

ARQUIVOS E DIRETÓRIOS

Rafael Vieira Coelho

(rafaelvc2@gmail.com)

Ruby

Aprenda a programar
na linguagem mais divertida



C Casa do
Código

LUCAS SOUZA

API (APPLICATION PROGRAMMING INTERFACE)

- Utilizaremos aqui a API do Ruby que lida com arquivos e diretórios.
- Usaremos esta API para salvarmos os objetos Livro do nosso sistema.
- Para isto, faremos serialização/deserialização de objetos

CLASSE FILE

A classe `File` da API do Ruby é uma abstração de objetos que são criados para representar um arquivo. Podemos, por exemplo, descobrir qual o tamanho, em bytes, de um arquivo salvo em disco:

```
1 arquivo_temporario = File.new("/tmp/arquivo")  
2 p arquivo_temporario.size # => 564
```


ESCREVENDO NO ARQUIVO

A API de arquivos possui uma grande variedade de métodos que utilizam blocos como forma de interação. Podemos criar um arquivo e incluir dados dentro dele com apenas uma linha de código:

```
1 File.open("/tmp/arquivo", "w") do |arquivo_temporario|  
  
2     arquivo_temporario.puts "primeira linha do meu  
arquivo"
```

Podemos usar o método write também. A única diferença é que ele não coloca a quebra de linha no final.

LENDO DO ARQUIVO

Dado que temos o arquivo texto /tmp/arquivo salvo, podemos abri-lo e imprimir cada uma de suas linhas:

```
1 arquivo_temporario = File.open "/tmp/arquivo", "r"

2 arquivo_temporario.each do |linha|

3     p linha # => "primeira linha do meu arquivo\n"

4 end
```

MODOS DE ABRIR UM ARQUIVO

O segundo parâmetro passado quando invocamos o método `open`, é o modo que desejamos abrir o arquivo. As maneiras mais comuns são:

- `r` - abre o arquivo somente para leitura
- `w` - abre o arquivo somente para escrita (sobrescreve todo o conteúdo do arquivo se o mesmo existir)
- `w+` - abre o arquivo tanto para leitura quanto para escrita (sobrescreve todo o conteúdo do arquivo se o mesmo existir)
- `a` - abre o arquivo somente para escrita (começa a escrita no final da última linha existente se o arquivo já existir)

SERIALIZAÇÃO DE OBJETOS

- Trata-se do processo de salvar um objeto no arquivo binário ou texto (xml, json, etc.).
- Desta forma, podemos recriar o objeto pronto (com os atributos preenchidos) quando lemos o arquivo.

FORMATO YAML

Vamos utilizar e explorar o formato humano de serialização que é representando pelo formato YAML (<http://ruby-doc.org/core/classes/YAML.html>). Qualquer objeto pode ser serializado para o formato YAML sem o mínimo esforço, gastando apenas algumas linhas de código:

SERIALIZANDO O OBJETO

```
1  require 'yaml'
2
3  require_relative 'livro'
4  require_relative 'biblioteca'
5  require_relative 'relatorio'
6
7  liv1 = Livro.new("Mauricio Aniche", "123454", "Suspense", 198.89)
8  liv2 = Livro.new "Tárcio Zemel", "452565", "Biografia", 53.21
9  liv3 = Livro.new("Rogério Amaral", "1111", "Suspense", 12.90)
10 liv4 = Livro.new "Alberto de Souza", "1232", "Biografia", 135.99
11 biblioteca = Biblioteca.new
12 biblioteca.adiciona(liv1, 247)
13 biblioteca.adiciona(liv2, 248)
14 biblioteca.adiciona(liv3, 249)
15 biblioteca.adiciona(liv4, 250)
16
17 biblioteca_serializado = YAML.dump(biblioteca)
18 p biblioteca_serializado
19
20 biblioteca = YAML.load(biblioteca_serializado)
21 p biblioteca
```

GRAVANDO/LENDO NO/DO ARQUIVO

```
1  require 'yaml'
2
3  class BancoDeArquivos
4
5      def salva(livro)
6          File.open('livros.yml', 'a') do |arquivo|
7              obj_serial = YAML.dump(livro)
8              arquivo.puts(obj_serial)
9              arquivo.puts ""
10         end
11     end
12
13     def carrega()
14         File.open("livros.yml", "r").map do |livro_serializado|
15             YAML.load livro_serializado
16         end
17     end
18 end
```

GRAVANDO/LENDO NO/DO ARQUIVO

```
1  require 'yaml'
2
3  require_relative 'livro'
4  require_relative 'biblioteca'
5  require_relative 'relatorio'
6  require_relative 'bd_arquivos'
7
8  liv1 = Livro.new("Mauricio Aniche", "123454", "Suspense", 198.89)
9  liv2 = Livro.new "Tárcio Zemel", "452565", "Biografia", 53.21
10 liv3 = Livro.new("Rogério Amaral", "1111", "Suspense", 12.90)
11 liv4 = Livro.new "Alberto de Souza", "1232", "Biografia", 135.99
12 biblioteca = Biblioteca.new
13 biblioteca.adiciona(liv1, 247)
14 biblioteca.adiciona(liv2, 248)
15 biblioteca.adiciona(liv3, 249)
16 biblioteca.adiciona(liv4, 250)
17
18 biblioteca_serializado = YAML.dump(biblioteca)
19 #p biblioteca_serializado
20
21 bd = BancoDeArquivos.new
22
23 bd.salva(biblioteca)
24 biblioteca = bd.carrega()
25
26 #biblioteca = YAML.load(biblioteca_serializado)
27 p biblioteca
```


EXERCÍCIO

- Acesse o link abaixo e encontre 2 exemplos de usos de métodos que não foram apresentados nos slides.
- **<https://ruby-doc.org/core-1.9.3/File.html>**