Bilhete de Identidade e Cartão do Cidadão

MODO DE CÁLCULO:

Se o número tiver só 6 ou 7 dígitos acrescente 0 (zeros) à esquerda até perfazer 8 digitos

MULTIPLIQUE O

8.º dígito por 2

7.º dígito por 3

6.º dígito por 4

5.º dígito por 5

4.º dígito por 6

3.º dígito por 7

2.º digito por 8

1.º dígito por 9

Adicione os resultados

Calcule o Módulo 11 do resultado, isto é, o resto da divisão do número por 11.

Se o resto for 0 ou 1, o dígito de controle será 0

Se for outro algarismo x, o dígito de controle será o resultado de 11 - x

Algoritmo do Digito de Controlo

public bool ValidateNumeroDocumentoCC(string numeroDocumento) { int sum = 0; bool secondDigit = false;

if(numeroDocumento.Length != 12) throw new ArgumentException("Tamanho inválido para número de documento.");

for (int i = numeroDocumento.Length-1; i >= 0; --i) { int valor = GetNumberFromChar(numeroDocumento[i]);

if (secondDigit) { valor \*= 2;

if (valor > 9) valor -= 9; }

sum += valor; secondDigit = !secondDigit;

}

return (sum % 10) == 0; }

public int GetNumberFromChar(char letter) { switch(letter) { case '0' : return 0; case '1' : return 1;

case '2' : return 2;

case '3' : return 3;

case '4' : return 4;

case '5' : return 5;

case '6' : return 6;

case '7' : return 7;

case '8' : return 8;

case '9' : return 9;

case 'A' : return 10;

case 'B' : return 11;

case 'C' : return 12;

case 'D' : return 13;

case 'E' : return 14;

case 'F' : return 15;

case 'G' : return 16;

case 'H' : return 17;

case 'I' : return 18;

case 'J' : return 19;

case 'K' : return 20;

case 'L' : return 21;

case 'M' : return 22;

case 'N' : return 23;

case 'O' : return 24;

case 'P' : return 25;

case 'Q' : return 26;

case 'R' : return 27;

case 'S' : return 28;

case 'T' : return 29;

case 'U' : return 30;

case 'V' : return 31;

case 'W' : return 32;

case 'X' : return 33;

case 'Y' : return 34;

case 'Z' : return 35;

}

throw new ArgumentException("Valor inválido no número de documento.");

}