

TRABALHO PRÁTICO 3

REALIZADO POR:
TATIANA DAMAYA (A50299)

ÍNDICE

PÁG 1

CAPA

PÁG 2

ÍNDICE

PÁG 3

INTRODUÇÃO

PÁG 4 - PÁG 14

**| PROGRAMAS COM UTILIZAÇÃO DE
MÉTODOS**

PÁG 15 - PÁG 21

**| UTILIZAÇÃO DE MÉTODOS
RECURSIVOS**

PÁG 22 - PÁG 33

| LINHA DE COMANDOS

PÁG 34

CONCLUSÃO

INTRODUÇÃO

"Um programa consiste num conjunto de instruções, para um computador seguir, para executar uma determinada tarefa".

Após a aprendizagem da criação de programas utilizando a linguagem de programação *Java* no editor/executor *Eclipse*, passámos à introdução de métodos estáticos na primeira parte deste trabalho, métodos recursivos na segunda parte e por último à introdução de linha de comandos, a qual podemos por em prática na resolução dos 15 exercícios deste último trabalho de (TI).

Durante este trabalho são usados em praticamente todos os códigos métodos como:

- `System.out` - permite escrever na consola.
- `Scanner keyboard = new Scanner(System.in)` - permite a leitura da consola
- `keyboard.nextInt()` - permite ler um inteiro da consola.
- `keyboard.nextLine()` - permite ler string da consola.
- `keyboard.close()` - permite fechar a ligação ao teclado.

Após as aprendizagens durante as aulas de TI, no IDE *Eclipse* e na Linha de comandos, ficámos devidamente autónomos para a realização dos programas pedidos no enunciado deste trabalho prático.

04

PROGRAMA 1

1. Crie uma classe (TP3-01-SomaDigitos) que implemente o método `int somaDigitos(int n)`. Este método deverá receber um número `n` e retornar a soma de todos os seus dígitos. Ex: `somaDigitos(1234)` deverá retornar o número 10.

EXPLICAÇÃO:

Para a implementação do método `int somaDigitos(int n)` comecei pela inicialização de uma nova variável `int soma`, que através de um ciclo "while" irá receber o resto da divisão do número introduzido pelo utilizador "int n" por 10, e somando-o com o seguinte, durante todo o ciclo, até que `n` deixe de ser maior que 0. Ainda dentro deste, a variável `n` está também a ser dividida por 10, para que esse último dígito, seja retirado a cada ciclo, após já ter sido adicionado à soma, isto é possível visto que a variável só recebe números inteiros. Após a conclusão do ciclo o método irá retornar a soma de todos os dígitos do número que o utilizador introduziu e será enviada para a consola através do método `main`.

05

TP3-01-SOMADIGITOS

```
TP3_05_Silabas.java  TP3_01_SomaDigitos.java X
1 package TI_TP3_A50299;
2 import java.util.Scanner;
3
4 public class TP3_01_SomaDigitos {
5
6     public static void main(String[] args) {
7         /** 1. Crie uma classe (TP3-01-SomaDigitos) que implemente o método int somaDigitos(int n).
8          * Este método deverá receber um número n e retornar a soma de todos os seus dígitos.
9          * Ex: somaDigitos(1234) deverá retornar o número 10.
10          */
11
12         System.out.println("...:TP3-01-SomaDigitos:...");
13         int num;
14         System.out.println("Introduza um número:");
15         Scanner keyboard = new Scanner(System.in);
16         num = keyboard.nextInt();
17         System.out.println("A soma de todos os dígitos do número [" + num + "] é: " + somaDigitos(num));
18         keyboard.close();
19     }
20
21     public static int somaDigitos(int n) {
22         int soma = 0;
23         while (n > 0) {
24             soma += (n % 10);
25             n = n / 10;
26         }
27         return soma;
28     }
29 }
```

```
Console X
<terminated> TP3_01_SomaDigitos [Java Application] /Users/tatianadamaya/.p2/pool/plugins/or
...:TP3-01-SomaDigitos:...
Introduza um número:
1234
A soma de todos os dígitos do número [1234] é: 10
```

EXECUÇÃO DO PROGRAMA

06

PROGRAMA 2

2. Crie uma classe (TP3-02-Arrays) que implemente os seguintes métodos: `int max(int[] array)` que retorna o valor maior de um array; `int min(int[] array)` que retorna o valor menor do array; `int sum(int[] array)` que retorna a soma de todos os números do array; `double avg(int[] array)` que retorna o valor médio do array.

EXPLICAÇÃO:

Para a implementação do método `int max(int[] array)`, comecei pela inicialização de uma nova variável `"int max"`, à qual associei o primeiro índice do array, que através de um ciclo `"for"` irá ser comparado a todos os índices até que a `length` do array acabe. Caso encontre um valor maior dentro do array, a variável `max` irá ser igualada a esse novo índice, através do operador `if`. Quando o ciclo acabar, a comparação irá terminar com o valor maior do array contido na variável `max`, a qual o método irá retornar.

Para o `int min(int[] array)`, as instruções serão as mesmas que o método `int max(int[] array)`, apenas troquei o nome da variável para `"min"` e no ciclo serão comparados os índices aos de menor valor.

Para o método `int sum(int[] array)`, após a inicialização da variável `sum`, através de um ciclo `for`, iremos somar cada índice do array ao seguinte, até que a `length` do array acabe. Após o ciclo acabar a variável `sum` será retornada. Para o método `double avg(int[] array)` usei uma variável `double "media"` que chama o método `int sum(int[] array)` e o divide pelo `length` do array, para assim termos a divisão da soma do array pela quantidade de valores existentes neste, calculando a média.

Após a conclusão dos métodos obteremos então o valor maior, menor, a soma e a média do array, que serão enviados para a consola ao chamar os métodos correspondentes no método `main`, onde para a interação com o utilizador, criei `system.out's` que perguntam ao utilizador qual o tamanho do array que deseja e os valores que quer introduzir neste, sendo assim este a criá-lo.

