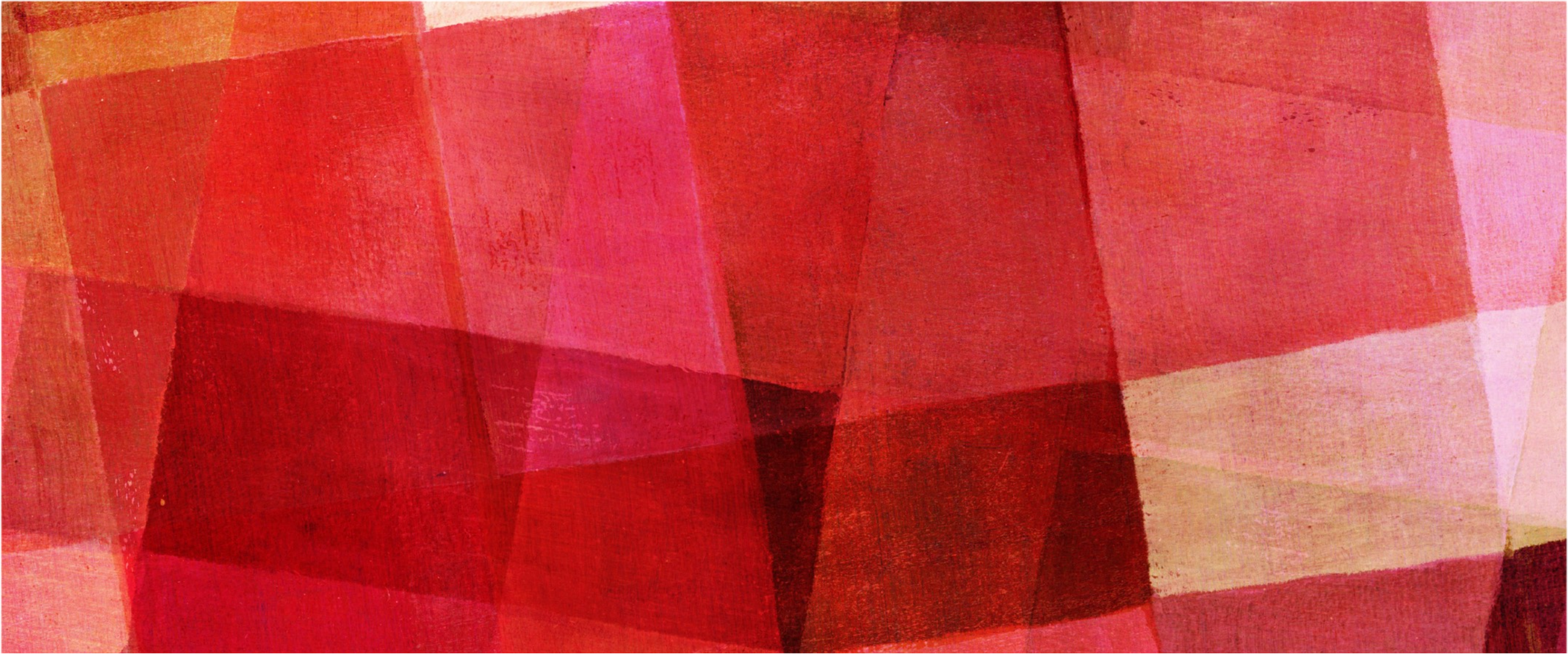

NANO CURSO DE RUBY

DevMaker

12 de Junho de 2017

(oin <3)

RUBY

- Sintaxe limpa
- 100% orientada a objetos
- Tipagem forte e dinâmica
- Amigável
- Documentação acessível
- Ruby <=> Java
- Testes
- Metaprogramação
- Exemplos

RUBY – SINTAXE LIMPA

```ruby
if true
  'if statement'
elsif false
  'else if, optional'
else
  'else, also optional'
end
```

```ruby
while a > b
  puts "A maior que B"
  a -= 1
end
```

```ruby
(1..10).each do |i|
  p i
end
```

```ruby
case grade
when 'A'
  puts 'Way to go kiddo'
when 'B'
  puts 'Better luck next time'
when 'C'..'F'
  puts 'Son, I am disappoint'
else
  puts 'Alternative grading system, eh?'
end
```

```ruby
begin
  # your code here
rescue NoMemoryError => exception_variable
  puts 'NoMemoryError raised', exception_variable
rescue RuntimeError => other_exception_variable
  puts 'RuntimeError raised'
else
  puts 'This runs if no exceptions were thrown at all'
ensure # == finally
  puts 'This code always runs no matter what'
end
```

