

A man wearing a grey hoodie, blue pants, and a grey cap is crouching on a dirt ground. He is holding a small, colorful object in his right hand. The background is a blurred, outdoor setting with a building and some debris.

**Pense 4 vezes  
antes de fazer  
herança**

**Leonardo Tegon - TDC Recife 2019**

***LEONARDO TEGON***

***twitter.com/tegonl***

***github.com/tegon***



***BIRIGUI***





# ***HERANÇ***

***PROGRAMAÇÃO  
ORIENTADA A OBJETOS***

***REPRESENTAÇÃO DO  
MUNDO REAL***



```
class Animal  
  def andar  
  end  
  def falar  
  end  
end
```

```
class Cachorro < Animal  
end
```

```
class Gato < Animal  
end
```



***PENSE EM UMA FRUTA***



pitaya desenho laranja tamara cereja melancia mirtilo cupuaçu lichia graviola figo ingá pinha maracuja amora ➔



Diferença entre fruto e fruta - Brasil Escola  
brasilecola.uol.com.br



Fruta. Os benefícios que previnem doenças e não...  
sol.sapo.pt



Fruta não engorda se consumida com moderação:...  
globoesporte.globo.com



Conheça os tipos de frutas e seus benefícios para...  
jaminealimentos.com



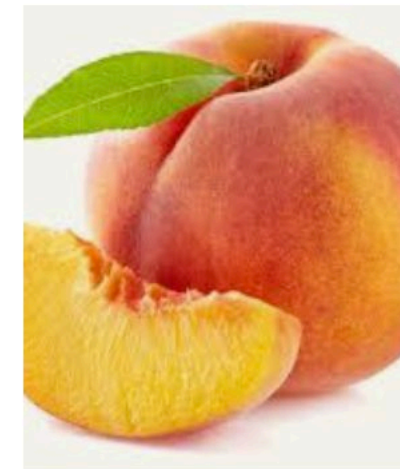
Frutas: conheça os benefícios das frutas ...  
brasilecola.uol.com.br



Saiba tudo sobre pitaya, a fruta do ...  
saude.abril.com.br



Mamão, uma fruta poderosa | Saúde é ...  
saude.abril.com.br



Fruta ou fruto? - Aluno...  
alunosonline.uol.com.br



Frutas: lista de 50 frutas mais consumidas e seus ...  
todamateria.com.br


**#1. OBJETOS DEVEM  
SER MODELADOS DO  
PONTO DE VISTA DE  
NEGÓCIO**



**NETFLIX**



# BROOKLYN NINE-NINE

👍 2017 14 5 Seasons 

Watch Season 5 Now

S5:E15 "The Puzzle Master"

Jake brings Amy in on an arson case connected to her celebrity crush, a crossword puzzle author. The Nine-Nine gets a sweet new detective car.

▶ NEXT EPISODE

+ MY LIST



Creators: Daniel J. Goor, Michael Schur



**N** SERIES

# explained

2019 **14** 2 Seasons **AD**

**New Episode Coming Thursday**

"Cults"

How do cults lure people in and exert control? Learn a cult's telltale signs, and how loneliness and life online makes indoctrination easier than ever.

**▶ NEXT EPISODE**

**+ MY LIST**



**N** SERIES  
**PATRIOT**  
**ACT**  
WITH HASAN MINHAJ

82% Match 2019 **16** 4 Volumes

**Watch All 4 Volumes Now**

Every Sunday, Hasan Minhaj brings an incisive and nuanced perspective to global news, politics and culture in his unique comedy series.

**▶ PLAY**

**+ MY LIST**



**Starring:** Hasan Minhaj

**Genres:** Political TV Shows, TV Comedies, US TV Shows

**This show is:** Cynical, Witty, Irreverent



Disney PIXAR  
**DIVERTIDA  
MENTE**

👍 2015 L 1h 34m

When young Riley's family moves across the country, her inner Joy corrals her other feelings into helping her cope with a new life -- and growing up.

▶ **PLAY**

✓ **MY LIST**



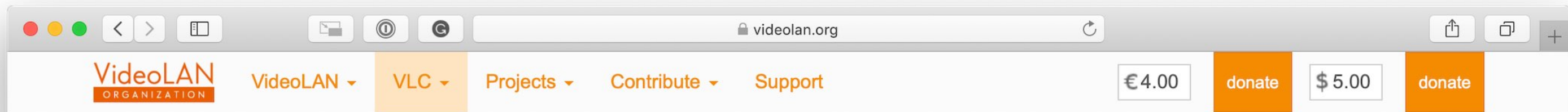
This wild ride through the wonderland of human emotions won an Academy Award and a Golden Globe for best animated film.

