

# Métodos e Gems

## Curso de Ruby - Métodos e Gems (Aula 5)



### Conteúdo

O conteúdo que será abordado na aula:

- **Métodos**
  - O que são?
  - Trabalhando com métodos
    - Como criar?
    - Parâmetros
    - Retorno
- **Gems**
  - O que são?
  - Trabalhando com gems
    - Como instalar uma gem
    - Novas Funcionalidades
    - Desinstalar uma gem
    - Informação
    - Bundler

MARCAR ESTA UNIDADE  
COMO CONCLUÍDA

## Métodos

### O que são?

Método é uma forma de organizar funções específicas de um programa. Caso necessário permite a reutilização de código, ou seja, evita escrever o mesmo código diversas vezes.

## Trabalhando com métodos

### Como criar?

1- Crie seu primeiro método em um arquivo chamado **method.rb**

```
----- Default
1 | def talk
2 |   puts 'Olá, como você está?'
3 | end
4 |
5 | talk
```

Para definir um método utiliza-se a palavra reservada **def** seguida pelo nome do método. Depois é escrito um conjunto de expressões, e por fim, a palavra **end** determina o término do método.

Para executar um método basta apenas escrever o seu nome.

### Parâmetros

1- Substitua o código de **method.rb** por

```
----- Default
1 | def talk(first_name, last_name)
2 |   puts "Olá #{first_name} #{last_name}, como você está?"
3 | end
4 |
5 | first_name = 'Leonardo'
6 | last_name = 'Scorza'
7 |
8 | talk(first_name, last_name)
```

Um método pode depender de um ou mais parâmetros para realizar determinada tarefa.

O exemplo define um método que precisa do primeiro e segundo nome para exibir uma frase. Tendo ciência disso, deve-se passar dois argumentos toda a vez que o método `talk` for chamado.

MARCAR ESTA UNIDADE  
COMO CONCLUÍDA



2- Crie um arquivo chamado **transito.rb** com o código

Default

```
1 def signal(color = 'vermelho')
2   puts "O sinal está #{color}"
3 end
4
5 signal
6
7 color = 'verde'
8 signal(color)
```

Definindo um valor padrão ao parâmetro, ele torna-se opcional.

Um método pode ser invocado mais de uma vez, em diferentes áreas do programa. Assim seu código está sendo reutilizado.

Quer mudar o código? Altere apenas o método!

## Retorno

1- Crie um arquivo chamado **return.rb**

Default

```
1 def compare(a, b)
2   a > b
3 end
4
5 a = 1
6 b = 2
7
8 result = compare(a, b)
9
10 puts "O resultado da comparação é '#{result}'"
```

O retorno de um método ruby é sempre o resultado de sua última instrução.

## Gems

### O que são?

Gem é um pacote que oferece funcionalidades a fim de resolver uma necessidade específica de um programa Ruby. Pense como o conceito de **biblioteca** em outras linguagens de programação.

**Trabalhando com gems** MARCAR ESTA UNIDADE  
COMO CONCLUÍDA

## Como instalar uma gem

1- Para instalar uma **gem** execute em seu terminal

Default

```
1 | gem install os
```

Você instalou a gem **os**. Ela é uma biblioteca para você extrair informações sobre o PC que você está usando (exp: Sistema operacional, quantidade de cores e etc).

## Novas Funcionalidades

1- Crie um arquivo chamado **os.rb** e adicione o código

Default

```
1 | require 'os'
2 |
3 | def my_os
4 |   if OS.windows?
5 |
6 |     "Windows"
7 |
8 |   elsif OS.linux?
9 |
10 |    "Linux"
11 |
12 |   elsif OS.mac?
13 |
14 |    "Osx"
15 |
16 |   else
17 |
18 |    "Não consegui identificar"
19 |   end
20 |
21 | end
22 |
23 | puts "Meu PC possui #{OS.cpu_count} cores, é #{OS.bits} bits e o sistema operacional é
```

No começo do arquivo, o require 'gem\_name' carrega os arquivos da gem. Isso possibilita a declaração de códigos com funcionalidades da biblioteca.

obs: Para saber como utilizar uma gem consulte sua documentação.

## Desinstalar uma gem

1- Para desinstalar uma gem utilize o comando

Default

```
1 | gem uninstall gem_name
```

Ex:

MARCAR ESTA UNIDADE  
COMO CONCLUÍDA

Default

```
1 | gem uninstall os
```



## Informação

1- Liste todas as gems instaladas na máquina rodando

Default

```
1 | gem list
```

## Bundler

Para ter controle sobre as dependências de um projeto contamos com uma ferramenta que procura e instala gems chamada Bundler.

