

## Programação Estruturada

**Professoras: Andréa Braga e Talita Berbel**

### **Projeto Final**

### **Sistema de Estoque de uma Montadora de Veículos**

---

#### **Informações importantes:**

- O projeto é composto por: 2 etapas parciais (20%) + entrega final (80%).
- Para cada entrega parcial é necessário enviar **até o prazo de 1 semana**, o arquivo fonte atual (formato .c ou .cpp) do projeto via CANVAS.
- O código fonte das entregas deverá conter **cabeçalho com nº do grupo, RA e nome dos integrantes do grupo (ordem alfabética)**.
- **Manter o nome e tipo das variáveis e os membros das estruturas.**
- Utilizar: **alocação dinâmica, ponteiros para memória e ponteiros para arquivos.**
- A entrega final será feita até no máximo dia **29/11 (turmas - manhã)** e **30/11 (turmas - noite)** via CANVAS.
- Projetos cujas cópias/colas forem identificadas não serão pontuados, portanto, a nota referente à Avaliação Final será 0 (**zero**).

## 1ª etapa - Fazer menu e rotina para:

- Arquivo: **concessionaria.bin** e Cadastro de no máximo 5 (cinco) concessionárias que acessam o sistema, contendo a estrutura:
  - Nº do registro (gerado automaticamente pelo sistema),
  - nome,
  - CNPJ,
  - endereço - é composto por uma estrutura com: logradouro, bairro, CEP, cidade, estado, fone e email,
  - quantidade de carros vendidos,
  - quantidade de carros reservados,
  - tabela 0 (irá conter o status e o nº de registro do 1º carro, se for o caso),
  - tabela 1 (irá conter o status e o nº de registro do 2º carro, se for o caso),
  - tabela 2 (irá conter o status e o nº de registro do 3º carro, se for o caso).
    - ❖ as quantidades de carros vendidos e reservados devem ser **0** inicialmente
    - ❖ as tabelas devem ser **união (OR)**
    - ❖ as tabelas devem ser inicializadas com “L” – livre

### Exemplo do Cadastro de Concessionárias

regLoja	Nome	CNPJ	Endereço	Sold	Reserved	tabela 0	tabela 1	tabela 2
1	Abrao Reze	xx.xxx.xxx/xxxx-xx	xxxxxxxxxx	0	0	L	L	L
2	Cobel	11.444.777/0001-61		0	1	R - 3	L	L
3	Automec	xx.xxx.xxx/xxxx-xx		0	0	L	L	L
4	Caiuás	xx.xxx.xxx/xxxx-xx		0	0	L	L	L
5	SAF	xx.xxx.xxx/xxxx-xx		0	0	L	L	L

- Consulta da situação da concessionária:
  - mostrar o nome e a quantidade de carros vendidos e reservados (se for o caso – mostrar as tabelas).
  - o acesso deve ser feito através do **CNPJ**

### Exemplo do Consulta de Concessionárias

nome	Sold	Reserved	tabela 0
Automec	0	1	R – 3

```
struct endereco    {
    char    logradouro[80];
    char    bairro[15];
    char    CEP[10];
    char    cidade[15];
    char    estado[3];
    char    fone[12];
    char    email[40];
};

struct infocarro    {
    char    sigla;
    int     regcarro;
};

union dadoscarro    {
    char    sigla;
    struct infocarro reservado;
};

struct loja    {
    int     regloja; //gerar automático
    char    nome[30];
    char    CNPJ[19];
    struct endereco end;
    int     sold;
    int     reserved;
    union   dadoscarro  tabela[3];
};
```

- Arquivo: **carro.bin** e Cadastro de no máximo 50 (cinquenta) carros da montadora contendo a estrutura:
  - Nº do registro (gerado automaticamente pelo sistema),
  - modelo,
  - cor,
  - preço e
  - status que poderá ser:
    - L – livre
    - R – reservado + CNPJ
  - status irá conter a sigla de status e/ou o CNPJ da loja que reservou o carro, se for o caso
    - ❖ status deve ser **união (OR)**
    - ❖ o status deve ser inicializado com “L” – livre

**Exemplo do Cadastro de Carros**

regCarro	Modelo	Cor	Valor	Status
1	Gol	Vermelho	51000	L
2	FIT	Preto	60000	L
3	Cruze	Prata	98000	R - 11.444.777/0001-61
4	Gol	Cinza	43000	L

- Consulta de carros da montadora – dois tipos de consulta:
  - **Total**
    - ❖ mostrar nº de registro, modelo, cor, valor e status dos carros

**Exemplo da Consulta de Carros - Total**

regCarro	Modelo	Cor	Valor	Status
1	Gol	Vermelho	51000	L
2	FIT	Preto	60000	L
3	Cruze	Prata	98000	R - 11.444.777/0001-61
4	Gol	Cinza	43000	L

- **Parcial - Status**
  - ❖ a escolha do **status (L ou R)**, deve ser feita pelo usuário
  - ❖ mostrar nº do registro, modelo, cor e valor.