07

TP3-02-ARRAYS

```
TP3_02_Arrays.java X
1 package TI_TP3_A50299;
2 import java.util.Arrays;
3
4
5 public class TP3_02_Arrays {
6
7     public static void main(String[] args) {
8         /** 2. Crie uma classe (TP3-02-Arrays) que implemente os seguintes métodos:
9          * int max(int[] array) que retorna o valor maior de um array;
10         * int min(int[] array) que retorna o valor menor do array;
11         * int sum(int[] array) que retorna a soma de todos os números do array;
12         * double avg(int[] array) que retorna o valor médio do array.
13         */
14
15         System.out.println("...:TP3-02-Arrays:...");
16
17         int n;
18
19         Scanner keyboard = new Scanner(System.in);
20
21         //criar array do utilizador
22         System.out.println("Introduza um valor para o número de posições do array:");
23
24         n = keyboard.nextInt();
25         int[] arr = new int[n];
26
27         System.out.println("Introduza " + n + " valores para as posições do array:");
28
29         for (int i = 0; i < arr.length; i++) {
30             arr[i] = keyboard.nextInt();
31         }
32         System.out.println("O Array formado foi : " + Arrays.toString(arr));
33         System.out.println("O número maior do Array é : " + max(arr));
34         System.out.println("O número menor do Array é : " + min(arr));
35         System.out.println("A soma do Array é : " + sum(arr));
36         System.out.println("A média do Array é : " + avg(arr));
37         keyboard.close();
38     }
39 }
```

```
//maior
public static int max(int[] array) {
    int max = array[0];
    for (int i = 0; i < array.length; i++) {
        if(array[i] > max) {
            max = array[i];
        }
    }
    return max;
}

//menor
public static int min(int[] array) {
    int min = array[0];
    for (int i = 0; i < array.length; i++) {
        if(array[i] < min) {
            min = array[i];
        }
    }
    return min;
}

//soma
public static int sum(int[] array) {
    int sum = 0;
    for (int i = 0; i < array.length; i++) {
        sum += array[i];
    }
    return sum;
}

//valor médio
public static double avg(int[] array) {
    double media = sum(array)/array.length;
    return media;
}
}
```

```
Console X
<terminated> TP3_02_Arrays [Java Application] /Users/tatianadamaya/.p2/pool/plugins/org.eclipse.justj.open
...:TP3-02-Arrays:...
Introduza um valor para o número de posições do array:
4
Introduza 4 valores para as posições do array:
4
7
2
9
O Array formado foi : [4, 7, 2, 9]
O número maior do Array é : 9
O número menor do Array é : 2
A soma do Array é : 22
A média do Array é : 5.0
```

EXECUÇÃO DO PROGRAMA

08

PROGRAMA 3

EXPLICAÇÃO:

09

EXECUÇÃO DO PROGRAMA

10

PROGRAMA 4

4. Crie um programa (TP3-04-Password) que peça uma password ao utilizador e verifique se é válida. Uma password válida deverá conter pelo menos 10 caracteres (letras e números) onde, no mínimo, dois caracteres são numéricos e três caracteres são letras. Para tal deverá criar os seguintes métodos: boolean `isLetter(char ch)` que verifica se o caracter `ch` é uma letra ("a-z", "A-Z"); boolean `isDigit(char ch)` que verifica se o caracter `ch` é um dígito numérico; boolean `isValid(String password)` que verifica se a password é válida ou não consoante as regras mencionadas.

EXPLICAÇÃO:

Para a implementação do método boolean `isDigit(char ch)`, usei um operador `if` que contém os intervalos das diferentes letras (maiúsculas e minúsculas) segundo a tabela de ASCII, isto para verificar se os caracteres introduzidos pertencem a este, retornando um boolean `true` se forem letras e `false` se não pertencerem.

Para o boolean `isDigit(char ch)`, as instruções serão as mesmas que o método boolean `isLetter(char ch)`, apenas troquei o intervalo de valores para os dos dígitos numéricos da tabela de ASCII.

Para o método boolean `isValid(String password)`, inicializei a variável boolean `isValid`, e consoante as regras mencionadas, para verificar se a password introduzida tem pelo menos 10 caracteres, usei um operador `if`, que verifica se a `length` desta é maior ou igual a 10, se sim passa para a próxima confirmação se não, o método irá retornar imediatamente o booleano `false`.

Para as restantes confirmações usei um ciclo `for` que passa por todos os caracteres da password introduzida através do `char ch = password.charAt(i)` que apanha cada índice da palavra, e que através de operadores `if` que chamam os métodos `isLetter()` e `isDigit()` irá com as variáveis `int countLetter` e `countDigit` contar a quantidade de letras e números que existem na palavra. Para a última confirmação um dos operadores `if` irá retornar se a password é válida ou não, confirmando se a contagem de letras é igual ou superior a 3 e de números maior ou igual a 2.

Após a conclusão da verificação da password no método `main` recorri a um operador ternário que através de um `System.out.println` indica ao utilizador na consola se a password é válida ou não através da verificação do método `isValid != true?`.

