

Loja de Automóveis

Lucas do Vale Santos

02 de Novembro de 2013

1. Introdução

Este trabalho consiste na elaboração de um programa que simule uma loja de automóveis, permitindo algumas operações básicas como armazenar dados de veículos, alterar preços e consultar os dados, através da manipulação de arquivos. Espera-se que através deste programa, seja aplicada e compreendida a noção de “persistência”.

2. Implementação do Programa

O programa desenvolvido foi escrito utilizando a linguagem de programação **Pascal** e sua biblioteca **Crt**, que é de fundamental importância para o funcionamento dos comandos **writeln** e **readln** (entrada e saída de dados), além do **clrsrc** que serve para apagar os dados impressos na tela que antecedeam este mesmo comando.

Outra biblioteca utilizada foi a **Sysutils**, para o uso do comando **deletefile** que permite apagar um arquivo.

Dentre os vários recursos que a linguagem proporciona, além dos citados anteriormente, tem-se o uso de estruturas de iteração indefinida (**do ... while**), estruturas condicionais (**if ... then ... else, case ... of**) e procedimentos(**procedure**). Outro recurso usado foi a criação de **types**, e foi feito também, o uso da passagem de parâmetros por referência.

Neste programa, foram utilizadas estruturas de dados homogêneas (**array**), cuja aplicação foi de essencial importância para guardas os diversos **registros**.

O programa utiliza também, a noção de **persistência** que consiste em manipular arquivos. Estes permitem guardar os dados trabalhados no programa na memória secundária, de maneira que não se perca os mesmos.

Não foi encontrado nenhum obstáculo para a implementação do programa. Algumas restrições foram aplicadas, como serão mostradas a seguir.

3. Casos de Uso

Aqui serão apresentados os casos de usos referentes a entrada, saída e o tratamentos de alguns dados.

O programa disponibiliza na tela de execução, um **menu** fixo, contendo opções e suas respectivas finalidades, como pode ser visto na Figura 1.

```
C:\Users\libs\Documents\LUCAS DO VALE\Trabalho3.exe
***** Loja de Automoveis *****
1 - Incluir carros;
2 - Editar preco de um carro;
3 - Consultar dados de um carro;
4 - Listar todos os carros de um fabricante;
5 - Listar todos os dados;
6 - Apagar arquivo existente;
7 - Sair

Digite uma opcao:
```

Figura 1: menu do programa

No programa principal temos a estrutura **case of** que permite alterar o fluxo do programa para uma operação que corresponde a uma opção que o usuário desejar. Veja a Figura 2.

```
663 begin
664     menu;
665
666     writeln;
667     writeln('Digite uma opcao: ');
668     readln(op);
669
670     clrscr;
671     menu;
672
673     while(op <> 7)do
674     begin
675     case op of
676     1: begin
677         inclusaoCarros(vetCarros,numeroCarros); //incluir nov
678     end;
679     2: begin
680         edicaoPreco; //edita o preco de um carro
681     end;
682     3: begin
683         consultaDados; //consulta os dados de um carro
684     end;
685     end;
686     end;
687     end;
```

Figura 2: uma visão parcial do programa principal

A primeira opção disponível permite que o usuário inclua novos carros e seus respectivos dados. Ao escolher essa opção, o procedimento **inclusaoCarros()**; é chamado, e dentro deste procedimento é onde ocorre a inclusão.

A princípio, a rotina padrão de tratamento de erros é desativada, para que seja possível verificar se o arquivo contém dados. Se a variável **IORESULT** retornar o valor 2, isto significa que ele deve abrir pela primeira vez o arquivo para leitura e escrita de dados, de maneira que possa inserir registros no seu início. Caso contrário, significa que já existem dados dentro do arquivo, e o programa simplesmente carrega esses dados. Observe a Figura 3.

