CONTRACTS.RUBY

Contracts for Ruby

@dodecaphonic

Come linguagens de café-da-manhã
É o pai do Mathias (que hoje faz dois anos)
Usa Emacs com toda sua ginga morena

A INDÚSTRIA DE SOFTWARE PRODUZ RESULTADOS DE BAIXA QUALIDADE

Volkswagen delays recall of 160,000 cars due to software glitch

By Reuters | 12 Apr, 2016, 12.06AM IST Post a Comment

READ MORE ON » Volkswagen | Skoda | Passat | German Federal Motor Transport Authority | europe

HAMBURG: Volkswagen will slightly delay the reca
cars in E
aimed a
work, it s

Checks
Transpo
consump
engines
update t
for the c

Software Bug Contributed to Blackout

Kevin Poulsen, SecurityFocus 2004-02-11

A previously-unknown software flaw in a widely-deplo
the devastating scope of the August 14th northeaster

Knight Capital Reports Net Loss After Software Error

by Whitney Kisling

October 17, 2012 – 12:58 PM BRT

Heartbleed

From Wikipedia, the free encyclopedia

For other uses, see Heartbleed (disambiguation).

Heartbleed is a security bug disclosed in April 2014 in the OpenSSL cryptography library, which is a widely used implementation of the Transport L
may be exploited regardless of whether the party using a vulnerable OpenSSL instance for TLS is a server or a client. It results from improper input
check) in the implementation of the TLS heartbeat extension,[3] thus the bug's name derives from "heartbeat".[4] The vulnerability is classified as a b
data can be read than should be allowed.[6]

Heartbleed is registered in the Common Vulnerabilities and Exposures system as CVE-2014-0160.[5] The federal Canadian Cyber Incident Respons
advising system administrators about the bug.[7] A fixed version of OpenSSL was released on April 7, 2014, on the same day Heartbleed was publi

At the time of disclosure, some 17% (around half a million) of the Internet's secure web servers certified by trusted authorities were believed to be v
the servers' private keys and users' session cookies and passwords.[8][9][10][11][12] The Electronic Frontier Foundation,[13] Ars Technica,[14] and Bruc
bug "catastrophic". *Forbes* cybersecurity columnist Joseph Steinberg wrote, "Some might argue that [Heartbleed] is the worst vulnerability found (at
since commercial traffic began to flow on the Internet."[16]

A British Cabinet spokesman recommended that "People should take advice on changing passwords from the websites they use... Most websites h
placed to advise what action, if any, people need to take."[17] On the day of disclosure, the Tor Project advised anyone seeking "strong anonymity o
from the Internet entirely for the next few days while things settle."[18]

Medical Devices: The Therac-25*

Nancy Leveson
University of Washington

1 Introduction

Between June 1985 and January 1987, a computer-controlled radiation therapy machine, called the Therac-25, massively overdosed six people. These accidents have been described as the worst in the 35-year history of medical accelerators [6].

NÃO É ENGRAÇADO, MAS A GENTE NÃO SE TOCA E RI ASSIM MESMO

Programação Orientada a Gambiarras

Nota: *Seu navegador poderá apresentar uma certa lentidão durante o processamento e decodificação das ga*

Nota: *o trecho seguinte está "compactado" de modo a despoluir visualmente o contexto da página toda.*

> *Corrigir todos os problemas em 2 minutos!*

Você após colocar em comentário todas as linhas que estavam com problemas

> *ERRO! PASSEI POR AKI!!.*

Maker sobre POG.

> *Esta é a solução ótima, não temos tempo para isso. Faça a solução boa! O maior inimigo do bom é o ótimo.*

Líder de Projeto sobre POG

> *É só fazer do jeito que eu to falando que entregamos na data correta.*

Diretor TI POG sobre projeto com tempo recorde!

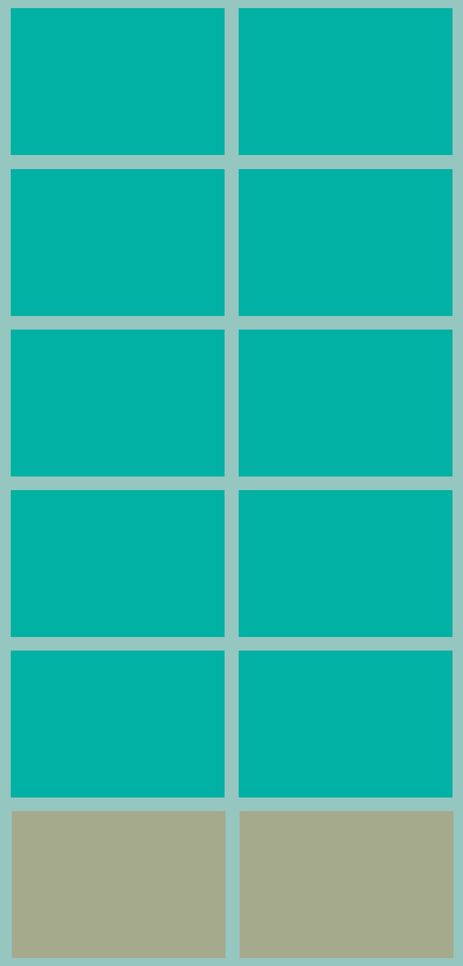
> *A culpa é do Hardware!*

desenvolvedor de software sobre POG

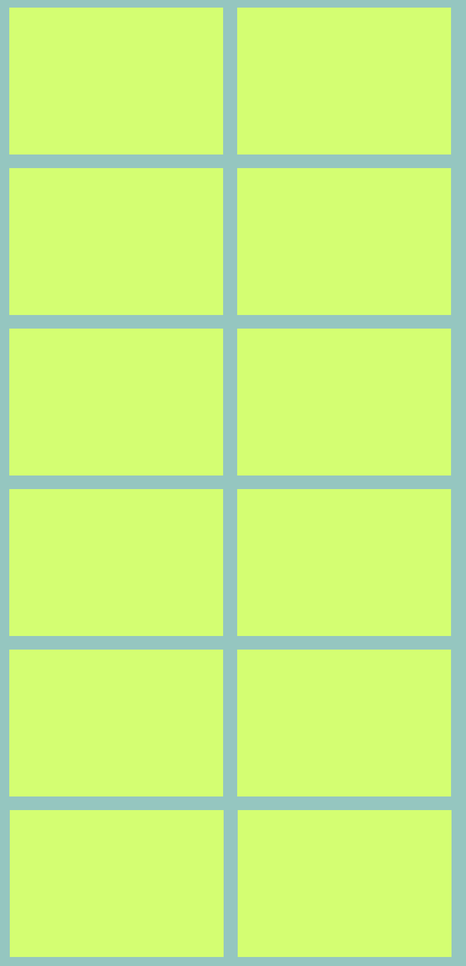
Índice [ocultar]

