

# Programa IT Academy - Processo Seletivo - Edição 15

Diogo Garbinato de Fagundes

March 27, 2022

## Contents

<b>1</b>	<b>Etapa</b>	<b>1</b>
1.1	Se a prova de CG for às 8:00, a prova de M deverá ser às: . . .	2
1.2	Se as provas CG e P forem respectivamente às 8:00 e 9:00, a prova de CE deve ser: . . . . .	2
1.3	Qual das seguintes afirmações é necessariamente verdadeira .	2
<b>2</b>	<b>Etapa</b>	<b>3</b>
2.1	Primeira funcionalidade . . . . .	3
2.2	Segunda funcionalidade . . . . .	4
2.3	Terceira funcionalidade . . . . .	5
2.4	Quarta funcionalidade . . . . .	6
2.5	Quinta funcionalidade . . . . .	7
<b>3</b>	<b>Autoavaliação</b>	<b>7</b>

## 1 Etapa

**Enunciado:** Para ganhar a bolsa os candidatos precisam fazer algumas provas, são elas: Conhecimento Gerais (CG), Conhecimento Específicos (CE), Português (P) e Matemática (M). Cada prova tem a duração de 50 minutos e podem ser alocadas de hora em hora. Devido ao número reduzido de fiscais:

- As provas serão num sábado, nos horários 8:00, 9:00, 10:00 e 11:00.
- A prova de (CG) deve ser às 8:00.

- A prova de (CE) deve ser após a prova de (P) e também após a prova de (CG).
- As provas de (M) e (CE) devem ser em horários consecutivos, nessa ordem.

**1.1 Se a prova de CG for às 8:00, a prova de M deverá ser às:**

- (A) 9:00
- (B) 10:00
- (C) 11:00

**Resposta:** Letra B. Pois a prova de M deve ser diretamente antes da de CE, e a prova de CE tem que ser em algum momento depois da prova de P, logo, a prova de P deve ser às 9:00, a prova M às 10:00 e a prova CE às 11:00

**1.2 Se as provas CG e P forem respectivamente às 8:00 e 9:00, a prova de CE deve ser:**

- (A) 10:00
- (B) 8:00
- (C) 11:00
- (D) 9:00

**Resposta:** Letra C. Como mencionado anteriormente, a prova de CE deve ser obrigatoriamente consecutiva a prova de M, então, como os horários 8:00 e 9:00 já foram alocados, a prova de M tem que ser às 10:00, e a prova de CE às 11:00

**1.3 Qual das seguintes afirmações é necessariamente verdadeira**

- (A) A prova P é após a prova de CE
- (B) A prova M pode ser realizada antes da prova P
- (C) A prova de CG é a primeira e a de CE é a última a ser realizada.
- (D) Se mudar a prova de CG para 10:00 então a prova de M deve ser às 8:00

**Resposta:** Letra C. Devido às regras dadas nos pontos levantados no enunciado, chegamos a uma ordem absoluta de como devem ser realizadas as provas. Começamos pela prova CG pois uma das regras é que a prova de CG deve ser às 8:00, e esse é o primeiro horário disponível, logo, é a primeira prova a ser realizada, assim como a alternativa C diz. Sabemos que a prova

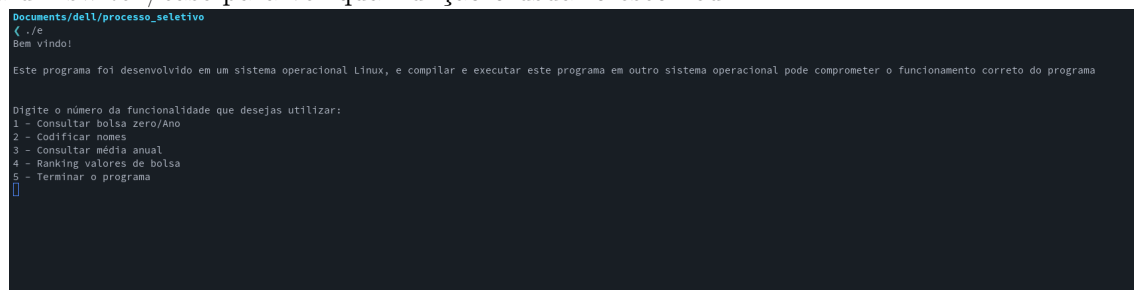
CE deve ser após a prova P e após a prova M por causa do terceiro e quarto pontos levantados no enunciado. Logo a prova de CE só pode ser a última a ser realizada, tornando a alternativa C necessariamente verdadeira.