11

TP3-04-PASSWORD

```
1 package TI_TP3_A50299;
2
3 import java.util.Scanner;
4
5 public class TP3_04_Password {
6
7     public static void main(String[] args) {
8         /** 4. Crie um programa (TP3-04-Password) que peça uma password ao utilizador e verifique se é válida.
9          * Uma password válida deverá conter pelo menos 10 caracteres (letras e números) onde, no mínimo, dois caracteres são numéricos e três caracteres são letras.
10          * Para tal deverá criar os seguintes métodos:
11          * boolean isLetter(char ch) que verifica se o caracter ch é uma letra ("a-z", "A-Z");
12          * boolean isDigit(char ch) que verifica se o caracter ch é um dígito numérico;
13          * boolean isValid(String password) que verifica se a password é válida ou não consoante as regras mencionadas.
14          */
15
16         System.out.println("...:TP3-04-Password:...");
17         String password;
18         System.out.println("Introduza uma password:");
19         Scanner keyboard = new Scanner(System.in);
20         password = keyboard.nextLine();
21         System.out.print("A sua password é " + (isValid(password) != true? "inválida!" : "válida!"));
22         keyboard.close();
23     }
24
25     public static boolean isLetter(char ch) {
26         if (ch >= 65 && ch <= 90 || ch >= 97 && ch <= 122) {
27             return true;
28         } else return false;
29     }
30
31     public static boolean isDigit(char ch) {
32         if (ch >= 48 && ch <= 57) {
33             return true;
34         } else return false;
35     }
36
37     public static boolean isValid(String password) {
38         boolean isValid = false;
39
40         if (password.length() >= 10) {
41             int countLetter = 0;
42             int countDigit = 0;
43
44             for(int i = 0; i < password.length(); i++) {
45                 char ch = password.charAt(i);
46
47                 if (isLetter(ch) == true) {
48                     countLetter++;
49                 }
50
51                 if (isDigit(ch) == true) {
52                     countDigit ++;
53                 }
54
55                 if (countLetter >= 3 && countDigit >= 2) {
56                     isValid = true;
57                 } else isValid = false;
58             }
59             return isValid;
60         }
61     }
62 }
```

```
Console X
<terminated> TP3_04_Password [Java Application] /Users/tatianadamaya/.p2/pool/plugins
...:TP3-04-Password:...
Introduza uma password:
Password1234
A sua password é válida!
```

```
Console X
<terminated> TP3_04_Password [Java Application] /Users/tatianadamaya/.p2/pool/plugins
...:TP3-04-Password:...
Introduza uma password:
P1234
A sua password é inválida!
```

EXECUÇÃO DO PROGRAMA

12

PROGRAMA 5

5. Crie um programa (TP3-05-Silabas) que conte o número de sílabas de uma palavra. O número de sílabas pode ser obtido (aproximadamente) pelo número de vogais não sucessivas existentes numa palavra. Por exemplo “valor” tem duas sílabas enquanto “biblioteca” tem 4 sílabas (note que “io” conta apenas uma vez). Use os seguintes métodos: boolean isVogal(char ch) que verifica se o caracter ch é uma vogal (“a, e, i, o, u”); int countVogais(String str) que retorna o número de vogais numa string; int countParesVogais(String str) que retorna o número de pares sucessivos de vogais; int silabas(String str) que calcula o número de sílabas (num. Vogais – num. Pares vogais).

EXPLICAÇÃO:

Para a implementação do método boolean isVogal(char ch), usei um operador if que contém os códigos das vogais (a,e,i,o,u) tanto em maiúsculas como minúsculas segundo a tabela de ASCII, isto para verificar se os caracteres introduzidos são vogais ou não, retornando true ou false.

Para o método int countVogais(String str), inicializei a variável int countVogais, a qual vai ser utilizada num ciclo for que percorre a length da string e através de um operador if que chama o método isVogal iremos contar o numero de vogais que existem na string e retornar esta contagem quando a length da string acabar.

Para int countParesVogais(String str) usamos também um ciclo for e um operador if que desta vez chama duas vezes o método isVogal() onde com a função str.charAt(i) e str.charAt(i+1) irá verificar se existe uma vogal seguida de outra dentro da string, caso isto seja true a variável countParesVogais irá somar 1 a cada par que houver e a sua contagem será retornada com o fim do ciclo for. Por último no int silabas(String str) usei então o cálculo da contagem de vogais menos a contagem de pares de vogais para retornar quantas silabas tem a palavra introduzida pelo utilizador.

13

TP3-05-SILABAS

```

1 package TI_TP3_A50299;
2
3 import java.util.Scanner;
4
5 public class TP3_05_Silabas {
6
7     public static void main(String[] args) {
8         /** 5. Crie um programa (TP3-05-Silabas) que conte o número de sílabas de uma palavra.
9          * 0 número de sílabas pode ser obtido (aproximadamente) pelo número de vogais não sucessivas existentes numa palavra.
10         * Por exemplo "valor" tem duas sílabas enquanto "biblioteca" tem 4 sílabas (note que "io" conta apenas uma vez).
11         * Use os seguintes métodos:
12         * boolean isVogal(char ch) que verifica se o caracter ch é uma vogal ("a", "e", "i", "o", "u");
13         * int countVogais(String str) que retorna o número de vogais numa string;
14         * int countParesVogais(String str) que retorna o número de pares sucessivos de vogais;
15         * int silabas(String str) que calcula o número de sílabas (num. Vogais - num. Pares vogais).
16         */
17
18         System.out.println("...:TP3-05-Silabas:...");
19         String palavra;
20         System.out.println("Introduza uma palavra:");
21         Scanner keyboard = new Scanner(System.in);
22         palavra = keyboard.nextLine();
23         System.out.print("A palavra " + palavra + " tem " + silabas(palavra) + " sílabas.");
24         keyboard.close();
25     }
26
27     public static boolean isVogal(char ch) {
28         if(ch == 97 || ch == 101 || ch == 105 || ch == 111 || ch == 117 || ch == 65 || ch == 69 || ch == 73 || ch == 79 || ch == 85) {
29             return true;
30         } else return false;
31     }
32
33     public static int countVogais(String str) {
34         int countVogais = 0;
35         for(int i = 0; i < str.length(); i++) {
36             if(isVogal(str.charAt(i)) == true) {
37                 countVogais++;
38             }
39         }
40         return countVogais;
41     }
42
43     public static int countParesVogais(String str) {
44         int countParesVogais = 0;
45         for(int i = 0; i < str.length() - 1; i++) {
46             if(isVogal(str.charAt(i)) == true && isVogal(str.charAt(i + 1)) == true) {
47                 countParesVogais++;
48             }
49         }
50         return countParesVogais;
51     }
52
53     public static int silabas(String str) {
54         int silabas = countVogais(str) - countParesVogais(str);
55         return silabas;
56     }
57 }

```

```

Console X
<terminated> TP3_05_Silabas [Java Application] /Users/tatianadamaya/.p2/pool/pli
...:TP3-05-Silabas:...
Introduza uma palavra:
biblioteca
A palavra biblioteca tem 4 sílabas.