Em seguida, os dados são copiados do arquivo para um **array de registros** de maneira que sejam manipulados na memória principal, como também mostra a Figura 3.

```

130 begin
131
132     continua := 1;
133
134     assign(catalogo,'catalogo.bin');
135     {$I-}
136     reset(catalogo);
137     {$I+}
138     if(IORESULT = 2)then //pela primeira que estiver sendo cr
139     begin
140         rewrite(catalogo);
141         write(catalogo,reg);
142         seek(catalogo,filepos(catalogo)-1);
143         numeroCarros := 0;
144     end
145
146     else //carrega o arquivo e seus respectivos dados existen
147     begin
148         seek(catalogo,filesize(catalogo));
149         numeroCarros := filesize(catalogo);
150         reset(catalogo);
151         i := 1;
152         while not eof(catalogo) do
153         begin
154
155             read(catalogo,reg);
156             vetCarros[i].modelo := reg.modelo;
157             vetCarros[i].placa := reg.placa;
158             vetCarros[i].fabricante := reg.fabricante;
159             vetCarros[i].ano := reg.ano;
160             vetCarros[i].preco := reg.preco;
161             i := i + 1;
162
163         end;
164
165     end;
166
167     close(catalogo);

```

Figura 3: manipulação dos dados contidos no arquivo

Feito isso, é solicitado ao usuário que entre os dados dos novos veículos, como mostra a Figura 4. Ainda no procedimento **inclusaoCarros()**; um outro procedimento é chamado para que faça a ordenação dos carros por ordem alfabética. Observe a Figura 5.

```

169 repeat //permanece aqui até que o usuário deseje ou até que o limite de registros
170
171     writeln;
172     writeln('Digite o nome do modelo: ');
173     readln(modelo);
174     writeln;
175     writeln('Digite a placa: ');
176     readln(placa);
177     writeln;
178     writeln('Digite o nome do fabricante: ');
179     readln(fabricante);
180     writeln;
181     writeln('Digite o ano de fabricacao: ');
182     readln(ano);
183     writeln;
184     writeln('Digite o preco: ');
185     readln(preco);
186     writeln;
187
188     incluiOrdenado(vetCarros,numeroCarros,modelo,placa,fabricante,ano,preco);
189     numeroCarros := numeroCarros + 1;
190
191     clrscr;
192     menu;
193
194     writeln('Deseja inserir outro carro? 1(sim)/0(nao)');
195     readln(continua);
196
197
198     if(numeroCarros = MAX)then //se o limite for atingido...
199     begin
200         writeln('Numero maximos de carros atingido!');
201     end;
202
203     clrscr;
204     menu;
205
206
207 until((numeroCarros = MAX) or (continua = 0));

```

C:\Users\vids\Documents\LUCAS DO VALE\trabalhos.exe

***** Loja de Automoveis *****

- 1 - Incluir carros;
- 2 - Editar preco de um carro;
- 3 - Consultar dados de um carro;
- 4 - Listar todos os carros de um fabricante;
- 5 - Listar todos os dados;
- 6 - Apagar arquivo existente;
- 7 - Sair

Digite o nome do modelo:
KA

Digite a placa:
LNB4081

Digite o nome do fabricante:
Ford

Digite o ano de fabricacao:
2009

Digite o preco:
19000

Figura 4: entrada de dados.

```

73 procedure incluiOrdenado(var vetCarros: vetor; var numeroCarros: integer; modelo: string; placa: string; fabricante: string; ano: integer; preco: real);
74 //este procedimento realiza a ordenação dos carros dentro do vetor
75 var
76     antes,depois: integer;
77
78 begin
79
80     if (numeroCarros = 0) then //se o vetor estiver vazio
81     begin
82         vetCarros [numeroCarros + 1].modelo := modelo;
83         vetCarros [numeroCarros + 1].placa := placa;
84         vetCarros [numeroCarros + 1].fabricante := fabricante;
85         vetCarros [numeroCarros + 1].ano := ano;
86         vetCarros [numeroCarros + 1].preco := preco;
87
88     end
89
90     else if (numeroCarros >= 1) then //se o vetor já estiver preenchido
91     begin
92         vetCarros [numeroCarros + 1].modelo := modelo;
93         vetCarros [numeroCarros + 1].placa := placa;
94         vetCarros [numeroCarros + 1].fabricante := fabricante;
95         vetCarros [numeroCarros + 1].ano := ano;
96         vetCarros [numeroCarros + 1].preco := preco;
97
98         depois := numeroCarros + 1;
99         antes := numeroCarros;
100
101         while (vetCarros [antes].modelo > vetCarros [depois].modelo) and (antes > 0) do //vai verificar se o elemento está desordenado, e portanto
102

```

Figura 5: procedimento para ordenação dos carros.

