

# COOPMoto&Táxi – Cooperativa de Motos & Táxis

Felipe Pinheiro de Oliveira

Engenharia de Computação – Universidade Estadual de Feira de Santana (UEFS)

Avenida Transnordestina, s/n – Novo Horizonte – 44.036-900 – Feira de Santana - BA

lipinhow\_fsa@hotmail.com

**Abstract.** *This report has been written based on a program developed in C programming language for a cooperative taxi and motorcycle taxi in order to improve control of customer information and the company cooperated well as schedule of races.*

**Resumo.** *Este relatório foi escrito com base num programa desenvolvido em linguagem de programação C para uma cooperativa de táxi e moto-táxi com o objetivo de melhorar o controle das informações de clientes e cooperados da empresa além do agendamento de corridas.*

## 1. Introdução

Com o avanço da tecnologia, muitas empresas têm informatizado todo o seu sistema possibilitando um melhor controle de suas informações e assim automatizando uma série de processos.

Para a realização do programa utilizou-se a linguagem de programação C que foi solicitada pelo problema. O programa tem como objetivo armazenar o cadastro de cliente, cooperados e corridas. Antes de se iniciar o programa, utilizou-se um algoritmo desenvolvido em forma de fluxograma feito anteriormente que descrevia passo a passo cada etapa do programa.

O relatório inicia-se comentando sobre a apresentação do problema explicando como ele foi proposto, seguido da fundamentação teórica onde é explicado as ferramentas da linguagem de programação C utilizadas na implementação do programa, logo após a metodologia contendo quais as ferramentas e materiais de estudo utilizados, as discussões dos resultados contendo a descrição detalhada de como foi desenvolvido o programa citando os resultados obtidos e problemas encontrados, por fim uma conclusão informando os aprendizados obtidos e a importância do problema no ambiente acadêmico.

## 2. Apresentação do problema

O problema propôs a criação de um programa que fosse capaz de atender as necessidades da empresa COOPMoto&Táxi que é uma cooperativa de moto-táxi e táxi, com o objetivo de agilizar o processo de atendimento das corridas possibilitando um maior controle das informações.

O programa deve ser capaz de realizar o cadastro de clientes, cooperados e corridas, contendo todos os dados e informações necessárias, além de ser capaz de gerar

um recibo de cada corrida. Todas essas informações devem ser salvas em um banco de dados temporário caso seja necessária a modificação de algum dado enquanto o programa permanecer aberto.

### **3. Fundamentação Teórica**

Segundo Brian W. Kernighan e Dennis M. Ritchie, C é uma linguagem de programação de finalidade geral que permite economia de expressão, modernos fluxos de controle e estruturas de dados e um rico conjunto de operadores. A falta de restrições e sua generalidade tornam a linguagem de programação C mais conveniente e eficaz para muitas tarefas do que linguagens supostamente mais poderosas.[1]

Para o desenvolvimento do problema foi necessário a utilização de diversos conhecimentos sobre a linguagem C que ainda não haviam sido utilizados anteriormente.

#### **2.1. Estruturas**

Uma estrutura é uma coleção de uma ou mais variáveis que podem ser de tipos diferentes, colocadas juntas sob um único nome para a sua manipulação. Elas ajudam a organizar dados, além de permitirem que um grupo de variáveis relacionadas possam ser tratadas como uma unidade ao invés de entidades separadas.

As estruturas são utilizadas geralmente quando se há a necessidade de se produzir um registro de algo que possua um ou diversos tipos de variáveis, um exemplo que pode ser utilizado é o cadastro de clientes de uma determinada empresa, onde seus dados farão parte da estrutura.

#### **2.2. Strings**

Strings nada mais são do que vetores de caracteres, elas são declaradas como sendo do tipo “char”, além disso, deve ser informado qual o seu tamanho através do comando “[ ]”, existem certos comandos utilizados para outros tipos de variáveis que não funcionam para strings, um exemplo pode ser o “strcmp” utilizado no problema, cujo seu objetivo é o de igualar duas variáveis do tipo string.

### **3. Metodologia**

Para a realização do problema utilizou-se o Dev C++, um compilador para a linguagem C. Este problema exigiu um estudo aprofundado, por meio de livros sobre algumas funções da linguagem de programação C que seriam úteis no desenvolvimento do sistema de controle da cooperativa proposta no problema.

A princípio utilizou-se o algoritmo em forma de fluxograma do primeiro problema que continha uma sequência dos passos necessários para o desenvolvimento do programa.

### **4. Discussão dos Resultados**

Feito uma análise detalhada do problema, deu-se início a solução, o fluxograma foi passado para a linguagem C. A princípio foram definidas as bibliotecas que seriam necessárias para os comandos utilizados ao longo do programa, logo após foram

definidas as estruturas de registro dos cooperados, clientes, tipos de corrida e recibo, através do comando “struct”, que foram declaradas fora da função principal sendo consideradas globais, e dentro de cada “struct” foram inseridos variáveis do tipo “string” (vetor de caracteres), cada um desses vetores corresponde a um tipo de dado, como pode ser visto na Figura 1 que contém a estrutura feita para os clientes.

```
struct cliente {  
  
    char cpf_cl[20];  
    char nome_cl[20];  
    char end_cl[20];  
    char end_dest_cl[20];  
    char tel_cl[10];  
  
}typedef cliente;
```