1- Crie um projeto chamado **first\_project**

Default

```
1 | mkdir first_project
2 | cd first_project
```

2- O bundler também é uma gem. Para instalá-lo rode

Default

```
1 | gem install bundler
```

3- Salve a lista de gems do projeto em um arquivo chamado **Gemfile**

Default

```
1 | source 'https://rubygems.org'
2 |
3 | gem 'os'
```

Na primeira linha é definido onde o bundle deve procurar pelas gems.

Depois é listado as dependências do projeto.

4- Instale estas gems com o comando.

Default

```
1 | bundle install
```

Assim, qualquer pessoa envolvida no projeto pode facilmente instalar suas dependências.

MARCAR ESTA UNIDADE  
COMO CONCLUÍDA

# Missões especiais

## Missão 1

Crie um programa que possua um método que resolva a potência dado um número base e seu expoente. Estes dois valores devem ser informados pelo usuário.

## Missão 2

Siga a documentação da gem `cpf_cnpj` para criar um programa que recebe como entrada um número de cpf e em um método verifique se este número é válido.

Link da documentação:

[https://github.com/fnando/cpf\\_cnpj](https://github.com/fnando/cpf_cnpj)

## Aula passada

## Código da missão 1

Default

```
1 array = []
2
3 i = 1
4
5 1..3.times do
6   print "Digite o #{i}º número: "
7   array.push gets.chomp.to_i
8
9   i += 1
10 end
11
12 array.each do |a|
13   result = a ** 2
14   puts "O resultado do número #{a} elevado a segunda potência é #{result}"
15 end
```

## Código da missão 2

Default

```
1 hash = {}
2
3 3.times do
4   print 'Informe uma chave: '
5   key = gets.chomp
6
7   print 'Informe seu valor: '
8   value = gets.chomp
9
10  hash[key] = value
```

```
11 end
12
13 hash.each do |k, v|
14   puts "Uma das chaves é #{k} e o seu valor é #{v}"
15 end
```

## Código da missão 3

Default

```
1 numbers = {a: 10, b: 30, c: 20, d: 25, e: 15}
2
3 maior_numero = 0
4 result = []
5
6 numbers.each do |key, value|
7   if value > maior_numero
8     maior_numero = value
9     result = [key, value]
10  end
11 end
12
13 puts "O maior número é o da chave #{result[0]} com o valor #{result[1]}"
```

## DISCUSSION



André Costa 2 MESES AGO

### DESAFIO 1

```
def potenciacao(number, exp)
  number ** exp
end

puts "Entre com um número: "
number = gets.chomp.to_i
puts "Entre com o expoente: "
exp = gets.chomp.to_i

puts potenciacao(number, exp)
-----
```

### DESAFIO 2

```
require "cpf_cnpj"

number = 99999999999
number2 = 29048281091

#puts CPF.generate
puts CPF.valid?(number)
puts CPE.valid?(number2)
```

MARCAR ESTA UNIDADE  
COMO CONCLUÍDA

REPLY



brunoraujo 4 MESES AGO



OneBitCode

require "cpf\_cnpj"

resultFinal = ""

loop do

puts resultFinal

puts "Resolução das missões dada em aula"

puts "Selecione um das seguintes opções"

puts "1 – Missão 1: Potência dado um número base e seu expoente"

puts "2 – Missão 2: Validar CPF informado"

puts "0 – Sair"

print "Opção: "

option = gets.chomp.to\_i

case option

when 1

def potencia(base, expoente)

base \*\* expoente

end

print "Digite um número base: "

base = gets.chomp.to\_i

print "Digite o expoente: "

expoente = gets.chomp.to\_i

result = potencia(base, expoente)

resultFinal = "O número base #{base} elevado ao número #{expoente} é  
#{result}"

when 2

def checkcpf(numeroCPF)

if CPF.valid?(numeroCPF)

return "O CPF é válido"

else

return "O CPF é inválido"

end

end

print "Digite um CPF: "

cpf = gets.chomp.to\_i

result = checkcpf(cpf)

resultFinal = result

when 0

MARCAR ESTA UNIDADE  
COMO CONCLUÍDA

Feedba





```
break if option == 0
else
result = "Opção não identificada"
end

#Comando que limpa o console
system "clear"
end
```

REPLY



Nicoli Félix 4 MESES AGO

```
def compare(a,b)
a ** b
end

print "Digite o primeiro número inteiro: "
a = gets.chomp.to_i

print "Digite o segundo número inteiro: "
b = gets.chomp.to_i

result = compare(a,b)

puts "O resultado da exponenciação é '#{result}' "
*****

require "cpf_cnpj"
print 'Digite seu CPF: '
number = gets.chomp

if CPF.valid?(number)
puts 'CPF válido'
else
puts 'CPF inválido'
end
```

