

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems



Programação Orientada a
Objetos I



Estruturas de Controle

Feedback

Curso de Ruby - Estrutu...



Conteúdo

Será abordado:

- Estruturas de Controle
- Condicional
 - If/Else/Elsif
 - Unless
 - Case
- Iteração
 - For
 - While
 - Times
 - Do/While (Loop)
- Iteração + Condicional

Estruturas de Controle

Tratam-se de códigos que escrevemos em nossos programas para analisar dados e decidir qual caminho seguir.

Divide-se em dois tipos,
Condicional e Iteração. Nesta aula você aprenderá como utilizá-los.

MARCAR ESTA UN
COMO CONCL

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

- O Início
- Conceitos Básicos
- Estruturas de Controle
- Collections
- Métodos e Gems
- Programação Orientada a Objetos I

Condisional

Tipo de estrutura de controle que executa um trecho de código dependendo do resultado de uma condição.

Abaixo, veja as Instruções Condicionais If, Else, Elsif, Unless e Case.

If

Expressão que verifica se uma condição é verdadeira(true), e a partir deste resultado determina se as instruções dentro de seu corpo serão ou não executadas.

1. Para entender como utiliza-lo crie um arquivo chamado if.rb e adicione o seguinte código

```
Default
1 day = 'Sunday'
2
3 if day == 'Sunday'
4   lunch = 'special'
5 end
6
7 puts "Lunch is #{lunch} today"
```

Leia a condição da seguinte forma:
Se o dia é Domingo, então o almoço é especial

2. Execute o programa (ruby if.rb) e veja que a condição é verdadeira, pois foi exibido na tela

```
Default
1 Today the lunch is special.
```

Else

MARCAR ESTA UN
COMO CONCL

Informa o que fazer quando a

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

verificação de uma condição if for falsa.

1. Para utilizar o else, substitua o código do arquivo if.rb por

```
Default
1 day = 'Saturday'
2
3
4 if day == 'Sunday'
5   lunch = 'special'
6 else
7   lunch = 'normal'
8 end
9
10 puts "Lunch is #{lunch} today"
```

Leia a condição da seguinte forma:

Se o dia é Domingo, então o almoço é especial
Senão, o almoço é normal.

2. Execute o programa e veja que nossa condição é falsa. O bloco de código dentro do Else então é executado e por fim exibido na tela.

```
Default
1 Today the lunch is normal.
```

Elsif

Utilizado quando há a necessidade de verificar mais de uma condição em um if.

1. Substitua o código do arquivo if.rb por

```
Default
1 day = 'Holiday'
2
3 if day == 'Sunday'
4   lunch = 'special'
5 elsif day == 'Holiday'
6   lunch = 'later'
7 else
8   lunch = 'normal'
9 end
```

MARCAR ESTA UN
COMO CONCL

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início

Conceitos Básicos

Estruturas de Controle

Collections

Métodos e Gems

Programação Orientada a
Objetos I

```
10 puts "Lunch is #{lunch} today"
11
```

Leia a condição da seguinte forma:

Se o dia é Domingo, então o almoço é especial.

Senão e se o dia é feriado, então o almoço é tarde.

Senão, o almoço é normal.

Feedback

Apenas nossa segunda condição é verdadeira.

2. Execute o programa e veja que o resultado será

```
1 Today the lunch is normal.
```

Default

Unless

Enquanto o if é executado quando sua condição é verdadeira, o unless ocorre de forma contrária. É executado apenas quando a condição é falsa.

1. Crie um arquivo chamado `unless.rb` e adicione o seguinte código

```
1 product_status = 'closed'
2
3 unless product_status == 'open'
4   check_change = 'can'
5 else
6   check_change = 'can not'
7 end
8
9 puts "You #{check_change} change th
```

Default

Leia da seguinte forma

A menos que o estado do produto seja aberto, a troca é possível.

MARCAR ESTA UN
COMO CONCL

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

- O Início
- Conceitos Básicos
- Estruturas de Controle
- Collections
- Métodos e Gems
- Programação Orientada a Objetos I

Senão, a troca não é possível.

2. O produto não foi aberto.
Execute o programa e veja que o resultado será

Default
1 | You can change the product

Feedback

Case

Instrução muito parecida com o if.
Ele se enquadra muito bem a situações com diversas condições.

1. Em um novo arquivo chamado case.rb adicione o código

Default
1 puts 'Digite o número do mês em que nasceu:'
2 month = gets.chomp.to_i
3
4
5 case month
6 when 1..3
7 puts 'Você nasceu no começo do ano'
8 when 9..12
9 puts 'Você nasceu no final do ano'
10 when 4..6
11 puts 'Você nasceu na primeira metade do ano'
12 when 7..9
13 puts 'Você nasceu na segunda metade do ano'
14 else
15 puts 'Não foi possível identificar seu mês'
16 end

Leia a condição da seguinte forma

Caso o mês informado esteja no intervalo entre 1 e 3, você nasceu no começo do ano
esteja no intervalo entre 9 e 12, você nasceu no final do

ano
esteja no intervalo entre 4 e 6, você nasceu na primeira metade do ano

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início

Conceitos Básicos

Estruturas de Controle

Collections

Métodos e Gems

Programação Orientada a Objetos I

esteja no intervalo entre 7 e 9, você nasceu na segunda metade do ano
Senão, não foi possível identificar o mês



Iteração

Tipo de estrutura de controle que gerencia quantas vezes um trecho de código será executado.
Abaixo, veja as instruções de iteração For, While, Times, Do/While.

For

Usado para percorrer uma coleção de elementos.

1. Crie um programa chamado `for.rb` com o seguinte código

```
Default
1 fruits = ['Maçã', 'Uva', 'Morango']
2
3 for fruit in fruits
4   puts fruit
5 end
```

No exemplo, a instrução `for` percorrerá todos os elementos da lista `fruits`. Em cada iteração, podemos acessar o elemento atual através da variável `fruit`.

2. Execute o programa e veja o nome da fruta cada vez que a repetição é executada.

```
Default
1 ruby for.rb
```

MARCAR ESTA UN
COMO CONCL

3. Agora, substitua o código de

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início

Conceitos Básicos

Estruturas de Controle

Collections

Métodos e Gems

Programação Orientada a Objetos I

for.rb por

```
1 fruits = ['Maçã', 'Uva', 'Morango']
2
3 fruit = "Laranja"
4
5 for fruit in fruits
6   puts fruit
7 end
8
9 puts fruit
```

Default

Feedback

CUIDADO

Ao executar o programa, note que a variável de iteração pode sobrescrever outra que esteja fora da estrutura de repetição.

While

Instrução que repete um bloco de código enquanto sua condição é verdadeira.

1. Crie um programa chamado **while.rb** com o seguinte código

```
1 x = 1
2
3 while x < 10
4   puts x
5   # Adiciona + 1 ao valor de x
6   x += 1
7 end
```

Default

Quando este programa é executado, a instrução **while** é repetida enquanto o valor de **x** for menor que 10.

Times

Executa uma repetição por um especificado número de vezes.

1. Em um novo programa MARCAR ESTA UN chamado **times.rb** adicione o COMO CONCL segue o código

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início

Conceitos Básicos

Estruturas de Controle

Collections

Métodos e Gems

Programação Orientada a Objetos I



```
Default
1 2.times do
2   puts 'Estou aprendendo times!'
3 end
4
5 names = ['João', 'Alfredo', 'Juca']
6 # Igual ao array, o times começa com 0
7 3.times do |index|
8   puts names[index]
9 end
```

Feedback

Execute o programa e perceba que a estrutura times:
Exibe a frase “Estou aprendendo times” 2 vezes
Exibe um índice do array name por 3 vezes

Do/While

Na verdade, no Ruby utilizamos uma estrutura de repetição chamada loop que faz o mesmo que o do/while em outras linguagens de programação. Ele cria um laço de repetição que só é parado quando uma instrução break for verdadeira.

1. Em um arquivo chamado loop.rb adicione o seguinte código.

```
Default
1 count = 1
2 loop do
3   puts count
4   break if count == 10
5   # Incrementa a variável count
6   count += 1
7 end
```

Foi criado uma estrutura de repetição que só será parada quando o valor da variável count for igual a 10. MARCAR ESTA UNA COMO CONCLUIDA

Execute o programa e veia

que ele contará de 1 a 10.

Feedback
:

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início

Conceitos Básicos

Estruturas de Controle

Collections

Métodos e Gems

Programação Orientada a
Objetos I

Iteração + Condicional

1. Veja um exemplo de como
utilizar estas duas estruturas em
conjunto, em `age.rb`, coloque:

```
Default
1 result = ''
2 loop do
3   puts result
4   puts 'Selecione uma das seguintes:'
5   puts '1- Descobrir a idade de uma pessoa'
6   puts '0- Sair'
7   print 'Opção: '
8
9   option = gets.chomp.to_i
10
11  if option == 1
12    print 'Digite o ano de nascimento: '
13    year_of_birth = gets.chomp.to_i
14    print 'Digite o ano atual: '
15    current_year = gets.chomp.to_i
16    age = current_year - year_of_birth
17    result = "Quem nasceu no ano de #{year_of_birth} tem #{age} anos."
18
19  elsif option == 0
20    break if option == 0
21  else
22    result = 'Opção inválida'
23  end
24  # Comando que limpa o console
25  system "clear"
26
27 end
```

Neste programa é criado
uma estrutura de repetição
que permite ao usuário
descobrir a idade de alguma
pessoa.
O programa é executado até
que a opção 0 que significa
sair seja escolhida.

Missões especiais

Utilizando as estruturas de
iteração e condição, crie uma
calculadora que ofereça ao usuário
a opção de Multiplicar.

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início

Conceitos Básicos

Estruturas de Controle

Collections

Métodos e Gems

Programação Orientada a
Objetos I

Dividir, Adicionar ou Subtrair dois números. Não se esqueça de também permitir que o usuário feche o programa.



