

**Universidade Federal de Itajubá**  
**Engenharia de Controle e Automação**  
**ECA419 – Laboratório de Robótica e Manufatura**  
**Exercício 01**

Baseando-se no projeto **eca419\_exercicio01.zip** fornecido, desenvolver uma aplicação console que contenha as seguintes operações:

- 1) Inserir nova pessoa;
- 2) Inserir novo documento;
- 3) Importar dados pessoais de um arquivo .CSV;
- 4) Atualizar documento;
- 5) Atualizar dados pessoais a partir de um arquivo .CSV;
- 6) Mostrar dados pessoais de uma pessoa específica;
- 7) Mostrar dados pessoais de todas pessoas cadastradas;
- 8) Mostrar dados pessoais filtrados por idade (filtro in-range inclusive);
- 9) Exportar dados cadastrados para um arquivo .CSV;
- 10) Remover pessoa;
- 11) Sair.

Os itens 7 e 8 devem ser apresentados em ordem alfabética (pelo nome da pessoa de interesse). Implementar classe Título de Eleitor. Completar o diagrama UML do modelo, para isso, baixe o software Dia Diagram Editor (Disponível em: <http://dia-installer.de/>). Lembre-se de que somente pessoas com idade maior ou igual a 16 anos podem ter título de eleitor e somente pessoas que possuem RG, CPF e com idade maior ou igual a 18 anos podem ter CNH.

Para executar o programa:

- 1) Compile o projeto (Ctrl+B).
- 2) Abra um terminal (Ctrl+T);
- 3) Navegue até a pasta build do projeto, usando o comando cd que aprendemos em sala de aula;
- 4) Execute o executável gerado a partir da compilação do projeto (comando: **./eca419\_exercicio01**);
- 5) Teste seu programa até verificar funcionamento da sua aplicação;
- 6) Faça as devidas alteração no código para corrigir os erros encontrados no passo anterior (5);
- 7) Repita os passos 4 a 6 até que seja verificado perfeito funcionamento da sua aplicação.

Teste final:

- 1) Compile o projeto (Ctrl+B).
- 2) Abra um terminal (Ctrl+T);
- 3) Navegue até a pasta build do projeto;
- 4) Execute o comando **script eca419\_exercicio01.txt** para armazenar as saídas de texto do seu programa;
- 5) Execute o executável gerado (comando: **./eca419\_exercicio01**) a partir da compilação do projeto);
- 6) Entre as seguintes entradas na sua aplicação (respeite a ordem):

1, <seus dados pessoais>, 2, <dados do seu RG sem emissão>, 2, <dados do seu CPF>, 6, <seu nome>, 3, dados.csv, 7, 8, 50, 80, 5, atualizar\_dados.csv, 7, 8, 26, 18, 2, <dados da sua CNH, se houver>, 4, <seu RG com emissão agora>, 6, 10, Fulano de Tal, 9, dados\_atualizados.csv, 11

- 7) Entre o comando **exit** para terminar o armazenamento das saídas de texto para o script criado;
- 8) Entre o comando **exit** novamente para fechar o terminal.

Compactar as seguintes pastas e arquivos em um arquivo com o seu nome:

|   |  |
|---|--|
| <pasta do projeto>/ <b>include</b>                      | (contem os arquivos .H do projeto)                                       |
| <pasta do projeto>/ <b>src</b>                          | (contem os arquivos .CPP do projeto, inclusive a aplicação desenvolvida) |
| <pasta do projeto>/ <b>eca419_exercicio01.dia</b>       | (diagrama UML atualizado do projeto)                                     |
| <pasta do projeto>/ <b>dados_atualizados.csv</b>        | (diagrama UML atualizado do projeto)                                     |
| <pasta do projeto>/ <b>CmakeLists.txt</b>               | (contem configuração para criar projeto CMake)                           |
| <pasta build do projeto>/ <b>eca419_exercicio01.txt</b> | (que se encontra na pastas build do projeto)                             |

Prazo para entrega: **22/09/2016**

Integrantes: **até 2 integrantes.** Entretanto, cada um deve executar o teste final de acordo com os seus dados e compactar os seus arquivos, conforme relação acima. Ambos devem entregar os arquivos no mesmo e-mail.

Entrega: **eca419.unifei@gmail.com**, contendo os arquivos comprimidos de cada integrante devidamente nomeados.