## 2 Etapa

Na resolução da segunda etapa eu escolhi utilizar a linguagem de programação C, e o console como interface. É importante ressaltar que este programa foi desenvolvido em um sistema operacional Linux, e consequentemente tentar compilar e executar o programa em outro sistema operacional como Windows ou macOS poderá gerar resultados diferentes.

Para desenvolver este programa eu comecei criando uma variável global que irá armazenar todo o arquivo .csv em um array bi-dimensional. Assim facilitando a manipulação dos dados, e tendo que abrir o arquivo para leitura apenas uma vez. Como C não possui Strings, utilizei um array de char bi-dimensional, sendo: [linha][caracter].

O método main abre o arquivo e caso tenha sucedido em encontrar o arquivo, lê e salva cada caractere do arquivo no array, através de dois `do...while`. Depois ele entra no “menu” onde imprime na tela as opções de funções e utiliza um `switch/case` para ver qual função o usuário escolheu.



```
Documents/dell/processo_seletivo
< ./e
Bem vindo!

Este programa foi desenvolvido em um sistema operacional Linux, e compilar e executar este programa em outro sistema operacional pode comprometer o funcionamento correto do programa

Digite o número da funcionalidade que deseja utilizar:
1 - Consultar bolsa zero/Ano
2 - Codificar nomes
3 - Consultar média anual
4 - Ranking valores de bolsa
5 - Terminar o programa

```

### 2.1 Primeira funcionalidade

Para solucionar a primeira funcionalidade eu criei um método auxiliar, que será reutilizado em outras funcionalidades também, o método “*procuraAno*”, ele recebe um inteiro e um *array* de *char* por parâmetro. O inteiro é a linha do arquivo que queremos começar a procurar o ano, isso será útil na terceira funcionalidade. O *array* de *char* é o ano que o usuário quer consultar. O método retorna um inteiro, que será a linha do arquivo onde se encontra o ano, ou 0 no caso de não encontrar o ano.

Esse método usa um laço do tipo *while* que se encerra caso termine o

arquivo, ou encontre o ano que está sendo procurado. O arquivo .csv no formato de texto utiliza ';' para separar as células (ou categorias) na linha, então para encontrar o ano eu utilizei um loop que vai lendo caracter por caracter a linha até encontrar o quarto ';' (no código se vê o número 59, pois esse é o número do ';' na tabela ASCII), pois após ele, os dígitos que vem são o ano de referência. Após encontrar o ano na linha, eu utilizo um *for* que compara os caracteres do ano na linha e o ano inserido pelo usuário. Cada vez que o laço encontra um caracter igual ele adiciona em um somatório; se no final do laço o somatório for 4 significa que todos os caracteres são iguais, ou seja, é o ano procurado. Daí o programa salva a linha do arquivo onde se encontra o ano desejado e retorna o método.

Depois de encontrar (ou não) o ano desejado, se imprime os dados do “bolsista zero” através de outro método auxiliar chamado *imprimeCaracteres*. Esse método recebe dois inteiros, que seriam, a linha e o local da linha para começar a imprimir. Ele imprime na tela caracter por caracter até encontrar o próximo ';', ou o fim da linha, depois retorna o local da linha onde estava (o caractere após ';'), para assim poder continuar indo seção por seção desejada das informações solicitadas no exercício.

