

UCG - Eng Computação - Programação II
 Programa: Validação de um CPF informado
 Marco Tulio Rodrigues Braga
 >>>>Principal.java

```

/** @author MT
import javax.swing.JOptionPane;
public class Principal {
    public static void main(String[] args) {
        String ncpfString;
        String dvString;

//Declarando os atributos
//"Chamando os atributos"
        CPF classeCPF;

//Alocando o objeto na memoria
        classeCPF = new CPF();

//Iniciando a interface com o usuario
//Lendo o numero do CPF
        ncpfString =
JOptionPane.showInputDialog("Digite o numero do
CPF\n(obs: Sem o Digito verificador)");

//Lendo o Digito Verificador do teclado
        dvString =
JOptionPane.showInputDialog("Digite o Digito
Verificador\n(obs:Apenas 2 digitos)");

//Chamei o metodo de conversão do CPF
        classeCPF.converterCPF();

//Chamei o metodo de somar a serie do cpf
        classeCPF.somaSerie();

//Chamada do metodo de verificar validade
//falta terminar
        if (
classeCPF.verifiValidade(true)) {

            JOptionPane.showMessageDialog( null, "O
CPF");
        }
    }
}

```

>>>Classe CPF.java

```

class CPF {
    //atributos
    public int ncpf1, ncpf2, ncpf3;
    public int ndv, ncpf, s1, s2, resto, dv1,
dv2, dvCalc;
    public int a, b, c, d, e, f, g, h, i, j,
k;
    public int rTmp1, rTmp2, rTmp3, rTmp4,
rTmp5, rTmp6, rTmp7, rTmp8;
    public String ncpfString;
    public String numberCPF, dvString;

//Descobrimdo o valor do cpf
    public void converterCPF(){
        ncpf1 = Integer.parseInt(ncpfString);

//Convertendo a String CPF digitado em Inteiro
        ncpf = Integer.parseInt(ncpfString);
//Convertendo a String Dv digitado em Inteiro
        ndv = Integer.parseInt(dvString);

        //descobrimdo valores do cpf
        a = ncpf/100000000;
        rTmp1 = (ncpf%100000000);
        b = rTmp1/10000000;
        rTmp2 = rTmp1%10000000;
        c = rTmp2/1000000;
        rTmp3 = rTmp2%1000000;
        d = rTmp3/100000;
        rTmp4 = rTmp3%100000;
        e = rTmp4/10000;
        rTmp5 = rTmp4%10000;
        f = rTmp5/1000;
        rTmp6 = rTmp5%1000;
        g = rTmp6/100;
        rTmp7 = rTmp6%100;
        h = rTmp7/10;
        i = rTmp7%10; //pq?=rTmp8 =
rTmp7%10;i = rTmp8/1; economizar nas contas
    }

//Obtendo os dgts verificadores
    public void somaSerie(){

//Aplicando regras da formula
        s1 = a*10 + b*9 + c*8 + d*7 + e*6
+ f*5 + g*4 + h*3 + i*2;
        resto = s1%11;
        if (resto==0 || resto==1) dv1 = 0;
        if (resto>=2 && resto<=10) dv1 =
11-resto;

        s1 = a*11 + b*10 + c*9 + d*8 + e*7
+ f*6 + g*5 + h*4 + i*3 + dv1*2;
        resto =0;
        resto = s2%11;
        if (resto==0 || resto==1) dv2 = 0;
        if (resto>=2 && resto<=10) dv2 =
11-resto;
    }

    public boolean verifiValidade(boolean
resp) {
        dvCalc = dv1*10 + dv2*1; //Digito
verificador Calculado
        if (dvCalc == ndv) resp=true;
        if (dvCalc != ndv) resp=false;
        return (resp);
    }
}

```