Por fim, os dados são copiados do array de registros e escritos no arquivo, atualizando o mesmo. A Figura 6 esclarece detalhes.

```

209     rewrite(catalogo);
210
211     for i := 1 to numeroCarros do //atualiza arquivo com os novos dados
212     begin
213         reg.modelo := vetCarros [i].modelo;
214         reg.placa := vetCarros [i].placa;
215         reg.fabricante := vetCarros [i].fabricante;
216         reg.ano := vetCarros [i].ano;
217         reg.preco := vetCarros [i].preco;
218
219         write(catalogo,reg);
220     end;
221
222     close(catalogo);

```

Figura 6: atualização do arquivo.

Ao digitar a opção 2, o usuário adquire a possibilidade de pesquisar por um modelo e editar ou não seu preço. O procedimento **edicaoPreco()** permite realizar tal operação.

Assim como está presente na Figura 3, é feita antes de tudo, uma cópia dos registros contidos no arquivo para um array de registros, para que sejam analisados e manipulados na memória principal. Em seguida, é requerido que o usuário entre com um modelo desejado e o programa pergunta se o usuário deseja ou não editar o valor (Figura 7). Várias restrições espalhadas ao longo do procedimento **edicaPreco()** fazem o tratamento dados, como no caso de existir mais de um modelo com mesmo nome (Figura 8), ou no caso de não existir tal modelo digitado (Figura 9).

```

writeln('Digite o nome um modelo: ');
readln(modelo);
encontrouPlaca := 0;
encontrouModelo := 0;
i := 0;

while(i < numeroCarros + 1)do //percorre todo o vetor
begin
    i := i + 1;

else //entra aqui se o modelo digitado pelo usuário for único
begin
    encontrouModelo := 1;
    clrscr;
    menu;

    writeln;
    writeln(vetCarros[i].modelo, ' ', vetCarros[i].placa, ' ', vetCarros[i].fabricante, ' ', vetCarros[i].preco);
    writeln('Deseja alterar seu preço? 1(sim)/0(nao)');
    readln(continua);

    clrscr;
    menu;

    if(continua = 1)then //só muda o preço se o usuário desejar
    begin
        writeln('Entre com o novo preço do carro: ');
        readln(vetCarros[i].preco);

        clrscr;
        menu;

    end;

end;

```

Figura 7: alteração do preço para modelo único.

```

while(i < numeroCarros + 1)do //percorre todo o vetor até o fim
begin
    i := i + 1;

    if(modelo = vetCarros[i].modelo) and (encontrouModelo = 0) and (i <= numeroCarros)then //verifica se existe mais de um carro desse modelo
    begin
        if(modelo = vetCarros[i+1].modelo) and (i+1 <= numeroCarros)then //verifica se existe mais de um carro desse modelo
        begin
            encontrouModelo := 1;
            clrscr;
            menu;

            writeln('Existe mais de um carro desse modelo!');
            writeln;

            j := i;

            while(modelo = vetCarros[j].modelo) and (j <= numeroCarros)do //imprime na tela todos os carros com o mesmo modelo
            begin
                writeln;
                writeln(vetCarros[j].modelo, ' ', vetCarros[j].placa, ' ', vetCarros[j].fabricante, ' ', vetCarros[j].preco);
                writeln;
                j := j + 1;

            end;

        end;

    end;

    while(modelo = vetCarros[j].modelo) and (j <= numeroCarros)do //imprime na tela todos os carros com o mesmo modelo
    begin
        writeln;
        writeln(vetCarros[j].modelo, ' ', vetCarros[j].placa, ' ', vetCarros[j].fabricante, ' ', vetCarros[j].preco);
        writeln;
        j := j + 1;

    end;

    if(placa = vetCarros[j].placa)then //verifica se a placa coincide com a do modelo
    begin
        clrscr;
        menu;

        writeln;
        encontrouPlaca := 1;
        writeln(vetCarros[j].modelo, ' ', vetCarros[j].placa, ' ', vetCarros[j].fabricante, ' ', vetCarros[j].preco);
        writeln;
        writeln('Deseja alterar seu preço? 1(sim)/0(nao)');
        readln(continua);

        clrscr;
        menu;

        if(continua = 1)then //só vai alterar o preço se o usuário desejar
        begin
            writeln('Entre com o novo preço do carro: ');
            readln(vetCarros[j].preco);

            clrscr;
            menu;

        end;

    end;

end;