REPLY



Fernando Galvão 9 MESES AGO

# Missões especiais

#Missão 1

```
# Crie um programa que possua um método que resolva a potência
# dado um número base e seu expoente.
# Estes dois valores devem ser informados pelo usuário.
```

```
def potencia(base, expoente)
  base ** expoente
end

print "\n\tCálculo de Potenciação\n"

print "\tDigite a base: "
base = gets.chomp.to_f

print "\tDigite o expoente: "
expoente = gets.chomp.to_f

resultado = potencia(base, expoente)
puts "\tPara a base de #{base} e o expoente de #{expoente} o seu
resultado é #{resultado}"
print "\t-----\n"

# Missão 2
# Siga a documentação da gem cpf_cnpj para criar um programa
# que recebe como entrada um número de cpf e em um método
# verifique se este número é válido.

# Link da documentação:
# https://github.com/fnando/cpf_cnpj

# gem install cpf_cnpj

# Foi instalado o gem "cpf_cnpj" dentro de uma arquivo/pasta.
# Com intuito de ter um ambiente isolado para melhor gerenciamento
# da gem.

require 'cpf_cnpj'

def cpfOK(num_cpf)
  if CPF.valid?(num_cpf)
    return 'CPF válido!'
  else
    return 'CPF inválido!'
  end
end

print "\t\tValidação do CPF\n"
print "\tDigite o seu cpf: "
num_cpf = gets.chomp.to_i

resultado = cpfOK(num_cpf)
print "\t#{resultado}\n"
print "\t-----\n"
```



JOAO.MOREIRA 11 MESES AGO

```
def values
  print 'Digite a base: '
  base = gets.chomp.to_i

  print 'Digite o expoente: '
  expoente = gets.chomp.to_i

  [base, expoente]
end

def calc_potencia(base, expoente)
  base**expoente
end

puts "O resultado da operacao é: #{calc_potencia(*values)}"
```

REPLY



Eduardo 11 MESES AGO

```
require "cpf_cnpj"

print "Digite seu CPF: "
number = gets.chomp

puts CPF.valid?(number) ? 'CPF válido!' : 'Erro: CPF inválido!'
```

REPLY



Eduardo 11 MESES AGO

```
def to_calculate(base, exponent)
  pot = 1;
  exponent.times do |i|
    pot *= base
  end
  pot
end

print "Digite a base para o cálculo: "
bas = gets.chomp.to_i

print "Digite a potência para o cálculo: "
exp = gets.chomp.to_i

puts "(#{bas}) elevado a #{exp} = #{to_calculate(bas, exp)}"
```

```
puts "#{bas} elevado a #{exp} = #{to_calculate(bas, exp)}"
```

REPLY

*Kalleby Santos* 12 MESES AGO**Missão 1:**

```
puts '—POTENCIA COM METODOS—'

def pow(base, exponent)
  return base ** exponent
end

def read_values
  print 'Enter with base: '
  base = gets.chomp.to_i

  print 'Enter with exponent: '
  exponent = gets.chomp.to_i

  return base, exponent
end

puts "Result is: #{pow(*read_values)}"
```

REPLY

*Daniel Lucas dos Reis Silva* 1 ANO AGO**require "cpf\_cnpj"**

```
class Cadastro
  attr_accessor :cpf
  def initialize(cpf=nil)
    @cpf=cpf
  end
  def valid_cpf(cpf)
    if CPF.valid?(cpf)
      puts "cpf valido"
    else
      puts "Cpf inválido"
    end
  end
end

puts "Digite o cpf/cnpj: \n"
cpf=gets.chomp.to_s
daniel=Cadastro.new(cpf)
daniel.valid_cpf(daniel.cpf)
```

MARCAR ESTA UNIDADE  
COMO CONCLUÍDA





*Ricardo* 1 ANO AGO

```
source 'http://rubygems.org'
gem 'check_cpf_cnpj', '~> 0.2.1'

require "check_cpf_cnpj"
def verifica_cpf cpf
  if CheckCpfCnpj.valid_cpf?(cpf)
    puts "O CPF #{CheckCpfCnpj.formatted_cpf(cpf)} é válido"
  else
    puts "O CPF #{CheckCpfCnpj.formatted_cpf(cpf)} é inválido"
  end
end

print "Digite o seu CPF (somente números): "
cpf=gets.chomp
puts verifica_cpf cpf
```

REPLY

*Jônatas G. Reis* 1 ANO AGO

Estou com o seguinte erro ao rodar o comando "gem install bundler" -> WARNING: You don't have /home/jgreis/.gem/ruby/2.7.0/bin in your PATH, gem executables will not run. Como arrumar?

