### **Sistema de Cadastro em Arquivos com Ordenação**

O objetivo deste trabalho é implementar um sistema de cadastro (com recursos para busca e alteração de dados) com armazenamento dos dados em **arquivos binários.** Além do código fonte, o grupo formado por **duas a três pessoas da mesma turma prática**, deverá entregar, via campus virtual, **o código-fonte da aplicação desenvolvida, um arquivo binário de dados de exemplo e um relatório.** O relatório deve ter o mínimo de formalismo, considerando os nomes dos componentes do grupo, uma fonte de tamanho razoável (preferencialmente 12), e com o máximo de 3 páginas. O relatório deverá obrigatoriamente conter:

* + Uma breve introdução sobre o trabalho em si;
  + Uma descrição **em alto nível** (isto é, sem código) de todas as estruturas utilizadas explicando a lógica do programa;
  + Uma descrição da ordem dos dados armazenados no arquivo;
  + Abordar de forma sucinta e objetiva os acertos e erros durante o desenvolvimento do trabalho;
  + Fazer uma conclusão, apresentando os resultados obtidos.

O arquivo de dados de exemplo deverá conter ao menos 50 itens cadastrados, o ideal é ter ao menos 100. O trabalho deverá ser implementado utilizando C++ e ser passível de compilação e execução em um Linux genérico qualquer.

Vocês irão elaborar a sua própria base, devendo existir, no mínimo, **cinco informações de, no mínimo, três tipos diferentes**. Além disso, **ao menos um dos campos deverá ser uma string com espaços em seu conteúdo**. A base será informada por cada grupo no momento da inscrição do grupo (tema e campos) em um fórum criado para essa finalidade no Campus Virtual: <https://campusvirtual.ufla.br/presencial/mod/forum/view.php?id=962597>.

A título de exemplo, tem-se a seguinte base de dados (clientes.csv - Clientes da Lojinha do Bruno):

| # nome; endereço; idade; profissão; renda mensal  "João Marcos"; "Rua Salvio Salvo, 38"; 35; padeiro; 2.500,00  "Diana Maria"; "Rua Tarmonelas de Jesus, 112"; 45; costureira; 1.500,00  "Aline Peçonha"; "Av. Veludo, 113"; 38; veterinária; 4.500,00  "Samira Tinoco"; "Rua Francisco Peres, 1383"; 29; secretária; 2.000,00  "Thiago Peixoto"; "Rua Salvio Salvo, 11"; 65; aposentado; 2.600,00 |
| --- |

Percebam que há cinco campos, sendo usados: strings (com e sem espaços), inteiro e ponto flutuante.

Cada tema poderá ser utilizado por mais de um grupo, desde que os itens cadastrados sejam diferentes. Assim verifique se algum grupo já não cadastrou o tema de interesse, para não repetir os campos utilizados. Não serão aceitos temas com campos repetidos, assim, caso duas equipes proponham o mesmo tema e campos similares, ele será atribuído ao grupo que sugeriu primeiro, no fórum de discussão sobre o projeto prático no campus virtual (<https://campusvirtual.ufla.br/presencial/mod/forum/view.php?id=962597>). Os times e temas já definidos serão informados também no Campus Virtual: <https://campusvirtual.ufla.br/presencial/mod/page/view.php?id=962601>.

A equipe deve ser formada até a data limite (12/6), após essa data limite, as pessoas que não tiverem grupos serão agrupadas pelos professores de forma aleatória, dentro da mesma turma, bem como os temas com os campos e poderão ser penalizados por isso.