```

```

...:TP3-05-Silabas:...
Introduza uma palavra:
valor
A palavra valor tem 2 sílabas.

```

EXECUÇÃO DO PROGRAMA

14

PROGRAMA 6

6. Crie um programa (TP3-06-Palindromo) que implemente o método recursivo boolean `palindromo(String str)`. Este método deverá devolver `true` caso a string seja um palíndromo, e `false` caso contrário. Um palíndromo é uma palavra que se lê da mesma forma da esquerda para a direita e ao contrário. Considere usar o método `str.substring(int a, int b)` que retorna uma sub-string entre os índices `a` e `b`.

EXPLICAÇÃO:

Para a implementação do método boolean `palindromo(String str)` comecei com a condição de paragem `str.length() == 1`, para que quando a palavra conter somente uma letra retornar que esta é um palindromo. Depois também temos a verificação de se caso a string tiver apenas 2 caracteres, verificar se o primeiro caractere é igual ao segundo e retornar `true`, caso isto se aplique. Para a verificação de uma string de maior comprimento, usamos o `str.charAt(0) == str.charAt(str.length()-1)` para verificar se o primeiro caracter da string é igual ao último e caso isto seja verdade, será aplicado a cada chamada do método com a diminuição da string a cada retorno usando `str.substring` que vai eliminando os caracteres inicial e último, após a confirmação de que estes são iguais. Caso estes não se aplicarem o metodo retornará no seu fim o boolean `false`.

15

TP3-06-PALINDROMO

```
TP3_06_Palindromo.java X
1 package TI_TP3_A50299;
2
3 import java.util.Scanner;
4
5 public class TP3_06_Palindromo {
6
7     public static void main(String[] args) {
8         /** 6. Crie um programa (TP3-06-Palindromo) que implemente o método recursivo boolean palindromo(String str).
9          * Este método deverá devolver true caso a string seja um palíndromo, e false caso contrário.
10         * Um palíndromo é uma palavra que se lê da mesma forma da esquerda para a direita e ao contrário.
11         * Considere usar o método str.substring(int a, int b) que retorna uma sub-string entre os índices a e b.
12         */
13
14         System.out.println("...:TP3-06-Palindromo:...");
15         String palavra;
16         System.out.println("Introduza uma palavra:");
17         Scanner keyboard = new Scanner(System.in);
18         palavra = keyboard.nextLine();
19         System.out.print("A palavra [" + palavra + (palindromo(palavra) != true? "]" não é um palíndromo!" : "] é um palíndromo!");
20         keyboard.close();
21     }
22
23     public static boolean palindromo(String str) {
24         if(str.length() == 1) return true;
25         else if (str.length() == 2 && str.charAt(0) == str.charAt(1)) return true;
26         else if(str.charAt(0) == str.charAt(str.length() - 1)) return palindromo(str.substring(1, str.length() - 1));
27         return false;
28     }
29 }
30 }
```

EXECUÇÃO DO PROGRAMA

Console X	Console X
<terminated> TP3_06_Palindromo [Java Application] /Users/tatianadama	<terminated> TP3_06_Palindromo [Java Application] /Users/tatianadamaya/
...:TP3-06-Palindromo:...	...:TP3-06-Palindromo:...
Introduza uma palavra:	Introduza uma palavra:
mata	ama
A palavra [mata] não é um palíndromo!	A palavra [ama] é um palíndromo!

16

PROGRAMA 7

7. Crie o programa (TP3-07-MaioresQue) que implemente o método recursivo `int maioresQue(int[] array, int val)`. Este método deverá retornar quantos números no array são maiores do que o número `val`. Por exemplo, para o array `{1, 2, 3, 4, 5}` existem 2 números maiores do que o 3 (i.e., o 4 e o 5). Considere usar o método `Arrays.copyOfRange(int[] original, int from, int to)` que retorna uma cópia do array original entre os índices `from` e `to`.

EXPLICAÇÃO:

Para a implementação do método recursivo `int maioresQue(int[] array, int val)` comecei com a condição de paragem `array.length == 0`, pois após o método `Arrays.copyOfRange(array, 1, array.length)` que usei a seguir o tamanho do array original vai sempre diminuindo a cada chamada, isto para que todos os valores do array sejam percorridos e comparados ao valor introduzido pelo utilizador (`val`), através de `array[0]`, que apanha o primeiro número de cada cópia do array e o compara com `val` através de um operador `if`. O método acaba então quando a `length` do array for 0 e já todos os valores tiverem sido comparados, sendo retornado o `count` que foi somando cada encontro de um número maior do que o `val`.