REPLY

*Jônatas G. Reis* 1 ANO AGO

Resolvido! Problema na instalação

REPLY



*Joseane* 1 ANO AGO

Tive o mesmo problema e não consegui resolver mesmo pesquisando no google! 😞

REPLY



*Thiago* 1 ANO AGO

1

MARCAR ESTA UNIDADE  
COMO CONCLUÍDA

~~def potencia(base, expoente)~~

ack

```
def potencia(base, expoente)
  return result = base ** expoente
end

puts 'Digite o número base: '
base = gets.chomp.to_i
puts 'Digite o expoente: '
expoente = gets.chomp.to_i

puts potencia(base, expoente)

-----

2
require 'cpf_cnpj'

def validaCPF(cpf)
  return CPF.valid?(cpf)
end

puts 'Digite o CPF: '
cpf = gets.chomp

if validaCPF(cpf)
  puts "O CPF #{cpf} é válido."
else
  puts "O CPF #{cpf} é inválido."
end
```

REPLY



*Gustavo* 2 ANOS AGO

eu estou fazendo os códigos da aula também já olhei a resolução na outra aula e o cpf digitado sempre da como invalido independente do numero

REPLY



*Juliana* 2 ANOS AGO

```
require "cpf_cnpj"

def cpf_valido(cpf)
  if CPF.valid?(cpf)
    puts "CPF válido"
  else
    puts "CPF inválido"
  end
end
```

MARCAR ESTA UNIDADE  
COMO CONCLUÍDA

end

REPLY



Juliana 2 ANOS AGO

```
def expoente(a,b)
a**b
end

puts 'Digite um numeros'
a = gets.chomp.to_i

puts 'Digite o expoente'
b = gets.chomp.to_i

resultado = expoente(a,b)

puts "Expoente é : #{resultado}"
```

REPLY



Juliana 2 ANOS AGO

```
def expoente(a,b)
a**b
end

puts 'Digite um numeros'
a = gets.chomp.to_i

puts 'Digite o expoente'
b = gets.chomp.to_i

resultado = expoente(a,b)

puts "Expoente é : #{resultado}"

require "cpf_cnpj"

def cpf_valido(cpf)
if CPF.valid?(cpf)
puts "CPF válido"
else
puts "CPF inválido"
end
end
end
```

```
puts 'Digite o CPF'
cpf = gets.chomp
```

MARCAR ESTA UNIDADE  
COMO CONCLUÍDA



cpf\_valido(cpf)

REPLY

*Gustavo Coelho* 2 ANOS AGO

#MISSÃO 1

```
def pot(num1, num2)
  return num1 ** num2
end

puts "digite um número: "
a = gets.chomp.to_i
puts "digite um expoente: "
b = gets.chomp.to_i

puts "#{a} elevado a potência de #{b} é #{pot(a, b)}"

#MISSÃO 2

require 'cpf_cnpj'
def verifier(cpf)
  if CPF.valid?(cpf)
    'válido'
  else
    'inválido'
  end
end

puts "digite um CPF"
cpf = gets.chomp
puts "seu CPF é #{verifier(cpf)}"
```

REPLY

*Ideilson* 2 ANOS AGO

===Missão 1===

```
def pot base, exp
  puts base ** exp
end

puts "Digite um número e um expoente: "
base = gets.chomp.to_i
exp = gets.chomp.to_i
pot base, exp

===Missão 2===
```

MARCAR ESTA UNIDADE  
COMO CONCLUÍDA





```
require "cpf_cnpj"

def validate_cpf number_cpf
  cpf = CPF.new(number_cpf, strict: true)
  if cpf.valid?
    puts "O CPF: #{cpf.formatted} é válido"
  else
    puts "O CPF: #{cpf.formatted} é inválido"
  end
end

print "Digite seu CPF: "
number_cpf = gets.chomp
validate_cpf number_cpf
```

REPLY

*Frankyston Lins* 2 ANOS AGO

<https://gist.github.com/frankyston/fe227422210b84961fdaf67083d9546>

REPLY



*Flávio S Ferreira* 2 ANOS AGO

### MISSÃO 1

```
def potencia(base, exp)
  base ** exp
end

print 'Digite base da potência: '
base = gets.chomp.to_i
print 'Digite expoente da potência: '
exp = gets.chomp.to_i

result = potencia(base, exp)

puts "#{base} elevado a #{exp} é igual a #{result}"
```

### MISSÃO 2

```
require 'cpf_cnpj'
```

```
def verifica_cpf(cpf)
```

```
  # desenvolver metodo
```

```
  if CPF.valid?(cpf)
```

```
    'VÁLIDO'
```

MARCAR ESTA UNIDADE  
COMO CONCLUÍDA

```
else
  'INVÁLIDO'
end
end

print 'Digite o CPF: '
cpf = gets.chomp

puts "CPF #{verifica_cpf(cpf)}"
```

