

**Projeto de Sistemas Distribuídos**  
**Cálculo dos dígitos verificadores de CPF e CNPJ**

**FEI**

Foi feito um programa em C++ que a partir de uma base de dados dada pelo professor, ele faz o cálculo dos dígitos verificadores de CPFs e CNPJs.

O Algoritmo se baseia numa função orquestradora que possui 3 passos.

1. Ler
2. Calcular
3. Mostrar e salvar resultados

1 – A leitura dos dados é feita a partir de uma quantidade de linhas pré-definidas (#define) no código. Após a leitura da linha os dados são classificados como CPF ou CNPJ baseado na quantidade de dígitos obtidos na linha lida, após a classificação os dígitos são armazenados em matrizes (Ex: m[CPF][DIGITOS]), uma para CPF e uma para CNPJ.

2 – Para fazer o cálculo dos dígitos verificadores foram criadas duas funções, uma para CPF outra para CNPJ, onde ambas recebem por parâmetros um intervalo (início e fim) do qual iram realizar o cálculo dos dígitos. Desta forma foi criado 4 Threads, utilizando a biblioteca <thread>, onde duas threads calculam os CPFs e as outras duas calculam os CNPJs lidos. Cada thread calcula metade da quantidade de CPFs ou CNPJs armazenados nas matrizes. Desta forma a quantidade de trabalho é dividida em cada thread, otimizando o algoritmo.

3 – Foi criado dois métodos para salvar e imprimir os dados no terminal, onde ambos possuem a opção de apenas mostrar o relatório no terminal e salvar os CPFs e CNPJs no arquivo de log ou mostrar todos CPFs e CNPJs calculados e o relatório no terminal.

- Um método imprime no terminal e salva os dados formatados, ou seja, o CPF seria mostrado e salvo no formato XXX.XXX.XXX-XX e o CNPJ seria mostrado e salvo no formato XX.XXX.XXX/XXXX-XX.
- O outro método imprime no terminal e salva os dados da mesma maneira que são vistos no programa (sem formatação), ou seja, os CPFs seriam uma sequência de 11 dígitos e o CNPJ seria uma sequência de 14 dígitos.

Independente da opção e do método escolhido, o programa sempre irá salvar os CPFs e CNPJs no arquivo “log.txt” que ficara no mesmo diretório do programa. Para alterar o método de formatação basta ir na função orquestradora e comentar a linha da chamada da função de print (Linha 576: Print com formatação e linha 577: Print sem formatação)

Nas figuras abaixo podemos ver dois exemplos da saída no terminal, uma do método formatado e outra do método sem formatação.

```
[~]Voce quer ver os CPFs e CNPJs no terminal ou apenas exibir o tempo das execucoes?[-]
OBS: Mesmo que escolha nao exibir os dados no terminal, sera gerado um arquivo "Log.txt" com todos CPFs e CNPJs calculados
0 - Nao
1 - Sim
0
Total de linhas lidas: 10006060
Total de CPFs: 5003030
Total de CNPJs: 5003030
0 tempo de excucao da leitura da base de dados foi de: 8369 m/s
0 tempo de excucao do calculo dos digitos na Thread halfcpf1 foi de: 85 m/s
0 tempo de excucao do calculo dos digitos na Thread halfcpf2 foi de: 85 m/s
0 tempo de excucao do calculo dos digitos na Thread halfcnpj1 foi de: 78 m/s
0 tempo de excucao do calculo dos digitos na Thread halfcnpj2 foi de: 77 m/s
0 tempo de excucao do calculo dos digitos no total foi de: 327 m/s
0 tempo de excucao da funcao de print foi de :182266 m/s

-----
Process exited after 192.4 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Figura 1 – Saída no terminal do método sem formatação.

```
[~]Voce quer ver os CPFs e CNPJs no terminal ou apenas exibir o tempo das execucoes?[-]
OBS: Mesmo que escolha nao exibir os dados no terminal, sera gerado um arquivo "Log.txt" com todos CPFs e CNPJs calculados
0 - Nao
1 - Sim
0
Total de linhas lidas: 10006060
Total de CPFs: 5003030
Total de CNPJs: 5003030
0 tempo de excucao da leitura da base de dados foi de: 7150 m/s
0 tempo de excucao do calculo dos digitos na Thread halfcpf1 foi de: 91 m/s
0 tempo de excucao do calculo dos digitos na Thread halfcpf2 foi de: 89 m/s
0 tempo de excucao do calculo dos digitos na Thread halfcnpj1 foi de: 82 m/s
0 tempo de excucao do calculo dos digitos na Thread halfcnpj2 foi de: 80 m/s
0 tempo de excucao do calculo dos digitos no total foi de: 343 m/s
0 tempo de excucao da funcao de print foi de :201760 m/s