Aula passada

Código da Missão 2

```
Default
1 print 'Digite seu nome: '
2 name = gets.chomp
3
4 print 'Digite sua idade: '
5 age = gets.chomp
6
7 puts "Seu nome é #{name} e você tem
```

Código da Missão 3

```
Default
1 print "Digite o primeiro número in"
2 # .to_i transforma a string em um i
3 number1 = gets.chomp.to_i
4
5 print "Digite o segundo número int
6
7
8 number2 = gets.chomp.to_i
9
10 puts 'Resultado das seguintes opçõ
11 result = number1 + number2
12 puts "Adição = #{result}"
13
14 result = number1 - number2
15 puts "Subtração = #{result}"
16
17 result = number1 * number2
18 puts "Multiplicação = #{result}"
19
20 result = number1 / number2
21 puts "Divisão = #{result}"
```

DISCUSSION



Willyan Ribeiro
Barbosa Martins

2 SEMANAS AGO

result = "

loop do
 puts result
 puts 'Seleciona uma'

MARCAR ESTA UN
COMO CONCL

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

```

das seguintes opções'
puts '1- Descobrir a
idade de uma pessoa'
puts '0- Sair'
print 'Opção: '

option =
gets.chomp.to_i

if option == 1
print 'Digite o ano de
nascimento: '
year_of_birth =
gets.chomp.to_i
print 'Digite o ano atual:'
current_year =
gets.chomp.to_i
age = current_year -
year_of_birth
result = "Quem nasceu
no ano de #
{year_of_birth}, tem #
{age} anos em #
{current_year}"
elsif option == 0
puts "
puts "Até logo!"
puts "
else
result = 'Opção inválida'
end

break if option == 0

#comando que limpa o
console
system "clear"
end

```

REPLY



Willyan Ribeiro MARCAR ESTA UN
Barbosa Martins COMO CONCL

2 SEMANAS AGO

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

O comando do arquivo `age.rb` indicado neste módulo não está funcionando bem quando escolhemos a opção “0”. Fiz umas modificações dentro do código e ficou melhor:

REPLY



Willyan Ribeiro

Barbosa Martins

2 SEMANAS AGO

Ao digitar minha dúvida aqui, a letra está em branco e o fundo da caixa de texto tbm está em branco. Por isso não consigo entender o que estou escrevendo.. kkk

REPLY



Marcos Vinicius de Oliveira Guilhon Rosa

4 SEMANAS AGO

```
result = "
loop do
  puts result
  puts 'Selecione uma das seguintes opções:'
  puts '1- Soma'
  puts '2- Subtracao'
  puts '3- Multiplicacao'
  puts '4- Divisao'
  puts '0- Sair'
```

print 'Opção: '

option =

MARCAR ESTA UN

COMO CONCL

gets.chomp.to_i

Feedback

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems



Programação Orientada a
Objetos I



```
if option == 1
print 'Digite o primeiro
numero: '
num1 = gets.chomp.to_i
print 'Digite o segundo
número: '
num2 = gets.chomp.to_i
final = num1 + num2
result = "O resultado é:
#{final}!"
```

```
elsif option == 2
print 'Digite o primeiro
numero: '
num1 = gets.chomp.to_i
print 'Digite o segundo
número: '
num2 = gets.chomp.to_i
final = num1 - num2
result = "O resultado é:
#{final}!"
```

```
elsif option == 3
print 'Digite o primeiro
numero: '
num1 = gets.chomp.to_i
print 'Digite o segundo
número: '
num2 = gets.chomp.to_i
final = num1 * num2
result = "O resultado é:
#{final}!"
```

```
elsif option == 4
print 'Digite o primeiro
numero: '
num1 = gets.chomp.to_i
print 'Digite o segundo
número: '
num2 = gets.chomp.to_i
final = num1 / num2
result = "O resultado é:
#{final}!"
```

MARCAR ESTA UN
COMO CONCL

#Final!

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas



REPLY

O Início



André Costa 2 MESES AGO

loop do
puts 'Digite um número:
'

Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos Inum1 = gets.chomp.to_i
puts 'Digite a operação:
'
operacao = gets.chomp
puts 'Digite o segundo
número:'
num2 = gets.chomp.to_i
result = "case operacao
when '+'
result = num1 + num2
when '-'
result = num1 - num2
when '*'
result = num1 * num2
when '/'
result = num1 / num2
else
result = 'Opção invalida'end
if result == "Opção invalida"
MARCAR ESTA UN
COMO CONCL

Feedback

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems



REPLY

Programação Orientada a
Objetos I

Lorena 4 MESES AGO

result = ""

```
loop do
  puts "Calculadora"
  puts "Selecione a
  operação que deseja
  fazer:"
  puts "1- Somar"
  puts "2- Subtrair"
  puts "3- Multiplicar"
  puts "4- Dividir"
  puts "0- Sair"
```

```
print "Opção
selecionada: "
```

```
option =
gets.chomp.to_i
case option
```

MARCAR ESTA UN
COMO CONCL

Feedback


Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Feedback

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems



Programação Orientada a
Objetos I



```
when 0
break
when 1
print 'Digite o primeiro
número: '
n1 = gets.chomp.to_i
print 'Digite o segundo
número: '
n2 = gets.chomp.to_i
puts 'Calculando...'
result = n1 + n2
puts "O resultado da
soma é: #{result}"
when 2
print 'Digite o primeiro
número: '
n1 = gets.chomp.to_i
print 'Digite o segundo
número: '
n2 = gets.chomp.to_i
puts 'Calculando...'
result = n1 - n2
puts "O resultado da
subtração é: #{result}"
when 3
print 'Digite o primeiro
número: '
n1 = gets.chomp.to_i
print 'Digite o segundo
número: '
n2 = gets.chomp.to_i
puts 'Calculando...'
result = n1 * n2
puts "O resultado da
multiplicação é: #
{result}"
when 4
print 'Digite o primeiro
número: '      MARCAR ESTA UN
n1 = gets.chomp.to_i
print 'Digite o segundo
número: '      COMO CONCL
```

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

print "Digite o segundo

número: '

n2 = gets.chomp.to_i
puts 'Calculando...'result = n1 / n2
puts "O resultado da
divisão é: #{result}"else
puts 'Opção inválida'
end
end

REPLY

Johnny andrews silva
barbosa

5 MESES AGO

```
ini = ""  
  
loop do  
  puts ini  
  calc = ["[1]Soma",  
          "[2]Subtração",  
          "[3]Adição", "[4]Divisão",  
          "[5]Modulo da divisão",  
          "[0]Sair", "Opção: "]  
  for calculadora in calc  
    puts calculadora  
  end  
  entrada =  
  gets.chomp.to_i  
  if entrada == 1  
    print "Digite um  
    número> "  
    n1 = gets.chomp.to_i  
    print "Digite outro  
    número> "  
    n2 = gets.chomp.to_i  
    s = n1 * n2  
  
    puts "A soma de #{n1} e  
    #{n2} é> #{s}" MARCAR ESTA UN  
    elsif entrada == 2 COMO CONCL  
    puts "Divisão"
```

Feedback

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

```

print "Digite um
número> "
n1 = gets.chomp.to_i
print "Digite outro
número> "
n2 = gets.chomp.to_i
s = n1 - n2
puts "A Subtração de #
{n1} e #{n2} é > #{s}"
elsif entrada == 3
print "Digite um
número> "
n1 = gets.chomp.to_i
print "Digite outro
número> "
n2 = gets.chomp.to_i
s = n1 + n2
puts "A Adição de #{n1}
e #{n2} é > #{s}"
elsif entrada == 4
print "Digite um
número> "
n1 = gets.chomp.to_i
print "Digite outro
número> "
n2 = gets.chomp.to_i
s = n1 / n2
puts "A Divisão de #{n1}
e #{n2} é > #{s}"
elsif entrada == 5
print "Digite um
número> "
n1 = gets.chomp.to_i
print "Digite outro
número> "
n2 = gets.chomp.to_i
s = n1 % n2
puts "O modulo da
divisão de #{n1} e #{n2}
é > #{s}" MARCAR ESTA UN
COMO CONCL
elsif entrada == 0

```

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

REPLY

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems



Programação Orientada a Objetos I



Pedro Couto

7 MESES AGO

##

Calculadora Básica ##

```
result = "
loop do
  puts result
  puts "
  puts
  *****
  puts 'Calculadora'
  puts 'Básica!!!'
  puts 'Selecione uma'
  puts 'das seguintes opções'
  puts '0- Sair'
  puts '1- Somar'
  puts '2- Subtrair'
  puts '3- Multiplicar'
  puts '4- Dividir'
  puts '5- Potenciação'
  print 'Opção: '

  option =
  gets.chomp.to_i

  case option
  when 0
    break
  when 1
    puts 'Legal! Você
    selecionou Soma.'
  when 2
    puts 'Digite o primeiro
    número:' MARCAR ESTA UN
  when 3
    puts 'Digite os dois
    números:' COMO CONCL
  when 4
    puts 'Digite os dois
    números:' COMO CONCL
  when 5
    puts 'Digite o expoente
    e a base:' COMO CONCL
  end
end
```

Feedback

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Feedback

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

```
gets.chomp.to_i
puts 'Digite o segundo
número: '
number2 =
gets.chomp.to_i
result = number1 +
number2
puts 'Resultado da
soma: '
when 2
puts 'Legal! Você
selecionou Subtrair.'
puts 'Digite o primeiro
número: '
number1 =
gets.chomp.to_i
puts 'Digite o segundo
número: '
number2 =
gets.chomp.to_i
result = number1 -
number2
puts 'Resultado da
subtração: '
when 3
puts 'Legal! Você
selecionou Multiplicar.'
puts 'Digite o primeiro
número: '
number1 =
gets.chomp.to_i
puts 'Digite o segundo
número: '
number2 =
gets.chomp.to_i
result = number1 *
number2
puts 'Resultado da
multiplicação: ' MARCAR ESTA UN
when 4 COMO CONCL
puts 'Legal! Você'
```

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems



Programação Orientada a
Objetos I



selecionou Dividir.'
puts 'Digite o primeiro
número: '
number1 =
gets.chomp.to_i
puts 'Digite o segundo
número: '
number2 =
gets.chomp.to_i
result =
number1/number2
puts 'Resultado da
divisão: '
when 5
puts 'Legal! Você
selecionou
Potenciação.'
puts 'Digite o primeiro
número: '
number1 =
gets.chomp.to_i
puts 'Digite o segundo
número: '
number2 =
gets.chomp.to_i
result = number1 **
number2
puts 'Resultado da
potenciação: '
else
puts 'Não foi possível
identificar'
end
end

REPLY



Jelson Correia de

Souza

8 MESES AGO

** INTERACTIVE

MARCAR ESTA UN
COMO CONCL

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%



Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems



Programação Orientada a Objetos I



```
CALCULATOR **'
puts "
result = "
loop do
  puts result
  puts "
  puts 'Select you option:
'
  puts "
  puts '1 – Calculate
Operation'
  puts '0 – Exit'
  puts "
  print "Option:"
  option =
  gets.chomp.to_i
  case option
  when 1
    print "Enter the first
number integer: "
    firstNumber =
    gets.chomp.to_i
    print "Enter the second
number integer: "
    secondNumber =
    gets.chomp.to_i
    puts "
    print "Select type
operation: "
    puts "
    puts '1 – Addition'
    puts '2 – Subtraction'
    puts '3 – Division'
    puts '4 – Multiplication'
    puts "
    puts 'loading ...'
    puts "
    MARCAR ESTA UN
    COMO CONCL
    typeOperation =
```

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

- O Início
- Conceitos Básicos

Estruturas de Controle

Collections

Métodos e Gems

Programação Orientada a Objetos I

```
gets.chomp.to_i
```

```
if typeOperation == 1
  result = firstNumber +
  secondNumber
```

```
elsif typeOperation == 2
  result = firstNumber -
  secondNumber
```

```
elsif typeOperation == 3
  result = firstNumber /
  secondNumber
```

```
elsif typeOperation == 4
  result = firstNumber *
  secondNumber
```

```
else
  puts 'Invalid operation
type!'
end
```

```
result = "The result of
operation is equal: #
{result}"
```

```
when 0
  puts 'Done! Bye, bye ...'
  break
else
  puts 'Option invalid!'
  system "clear"
end
system "clear"
end
```

