

Lista Avaliativa 2

Este trabalho deve ser realizado individualmente e enviado pelo portal didático como um arquivo compactado contendo o código-fonte (arquivo .c) do programa. O nome do arquivo enviado deve ser formado pelo nome completo do aluno usando underline (_) entre palavras e sem acentuação. Por exemplo, João Souza Silva fez o trabalho, então o nome do arquivo é joao_souza_silva.zip. A data de entrega da lista é **02/12/2020**. A lista vale 25 pontos.

O trabalho da lista 2 consiste em **modificar** e **estender** o programa que foi desenvolvido na lista 1. O sistema desenvolvido por vocês deve agora suportar **o cadastro e a visualização de várias instâncias das três entidades do sistema. O sistema deve armazenar todas as instâncias criadas para as entidades**. Por exemplo, se o seu sistema tem as seguintes entidades Cliente, Produto e Venda, em um determinado instante de uso do sistema poderemos ter 212 clientes, 590 produtos e 987 vendas cadastradas. Então, seriam 212 instâncias de Cliente, 590 instâncias de Produto e 987 instâncias de Venda. Lembre-se que todas essas instâncias devem estar armazenadas na memória durante a execução do programa. Obs: Considere que o número máximo de instâncias/ocorrências para cada entidade será igual a 1000.

Deve ser definida uma struct para cada entidade e os atributos da entidade devem ser armazenados como campos da struct.

Além das funções de cadastro e visualização/impressão dos dados das entidades, devem ser implementadas as seguintes funções:

- Uma função para **procurar uma entidade** dado um dos atributos dela. Por exemplo, no caso do sistema acima, deve ser implementada uma função para procurar um Cliente a partir do nome dele (**o atributo usado na busca será escolhido por vocês**) e também funções para procurar um Produto e uma Venda baseado nos atributos escolhidos. As funções de busca devem **imprimir os dados da entidade e retornar a entidade** caso ela seja encontrada ou retornar uma struct vazia (inicializada com dados inválidos, por exemplo) caso não tenha sido encontrada.
- Uma função para **remover uma entidade** do sistema dado um dos atributos dela (o atributo pode ser escolhido por vocês). Após a remoção, os dados da entidade removida não estarão mais disponíveis. Por exemplo, remover um Cliente dado o nome dele.

Essas funções (cadastro, impressão, procura e remoção) deverão estar disponíveis nos submenus de cada entidade.

O sistema deve estar organizado e as **funcionalidades** dele devem ser separadas em **funções**. Será avaliada a organização do código e como as funções foram definidas e estão sendo chamadas/usadas.

Obs: Para você conseguir fazer a lista 2 você deverá arrumar todos os erros que existirem na sua implementação da lista 1 e implementar tudo aquilo que não havia sido implementado.