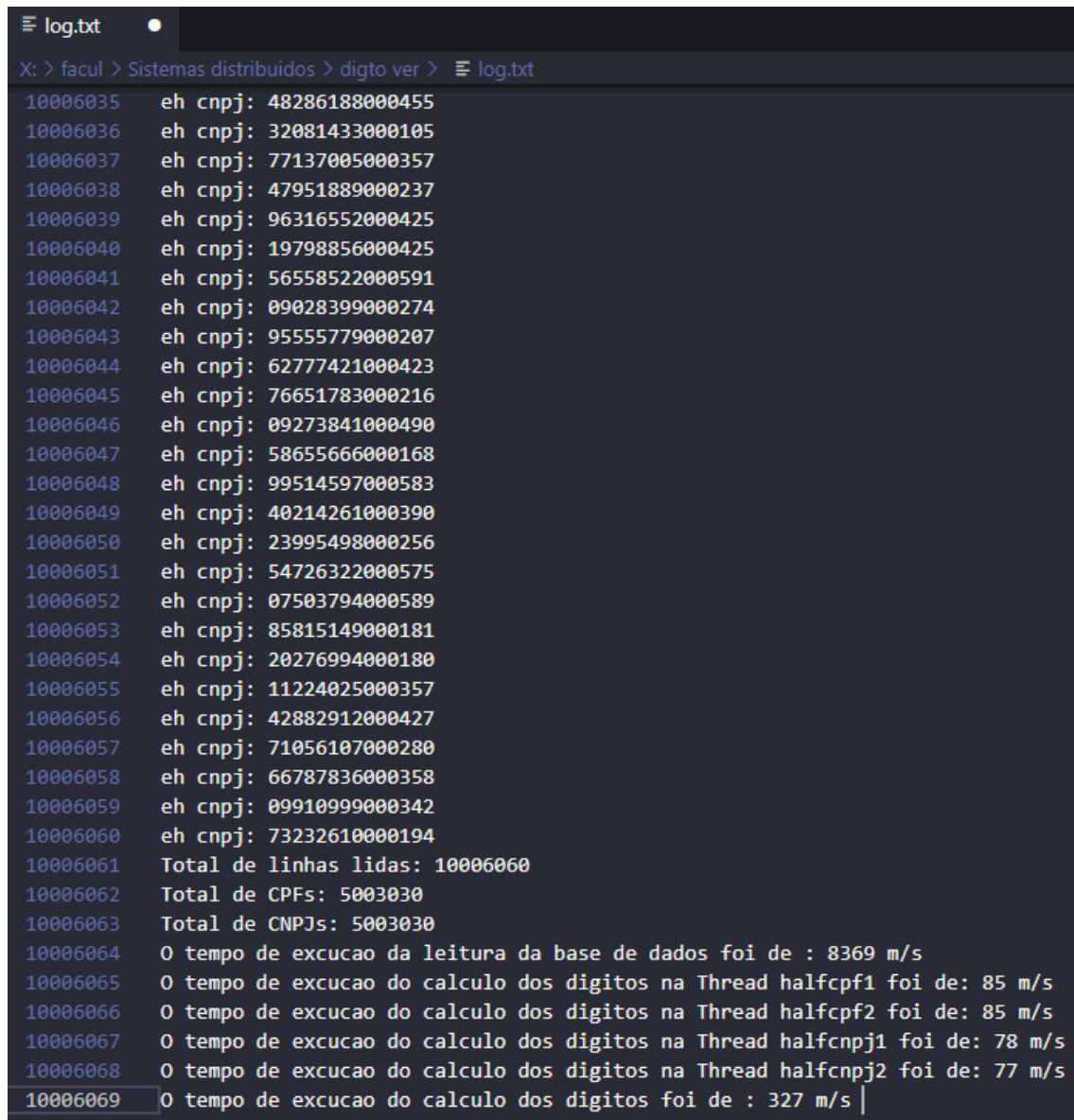
-----
Process exited after 210.7 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Figura 2 - Saída no terminal do método com formatação.

Podemos observar que o tempo da função de impressão e gravação dos dados no arquivo foi maior no método com formatação, isso se deve a lógica utilizada para imprimir os dados de forma formatada sem alterar as matrizes de dados dos CPFs e dos CNPJs.

Podemos observar nas imagens também o modelo do relatório que sempre é impresso no terminal e salvo no arquivo de log. O relatório contém o total de linhas lidas do arquivo de base de dados, o total de CPFs calculados, o total de CNPJs calculados, o tempo gasto com a leitura e armazenamento dos dados nas matrizes, o tempo que cada uma das quatro threads demorou para fazer o cálculo dos dígitos verificadores, o tempo total gasto com o cálculo, e o tempo gasto com a função de impressão e gravação dos dados no arquivo de log.