REPLY



Filipe 8 MESES AGO

#calculadora

#eu seu que fez uma
complicação doida aqui
kk

MARCAR ESTA UN
COMO CONCL

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

RESULT -

```
loop do
  puts result
  puts 'Selecione uma
  das seguintes opções'
  puts '1- Usar
  calculadora'
  puts '0- Sair'
  print 'Opção: '
```



```
option =
gets.chomp.to_i
```

```
system "clear"
```

```
if option == 1
#pede para que o
usuário entre com os
números para serem
calculados
puts "Informe os
números a serem
usados no cálculo."
```

```
print "Digite um
número: "
```

```
number1 =
gets.chomp.to_i
print "Digite outro
número: "
```

```
number2 =
gets.chomp.to_i
#pergunta qual
operação o usuário
quer realizar
puts "Selecione a
operação: "
```

```
puts "1- para
multiplicar"
puts "2- para dividir"
puts "3- para somar"
```

```
puts "4- para subtrair"
operator =
```

```
gets.chomp.to_i
MARCAR ESTA UN
COMO CONCL
```

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems



Programação Orientada a
Objetos I



Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

```
case operator
when 1
  result = number1 *
  number2
  print "O resultado da
multiplicação é: "
when 2
  result = number1 /
  number2
  print "O resultado da
divisão é: "
when 3
  result = number1 +
  number2
  print "O resultado da
soma é: "
when 4
  result = number1 -
  number2
  print "O resultado da
subtração é: "
else
  puts "Opção inválida"
end

elsif option == 0
break if option == 0
else
  result = 'Opção inválida'
end

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

REPLY



Filipe 8 MESES

AGO

MARCAR ESTA UNIÃO CONCLUI
eu sei que
eu fiz uma

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início

Conceitos Básicos

Estruturas de Controle

Collections

Métodos e Gems

Programação Orientada a Objetos I

complicação
doida
aqui*****

REPLY

Feedback



JOSEFRAN OLIVEIRA
DE ARAUJO

8 MESES AGO

`resultado = ""`

```
loop do
  puts resultado
  puts "escolha uma das
  opções a baixo"
  puts "1- adição"
  puts "2- subtração"
  puts "3- multiplicação"
  puts "4- divisão"
  puts "5- sair"
```

```
opção =
gets.chomp.to_i
# é isso ?
if opção == 1
  print "digite o primeiro
  número: "
  numero1 =
  gets.chomp.to_i
  print "digite o segundo
  número: "
  numero2 =
  gets.chomp.to_i
  resultado = numero1 +
  numero2
  puts "o resultado da
  adição de #{numero1} +
  #{numero2} é #
  {resultado}"
```

```
# então é isso?
elsif opção == 2
  print "digite o primeiro
```

MARCAR ESTA UN
COMO CONCL

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início

Conceitos Básicos

Estruturas de Controle

Collections

Métodos e Gems

Programação Orientada a Objetos I

numero:

numero1 =

gets.chomp.to_

print "digite o segundo
número: "

numero2 =

gets.chomp.to_i

resultado = numero1 -
numero2puts "o resultado da
subtração de #

{numero1} - #

{numero2} é #

{resultado}"

elsif opção == 3
print "digite o primeiro
número: "

numero1 =

gets.chomp.to_i

print "digite o segundo
número: "

numero2 =

gets.chomp.to_i

resultado = numero1 *
numero2puts "o resultado da
multiplicação de #

{numero1} * #

{numero2} é #

{resultado}"

#É IGUAL A ISSO

elsif opção == 4

print "digite o primeiro
número: "

numero1 =

gets.chomp.to_i

print "digite o segundo
número: "

numero2 =

gets.chomp.to_i

resultado = numero1 /

Feedback

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início

Conceitos Básicos

Estruturas de Controle

Collections

Métodos e Gems

Programação Orientada a Objetos I

```
numero2
puts "o resultado da
divisão de #{numero1} /
#{numero2} é #
{resultado}"
```

```
elsif opção == 5
break
else #SE NAO
resultado = "opção
inválida"
end
```

```
system clear
end
```

REPLY

Rafael Tino 9 MESES AGO
result = "

```
loop do
puts result
puts 'Selecione a
operação que deseja
realizar:'
puts '1- Adição'
puts '2- Subtração'
puts '3- Multiplicação'
puts '4- Divisão'
puts '0- Sair'
print 'Opção: '

option =
gets.chomp.to_i

case option
```

```
when 0
break
when 1
```

```
print 'Digite um número:
'
'MARCAR ESTA UN
num1 = gets.chomp.to_i
COMO CONCL
print 'Digite outro'
```

Feedback

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

- O Início
- Conceitos Básicos

- Estruturas de Controle

- Collections

- Métodos e Gems

- Programação Orientada a Objetos I

```

print 'Digite outro
número: '
num2 = gets.chomp.to_i
resultado = num1 +
num2
result = "#{num1} + #
{num2} = #{resultado}"
when 2
print 'Digite um número:
'

```

```

num1 = gets.chomp.to_i
print 'Digite outro
número: '
num2 = gets.chomp.to_i
resultado = num1 -
num2
result = "#{num1} - #
{num2} = #{resultado}"
when 3
print 'Digite um número:
'

```

```

num1 = gets.chomp.to_i
print 'Digite outro
número: '
num2 = gets.chomp.to_i
resultado = num1 *
num2
result = "#{num1} x #
{num2} = #{resultado}"
when 4
print 'Digite o dividendo:
'

```

```

num1 = gets.chomp.to_i
print 'Digite divisor: '
num2 = gets.chomp.to_i
resultado = num1 /
num2
resto = num1 % num2
result = "#{num1} / #
{num2} = #{resultado} e
sobra #{resto}"

```

MARCAR ESTA UN
COMO CONCL

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

[REPLY](#)

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems



Programação Orientada a Objetos I



```

else
  result 'Opção inválida'
end
system "clear"
end

```

Rafael Tino 9 MESES AGO

```

result = "
loop do
  puts result
  puts 'Selecione a
  operação que deseja
  realizar:'
  puts '1- Adição'
  puts '2- Subtração'
  puts '3- Multiplicação'
  puts '4- Divisão'
  puts '0- Sair'
  print 'Opção: '

  option =
  gets.chomp.to_i
  if option == 1
    print 'Digite um número:
    '
    num1 = gets.chomp.to_i
    print 'Digite outro
    número: '
    num2 = gets.chomp.to_i
    resultado = num1 +
    num2
    result = "#{num1} + #
    {num2} = #{resultado}"
  elsif option == 2
    print 'Digite um número:
    '
    num1 = gets.chomp.to_i
    print 'Digite outro COMO CONCL
    número: '

```

MARCAR ESTA UN

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

```
num2 = gets.chomp.to_i
resultado = num1 -
num2
result = "#{num1} - #"
{num2} = #{resultado}"
elsif option == 3
print 'Digite um número:
'
```

```
num1 = gets.chomp.to_i
print 'Digite outro
número: '
num2 = gets.chomp.to_i
resultado = num1 *
num2
result = "#{num1} x #"
{num2} = #{resultado}"
elsif option == 4
print 'Digite o dividendo:
'
```

```
num1 = gets.chomp.to_i
print 'Digite divisor: '
num2 = gets.chomp.to_i
resultado = num1 /
num2
resto = num1 % num2
result = "#{num1} / #"
{num2} = #{resultado} e
sobra #{resto}"
elsif option == 0
break
else
result 'Opção inválida'
end
system "clear"
end
```

REPLY



Renan Amorim MENSAS
ESTA UN
AGO
COMO CONCL
loop do

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems



Programação Orientada a
Objetos I



```
puts 'Selecione uma  
das seguintes opções'  
puts '1- Somar'  
puts '2- Subtrair'  
puts '3- Multiplicar'  
puts '4- Dividir'  
puts '0- Sair'  
print 'Opção: '
```

```
option =  
gets.chomp.to_i  
  
if option == 1  
print 'Digite o primeiro  
número: '  
number1 =  
gets.chomp.to_i  
print 'Digite o segundo  
número: '  
number2 =  
gets.chomp.to_i  
result = number1 +  
number2  
puts "O resultado da  
soma de #{number1} +  
#{number2} é #{result}"
```

```
elsif option == 2  
print 'Digite o primeiro  
número: '  
number1 =  
gets.chomp.to_i  
print 'Digite o segundo  
número: '  
number2 =  
gets.chomp.to_i  
result = number1 -  
number2  
puts "O resultado da  
soma de #{number1} -  
#{number2} é #{result}"
```

MARCAR ESTA UN
COMO CONCL

```
elsif option == 3
```

Feedback

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems



Programação Orientada a
Objetos I



```
print 'Digite o primeiro
número: '
number1 =
gets.chomp.to_i
print 'Digite o segundo
número: '
number2 =
gets.chomp.to_i
result = number1 *
number2
puts "O resultado da
soma de #{number1} *
#{number2} é #{result}"
```

```
elsif option == 4
print 'Digite o primeiro
número: '
number1 =
gets.chomp.to_i
print 'Digite o segundo
número: '
number2 =
gets.chomp.to_i
result = number1 /
number2
puts "O resultado da
soma de #{number1} /
#{number2} é #{result}"
```

```
elsif option == 0
break
else
puts 'Opção Inválida'
break
end
end
```

REPLY



Tatiane Reis

9 MESES AGO

MARCAR ESTA UN

loop do

COMO CONCL

puts ' Calculadora'

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início

Conceitos Básicos

Estruturas de Controle

Collections

Métodos e Gems

Programação Orientada a Objetos I

```
puts 'Escolha uma das opções abaixo'
puts '1 - Adição 2 -
Subtração 3 -
Multiplicação 4 -
Divisão 5- Sair'
```

```
opcao =
gets.chomp.to_i
```

```
if opcao == 1
print 'Número: '
num= gets.chomp.to_i
print 'Número: '
num2 = gets.chomp.to_i
result = num + num2
puts "Resultado: #
{num} + #{num2} = #
{result}"
```

```
elsif opcao == 2
print 'Número: '
num= gets.chomp.to_i
print 'Número: '
num2 = gets.chomp.to_i
result = num - num2
puts "Resultado: #
{num} - #{num2} = #
{result}"
```

```
elsif opcao == 3
print'Número: '
num= gets.chomp.to_i
print 'Número: '
num2 = gets.chomp.to_i
result = num * num2
puts "Resultado: #
{num} * #{num2} = #
{result}"
```

```
elsif opcao == 4
print 'Número: '
```

MARCAR ESTA UN
COMO CONCL

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início

Conceitos Básicos

Estruturas de Controle

Collections

Métodos e Gems

Programação Orientada a Objetos I

num= gets.chomp.to_i