A aplicação desenvolvida deverá possuir uma série de características, listadas a seguir. Para fins de exemplificação, suponha que os dados a serem armazenados possuem, entre seus campos, as seguintes informações: nome, idade e endereço:

1. Os dados devem ser armazenados em um arquivo tipado, mas com capacidade de importação e exportação dos dados para o formato CSV.
2. Deve ser possível ordenar os dados por pelo menos dois dos campos do registro. No exemplo dado, a aplicação poderia dar a opção de ordenar por nome ou idade. A ordenação deverá ser feita em memória (todos os dados do arquivo são lidos em um vetor na memória, ordenados e então novamente gravados no disco). Para ordenação, deverá ser utilizado um dos métodos eficientes: quick sort, merge sort ou shell sort.
3. A aplicação deverá ser capaz de remover dados existentes diretamente no arquivo. Recomendamos que a remoção seja feita de forma lógica, ou seja: o objeto não é apagado diretamente do arquivo, mas apenas marcado (com uma chave negativa, por exemplo). Nesse caso, o espaço pode (e deve) ser utilizado em uma futura inserção. O elemento também deve ser removido totalmente quando o arquivo for novamente reordenado.
4. Deve ser possível inserir um novo elemento. Nesse caso, a inserção será realizada ou no final do arquivo, ou sobrescrevendo algum dado marcado como removido. Opcionalmente, vocês também podem optar por fazer uma inserção ordenada, para facilitar o uso de busca binária (ou seja, o elemento é inserido de forma a manter os dados ordenados).
5. Deve ser possível buscar um registro, utilizando-se um dos campos. Nesse caso, deve ser possível fazer a busca por pelo menos duas formas diferentes (no nosso exemplo: por nome ou por idade ou por endereço). A busca, que pode ser binária ou sequencial, deve ser realizada diretamente no arquivo binário. Note, que para fazer busca binária, os dados deverão estar ordenados.
6. Deve ser possível imprimir tanto o arquivo inteiro, como um trecho informado pelo usuário. Por exemplo, deve ser possível imprimir do décimo ao vigésimo elemento do arquivo. A impressão deve ser realizada na ordem em que estão armazenados. Para uma impressão ordenada, é necessário, portanto, ordenar o arquivo primeiro.

Seguem as instruções e recomendações gerais para o desenvolvimento do projeto prático:

* Os códigos devem ser implementados em C++, sem uso de "goto" no trabalho. **O comando "break" só é aceito em "switch"** (qualquer outro uso deve ser justificado antes com os professores). Também é recomendável não usar "continue". Variáveis globais que poderiam ser evitadas também reduzem grandemente o valor de seu trabalho.
* Portabilidade é uma questão chave: você deve programar lembrando-se que sua implementação será corrigida em Linux. Não é exigido interface gráfica, mas se você quiser implementá-la, faça-o de forma que o sistema final fique portável.
* Boa organização e indentação do arquivo contam pontos na nota final. Má organização e indentação ruim retiram pontos. Vocês têm liberdade para escolherem o estilo de indentação que julgarem mais adequado, mas respeitem o estilo escolhido. Recomendamos o uso de espaços para indentação, mas você pode usar tabulações, desde que não misture os dois tipos.
* É necessário que o código-fonte seja entregue com cabeçalho identificando os membros da equipe e o tema do trabalho. Além disso, variáveis, tipos e funções desenvolvidas pelo grupo devem ter nomes adequados, evitando comentários desnecessários. Trechos mais complexos do código, entretanto, precisam ser comentados adequadamente, permitindo sua melhor compreensão.

Entre os critérios que serão utilizados para avaliação do projeto prático, destacam-se, além dos itens já citados:

* boa apresentação do projeto no geral (código e relatório);
* escolhas adequadas (ex: escolha das estruturas, uso adequado de entrada e saída de dados, etc.);
* implementação de recursos não solicitados, que mostrem esforço da equipe, como uma boa interface para o usuário;
* relatório bem escrito, formal e **em formato pdf**;
* Entrega antecipada (entrega até 19/6 para Etapa 1 e 16/7 para Etapa 2 - no caso de vários envios será considerada a data do último envio).