17

```
TP3_06_Palindromo.java  TP3_05_Silabas.java  TP3_07_MaioresQue.java  TP3_02_Arrays.java
1 package TI_TP3_A50299;
2
3 import java.util.Arrays;
4 import java.util.Scanner;
5
6 public class TP3_07_MaioresQue {
7     public static void main(String[] args) {
8         /** 7. Crie o programa (TP3-07-MaioresQue) que implemente o método recursivo int maioresQue(int[] array, int val).
9          * Este método deverá retornar quantos números no array são maiores do que o número val.
10         * Por exemplo, para o array {1, 2, 3, 4, 5} existem 2 números maiores do que o 3 (i.e., o 4 e o 5).
11         * Considere usar o método Arrays.copyOfRange(int[] original, int from, int to) que retorna uma cópia do array original entre os índices from e to.
12         */
13
14         System.out.println("...:TP3-07-MaioresQue:...");
15
16         int n, val;
17
18         Scanner keyboard = new Scanner(System.in);
19
20         //criar array do utilizador
21         System.out.println("Introduza um valor para o número de posições do array:");
22         n = keyboard.nextInt();
23         int[] arr = new int[n];
24         System.out.println("Introduza " + n + " valores para as posições do array:");
25         for (int i = 0; i < arr.length; i++) {
26             arr[i] = keyboard.nextInt();
27         }
28         System.out.println("Introduza o valor a ser comparado:");
29         val = keyboard.nextInt();
30
31         System.out.println("O Array formado foi: " + Arrays.toString(arr));
32         System.out.println("Existem " + maioresQue(arr, val) + " números no array que são maiores do que o número " + val + "!");
33
34         keyboard.close();
35     }
36
37     public static int maioresQue(int[] array, int val) {
38         if (array.length == 0) return 0;
39         int count = maioresQue(Arrays.copyOfRange(array, 1, array.length), val);
40         if (array[0] > val) count++;
41         return count;
42     }
43 }
```

```
Console X
<terminated> TP3_07_MaioresQue [Java Application] /Users/tatianadamaya/.p2/pool/plugins/org.eclipse.justj.openjdk.hotsp
...:TP3-07-MaioresQue:...
Introduza um valor para o número de posições do array:
5
Introduza 5 valores para as posições do array:
1
2
3
4
5
Introduza o valor a ser comparado:
3
O Array formado foi: [1, 2, 3, 4, 5]
Existem 2 números no array que são maiores do que o número 3!
```

EXECUÇÃO DO PROGRAMA

18

PROGRAMA 8

8. Crie o programa (TP3-08-RemoveEspacosConsecutivos) que implemente o método recursivo `String removeEspacosConsecutivos(String str)`. Este método deverá retornar a string original sem os espaços consecutivos. Por exemplo, para a string “ab cd” o método deverá retornar a string “ab cd”.

EXPLICAÇÃO:

Para a implementação do método recursivo `String removeEspacosConsecutivos(String str)`, comecei com a condição de paragem `str.length() == 0`. A seguir para remover os espaços usei o operador `if` com `(str.charAt(0) == ' ' && str.charAt(1) == ' ')`, para que com o percorrer da string, que é retornada em cada chamada sem os caracteres já verificados através de uma substring, podermos percorrer toda a palavra e perceber se o primeiro índice é um espaço e se o seguinte também. O método acaba então quando a `length` da string for 0 e já todos os caracteres tiverem sido verificados, eliminando qualquer espaço que esteja seguido de outro.

19

TP3-08-REMOVEESPACOSCONSECUTIVOS

```
TP3_06_Palindromo.java TP3_05_Silabas.java TP3_07_MaioresQue.java *TP3_08_RemoveEspacosConsecutivos.java X
1 package TI_TP3_A50299;
2
3 import java.util.Scanner;
4
5 public class TP3_08_RemoveEspacosConsecutivos {
6
7     public static void main(String[] args) {
8         /** 8. Crie o programa (TP3-08-RemoveEspacosConsecutivos) que implemente o método recursivo String removeEspacosConsecutivos
9          * Este método deverá retornar a string original sem os espaços consecutivos.
10          * Por exemplo, para a string "ab cd" o método deverá retornar a string "ab cd".
11          */
12
13         System.out.println("...:TP3-08-RemoveEspacosConsecutivos:...");
14         String palavra;
15         System.out.println("Introduza uma palavra:");
16         Scanner keyboard = new Scanner(System.in);
17         palavra = keyboard.nextLine();
18         System.out.print(removeEspacosConsecutivos(palavra));
19         keyboard.close();
20     }
21
22     public static String removeEspacosConsecutivos(String str) {
23         if(str.length() == 0) return "";
24         if(str.charAt(0) == ' ' && str.charAt(1) == ' ') return removeEspacosConsecutivos(str.substring(1));
25         return str.charAt(0) + removeEspacosConsecutivos(str.substring(1));
26     }
27 }
28 }
```

EXECUÇÃO DO PROGRAMA

```
Console X
<terminated> TP3_08_RemoveEspacosConsecutivos [Java Application] /Users/tatianadar...
...:TP3-08-RemoveEspacosConsecutivos:...
Introduza uma palavra:
ab cd
ab cd
```

20

PROGRAMA 9

9. Crie o programa (TP3-09-SilabasRecurativas) que reimplente o método `int silabas(String str)` de forma recursiva. Tenha atenção que duas vogais seguidas devem contar apenas como uma sílaba. Utilize o método boolean `isVogal(char ch)` para facilitar a lógica do programa.

EXPLICAÇÃO:

Após a adição do método boolean `isVogal`, que através da verificação de um caractere detecta se este é uma vogal ou não. Para a implementação do método recursivo método `int silabas(String str)`, comecei com a condição de paragem `str.length() == 1`. A seguir para verificar as vogais usei a variável `count` que conta as vogais existentes e se existe uma vogal a seguir a outra, para que com o percorrer da string, que é retornada em cada chamada sem os caracteres já verificados, através de uma substring, possibilitando percorrer toda a palavra e retornar no fim a conta de silabas efetuada.

