



UNIVERSIDADE FEDERAL DE SANTA CATARINA - CAMPUS JOINVILLE  
CENTRO TECNOLÓGICO DE JOINVILLE  
PROGRAMAÇÃO III – 2021/1

## SIMULADOR DE TRANSPORTE E LOGÍSTICA

Breno Pena de Almeida, Vitor Uchikawa

### RESUMO

*Este trabalho foi feito com o intuito de desenvolver um algoritmo que simula um sistema de logística de uma transportadora. Esse sistema verifica se quais caminhões e quais motoristas estão em serviço, a partir dos dados seus dados informados pelo o usuário. Dessa forma, se o programa identificar algum caminhão e algum motorista disponível, o algoritmo abre uma ordem de serviço e altera a situação atual que ambos se encontram.*

**Palavras-chave:** Logística, algoritmo, caminhão, motorista.

### 1 FUNCIONAMENTO

Foram criadas as classes bases “endereço” e “data” que seriam úteis na criação de posteriores classes. “Data” armazena um dia, mês e ano e foi feita de modo a impedir datas inválidas, como meses acima de 12. A classe “endereço” contém os atributos necessários para identificar uma localização (rua, estado, país e CEP), também impedindo set inválido de CEP (tamanho errado e não numerais). Em seguida foram criadas as classes bases “pessoa” e “veículo”. “Pessoa” tem os dados de um indivíduo relacionado governamentalmente, com nome, gênero (pertencente a classe enum “gêneros” que admite masculino, feminino e n/a (not available)) e CPF, que impede set inválido com tamanho discrepante e inserção de caracteres não numéricos. “Veículo” tem atributos pertencentes a veículos que estariam cadastrados no governo como RENAVAM (impede set de não numérico e tamanho inválido), placa (impede set de tamanho inválido) e data da última revisão (composição com classe “data”). Em seguida foram criadas as classes “motorista” e “caminhão”. “Motorista” herda a classe “pessoa” e adiciona atributos relacionados ao trabalho do indivíduo, sendo esses os anos de experiência, o número da carteira de motorista (impede set de não numerais e tamanho errado), vencimento da carteira de motorista (composição com “data”), e status de ocupado (se

se True, está atualmente ocupado para uma entrega). “Caminhão” herda de “veículo” e armazena o modelo, número atribuído pela empresa ao automóvel, e status de sendo usado (se true está em uma entrega). As classes citadas possuem funções membro de set e get para manipulação dos dados que se possível a não atribuição de um argumento (atribuição inválida) é retornado false, e sobrecarga do operador de inserção “<<” para fácil impressão em terminal. Foi finalmente criada uma classe “ordem” onde ficará a ordem de serviço com atributos do tipo “caminhão” e “motorista” que serão separados para o trabalho (composição de “motorista” e “caminhão”), atributos da classe “data” para quando o pedido iria sair e quando chegaria ao destino, e por fim um atributo de observações para identificação da carga. Esta classe tem funções de get e set, sendo que se possível utilizar o motorista ou caminhão fornecido será ele será definido para ocupado, e ele já estiver ocupado será retornado false pela função.

Por fim, foi criada a classe Lista\_Geral, visando que essa classe possui o intuito de fazer com que o programa interaja com as outras classes. A Lista geral possui três vetores que irão usar para guarda os dados de motorista, caminhão e ordem e, além disso, ela também possui outros atributos que irão auxiliar na manipulação dos dados dos vetores. A lista Geral possui funções membros que manipula os dados dos vetores de acordo com a necessidade do usuário. As funções membros insere caminhão, insere motorista e insere ordem, insere os objetos nos vetores respectivos, que corresponde ao banco de dados da empresa. Onde possui os dados dos caminhões, motorista e serviços, os dados são inserido a parti da função push.back, essa função ela faz parte da biblioteca vector, ela insere elementos no final do vetor. Em Motorista, será inserido os dados de CPF, n° carteira de motorista, data vencimento, situação que ele se encontra, nome e o gênero. Caminhão será inserido os dados, idade, modelo, placa, remavam e data da última revisão. Ordem de serviço será inserido os dados, data de saída, data de entrega, frete e endereço. A função tamanho, imprime o tamanho dos vetores através da função size(), que é encontra na biblioteca vector.

