodos

Métodos devem ter verbos

Classes não devem conter verbos

Use o contexto

Uma palavra por conceito



Q?

```
int a = l;  
if ( 0 == l )  
a = 01;  
else  
l = 01;
```


NOMES SIGNIFICATIVOS

Nomes com intenção

Evite desinformação

Nomes pronunciáveis

Evite trocadilhos

Evite prefixos de métodos

Métodos devem ter verbos

Classes não devem conter verbos

Use o contexto

Uma palavra por conceito



Q?

```
int a = l;  
if ( 0 == l )  
a = 01;  
else  
l = 01;
```

hp

NOMES SIGNIFICATIVOS

Nomes com intenção

Evite desinformação

Nomes pronunciáveis

Evite trocadilhos

Evite prefixos de métodos

Métodos devem ter verbos

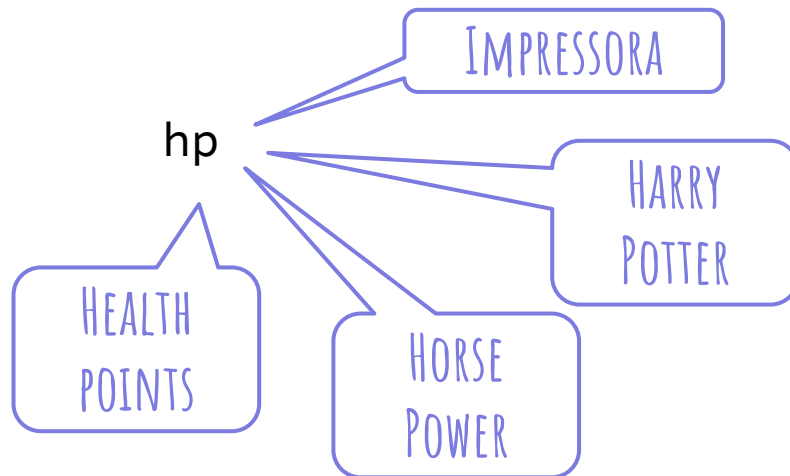
Classes não devem conter verbos

Use o contexto

Uma palavra por conceito

Q?

```
int a = l;  
if ( 0 == l )  
a = 01;  
else  
l = 01;
```



NOMES SIGNIFICATIVOS

Nomes com intenção

Evite desinformação

Nomes pronunciáveis

Evite trocadilhos

Evite prefixos de métodos

Métodos devem ter verbos

Classes não devem conter verbos

Use o contexto

Uma palavra por conceito

NOMES SIGNIFICATIVOS

Nomes com intenção

Evite desinformação

Nomes pronunciáveis

Evite trocadilhos

Evite prefixos de métodos

Métodos devem ter verbos

Classes não devem conter verbos

Use o contexto

Uma palavra por conceito

amdhms

NOMES SIGNIFICATIVOS

Nomes com intenção

Evite desinformação

Nomes pronunciáveis

Evite trocadilhos

Evite prefixos de métodos

Métodos devem ter verbos

Classes não devem conter verbos

Use o contexto

Uma palavra por conceito



NOMES SIGNIFICATIVOS

Nomes com intenção

Evite desinformação

Nomes pronunciáveis

Evite trocadilhos

Evite prefixos de métodos

Métodos devem ter verbos

Classes não devem conter verbos

Use o contexto

Uma palavra por conceito



NOMES SIGNIFICATIVOS

Nomes com intenção

Evite desinformação

Nomes pronunciáveis

Evite trocadilhos

Evite prefixos de métodos

Métodos devem ter verbos

Classes não devem conter verbos

Use o contexto

Uma palavra por conceito

NOMES SIGNIFICATIVOS

Nomes com intenção

Evite desinformação

Nomes pronunciáveis

Evite trocadilhos

Evite prefixos de métodos

Métodos devem ter verbos

Classes não devem conter verbos

Use o contexto

Uma palavra por conceito

NOMES SIGNIFICATIVOS

Nomes com intenção

Evite desinformação

Nomes pronunciáveis

Evite trocadilhos

Evite prefixos de métodos

Métodos devem ter verbos

Classes não devem conter verbos

Use o contexto

Uma palavra por conceito

NOMES SIGNIFICATIVOS

Nomes com intenção

Evite desinformação

Nomes pronunciáveis

Evite trocadilhos

Evite prefixos de métodos

Métodos devem ter verbos

Classes não devem conter verbos

Use o contexto

Uma palavra por conceito

Se o seu método precisa de prefixo para as variáveis, ele já está muito grande

NOMES SIGNIFICATIVOS

Nomes com intenção

Evite desinformação

Nomes pronunciáveis

Evite trocadilhos

Evite prefixos de métodos

Métodos devem ter verbos

Classes não devem conter verbos

Use o contexto

Uma palavra por conceito

NOMES SIGNIFICATIVOS

Nomes com intenção

Evite desinformação

Nomes pronunciáveis