**Figura 1. Estrutura de Clientes**

Logo após foram declaradas as variáveis que seriam utilizadas ao longo do programa dentro da função principal, além de declarar variáveis do tipo das estruturas declaradas acima da função “main”, depois foi construído um menu contendo as opções dentro de uma estrutura de laço “while” que se repetirá enquanto a condição proposta nele estiver sendo satisfeita, neste menu contém o cadastro e busca de clientes e cooperados além do agendamento das corridas e emissão de um recibo. O cadastro de clientes e cooperados foram criados utilizando as estruturas criadas anteriormente, dentro de um laço “do, while” para que essa condição seja satisfeita pelo menos uma vez além de ser possível a realização de quantos cadastros fossem necessários, lembrando que o limite de cadastros foram limitados para apenas 5, para esta limitação utilizou-se um laço de comando “for” e dentro dele utilizou-se o comando de entrada “gets” para que pudesse ser lido quando fosse necessário o uso de espaços entre uma palavra e outra, além disso foi necessário o comando “fflush(stdin)” para limpar o *buffer* de entrada que serviu para reparar alguns erros que apareceram no momento do registro dos dados. As variáveis por fazerem parte de uma estrutura tiveram de ser utilizadas da forma exemplificada na Figura 2.

```
printf ("=====CADASTRO DE CLIENTES=====");  
printf ("\nCadastro de Cliente Nº%d\n", cadcl+1);  
fflush(stdin);  
printf ("Digite o CPF: ");  
gets (cl_1[cadcl].cpf_cl);fflush(stdin);  
printf ("Digite o nome: ");  
gets (cl_1[cadcl].nome_cl);fflush(stdin);  
printf ("Digite o endereço do cliente: ");  
gets (cl_1[cadcl].end_cl);fflush(stdin);  
printf ("Digite o endereço de destino: ");  
gets (cl_1[cadcl].end_dest_cl);fflush(stdin);  
printf ("Digite o telefone: ");  
gets (cl_1[cadcl].tel_cl);fflush(stdin);
```

**Figura 2. Cadastro Clientes**

Depois foram implementados vários comandos “printf” para imprimir na tela os dados registrados além do comando “scanf” para fornecer a possibilidade de se realizar outro cadastro ou sair do laço e retornar ao menu principal. Foi necessário o implemento de condições através do comando “if” para que não se pudesse realizar o cadastro de clientes e cooperados com o mesmo CPF, pois ele seria usado futuramente no momento da busca, para a comparação destas “strings” utilizou-se o comando “strcmp”.

As corridas foram divididas em duas opções, normais e pré-agendadas, esta segunda opção foi posta caso a corrida tenha sido solicitada para ser realizada em um momento futuro, cada tipo de corrida foi declarada em estruturas diferentes. Dentro da opção de cada corrida, deveria ser feito uma fila de cooperados, para que existisse uma ordem estabelecida, sendo que ao terminar cada corrida o cooperado deve ser transportado para o final da fila, para isto houve a tentativa de se implementar um laço “for” fazendo com que o registro das corridas fossem feitas na ordem de registros dos cooperados, porém não seria possível fazer com que o primeiro cooperado retornasse para o final da fila, além disso ocorreu um erro pois não seria possível retornar ao menu principal no momento do registro das corridas.

Foi necessária também a criação de um recibo para ser impresso na tela contendo os dados de cada corrida, para sua implementação utilizou-se o comando de entrada “gets” e o de saída “printf” para exibir o recibo na tela.

Na busca de clientes e cooperados definiu-se que ela seria realizada através do CPF, ao digitá-lo seria feito uma comparação com todos os outros CPFs de cada categoria, caso encontre algum igual imprimiria na tela o restante dos dados daquele cliente ou cooperado, para esta comparação utilizou-se mais uma vez o comando “strcmp”, caso não encontrasse nenhum CPF igual imprimiria uma mensagem informando o ocorrido. Ao se concluir a busca há a opção também da alteração do cadastro caso seja necessário, isto foi possível fazendo com que as alterações ocupassem o mesmo espaço no vetor que antes constavam os dados antigos.

## **Conclusões**

Este relatório discutiu sobre um programa para ser implementado numa cooperativa de moto-táxis e táxis, este sistema deveria agilizar todos os processos necessários para a empresa possibilitando um controle maior das informações por parte da gerência. Para a solução destas propostas criou-se um sistema de cadastros e buscas além de geração de recibos.

Este problema foi muito importante na aprendizagem da linguagem de programação C, ao solucioná-lo foi possível adquirir novos conhecimentos sobre a linguagem além de despertar no estudante o autodidatismo fazendo com que ele busque soluções eficazes para a solução do problema.

O programa cumpre com as funções solicitadas pelo problema, ao não ser pela fila de cooperados nas corridas, pois eles deveriam estar em ordem e ao retornarem para a cooperativa serem postos no final da fila.

Este problema abre margem ao uso de outras funcionalidades, pois ele não armazena os dados por tempo ilimitado, ao ser fechado ele perde todas as informações cadastradas, este sistema poderá ser implantado em situações futuras.

## **8. Referências**

- [1] C: a linguagem de programação - padrão ANSI ; 15 ed ; Brian W. Kernighan & Dennis N. Ritchie.