RUBY - 100% OO

```
irb(main):009:0> 1.class
=> Integer
irb(main):010:0> "1".class
=> String
irb(main):011:0> 1.0.class
=> Float
irb(main):012:0> class Exemplo; end
=> nil
irb(main):013:0> e = Exemplo.new
=> #<Exemplo:0x007fff229fe128>
irb(main):014:0> e.class
=> Exemplo
irb(main):015:0> self
=> main
irb(main):016:0> self.class
=> Object
irb(main):017:0>
```

RUBY - 100% OO

```
irb(main):017:0> e.class
=> Exemplo
irb(main):018:0> Exemplo.class
=> Class
irb(main):019:0> m = Exemplo.method(:new)
=> #<Method: Class#new>
irb(main):020:0> m.class
=> Method
irb(main):021:0>
```

RUBY – TIPAGEM FORTE & DINÂMICA

```
irb(main):016:0> defined? 1
=> "expression"
irb(main):017:0> defined? true
=> "true"
irb(main):018:0> defined? nil
=> "nil"
irb(main):019:0> defined? a
=> nil
irb(main):020:0> a = 1.5 + 1
=> 2.5
irb(main):021:0> defined? a
=> "local-variable"
irb(main):022:0> a.class
=> Float
irb(main):023:0> a = [1,2,3]
=> [1, 2, 3]
irb(main):024:0> a.class
=> Array
irb(main):025:0>
```

RUBY – TIPAGEM FORTE & DINÂMICA

```
irb(main):049:0> 1 + 1
=> 2
irb(main):050:0> b = 1 + 1.5
=> 2.5
irb(main):051:0> b.class
=> Float
irb(main):052:0> b + 1
=> 3.5
irb(main):053:0> b *= 2
=> 5.0
irb(main):054:0> b
=> 5.0
irb(main):055:0> b + '3'
TypeError: String can't be coerced into Float
        from (irb):55:in `+'
        from (irb):55
        from /Users/aramisf/.rbenv/versions/2.4.0/bin/irb:11:in `<main>'
irb(main):056:0> '4' + b
TypeError: no implicit conversion of Float into String
        from (irb):56:in `+'
        from (irb):56
        from /Users/aramisf/.rbenv/versions/2.4.0/bin/irb:11:in `<main>'
irb(main):057:0>
```

RUBY - AMIGÁVEL

```
irb(main):003:0> "33".to_f
=> 33.0
irb(main):004:0> "33".to_i
=> 33
irb(main):005:0> "12.34".to_i
=> 12
irb(main):006:0> 1.897.round
=> 2
irb(main):007:0> 1.89752.round(2)
=> 1.9
irb(main):008:0> 1.89752.round(3)
=> 1.898
irb(main):009:0> 1.89752.truncate(2)
=> 1.89
irb(main):010:0> a = [1,2,3]
=> [1, 2, 3]
irb(main):011:0> a.push("a")
=> [1, 2, 3, "a"]
irb(main):012:0> a.pop
=> "a"
irb(main):013:0> a
=> [1, 2, 3]
irb(main):014:0> a.unshift("unshifted")
=> ["unshifted", 1, 2, 3]
irb(main):015:0> a.shift
=> "unshifted"
irb(main):016:0> a
=> [1, 2, 3]
irb(main):017:0>
```

RUBY - DOCUMENTAÇÃO ACESSÍVEL

```
irb(main):007:0> help Hash
```

```
irb(main):028:0> help "Hash::new"
```

```
= Hash < Object

------------------------------------------------------------
= Includes:
Enumerable (from ruby core)

(from ruby core)
------------------------------------------------------------
A Hash is a dictionary-like collection of unique keys and
called associative arrays, they are similar to Arrays, but
integers as its index, a Hash allows you to use any object

Hashes enumerate their values in the order that the corres
inserted.

A Hash can be easily created by using its implicit form:

  grades = { "Jane Doe" => 10, "Jim Doe" => 6 }

Hashes allow an alternate syntax for keys that are symbols

  options = { :font_size => 10, :font_family => "Arial" }
```

RUBY <=> JAVA

- Possui coletor de lixo
- Tipagem forte de Objetos
- Métodos públicos, privados e protegidos
- Documentação Embutida

RUBY <=> JAVA

- Ruby é interpretada:
  - Não há verificação estática de tipo
  - Brinque à vontade usando o interpretador
- Variáveis são apenas rótulos, não possuem um tipo associado

RUBY – TESTES

- Test::Unit
  - Acompanha a implementação padrão da linguagem
  - Assertiva

RUBY - METAPROGRAMAÇÃO

- Programas que escrevem programas
- É um outro mundo
- Utilidade
  - DSL
  - Frameworks
  - Dinamicidade