Evite trocadilhos

Evite prefixos de métodos

Métodos devem ter verbos

Classes não devem conter verbos

Use o contexto

Uma palavra por conceito

Métodos são ações. Os verbos ajudam a descrever o comportamento

NOMES SIGNIFICATIVOS

Nomes com intenção

Evite desinformação

Nomes pronunciáveis

Evite trocadilhos

Evite prefixos de métodos

Métodos devem ter verbos

Classes não devem conter verbos

Use o contexto

Uma palavra por conceito

NOMES SIGNIFICATIVOS

Nomes com intenção

Evite desinformação

Nomes pronunciáveis

Evite trocadilhos

Evite prefixos de métodos

Métodos devem ter verbos

Classes não devem conter verbos

Use o contexto

Uma palavra por conceito

Classes não performam ações.

NOMES SIGNIFICATIVOS

Nomes com intenção

Evite desinformação

Nomes pronunciáveis

Evite trocadilhos

Evite prefixos de métodos

Métodos devem ter verbos

Classes não devem conter verbos

Use o contexto

Uma palavra por conceito

NOMES SIGNIFICATIVOS

Nomes com intenção

Evite desinformação

Nomes pronunciáveis

Evite trocadilhos

Evite prefixos de métodos

Métodos devem ter verbos

Classes não devem conter verbos

Use o contexto

Uma palavra por conceito

NOMES SIGNIFICATIVOS

Nomes com intenção

Evite desinformação

Nomes pronunciáveis

Evite trocadilhos

Evite prefixos de métodos

Métodos devem ter verbos

Classes não devem conter verbos

Use o contexto

Uma palavra por conceito

NOMES SIGNIFICATIVOS

Nomes com intenção

Evite desinformação

Nomes pronunciáveis

Evite trocadilhos

Evite prefixos de métodos

Métodos devem ter verbos

Classes não devem conter verbos

Use o contexto

Uma palavra por conceito

```
delete()  
remove()  
erase()
```

NOMES SIGNIFICATIVOS

Nomes com intenção

Evite desinformação

Nomes pronunciáveis

Evite trocadilhos

Evite prefixos de métodos

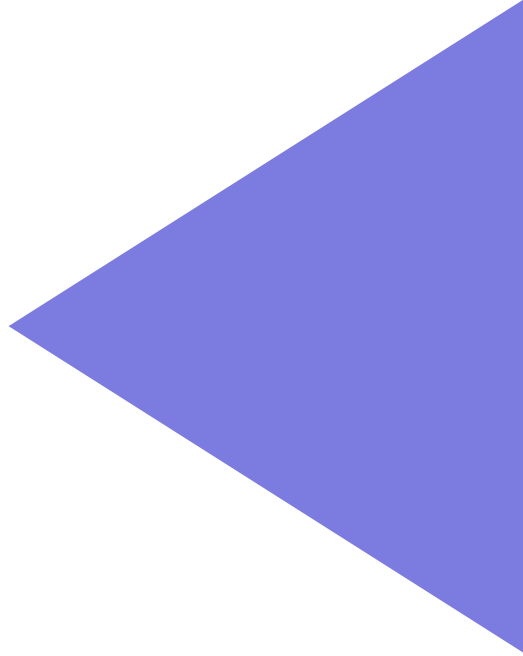
Métodos devem ter verbos

Classes não devem conter verbos

Use o contexto

Uma palavra por conceito

FUNÇÕES



FUNÇÕES

Pequenas!

Faça apenas uma coisa

Poucos argumentos

Sem efeitos colaterais

Fazer algo ou responder algo

Don't repeat yourself

FUNÇÕES

Pequenas!

Faça apenas uma coisa

Poucos argumentos

Sem efeitos colaterais

Fazer algo ou responder algo

Don't repeat yourself

Ainda menores!

FUNÇÕES

Pequenas!

Faça apenas uma coisa

Poucos argumentos

Sem efeitos colaterais

Fazer algo ou responder algo

Don't repeat yourself

FUNÇÕES

Pequenas!

Faça apenas uma coisa

Poucos argumentos

Sem efeitos colaterais

Fazer algo ou responder algo

Don't repeat yourself

Devem fazer uma coisa.
Fazê-la bem.
E fazer apenas isso.

FUNÇÕES

Pequenas!

Faça apenas uma coisa

Poucos argumentos

Sem efeitos colaterais

Fazer algo ou responder algo

Don't repeat yourself

FUNÇÕES

Pequenas!

Faça apenas uma coisa

Poucos argumentos

Sem efeitos colaterais

Fazer algo ou responder algo

Don't repeat yourself

Quantidade
mínima de
elementos

Fazendo mais
de uma coisa

FUNÇÕES

Pequenas!

Faça apenas uma coisa

Poucos argumentos

Sem efeitos colaterais

Fazer algo ou responder algo

Don't repeat yourself

FUNÇÕES

Pequenas!