CUSTO X BENEFÍCIO

CUSTO X BENEFÍCIO

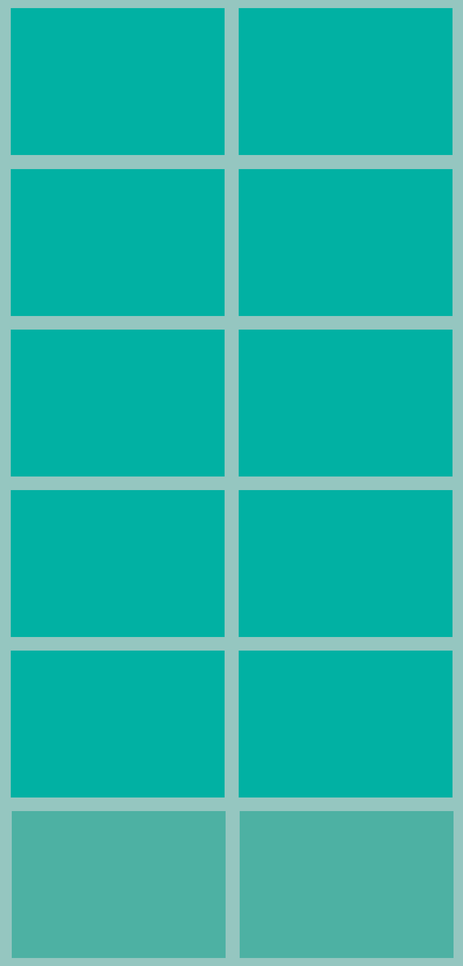


Qualidade

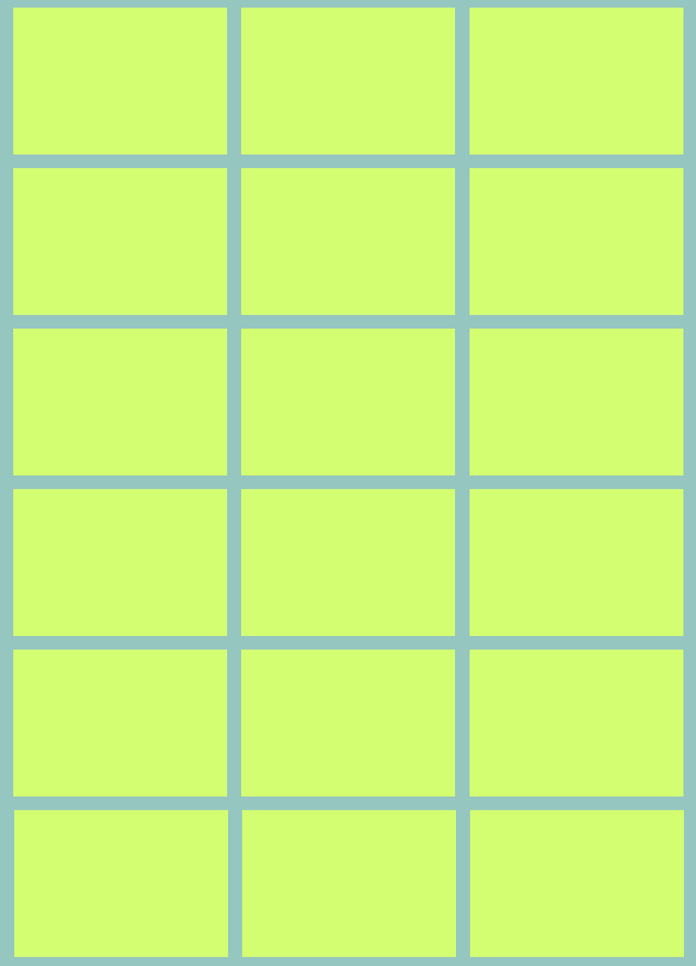


Custo Total

CUSTO X BENEFÍCIO

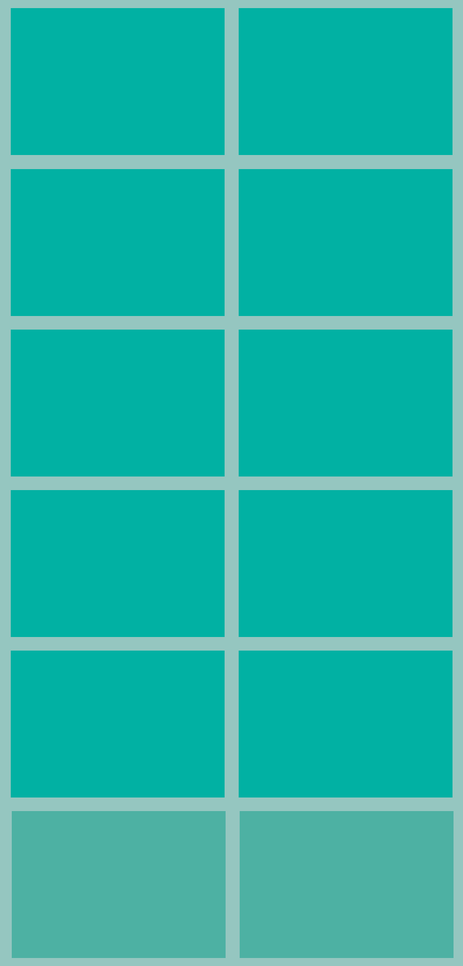


Qualidade

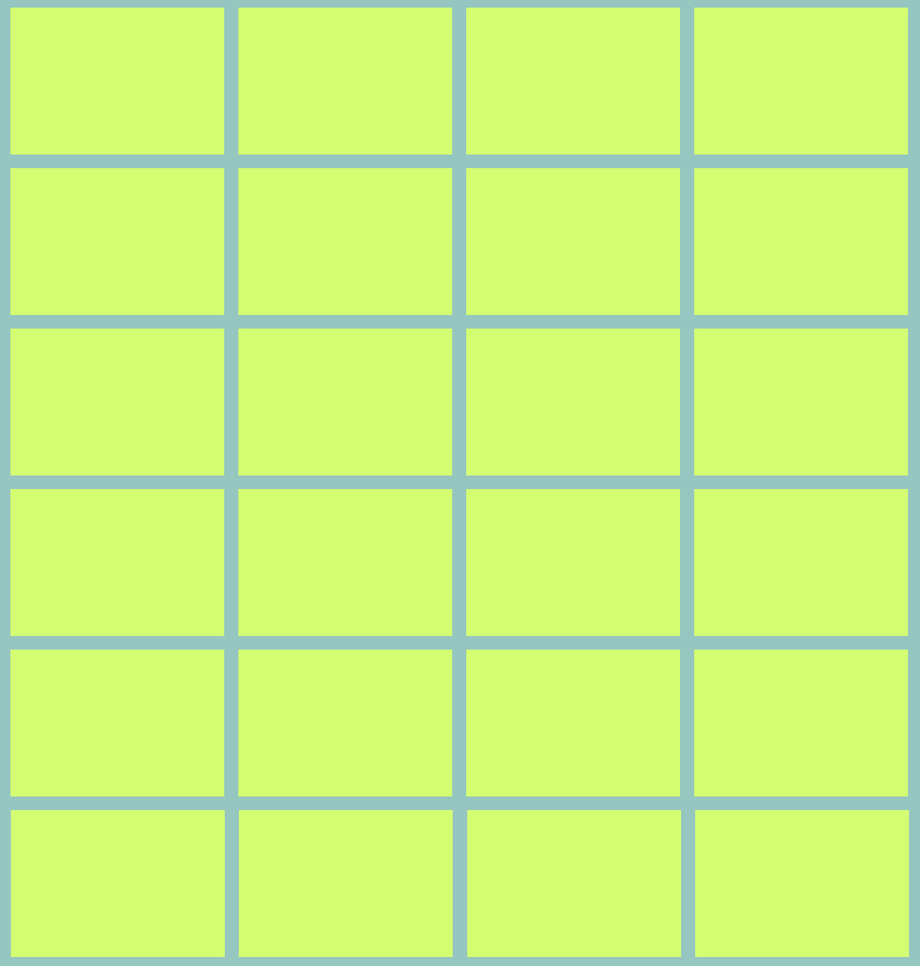


Custo Total

CUSTO X BENEFÍCIO

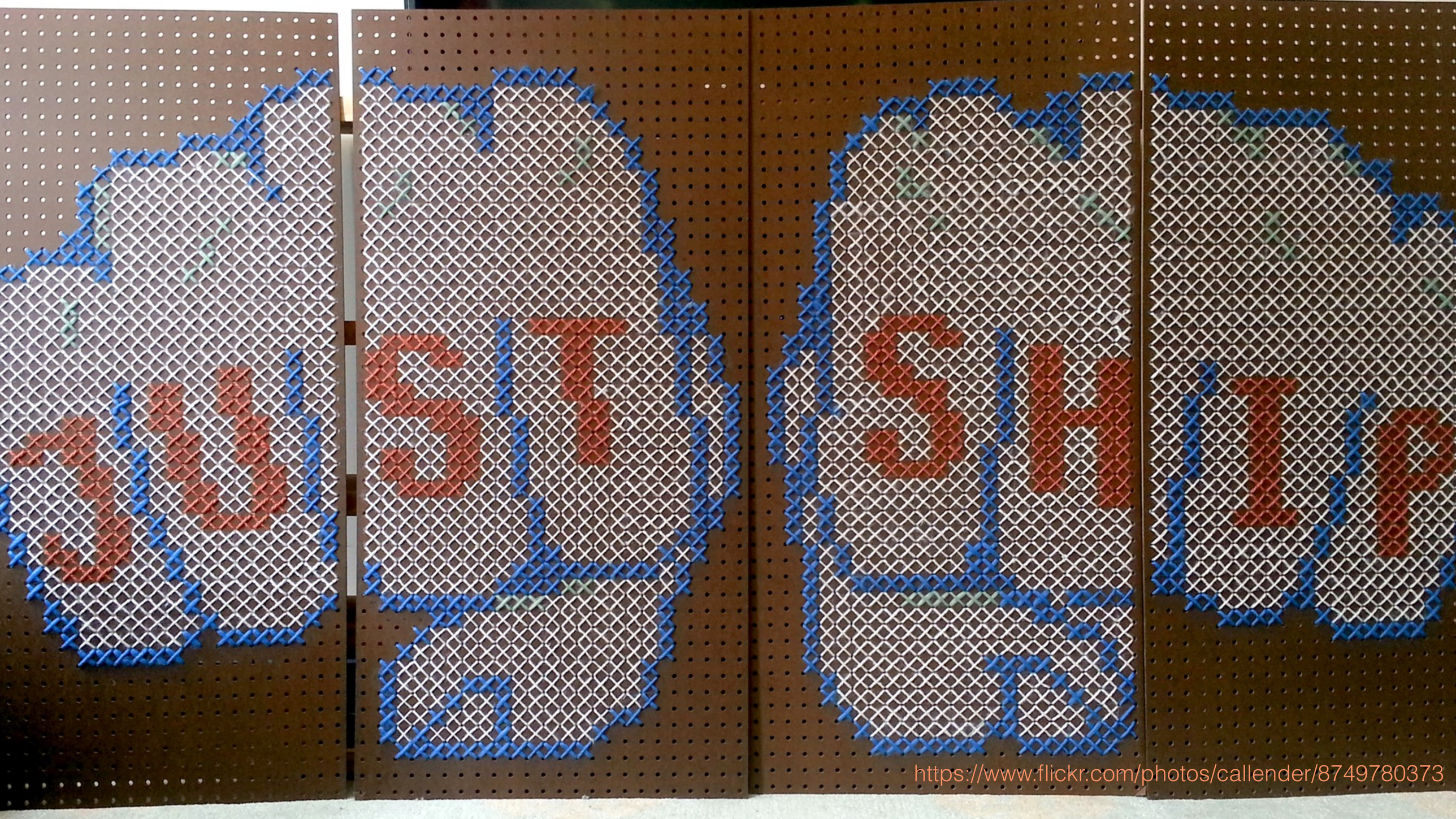


Qualidade



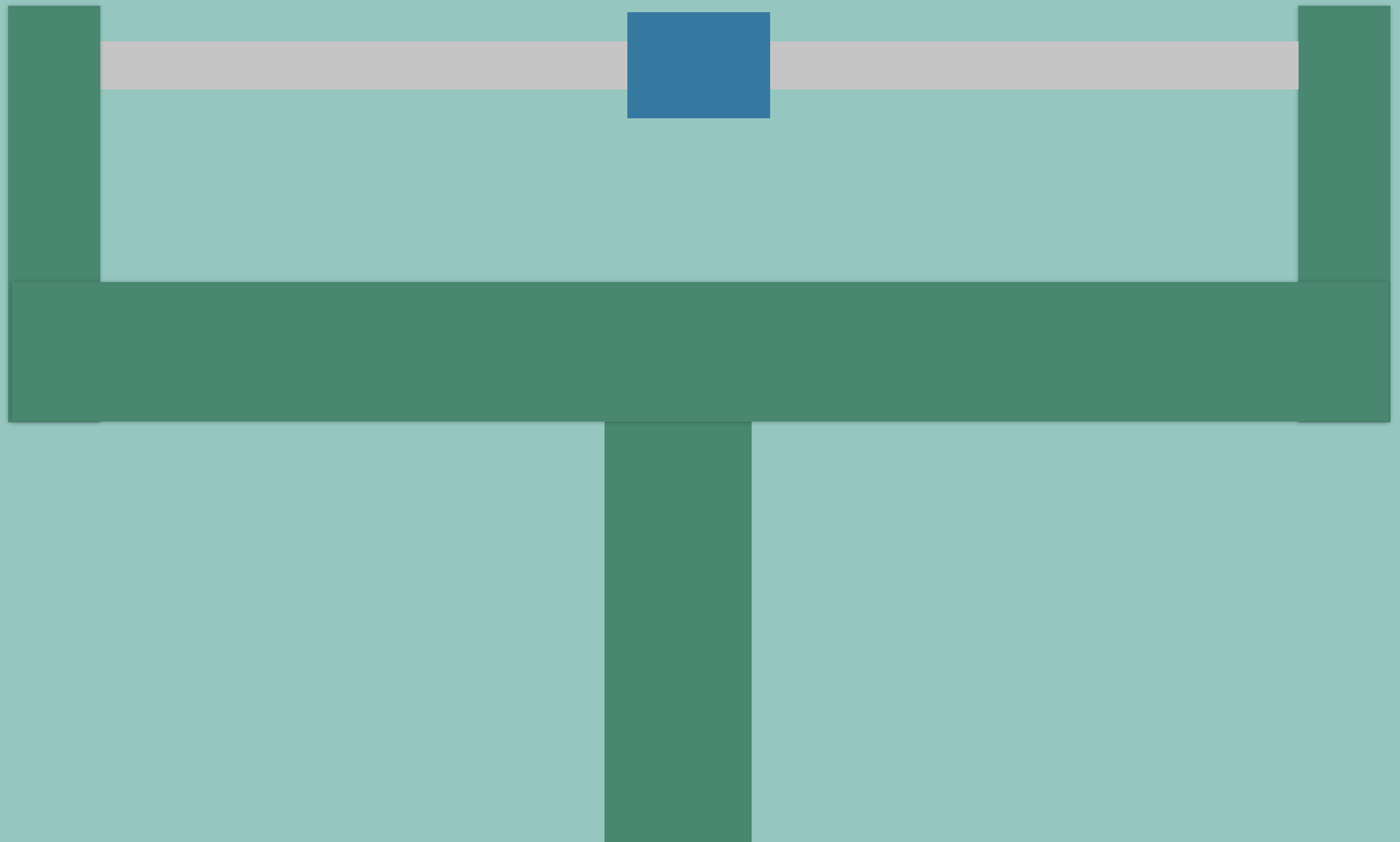
Custo Total

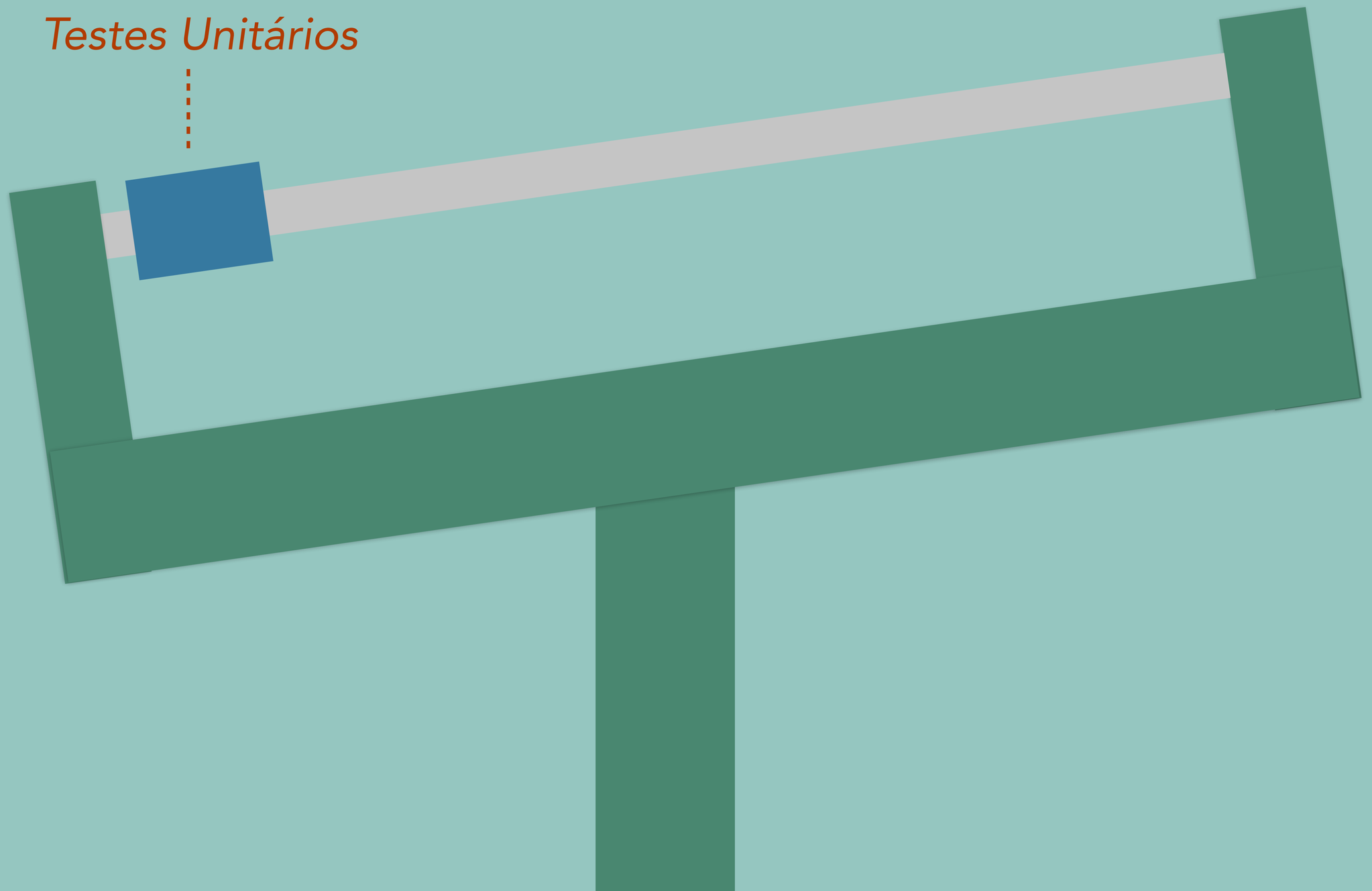