A função imprime banco de dados ela irá imprimir os dados dos respectivos vetores através de um for, função ela imprimir os dados de motorista, caminhão e ordem a parti da sobrecarga do operador << . Ao adicionar este operador, a função irá imprimir os dados do objeto inserido no vetor. O for usado para essa função, irá ter como condição a variável max que receberá o valor retornado pela função imprime tamanho, o mesmo ocorre para os outros for usado na classe Lista geral.

Em seguida temos a função fazer pedido, ela irá abrir a ordem de serviço, primeiramente a função irá verificar se se o vetor banco de dados ordem está vazio, ou seja se tem alguma serviço cadastrada no sistema, se o vetor não estiver vazio, ou seja, algum serviço cadastrado no sistema, o programa irá verificar em seguida se existe caminhões e motoristas registrados no sistema, ou seja se o vetor banco de dados motorista e caminhão está vazio, s e estiver vazio irá imprimir uma mensagem que não possui caminhão e motorista registrado. Se a empresa possui algum caminhão e motorista registrado, ela irá verificar a situação de cada caminhão e motorista registrado, se encontrar algum caminhão e motorista sem serviço, abrirá uma ordem de serviço. Caso não tenha caminhão sem serviço, ela irá imprimir que não tem caminhão disponível, caso não tenha motorista sem serviço, ela irá imprimir que não tem motorista disponível, caso não tenha caminhões e motoristas sem serviço ela irá imprimir que não caminhão e motoristas disponíveis. Os motoristas não possuem seus próprios caminhões, pode ocorrer de empresa ter mais caminhões do que motorista ou mais motorista do que caminhões, para que uma ordem de serviço seja aberta, deve

ter um motorista e um caminhão que não esteja trabalhando. Tendo essa condição respeitada o algoritmo irá abrir uma ordem de serviço para aquele caminhão e motorista, imprimirá a ordem de serviço e automaticamente irá mudar a situação que eles se encontram no sistema, ou seja, o motorista e caminhão estando sem serviço, passara a ficar em serviço a parti desse momento. Ao abrir a ordem de serviço, a ordem irá sair do banco de dados da empresa, ou seja, irá ser removido do vetor devido o serviço ser realizado. O vetor da ordem simula uma fila, ou seja, o primeiro que entra, é o primeiro que sai, portando os serviços são feitos a parti da ordem de chegada, e assim que forem realizados eles são removidos do sistema, ou seja do vetor.

A função `mudar_situacao_motorista` e `mudar_situacao_caminhao`. permitirá que o usuário mude a situação que o caminhão e motorista se encontra, caso o algoritmo inicia no momento que um caminhão ou motorista esteja em serviço ou não. O usuário poderá mudar automaticamente a situação que ambos se encontram, no caso motorista, o usuário irá o CPF daquele motorista a função irá buscar o CPF do motorista a parti de um `for`, que ao encontra ela imprimirá o nome do motorista que foi modificado, e a lista atualizada. A função `motorista_imprime` imprime a mensagem de acordo como o gênero. Por exemplo se mandar imprimir Laura, ela irá imprimir A situação da motorista Laura foi mudada, e se mandar imprimir Breno, ela irá imprimir a situação do motorista Breno foi mudada. Caso não encontre o motorista relacionado ao CPF ela irá imprimir uma mensagem não foi encontrado motoristas com o CPF. A função `mudar_situacao_caminhao` ela também emitira a mensagem que o caminhão com a placa foi mudado a situação e caso não encontre o caminhão emitira uma mensagem que não encontrou caminhão relacionado aquela placa.

As funções `remove`, permite que um usuário remova motorista e caminhão do sistema da empresa, ou seja ele poderá remover um motorista e caminhão em uma posição desejada no vetor, essa remoção é feita pela função `erase()` que se encontra dentro da biblioteca `vector`. Para remover o motorista o usuário insere o CPF do motorista a ser removido, e o programa irá buscar esse CPF através de um `for`, caso encontre o programa irá removê-lo e emitira a mensagem que o motorista foi removido, caso não encontre imprimira a mensagem não possui motoristas relacionado a esse CPF, essa mensagem mudará de acordo com o gênero do motorista como as funções `mudar_situacao` explicada acima. O mesmo ocorre para caminhão, mas para remover o caminhão desejado é necessário inserir a placa para buscar o respectivo caminhão. A função `tamanho_caminhao_disponivel()`, `tamanho_motorista_disponivel` ela retorna a quantidade de caminhões e motorista que tem não estão em serviço. Já a funções `tamanho_caminhao_n_disponivel` e `tamanho_motorista_n_disponivel` ela retorna à quantidade de caminhões que estão em serviço.