Faça apenas uma coisa

Poucos argumentos

```
checkLoginInfo()  
----openSession()
```

Sem efeitos colaterais

Fazer algo ou responder algo

Don't repeat yourself

FUNÇÕES

Pequenas!

Faça apenas uma coisa

Poucos argumentos

Sem efeitos colaterais

Fazer algo ou responder algo

Don't repeat yourself

FUNÇÕES

Pequenas!

Faça apenas uma coisa

Poucos argumentos

Sem efeitos colaterais

Fazer algo ou responder algo

Don't repeat yourself

Ou muda o estado do objeto ou
retorna uma informação
relevante

FUNÇÕES

Pequenas!

Faça apenas uma coisa

Poucos argumentos

Sem efeitos colaterais

Fazer algo ou responder algo

Don't repeat yourself

FUNÇÕES

Pequenas!

Faça apenas uma coisa

Poucos argumentos

Sem efeitos colaterais

Fazer algo ou responder algo

Don't repeat yourself

Don't repeat yourself

FUNÇÕES

Pequenas!

Faça apenas uma coisa

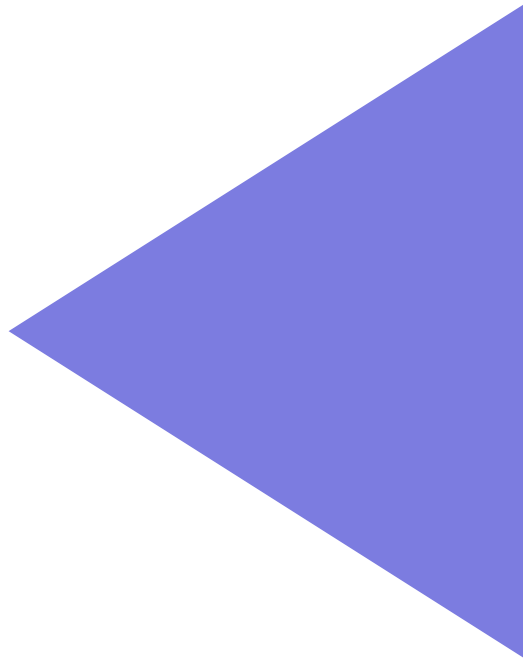
Poucos argumentos

Sem efeitos colaterais

Fazer algo ou responder algo

Don't repeat yourself

COMENTÁRIOS



COMENTÁRIOS



"Now! ... *That* should clear up
a few things around here!"

COMENTÁRIOS

Comentários não
resolvem código ruim



"Now! ... That should clear up
a few things around here!"

COMENTÁRIOS

Código auto-explicativo

Comentários de licença

Explicação de intenção

TODO

Docstrings

COMENTÁRIOS

Código auto-explicativo

Comentários de licença

Explicação de intenção

TODO

Docstrings

COMENTÁRIOS

Código auto-explicativo

Comentários de licença

Explicação de intenção

TODO

Docstrings

```
// Check to see if the employee is eligible for full benefits  
if ((employee.flags & HOURLY_FLAG) && (employee.age > 65))
```

ou

```
if (employee.isEligibleForFullBenefits())
```

COMENTÁRIOS

Código auto-explicativo

Comentários de licença

Explicação de intenção

TODO

Docstrings

COMENTÁRIOS

Código auto-explicativo

Comentários de licença

Explicação de intenção

TODO

Docstrings

README

COMENTÁRIOS

Código auto-explicativo

Comentários de licença

Explicação de intenção

TODO

Docstrings

COMENTÁRIOS

Código auto-explicativo

Comentários de licença

Explicação de intenção

TODO

Docstrings

COMENTÁRIOS

Código auto-explicativo

Comentários de licença

Explicação de intenção

TODO

```
//This is our best attempt to get a race condition  
//by creating large number of threads.
```

Docstrings

COMENTÁRIOS

Código auto-explicativo

Comentários de licença

Explicação de intenção

TODO

Docstrings

COMENTÁRIOS

Código auto-explicativo

Comentários de licença

Explicação de intenção

TODO

Docstrings

Issue

Backlog

Documentação de dev

COMENTÁRIOS

Código auto-explicativo

Comentários de licença

Explicação de intenção

TODO

Docstrings

COMENTÁRIOS

Código auto-explicativo

Comentários de licença

Explicação de intenção

TODO

Docstrings

COMENTÁRIOS

Código auto-explicativo

Comentários de licença

Explicação de intenção

TODO

Docstrings

```
def complex(real=0.0, imag=0.0):  
    """Form a complex number.  
  
    Keyword arguments:  
    real -- the real part (default 0.0)  
    imag -- the imaginary part (default 0.0)  
    """  
  
    if imag == 0.0 and real == 0.0:  
        return complex_zero  
  
    ...
```

COMENTÁRIOS

Código auto-explicativo

Comentários de licença

Explicação de intenção

TODO

Docstrings