CULTURA DE STARTUPS = STATUS QUO


https://www.flickr.com/photos/callender/8749780373

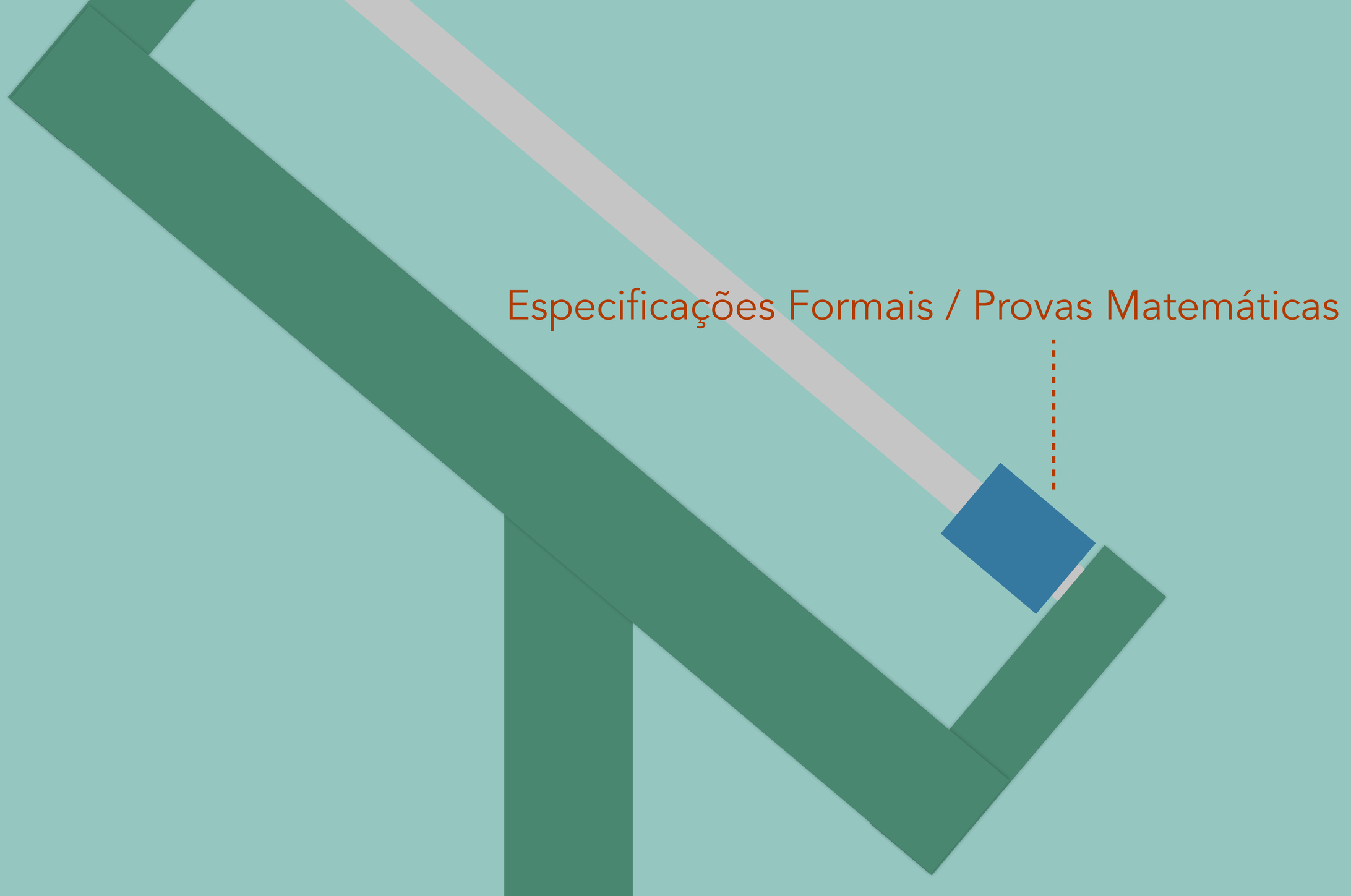
RUBY, PHP, PYTHON, JAVASCRIPT: PROCESSOS HUMANOS > COMPUTADORES E ALGORITMOS

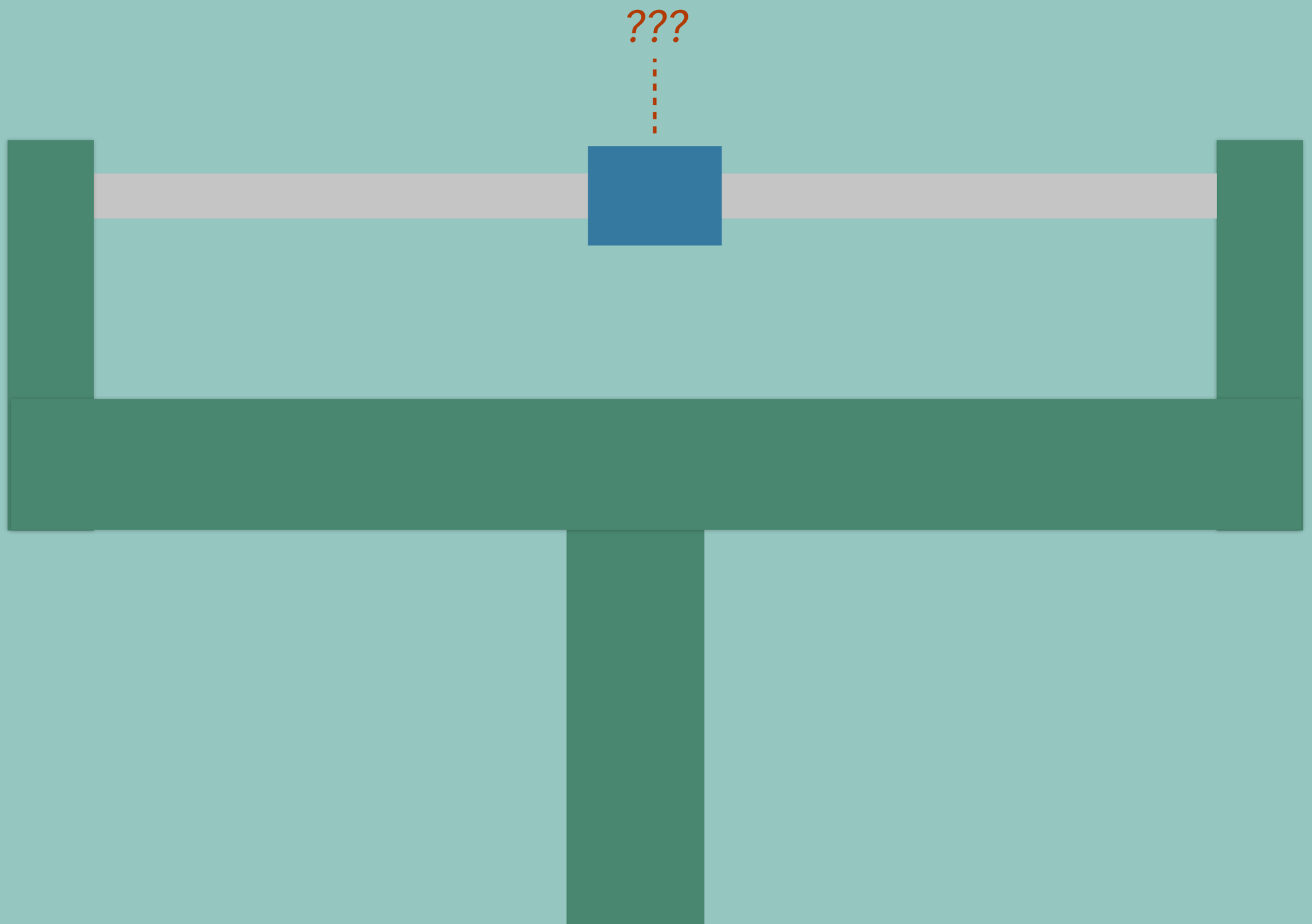
VERIFICAÇÃO FORMAL




Testes Unitários


Property-based testing


Especificações Formais / Provas Matemáticas



DESIGN BY CONTRACT

TIPOS DE ENTRADA

TIPOS DE ENTRADA

TIPOS DE SAÍDA

TIPOS DE ENTRADA