```

Figura 8: alteração do preço para modelos repetidos.

```

while(i <= numeroCarros + 1)do //percorre até o fim do vetor
begin
    i := i + 1;

    if(encontrouModelo = 0) and (modelo < vetCarros[i].modelo) and (i < numeroCarros)then //se o modelo não existe, entra aqui para mostrar o próximo carro da lista em ordem alfabética
    begin
        encontrouModelo := 1; //para que não entre novamente nessa condição/encontrou o próximo em ordem
        writeln;
        writeln('O modelo ', modelo, ' não existe! Este é o próximo carro da lista em ordem alfabética: ');
        writeln;
        writeln(vetCarros[i].modelo, ' ', vetCarros[i].placa, ' ', vetCarros[i].fabricante, ' ', vetCarros[i].preco);
        writeln('Deseja alterar seu preço? 1(sim)/0(nao)');
        readln(continua);

        clrscr;
        menu;

        if(continua = 1)then //só altera o preço se o usuário desejar
        begin
            writeln('Entre com o novo preço do carro: ');
            readln(vetCarros[i].preco);

            clrscr;
            menu;

        end;

    end;

end;

```

Figura 9: inexistência de modelo.

Por fim, o arquivo é atualizado com o(s) novo(s) preço(s), assim como acontece na Figura 6.

Através da opção 3, é possível que o usuário consulte os dados de algum modelo, como a placa, o fabricante, o ano e o preço. Ao escolher esta opção, o procedimento **consultaDados** é chamado.

Dentro deste procedimento, mais uma vez, a primeira tarefa a ser realizada é a cópia dos registros do arquivo para o array de registros (vide Figura 3). Desta vez, não é necessário atualizar o arquivo, pois nada será alterado, apenas consultado.


Depois de copiados, os registros do array serão comparados com um modelo que o usuário digitar. Ao encontrar determinado modelo, são impressos na tela todos os seus dados, como mostra a Figura 10.

```
writeln('Entre com um modelo: ');
readln(modelo);

clrscr;
menu;

i := 0;
verProximo := 0;
encontrou := 0;
while(i < numeroCarros + 1)do //permanece aqui até atingir o fim do vetor
begin
    i := i + 1;

    if(modelo = vetCarros[i].modelo) and (i <= numeroCarros)then //verifica se o modelo digitado é igual ao modelo contido no campo do vetor
    begin
        encontrou := 1;
        writeln;
        writeln(vetCarros[i].modelo, ' ', vetCarros[i].placa, ' ', vetCarros[i].fabricante, ' ', vetCarros[i].ano, ' R$ ', vetCarros[i].preco:2:2);
        writeln;
    end
end
```



***** Loja de Automoveis *****

1 - Incluir carros;
2 - Editar preco de um carro;
3 - Consultar dados de um carro;
4 - Listar todos os carros de um fabricante;
5 - Listar todos os dados;
6 - Apagar arquivo existente;
7 - Sair

Entre com um modelo:
Santana

***** Loja de Automoveis *****

1 - Incluir carros;
2 - Editar preco de um carro;
3 - Consultar dados de um carro;
4 - Listar todos os carros de um fabricante;
5 - Listar todos os dados;
6 - Apagar arquivo existente;
7 - Sair

Santana LOT3781 Volkswagen 2000 R\$ 11000.00

Figura 10: entrada e impressão do modelo correspondente.