**Exemplo da Consulta de Carros – Parcial (Status: L)**

regCarro	Modelo	Cor	Valor	Status
1	Gol	vermelho	51000	L
2	FIT	preto	60000	L
4	Gol	cinza	43000	L

○ **Parcial - Modelo**

- ❖ a escolha do **modelo**, deve ser feita pelo usuário
- ❖ mostrar nº do registro, modelo, cor, valor e status.

**Exemplo da Consulta de Carros – Parcial (Modelo: Gol)**

regCarro	Modelo	cor	Valor	Status
1	Gol	vermelho	51000	L
4	Gol	cinza	43000	L

```
struct infoloja    {
    char    sigla;
    char    CNPJ[19];
};

union dadosloja    {
    char    sigla;
    struct infoloja    reserva;
};

struct montadora    {
    int    regcarro; //gerar automático
    char    modelo[20];
    char    cor[10];
    float    valor;
    union dadosloja    status;
};
```

## 2ª etapa – Fazer menu e rotina para:

- Início da reserva:
  - a concessionária só poderá reservar 3 carros
    - ❖ verificar se **reserved**  $\leq 3$ , caso positivo, alterar:
      - **cadastro de concessionárias:**
        - incrementar **reserved**
        - tabela: **R** e cadastrar nº do registro do carro
  - a consulta dos carros disponíveis é opcional, portanto deve-se verificar se a consulta parcial deverá ser realizada ou não.
  - a reserva deve ser feita através do **nº do registro do carro**
    - ❖ verificar se status **L (livre)**, caso positivo, alterar:
      - **cadastro de carros** – status: **R** e cadastrar CNPJ.
  - efetivar as alterações no arquivo binário.
  
- Término da reserva:
  - o término da reserva deve ser feito através da **nº do registro do carro**
  - mostrar modelo, cor, valor e status.
    - ❖ verificar se status **R (reservado)**, caso positivo:
    - ❖ verificar se o carro será comprado ou liberado da reserva
      - **cadastro de carros:**
        - **comprado:** modelo e cor = **vago**, valor = **0** e status: **L**.
        - **liberado:** status: **L**
      - **cadastro de concessionárias:**
        - decrementar **reserved**
        - incrementar **sold** – caso carro vendido
        - tabela: **L**
  - efetivar as alterações no arquivo binário.

### 3ª etapa – Fazer menu e rotina para:

- Histórico de Vendas
  - No término da reserva, caso o carro seja vendido para a concessionária deverá ser gerado um histórico da venda, com os dados descritos na estrutura abaixo.
  - No término da reserva deve-se digitar a data da venda (dia, mês e ano).
  - Já os dados da concessionária e do carro são originados dos arquivos binários já existentes.
  - Nome do arquivo: **historico.bin**

#### Exemplo do Histórico

regHist	regCarro	Modelo	Cor	Valor	regLoja	Nome	CNPJ	dataVenda
1	7	Gol	Preto	52000	1	Cobel	11.444.777/0001-61	30/01/2019
2	10	Onix	Prata	42000	2	Automec	xx.xxx.xxx/xxxx-xx	02/05/2019

```
struct data      {
    int dia;
    int mes;
    int ano;
};

struct historicoVendas {
    int reghist;
    int regcarro;
    char modelo[20];
    char cor[10];
    float valor;
    int regloja;
    char nome[30];
    char cnpj[19];
    struct data dataVenda;
};
```

- Consulta de Histórico de Vendas – mostrar todas as informações referentes as seguintes consultas:
  - Carros comprados por concessionária: recebe o **CNPJ** da concessionária;
  - Concessionárias que compraram determinado modelo de carro: recebe o **modelo** do carro.
- Alteração de cadastro de Concessionárias:
  - o acesso deve ser feito através do **CNPJ**

- mostrar regLoja, nome, endereço, quantidade de carros vendidos, quantidade de carros reservados e tabelas.
  - permite alterar **nome** e/ou **CNPJ** da concessionária
- Alteração de cadastro de Carros:
  - o acesso deve ser feito através do **regCarro**
  - mostrar modelo, cor, valor e status
  - permite alterar **modelo**, **cor** ou **valor** do carro