TIPOS DE SAÍDA

ERROS E EXCEÇÕES

TIPOS DE ENTRADA

TIPOS DE SAÍDA

ERROS E EXCEÇÕES

EFEITOS COLATERAIS

TIPOS DE ENTRADA

PRÉ-CONDIÇÕES

TIPOS DE SAÍDA

ERROS E EXCEÇÕES

EFEITOS COLATERAIS

TIPOS DE ENTRADA

PRÉ-CONDIÇÕES

TIPOS DE SAÍDA

PÓS-CONDIÇÕES

ERROS E EXCEÇÕES

EFEITOS COLATERAIS

TIPOS DE ENTRADA

PRÉ-CONDIÇÕES

TIPOS DE SAÍDA

PÓS-CONDIÇÕES

ERROS E EXCEÇÕES

INVARIANTES

EFEITOS COLATERAIS

TIPOS DE ENTRADA

PRÉ-CONDIÇÕES

TIPOS DE SAÍDA

PÓS-CONDIÇÕES

ERROS E EXCEÇÕES

INVARIANTES

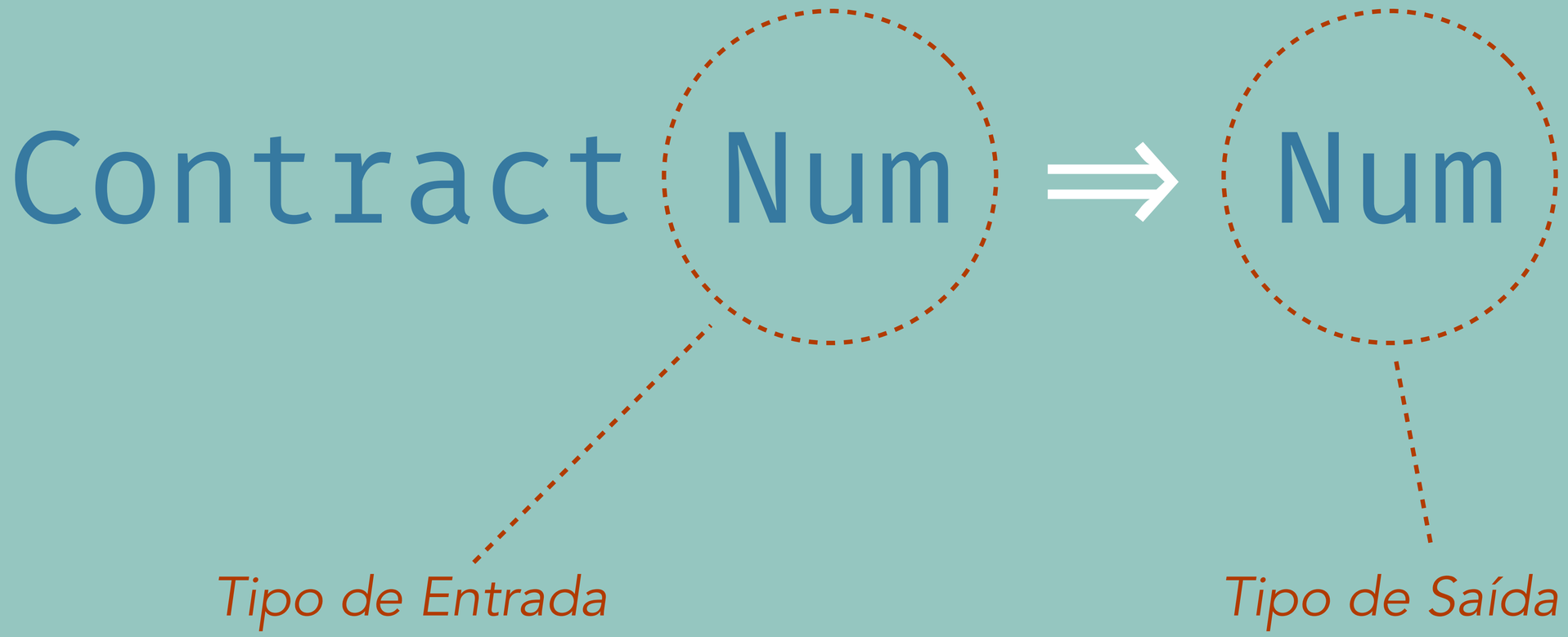
EFEITOS COLATERAIS

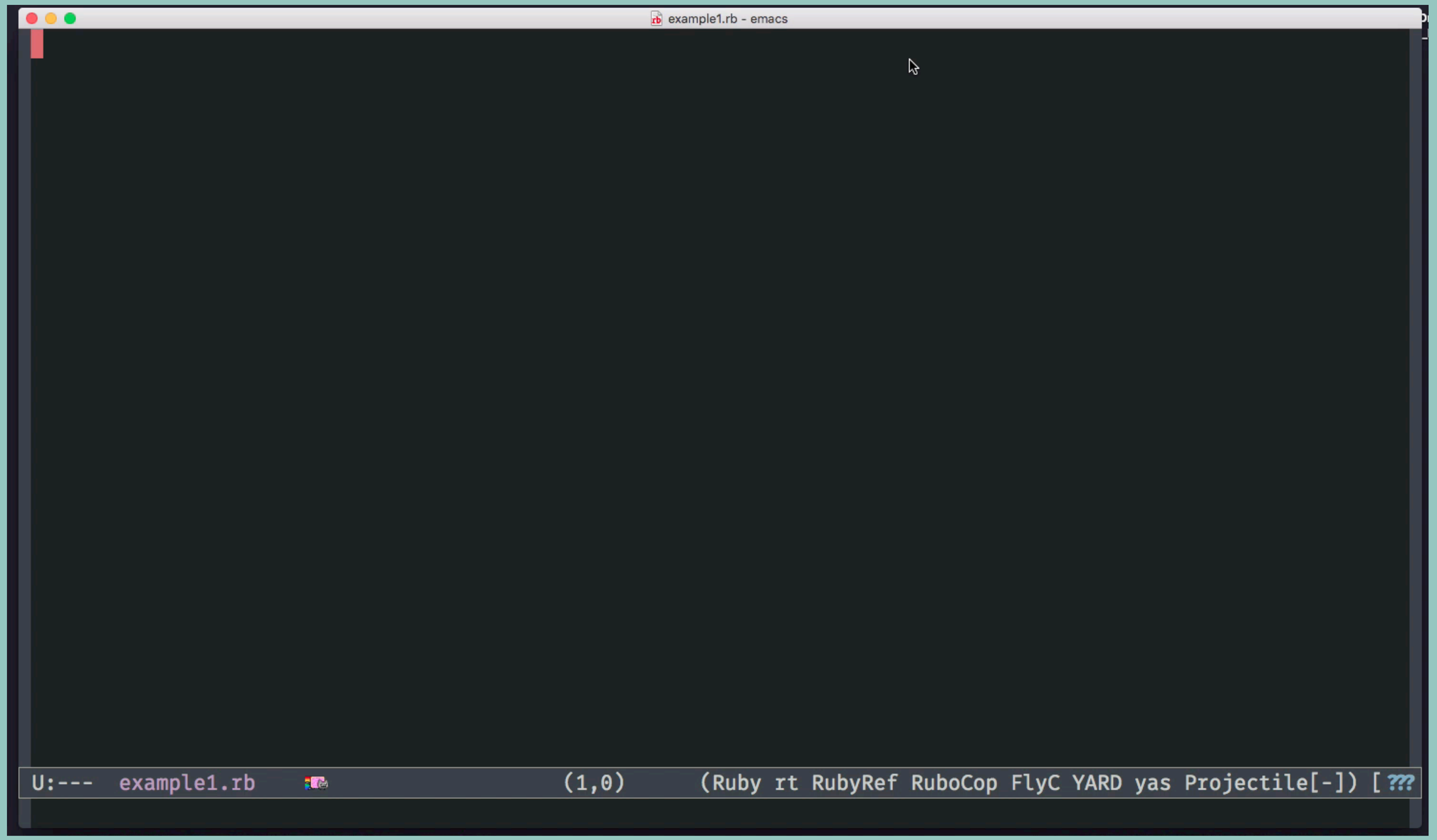
PERFORMANCE

```
put (x: ELEMENT; key: STRING) is
      -- Insert x so that it will be retrievable through key.
    require
      count ⩽ capacity
      not key.empty
    do
      ... Some insertion algorithm ...
    ensure
      has (x)
      item (key) = x
      count = old count + 1
    end
```