REPLY

*VINICIUS LEITE LIMA* 2 ANOS AGO

Missao 1

```
def potencia(base, expoente)
  puts "Resultado: #{base ** expoente}"
end
```

```
puts 'Digite o numero da base'
base = gets.chomp.to_f
```

```
puts 'Digite o numero do expoente'
expoente = gets.chomp.to_f
```

```
potencia(base,expoente)
```

Missao 2

```
require "cpf_cnpj"
```

```
def verifica_cpf(cpf)
```

```
  if CPF.valid?(cpf)
```

```
    "valido"
```

```
  else
```

```
    "invalido"
```

```
  end
```

```
end
```

```
puts 'digite o numero do cpf'
```

```
cpf = gets.chomp
```

```
puts "O cpf é #{verifica_cpf(cpf)}"
```

REPLY

*Rafael* 2 ANOS AGO

Valeu!

MARCAR ESTA UNIDADE  
COMO CONCLUÍDA



Rafael 2 ANOS AGO

Leonardo conseguiu resolver problemas relacionados as extensões e instalação de gems,laeu!

REPLY



Rafael 2 ANOS AGO

opss e versão do ubuntu é 19.04 digitei errado

REPLY



Rafael 2 ANOS AGO

olá leonardo,venho a tempos vendo o trabalho de vcs e tá show,parei nesta aula não consigo instalar as gem no meu console,estou no ubuntu 19.04.0 e usando o visual code,ele me uma mensagem dizendo que não tenho permissão para acessar esta documentação,sabe me dizer o que faço?

REPLY



Leonardo Scorza 2 ANOS AGO

Olá Rafa, beleza?

Fico feliz que esteja fazendo nosso curso 😊

Qual erro aparece para você?

Abraço

REPLY



Rafael 2 ANOS AGO

to curtindo o ruby,tenho uma breve experiencia com c e java mas achei o ruby mais amigavel,agora to nessa não consigo instalar nenhuma gem

REPLY



Rafael 2 ANOS AGO

vou tentar explicar ele fala que não tenho

acesso a documentação deste diretorio mas tenho uma lista de gem que aparecem quando chamo o comando para listarem o qual não lembro agora e não consigo instalar nenhuma

MARCAR ESTA UNIDADE  
COMO CONCLUÍDA

tembre agora e não consegue instalar nenhuma  
como o minitest sempre da o mesmo erro

REPL 

Leonardo Scorza 2 ANOS AGO

Entendi 😊

Você está em qual sistema operacional?

Você está usando o rvm ou rbenv?

REPLY



Rafael 2 ANOS AGO

estou usando o ubuntu18.04.3 LTS,mas eu consegui instalar as gem,dei uma pesquisada e achei pois o erro era pq o diretorio aonde as gems deveriam ser instaladas nao tinha permissão para escrita mesmo usando chmod 777 ele negava,mas c eu usa-se sudo,por exemplo,sudo gem install nome\_da\_gem ai funcionava,mas agora to com outro problema,não consigo instalar as extensões no vscode,por isso não consigo debbugar corretamente.Estou usando rvm.

REPLY



Rafael 2 ANOS AGO

desculpe.estou usando rbenv

REPLY

MARCAR ESTA UNIDADE  
COMO CONCLUÍDA



Edward 3 ANOS AGO

Missão 1

Crie um programa que possua um método que resolva a potência dado um número base e seu expoente. Estes dois valores devem ser informados pelo usuário.

```
def potenciacao(numero, expoente)
  unless numero == 0 && expoente == 0
    puts "O resultado de #{numero} elevado a #{expoente} foi: #{numero**expoente}"
  else
    puts 'Desculpa, mas é impossível fazer a potenciação sem números'
  end
end

print 'Digite o número a ser elevado: '
numero = gets.chomp.to_i
print 'Agora digite o número do expoente: '
expoente = gets.chomp.to_i
potenciacao(numero, expoente)
```

## Missão 2

Siga a documentação da gem `cpf_cnpj` para criar um programa que recebe como entrada um número de cpf e em um método verifique se este número é válido.

```
require 'cpf_cnpj'

def valida_cpf_cnpj(numero)
  tipo_documento = ""
  numero_formatado = 0
  unless numero == 0
    #CPF.valid?(numero)
    if numero.to_s.length == 11
      tipo_documento = 'CPF'
      numero_cpf = CPF.new(numero)
      numero_formatado = numero_cpf.formatted
      if numero_cpf.valid?
        documento_status = 'Válido'
      else
        documento_status = 'Inválido'
      end
    elsif numero.to_s.length == 14
      tipo_documento = 'CNPJ'
      numero_cnpj = CNPJ.new(numero)
```



```
numero_formatado = numero_cnpj.formatted

if numero_cnpj.valid?
  documento_status = 'Válido'
else
  documento_status = 'Inválido'
end
else
  puts 'O número digitado é maior que a quantidade de número para um
  CPF ou CNPJ'
  exit
end

puts "O #{tipo_documento} digitado foi #{numero_formatado} e é um
  documento #{documento_status}"
else
  puts 'É necessário digitar um número'
end
end

print 'Digite um número de CPF ou CNPJ: '
numero_documento = gets.chomp.to_i
valida_cpf_cnpj(numero_documento)
```

