

# MC102W - Algoritmos e programação de computadores

Prof. Rafael C. S. Schouery — 1s2020

## EXAME

**Entrega:** Entregue pelo Google Classroom, na tarefa intitulada “Exame”, até o dia **25/08/2020**. Você pode entregar um arquivo zip se usar mais de um arquivo `.py`.

**Objetivo:** O objetivo do exame é verificar se o aluno aprendeu os conceitos básicos da disciplina. Você deve desenvolver um programa como descrito na próxima página. Você não será avaliado apenas se o programa faz o que é pedido, mas também pelo seu domínio dos conceitos aprendidos. Em particular:

- Observe se o seu programa está bem legível, principalmente em termos de nomes de variáveis e funções. Evite repetições desnecessárias de códigos e use adequadamente comentários e *docstrings*.
- Observe o uso adequado das estruturas de dados abordadas (lista, dicionário, conjunto, tuplas, matrizes, etc...)
  - Você não é obrigado a usar todas, mas você perderá nota se usar uma estrutura quando deveria usar outra ou se o seu código ficasse muito mais simples se utilizasse umas dessas estruturas, por exemplo.
- Observe o uso adequado de Classes, Objetos e Funções.
  - Dica: Classes, em geral, têm métodos e atributos. Classes sem métodos podem ser úteis, mas as vezes podem ser substituídas por dicionários.
  - Dica: Funções ajudam muito a evitar repetição de código e simplificam bastante o entendimento, principalmente se você usar bons nomes.

**Regras:** Você deve seguir as seguintes regras ou então terá desconto na nota final:

- Você deve implementar suas próprias funções de ordenação ao invés de utilizar funções de ordenação prontas como `list.sort()` e `sorted()`.
- Você deve implementar suas próprias funções de leitura e escrita de arquivos no formato `.csv` (o formato é descrito na próxima página).
- Você deve ter pelo menos uma função recursiva no seu código.
  - Os algoritmos de ordenação recursivos apresentados no curso podem ser usados, mas não contam para o cumprimento desse critério.
- Você deverá utilizar busca binária em alguma parte do seu código.

Note que não são fornecidos exemplos de entrada e saída neste exame. Assim, você pode fazer a entrada e saída como quiser, porém você deve ser extremamente claro nas mensagens para o usuário para que o exame possa ser corrigido adequadamente.

# Sistema de Alunos

A DAC contratou você para fazer um novo sistema de gerenciamento de alunos. Apesar de um sistema desse contar com inúmeras complexidades como uso de banco de dados, interface web, emissão de documentos, etc, em um primeiro momento, você deverá implementar apenas algumas funcionalidades com o objetivo de apresentar um protótipo.

As informações do sistema serão lidas e salvas em arquivos no formato csv (*comma separated values* - valores separados por vírgula). Tais arquivos são muito populares e fáceis de lidar, funcionando basicamente como uma planilha (do Excel, Google Sheets, etc). Abaixo está um exemplo de um arquivo csv com três colunas e seis linhas:

```
RA, Nota, Disciplina
123456, 7.8, MC102
123456, 8.2, MC202
123457, 10.0, MC202
123457, 6.2, MC102
123458, 7.0, MC322
```

No exemplo acima temos três colunas chamadas RA, Nota e Disciplina e cinco *registros* (cada linha após a primeira, já que a primeira nos diz o nome da coluna). Assim, acompanhando as vírgulas e os nomes das colunas, podemos ver que o aluno de RA 123456 tirou Nota 7.8 na Disciplina MC102 e esse mesmo aluno tirou Nota 8.2 na Disciplina MC202.

Sua tabela terá mais colunas do que o exemplo acima, pois você precisará armazenar para cada registro: RA (**int**), Nota (**float**), Disciplina (**string**), Turma (**string**), Semestre (**int**, sendo que 1 representa o primeiro semestre e 2 para representa o segundo semestre), e Ano (um **int**). Além disso, os valores em si não vão conter vírgula, então não precisa se preocupar com isso.

Um *oferecimento* de uma Disciplina é uma combinação de Turma, Semestre e Ano. Por exemplo, você está matriculado em MC102, turma W do semestre 1 de 2020. Ou seja, essa combinação W, semestre 1 e 2020 determina um oferecimento de MC102. Dessa forma, conseguimos diferenciar outros oferecimentos como a turma A desse mesmo semestre e ano ou a turma W do semestre 1 de 2019.

Note que um aluno pode cursar mais de um oferecimento da mesma Disciplina.

A interface com o usuário será através do terminal, sempre apresentando menus de opções para que o usuário decida o que fazer a seguir. Abaixo listamos quais são as operações que o usuário pode fazer:

- Carregar uma tabela no formato .csv.
  - O usuário informará o caminho do arquivo.
  - Informe ao usuário se o arquivo não existir.
- Inserir um novo registro.

- Apagar um registro.
  - O usuário fornece RA, Disciplina, Turma, Semestre e Ano.
  - Se um aluno tiver duas ou mais Notas para essa mesma combinação, apague qualquer uma delas ou evite que isso aconteça na inserção de novos registros.
- Salvar os dados atuais no formato .csv.
  - O usuário informará o caminho do arquivo.
  - Peça confirmação do usuário para sobrescrever se o arquivo já existir.
- Imprimir todos os valores de RA na tabela sem repetição.
- Imprimir o histórico escolar de um RA dado.
  - Você deve mostrar as informações Disciplina, Turma, Semestre, Ano e Nota cronologicamente, separada por Semestre. Note que, como um aluno pode ter participado de vários oferecimentos de uma mesma Disciplina, você deverá mostrar todos esses oferecimentos.
- Imprimir os pares RA e Nota, ordenado por Nota, de um oferecimento, bem como a Nota média no final.
  - O usuário informa a Disciplina, Turma, Semestre e Ano.
- Imprimir, para cada RA que cursou uma Disciplina dada, qual foi o último oferecimento desta Disciplina cursada pelo RA e qual foi sua Nota.
  - Como um RA pode ter cursado uma mesma Disciplina várias vezes, queremos saber quando foi a última vez e que Nota ele tirou.
- Sair do programa.