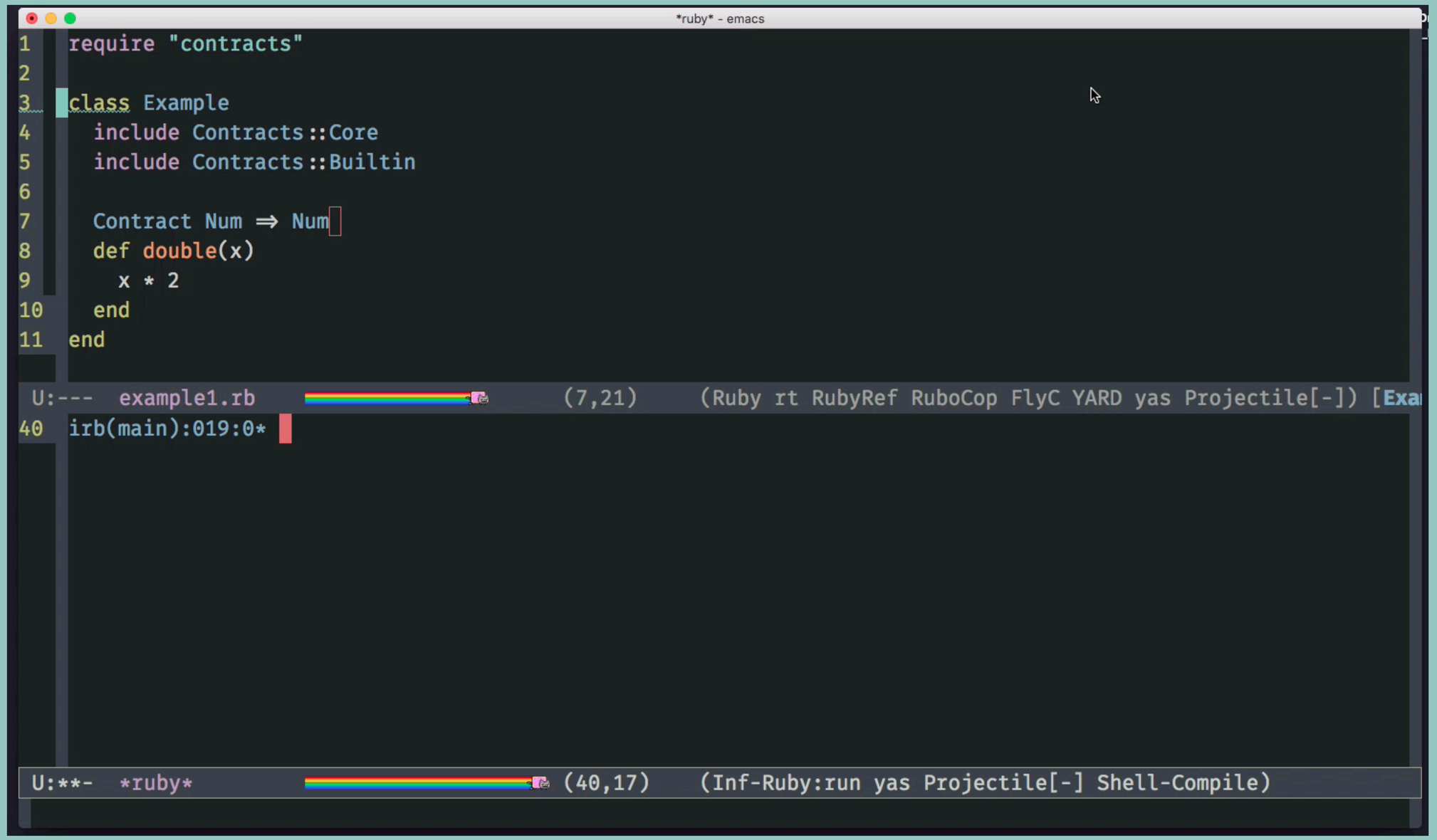
CHECAGEM EM TEMPO DE EXECUÇÃO

CONTRACTS.RUBY

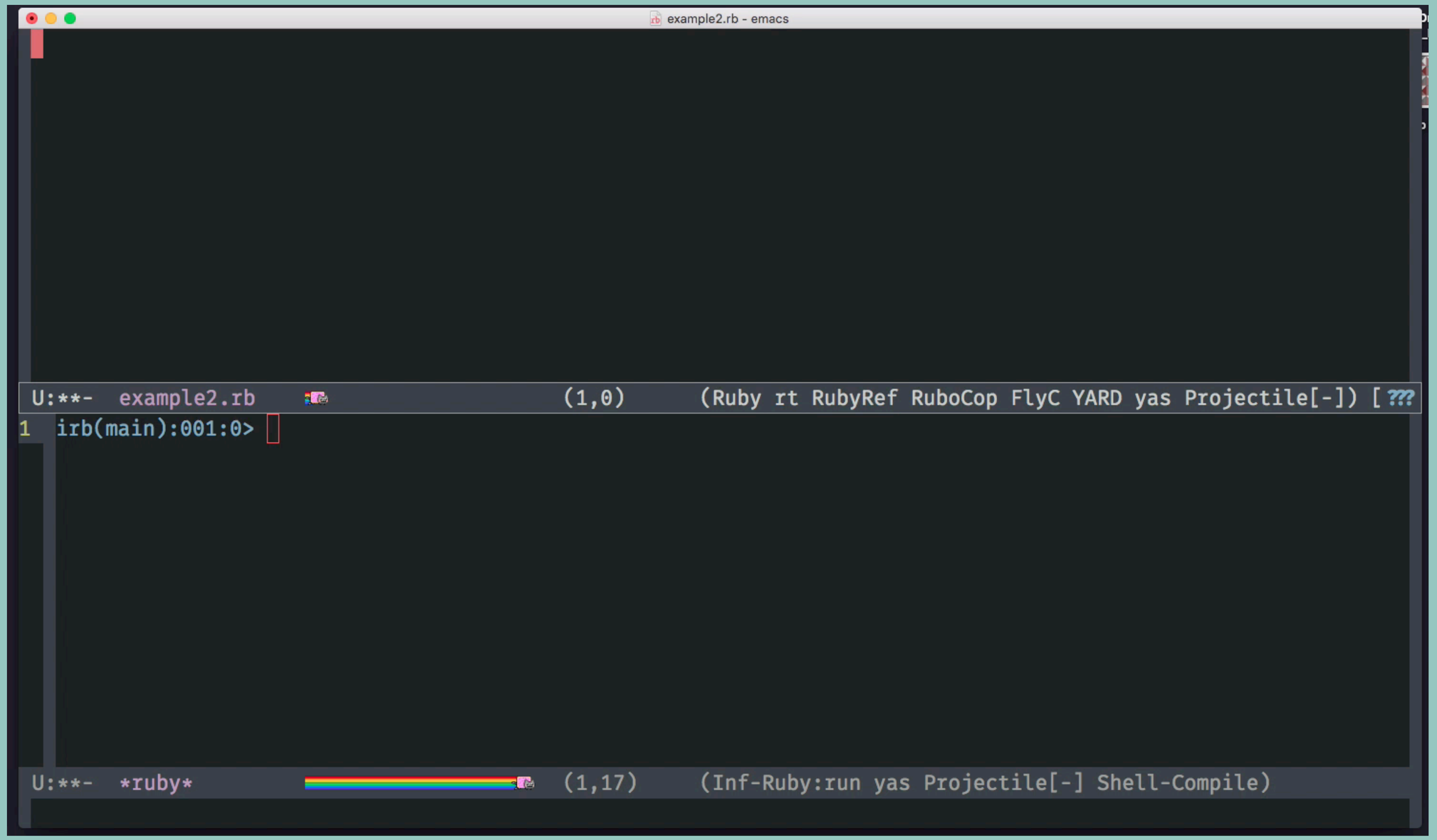

Contract
Num
⇒
Num
Tipo de Entrada
Tipo de Saída


example1.rb - emacs
U:--- example1.rb
(1,0)
(Ruby rt RubyRef RuboCop FlyC YARD yas Projectile[-]) [

