**UNIVERSIDADE DA BEIRA INTERIOR**

Logo

Description automatically generated

Relatório do 1° trabalho - Linguagens de Scripting

ProfessorPaulAndrew Crocker

Filipa Barreiro, N° 44446

J. Francisco Oliveira, N° 43771

Larissa Goularte, N° 43070

1. **Introdução**

O trabalho desenvolvido trata-se de um programa que permite efetuar a gestão do estoque de uma empresa de comércio de automóveis usados, sendo utilizados a linguagem de comandos Bash e awk, bem como o comando dialog. Neste programa é possível adicionar estoque, remover estoque, fazer alterações nos dados dos veículos, visualizar os automóveis por critério de organização, gerar relatórios referentes ao estoque e também criar, restaurar e apagar cópias de segurança do ficheiro onde as informações dos veículos estão armazenadas.

O trabalho foi realizado no âmbito da cadeira de Linguagens de Scripting e tem como objetivo o desenvolvimento dos conhecimentos da linguagem e comandos do BASH Shell, bem como outros assuntos e conteúdos apresentados em aula.

1. **Funcionalidades**
   1. **Menu Principal**

**Graphical user interface, text, application

Description automatically generated**

O menu principal foi desenvolvido em uma função chamada menuPrincipal, sendo a única função executada ao inicializar o programa. Foi utilizado o comando --menu do dialog para a apresentação das opções, bem como o comando --stdout para obter o input do utilizador. Para a execução das opções, foi utilizado o comando case; Em cada opção do comando está a função que é executada quando o utilizador seleciona o item desejado.

* 1. **Compra**

**Graphical user interface, application, Teams

Description automatically generated Graphical user interface, application, Teams

Description automatically generated** Graphical user interface, application, Teams

Description automatically generatedGraphical user interface, application

Description automatically generated

Na opção de compra é solicitado ao utilizador introduzir a matrícula, marca, modelo, ano, tipo, preço e custo de restauro do veículo a ser comprado.

Para obter cada informação, foi utilizado o comando --inputbox, que permite ao utilizador introduzir texto, e o comando --stdout para guardar, na sua respectiva variável, a informação introduzida. Os comandos echo e >> foram utilizados para imprimir as variáveis no ficheiro da base de dados.

Ao terminar de introduzir as informações, foi utilizado o comando --yesno do dialog e um if/else para perguntar ao utilizador se deseja adicionar outro veículo, caso a resposta seja sim então a função de compra é executada novamente, caso contrário a função do menu principal é executada.

* 1. **Venda**

Para realizar as vendas, é inicialmente solicitado ao utilizador que introduza a matrícula do veículo que deseja vender, bem como o preço e data da venda. Com o comando grep, o veículo é localizado pela sua matricula no ficheiro de base de dados e suas informações são armazenadas em uma variável.

Com os comandos echo e >> as informações do veículo junto das variáveis que contêm as informações da venda são impressas em um ficheiro de base de dados de venda. O comando grep -v armazena as informações de todos os veículos da base de dados, exceto o que foi vendido, em um ficheiro temporário que será utilizado para substituir o ficheiro da base de dados com o comando mv.

* 1. **Atualizar preço de restauro**

**Graphical user interface, text, application

Description automatically generated Graphical user interface, text, application

Description automatically generated Graphical user interface, text, application, chat or text message

Description automatically generated**

Para alterar o preço de restauro, primeiramente é utilizado o comando –inputbox e –stdout do dialog em uma variável para solicitar e guardar a matrícula do veículo que deseja realizar a alteração. Depois, são utilizados os mesmos comandos em outra variável para solicitar o novo valor.

O comando grep e cut são utilizado para pesquisar a matrícula do veículo no ficheiro da base de dados e obter as outras informações referentes ao veículo (exceto o valor de restauro), que são guardadas em variáveis. Com o comando grep e a opção -v, são redirecionadas as informações dos outros veículos para um ficheiro temporário.