```
print 'Número: '
num2 = gets.chomp.to_i
result = num / num2
puts "Resultado: #
{num} / #{num2} = #
{result}"
```

elsif opcao == 5

break

else

puts'Opção Inválida'

break

end

end

REPLY

Ivan Júnior Carlota 9

MESES AGO

result = "

loop do

puts result

puts 'Seja bem vindo a
Calculadora de dois
números!'

puts '1- Adição'

puts '2- Subtração'

puts '3- Multiplicação'

puts '4- Divisão'

puts '0- Sair'

print 'Opção: '

option =

gets.chomp.to_i

case option

when 0

break

when 1

print 'Digite o 1 número:
'

MARCAR ESTA UN
COMO CONCL

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

```
number1 =
gets.chomp.to_i
print 'Digite o 2 número:
'
```

```
number2 =
gets.chomp.to_i
addition = number1 +
number2
result = "A Adição de #
{number1} e #
{number2}, é: #
{addition}."
```

```
when 2
print 'Digite o 1 número:
'
```

```
number1 =
gets.chomp.to_i
print 'Digite o 2 número:
'
```

```
number2 =
gets.chomp.to_i
```

```
subtraction = number1
- number2
result = "A Subtração de
#{number1} e #
{number2}, é: #
{subtraction}."
```

```
when 3
print 'Digite o 1 número:
'
```

```
number1 =
gets.chomp.to_i
print 'Digite o 2 número:
'
```

```
number2 =
gets.chomp.to_i
```

```
multiplication = MARCAR ESTA UNI
COMO CONCL
number1 * number2
result = "A Multiplicação
```

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

de #{number1} e #
{number2}, é: #
{multiplication}.”

when 4
print ‘Digite o 1 número:
,

number1 =
gets.chomp.to_i
print ‘Digite o 2 número:
,

number2 =
gets.chomp.to_i
division = number1 /
number2
result = “A Divisão de #
{number1} e #
{number2}, é: #
{division}.”

else
result = ‘Opção inválida’
end
Comando que limpa o
console
system “clear”
end

REPLY

Renata Souza 10 MESES
AGO

result = “ ”
loop do
puts result
puts “Selecione a
operação que deseja
fazer:”

puts “1- Soma”

puts “2- Subtração”

MARCAR ESTA UN

puts “3- Divisão”

COMO CONCL

puts “4- Multipliacação”

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



```
puts "Digite um número inteiro: "
num1 = gets.chomp.to_i
```

Conceitos Básicos



```
num2 = gets.chomp.to_i
```

Estruturas de Controle



```
puts "A soma entre # {num1} e #{num2} é igual a #{num1 + num2}"
```

Collections



```
break
```

Métodos e Gems



```
when 2
  puts "Digite um número inteiro: "
  num1 = gets.chomp.to_i
```

Programação Orientada a Objetos I



```
puts "Digite outro número inteiro: "
num2 = gets.chomp.to_i
```

```
puts "A Subtração entre #{num1} e #{num2} é igual a #{num1 - num2}"
```

```
break
```

```
when 3
```

```
  puts "Digite um número inteiro: "
  num1 = gets.chomp.to_i
```

```
  puts "Digite outro número inteiro: "
```

MARCAR ESTA UNIÃO COMO CONCLUIDA

```
  num2 = gets.chomp.to_i
```

Feedback

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

REPLY

```
puts "A Divisão entre #  
{num1} e #{num2} é  
igual a #{num1 /  
num2}"  
break  
when 4  
puts "Digite um número  
inteiro:"  
num1 = gets.chomp.to_i
```

```
puts "Digite outro  
número inteiro:"  
num2 = gets.chomp.to_i
```

```
puts "A Multipliacaçao  
entre #{num1} e #  
{num2} é igual a #  
{num1 * num2}"
```

```
break  
else
```

```
puts "Opção inválida"  
break  
system "clear"  
end  
end
```



Eulania Soares de
Oliveira Costa

12 MESES AGO

```
loop do  
  puts "\n >>> Olá, você  
  está utilizando a  
  calculadora inteligente!  
<< O resultado da soma  
  é #{res} \n"  
  elsif ope == 2  
    puts "Qual o primeiro
```

```
número?"  
  num1 = gets.chomp.to_i  
  puts "Qual o segundo  
  número?"
```

Feedback

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

```
num2 = gets.chomp.to_i
res = num1 - num2
```

```
print "-> O resultado da
subtração é #{res} \n"
elsif ope == 3
```

```
puts "Qual o primeiro
número?"
```

```
num1 = gets.chomp.to_i
puts "Qual o segundo
número?"
```

```
num2 = gets.chomp.to_i
res = num1 * num2
print "-> O resultado da
multiplicação é #{res}
\n"
```

```
elsif ope == 4
puts "Qual o primeiro
número?"
```

```
num1 = gets.chomp.to_i
puts "Qual o segundo
número?"
```

```
num2 = gets.chomp.to_i
res = num1 / num2
print "-> O resultado da
divisão é #{res} \n"
```

```
elsif ope == 0
break
else
print "-> Opção inválida!
\n"
end
end
```

REPLY



Eulania Soares de
Oliveira Costa

12 MESES AGO

loop do MARCAR ESTA UN
 puts "\n >> Olá, você
 está utilizando a

Feedback

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início

Conceitos Básicos

Estruturas de Controle

Collections

Métodos e Gems

Programação Orientada a Objetos I

calculadora inteligente!

<< O resultado da soma é #{res} \n"

elsif ope == 2

puts "Qual o primeiro número?"

num1 = gets.chomp.to_i

puts "Qual o segundo número?"

num2 = gets.chomp.to_i

res = num1 - num2

print "-> O resultado da subtração é #{res} \n"

elsif ope == 3

puts "Qual o primeiro número?"

num1 = gets.chomp.to_i

puts "Qual o segundo número?"

num2 = gets.chomp.to_i

res = num1 * num2

print "-> O resultado da multiplicação é #{res}

\n"

elsif ope == 4

puts "Qual o primeiro número?"

num1 = gets.chomp.to_i

puts "Qual o segundo número?"

num2 = gets.chomp.to_i

res = num1 / num2

print "-> O resultado da divisão é #{res} \n"

elsif ope == 0

break

else

print "-> Opção inválida!

\n"

end

end

MARCAR ESTA UN
COMO CONCL

REPLY

Ruby Puro

**Tempo Restante : UNLIMITED
ACCESS**

20.00%
▼

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems



Programação Orientada a
Objetos I



Heverton Santos
Costa

12 MESES AGO

Feedback
:

Utilizando as
estruturas de iteração e
condição crie uma
calculadora que
ofereça ao usuário a
opção de multiplicar,
dividir, adicionar ou
subtrair dois números.

Não se esqueça
de também permitir
que o usuário feche o
programa.

result = "

```
loop do
  print result
  puts "
  puts 'Calculadora:'
  puts '1 – Soma'
  puts '2 – subtração'
  puts '3 – Multiplicação'
  puts '4 – Divisão'
  print "-----\n\n"
```

```
puts '0 – Caso deseje
sair'
puts 'Escoha a opção
que deseja: '
option =
gets.chomp.to_i
```

```
if option == 0
break
```

end

```
case option
when 1..4
```

MARCAR ESTA UN
COMO CONCL

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início

Conceitos Básicos

Estruturas de Controle

Collections

Métodos e Gems

Programação Orientada a Objetos I

```

print "\n\n"
puts 'Digite o primeiro
número: '
num1 = gets.chomp.to_i

puts 'Digite o segundo
número : '
num2 = gets.chomp.to_i

if option == 1
total = num1 + num2
result = 'O resultado da
SOMA é: ' + total.to_s +
"\n\n"

elsif option == 2
total = num1 - num2
result = 'O resultado da
SUBTRAÇÃO é: ' +
total.to_s + "\n\n"

elsif option == 3
total = num1 * num2
result = 'O resultado da
MULTIPLICAÇÃO é: ' +
total.to_s + "\n\n"

elsif option == 4
total = num1.to_f /
num2
result = 'O resultado da
DIVISÃO é: ' +
total.to_f.to_s + "\n\n"
end
end

system 'clear'
end

```

REPLY



ALEXIS 12 MESES AGO

result MARCAR ESTA UN
COMO CONCL

loop do

Feedback

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

```

puts result
puts "\n"
puts 'Selecione uma
das seguintes opções:'
puts '1 para calcular'
puts '2 para sair'

option =
gets.chomp.to_s

if option == '1'
puts 'Digite um número:
'
num1 = gets.chomp.to_i

puts 'Digite outro
número: '
num2 = gets.chomp.to_i

puts 'Qual a operação?
+, -, /, *'
op = gets.chomp.to_s

case op
when '+'
result = num1 + num2
puts "O resultado é #
{result}"

when '-'
result = num1 - num2
puts "O resultado é #
{result}"

when '/'
result = num1 / num2
puts "O resultado é #
{result}"

when '*'
result = num1 * num2
puts "O resultado é #
{result}"

end
MARCAR ESTA UN
elsif option == '2' COMO CONCL
break

```

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

- O Início
- Conceitos Básicos
- Estruturas de Controle
- Collections
- Métodos e Gems
- Programação Orientada a Objetos I

```
else
result='invalido digite
uma das opções
dadas!'
end
end
```

REPLY



ALEXIS 12 MESES AGO

```
def display
condicao =
resultado = 0
puts "Digite uma das
opções abaixo:"
puts '1 para realizar
operação'
puts '2 para sair'

condicao
=gets.chomp.to_s
if condicao =='1'
puts 'digite o primeiro
número:'
num1 = gets.chomp.to_i
puts 'Digite o segundo
número:'
num2 = gets.chomp.to_i
puts 'Qual a operação
matematica (+, -, /, *)'
será efetuada:'
operacao = gets.chomp
case operacao
when '+'
resultado = num1 +
num2

```

**MARCAR ESTA UNI
COMO CONCL
diplay**

Feedback

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início

Conceitos Básicos

Estruturas de Controle

Collections

Métodos e Gems

Programação Orientada a Objetos I

```
when '-'
resultado = num1 -
num2
puts "A subtração de #{num1} - #{num2} é
igual a #{resultado}"
display
```

```
when '/'
resultado = num1/num2
puts "A divisão de #{num1} / #{num2} é
igual a #{resultado}"
display
```

```
when '*'
resultado = num1 *
num2
puts "A multiplicação
de #{num1} + #{num2}
é igual a #{resultado}"
display
end
```

```
elsif condicao == '2'
puts 'SAINDO'

else
puts "Não sei o que isto
quer dizer"
display
end
end

display
```

REPLY



Roger Albuquerque 1

ANO AGO

loop do
 puts 'Calculadora -'
 Escolha uma opção'

MARCAR ESTA UN
COMO CONCL

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems



Programação Orientada a Objetos I



```

puts "TIAUIÇÃO -"
[2]Subtração -
[3]Multiplicação -
[4]Divisão - [0]Sair'
print 'Qual sua opção: '
value = gets.chomp.to_i

case value
when 1..4
    print 'Digite o primeiro
    numero: '
    num1 = gets.chomp.to_i
    print 'Digite o segundo
    numero: '
    num2 = gets.chomp.to_i

    if value == 1
        result = num1 + num2
    elsif value == 2
        result = num1 - num2
    elsif value == 3
        result = num1 * num2
    elsif value == 4
        result = num1 / num2
    end

    puts "O resultado do
    calculo: #{result}"
    puts ""
    when 0
        break
    else
        puts 'Opção inválida'
        break
    end
end

```

REPLY



Rafael Naves 1 ANO AGO

result = "
MARCAR ESTA UN
COMO CONCL

loop do
 puts result

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems



Programação Orientada a Objetos I