Caso haja mais de um modelo com mesmo nome, o programa oferecerá a possibilidade de o usuário consultar um a um até não haja mais nenhum igual (Figura 11). Outra restrição importante é que quando o modelo que o usuário fornecer não constar no array, o programa emitirá um aviso (Figura 12).

```

if(modelo = vetCarros[i].modelo) and (i <= numeroCarros)then //verifica se o modelo digitado é igual ao mo
begin
    encontrou := 1;
    writeln;
    writeln(vetCarros[i].modelo, ' ', vetCarros[i].placa, ' ', vetCarros[i].fabricante, ' ', vetCarros[i].ano, ' R
    writeln;

    if(modelo = vetCarros[i+1].modelo)then //verifica se o proximo campo do vetor também contém o mesmo mo
    begin
        writeln('Ainda existem outros carros do mesmo modelo. Deseja ver o proximo?1(sim)/0(nao)');
        readln(verProximo);

        i := i + 1;

        while((modelo = vetCarros[i].modelo) and (verProximo = 1))do //permanece aqui ate que não haja mai
        begin
            clrscr;
            menu;

            writeln;
            writeln(vetCarros[i].modelo, ' ', vetCarros[i].placa, ' ', vetCarros[i].fabricante, ' ', vetCarros[i]
            writeln;
            i := i + 1;
            if(modelo = vetCarros[i].modelo)then //se ainda existir um modelo igual na proxima posição...
            begin
                writeln('Ainda existem outros carros do mesmo modelo. Deseja ver o proximo?1(sim)/0(nao)');
                readln(verProximo);
            end;
        end;

        encontrou := -1; //a variável sofre esta atribuição se pelo menos um modelo dos repetidos já tiver
    end;
end;

```

Palio LF07765 Fiat 2007 R\$ 16500.00

Ainda existem outros carros do mesmo modelo. Deseja ver o proximo?1(sim)/0(nao)

Figura 11: quando há mais de um modelo com mesmo nome.

```

else if ((modelo < vetCarros[i].modelo) or (i > numeroCarros)) and (encontrou = 0) then
begin
    clrscr;
    menu;

    writeln('Este modelo nao existe!');
    writeln;
    encontrou := -1; //neste caso a variável sofre esta atribuição para indicar que o mo
end;

```

Figura 12: inexistência de modelo.

A opção 4 permite que, através do procedimento **listagemFabricante**, o usuário entre com um fabricante qualquer, e logo em seguida, se este existir, o programa exiba na tela todos os modelos deste fabricante. Caso contrário, o usuário é alertado. Este procedimento basicamente lê cada registro do arquivo e compara o campo “fabricante” com o fabricante desejado. Observe a Figura 13.

A opção 5 é bastante semelhante à anterior, com única diferença de que, através do procedimento **listagemGeral**, o usuário tem a possibilidade de contemplar todos os dados de todos os carros presentes no arquivo (Figura 14). Além disso, como neste procedimento é preciso imprimir diferentes modelos de diferentes fabricantes, foi feito o uso da função **gotoxy()** para alinhar corretamente os dados.

```

begin
  assign(catalogo,'catalogo.bin');
  reset(catalogo);

  encontrou := 0;

  writeln('Digite um fabricante: ');
  readln(fabricante);

  clrscr;
  menu;

  while not eof(catalogo) do
  begin
    read(catalogo,reg);  //a cada Loop faz a leitura de um registro do arquivo
    if(fabricante = reg.fabricante)then  //se o fabricante informado corresponder com o fabric
    begin
      encontrou := 1;
      writeln;
      writeln(reg.modelo,' ',reg.placa,' ',reg.fabricante,' ',reg.ano,' R$ ',reg.preco:2:2);
    end;
  end;

  if(encontrou = 0)then
  begin
    writeln('Este fabricante nao existe!');
  end;

  close(catalogo);

  writeln;
end;

```

Figura 13: listagem de carros e dados por fabricante.

```

619 assign(catalogo,'catalogo.bin');
620 reset(catalogo);
621
622 gotoxy(1,13);
623 write('Modelo');
624 gotoxy(11,13);
625 write('Placa');
626 gotoxy(21,13);
627 write('Fabricante');
628 gotoxy(35,13);
629 write('Ano');
630 gotoxy(41,13);
631 write('Preco');
632
633 i := 0;
634
635 while not eof(catalogo)do  //Lê o
636 begin
637   read(catalogo,reg);
638   gotoxy(1,15+i);
639   write(reg.modelo);
640   gotoxy(11,15+i);
641   write(reg.placa);
642   gotoxy(21,15+i);
643   write(reg.fabricante);
644   gotoxy(35,15+i);
645   write(reg.ano);
646   gotoxy(41,15+i);
647   write('R$',reg.preco:2:2);
648   i := i + 2;
649 end;
650
651 close(catalogo);

