

```

import java.util.Scanner;

public class AFDConsole {
    static private AFD afd;
    static String[] palavras;
    public static void main(String[] args) {

        afd = new AFD();
        Scanner entrada = new Scanner(System.in);
        definindoEstados(entrada);
        definindoAlfabeto(entrada);
        definindoTransicoes(entrada);
        definindoEstadosIniciais(entrada);
        definindoEstadosFinais(entrada);
        definindoPalavras(entrada);
        afd.testes(palavras);
        entrada.close();

    }

    private static void definindoEstados(Scanner entrada){
        System.out.println("-> Definindo estados");
        String estados = entrada.nextLine();
        for(char ID : estados.toCharArray()){
            if(ID != ' '){
                Estado estado = new Estado(ID);
                afd.adicionarEstado(estado);
            }
        }
    }

    private static void definindoAlfabeto(Scanner entrada){
        System.out.println("-> Definindo alfabeto");
        String alfabeto = entrada.nextLine();
        for(char token : alfabeto.toCharArray()){
            if(token != ' ')
                afd.adicionarTokenAlfabeto(token);
        }
    }

    private static void definindoTransicoes(Scanner entrada){
        System.out.println("-> Definindo transicoes");
        while(true){
            String transicao = entrada.nextLine();
            if(transicao.compareTo("#") == 0)
                break;

            Estado origem = null, destino = null; // estados da transicao
            char tokenConsumido = '\0'; // token lido na transicao
            int index = 0;
            for(char token : transicao.toCharArray()){
                if(token != ' '){
                    // origem da transicao
                    if(index == 0)

```

```

        origem = afd.estadoPorId(token);
    else if (index == 1)
        destino = afd.estadoPorId(token);
    else
        tokenConsumido = token;
    index++; // incrementa o contador
}
}

if(origem != null) {
    // adicionando transicao
    Transicao _transicao = new Transicao(tokenConsumido, destin
    origem.adicionarTransicao(_transicao);
} else {
    System.out.println(String.format("Erro %s estado nao existe
AFD_ERRORS.ERRO_03));
}
}

static void definindoEstadosIniciais(Scanner entrada){
    System.out.println("-> Definindo estados iniciais");
    String estados = entrada.nextLine();
    for(char token : estados.toCharArray()){
        if(token != ' ') {
            Estado estado = afd.estadoPorId(token);
            if(estado != null)
                estado.estadoInicial = true;
        }
    }
}

static void definindoEstadosFinais(Scanner entrada){
    System.out.println("-> Definindo estados finais");
    String estados = entrada.nextLine();
    for(char token : estados.toCharArray()){
        if(token != ' ') {
            Estado estado = afd.estadoPorId(token);
            if(estado != null)
                estado.estadoFinal = true;
        }
    }
}

static void definindoPalavras(Scanner entrada){
    System.out.println("-> palavras a serem testadas");
    String _palavras = entrada.nextLine();
    palavras = _palavras.split(" ");
}
}

```