

```

package main;

//importar o pacote que permite a entrada de dados, classe Scanner.
import java.util.Scanner;

//determinando a classe
public class CalculoIP{

    //variáveis para todo o sistema

    static Scanner sc = new Scanner (System.in);

    static boolean cont = true;

    static int [] ipVal;

    static int prefix;

    static String mask;

    static int nHosts;


    //criação do método
    public static void main(String[] args) {

        boolean ipV;

        String ip;

        String classe;


        //executar caso o IP digitado for inválido
        do{

            System.out.println("Digite um endereço de IP válido (ou 'SAIR' para sair do programa):
");

            ip = sc.next();

            ipV = validarIP(ip);

        }while(ipV == false && cont == true);


        if(cont == true){

            classe = testeClasse();

            do{

                System.out.println("Digite o prefixo da rede: ");

```

```

        prefix = sc.nextInt();
    }while (prefix <1 && prefix >32);
    if (classe.equals("C") && prefix>24 && prefix<31){
        validarPrefix();
        imprimirTab();
    }
}

System.out.println("FIM da Bagaça.");
}

```

//determinar qual classe do IP

```

private static String testeClasse() {
    String classe;

    // testar o valor do 1º octeto do IP
    if(ipVal[0]<128){
        classe = "A";

        System.out.println("IP Classe "+classe);

        return classe;
    }else if(ipVal[0]<192){
        classe = "B";

        System.out.println("IP Classe "+classe);

        return classe;
    }else if(ipVal[0]<224){
        classe = "C";

        System.out.println("IP Classe "+classe);

        return classe;
    }else if(ipVal[0]<240){
        classe = "D";

        System.out.println("IP Classe "+classe);

        return classe;
    }else{

```

```

        classe = "E";

        System.out.println("IP Classe "+classe);

        return classe;
    }
}

```

//atribuir a Máscara e os números de Hosts para cada prefixo

```

private static void validarPrefx() {

    //a quantidade de hosts é 2 elevado a quantidade de bits que sobraram do prefixo

    nHosts = (int) Math.pow(2,32-prefix);

    int masc = 0;

    int i=7;

    int j=32-prefix;

    //while fará o calculo do número final da máscara, calculando a potência de cada bit
    ligado no último octeto

    while(i>=j){

        masc += Math.pow(2, i);

        i--;

    }

    mask="255.255.255."+masc;

}

```

```

private static void imprimirTab() {

    //reunir os 3 primeiros octetos do IP

    String ipInicio = ipVal[0]+"."+ipVal[1]+"."+ipVal[2]+".";

    int oct4 = 0;

    int broad = nHosts;

    //imprimir a máscara

    System.out.println("Mascara: "+mask);

    //calcula o Broadcast

```

```

int x = 0;

int y = 0;

while (x <= ipVal[3]){

    x+=nHosts;

    y = (x-1);

}

System.out.println("Broadcast: "+ipInicio+y);

int i = 1;

//impressão das sub-redes

System.out.println(i+" - Sub-Rede: "+ipInicio+oct4+" | Host: "+ipInicio+(oct4+1)+ " à
" +(broad-2)+ " | Broadcast: "+ipInicio+(broad-1));

    oct4 += nHosts;

    broad += nHosts;

    i++;

    while(oct4 < 255);

}

//pedir número ip e o valida

private static boolean validarIP(String ip) {

    String[] vetorS;

    boolean valido =true;

    //if para saber se o usuario deseja continuar

    if(ip.equalsIgnoreCase("sair")){

        cont=false;

        return false;

    }

    //função Split. quebra a String sempre que aparecer um ponto

    vetorS=ip.split("\\.");

    // for para cada caracter em cada posição do vetor

    for(int i=0;i<vetorS[i].length();i++){

        //teste para saber se é caracter ou número

```

```

        if(!Character.isDigit(vetorS[i].charAt(i))){
            valido = false;
            break;
        }
    }

//criar um vetor de inteiros com o tamanho do vetor de String
int [] vetorI = new int [vetorS.length];

if(valido==true){
    //for para converter String para Int
    for(int i=0;i<vetorS.length;i++){
        vetorI[i] = Integer.decode(vetorS[i]);
    }
    //if testa se o vetorI possui 4 espaços
    if(vetorI.length!=4){
        valido=false;
    }
    //for que testa se o número está entre 0 e 255
    for (int i=0; i<vetorI.length;i++){
        if(vetorI[i]<0 || vetorI[i]>255){
            valido=false;
            break;
        }
    }
}

if(valido==false){
    System.out.println("IP Invalido");
    valido=false;
    return valido;
}else{
    ipVal=vetorI;
}

```

```
        valido=true;
        return valido;
    }
}
```