**Starring:** Amy Poehler, Phyllis Smith, Richard Kind

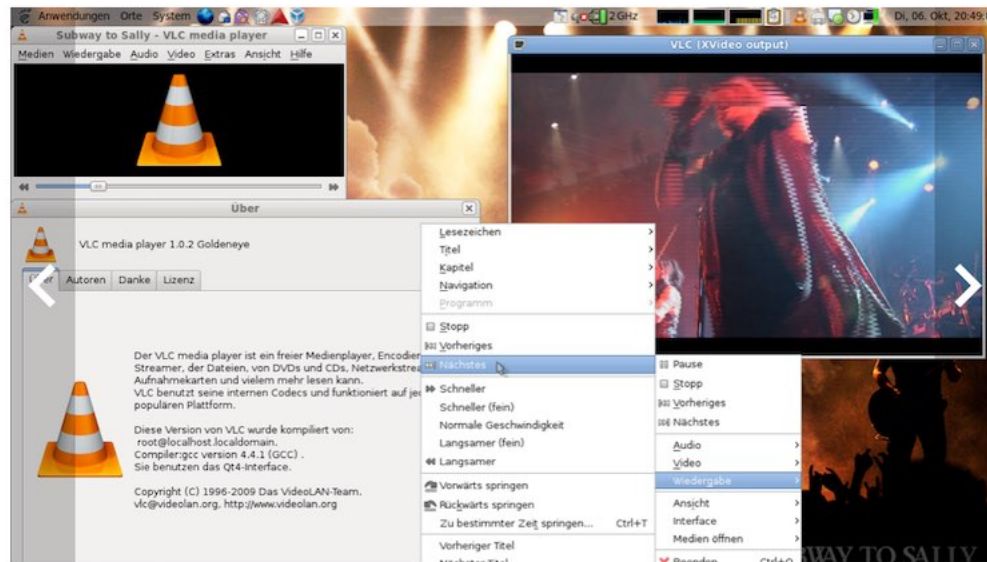
**Genres:** Family Comedies, Children & Family Movies, Disney Movies

**This movie is:** Family-friendly, Emotional





VideoLAN, a project and a **non-profit organization**.



## VLC media player

VLC is a free and open source cross-platform multimedia player and framework that plays most multimedia files as well as DVDs, Audio CDs, VCDs, and various streaming protocols.

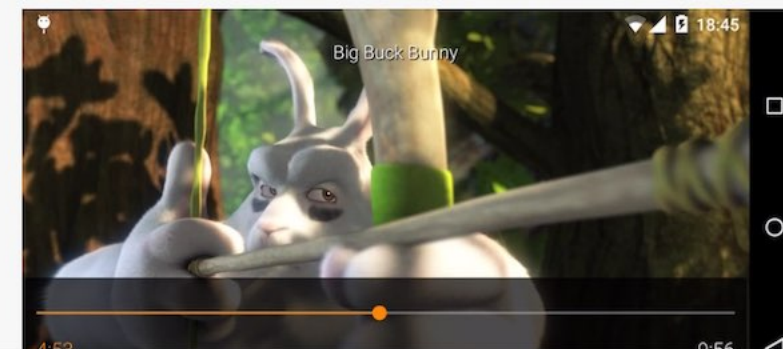
[Download VLC](#)

Version 3.0.8 • macOS • 48 MB  
36,763,768 downloads so far



### Simple, fast and powerful

- ✓ **Plays everything** - Files, Discs, Webcams, Devices and Streams.
- ✓ **Plays most codecs with no codec packs needed** - MPEG-2, MPEG-4, H.264, MKV, WebM, WMV, MP3...
- ✓ **Runs on all platforms** - Windows, Linux, Mac OS X, Unix, iOS, Android ...