```

puts '..:Calculadora
Digital:..'
puts 'Selecione uma
das seguintes opções:'
puts '1 – Somar!'
puts '2 – Subtrair!'
puts '3 – Multiplicar!'
puts '4 – Dividir!'
puts '0 – Sair'
puts 'Opção:'

option =
gets.chomp.to_i
puts 'Insira o 1º valor'
value1 =
gets.chomp.to_i
puts 'Insira o 2º valor'
value2 =
gets.chomp.to_i

if option == 1
result = "#{value1} + #
{value2} = #
{value1+value2}"
elsif option == 2
result = "#{value1} - #
{value2} = #{value1-
value2}"
elsif option == 3
result = "#{value1} * #
{value2} = #
{value1*value2}"
elsif option == 4
result = "#{value1} / #
{value2} = #
{value1/value2}"
elsif option == 0
break
else
result = "Opção MARCAR ESTA UN
inválida!" COMO CONCL
end

```

[Feedback](#)

Ruby Puro

Tempo Restante : UNLIMITED

ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems



Programação Orientada a Objetos I



system "clear"
end

REPLY

Feedback



Igor Henrique Morais 1

ANO AGO

```
loop do
  puts 'Bem-vindo a
  calculadora foda da
  galera!'
  puts '1 – Somar'
  puts '2 – Subtrair'
  puts '3 – Dividir'
  puts '4 – Multiplicar'
  puts '0 – SAIR'
  puts 'Digite uma das
  opções acima: '
```

```
opcao =
gets.chomp.to_i

case opcao
when 1
  puts 'Digite o primeiro
  valor:'
  soma1 =
  gets.chomp.to_f
  puts 'Digite o segundo
  valor:'
  soma2 =
  gets.chomp.to_f
  resultado_soma =
  soma1 + soma2
  puts "O resultado da
  soma é #
  {resultado_soma}"
when 2
  puts 'Digite o primeiro
  valor:'
```

sub1 = gets.chomp.to_f
 puts 'Digite o segundo
 valor:'

MARCAR ESTA UN
COMO CONCL

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

valor.

Feedback

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

Regina Fonseca 1 ANO

AGO

loop do
puts 'escolha uma
opção:'
puts '1 – iniciar
calculadora'
puts '0 – sair'
puts 'opção:'

opcao =
gets.chomp.to_i

if opcao == 1
print 'entre com o
primeiro número:'
n1 = gets.chomp.to_i
print 'entre com o
segundo número:'
n2 = gets.chomp.to_i

puts 'qual operação?'
puts '1 – SOMA'
puts '2 – SUBTRAÇÃO'
puts '3 –
MULTIPLICAÇÃO'
puts '4 – DIVISÃO'
operacao =
gets.chomp.to_i

if operacao == 1
puts "a SOMA de #{n1}
com #{n2} é: #{n1 +
n2}"
elsif operacao == 2
puts "a SUBTRAÇÃO de
#{n1} com #{n2} é: #{n1
- n2}"
elsif operacao == 3

~~puts "a
MULTIPLICAÇÃO de #
{n1} por #{n2} é: #{n1 *
n2}"~~

MARCAR ESTA UN
COMO CONCL

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início

Conceitos Básicos

Estruturas de Controle

Collections

Métodos e Gems

Programação Orientada a Objetos I

Feedback

```
elsif operacao == 4
  puts "a DIVISÃO de #
{n1} por #{n2} é: #{n1 /
n2}"
else
  puts 'opção inválida'
end
```

```
elsif opcao == 0
  break
else
  puts 'opção inválida'
end
end
```

REPLY



Fernanda Conde 1 ANO AGO

=begin
criar uma calculadora
que ofereça ao usuário
a opção de multiplicar,
dividir,
adicionar ou subtrair
dois números.
=end

```
puts "### programa de
calculadora ###"
print "digite o primeiro
numero: "
num1 = gets.chomp.to_i
print "digite o segundo
numero: "
num2 = gets.chomp.to_i
loop do
```

puts "1 - soma"

puts "2 - subtração"

puts "3 - multiplicação"

puts "4 - divisão"

puts "0 - sair"

MARCAR ESTA UN
CÓMO CONCL

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

```
print "escolha uma
opção:"
op = gets.chomp.to_i
```

```
case op
when 1
soma = num1 + num2
puts "a soma de #
{num1} e #{num2} é: #
{soma}"
```

```
when 2
subt = num1 - num2
puts "a subtração de #
{num1} e #{num2} é: #
{subt}"
```

```
when 3
mult = num1 * num2
puts "a multiplicação de
#{num1} e #{num2} é: #
{mult}"
```

```
when 4
div = num1 / num2
puts "a divisão de #
{num1} e #{num2} é: #
{div}"
```

```
when 0
break
else
puts "### digite uma
opção válida ###"
end
end
```

REPLY



Fernanda Conde 1 ANO

AGO

<https://github.com/fernandaconde> MARCAR ESTA UN
COMO CONCL

REPLY

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems



Programação Orientada a
Objetos I



Lauriano Silva 1 ANO AGO

resultado_total = "

```
loop do
  puts resultado_total
  puts 'Selecione uma
  das seguintes opções'
  puts '1 – Somar'
  puts '2 – Subtrair'
  puts '3 – dividir'
  puts '4 – multiplicar'
  puts '0 – sair'
  print 'Opção: '
```

```
option =
gets.chomp.to_i
if option == 1
print 'Digite aqui o
primeiro numero: '
x = gets.chomp.to_i
print 'Digite aqui o
segundo numero: '
y = gets.chomp.to_i
re = x + y
resultado_total = "O
valor da adição é: #{re}"
```

```
elsif option == 2
print 'Digite aqui o
primeiro numero: '
x = gets.chomp.to_i
print 'Digite aqui o
segundo numero: '
y = gets.chomp.to_i
re = x - y
resultado_total = "O
valor da subtração é: #
{re}"
```

elsif option == 3

print 'Digite aqui o
primeiro numero: 'COMO CONCL
x = gets.chomp.to_i

Feedback

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

```
print 'Digite aqui o
segundo numero: '
y = gets.chomp.to_i
re option = x / y
resultado_total = "O
valor da divisão é: #{re}"
```



elsif option == 4

print 'Digite aqui o

primeiro numero: '

x = gets.chomp.to_i

print 'Digite aqui o

segundo numero: '

y = gets.chomp.to_i

re = x * y

resultado_total = "O

valor da multiplicação

é: #{re}"

elsif option == 0

break

else

break

end

system = 'clean'

end

REPLY



Cristiano Bosenbecker

Hellwig

1 ANO AGO

result = "

loop do

puts result

puts "Calculadora"

puts '1 + => Somar'

puts '2 - => Subtrair'

puts '3 * => Multiplicar'

puts '4 / => Dividir'

puts '0 => Sair'

MARCAR ESTA UN

COMO CONCL

puts 'Opcao: '

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Feedback

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

```

op = gets.chomp.to_i

if op == 1
  print 'Primeiro Valor: '
  value1 =
  gets.chomp.to_i
  print 'Segundo Valor: '
  value2 =
  gets.chomp.to_i
  soma = value1 + value2
  result = "Resultado de #{value1} + #{value2} = #{soma}"
end

elsif op == 2
  print 'Primeiro Valor: '
  value1 =
  gets.chomp.to_i
  print 'Segundo Valor: '
  value2 =
  gets.chomp.to_i
  subtrai = value1 - value2
  result = "Resultado de #{value1} - #{value2} = #{subtrai}"
end

elsif op == 3
  print 'Primeiro Valor: '
  value1 =
  gets.chomp.to_i
  print 'Segundo Valor: '
  value2 =
  gets.chomp.to_i
  multiplica = value1 *
  value2
  result = "Resultado de #{value1} * #{value2} = #{multiplica}"
end

```

```

elsif op == 4
  print 'Primeiro Valor: '
  value1 =

```

MARCAR ESTA UN
COMO CONCL

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início

Conceitos Básicos

Estruturas de Controle

Collections

Métodos e Gems

Programação Orientada a Objetos I

```
gets.chomp.to_i
print 'Segundo Valor: '
value2 =
gets.chomp.to_i
dividi = value1 / value2
result = "Resultado de #
{value1} / #{value2} = #
{dividi}"
```

```
elsif op == 0
break
else
puts 'Opção invalida'
end
system "clear"
end
```

REPLY



Douglas 1 ANO AGO
Realizei a missão utilizando “Case”. Gostaria de saber se é possível solicitar uma única vez os números de entrada em vez de solicitar dentro de cada estrutura do “when”. Segue meu código com a resolução:

```
result = "
loop do
puts result
puts 'Qual operação matemática você deseja fazer?'
puts '1 – Soma'
puts '2 – Subtração'
puts '3 – Multiplicação'
puts '4 – Divisão'
MANCAR ESTA UN
COMO CONCL
puts '0 – Sair'
```

Feedback

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

```

print 'Opção: '
option =
gets.chomp.to_i

case option
when 0
break
when 1
print 'Digite o primeiro
número: '
num1 = gets.chomp.to_i
print 'Digite o segundo
número: '
num2 = gets.chomp.to_i
res = num1+num2
result = "O resultado da
soma é #{res}!"
when 2
print 'Digite o primeiro
número: '
num1 = gets.chomp.to_i
print 'Digite o segundo
número: '
num2 = gets.chomp.to_i
res = num1-num2
result = "O resultado da
subtração é #{res}!"
when 3
print 'Digite o primeiro
número: '
num1 = gets.chomp.to_i
print 'Digite o segundo
número: '
num2 = gets.chomp.to_i
res = num1*num2
result = "O resultado da
multiplicação é #{res}!"

when 4
print 'Digite o primei
número: '
num1 = gets.chomp.to_i

```

Feedback

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



REPLY

Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

```
print 'Digite o segundo
número: '
num2 = gets.chomp.to_i
res = num1/num2
result = "O resultado da
divisão é #{res}!"
```

else

result = 'Opção inválida'

end

Comando que limpa o
console

system "clear"

end

Thiago 1 ANO AGO

puts

'Calculadora Ruby'

result = "

loop do

puts result

puts 'Escolha a opção
desejada:'

puts '1 – Calcular'

puts '0 – Sair'

puts 'Opçao:'

option =

gets.chomp.to_i

operation = "

if option == 1

puts 'Digite o primeiro
numero: '

number01 =

gets.chomp.to_f

loop do

puts 'Digite a operação

(+ - / *)'

operation = gets.chomp

MARCAR ESTA UN
COMO CONCL

break if (operation ==

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

0 Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

```
'+') or (operation == '-')
or (operation == '/') or
(operation == '*'))
puts 'Opção inválida,
digite novamente.'
end
```

