

Menu: o método utilizado (switch-case) torna o menu mais eficiente testando apenas a opção pretendida. A estrutura de controlo utilizada (while) é usada para percorrer o menu até pretendermos sair do programa (Opção 5 (sair)).

Opção 1(Contar as palavras): para esta opção foi criado vários *Arrays*: um para conter apenas o texto só com as palavras (*textoPalavras*), outro que contem apenas os algarismos (*textoNumeros*), outro para juntar estes dois (base) (*otexto*), outro para todos os caracteres que podem existir (pontuação) e outro para os números que podem existir (*num*). Seguidamente houve a criação de um ciclo *for* para percorrer todas as pontuações já definidas anteriormente e, utilizando o *replace*, uma daquelas pontuações que podem existir são substituídas por um vazio, contrariamente o *replaceAll* troca os espaçamentos múltiplos por apenas um espaçamento. O método *split* vai percorrer a frase até encontrar um espaço, no espaço encontrado a palavra que antecede a esse espaço é colocada no *array* base e assim sucessivamente. Após isto é criado um ciclo *for* cujo seu objetivo é percorrer o *array* dos números e dentro deste ciclo é criado outro para percorrer o texto (base) para contar os números dos algarismos através de um incremento. Depois temos de criar uma variável que nos vai indicar se o elemento do *array* é ou não um numero, mas para isso é necessário criar um *array* com o tamanho que cada palavra deve ter. Uma vez que já se sabe o tamanho, este ciclo, vai servir para percorrer todas as palavras (elementos) do *array* se neste ciclo se achou a palavra a variável é alterada para falsa. Agora, para percorrer os números de 0 a 9 é criado outro ciclo *for* e se a primeira letra for um número é porque não é uma palavra, se não achou um número é porque é uma palavra.

Opção 2(Frequências Globais): nesta opção pretende-se a frequência absoluta e a frequência relativa. Deste modo criou-se uma função (*fAbsolutaRelativa*) onde dentro desta criou-se um *array* (*listaPalavras*) que contem todas as palavras sem repetição onde se determinou o seu tamanho. Seguidamente criou-se uma variável (*numeroPalavras*) que determina o numero de vezes que cada palavra aparece na frase. Criou-se um ciclo *for* para percorrer cada elemento do *array* criado anteriormente. Para verificar se uma palavra já tinha sido contada ou não criou-se uma variável *boolean*. Após esta variável criou-se novamente um ciclo *for* para percorrer o texto da frase. A variável *counter* criada foi com o intuito de contar o numero de vezes que cada variável aparecia. De seguida cria-se outro ciclo *for* para percorrer o *array* (*listaPalavras*) e dentro de este ciclo cria-se um *if* para verificar se cada elemento que estivermos a ver está na lista, se estiver é porque este existe, se não existe então vamos adicionar á lista. A criação de outro *for* é fundamental para percorrer de novo o *array* (*otexto*) e verificar se encontrou de novo para contar quantas vezes o elemento aparece. Após isto vai armazenar a contagem no *array* (*numeroPalavras*) e ir a cada elemento desse *array*. Para imprimir o resultado criou-se outro ciclo *for* onde se calculou a percentagem de cada um dos elementos e se essa percentagem é igual a zero então não se deve mostrar nada se não mostra-se. No final é chamado a função pausa para efetuar uma pausa e fazer com que apareça “Prima enter para continuar...”

Opção 3(Palindromes e Capicuas): nesta opção criou-se um ciclo *for* que serviu para percorrer cada palavra do texto principal. Criou-se também uma estrutura *if* para determinar se uma palavra é uma palindrome ou uma capicua. Posteriormente cria-se uma função onde dentro desta criam-se variáveis (*len/half*) onde a variável (*len*) determina o tamanho do *array* escolhido ou atual e a variável "*half*" divide este mesmo *array* em metade para facilitar o encontro se existir palindromes ou capicuas.

O ciclo *for* criado depois das variáveis mencionadas acima serviu para analisar apenas metade do elemento do *array*, seguidamente o ciclo *if* é criado de modo a comparar os caracteres do *array* escolhido na variável *len* com a outra metade de caracteres deste mesmo *array*.

Opção 4 (Ordenar Palavras): nesta opção foi utilizado o algoritmo de ordenação *ShellSort* para ordenar alfabeticamente as letras que constituíam cada palavra da frase dado no código. A utilização deste algoritmo consistiu em dividir a frase em várias partes e em cada parte ordenar as letras alfabeticamente de modo a que cada palavra ficasse ordenada. Criou-se um *array* (*letra*) onde posteriormente se dividiu a frase em partes e cada parte digamos que ficou armazena cada uma numa parte diferente. Criou-se uma variável (*intervalo*) onde cada palavra da frase que tinha sido armazena em partes diferentes no *array* para dividir cada palavra em duas partes ordenando cada caracter da mesma por ordem alfabética.

O *array* (*troca*) foi criado com o intuito de trocar a ordem de cada caracter de cada parte composto por uma palavra anteriormente dividida através do *array* (*letra*) de modo a que no final seja imprimida a frase onde em os caracteres de cada palavra fossem ordenados alfabeticamente.