21

TP3-09-SILABASRECURSIVAS

```
TP3_09_SilabasRecursivas.java X
1 package TI_TP3_A50299;
2
3 import java.util.Scanner;
4
5 public class TP3_09_SilabasRecursivas {
6
7     public static void main(String[] args) {
8         /** 9. Crie o programa (TP3-09-SilabasRecursivas) que reimplente o método int silabas(String str) de forma recursiva.
9          * Tenha atenção que duas vogais seguidas devem contar apenas como uma sílaba.
10         * Utilize o método boolean isVogal(char ch) para facilitar a lógica do programa.
11         */
12         System.out.println("...:TP3-09-SilabasRecursivas:...");
13         String palavra;
14         System.out.println("Introduza uma palavra:");
15         Scanner keyboard = new Scanner(System.in);
16         palavra = keyboard.nextLine();
17         System.out.print("A palavra [" + palavra + "] tem " + silabas(palavra) + " silabas!");
18         keyboard.close();
19     }
20
21     public static int silabas(String str) {
22         if (str.length() == 1) return 0;
23         int count = silabas(str.substring(1));
24         if(isVogal(str.charAt(0)) == true && isVogal(str.charAt(1)) == false) count ++;
25         if(isVogal(str.charAt(0)) == true && isVogal(str.charAt(1)) == true) count ++;
26         return count;
27     }
28
29     public static boolean isVogal(char ch) {
30         if(ch == 97 || ch == 101 || ch == 105 || ch == 111 || ch == 117 || ch == 65 || ch == 69 || ch == 73 || ch == 79 || ch == 85) {
31             return true;
32         } else return false;
33     }
34 }
35 }
```

```
Console X
<terminated> TP3_09_SilabasRecursivas [Java Application] /Users/tatianadamaya/.p2/po
...:TP3-09-SilabasRecursivas:...
Introduza uma palavra:
biblioteca
A palavra [biblioteca] tem 4 silabas!
```

```
Console X
<terminated> TP3_09_SilabasRecursivas [Java Application] /Users/tatianadamaya/.p2/po
...:TP3-09-SilabasRecursivas:...
Introduza uma palavra:
valor
A palavra [valor] tem 2 silabas!
```

EXECUÇÃO DO PROGRAMA

22

PROGRAMA 10

10. Crie o programa (TP3-10-Normalizar) que implemente o método `String normalizar(String str)` para normalizar nomes de uma forma recursiva. Para a resolução do problema deverá assumir que qualquer nome a seguir a um (ou mais) espaços deverá começar por um carácter maiúsculo seguido de vários caracteres minúsculos. Ignore os casos especiais como “de”, “das”, “e”, etc. Como exemplo, o nome “aníbal santos” (com um ou mais espaços entre os dois nomes) deverá retornar “Aníbal Santos”. Note que poderá ter que incluir um espaço antes da primeira palavra para que o algoritmo recursivo a consiga detectar e normalizar.

EXPLICAÇÃO:

Para a implementação do método recursivo `String normalizar(String str)`, comecei com a condição de paragem `str.length() == 0`. A seguir criei uma `String c` para que esta através de uma substring se iguale a cada caractere do nome que o utilizador introduziu.

Para alterar cada caractere em maiúscula ou minúscula, usei os operadores `if`, no qual um verifica se o caractere é um espaço, caso isto aconteça, este irá retornar a substring entre o caracter e o a seguir para maiuscula e retornar uma nova string a partir desse caractere.

Caso o caractere não seja um espaço, este vai ser minúsculo e a string irá ser diminuida a cada chamada até que a `length` da palavra acabe acabe.

23

TP3-10-NORMALIZAR

```
TP3_03_EscreveDigit... TP3_02_Arrays.java TP3_06_Palindromo... TP3_01_SomaDigitos.j... TP3_09_SilabasRec... TP3_08_RemoveEspa...
1 package TI_TP3_A50299;
2
3 import java.util.Scanner;
4 public class TP3_10_Normalizar {
5     /**Crie o programa (TP3-10-Normalizar) que implemente o método String normalizar(String str) para normalizar nomes de uma forma
6      * Para a resolução do problema deverá assumir que qualquer nome a seguir a um (ou mais) espaços deverá começar por um
7      * Ignore os casos especiais como "de", "das", "e", etc.
8      * Como exemplo, o nome "anibal santos" (com um ou mais espaços entre os dois nomes) deverá retornar "Anibal Santos".
9      * Note que poderá ter que incluir um espaço antes da primeira palavra para que o algoritmo recursivo a consiga detectar
10     */
11     public static void main(String[] args) {
12         System.out.println("...:TP3-10-Normalizar:...");
13         String palavra;
14         System.out.println("Introduza um nome:");
15         Scanner keyboard = new Scanner(System.in);
16         palavra = keyboard.nextLine();
17         System.out.print("O nome normalizado fica:");
18         System.out.print(" " + normalizar(palavra));
19         keyboard.close();
20     }
21
22     public static String normalizar(String str) {
23         if (str.length() == 0) return "";
24         String c = str.substring(0, 1);
25         if (str.length() > 0 && c.equals(" ")) return str.substring(0,2).toUpperCase() + normalizar(str.substring(2)) + "";
26         if (str.length() > 0) return c.toLowerCase() + normalizar(str.substring(1));
27         return "";
28     }
29 }
30 }
```

```
Console X
<terminated> TP3_10_Normalizar [Java Application] /Users/tatianadamaya/.p
...:TP3-10-Normalizar:...
Introduza um nome:
  anibal santos
O nome normalizado fica: Anibal Santos
```

EXECUÇÃO DO PROGRAMA

24

EXERCÍCIO 11

11. Navegue até à pasta inicial (C:\TP3CMD) e crie os seguintes directórios e sub-directórios:


1. • América • Brasil
2. • Canada
3. • EUA • Ásia
4. • China
5. • India • Europa
6. • Espanha • França
7. • Portugal

EXPLICAÇÃO:

Após navegar com o comando `cd ..`, criei a pasta TP3CMD através do comando `mkdir`, e entrei nela com o comando `cd` criando novamente com o comando `mkdir` os directórios e sub-directórios descritos no exercício, verificando se estes foram criadas corretamente com o comando `dir` e navegando entre as pastas com os comando `cd` para entrar nelas e `cd ..` para sair.

25

EXERCÍCIO 11

 Select Command Prompt

Microsoft Windows [Version 10.0.19044.2364]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nunoh>cd ..

C:\Users>cd ..