Nas figuras abaixo podemos ver dois exemplos dos dados gravados no arquivo de log, uma do método formatado e outra do método sem formatação.



```
log.txt
X: > facul > Sistemas distribuidos > dígito ver > log.txt
10006035 eh cnpj: 48286188000455
10006036 eh cnpj: 32081433000105
10006037 eh cnpj: 77137005000357
10006038 eh cnpj: 47951889000237
10006039 eh cnpj: 96316552000425
10006040 eh cnpj: 19798856000425
10006041 eh cnpj: 56558522000591
10006042 eh cnpj: 09028399000274
10006043 eh cnpj: 95555779000207
10006044 eh cnpj: 62777421000423
10006045 eh cnpj: 76651783000216
10006046 eh cnpj: 09273841000490
10006047 eh cnpj: 58655666000168
10006048 eh cnpj: 99514597000583
10006049 eh cnpj: 40214261000390
10006050 eh cnpj: 23995498000256
10006051 eh cnpj: 54726322000575
10006052 eh cnpj: 07503794000589
10006053 eh cnpj: 85815149000181
10006054 eh cnpj: 20276994000180
10006055 eh cnpj: 11224025000357
10006056 eh cnpj: 42882912000427
10006057 eh cnpj: 71056107000280
10006058 eh cnpj: 66787836000358
10006059 eh cnpj: 09910999000342
10006060 eh cnpj: 73232610000194
10006061 Total de linhas lidas: 10006060
10006062 Total de CPFs: 5003030
10006063 Total de CNPJs: 5003030
10006064 O tempo de excução da leitura da base de dados foi de : 8369 m/s
10006065 O tempo de excução do calculo dos digitos na Thread halfcpf1 foi de: 85 m/s
10006066 O tempo de excução do calculo dos digitos na Thread halfcpf2 foi de: 85 m/s
10006067 O tempo de excução do calculo dos digitos na Thread halfcnpj1 foi de: 78 m/s
10006068 O tempo de excução do calculo dos digitos na Thread halfcnpj2 foi de: 77 m/s
10006069 O tempo de excução do calculo dos digitos foi de : 327 m/s |
```

Figura 3 – Dados gravados no arquivo de log do método sem formatação

```
log.txt
X: > facul > Sistemas distribuidos > digto ver > log.txt
10006035 eh cnpj: 48.286.188/0004-55
10006036 eh cnpj: 32.081.433/0001-05
10006037 eh cnpj: 77.137.005/0003-57
10006038 eh cnpj: 47.951.889/0002-37
10006039 eh cnpj: 96.316.552/0004-25
10006040 eh cnpj: 19.798.856/0004-25
10006041 eh cnpj: 56.558.522/0005-91
10006042 eh cnpj: 09.028.399/0002-74
10006043 eh cnpj: 95.555.779/0002-07
10006044 eh cnpj: 62.777.421/0004-23
10006045 eh cnpj: 76.651.783/0002-16
10006046 eh cnpj: 09.273.841/0004-90
10006047 eh cnpj: 58.655.666/0001-68
10006048 eh cnpj: 99.514.597/0005-83
10006049 eh cnpj: 40.214.261/0003-90
10006050 eh cnpj: 23.995.498/0002-56
10006051 eh cnpj: 54.726.322/0005-75
10006052 eh cnpj: 07.503.794/0005-89
10006053 eh cnpj: 85.815.149/0001-81
10006054 eh cnpj: 20.276.994/0001-80
10006055 eh cnpj: 11.224.025/0003-57
10006056 eh cnpj: 42.882.912/0004-27
10006057 eh cnpj: 71.056.107/0002-80
10006058 eh cnpj: 66.787.836/0003-58
10006059 eh cnpj: 09.910.999/0003-42
10006060 eh cnpj: 73.232.610/0001-94
10006061 Total de linhas lidas: 10006060
10006062 Total de CPFs: 5003030
10006063 Total de CNPJs: 5003030
10006064 O tempo de excucao da leitura da base de dados foi de : 7150 m/s
10006065 O tempo de excucao do calculo dos digitos na Thread halfcpf1 foi de: 91 m/s
10006066 O tempo de excucao do calculo dos digitos na Thread halfcpf2 foi de: 89 m/s
10006067 O tempo de excucao do calculo dos digitos na Thread halfcnpj1 foi de: 82 m/s
10006068 O tempo de excucao do calculo dos digitos na Thread halfcnpj2 foi de: 80 m/s
10006069 O tempo de excucao do calculo dos digitos foi de : 343 m/s
```

Figura 4 - Dados gravados no arquivo de log do método com formatação

O código-fonte foi publicado no Github e está disponível no link:

<https://github.com/Guizith/DigValidaPFPePJ>

Um vídeo explicativo sobre o código-fonte foi publicado no Youtube e está disponível no

link: <https://www.youtube.com/watch?v=-RTmLoYgXMg&feature=youtu.be>

Link de referência da lógica usada para o cálculo dos dígitos:

<https://campuscode.com.br/conteudos/o-calculo-do-digito-verificador-do-cpf-e-do-cnpj>

Link do site utilizado para confirmação dos dígitos:

<http://ghiorzi.org/DVnew.htm>