Os três primeiros dados são consecutivos, então eu basicamente só chamo esse método auxiliar 3 vezes seguidas. Depois precisamos imprimir o valor da bolsa, que fica 7 células a frente da última que mostramos na tela (entidade de ensino), por isso utilizei um *while* que percorre a linha até encontrar mais 6 ';', depois mostro na tela qual é a moeda, pulo o ';', e mostro o número do valor da bolsa.

```
Digite o número da funcionalidade que deseja utilizar:
1 - Consultar bolsa zero/Ano
2 - Codificar nomes
3 - Consultar média anual
4 - Ranking valores de bolsa
5 - Terminar o programa
1

~ Consultar bolsa zero/Ano ~
Digite o ano que você deseja consultar os dados do bolsista zero: 2016
NOME: ALEXANDRE RIBEIRO NETO
CPF: ***.195.647-**
ENTIDADE DE ENSINO: UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
VALOR DA BOLSA: R$ 765
```

## 2.2 Segunda funcionalidade

Para desenvolver a segunda funcionalidade do programa eu primeiro analisei o padrão de codificação que foi utilizado nos exemplos do enunciado. Cheguei a conclusão de que primeiro devemos inverter todos os caracteres, menos o primeiro e o último, depois devemos trocar a letra pela próxima do alfabeto,

e caso a letra seja Z (última do alfabeto) devemos trocar pela primeira letra, a letra A.

Eu começo o método da funcionalidade criando um array de char para armazenar o nome que o usuário deseja procurar e utilizei o *fgets* para ler o input do usuário, pois essa função lê o input até uma quebra de linha, diferentemente do *scanf*, que ao ter um espaço irá parar de ler o input. Depois criei mais um método auxiliar chamado *'procuraUauraio'*, que recebe o nome digitado por parâmetro. A lógica por trás desse método é similar a do *'procuraAno'*, a diferença é que o nome do usuário sempre começa na posição 0 do array do arquivo, assim não sendo necessário esse contador.

Depois, se o nome tiver sido encontrado no arquivo passamos para outro método auxiliar chamado *'codificaNome'*, que recebe por parâmetro a linha onde se encontra o bolsista. Primeiro eu armazeno o nome do bolsista em dois arrays de char, o primeiro é para referência, e o segundo será modificado para codificar o nome. Depois, utilizo um *for* com o tamanho do nome, e vou trocando cada letra com a sua letra espelhada (por exemplo primeira letra troca com a última letra, segunda letra com a penúltima e assim por diante).

Depois uso mais um *for*, que pega caracter por caracter e substituí pelo próximo do alfabeto (utilizando a tabela ASCII é somar 1), mas antes verificar se o caracter não é um espaço em branco ou uma letra Z, se for a letra Z substituo por A, e se for espaço em branco não faço nada. Por fim imprime na tela o nome codificado. Depois imprimo os outros dados do bolsista requeridos no exercício, utilizando a mesma lógica da primeira funcionalidade.

```

Digite o número da funcionalidade que deseja utilizar:
1 - Consultar bolsa zero/Ano
2 - Codificar nomes
3 - Consultar média anual
4 - Ranking valores de bolsa
5 - Terminar o programa
2

~ Codificar Nomes ~
Digite o nome (ou parte do nome) da pessoa deseja buscar (apenas em letras maiúsculas e sem acentos)
PEDRO
NOME CODIFICADO: QBJE TBUOBE FVRJSOFI PSEFS
ENTIDADE DE ENSINO: UNIVERSIDADE FEDERAL DE SERGIPE
ANO: 2016
VALOR DA BOLSA: R$ 1100
```

## 2.3 Terceira funcionalidade

Para a terceira funcionalidade é utilizado o método auxiliar *'procuraAno'* já explicado na primeira funcionalidade. Caso encontre o ano, entra em um loop *while* até não serem encontrados dados no ano desejado (assim o método auxiliar retornará 0). Nesse loop é utilizado um contador, que cada vez que encontra o ano, soma 1, além disso utilizo uma variável somatória, que toda

vez que é encontrado o ano, soma o valor da bolsa com o próprio somatório. Depois de terminar o loop divide o somatório total pela quantidade de vezes que o ano foi encontrado, assim obtendo a média de bolsa daquele ano.

```
Digite o número da funcionalidade que deseja utilizar:
1 - Consultar bolsa zero/Ano
2 - Codificar nomes
3 - Consultar média anual
4 - Ranking valores de bolsa
5 - Terminar o programa
3

~ Consultar média anual ~
Digite o ano que você deseja consultar: 2013
Media: R$ 935
```