O arquivo `main.cpp` ele possuirá as funções `testevu()` e `testebre()` são as funções que teste. A principal `main` do programa criara o objeto `lista` 1, e inicialmente imprimira as opções que o usuário possui para manipulação dos dados, um menu. As opção são:

Cadastrar motorista.

Cadastrar caminhão.

Remover motorista.

Remover caminhão.

Visualizar banco de dados.  
Abrir ordem de serviço.  
Cadastrar serviço.  
Visualizar lista de serviços.  
Remover serviço  
Configurações  
Sair do programa.

Cada uma dessas opções é dada por um valor que o usuário digitará. Para selecionar cada opção é usado um switch dentro de while, o while faz com que quando o usuário digitar 11 o programa irá se encerrar. No caso do switch cada case corresponde a uma opção. No caso da opção cadastrar motorista caminhão e serviço, o usuário irá inserir os dados que entrarão em algumas variáveis, e por fim chamaras respectivas função insere motorista caminhão e ordem, passando como argumento essas variáveis. As funções remover caminhão e motorista ela irá remover os caminhões e motorista chamado a função remover\_motorista e remover\_caminhão na classe lista geral, também ela irá chamar as respectivas funções tamanho banco de dados para que se o vetor estiver vazio, ela não remova, e irá imprimir uma mensagem que não possível remover. O mesmo ocorre com a opção remover serviço, em que o usuário digita a observação do serviço do serviço, e o programa chama a função remover\_ordem da classe lista geral.

A opção visualizar banco de dados, ela imprimir o banco de dados da empresa, imprimirá a quantidade de motorista registrado na empresa, chamando a função tamanho banco de dados motorista da classe lista geral e o tamanho de caminhão registrado chamado a função tamanho banco de dados caminhão da classe lista geral. Também irá imprimir as quantidades de motorista e caminhões em serviço e sem serviço chamando as respectivas funções da classe lista geral.

Ela imprimira a lista de motorista e caminhões registrado, chamado as funções imprime banco de dados, caso não tiver caminhão e motorista registrado, imprime que a lista está vazia. A opção banco de dados permitirá que modifica a situação de caminhão e motorista, chamando as funções mudar situação caminhão e mudar situação motorista da classe lista geral.

A opção abrir ordem de serviço irá chamar a função fazer pedido na classe lista geral e imprimirá a ordem de pedido aberta, com o caminhão, motorista disponíveis, cadastrado na opção cadastrar caminhão e motorista e o primeiro serviço cadastrado na opção cadastrar serviço.

A opção visualizar lista de serviço, ela irá imprimir a quantidade de serviços registrado chamando a função tamanho banco de dados ordem da classe lista geral e imprimira a lista de serviço chamado a função imprime banco de dados ordem da classe lista geral.

A opção remover serviço irá verificar se possui algum serviço registrado, a parti do tamanho que é retornado pela função tamanho banco de dados ordem se o vetor estiver vazio, ela vai imprimir a mensagem que não existe serviços cadastrado para ser removido. Caso contrário, ela irá remover o serviço como foi situado acima.

A opção configuração ela mudará o nome da empresa cadastrada, inicialmente a empresa chama Sem nome,

mas poderá ser mudado dessa opção, e verificado no banco de dados.

Por fim a opção sair do programa ela irá sair do programa, finalizando o while.

Para utilização do programa é necessário ter os seguintes programas instalados: compilador GCC (<https://gcc.gnu.org/>), CMAKE <https://cmake.org/>. Para a utilização do programa execute a tarefa make em um terminal com diretório na pasta dos arquivos fornecidos com o makefile.

## REFERÊNCIAS

DEITEL, H. M.; DEITEL, P. J. **C++ : como programar**. 5. ed. São Paulo: Pearson Prentice Hall, 2006. 1163 p. ISBN 9788543013732