```
puts 'Digite o segundo
numero: '
```

```
number02 =
```

```
gets.chomp.to_f
```

```
if operation == '+'
result = number01 +
```

```
number02
```

```
elsif operation == '-'
result = number01 -
```

```
number02
```

```
elsif operation == '/'
result = number01 /
```

```
number02
```

```
else
```

```
result = number01 *
```

```
number02
```

```
end
```

```
system 'clear'
```

```
puts "#{number01} #
```

```
{operation} #
```

```
{number02} = #{result}"
```

```
elsif option == 0
break if option == 0
```

```
else
```

```
result = 'Opção
```

```
inválida!'
```

```
system 'clear'
```

```
end
```

```
end
```

REPLY



Eduardo abraao 1 ANO

MARCAR ESTA UN

AGO

COMO CONCL

UÍLIO

loop do

n "1 – Para adicionar: "

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

p "2 – Para subtrair:
p "3 – Para multiplicar:
p "4 – Para dividir:
p "0 – Sair"

calculadora =
gets.chomp.to_i

break if calculadora ==
0

case calculadora
when 1
p "Digite o primeiro
valor:
num1 = gets.chomp.to_i
p "Digite o segundo
valor:
num2 = gets.chomp.to_i

p num1+num2
when 2
p "Digite o primeiro
valor:
num1 = gets.chomp.to_i

p "Digite o segundo
valor:
num2 = gets.chomp.to_i
p num1-num2
when 3

p "Digite o primeiro
valor:
num1 = gets.chomp.to_i
p "Digite o segundo
valor:
num2 = gets.chomp.to_i
p num1*num2
when 4

p "Digite o primeiro
valor:
num1 = gets.chomp.to_i
p "Digite o segundo
valor:
num2 = gets.chomp.to_i

MARCAR ESTA UN
COMO CONCL

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

REPLY

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

Joaо Guilherme 2 ANOS
AGO

```

result = "
loop do
puts "Calculadora"
puts "1 – para
continuar"
puts "0 – Para Sair"
op1 = gets.chomp.to_i
if op1 ==1
x = 'sim'
while x == 'sim'
system "clear"
puts "1 – Para Somar"
puts "2 – Para Subtrair"
puts "3 – Para
Multiplicar"
puts "4 – Para Dividir"
puts "0 – Para Voltar ao
menu anterior"
puts "Digite a Opção"
op2 = gets.chomp.to_i
system "clear"
case op2
when 1
puts "Você escolheu
soma"
puts "Digite o Primeiro
numero inteiro"
num1 = gets.chomp.to_i
puts "Digite o Segundo
numero inteiro" MARCAR ESTA UN
COMO CONCL
num2 = gets.chomp.to_i

```

Feedback

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início

Conceitos Básicos

Estruturas de Controle

Collections

Métodos e Gems

Programação Orientada a Objetos I

```

num2 = gets.chomp.to_i
operacao = num1 +
num2
puts result = "A soma é
#{operacao}"
when 2
puts "Você escolheu
subtração"
puts "Digite o Primeiro
numero inteiro"
num1 = gets.chomp.to_i
puts "Digite o Segundo
numero inteiro"
num2 = gets.chomp.to_i
operacao = num1 -
num2
puts result = "A
subtração é #
{operacao}"
when 3
puts "Você escolheu
Multiplicação"
puts "Digite o Primeiro
numero inteiro"
num1 = gets.chomp.to_i
puts "Digite o Segundo
numero inteiro"
num2 = gets.chomp.to_i
operacao = num1 *
num2
puts result = "A
Multiplicação é #
{operacao}"
when 4
puts "Você escolheu
Divisão"
puts "Digite o Primeiro
numero inteiro"
num1 = gets.chomp.to_i
puts "Digite o Segundo
numero inteiro"
  
```

MARCAR ESTA UN
COMO CONCL

Feedback

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início

Conceitos Básicos

Estruturas de Controle

Collections

Métodos e Gems

Programação Orientada a Objetos I

Feedback

```
num2 = gets.chomp.to_i
operacao = num1 /
num2
puts result = "A divisão
é #{operacao}"
else
puts "Opção Inválida"
end
```

```
print "Gostaria de
realizar outra operação?
Sim ou não: "
x = gets.chomp
end
```

```
else
break
end
```

```
system "clear"
end
```

REPLY



Caio Henrique Porcel 2
ANOS AGO

Eu fiz por meio de
funções que interagem
entre si

Ficou dessa forma:

```
def menu
puts
```

```
"-----"
```

```
puts "Bem vindo à
calculadora! "
```

```
puts "Selecione uma
das opções: "
```

```
puts "[0] -> Adição "
```

```
puts "[1] -> Subtração "
```

```
puts "[2] ->
```

```
Multiplicação"
```

```
puts "[3] -> Divisão"
```

```
puts "[4] -> Sair"
```

MARCAR ESTA UN
COMO CONCL

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems



Programação Orientada a
Objetos I



print "Qual escolhe?"

escolha =

gets.chomp.to_i

if escolha == 0

system "clear"

adicao()

end

if escolha == 1

system "clear"

subtracao()

end

if escolha == 2

system "clear"

multi()

end

if escolha == 3

system "clear"

divisao()

end

if escolha == 4

system "clear"

exit

else

puts "Digite uma opção
válida"

menu

end

end

def adicao

puts "- ADIÇÃO -"

print "Digite o primeiro
número: "

num1 = gets.chomp.to_i

print "Digite o segundo
número: "

num2 = gets.chomp.to_i

soma = num1 + num2

puts "A soma entre #
{num1} e #{num2} é #
{soma}"

MARCAR ESTA UN
COMO CONCL

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

SOMAS.

menu

end

def subtracao

puts “— SUBTRAÇÃO —”

print “Digite o primeiro
número: ”

num1 = gets.chomp.to_i

print “Digite o segundo
número: ”

num2 = gets.chomp.to_i

sub = num1 - num2

puts “A subtração entre
#{num1} e #{num2} é #
{sub}.”

menu

end

def multi

puts “—

MULTIPLICAÇÃO —”

print “Digite o primeiro
número: ”

num1 = gets.chomp.to_i

print “Digite o segundo
número: ”

num2 = gets.chomp.to_i

mult = num1 * num2

puts “A multiplicação
entre #{num1} e #
{num2} é #{mult}.”

menu

end

def divisao

puts “— DIVISÃO —”

print “Digite o primeiro
número: ”

num1 =

gets.chomp.to_f

MARCAR ESTA UN
COMO CONCLprint “Digite o segundo
número: ”

Feedback

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

- O Início
- Conceitos Básicos
- Estruturas de Controle
- Collections
- Métodos e Gems
- Programação Orientada a Objetos I

```
num2 =
gets.chomp.to_f
div = num1 / num2
puts "A divisão entre #{num1.to_i} e #{num2.to_i} é #{div}."
menu
end
menu
```

REPLY



Hygor Pereira 2 ANOS
AGO

É tão bacana quando a coisa está fluindo que a gente até se empolga. Pelos comentários a maioria usou o loop, ai resolvi usar o while para ficar com uma dinâmica mais agradável na parte de dar a opção para sair, exigiu mais código e mais interação do usuário, mas é uma outra alternativa bacana pra mudar a parte visual.

```
resp =
while resp != '2'
  puts '+-----+
  +-----+
  puts 'I C A L C U L A D O
  R A I'
  puts '+-----+
  +-----+ MARCAR ESTA UN
  +-----+ COMO CONCL
  puts 'Informe os dados'
```

Feedback

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

```
para a operação'  
print "Priemiro número:  
"
```

```
n1 = gets.chomp.to_i  
print "Segundo número:  
"
```

```
n2 = gets.chomp.to_i  
puts '+
```

```
+'  
puts '| [+] Adição |'  
puts '| [-] Subtração |'  
puts '| [*] Multiplicação |'  
puts '| [/] Divisão |'
```

```
Exponenciação |'  
puts '+
```

```
+'  
print 'Escolha a  
operação de acordo  
com as opções: '  
op = gets.chomp  
puts '+
```

```
+'  
if op == '+'  
resultado = n1 + n2  
elsif op == '-'  
resultado = n1 - n2  
elsif op == '*'  
resultado = n1 * n2  
elsif op == '/'  
r1 = n1 / n2  
r2 = n1 % n2  
resultado = "#{r1} resto
```

```
#'{r2}' MARCAR ESTA UN  
elsif op == '^' COMO CONCL  
resultado = n1 ** n2
```

Feedback

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início

Conceitos Básicos

Estruturas de Controle

Collections

Métodos e Gems

Programação Orientada a
Objetos I

```
else
  resultado = 'Operação
  inválida'
end
puts "#{n1} #{op} #{n2}
= #{resultado}"
puts '+
```

```
+
print 'Informe [1] para
Limpar ou [2] para Sair: '
resp = gets.chomp
system 'clear'
end
```

REPLY



Jose Baito Camacho 2

ANOS AGO

```
result =
resultCalc =
loop do
  puts result
  puts resultCalc
  puts 'Selecione uma
  das seguintes opções'
  puts '1- Abrir a
  calculadora'
  puts '0- Sair'
  print 'Opção: '

  option =
  gets.chomp.to_i
```

```
if option == 1
  print 'Informe o 1º nº: '
  n1 = gets.chomp.to_i
  print 'Informe o 2º nº: '
  n2 = gets.chomp.to_i
```

```
system "cls" MARCAR ESTA UN
loop do COMO CONCL
  puts 'Selecione uma
```

Feedback

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

das seguintes opções'
puts '1- Multiplicação'
puts '2- Adição'
puts '3- Subtração'
puts '4- Divisão'
puts '0- Sair'
print 'Opção: '

```
optionCalc =
gets.chomp.to_i
if optionCalc == 1
resultCalc = "#{n1} X #
{n2} = #{n1 * n2}"
result = "
break
elsif optionCalc == 2
resultCalc = "#{n1} + #
{n2} = #{n1 + n2}"
result = "
break
elsif optionCalc == 3
resultCalc = "#{n1} - #
{n2} = #{n1 - n2}"
result = "
break
elsif optionCalc == 4
resultCalc = "#{n1} / #
{n2} = #{n1 / n2}"
result = "
break
elsif optionCalc == 0
break
else
resultCalc = 'Opção
inválida'
end
system "cls"
end
```

elsif option == 0 MARCAR ESTA UN
break COMO CONCL
else

Feedback

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

```
result = 'Opção inválida'
resultCalc = "
end
system "cls"
end
```

REPLY

Feedback



Zilmar Moraes 2 ANOS
AGO

```
result = "
loop do
puts result
puts "
puts 'Selecione uma
das operações para
realizar o cálculo: '
puts '1 – Adição'
puts '2 – Subtração'
puts '3 – Multiplicação'
puts '4 – Divisão'
puts '0 – Sair'
puts "
```

```
option =
gets.chomp.to_i2
if option != 0
puts "
print 'Informe o primeiro
número: '
num1 = gets.chomp.to_i
```

```
print 'Informe o
segundo número: '
num2 = gets.chomp.to_i
case option
when 1
result = "Resultado da
operação escolhida é: #  
{num1 + num2}" COMO CONCL
when 2
```

MARCAR ESTA UN

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

REPLY



Eduardo 2 ANOS AGO

result = "