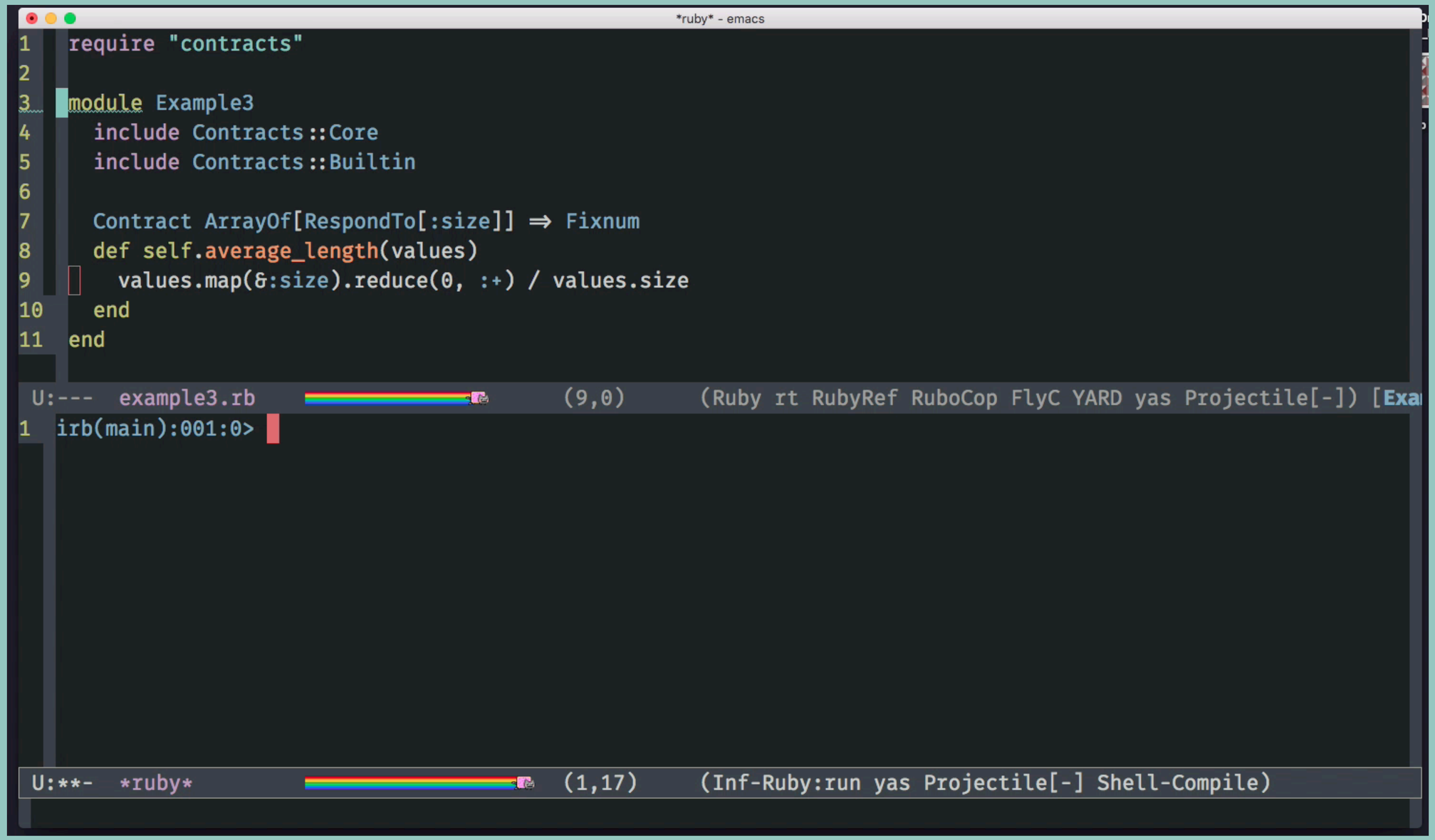
DOCUMENTAÇÃO


ruby - emacs
require "contracts"
class Example
include Contracts::Core
include Contracts::Builtin
Contract Num ⇒ Num
def double(x)
x * 2
end
end
U:--- example1.rb (7,21) (Ruby rt RubyRef RuboCop FlyC YARD yas Projectile[-])
irb(main):019:0*
U:**- *ruby* (40,17) (Inf-Ruby:run yas Projectile[-] Shell-Compile)

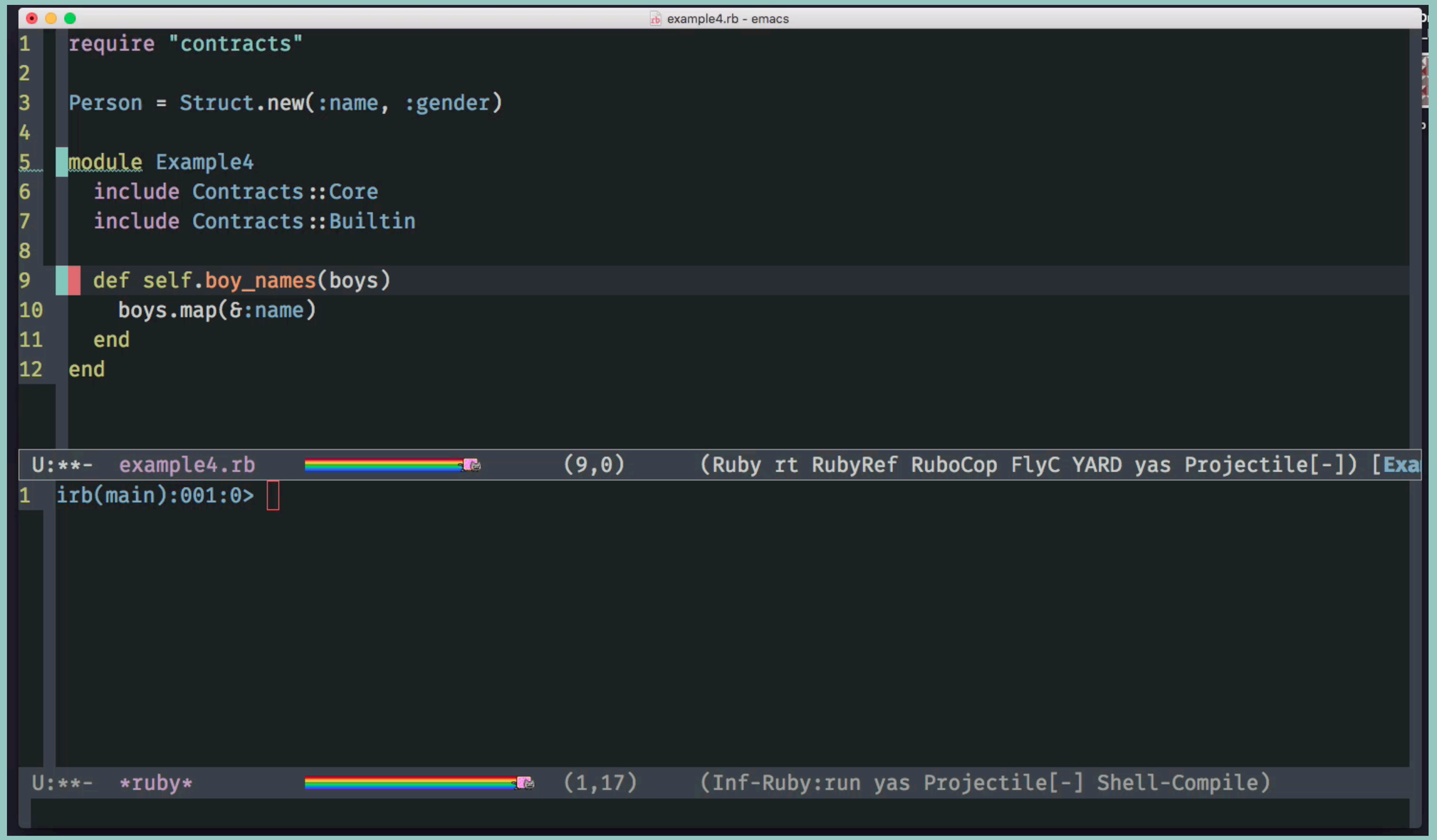
```
Contract ArrayOf[String] ⇒ Fixnum
def average_length(strings)
  strings.map(&:size).reduce(0, :+) / strings.size
end
```

