

Programação III – Validando CPF

Validação de CPF*

O CPF tem 11 dígitos, os dois últimos são chamados de dígitos verificadores e são consequência dos 9 anteriores. Isso quer dizer que você pode idealizar qualquer combinação de 9 dígitos e só precisa calcular os dois últimos dígitos para ter um CPF válido.

O primeiro dígito é o um número que depende do resto da soma de uma multiplicação de cada um dos nove dígitos por 10, 9, ... 3, 2 por 11. E a do segundo é o mesmo esquema, porém inclui o primeiro dígito verificador, e a multiplicação começa em 11 (11, 10 ... 3, 2).

Matematicamente falando, o primeiro dígito é 0 se o resto da divisão de $(10 * a + 9 * b + 8 * c + 7 * d + 6 * e + 5 * f + 4 * g + 3 * h + 2 * i)$ por 11 for menor que 2 e é a diferença entre 11 e o resto da divisão caso contrário; sendo **abcdefghi** os nove primeiros números do CPF.

Para exemplificar utilizamos 123456789 como os primeiros nove dígitos do CPF.

Dígitos do CPF	1	2	3	4	5	6	7	8	9
Multiplicadores	10	9	8	7	6	5	4	3	2
Produtos	10	18	24	28	30	30	28	24	18

*** Para calcular o primeiro dígito verificador soma-se os produtos e dividimos por 11. Caso o resto da divisão seja menor que 2 o dígito verificador é 0, senão é a diferença de 11 e o próprio resto.**

Com esta tabelinha fica bem mais fácil o cálculo do primeiro dígito verificador, usando módulo 11:

$$10 + 18 + 24 + 28 + 30 + 30 + 28 + 24 + 18 = 210$$

$$210 \bmod 11 = 1$$

Como 1 é menor que 2, o dígito é **0**.

Vamos ao próximo dígito verificador.

Dígitos do CPF	1	2	3	4	5	6	7	8	9	0
Multiplicadores	11	10	9	8	7	6	5	4	3	2
Produtos	11	20	27	32	35	36	35	32	27	0

Somando os produtos temos:

$$11 + 20 + 27 + 32 + 35 + 36 + 35 + 32 + 27 + 0 = 255$$

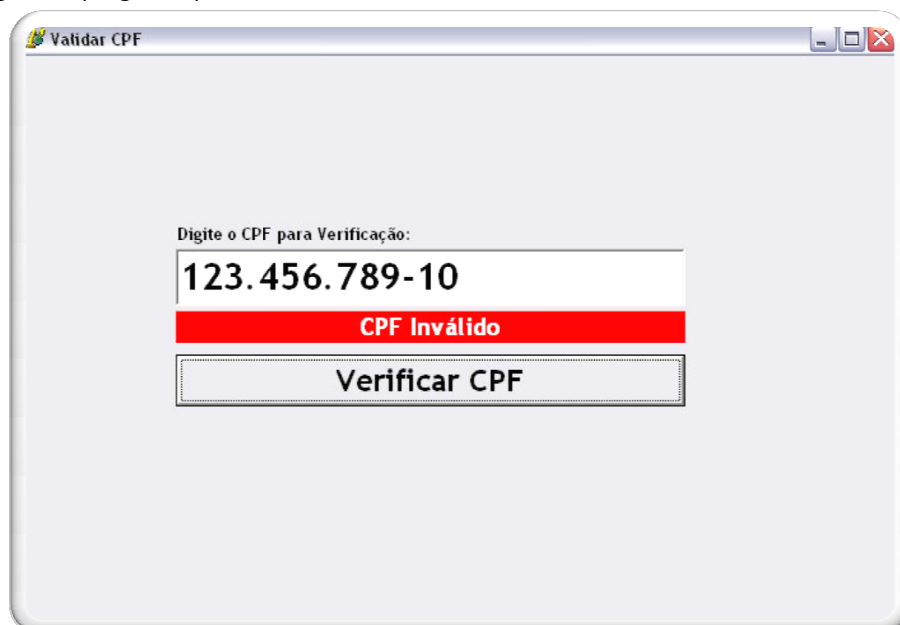
$$255 \bmod 11 = 2$$

Como 2 não é menor que 2, o segundo dígito verificador é $11 - 2 = 9$.

Assim, o CPF completo é **123456789-09**.

* Texto retirado da Internet

Abaixo segue a imagem do programa para Validar CPF



Código para Delphi (BitBtn)

```
Var
  dig1,dig2,i,cal : integer;
  digitos         : array [1..11] of integer;
  controle        : boolean;
Begin
  controle := false;
  For i := 1 to length(edit1.text) - 3 do
  Begin
    If i < 4 then
      digitos[i] := StrToInt(edit1.text [i])
    Else
      If (i >= 4) and (i < 7) then
        digitos[i] := StrToInt(edit1.text [i+1])
      Else
        If (i >= 7) and (i < 10) then
          digitos[i] := StrToInt(edit1.text [i+2])
        Else
          If (i >= 10) then
            digitos[i] := StrToInt(edit1.text [i+3]);
  End;
  // Verifica se os digitos são iguais
  For i := 1 to 10 do
    If digitos[i] <> digitos[i+1] then
      Begin
        controle := true;
        break;
      End;
  If controle then
  Begin
    controle := false;

    // Primeiro Digito
    cal :=(10 * digitos[1])+(9 * digitos[2])+(8 * digitos[3])+(7 * digitos[4])+(6 * digitos[5])+
    (5 * digitos[6])+(4 * digitos[7])+(3 * digitos[8])+(2 * digitos[9]);
    dig1 := 11-(cal mod 11);
    if dig1 >= 10 then
      dig1 := 0;

    // Segundo Digito
    cal :=(11 * digitos[1])+(10 * digitos[2])+(9 * digitos[3])+(8 * digitos[4])+(7 * digitos[5])+
    (6 * digitos[6])+(5 * digitos[7])+(4 * digitos[8])+(3 * digitos[9])+(2 * dig1);
    dig2 := 11-(cal mod 11);
    If dig2 >= 10 then
      dig2 := 0;

    If ( edit1.text[13] = IntToStr(dig1) )and(edit1.text [14] = IntToStr(dig2) ) then
      Begin
        LabelErro.Caption:='CPF Válido';
        LabelErro.Color:=ClGreen;
        LabelErro.Font.Color:=ClWhite;
      End
    Else
      Begin
        LabelErro.Caption:='CPF Inválido';
        LabelErro.Color:=ClRed;
        LabelErro.Font.Color:=ClWhite;
      End;
  End;
  Else
  Begin
    LabelErro.Caption:='CPF Inválido';
    LabelErro.Color:=ClRed;
    LabelErro.Font.Color:=ClWhite;
  End;
End;
```