# **#2. PRINCÍPIO DE SUBSTITUIÇÃO DE LISKOV**

**O MESMO CÓDIGO QUE  
FUNCIONA COM UM  
TIPO, DEVE FUNCIONAR  
COM UM SUBTIPO**

```
passaros.each do |passaro|  
  passaro.voar  
end
```



```
passaros.each do |passaro|  
  passaro.voar  
end
```

⇒ **NotImplementedError: pinguins não voam**

```
passaros.each do |passaro|  
  passaro.voar unless passaro.is_a?(Pinguim)  
end
```

***MENSAGENS QUE  
RECEBE E ENVIA***

***FORTE ACOPLAMENTO***

**#3. NÃO DEVE SER  
SOMENTE PARA  
REUTILIZAR CÓDIGO**

```
class TextField
  def initialize(name, size, value)
    @name, @size, @value = name, size, value
  end

  def render
    ...
  end
end
```

```
class TextField
  def initialize(name, size, value)
    @name, @size, @value = name, size, value
  end

  def render
    ...
  end
end
```

```
class DateField < TextField
  def initialize(name, size, value)
    super(name, size, format_date(value))
  end

  def format_date(value)
    value.strftime("%Y-%m-%d")
  end
end
```



```
class DateField < TextField
  def initialize(name, size, value)
    super(name, size, format_date(value))
  end

  def format_date(value)
    value.strftime("%Y-%m-%d")
  end
end
```

```
class DateField < TextField
  def initialize(name, size, value)
    super(name, size, format_date(value))
  end

  def format_date(value)
    value.strftime("%Y-%m-%d")
  end
end
```

```
class TimeField < DateField
  def format_date(value)
    value.strftime("%T.%L")
  end
end
```

```
class TimeField < DateField
  def format_date(value)
    value.strftime("%T.%L")
  end
end
```

***CUIDADO COM  
ESPECIALIZAÇÕES  
PEQUENAS***

```
class TimeField < DateField
  def format_date(value)
    ...
  end
end
```

***CUIDADO AO REMOVER  
COMPORTAMENTOS***

```
class DateField < TextField
  def initialize(name, size, value)
    super(name, size, format_date(value))
  end

  def format_date(value)
    value.strftime("%Y-%m-%d")
  end
end
```



```
class DateField < TextField
  def initialize(name, value)
    super(name, nil, format_date(value))
  end

  def format_date(value)
    value.strftime("%Y-%m-%d")
  end
end
```



# COMPOSIÇÃO

```
class TextField
  def initialize(name, size, value, formatter)
    @name, @size = name, size
    @value = formatter.format(value)
  end

  def render
    ...
  end
end
```

```
class DateFormatter  
  def self.format(value)  
    value.strftime("%Y-%m-%d")  
  end  
end
```

***DON'T REPEAT***

***YOURSELF***



***DON'T REPEAT  
YOURSELF?***

# **#4. THE RULE OF *THREE***



***TRÊS LUGARES AO INVÉS  
DE DOIS***

# Refactoring

## Improving the Design of Existing Code

by Martin Fowler

(with Kent Beck)

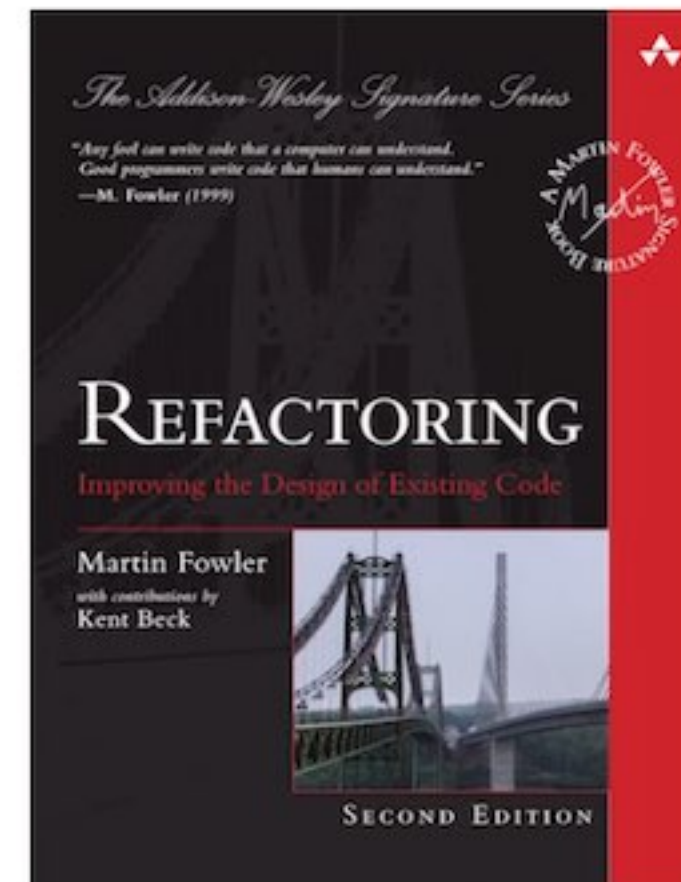
Refactoring is a controlled technique for improving the design of an existing code base. Its essence is applying a series of small behavior-preserving transformations, each of which "too small to be worth doing". However the cumulative effect of each of these transformations is quite significant. By doing them in small steps you reduce the risk of introducing errors. You also avoid having the system broken while you are carrying out the restructuring - which allows you to gradually refactor a system over an extended period of time.