REPLY

*ranx* 3 ANOS AGO

## # Missão 1

```
# Crie um programa que possua um método que resolva
# a potência dado um número base e seu expoente.
# Estes dois valores devem ser informados pelo usuário.
```

```
def calc_exp base, expoente
  base.to_f ** expoente.to_f
end
```

```
printf "Informe a base: "
base = gets.strip

printf "Informe o expoente: "
exp = gets.strip
```

```
puts "#{base} elevado a #{exp} é igual a: %0.2f" % calc_exp(base, exp)
```

## # Missão 2

```
# Siga a documentação da gem cpf_cnpj para
```

MARCAR ESTA UNIDADE  
COMO CONCLUÍDA

```
# criar um programa que recebe como entrada
# um número de cpf e em um método verifique
# se este número é válido.

require "cpf_cnpj"

def validate_cpf number
  CPF.new(number).valid? ? "válido" : "incorreto"
end

printf "Informe o número de CPF: "

puts "Número de CPF #{validate_cpf(gets.strip)}"
```

REPLY



*Leandro Alves* 3 ANOS AGO

```
def calcula_pow(number, number2)
  puts number ** number2
end

puts "Informe o primeiro número"
first_number = gets.chomp.to_i

puts "Informe o expoente"
second_number = gets.chomp.to_i

calcula_pow(first_number,second_number)

-----

require "cpf_cnpj"

def valida_cpf(cpf)
  if CPF.valid?(cpf)
    puts "CPF válido"
  else
    puts "CPF inválido"
  end
end

puts "Informe o número de CPF:"
valida_cpf(gets.chomp)
```

REPLY

*Karlysson Alves* 3 ANOS AGO

MARCAR ESTA UNIDADE  
 COMO CONCLUÍDA  
 ----- missão parte 1

```
puts 'Digite um numero Base:'
base = gets.chomp.to_i

puts 'Digite um numero Expoente:'
expoente = gets.chomp.to_i

def calculo(b, e)
  result = b**e
  puts "O resultado do cálculo de Base [{b}] elevado ao Expoente [{e}]
  é: [{result}]"
end

calculo(base, expoente)

puts '----- missão parte 2
-----'

require 'cpf_cnpj'

puts 'Digite o numero do seu CPF: '
cpf = gets.chomp.to_s

def valida(numero)
  CPF.valid?(numero)
end

puts "O CPF é: #{valida(cpf)}"
```

REPLY



Priscila Ferreira Bezerra 3 ANOS AGO

# Missões especiais

# Missão 1

# Crie um programa que possua um método que resolva a potência dado um número base e seu expoente. Estes dois valores devem ser informados pelo usuário.

```
def pot (n1, n2)
  result = n1 ** n2
  puts "The result is: #{result}"
end
print "Insert first number: "
n1 = gets.chomp.to_i
print "Insert second number: "
n2 = gets.chomp.to_i
pot(n1, n2)
```

MARCAR ESTA UNIDADE  
COMO CONCLUÍDA



```
puts ""
puts "-----"
puts ""
```

## # Missão 2

# Siga a documentação da gem `cpf_cnpj` para criar um programa que recebe como entrada um número de cpf e em um método verifique se este número é válido.

# Link da documentação:

# [https://github.com/fnando/cpf\\_cnpj](https://github.com/fnando/cpf_cnpj)

```
def valid_number(cpf_n)
  if CPF.valid?(cpf_n)
    puts "Valido"
  else
    puts "invalido"
  end
end
```

```
print "insira o seu cpf"
cpf_n = gets.chomp
valid_number(cpf_n)
```

REPLY



Ricardo Araujo 3 ANOS AGO

# Siga a documentação da gem `cpf_cnpj` para criar um programa

# que recebe como entrada um número de cpf e em um  
# método verifique se este número é válido.

```
require "cpf_cnpj"

puts "Digite um CPF: "
cpf = CPF.new(gets.chomp)

def verificar_cpf(cpf)
  if cpf.valid?
    puts "CPF valido"
    puts "#{cpf.formatted}"
  else
    puts "CPF invalido"
  end
end
```

MARCAR ESTA UNIDADE  
COMO CONCLUÍDA

verificar\_cpf(cpf)