**A entrega do projeto deve ser feita sempre através de um único arquivo compactado.** (preferência por zip, 7z, tar.gz ou tar.bz -- **\*\*não usar rar\*\***), sendo que o nome do arquivo deve conter os primeiros nomes dos alunos e o trabalho, por exemplo: joao\_pedro\_maria-projeto\_ferramentas.tar.gz. Arquivos fora do formato não serão corrigidos. A atividade deverá ser enviada em espaço específico, criado para este fim no campus virtual. Somente um membro do grupo deverá enviar o trabalho.

**Observação importante**: Utilizaremos ferramentas de detecção automática de plágios nos trabalhos enviados. Trabalhos com suspeitas de plágio serão zerados e os responsáveis poderão sofrer processo disciplinar junto à CSI-ICET.

**Prazo de entrega**: Data limite para cadastro do grupo com o tema: 12/06  
 Primeira etapa (Checkpoint): 25 de Junho de 2023

Segunda etapa: 23 de Julho de 2023

**Sugestão de Temas**

**Bandas de rock**

* Identificador (inteiro)
* Nome (string com espaços)
* Estilo musical (string simples)
* Ano de criação (inteiro curto)
* Número de Músicas famosas (inteiro curto)

**Animais exóticos**

* Identificador (inteiro)
* Nome (string com espaços)
* País de origem (string com espaços)
* Classe a que pertence (string)
* Tempo médio de vida (inteiro curto)

**Séries televisivas**

* Identificador (inteiro)
* Nome (string com espaços)
* País de origem (string com espaços)
* Ano de estréia (inteiro curto)
* Elenco principal (string com espaços)

**Constelações**

* Identificador (inteiro)
* Nome (string com espaço)
* Ano de descobrimento (inteiro curto)
* Distância da Terra (float)
* Hemisfério (char)

**Frutas tropicais**

* Identificador (inteiro)
* Nome (string com espaço)
* Região de origem (string com espaço)
* Calorias (inteiro)
* Coloração (string)

**Brinquedos antigos**

* Identificador
* Nome
* Marca
* Ano de lançamento
* Descrição

**Filmes Musicais**

* Identificador
* Nome
* Ano de lançamento
* Elenco
* Sinopse

**Games**

* Identificador
* Nome
* Ano de lançamento
* Plataforma
* Descrição

**Compositores clássicos**

* Identificador
* Nome
* Ano de nascimento
* Estilo
* Principal obra

**Astros de Hollywood**

* Identificador
* Nome
* Ano de nascimento
* Principal filme
* Prêmios recebidos

**Filósofos**

* Identificador
* Nome
* Ano de Nascimento
* Principais ideias
* Livro mais conhecido

**Músicos de Jazz**

* Identificador
* Nome
* Estilo
* Instrumento
* Banda em que toca

**Personagem de desenho animado**

* Identificador
* Nome
* Criador
* Título do desenho
* Principais relações

**Carros**

* Identificador
* Nome
* Marca
* Ano de lançamento
* Descrição

**Plantas**

* Identificador
* Nome
* Região de origem
* Nome científico
* Descrição

**Minerais**

* Identificador
* Nome
* Cor
* Propriedade eletromagnética
* Descrição

**Empresas**

* Identificador
* Nome
* Ano de fundação
* Porte
* Área de negócio

**Robô**

* Identificador
* Tipo
* Ano de criação
* Descrição
* Principal uso

**Sensores**

* Identificador
* Nome
* Tipo
* Aplicação
* Descrição

**Países**

* Identificador
* Nome
* População (quantidade)
* Idioma falado
* Descrição

**Rodovias**

* Identificador
* Nome ou código
* Extensão
* Ano de criação
* Descrição

**Telefones e smartphones**

* Identificador
* Nome ou código
* Fabricante
* Ano de criação
* Descrição

**Roupas**

* Identificador
* Tipo
* Tamanho
* Cor
* Descrição

**Universidades**

* Identificador
* Nome
* Quantidade de cursos
* Cidade (local)
* Posição em algum ranking