

# **Cartão de Cidadão**

## **Validação de Número de Documento do Cartão de Cidadão**

26 de Janeiro de 2009

Versão 1.0

**AMA**

## ÍNDICE

<b>1. INTRODUÇÃO .....</b>	<b>3</b>
<b>2. NÚMERO DE DOCUMENTO CARTÃO DE CIDADÃO .....</b>	<b>4</b>
2.1. COMPOSIÇÃO .....	4
2.2. VALIDAÇÃO <i>CHECK DIGIT</i> .....	4
2.3. TABELA DE CONVERSÃO DO ALFABETO .....	5
2.4. EXEMPLO DE CÓDIGO <i>C#</i> .....	6

## 1. Introdução

O Cartão de Cidadão possui um número de documento, constituído por diversos elementos base, ao qual se pode utilizar um método de validação de *check digit*, de modo a evitar falha aquando da recolha do mesmo.

Nos capítulos seguintes é abordado a constituição do número de documento, a fórmula de validação *check digit*, bem como as tabelas de conversão e um exemplo de código demonstrativo da validação.

## 2. Número de Documento Cartão de Cidadão

### 2.1. Composição

O número de documento é constituído por 4 elementos distintos, sendo que se encontra visível na parte frontal do Cartão de Cidadão.

Formato: **DDDDDDDD C AAT**

Onde:

**D** - Número de Identificação Civil [0.. 9]

**C** - *Check Digit* do Número de Identificação Civil [0.. 9]

**A** - Versão [A.. Z, 0.. 9]

**T** - *Check Digit* Número de Documento [0.. 9]

### 2.2. Validação *Check Digit*

A fórmula de validação efectua a verificação usando o número de documento completo. O número de documento tem de passar os seguintes testes:

1. Efectuando uma contagem da direita para a esquerda do número de documento, duplicar o valor de cada 2º elemento, sendo que as letras deverão ser substituídas de acordo com a tabela de conversão presente neste documento;
2. No caso do resultado da duplicação ser igual ou superior a 10, subtrair 9 ao seu valor;
3. Somar a totalidade dos valores obtidos;
4. Ao valor obtido, deve-se calcular o resto da sua divisão por 10, sendo que se esse valor for 0 o número de documento é válido.

Como exemplo, e usando o número de documento 00000000 0 ZZ4, iniciamos por retirar os espaços existentes. De seguida, deverão ser substituídas as letras pelo seu respectivo valor, sendo que para o caso concreto, ficaríamos com o valor 000000000**(35)(35)**4.

[000000000**(35)(35)**4]

**Passo 1 e 2:**  $(0 \times 2) = 0, (0 \times 2) = 0, (0 \times 2) = 0, (0 \times 2) = 0, (0 \times 2) = 0, ((35 \times 2) = 70 - 9) = 61$

**Passo 3:**  $(0) + 0 + (0) + 0 + (0) + 0 + (0) + 0 + (0) + 35 + (61) + 4 = 100$

**Passo 4:**  $100 \bmod 10 = 0$  sendo que confirma que o valor é válido.

## 2.3. Tabela de Conversão do Alfabeto

As letras do alfabeto presentes no número de documento devem ser convertidas para dígitos, de acordo com a tabela seguinte, para cálculo do *check digit*.

Letra	Valor
A	10
B	11
C	12
D	13
E	14
F	15
G	16
H	17
I	18
J	19
K	20
L	21
M	22
N	23
O	24
P	25
Q	26
R	27
S	28
T	29
U	30
V	31
W	32
X	33
Y	34
Z	35

## 2.4. Exemplo de Código C#

```
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.");
}
```