REPLY



Matheus Ribeiro da Fonseca 3 ANOS AGO

No exemplo de gens para descobrir o sistema operacional não rodou e deu essa mensagem de erro: O sistema\_operacional.rb:7:in my\_os': undefined method linux' for OS:Class (NoMethodError)  
Did you mean? linux?  
from Sistema\_operacional.rb:19:in ``  
obs: eu instalei no terminal gem install os

REPLY



Denilson Silva 3 ANOS AGO

```
require "cpf_cnpj"

numero = "111.111.111-11"

def validar(num)
  tipo = CPF.valid?(num, strict: true)
  if tipo == true
    "Válido"
  elsif tipo == false
    "Falso"
  else
    "Indefinido"
  end
end

print validar(numero)
print "\n"

def potencia(n1, n2)
  n1 ** n2
end

print "Digite sua Base: "
n1 = gets.chomp.to_i
print "Digite sua Potencia: "
n2 = gets.chomp.to_i

print "O resultado: #{potencia(n1, n2)} sua Base: #{n1}, sua Potencia #{n2}"
```

MARCAR ESTA UNIDADE

COMO CONCLUI DA

```
print "\n"
```

REPLY

*Marcio* 3 ANOS AGO**desafio 1**

```
-----  
def potencia(n1,n2)  
  return n1 ** n2  
end  
  
puts "digite sua base"  
base = gets.chomp.to_i  
puts "digite o expoente"  
expoente = gets.chomp.to_i  
  
result = potencia(base,expoente)  
  
puts "#{base} elevado a #{expoente} = #{result}"  
  
desafio 2  
-----  
require 'cpf_cnpj'  
  
def valid(cpfNumber)  
  if CPF.valid?(cpfNumber)  
    " é Valido"  
  else  
    "não é Validao"  
  end  
end  
  
puts 'digite seu cpf'  
  
cpf = gets.chomp  
  
puts "o cpf digitado #{valid(cpf)}"
```

REPLY

*Gabriel* 3 ANOS AGO**def Potencia n1,n2****n1\*\*n2****end****numeros = []****MARCAR ESTA UNIDADE  
COMO CONCLUÍDA****puts "Digite a base da potencia:"**

```

numeros.push(gets.chomp)
puts "Digite a potencia:"
numeros.push(gets.chomp)

puts "Resultado da potencia foi #{Potencia numeros[0].to_i,
numeros[1].to_i}"

```

---

```

require "cpf_cnpj"

puts '***** Desafio 2 *****'
puts 'Digite um CPF para verificação'
cpf = CPF.new(gets.chomp)

if(cpf.valid?)
  puts "CPF #{cpf.formatted} valido!"
else
  puts 'CPF inválido!'
end

```

REPLY



*Gabriel Henrique* 3 ANOS AGO

```
require 'cpf_cnpj'
```

```

def validar numero
  if CPF.valid?(numero, strict:true)
    puts "O CPF é válido!"
  else
    puts "O CPF não é valido"
  end
end

puts "Informe o número do CPF: "
numero = gets.chomp
validar numero

```

REPLY

*Alan Maxwell* 3 ANOS AGO

A única maneira que rodou direito no meu foi essa:

```
require 'cpf_cnpj'
```

```

def registra_cpf(number)
  CPF.new(number)
end

```

MARCAR ESTA UNIDADE  
COMO CONCLUÍDA

```

def valida_cpf(number)
  CPF.valid?(number)
end

puts 'Registre seu CPF:'
print 'Digite o CPF: '
cpf = gets.chomp

registra_cpf(cpf)

puts 'Validação..'

if valida_cpf(cpf)
  puts "Seu CPF é valido"
else
  puts "CPF inválido"
end

```

REPLY



*Raphael Filipe* 3 ANOS AGO

require "cpf\_cnpj"

```

def registrar
  puts "Insira seu CPF"
  cpfr = gets.chomp
  CPF.new (cpfr)
end

def comparar
  puts "Digite Seu CPF"
  cpfg = gets.chomp
  if CPF.valid?(cpfg, strict: true)
    puts 'Seu CPF está correto!'
  else
    puts 'Numero de CPF invalido'
  end
end

puts "Aperte 1 P/ CADASTRAR"
puts "Aperte 2 P/ VALIDAR"

```

```

opcao = gets.chomp.to_i
if opcao == 1
  registrar

```

MARCAR ESTA UNIDADE  
COMO CONCLUÍDA

registrar

```
elsif opcao == 2
  comparar
else
  puts "Opção Invalida"
end
```

REPLY



Raphael Filipe 3 ANOS AGO

exemplo para se basearem

```
require "cpf_cnpj"

def registrar
  puts "Insira seu CPF"
  cpfr = gets.chomp
  CPF.new (cpfr)
end

def comparar
  puts "Digite Seu CPF"
  cpfg = gets.chomp
  if CPF.valid?(cpfg, strict: true)
    puts 'Seu CPF está correto!'
  else
    puts 'Numero de CPF invalido'
  end
end

puts "Aperte 1 P/ CADASTRAR"
puts "Aperte 2 P/ VALIDAR"

opcao = gets.chomp.to_i

if opcao == 1
  registrar
else
  comparar
else
  puts "Opção Invalida"
end
```