```
print 'Enter the one
number: '
one_number =
gets.chomp.to_i
```

```
print 'Enter the second
number: '
second_number =
gets.chomp.to_i
```

```
loop do
puts result
puts 'Select one of the
following options'
puts '1 – Addition'
```

```
puts '2 – Subtraction'
puts '3 – multiplication'
puts '4 – Division'
```

MARCAR ESTA UN
COMO CONCL

Ruby Puro

Tempo Restante : UNLIMITED

ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems



Programação Orientada a Objetos I



```

puts '0 - Go out'

option =
gets.chomp.to_i

if option == 1
addition = one_number
+ second_number

result = "the result of
addiction is: #
{addition}"

elsif option == 2
subtraction =
one_number -
second_number

result = "the result of
subtraction is: #
{subtraction}"

elsif option == 3
multiplication =
one_number *
second_number

result = "the result of
multiplication is: #
{multiplication}"

elsif option == 4
division = one_number /
second_number

result = "the result of
division is: #{division}"

elsif option == 0
break

else
print 'invalid options'
end
MARCAR ESTA UN
COMO CONCL
system "clear"

```

Feedback

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems



Programação Orientada a
Objetos I



end

REPLY



Gustavo 2 ANOS AGO
resultado = "

```
loop do
  puts resultado
  puts 'Selecione uma
  das seguintes opções'
  puts '1- Somar dois
  numero'
  puts '2- Subtrair dois
  numero'
  puts '3- Multiplicar dois
  numero'
  puts '4- Dividir dois
  numero'
  puts '0- Para sair do
  programa'

  option =
  gets.chomp.to_i
```

```
case option
when 1
  string_operacao = '{
    Soma }'
when 2
  string_operacao = '{
    Subtração }'
when 3
  string_operacao = '{
    Multiplicação }'
when 2
  string_operacao = '{
    Divisão }'
when 0
  string_operacao = ""

end
```

MARCAR ESTA UN
COMO CONCL

if option == 1

Feedback

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

- O Início
- Conceitos Básicos
- Estruturas de Controle
- Collections
- Métodos e Gems
- Programação Orientada a Objetos I

```
print 'Digite o primeiro
```

```
numero: '
```

```
numero1 =
```

```
gets.chomp.to_i
```

```
print 'Digite o Segundo
```

```
numero: '
```

```
numero2 =
```

```
gets.chomp.to_i
```

```
soma = numero1 +
```

```
numero2
```

```
resultado = "0
```

```
Resultado da operação
```

```
#{string_operacao} e #{soma}"
```

```
elsif option == 2
```

```
print 'Digite o primeiro
```

```
numero: '
```

```
numero1 =
```

```
gets.chomp.to_i
```

```
print 'Digite o Segundo
```

```
numero: '
```

```
numero2 =
```

```
gets.chomp.to_i
```

```
s_sub = numero1 -
```

```
numero2
```

```
resultado = "0
```

```
Resultado da operação
```

```
#{string_operacao} e #{s_sub}"
```

```
elsif option == 3
```

```
print 'Digite o primeiro
```

```
numero: '
```

```
numero1 =
```

```
gets.chomp.to_i
```

```
print 'Digite o Segundo
```

```
numero: '
```

```
numero2 =
```

```
gets.chomp.to_i
```

```
s_multi = numero1 *
```

```
numero2
```

```
resultado = "0
```

MARCAR ESTA UN

COMO CONCL

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

REPLY

Mikael Bernardo 2 ANOS
AGO

Espero que o meu
código tenha ficado
bom, eu estava vendo
uns comentários e a
maioria foi de if else e
elsif, porém tentei fazer
com Case, e testando
chegou a funcionar,
então segue meu

código.

MARCAR ESTA UN
COMO CONCL

result = "

Resultado da operação

#{string_operacao} e #
{s_multi}"

elsif option == 4

print 'Digite o primeiro
numero: '

numero1 =

gets.chomp.to_i

print 'Digite o Segundo
numero: '

numero2 =

gets.chomp.to_i

s_div = numero1 /
numero2

resultado = "0

Resultado da operação

#{string_operacao} e #
{s_div}"

elsif option == 0

break if option == 0

else

puts "Opção invalida"

end

end

Feedback
:

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

```

loop do
  puts result
  puts 'Selecione uma
  das seguintes opções
  aritméticas:'
  puts '1 - Soma'
  puts '2 - Subtrair'
  puts '3 - Multiplicar'
  puts '4 - Dividir'
  puts '5 - Sair'
  puts 'Opção:'

  option =
  gets.chomp.to_i

  case option
  when 1
    print "Digite o primeiro
    número inteiro: "
    number1 =
    gets.chomp.to_i
    print "Digite o segundo
    numero inteiro: "
    number2 =
    gets.chomp.to_i
    puts 'Resultado das
    seguintes opções entre
    eles'
    result = number1 +
    number2
    puts "Adição = #
    {result}"
  when 2
    print "Digite o primeiro
    número inteiro: "
    number1 =
    gets.chomp.to_i
    print "Digite o segundo
    numero inteiro: "
    number2 = MARCAR ESTA UN
    gets.chomp.to_i COMO CONCL
  end
end
  
```

Feedback

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

```

puts 'Resultado das
seguintes opções entre
eles'
result = number1 -
number2
puts "Subtração = #
{result}"
when 3
print "Digite o primeiro
número inteiro: "
number1 =
gets.chomp.to_i
print "Digite o segundo
numero inteiro: "
number2 =
gets.chomp.to_i
puts 'Resultado das
seguintes opções entre
eles'
result = number1 *
number2
puts "Multiplicação = #
{result}"
when 4
print "Digite o primeiro
número inteiro: "
number1 =
gets.chomp.to_i
print "Digite o segundo
numero inteiro: "
number2 =
gets.chomp.to_i
puts 'Resultado das
seguintes opções entre
eles'
result = number1 /
number2
puts "Divisão = #
{result}"      MARCAR ESTA UN
when 5          COMO CONCL

```



Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

system clear

break

else

result = 'Opção'

Inválida!

end

end



REPLY

 marcelo valada 2 ANOS AGO

```
puts "Qual operação
deseja fazer?"
puts "1 para somar"
puts "2 para subtrair"
puts "3 para multiplicar"
puts "4 para dividir"
puts "0 para sair"
operacao =
gets.chomp.to_i
```

```
while operacao != 0
system "clear"
```

```
if operacao == 1
puts "SOMANDO..."
print "qual o primeiro
numero? "
num1 = gets.chomp.to_i
print "qual o segundo
numero? "
num2 = gets.chomp.to_i
```

```
total = num1 + num2
puts "#{total}"
```

```
puts "deseja continuar?
1-sim ou 2-nao "
print "resposta: "
opcao =
```

```
gets.chomp.to_i
if opcao == 2
break
MARCAR ESTA UN
COMO CONCL
```

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems



Programação Orientada a
Objetos I



Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%



Aulas

O Início



```
puts "deseja continuar?"
```

```
1-sim ou 2-nao "
```

```
print "resposta:"
```

```
opcao =
```

```
gets.chomp.to_i
```

```
if opcao == 2
```

```
break
```

```
end
```

```
elsif operacao == 3
```

```
puts
```

```
"MULTIPLICANDO..."
```

```
print "qual o primeiro
```

```
numero? "
```

```
num1 = gets.chomp.to_i
```

```
print "qual o segundo
```

```
numero? "
```

```
num2 = gets.chomp.to_i
```

```
total = num1 * num2
```

```
puts "#{total}"
```

```
puts "deseja continuar?"
```

```
1-sim ou 2-nao "
```

```
print "resposta:"
```

```
opcao =
```

```
gets.chomp.to_i
```

```
if opcao == 2 MARCAR ESTA UN
```

```
COMO CONCL
```

```
end
```

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

```

elsif operacao == 4
puts "DIVIDINDO..."
print "qual o primeiro
numero? "
num1 = gets.chomp.to_i
print "qual o segundo
numero? "
num2 = gets.chomp.to_i
total = num1 / num2
puts "#{total}"

puts "deseja continuar?
1-sim ou 2-nao "
print "resposta: "
opcao =
gets.chomp.to_i
if opcao == 2
break
end
else
puts "operação
inválida!"
system "clear"
end
end

```

REPLY

Agner 2 ANOS AGO

Meu código pro
desafio ficou da
seguinte forma:

```

option = ''
loop do
puts 'Escolha uma das
seguintes opções'
puts '1 - SOMAR'

```

```

puts '2 - SUBTRAÇÃO'
puts '3 - MULTIPLICAÇÃO'

```

MARCAR ESTA UN
COMO CONCL

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início

Conceitos Básicos

Estruturas de Controle

Collections

Métodos e Gems

Programação Orientada a Objetos I

```

puts '4 – DIVISÃO'
puts '0 – SAIR'
print "Opção: "

option =
gets.chomp.to_i

if option == 1
system "clear"
print 'Qual o valor do
primeiro número: '
n1 = gets.chomp.to_i
print 'Qual o valor do
segundo número: '
n2 = gets.chomp.to_i
res = n1 + n2
puts "A soma entre #
{n1} e #{n2} é: #
{n1+n2}"
puts ''

elsif option == 2
system "clear"
print 'Qual o valor do
primeiro número: '
n1 = gets.chomp.to_i
print 'Qual o valor do
segundo número: '
n2 = gets.chomp.to_i
puts "A subtração entre
#{n1} e #{n2} é: #{n1 -
n2}"
puts ''

elsif option == 3
system "clear"
print 'Qual o valor do
primeiro número: '
n1 = gets.chomp.to_i
print 'Qual o valor do
segundo número: '
n2 = gets.chomp.to_i
puts "A multiplicação
entre #{n1} e #{n2} é: #

```

MARCAR ESTA UN
COMO CONCL

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início

Conceitos Básicos

Estruturas de Controle

Collections

Métodos e Gems

Programação Orientada a Objetos I

Feedback

```
{n1 * n2}"
puts ''
elsif option == 4
system "clear"
print 'Qual o valor do
primeiro número: '
n1 = gets.chomp.to_i
print 'Qual o valor do
segundo número: '
n2 = gets.chomp.to_i
puts "A multiplicação
entre #{n1} e #{n2} é: #
{n1 / n2}"
puts ''
elsif option == 0
break
else
puts 'OPÇÃO INVÁLIDA'
end
end
```

REPLY



Andre Felipe 2 ANOS AGO

#Utilizando as
estruturas de iteração e
condição, crie uma
calculadora que ofereça
#ao usuário a opção de
Multiplicar, Dividir,
Adicionar ou Subtrair
dois números.
Não se esqueça de
também permitir que o
usuário feche o
programa.

result = "