São utilizados os comandos echo e >> para então imprimir no ficheiro temporário as variáveis com as informações do veículo, juntamente da variável que contém o novo valor de restauro.

Com o comando mv é feita a substituição do ficheiro da base de dados (que ainda contém o veículo com o preço de restauro antigo) pelo ficheiro temporário (que contém o veículo com o preço de restauro atualizado).

* 1. **Alterar dados**

Na opção de alterar dados, é apresentado um menu que permite escolher qual informação deseja alterar. Assim como no menu principal, foi utilizado o comando --menu do dialog para a apresentação das opções,e o comando --stdout para obter o input do utilizador. Para executar as opções, foi utilizado o comando case; Em cada opção do comando está a função que é executada quando o utilizador seleciona o item desejado.

A alteração dos dados é feita de forma semelhante a alteração do preço de restauro. Neste caso, para cada opção de alteração, existe uma função diferente.

Primeiramente, é solicitada a matrícula do veículo que deseja realizar alterações através dos comandos --inputbox e --stdout do dialog em uma variável. É utilizada esta mesma maneira para solicitar e armazenar a nova informação.

Os comandos grep e cut são utilizados em pipe para pesquisar a matrícula no ficheiro de base de dados e extrair e guardar em variáveis as outras informações acerca do veículo. Novamente com o comando grep e a opção -v, são redirecionados todas as informações dos outros veículos para um ficheiro temporário. Este ficheiro temporário receberá através dos comandos echo e >> as variáveis que contém as informações do veículo que foi realizado alteração, juntamente da variável que contém a nova informação.

O comando mv é utilizado para substituir o ficheiro da base de dados com as informações antigas pelo ficheiro temporário que contém as informações atualizadas.

* 1. **Visualizar automóveis**

Graphical user interface, application

Description automatically generated

Ao selecionar a opção de visualizar automóveis, é apresentado um menu onde o utilizador poderá escolher qual o critério de organização deseja, ou seja, pela matricula, marca, modelo, ano ou tipo.

Assim como no menu principal, foi utilizado o comando --menu do dialog para apresentar as opções e o comando --stdout para obter o input do utilizador. Para a execução das opções também foi utilizado o comando case, contendo a função de visualização relacionada em cada opção.

Em cada função de visualização, foi utilizado o comando sort no ficheiro da base de dados (caso seja por tipo, o comando sera executado no ficheiro dos tipos de automóveis), utilizando as opções -t para especificar o separador e a opção -k para especificar a coluna. Este comando é guardado em uma variável que será utilizada no comando --msgbox do dialog para imprimir os veículos organizados conforme a opção escolhida.

Depois de concluir, o programa volta para o menu de visualização.

* 1. **Gestão de base de dados**
  2. **Relatórios**
     1. **Veículos Vendidos**

Para o relatório dos veículos vendidos foi utilizado o comando cat no ficheiro da base de dados de vendas em uma variável. Esta variável é então utilizada no comando --msgbox do dialog.

* + 1. **Veiculos em Stock**

Utilizado o comando cat no ficheiro de base de dados e armazenado em uma variável que é utilizada no --msgbox do dialog.

* + 1. **Numero de Veiculos em Stock**

Foi utilizado o comando grep com a opção -c no ficheiro de base de dados.

* + 1. **Numero de Veiculos Vendidos**

Foi utilizado o comando grep com a opção -c no ficheiro de base de dados de venda.

* + 1. **Veiculo mais antigo em Stock**

Foi utilizado o comando head com a opção -n no ficheiro de base de dados.

* + 1. **Total lucro**

Para o calculo do lucro, foi utilizado o comando cut para extrair as colunas que contêm o valor de venda, compra e restauro. Os valores em cada coluna são então somados em um for loop, a soma dos valores de compra são adicionados com a soma dos valores de restauro, esta soma é então subtraída da soma dos valores de venda e armazenados em uma variável que é utilizada em uma –msgbox do dialog.