

Introdução ao Ruby on Rails

by Jackson Pires



Entendendo e usando Gems

O que são Gems?

O que são Gems?

- Gems são bibliotecas Ruby
- Site oficial que hospeda as Gems: <https://rubygems.org/>
- Documentação: <https://guides.rubygems.org/rubygems-basics/>
- O comando `gem` é usado para interagir com as gems (instalar, remover, criar, etc...)

The background is a solid red color with a complex, abstract pattern of overlapping, semi-transparent geometric shapes, primarily triangles and polygons, creating a textured, crystalline effect.

RubyGems.org

RubyGems.org

- Pesquise gems/Explore
- Verifique a home page
- Verifique a documentação

Instalando e usando a primeira Gem

Instalando e usando a primeira Gem

Para instalar uma gem use `gem install <nome da gem>`, **ex:**

```
$> gem install cpf_utils
```


Instalando e usando a primeira Gem

Para usar uma gem após sua instalação é necessário carregá-la no projeto ou IRB. Veja o exemplo:

```
require 'cpf_utils'
```

Verifique sempre a documentação da gem para entender o que é possível fazer com ela.

https://github.com/jacksonpires/cpf_utils

Instalando e usando a primeira Gem

```
# Para gerar um número de CPF:
```

```
CpfUtils.cpf => # "45698394823"
```

```
# Para gerar um CPF formatado:
```

```
CpfUtils.cpf_formatted => # "456.983.948-23"
```

```
# Para verificar se um CPF é válido:
```

```
CpfUtils.valid_cpf?("47238051923") => # true
```

Comandos comuns

Comandos comuns

Para listar as gems instaladas: `gem list`

Para listar as gems instaladas e filtrar por nome da gem: `gem list <name>`

Para remover uma gem: `gem uninstall <gem>`

Instalando uma gem localmente:

```
gem install --local path_to_gem/filename.gem
```

Entendendo o arquivo .gem

Entendendo o arquivo .gem

O que é um arquivo .gem?

Entendendo o arquivo .gem

O que é um arquivo **.gem**?

Altere a extensão para **.zip** e veja a mágica.

Em outras palavras um arquivo .gem é o projeto ruby compactado e disponibilizado no servidor **rubygems.org**

Criando um script para verificar CPF's

Criando um script para verificar CPF's

```
$> mkdir verifica_cpfs
```

```
$> cd verifica_cpfs
```

```
$> touch app.rb
```

```
$> code .
```

Criando um script para verificar CPF's

```
#app.rb  
  
require 'cpf_utils'  
  
puts "Digite seu CPF:"  
  
cpf = gets.chomp  
  
valid_cpf = CpfUtils.valid_cpf?(cpf)  
  
valid_cpf ? puts("Seu CPF é válido") : puts("Oops, CPF  
inválido!") "
```

Criando um script para verificar CPF's

Código refatorado...

```
#app.rb
```

```
require 'cpf_utils'
```

```
puts "Digite seu CPF:"
```

```
cpf = gets.chomp
```

```
cpf.valid_cpf? ? puts("Seu CPF é válido") : puts("Oops, CPF  
inválido!)"
```

Criando um script para verificar CPF's

E se por algum motivo a pessoa que for usar o script não tiver o `cpf_utils` instalado? Tem como pré-definir quais gems são necessárias para um projeto?

Criando um script para verificar CPF's

E se por algum motivo a pessoa que for usar o script não tiver o `cpf_utils` instalado? Tem como pré-definir quais gems são necessárias para um projeto?

Sim, existem uma gem chamada `bundler` que tem esse propósito. Vamos entender como ela funciona.

Conhecendo o Bundler

Conhecendo o Bundler

- Site oficial: <https://bundler.io/>

Conhecendo o Bundler

Comece instalando o bundler

```
$> gem install bundler
```

```
$> bundler -v
```


Conhecendo o Bundler

Crie um arquivo `Gemfile` (ou use o `bundle init`) e adicione o seguinte conteúdo:

```
# Gemfile

source "https://rubygems.org"

gem "cpf_utils"
```

Conhecendo o Bundler

Para instalar as gems listadas no arquivo `Gemfile` basta rodar o comando

```
$> bundle install
```

Indicando versões específicas de Gems

Indicando versões específicas de Gems

É possível indicar versões exatas ou intervalos de versões nos Gemfile's

<https://bundler.io/guides/gemfile.html#gemfiles>

O arquivo Gemfile.lock

O arquivo Gemfile.lock

O arquivo `Gemfile.lock` "trava"/registra a versão usada no momento que o projeto foi desenvolvido, assim pode-se garantir que quem for usar o projeto em um outro momento terá a mesma versão da biblioteca que foi usada no momento do desenvolvimento bem como suas dependências, evitando assim incompatibilidade de código.

https://bundler.io/guides/using_bundler_in_applications.html#gemfilelock

Exercícios

Exercícios

- 1) Pesquise e use uma gem no IRB
- 2) Crie um pequeno script usando uma gem
- 3) O que acontece se alguém pegar um projeto que já possui um arquivo `Gemfile.lock`, apagá-lo e rodar novamente o comando `bundle install`?