My book describes the process of refactoring and spends most of its time explaining how to do the various refactorings - the behavior preserving transformations. The book opens with a simple example that describes the whole process. There are then some introductory chapters that discuss broader issues around refactoring, the "code smells" that

See on

amazon

informIT



**REGRA NÃO, GUIA**

**Duplication is far cheaper  
than the wrong abstraction**

**– Sandi Metz**



***RECONSIDERE A  
ABORDAGEM***

**JUNTE SEU TIME E  
CONVERSE SOBRE O  
DOMÍNIO**

***JUNTE SEU TIME E  
CONVERSE SOBRE OS  
CASOS DE USO***



**DEIXE AS COISAS IGUAIS  
PARA ACHAR O QUE É  
DIFERENTE**

**A chegada de uma nova  
feature que não encaixa na  
abstração atual é o melhor  
momento para refatorar**

***VALEU!!***

***twitter.com/tegonl***

***github.com/tegon***

# **REFERÊNCIAS**

- <https://thoughtbot.com/blog/reusable-oo-composition-vs-inheritance>
- <https://medium.com/@rdsbhas/10-modern-software-engineering-mistakes-bc67fbef4fc8>
- <https://www.bennadel.com/blog/2483-object-thinking-by-david-west.htm>
- <https://www.bennadel.com/blog/3108-elegant-objects-by-yegor-bugayenko.htm>
- <https://www.sandimetz.com/blog/2016/1/20/the-wrong-abstraction>
- <https://twitter.com/sandimetz/status/496627913010470912?lang=en>
- <https://www.youtube.com/watch?v=8bZh5LMaSmE>
- <https://www.youtube.com/watch?v=29MAL8pJImQ>
- [https://www.youtube.com/watch?v=\\_f2LYPpueAY](https://www.youtube.com/watch?v=_f2LYPpueAY)
- <https://www.youtube.com/watch?v=CXyNZYD007Q>

# **REFERÊNCIAS**

- <https://www.deconstructconf.com/2018/sandi-metz-polly-want-a-message>
- <https://programmingisterrible.com/post/176657481103/repeat-yourself-do-more-than-one-thing-and>
- <https://www.thoughtworks.com/insights/blog/composition-vs-inheritance-how-choose>
- <https://www.rubypigeon.com/posts/refactoring-inheritance-composition-data/>
- <https://www.tomdalling.com/blog/software-design/solid-class-design-the-liskov-substitution-principle/>
- <https://super.abril.com.br/mundo-estranho/tomate-e-fruta/>
- [https://en.wikipedia.org/wiki/Ruleofthree\(computerprogramming\)](https://en.wikipedia.org/wiki/Ruleofthree(computerprogramming))
- <https://en.wikipedia.org/wiki/Liskovsubstitutionprinciple>
- <https://pt.wikipedia.org/wiki/Herança>

# ***CRÉDITOS DAS FOTOS***

- Olhar Angolano: <https://unsplash.com/photos/95fbuDeudRg>
- Sergio Souza: <https://unsplash.com/photos/gTSi041JpVc>
- Jason Roberts: <https://unsplash.com/photos/6AQY7p01IS0>
- Ricardo Viana: <https://unsplash.com/photos/-tYsPFKMm7g>
- Alex Ghizila: [https://unsplash.com/photos/UD\\_j10SKj5g](https://unsplash.com/photos/UD_j10SKj5g)
- Thibault Penin: <https://unsplash.com/photos/3HInbCmQ8ro>
- Dmitry Bayer: <https://unsplash.com/photos/c5Nzpj1sn4>
- Michał Parzuchowski: <https://unsplash.com/photos/geNNFqfvw48>
- Perry Grone: <https://unsplash.com/photos/lbLgFFIADrY>