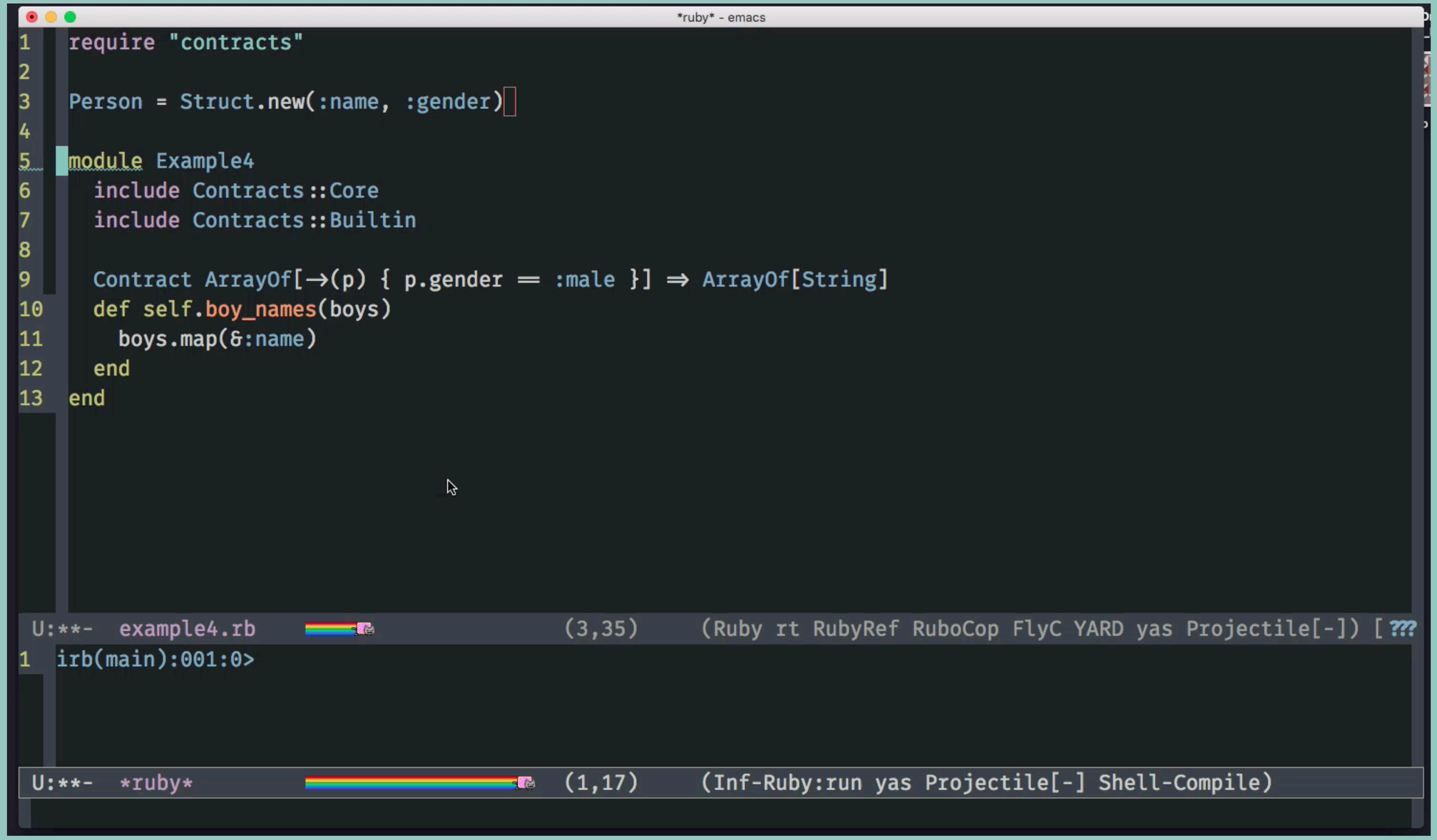

example2.rb - emacs
U:**- example2.rb (1,0) (Ruby rt RubyRef RuboCop FlyC YARD yas Projectile[-]) [???
1 irb(main):001:0>
U:**- *ruby* (1,17) (Inf-Ruby:run yas Projectile[-] Shell-Compile)

```
Contract ArrayOf[RespondTo[:size]] ⇒ Fixnum
def average_length(values)
  values.map(&:size).reduce(0, :+) / strings.size
end
```

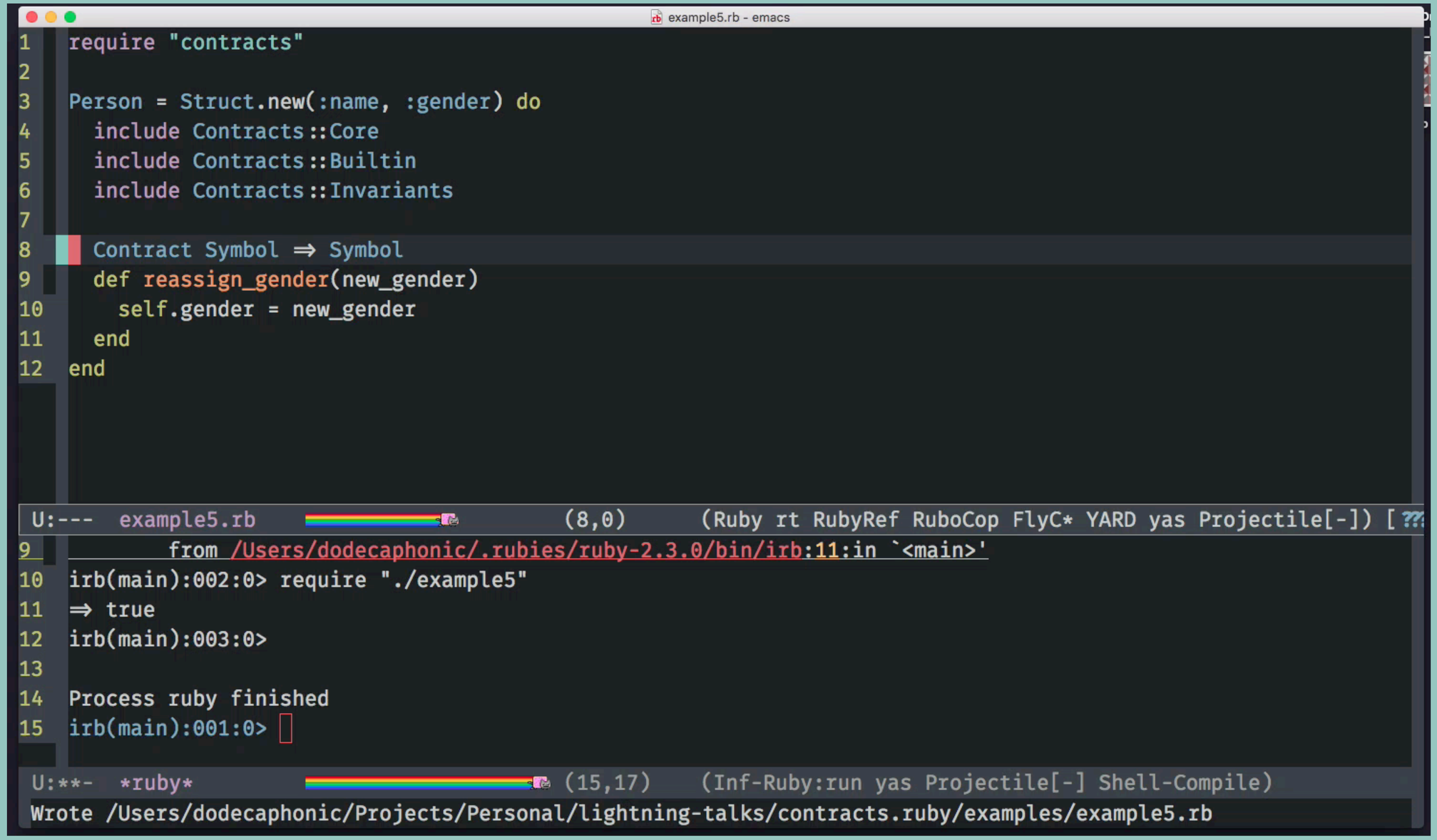

ruby - emacs
require "contracts"
module Example3
include Contracts::Core
include Contracts::Builtin
Contract ArrayOf[RespondTo[:size]] => Fixnum
def self.average_length(values)
values.map(&:size).reduce(0, :+) / values.size
end
end
U:--- example3.rb (9,0) (Ruby rt RubyRef RuboCop FlyC YARD yas Projectile[-])
irb(main):001:0>
U:**- *ruby* (1,17) (Inf-Ruby:run yas Projectile[-] Shell-Compile)

CONTRATOS PERSONALIZADOS


example4.rb - emacs
require "contracts"
Person = Struct.new(:name, :gender)
module Example4
include Contracts::Core
include Contracts::Builtin
def self.boy_names(boys)
boys.map(&:name)
end
end
U:**- example4.rb (9,0) (Ruby rt RubyRef RuboCop FlyC YARD yas Projectile[-]) [Exa
irb(main):001:0>
U:**- *ruby* (1,17) (Inf-Ruby:run yas Projectile[-] Shell-Compile)


ruby - emacs
require "contracts"
Person = Struct.new(:name, :gender)
module Example4
include Contracts::Core
include Contracts::Builtin
Contract ArrayOf[→(p) { p.gender == :male }] ⇒ ArrayOf[String]
def self.boy_names(boys)
boys.map(&:name)
end
end
U:**- example4.rb (3,35) (Ruby rt RubyRef RuboCop FlyC YARD yas Projectile[-]) [???
irb(main):001:0>
U:**- *ruby* (1,17) (Inf-Ruby:run yas Projectile[-] Shell-Compile)

INVARIANTES


example5.rb - emacs
require "contracts"
Person = Struct.new(:name, :gender) do
include Contracts::Core
include Contracts::Builtin
include Contracts::Invariants
Contract Symbol ⇒ Symbol
def reassign_gender(new_gender)
self.gender = new_gender
end
end
U:--- example5.rb (8,0) (Ruby rt RubyRef RuboCop FlyC* YARD yas Projectile[-])
from /Users/dodecaphonic/.rubies/ruby-2.3.0/bin/irb:11:in `<main>'
irb(main):002:0> require "./example5"
⇒ true
irb(main):003:0>
Process ruby finished
irb(main):001:0>
U:**- *ruby* (15,17) (Inf-Ruby:run yas Projectile[-] Shell-Compile)
Wrote /Users/dodecaphonic/Projects/Personal/lightning-talks/contracts.ruby/examples/example5.rb

MAIS DETALHES

https://en.wikipedia.org/wiki/Design_by_contract
https://www.eiffel.com/values/design-by-contract/introduction/
https://egonschiele.github.com/contracts.ruby

OBRIGADO.