



Lázaro José Rodrigues Júnior  
Carlos Eduardo de Sousa

**TRABALHO DE ALGORITMOS E ESTRUTURAS DE DADOS I**  
(Trabalho final)

BambuÍ-MG  
Engenharia de Computação  
2019

## Pet shop “Happy Dog”

A ideia de implementação do programa surgiu da necessidade de facilitar o comércio em uma pet shop especializada para cachorros, registrando de forma fácil os clientes, os atendimentos e o dinheiro circulado no negócio.

Para atender estas demandas desenvolvemos um programa de fácil uso capaz de atender todas as necessidades deste usuário, retornando resultados rápidos e precisos, que facilitarão por completo seu cotidiano.

A seguir são apresentadas as principais estruturas utilizadas na montagem do programa:

**“Int main ()”** - Bloco principal do programa. Tem a função de chamar o procedimento “menu”.

**“Void menu ()”** – Procedimento responsável por interagir com o usuário. É através dele que conseguiremos dar fluidez ao programa. Nele encontramos a estrutura de repetição *do/while* e um *switch case* que possui todas as ações a serem tomadas, e é responsável por direcionar o usuário a uma função, dependendo de suas escolhas. O usuário possui 6 opções que poderão auxiliar em suas necessidades diárias (explicaremos melhor cada uma das funções com mais detalhes no decorrer deste trabalho), são elas:

**Caso 1:** Coletar os dados de um novo cliente.

**Caso 2:** Registrar um atendimento.

**Caso 3:** Finalizar um atendimento.

**Caso 4:** Consultar um data.

**Caso 5:** Consultar um cliente.

**Caso 6:** Finalizar o programa.

Se por ventura o usuário digitar um número que não corresponde a nenhuma das operações, o programa o adverte e retorna novamente ao menu, mostrando as opções válidas.

**“Void arte ()”** - Procedimento responsável pela criação do cabeçalho, dá um visual mais agradável ao programa. Composto por uma série de comandos de saída e caracteres especiais que formam o nome da pet shop “Happy Dog”.

**“Int cad ()”** - Função responsável por cadastrar dados em um arquivo de texto (“cadastro.txt”), que poderão ser lidos posteriormente. É acionada quando o usuário desejar armazenar os dados de seus clientes, tais como: nome do dono, nome do animal, raça do animal, cor do animal, porte do animal, sexo do animal, se o animal é vacinado, e a data do cadastro.

Inicialmente encontramos um comando de abertura de arquivo, depois a declaração de um vetor do tipo *string* que armazenará os dados a serem guardados (“cadastro[8]”), um vetor de constante do tipo literal que mostrará as opções (“cabecalho[8]”), e por fim, variáveis que serão usadas no processo de armazenamento de dados.

Os dados são armazenados no vetor através da estrutura de repetição *for* que vai se repetir até que ele seja preenchido. Após total preenchimento do vetor se inicia a fase de registro de cada item, que funciona da seguinte maneira: dentro de um laço de repetição é atribuído a variável “escreve” o valor do vetor(“cadastro[i]”) mais um “;” que vai ser responsável por separar cada dado do cliente. Ao final do laço de repetição a variável “escreve” registra o dado no arquivo. Quando terminado o vetor, o programa muda para linha seguinte que poderá posteriormente armazenar outros dados da mesma forma. Ou seja, cada linha do arquivo corresponderá aos dados de um cliente, que poderão ser lidas quando necessário.

A função também possui um *do/while* que é encarregado de manter ou não o usuário na função de cadastramento. Se o usuário desejar cadastrar outro cliente é inserido a opção (1), caso queira voltar ao menu principal a opção (0).

**“Int reg ()”** – Função responsável por registrar um atendimento salvando os dados em um arquivo de texto (“registro.txt”). O funcionamento desta função basicamente se assemelha ao da função “int cad ()” registrando os dados em arquivo, porém, com algumas peculiaridades. Em seu escopo possuem variáveis que vão armazenar dados em um outro arquivo chamado “valores.txt”. Para salvar os dados neste arquivo é acionada a função “int valor” que receberá por parâmetro o tipo de tosa, o tipo de vacina e um código. O código recebido pela função será o mesmo informado em uma das opções de atendimento, tal código poderá ser criado pelo próprio usuário contando que seja único. Este código será utilizado posteriormente e corresponderá a um atendimento feito.

**“Int Faten ()”** – Quando o usuário desejar finalizar um atendimento, ou seja, gerar o valor da compra juntamente com os dados do cliente, é acionada a função “int Faten ()”. Inicialmente a função solicita que seja digitado o código de registro (o mesmo criado anteriormente). Ao receber o código se inicia o processo de leitura de dados armazenados em arquivo. Neste processo um arquivo específico é aberto e utilizando laços de repetição é lido e os valores armazenados em um vetor ignorando os “;”. Depois o código informado é comparado aos armazenados no vetor, se os códigos forem iguais a função compara os dados de atendimento, caso o cliente opte por algum tipo de tosa a função “int tosa” retorna o valor da tosa, da mesma forma “int vacina” retornar o valor da vacina. Quando todos os valores são coletados a função envia para “void boleto” o valor total, os dados do cliente, o tipo de tosa e vacina para que ele possa gerar o boleto finalizando assim um atendimento.

**“Int Dcon ()”** – Ao iniciar esta função imediatamente é solicitado ao usuário uma data que deve estar no formato xx/xx/xxxx. Depois inicia-se o processo de leitura que já foi citado anteriormente. Ao final deste processo, os registro que foram armazenados em um vetor são comparados com a data informada, se as datas forem iguais o programa retorna todos os dados de clientes que se registraram na data situada. Se a data digitada não for encontrada, é retornado uma mensagem informando que a data não se encontra nos registros.

**“Int Ccon ()”** – Função responsável por retornar os dados de um cliente específico. Possui basicamente a mesma estrutura de comparação da função “int Dcon” com estruturas de leitura de dados e condicionais.