

## Projeto Avaliativo Final – Lista de Contatos

A avaliação final de nossa disciplina consistirá na elaboração de um programa para inserção, controle e manipulação de uma lista de contatos de Clientes. Esta lista armazenará estas informações para a empresa fictícia ACME S.A., e contará com informações tais como: número de identificação do cliente, nome do cliente, empresa onde trabalha, departamento, telefones fixo e celular e e-mail.

Seu programa deverá recuperar os dados armazenados de seções de trabalho anteriores e possibilitar a inserção de novos clientes, remoção de clientes existentes, edição dos dados de clientes e exibição da lista de clientes. Ao final de uma seção de trabalho, o programa deverá ser capaz de salvar em arquivo o conteúdo da lista de contatos, para a continuação do trabalho posteriormente em nova seção.

## Estruturas de dados do Aplicativo

Seu aplicativo deverá ser baseado na estrutura de dados do tipo Lista Dinâmica, desenvolvida em sala de aula, e contar com as funções que manipulam os desta lista **devidamente alteradas**, para que sejam pertinentes e estejam ajustadas aos requisitos do seu programa, não sendo necessário a implementação de funções que não sejam úteis.

Os nós da Lista Dinâmica necessariamente precisam ter o formato abaixo para sua utilização na Lista Dinâmica:

Estrutura Cliente (struct)	Ponteiro prox.
Código Nome Empresa Departamento Telefone Celular e-mail	Ponteiro para a próxima estrutura

A ordenação dos elementos na lista deverá ser crescente e baseada no atributo (campo) código, portanto, só há a necessidade de contar com as funções de inserção e remoção ordenadas. A implementação da Lista Dinâmica, é de escolha livre entre os três tipos semelhantes vistos em sala de aula: Lista Dinâmica Ligada, Lista Dinâmica Duplamente Ligada ou Lista Dinâmica Circular.

Para o caso de se utilizar a Lista Dinâmica Duplamente ligada, a estrutura acima diferirá um pouco: é necessária a inclusão de mais um ponteiro que apontará para o elemento anterior.

Seu aplicativo deverá ser capaz de ler e gravar em um arquivo, **utilizando o modo gravação binária**, estruturas completas com todos os dados dos clientes.

## Funcionalidades do Aplicativo

Todas as funcionalidades do seu programa deverão estar implementadas em funções específicas que realizarão uma funcionalidade específica. Por exemplo, uma função para realizar a leitura/armazenamento dos dados em arquivo, uma função para impressão em tela, uma função para edição, etc. Modularize seu programa.

O aplicativo deverá ser implementado com um menu que terá suas escolhas acessíveis por números, ou seja, um menu com as opções numeradas onde o usuário poderá escolher a funcionalidade que desejar. Este menu deverá ser exibido **ininterruptamente** e, após a execução de cada funcionalidade, seu programa deverá apresentá-lo novamente. O menu deverá contar ainda com uma opção específica para encerramento da aplicação, lembrando que neste ponto deve-se tomar o cuidado com a **perda de dados**.

O menu terá que contar no **mínimo**, com as seguintes opções:

- Inserção de novo contato;
- Gerar e exibir relatório de contatos em forma de lista total
- Gerar e exibir relatório individual com busca por identificador;
- Gerar e exibir relatório com busca por nome (neste caso exibir todos com o mesmo nome);
- Edição de dados do contato, escolhido por identificador;
- Remover contato, escolhido por identificador;
- Sair e encerrar o programa, porém, salvando toda a lista para posterior nova seção de trabalho, antes de seu encerramento.

## Opções do menu

### Inserção de novo contato

Esta opção do menu tratará da inserção de novos clientes na aplicação. Seus dados serão armazenados em estrutura específica, que no momento da inserção na Lista será atribuída a parte de dados da estrutura principal (nó), que fará parte da Lista Dinâmica.

