

### TRABALHO 3 - LISTAS ENCADEADAS

Implemente um sistema em C para controle de locação de veículos a clientes. O sistema deve atender aos seguintes requisitos:

- Deve ser possível cadastrar **veículos**: incluir veículos e listar todos os veículos.
  - Sobre cada veículo, guardar: placa, marca, modelo, ano de fabricação, quilometragem atual, valor da diária, e uma flag “disponibilidade” que informa se o veículo está disponível para locação (um veículo está disponível para locação se ele não está locado no dia atual).
- Deve ser possível cadastrar **clientes**: incluir clientes e listar todos os clientes.
  - Sobre cada cliente, guardar: CNH, nome e telefone.
- Deve ser possível **realizar a locação de veículos por clientes**: locar veículo (retirar), devolver veículo e listar todas as locações.
  - Sobre cada locação, guardar: cliente, veículo, dia da retirada, dia da devolução e valor total pago.

Restrições:

- 1) Um veículo só pode ser locado se estiver disponível na data solicitada.
- 2) A quilometragem atual do veículo deve ser solicitada pelo programa e atualizada pelo usuário sempre que o veículo for devolvido.
- 3) A flag “disponibilidade” deve ser atualizada automaticamente pelo programa sempre que o veículo for locado (*indisponível*) e devolvido (*disponível*).
- 4) O valor total pago deve ser calculado automaticamente pelo programa, com base no valor da diária e nos dias locados.
- 5) Uma locação só pode ser feita por um cliente já cadastrado e um veículo já cadastrado;
- 6) Não pode haver duplicidade de informações sobre clientes e veículos. Por exemplo, uma locação não armazena/replica os dados de clientes e veículos, mas guarda apenas ponteiros que apontam para os respectivos cliente e veículo já previamente cadastrados.

Faça um menu de opções que permita ao usuário realizar as operações descritas acima e gerar os seguintes relatórios:

- R1) Listar todas as locações ativas (clientes ainda não devolveram os carros) em uma certa data (a data é informada pelo usuário);
- R2) Listar todas as locações já realizadas por um cliente (a CNH do cliente é informada pelo usuário);
- R3) Listar o faturamento da locadora por mês (mês é informado pelo usuário). Por faturamento, entenda o total arrecadado em locações.
- R4) Listar a placa dos 3 veículos mais rodados (quilometragem mais alta).
- R5) Listar a placa e o modelo de todos os veículos disponíveis (não estão locados).

Sugere-se que o menu principal seja estruturado conforme abaixo:

1. incluir veículos
2. listar veículos
3. incluir clientes
4. listar clientes
5. realizar locação
6. devolver veículo
7. listar locações

### Observações para este trabalho:

- **data da entrega: 4 de outubro;**
- para manipulação de datas e diferença entre datas, olhar/utilizar código pronto disponibilizado no site;
- não colocar restrições de formatação (máscaras) para as entradas de dados. São ótimas em sistemas reais, mas dificultam para eu fornecer a entrada de dados para corrigir (uso entrada de dados padronizada para todos os alunos);
- evitem limpeza de telas após entradas de dados, pois dificultam para eu acompanhar as entradas já fornecidas e para conferir os resultados esperados;
- os dados devem ser armazenados **obrigatoriamente em listas encadeadas**;
- podem usar listas simplesmente ou duplamente encadeadas;
- deve ter um menu de opções que permita ao usuário escolher qual ação deseja executar;
- as *structs* devem sempre ser alocadas dinamicamente;
- **o código deve ser organizado em diferentes arquivos e funções (sugestão: main.c, lista.c, lista.h, menu.c, menu.h);**
- trabalhos atrasados não serão corrigidos - se você não terminou tudo a tempo, entregue o que conseguiu fazer;
- formato da entrega: enviar os arquivos fonte .c e .h (não anexar arquivos executáveis!) para deise@inf.ufsm.br;
- trabalho em grupos de até 2 alunos. Enviar um e-mail por dupla, identificando a disciplina, nome dos alunos e número do trabalho (trabalho 3, por exemplo);
- caso não receba confirmação do recebimento do e-mail em 24hs, contate novamente o professor (reclamações posteriores sobre possíveis problemas no envio do e-mail não serão aceitas);
- cópias da internet e/ou colegas de outras duplas anulam a nota do trabalho de ambos os grupos;
- trabalhos com erros de compilação não serão corrigidos;
- professor usará o Dev-C ou Code Blocks para corrigir os trabalhos. Podem usar qualquer IDE para implementar o trabalho, mas certifiquem-se de que os arquivos estão compilando corretamente em alguma destas IDEs;
- **obrigatoriamente a solução de vocês deve fazer uso de funções e passagem de parâmetros** (eventuais respostas que pareçam corretas na execução, mas não usem funções e passagem de parâmetros, serão consideradas erradas);
- **não devem ser usadas variáveis globais;**
- a qualquer momento, vocês podem ser chamados para responder questionamentos sobre o trabalho entregue. Quando chamados, os alunos devem apresentar/responder aos questionamentos do professor, os quais farão parte da nota do trabalho;
- juntamente com o trabalho deve ser entregue um arquivo readme.txt que descreve sucintamente o que foi e o que não foi feito do enunciado do trabalho;
- peso do trabalho: verificar no site da disciplina em “Avaliação”.