## 2.4 Quarta funcionalidade

Para desenvolver a quarta funcionalidade eu criei um array bidimensional de inteiros, no tamanho 6x2, os 3 primeiros (dos 6) são os 3 bolsistas com as maiores bolsas, será armazenado a linha do bolsista e o valor de sua bolsa. Os últimos 3 são os bolsistas com as menores bolsas. Primeiro eu coloco o valor de 0 no array das maiores bolsas, assim qualquer bolsista terá uma bolsa maior, e coloco um valor absurdamente alto no das menores bolsas, assim qualquer aluno terá uma bolsa menor. Eu criei outro método auxiliar chamado '*valorBolsa*' que recebe por parâmetro a linha do bolsista que queremos saber a bolsa. O método, primeiro encontra o décimo ',' pois depois dele se encontram os números do valor da bolsa. Como não é fixo a quantidade de caracteres que a bolsa vai ter (algumas bolsas são 3 outras são 4), tive que desenvolver uma lógica para a conversão de char para int, onde não precisamos saber a quantidade de dígitos previamente. Basicamente, ele armazena o dígito em uma variável, e se não for o último dígito da linha, multiplica por 10, assim sempre tendo as casas decimais corretamente quando somarmos com o próximo dígito (por exemplo: 1500 ->  $1 * 10 = 10$ ;  $10 + 5 = 15$ ;  $15 * 10 = 150$ ;  $150 + 0 = 150$ ;  $150 * 10 = 1500$ ;  $1500 + 0 = 1500$ ). Por fim, o método retorna o valor da bolsa em inteiro.

A lógica para encontrar os alunos com maiores e menores bolsas é bem simples; ele compara a bolsa do bolsista atual com as bolsas dos bolsistas já armazenados no array 1 por 1, se for maior do que algum dos 3 primeiros troca, trocando os outros também que vem depois (caso não seja o terceiro mais baixo ou terceiro mais alto).

```

Digite o número da funcionalidade que deseja utilizar:
1 - Consultar bolsa zero/Ano
2 - Codificar nomes
3 - Consultar média anual
4 - Ranking valores de bolsa
5 - Terminar o programa
4

~ Ranking valores de bolsa ~

As pessoas com as três maiores bolsas são:
NOME: LIAMARA SCORTEGAGNA ; BOLSA: 1500
NOME: CESAR ALFREDO CARDOSO ; BOLSA: 1500
NOME: IZABEL PATRICIA MEISTER ; BOLSA: 1500

As pessoas com as três menores rendas são:
NOME: ALEXANDRE RIBEIRO NETO ; BOLSA: 765
NOME: LUIZ GUSTAVO RIBEIRO ROLANDO ; BOLSA: 765
NOME: ANTONIO RODRIGUES DE ARAUJO ; BOLSA: 765

```

## 2.5 Quinta funcionalidade

Eu fiz o menu principal um loop do tipo *while*, e o programa só encerra quando o usuário enviar o número 5, assim permitindo que ele encerre o programa.

```

Digite o número da funcionalidade que deseja utilizar:
1 - Consultar bolsa zero/Ano
2 - Codificar nomes
3 - Consultar média anual
4 - Ranking valores de bolsa
5 - Terminar o programa
5
Até mais!

Documents/dell/processo_seletivo took 2m52s
>

```

## 3 Autoavaliação

Achei o exercício, tanto a etapa 1 quanto a etapa 2 relativamente fácil e divertido, a etapa 1 consegui realizar sem nenhuma dificuldade. Já na etapa dois eu tive uma dificuldade maior para começar a desenvolver o programa, mas quando eu tive a ideia de armazenar o arquivo inteiro em um array bi-dimensional tive mais facilidade em pensar nas soluções das funcionalidades.

Tive que me esforçar bastante para desenvolver a primeira funcionalidade, mas após essa, todas as outras seguem a mesma lógica, então só tive que adaptar para cada particularidade de cada funcionalidade. Para testes, realizei testes com diversas entradas de dados, tanto dados válidos quanto inválidos e chequei se estavam de acordo com o documento .csv