### Geração de relatório – Lista Total

A aplicação deverá contar com uma listagem em tela, na forma de relatório, de todos os clientes armazenados na Lista Dinâmica. O objetivo desta funcionalidade além de visualizar todos os clientes, está no fato de que através deste relatório, será possível a visualização dos códigos de identificação dos clientes a serem editados ou excluídos. Estes códigos então serão utilizados nestas outras funcionalidades.

### Geração de relatório individual – busca por identificador

Este relatório funciona de forma individual para confirmação de dados dos clientes. Entra-se com sua identificação e a aplicação exibe todos os seus dados para verificação.

### Geração de relatório – busca por nome

De forma similar ao anterior, este relatório busca os clientes por nome, sendo também mais uma forma de identificação dos clientes na base de dados. Vale ressaltar que nesta função é possível encontrar mais de um cliente com o mesmo nome. Neste caso, todos com o mesmo nome devem ser exibidos. Um detalhe deve ser observado: A linguagem C é **Case Sensitive**, tome cuidado com as buscas.

### Edição de dados do contato – busca por identificador

Esta funcionalidade permite a busca, exibição e posterior edição de um cliente. Ela deve ser disparada pelo fornecimento do código de identificação do cliente.

Após o cliente ser encontrado na lista dinâmica onde está armazenado, seus dados devem ser exibidos para confirmação juntamente com a mensagem se o usuário realmente deseja editar seus dados. A edição em caso afirmativo, deve acontecer enquanto a **aplicação ainda exibe os dados antigos** por uma questão de facilidade de edição.

Em caso negativo para a edição, a aplicação deverá retornar ao menu principal.

### Remoção de contato – busca por identificador

O funcionamento desta opção do menu é similar ao anterior. A aplicação executará uma busca pelo código de identificação do cliente armazenado na lista dinâmica, e o exibirá e em tela. Em seguida exibe mensagem confirmando se realmente o usuário deseja remove-lo.

### Sair do programa – encerramento com segurança

Esta opção encerra a aplicação, simplesmente sai do programa. Mas lembre-se deve haver aqui, antes do encerramento do aplicativo, uma rotina a ser executada para a gravação dos dados presentes na Lista Dinâmica no arquivo de dados, de forma que não haja perda de informações.

## Regras:

- Implementar o código utilizando somente Linguagem C;
- Utilizar estruturas de dados do tipo Lista Dinâmica (3 opções) – vetores convencionais para armazenamento dos dados dos clientes e a não utilização de alocações de memória, **invalidam** o projeto;
- A lista de contatos, quando entregue para avaliação, deverá possuir no mínimo 10 registros de clientes previamente gravados em seu arquivo de dados;
- O arquivo utilizado para salvar os clientes deve ser entregue juntamente com o projeto, com clientes armazenados, obviamente;
- Extremamente importante informar, com comentários, **o que e como** as rotinas (funções) funcionam (lembrem-se: eu não conheço o seu raciocínio, portanto comentem!!!), **vale pontos**;
- **Projetos copiados (e o Trabalho que serviu de fonte) terão nota 0 – Lembre-se: todos os projetos de semestres anteriores estão arquivados**;
- O projeto poderá ser realizado em dupla ou de forma individual.
- Use o bom senso, não complique! Implemente uma funcionalidade por vez, facilitando assim a correção de possíveis erros durante o seu desenvolvimento.

## O Trabalho deve ser entregue com:

- **Todos os arquivos do projeto** (executáveis, fontes, bibliotecas, arquivos gerados pela IDE CodeBlocks e módulos de funções), todos bem indentados e comentados quando for o caso, em data especificada na plataforma AVA/Moodle;
- Um relatório deverá ser entregue junto com o projeto em formato PDF e conter:
  - Decisões de implementação tomadas, uma visão geral do funcionamento do programa, comentários sobre os testes executados, etc.
  - Manual de utilização do programa;
  - Resultado dos testes, quais foram os problemas, como foram resolvidos, etc.
  - A falta deste relatório acarretará penalização de 2,00 pontos na nota final do projeto.

**Não há a possibilidade de alteração da data de entrega.**