C:\>mkdir TP3CMD

C:\>dir

Volume in drive C has no label.
Volume Serial Number is 6027-6E67

Directory of C:\

04/12/2022	21:14	<DIR>	computationalsoftwares
25/05/2021	07:03	<DIR>	DOS-GAMES
05/08/2021	13:25	<DIR>	Drivers
24/05/2021	22:21	<DIR>	Intel
11/06/2022	22:37	<DIR>	MinGW
22/09/2005	23:39		894,976 msdia80.dll
14/12/2022	12:37	<DIR>	Program Files
02/09/2022	10:35	<DIR>	Program Files (x86)
05/11/2022	00:30	<DIR>	Python38
13/03/2022	04:32	<DIR>	Recovery
15/12/2019	10:06	<DIR>	Sandbox
28/12/2022	14:44	<DIR>	TP3CMD
02/09/2022	16:07	<DIR>	Users
28/12/2022	14:03	<DIR>	WINDOWS
		1 File(s)	894,976 bytes
		13 Dir(s)	13,657,448,448 bytes free

C:\>cd TP3CMD

C:\TP3CMD>mkdir América

C:\TP3CMD>cd América

C:\TP3CMD\América>mkdir Brasil

C:\TP3CMD\América>mkdir Canada

C:\TP3CMD\América>mkdir EUA

C:\TP3CMD\América>dir

Volume in drive C has no label.
Volume Serial Number is 6027-6E67

Directory of C:\TP3CMD\América

28/12/2022	14:49	<DIR>	.
28/12/2022	14:49	<DIR>	..
28/12/2022	14:48	<DIR>	Brasil
28/12/2022	14:48	<DIR>	Canada
28/12/2022	14:49	<DIR>	EUA
		0 File(s)	0 bytes
		5 Dir(s)	13,927,497,728 bytes free

C:\TP3CMD\América>cd ..

C:\TP3CMD>mkdir Ásia

C:\TP3CMD>cd Ásia

C:\TP3CMD\Ásia>mkdir China

C:\TP3CMD\Ásia>mkdir India

C:\TP3CMD\Ásia>cd ..

 Command Prompt

C:\TP3CMD>mkdir Ásia

C:\TP3CMD>cd Ásia

C:\TP3CMD\Ásia>mkdir China

C:\TP3CMD\Ásia>mkdir India

C:\TP3CMD\Ásia>cd ..

C:\TP3CMD>mkdir Europa

C:\TP3CMD>cd Europa

C:\TP3CMD\Europa>mkdir Espanha

C:\TP3CMD\Europa>mkdir França

C:\TP3CMD\Europa>mkdir Portugal

C:\TP3CMD\Europa>dr

'dr' is not recognized as an internal or external command,
operable program or batch file.

C:\TP3CMD\Europa>dir

Volume in drive C has no label.
Volume Serial Number is 6027-6E67

Directory of C:\TP3CMD\Europa

28/12/2022	14:55	<DIR>	.
28/12/2022	14:55	<DIR>	..
28/12/2022	14:54	<DIR>	Espanha
28/12/2022	14:55	<DIR>	França
28/12/2022	14:55	<DIR>	Portugal
		0 File(s)	0 bytes
		5 Dir(s)	14,748,708,864 bytes free

EXECUÇÃO DA LINHA DE COMANDOS

26

EXERCÍCIO 12

12. A partir da pasta C:\TP3CMD, navegue até ao directório de Portugal (usando um único comando), e use o comando “echo 10M > pop.txt” para criar o ficheiro “pop.txt” com o número 10M que corresponde ao número de habitantes em Portugal. A partir daqui navegue até aos directórios de Espanha e França e repita o mesmo processo para indicar as populações de Espanha (47M) e de França (68M). Utilize o comando “type filename” para verificar os conteúdos dos ficheiros na linha de comandos.

EXPLICAÇÃO:

Após entrar no directório de Portugal com o comando cd, usei o comando “echo 10M > pop.txt” para criar o ficheiro “pop.txt” com o número 10M que corresponde ao número de habitantes em Portugal e repeti o mesmo processo nos directórios de Espanha e França tal como descrito no enunciado verificando também com o type pop.txt para verificar o conteúdo dos ficheiros.

27

EXERCÍCIO 12

```
C:\TP3CMD\Europa>cd ..  
  
C:\TP3CMD>cd Europa  
  
C:\TP3CMD\Europa>cd Portugal  
  
C:\TP3CMD\Europa\Portugal>echo 10M > pop.txt  
  
C:\TP3CMD\Europa\Portugal>cd ..  
  
C:\TP3CMD\Europa>cd Espanha  
  
C:\TP3CMD\Europa\Espanha>echo 47M > pop.txt  
  
C:\TP3CMD\Europa\Espanha>cd..  
  
C:\TP3CMD\Europa>cd França  
  
C:\TP3CMD\Europa\França>echo 68M > pop.txt  
  
C:\TP3CMD\Europa\França>type pop.txt  
68M  
  
C:\TP3CMD\Europa\França>
```

EXECUÇÃO DA LINHA DE COMANDOS

28

EXERCÍCIO 13

13. A partir da pasta C:\TP3CMD, copie o ficheiro “pop.txt” do directório de Portugal para os directórios do Brasil, Canadá e EUA.

EXPLICAÇÃO:

Após entrar na pasta TP3CMD usei o comando
" copy Europa\Portugal\pop.txt
América\Brasil\. " para copiar o ficheiro
pop.txt do directório de Portugal para o do
Brasil, Canadá e EUA.

29

EXERCÍCIO 13

```
C:\TP3CMD>copy Europa\Portugal\pop.txt América\Brasil\  
1 file(s) copied.
```

```
C:\TP3CMD>cd América\Brasil\
```

```
C:\TP3CMD\América\Brasil>dir  
Volume in drive C has no label.  
Volume Serial Number is 6027-6E67
```

```
Directory of C:\TP3CMD\América\Brasil
```

```
28/12/2022  15:30    <DIR>          .  
28/12/2022  15:30    <DIR>          ..  
28/12/2022  15:03                6 pop.txt  
                1 File(s)                6 bytes  
                2 Dir(s)  14,794,260,480 bytes free
```

```
C:\TP3CMD\América\Brasil>cd ..
```

```
C:\TP3CMD\América>cd..
```

```
C:\TP3CMD>copy Europa\Portugal\pop.txt América\Canada\  
1 file(s) copied.
```

```
C:\TP3CMD>copy Europa\Portugal\pop.txt América\EUA\  
1 file(s) copied.
```

EXECUÇÃO DA LINHA DE COMANDOS

30

EXERCÍCIO 14

14. A partir da pasta C:\TP3CMD\América, renomeie o directório “Canada” para “Canadá” (com acento no último “a”). Ainda na pasta América, modifique, usando o comando “echo” a população do Canadá para “250M” (ficheiro “pop.txt”). Mantendo-se na mesma pasta “América”, copie o ficheiro “pop.txt” do Canadá para a China. Finalmente navegue até à pasta “China” e verifique o conteúdo do ficheiro.

