

# Linguagens de Programação

## Trabalho 2

---

Aluno: Vinícius Aguiar Figueiredo  
Professor: Miguel Elias Mitre Campista

20 de outubro de 2017

# 1 INTRODUÇÃO

O programa desenvolvido na linguagem Perl possui por objetivo fornecer ao usuário uma série de opções de validação de dados, enumerados na seção a seguir, retornando um arquivo de texto que explicita quais passaram e quais não passaram nos testes de validação.

## 2 IMPLEMENTAÇÃO DO PROGRAMA

O script principal, **main.pl**, possui 5 funções de análise principais, que usam as bibliotecas **libCPF.pm** e **libHandler.pm**, escritas para aumentar a modularização do código e facilitar seu comportamento, e também o código de interação com o usuário, que é apresentado a um menu. A primeira biblioteca contém algumas das funções usadas pela subrotina **analiseCPF** em seus testes de validação. A segunda biblioteca contém uma função de gerenciamento muito útil nas funções de análise principais, a função **arrayToFile**. A seguir enumero as funções de análise principal, seu comportamento geral e seus testes de validação de forma sucinta, uma breve documentação simplificada do script.

1. **analiseNumCel(\$filename)** - A validação de números de telefone e celular se mostrou bem complexa, por isso houve divisão da expressão regular em 3 principais, que são responsáveis pela validação do número:
  - a) `/(\+?\(\d{2}\)\d{4,5}\-?\d{4})/g`
  - b) `/(\+\d{13})/g`
  - c) `/(\d{4,5}\-?\d{4})/g`
2. **analiseCPF(\$filename, \$usarAlg)** - A validação de CPFs é a única validação que apresenta ao usuário uma outra opção além do nome do arquivo, **\$usarAlg** quando **verdadeiro** usa o algoritmo de módulo-11 do CPF para criar uma validação ainda mais rigorosa dos dados. Já a validação de formatação é mais generosa e permite valores em formatos bem distintos, utilizou-se do regex:  
`/([\d]{3}[\.\.]?[\d]{3}[\.\.]?[\d]{3}[\-]?[\d]{2})/g`
3. **analiseEmail(\$filename)** - A validação de emails se deu com uma expressão branda, capaz de autenticar emails que possuem domínios variados e/ou múltiplos, como os .poli.ufrj.br, além disso, obedecem-se algumas regras internacionais de permissão de criação de emails, usando-se a expressão:  
`/([A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{1,})/gi`  
Onde se preferiu usar o modificador de insensibilidade de caixa-alta para dar mais legibilidade à expressão.
4. **analiseCEP(\$filename)** - A validação de CEPs mostrou-se bem simples, foi escrita uma expressão que permite formatos com ou sem hífen, para abranger mais casos e validar de forma mais coerente os dados.

```
/(\d{5}-?\d{3})/g
```

5. `analisePlacaRJ($filename)` - Inicialmente a ideia era criar uma análise de RGs, como registrado no trabalho 1, entretanto descobri que não há uniformidade em RGs, sendo uma validação impossível sem acesso aos servidores com a informação dos registros armazenada. Substituí a ideia por uma análise de placas de carro do RJ, a expressão regular é simples, abrangente, e usa um conceito ainda não utilizado, do uso de `[]` na expressão, para permitir apenas placas que comecem com K ou F:

```
/([KF]\w{2}\-\d{4})/g
```

### 3 O PROGRAMA EM USO

Ao executar o script o usuário é apresentado a um menu de opções, onde se espera que seja escolhida uma das análises de validação, o programa aguarda o usuário inserir uma opção válida. Ao ser escolhida a opção, é perguntando ao usuário o nome do arquivo de texto a ser analisado. Com a conclusão da análise o programa se encerra, sem antes informar sobre a conclusão com êxito da criação de um novo arquivo, que por padrão é chamado de *modified + filename*.

```
vinicius@debian:~/lingprogscripsts/trab$ perl sketch.pl
Este é um gerenciador de validações de dados
Escolha uma das opções digitando um número:
1. Validador de Números de Celulares
2. Validador de CPFs
3. Validador de e-mails
4. Validador de CEPs
5. Validador de placas de carro do RJ
4
Opção escolhida: 4. Validador de CEPs
Digite o nome do arquivo a ser validado: ceps.txt
Análise concluída! Arquivo novo gerado!
vinicius@debian:~/lingprogscripsts/trab$ cat modifiedceps.txt
Válido      | 21020330
Válido      | 21020-330
Válido      | 26820-850
Inválido    | 28550.520
Inválido    | 35550-120a
Válido      | 45550450
Inválido    | 245504554
vinicius@debian:~/lingprogscripsts/trab$
```

Na imagem há um exemplo de execução da opção 4., também é mostrada na imagem o conteúdo do arquivo novo. O arquivo `ceps.txt` usado está incluso na pasta do trabalho.

No caso de ser chamada a opção do Validador de CPFs, o usuário é questionado sobre a utilização do algoritmo, sendo feita uma recomendação para que se use esta opção:

```

vinicius@debian:~/lingprogs/scripts/trab$ perl sketch.pl
Este é um gerenciador de validações de dados
Escolha uma das opções digitando um número:
1. Validador de Números de Celulares
2. Validador de CPFs
3. Validador de e-mails
4. Validador de CEPs
5. Validador de placas de carro do RJ
2
Opção escolhida: 2. Validador de CPFs
Digite o nome do arquivo a ser validado: cpfs.txt
Deseja usar validação por algoritmo do CPF? (Recomendado)
1 para SIM e 0 para NÃO: 1
Análise concluída! Arquivo novo gerado!
vinicius@debian:~/lingprogs/scripts/trab$ cat modifiedcpfs.txt
Válido | 15947717743
Válido | 159.47717743
Válido | 159.477.177-43
Válido | 159477177-43
Inválido | 159.477.177.43
Inválido | 159477177433
Inválido | 2588/844
Inválido | oi159478511
Inválido | a154785633
Inválido | 11111111111
Inválido | 15947717744
Inválido | 159.477.177-44
Válido | 12472226594
vinicius@debian:~/lingprogs/scripts/trab$ █

```

Note que, por mais que o penúltimo CPF obedeça à formatação, não foi validado, por não ter passado no teste de soma.

## 4 COMENTÁRIOS E CONCLUSÃO

Foram incluídos na pasta do trabalho todos os arquivos de texto usados para os testes das cinco funções. Uma pequena observação: a execução do script nas imagens está com um nome diferente do que foi dado após o término do código, a execução agora, seria evidentemente dada por *perl main.pl*. A opção de não exibir ao usuário diretamente o arquivo criado na tela é uma opção de segurança contra arquivos muito grandes, que poderiam congelar a linha de comando ou atrapalhar o usuário com excesso de informação. Ao longo do projeto a função de números de celular se mostrou bem inconsistente e às vezes pouco confiável, apesar de haver uma complexidade muito grande quando os dados são livres de restrições acredito que a expressão possa ser melhorada com mais testes. Além dessa res-

salva o programa se comporta bem e cumpre de forma consistente o seu propósito. Para análise semântica e variações das regex, usei intensivamente o site <https://regex101.com/>.