```
loop do
puts "##### MARCAR ESTA UN
resultado da sua COMO CONCL
equação é #{result}
```

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início	<input checked="" type="radio"/>	##### puts "" puts "" puts '#####' Bem vindo a caulculadora #####'
Conceitos Básicos	<input checked="" type="radio"/>	puts ' antes de começarmos preciso que digite dois valores que deseja fazer uma operação' puts ' em seguida seleciona a operação que deseja efetuar' puts ""
Estruturas de Controle	<input type="radio"/>	print ' Digite o primeiro valor: num1 = gets.chomp.to_i puts ""
Collections	<input type="radio"/>	print ' Digite o segunda valor: num2 = gets.chomp.to_i puts ""
Métodos e Gems	<input type="radio"/>	puts 'Escolha uma das opções a baixo' puts '1 – Efetuar uma soma.' puts '2 – Efetuar uma subtração.' puts '3 – Efetuar uma divisão.' puts '4 – efetuar uma multiplicação.' puts '5 – para o programa'
Programação Orientada a Objetos I	<input type="radio"/>	print 'Escolha uma das opções digitando seu numero:'
		MARCAR ESTA UN COMO CONCL

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%



Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

```
option =
gets.chomp.to_i

case option
when 1
result = num1+num2
puts "O valor da soma é
#{result}"
when 2
result = num1-num2
puts "O valor da
subtração é #{result}"
when 3
result= num1/num2
puts "O resultado da
divisão dos dois
numeros é #{result}"
when 4
result = num1*num2
puts "O resultado da
multiplicação é #
{result}"
when 5
break
else
puts 'opção invalida'
end
system "clear"
end
```

REPLY



Andre Felipe 2 ANOS AGO

o meu
fico dessa forma

result = "

loop do

~~puts "#####"~~ eresultado da sua
equação é #{result}MARCAR ESTA UN
COMO CONCL

#####"

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%



Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems



Programação Orientada a
Objetos I



```
puts ""  
puts ""  
puts #####  
Bem vindo a  
caulculadora  
#####  
puts ' antes de  
começarmos preciso  
que digite dois valores  
que deseja fazer uma  
operação'  
puts ' em seguida  
seleciona a operação  
que deseja efetuar'  
puts ""  
print ' Digite o primeiro  
valor: '  
num1 = gets.chomp.to_i  
puts ""  
print ' Digite o segunda  
valor: '  
num2 = gets.chomp.to_i  
puts ""  
puts 'Escolha uma das  
opções a baixo'  
puts '1 – Efetuar uma  
soma.'  
puts '2 – Efetuar uma  
subtração.'  
puts '3 – Efetuar uma  
divisão.'  
puts '4 – efetuar uma  
multiplicação.'  
puts '5 – para o  
programa'  
  
print 'Escolha uma das  
opções digitando seu  
numero: ' MARCAR ESTA UN  
option = COMO CONCL  
option =
```

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%



Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

```
case option
when 1
result = num1+num2
puts "O valor da soma é #{result}"
when 2
result = num1-num2
puts "O valor da
subtração é #{result}"
when 3
result= num1/num2
puts "O resultado da
divisão dos dois
numeros é #{result}"
when 4
result = num1*num2
puts "O resultado da
multiplicação é #
{result}"
when 5
break
else
puts 'opção invalida'
end
system "clear"
end
```

REPLY



Leonardo 2 ANOS AGO
resultado = "

"
loop do
puts resultado
puts
puts

puts "Bem vindo a
CALCULADORA!!!" MARCAR ESTA UN
COMO CONCL
puts "Seleciona uma

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

das opções abaixo:
puts "1 – Cálculo de soma de dois números."

puts "2 – Cálculo de diminuição de dois números."

puts "3 – Cálculo de multiplicação de dois números."

puts "4 – Cálculo de divisão de dois números."

puts "5 – Cálculo de expoente de dois números."

puts "6 – Sair do calculadora."

print "Digite a opção desejada: "

```
option =
gets.chomp.to_i
case option
```

when 1
print "Digite o primeiro número: "

n1 = gets.chomp.to_i
print "Digite o segundo número: "

n2 = gets.chomp.to_i
r1 = n1 + n2
resultado = "O valor da soma foi de #{r1}."

when 2
print "Digite o primeiro número: "
n1 = gets.chomp.to_i

print "Digite o segundo número: "
n2 = gets.chomp.to_i

MARCAR ESTA UN

COMO CONCL

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I $rZ = nZ - nI$ resultado = "O valor da
diminuição foi de #{r2}."

when 3

print "Digite o primeiro
número: "n1 = gets.chomp.to_i
print "Digite o segundo
número: "n2 = gets.chomp.to_i
r3 = n1 * n2resultado = "O valor da
multiplicação foi de #
{r3}."

when 4

print "Digite o primeiro
número: "n1 = gets.chomp.to_i
print "Digite o segundo
número: "n2 = gets.chomp.to_i
r4 = n1 / n2resultado = "O valor da
divisão foi de #{r4}."

when 5

print "Digite o primeiro
número: "n1 = gets.chomp.to_i
print "Digite o segundo
número: "n2 = gets.chomp.to_i
r5 = n1 ** n2resultado = "O valor da
expoente foi de #{r5}."

when 6

break

else

resultado =

MARCAR ESTA UN
COMO CONCL

"Infelizmente essa

Feedback

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início

Conceitos Básicos

Estruturas de Controle

Collections

Métodos e Gems

Programação Orientada a Objetos I

opção não existe.”

end

system “clear”

end

REPLY

Feedback



Gustavo Coelho 2 ANOS AGO

#MISSÃO

result = ""

loop do

puts "1-calcular\n0-sair\n"

op = gets.chomp.to_i

if op == 0

break

elsif op == 1

puts "digite um número:\n"

num1 =

gets.chomp.to_f

puts "1-adição\n2-subtração\n3-multiplicação\n4-divisão\n"

cal = gets.chomp.to_i

puts "\ndigite outro número:\n"

num2 =

gets.chomp.to_f

case cal

when 1

puts "resultado da soma é: #{num1 + num2}"

when 2

puts "resultado da subtração é: #{num1 - num2}"

MARCAR ESTA UNI
COMO CONCL

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início

Conceitos Básicos

Estruturas de Controle

Collections

Métodos e Gems

Programação Orientada a
Objetos I

when 3

```
puts "resultado da
multiplicação é: #{num1 * num2}"
```

when 4

```
puts "resultado da
divisão é: #{num1 / num2}"
```

else

```
puts "opção inválida"
```

end

else

```
puts "opção inválida"
```

end

end

REPLY



Kenzo 2 ANOS

AGO

Minha
Solução

```
option = 'n'
while option
!= 'sair'
```

```
print 'Digite o
primeiro
valor: '
frst =
gets.chomp.to
print 'Digite o
segundo
valor: '
scnd =
gets.chomp.to
```

```
puts
'Selecione a
```

sua opção: '

 MARCAR ESTA UN
 puts '+' COMO CONCL
 puts '-'

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

```

puts '*'
puts '/'
print'Sua
opção: '
option =
gets.chomp
case option
when 1..4
when '+'
puts
"Resultado: #
{frst + scnd}"
when '-'
puts
"Resultado: #
{frst - scnd}"
when '*'
puts
"Resultado: #
{frst * scnd}"
when '/'
puts
"Resultado: #
{frst / scnd}"
end
print 'Digite
"sair" para
sair: '
option =
gets.chomp
end

puts
'Saindo...'

```

REPLY

Frankyston Lins 2 ANOS AGO

MARCAR ESTA UN
Resposta do desafio: COMO CONCL
<https://gist.github.com/fr>

REPLY

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems

Programação Orientada a
Objetos I

Matheus Araujo

Miranda

2 ANOS AGO

Feedback

result = "

```
loop do
  puts result
  puts 'Bem vindo a
  Calculadora Teste,
  selecione o que deseja
  fazer.'
  puts '1 – Mutiplicação'
  puts '2 – Divisão'
  puts '3 – Soma'
  puts '4 – Subtração'
  puts '0 – Para sair da
  calculadora'
```

```
option =
gets.chomp.to_i
```

```
case option
when 1
  puts 'Digite o Primeiro
  numero'
  number1 =
  gets.chomp.to_i
  puts 'Digite o segundo
  numero'
  number2 =
  gets.chomp.to_i
  mutiplicacao =
  (number1 * number2)
  result = "A Mutiplicação
  dos dois numeros é: #
  {mutiplicacao}"
when 2
```

```
puts 'Digite o Primeiro
numero'
number1 =
```

MARCAR ESTA UN
COMO CONCL

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Feedback

Aulas

O Início



Conceitos Básicos



Estruturas de Controle



Collections



Métodos e Gems



Programação Orientada a
Objetos I



```
gets.chomp.to_i
puts 'Digite o segundo
numero'
number2 =
gets.chomp.to_i
divisao = (number1 /
number2)
result = "A Divisão dos
dois numeros é: #
{divisao}"
when 3
puts 'Digite o Primeiro
numero'
number1 =
gets.chomp.to_i
puts 'Digite o segundo
numero'
number2 =
gets.chomp.to_i
soma = (number1 +
number2)
result = "A Soma dos
dois numeros é: #
{soma}"
when 4
puts 'Digite o Primeiro
numero'
number1 =
gets.chomp.to_i
puts 'Digite o segundo
numero'
number2 =
gets.chomp.to_i
subtracao = (number1 -
number2)
result = "A Subtração
dos dois numeros é: #
{subtracao}"
```

when 0 MARCAR ESTA UN
break if option == 0 COMO CONCL
else

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

O Início

Conceitos Básicos

Estruturas de Controle

Collections

Métodos e Gems

Programação Orientada a Objetos I

result = 'Opção Inválida'

end

system "clear"

end

REPLY

Feedback



Carlos Valente 2 ANOS
AGO

resultado = "

loop do

puts resultado

puts "Selecione uma das opções abaixo:"

puts

"=====

puts "[0] Sair"

puts "[1] Multiplicar"

puts "[2] Dividir"

puts "[3] Adicionar"

puts "[4] Subtrair"

opcao =

gets.chomp.to_i

if opcao == 0

break

elsif opcao 4

resultado = "Opção Inválida"

else

print 'Digite o primeiro valor: '

num1 = gets.chomp.to_i

print 'Digite o segundo valor: '

num2 = gets.chomp.to_i

if opcao == 1

~~valor = num1 * num2~~

resultado = "A MARCAR ESTA UN Multiplicação entre # COMO CONCL {num1} e #{num2} é"

Ruby Puro

Tempo Restante : UNLIMITED
ACCESS

20.00%

Aulas

- O Início
- Conceitos Básicos
- Estruturas de Controle
- Collections
- Métodos e Gems
- Programação Orientada a Objetos I

```
igual a #{valor}"  
elsif opcao == 2  
valor = num1 / num2  
resultado = "A Divisão  
entre #{num1} e #{  
num2} é igual a #{  
valor}"  
elsif opcao == 3  
valor = num1 + num2  
resultado = "A Soma  
entre #{num1} e #{  
num2} é igual a #{  
valor}"  
else  
valor = num1 - num2  
resultado = "A Diferença  
entre #{num1} e #{  
num2} é igual a #{  
valor}"  
end  
end  
system "clear"  
end
```

REPLY

LOAD MORE (70)
ASK QUESTION

MARCAR ESTA UN
COMO CONCL