EXPLICAÇÃO:

Após entrar na pasta América para renomear o directório Canada usei o comando rename. Com o comando echo modifiquei o pop.txt para 250M e copiei-o do Canadá para a China com o comando " copy Canadá\pop.txt ..\Ásia\China\."

31

EXERCÍCIO 14

```
C:\TP3CMD>cd América

C:\TP3CMD\América>rename Canada Canadá

C:\TP3CMD\América>dir
Volume in drive C has no label.
Volume Serial Number is 6027-6E67

Directory of C:\TP3CMD\América

28/12/2022  15:41    <DIR>          .
28/12/2022  15:41    <DIR>          ..
28/12/2022  15:30    <DIR>          Brasil
28/12/2022  15:34    <DIR>          Canadá
28/12/2022  15:35    <DIR>          EUA
                0 File(s)                0 bytes
                5 Dir(s) 14,792,052,736 bytes free

C:\TP3CMD\América>echo 250M > Canadá\pop.txt

C:\TP3CMD\América>copy Canadá\pop.txt China\
The system cannot find the path specified.
0 file(s) copied.

C:\TP3CMD\América>copy Canadá\pop.txt ..\Ásia\China\
1 file(s) copied.

C:\TP3CMD\América>cd ..\Ásia\China

C:\TP3CMD\Ásia\China>dir
Volume in drive C has no label.
Volume Serial Number is 6027-6E67

Directory of C:\TP3CMD\Ásia\China

28/12/2022  16:04    <DIR>          .
28/12/2022  16:04    <DIR>          ..
28/12/2022  16:01             7 pop.txt
                1 File(s)                7 bytes
                2 Dir(s) 14,777,368,576 bytes free

C:\TP3CMD\Ásia\China>type pop.txt
250M
```

EXECUÇÃO DA LINHA DE
COMANDOS

32

PROGRAMA 15

15. A partir da pasta C:\TP3CMD execute o comando “tree /f” para listar a árvore de directórios e ficheiros. Remova a pasta Ásia e todo o seu conteúdo (explore o comando “rd /s”). Por fim, volte a utilizar o comando “tree” para verificar que a pasta Ásia foi removida e, finalmente, remova a pasta C:\TP3CMD.

EXPLICAÇÃO:

Ao entrar na pasta C:\TP3CMD., percebi que após algumas tentativas o comando indicado no enunciado “tree /f” não estava a funcionar na minha linha de comandos, por isso com alguma pesquisa encontrei o comando “dir /s /b /a-d” onde consegui obter o mesmo resultado que o enunciado pretendia e listei a árvore de directórios e ficheiros. Nesta encontram-se também os prints que ia fazendo dos exercícios, pois guardei-os na mesma pasta. Com o comando “rd /s” removi a pasta da Ásia, voltei a verificar se foi removida com o comando “dir /s /b /a-d” e por fim removi também com o comando “rd /s” a pasta inicialmente criada C:\TP3CMD, acabando assim os exercícios desta parte de linha de comandos.

33

PROGRAMA 15

```
C:\TP3CMD>dir /s /b /a-d
C:\TP3CMD\scree11.PNG
C:\TP3CMD\screen11e12.jpg
C:\TP3CMD\screen13.jpg
C:\TP3CMD\América\Brasil\pop.txt
C:\TP3CMD\América\Canadá\pop.txt
C:\TP3CMD\América\EUA\pop.txt
C:\TP3CMD\Europa\Espanha\pop.txt
C:\TP3CMD\Europa\França\pop.txt
C:\TP3CMD\Europa\Portugal\pop.txt
C:\TP3CMD\Ásia\China\pop.txt

C:\TP3CMD>cd Ásia

C:\TP3CMD\Ásia>cd ..

C:\TP3CMD>rd Ásia /s
Ásia, Are you sure (Y/N)? Y

C:\TP3CMD>dir /s /b /a-d
C:\TP3CMD\scree11.PNG
C:\TP3CMD\screen11e12.jpg
C:\TP3CMD\screen13.jpg
C:\TP3CMD\América\Brasil\pop.txt
C:\TP3CMD\América\Canadá\pop.txt
C:\TP3CMD\América\EUA\pop.txt
C:\TP3CMD\Europa\Espanha\pop.txt
C:\TP3CMD\Europa\França\pop.txt
C:\TP3CMD\Europa\Portugal\pop.txt

C:\TP3CMD>cd ..

C:\>rd TP3CMD/s
TP3CMD, Are you sure (Y/N)? Y

C:\>S
```

EXECUÇÃO DA LINHA DE COMANDOS

CONCLUSÃO

Para a conclusão deste relatório, escrevo que a execução tanto deste trabalho, tal como dos passados, ajudaram-me imenso a ter uma melhor noção de como se processa um programa, e de todas as linguagens de código existentes, visto que foi a minha primeira abordagem na área.

Decifrar sozinha os exercícios e estudar os slides e apontamentos de aulas, prepararam-me não só para outras unidades curriculares como para a própria, à medida que o grau de exigência foi subindo.

Termino este trabalho/relatório agradecendo por esta unidade curricular existir e por a forma de como esta é lecionada, que prepara da melhor forma possível especialmente quem está a começar.

Com a ajuda destas aulas, sinto-me mais preparada para os restantes semestres deste curso, sempre com o apoio e relembrar da frase "Dividir para conquistar!"

Obrigada!