MARCAR ESTA UNIDADE  
COMO CONCLUÍDA

REPLY



Yuri Tavares 3 ANOS AGO

### Missão 1

```
def power(base_number, exponent)
  result = base_number**exponent
  puts "O resultado é #{result}"
end

puts "Favor digite um numero!"
base_number = gets.chomp.to_i
puts 'Agora qual numero que deseja eleva-lo?'
exponent = gets.chomp.to_i

power base_number, exponent
```

-----

### Missão 2

```
require 'cpf_cnpj'

puts 'Favor insira o numero do seu CPF!'
cpf = CPF.new(gets.chomp.to_i)

if cpf.valid?
  puts 'Seu CPF está correto!'
  puts "#{cpf.formatted}"
else
  puts 'Numero de CPF invalido'
end
```

REPLY



Moisés Tedeschi de Melo 3 ANOS AGO

### 1º Exercício – Code:

```
def potencia(x, y)
  return x ** y
end

print 'Entre com a base: '
base = gets.chomp.to_i
print 'Entre com a potência: '

pot = gets.chomp.to_i

resultado = potencia(base, pot)
puts "Resultado do cálculo: #{resultado}"
```

MARCAR ESTA UNIDADE  
COMO CONCLUÍDA

## 2º Exercício – Code:

#Autor Moisés Tedeschi

#2018

#Biblioteca usada: gem 'cpf\_cnpj', '~&gt; 0.2.1'

require "cpf\_cnpj"

def valida\_cpf(pf)

if CPF.valid?(pf)

puts "O CPF: #{pf} é Válido!"

else

puts "O CPF está inválido!"

end

end

def valida\_cnpj(pj)

if CNPJ.valid?(pj)

puts "O CNPJ: #{pj} é Válido!"

else

puts "O CNPJ está inválido!"

end

end

puts "Qual o tipo de pessoa?"

puts '1 – Pessoa Física'

puts '2 – Pessoa Juridica'

op = gets.chomp.to\_i

if op == 1

print 'Informe o CPF: '

pf = gets.chomp

validacao\_pf = valida\_cpf(pf)

puts validacao\_pf

elsif op == 2

print 'Informe o CNPJ: '

pj = gets.chomp

validacao\_pj = valida\_cnpj(pj)

puts validacao\_pj

else

print 'Opção inválida!'

end

MARCAR ESTA UNIDADE  
COMO CONCLUÍDA



Jakeline 3 ANOS AGO

O meu código só mostra cpf inválido.

```
require "cpf_cnpj"

def verification(cpf)
  if CPF.valid?(cpf)
    puts "Seu CPF é válido."
  else
    puts "CPF inválido!"
  end
end

print "Digite o número do seu CPF: "
cpf = gets.chomp.to_i

result = verification(cpf)

puts result
```

REPLY



Gabriel Henrique 3 ANOS AGO

tire o ".to\_i" do gets

REPLY



Arthur Guedes 3 ANOS AGO

Pode ser que você esteja convertendo um número float por um inteiro

REPLY



Aroldo 3 ANOS AGO

require "cpf\_cnpj"

```
def my_cpf(cpf_number)
  if CPF.valid?(cpf_number, strict: true)
    return "OK"
  else
    return "false"
  end
end
```

```
puts "Digite o seu cpf"
```

MARCAR ESTA UNIDADE  
COMO CONCLUÍDA



```
cpf_number = gets.chomp.to_i
result = my_cpf(cpf_number)
puts result
```

REPLY

*Aroldo* 3 ANOS AGO**Missao 1**

```
def potencia(a,b)
a ** b
end

puts "Digite o primeiro numero"
a = gets.chomp.to_i
puts "Digite o segundo numero"
b = gets.chomp.to_i

result = potencia(a, b)

puts " a potencida eh #{result}"
```

REPLY

*Aroldo* 3 ANOS AGO**Leonardo, podemos postar aqui os códigos das missões?**

REPLY

*Leonardo Scorza* 3 ANOS AGO**Pode sim Aroldo 😊**

REPLY

*Bruno Lucas da Silva Fernandes* 3 ANOS AGO**Esse curso vai ter certificação?**

REPLY

ASK QUESTION

MARCAR ESTA UNIDADE  
COMO CONCLUÍDA