```

Modelo	Placa	Fabricante	Ano	Preco
Brava	LFG7765	Fiat	2000	R\$8000.00
C4	LIL5027	Citroen	2010	R\$45000.00
Civic	KJF4470	Honda	2010	R\$51500.00
Corsa	KFG1467	Chevrolet	2009	R\$8000.00
Corsa	KUI4499	Chevrolet	2003	R\$10200.00
Fiesta	L0L2692	Ford	2001	R\$11500.00
Fiesta	LTE5580	Ford	2012	R\$28900.00
Gol	KDF6120	Volkswagen	2010	R\$18900.00
KA	LNB4081	Ford	2009	R\$19000.00
Kombi	2008	Volkswagen	2008	R\$26800.00
Monza	LFG1261	Chevrolet	1995	R\$4500.00
Omega	LOG4545	Chevrolet	1995	R\$7500.00
Palio	LF07765	Fiat	2007	R\$16500.00
Palio	LJA5447	Fiat	2002	R\$10000.00
Palio	LKA5634	Fiat	2002	R\$9500.00
Santana	LOT3781	Volkswagen	2000	R\$11000.00
Zafira	KRE2233	Chevrolet	2013	R\$55000.00

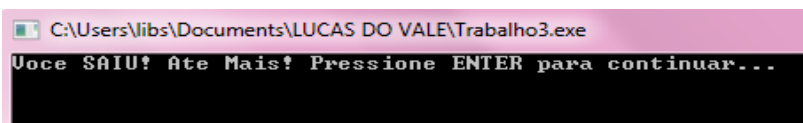
Figura 14: listagem de todos os dados contidos no arquivo.

O programa também disponibiliza a opção 6, para efeito de limpeza. Esta opção chama uma função **deletefile** que deleta o atual arquivo (Figura 15).

```
6: begin
    deletefile('catalogo.bin'); //deleta o arquivo
    writeln('O arquivo catalogo.bin foi excluido!');
    writeln;
end
```

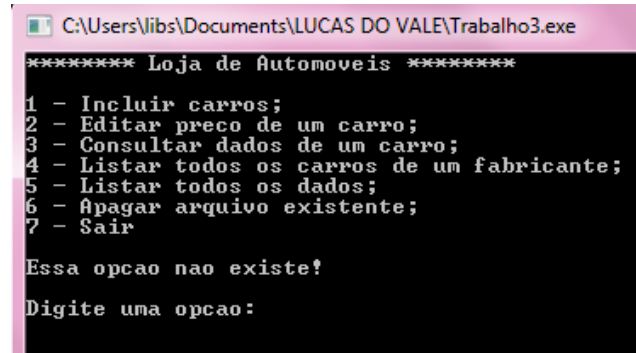
Figura 15: exclusão do arquivo.

O programa disponibiliza a opção 7. Após digitar essa opção, o usuário sai do programa, e é impressa na tela uma mensagem de finalização, como está demonstrado na Figura 16. Por fim, caso o usuário digite uma opção diferente das anteriores, o programa exibe uma mensagem, alertando sobre a inexistência da opção, como está esclarecido na Figura 17.



```
C:\Users\libs\Documents\LUCAS DO VALE\Trabalho3.exe
Voce SAIU! Ate Mais! Pressione ENTER para continuar...
```

Figura 16: alerta de saída do programa.



```
C:\Users\libs\Documents\LUCAS DO VALE\Trabalho3.exe
***** Loja de Automoveis *****
1 - Incluir carros;
2 - Editar preco de um carro;
3 - Consultar dados de um carro;
4 - Listar todos os carros de um fabricante;
5 - Listar todos os dados;
6 - Apagar arquivo existente;
7 - Sair
Essa opcao nao existe!
Digite uma opcao:
```

Figura 17 : quando uma opção inválida é digitada.

4. Conclusão

O desenvolvimento do código em si não foi tão simples, visto que o processo para se trabalhar com ordenação alfabética não é nada trivial. As restrições também devem ser consideradas nos momentos em que sejam identificadas incoerências, tomadas as entradas de dados. Visto que aplicação do conceito de persistência foi concebida com sucesso, entre outros aspectos, podemos concluir que o programa cumpriu com sua finalidade.