



INSTITUTO POLITECNICO NACIONAL

ESCUA SUPERIOR DE COMPUTO

**“Practica 1: Diseño e implementación de las clases AFN,
AFD.”**

MIEMBROS:

González Vázquez Cristian Armando

PROFESOR DE U.A.

Saucedo Delgado Rafael Norman

Unidad de Aprendizaje:

Análisis y Diseño de Algoritmos



Desarrollo

1. ¿Qué compilador de lenguaje C++ usaste?

El compilador que se utilizó para el lenguaje fue MinGW.

2. ¿Qué versión del compilador de lenguaje C++ usaste?

La versión del compilador que fue utilizado fue la 6.3.0

```
PS C:\Users\name\Documents\GitHub\Compiladores2021\Practical1> g++ --version
g++.exe (MinGW.org GCC-6.3.0-1) 6.3.0
Copyright (C) 2016 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

3. ¿Qué es el diseño orientado a objetos?

Es un paradigma de programación el cual se basa en los conceptos de clases y objetos. Sirve para poder estructurar un programa en pequeños y simples piezas de códigos como plantillas (clases), de las cuales se usan para crear instancias de estos (objetos).

4. ¿Qué hace un AFD?

Es un autómata que, por cada símbolo de entrada, uno puede determinar a qué estado se moverá.

5. ¿Qué hace un AFND?

Es un autómata que, por cada símbolo de entrada, la máquina se puede mover a cualquiera de las combinaciones de sus estados, es decir que no se puede determinar el estado al cual se moverá.

6. ¿Cuál es tu usuario de github?

Mi usuario es **cristiangv97**.

CODIGO:

```
#include <iostream>
#include <stdlib.h>

class automataFD{
private: // Atributos
    string conjuntoEstados[]; // Conjunto finito de estados
    string alfabeto[]; // Alfabeto
    string estadoInicial; // Estado inicial en el cual el automata empezara
    string estadosFinales[]; // Estados finales de aceptacion por el automata

public:
    void automataFD(string); // metodo donde se asignara la cadena de entrada al automata
    void movimientoCadena(string); // metodo donde se ira moviendo el apuntador de toda la cadena
    void validacionCadena(string); // metodo donde se procesara y validara el simbolo de entrada
    void estadoFinal(); // metodo donde se regresara el estado en el cual termino el automata
};

class automataFND{
private: // Atributos
    string conjuntoEstados[]; // Conjunto finito de estados
    string alfabeto[]; // Alfabeto
    string estadoInicial; // Estado inicial en el cual el automata empezara
    string estadosFinales[]; // Estados finales de aceptacion por el automata

public:
    void automataFD(string); // metodo donde se asignara la cadena de entrada al automata
    void movimientoCadena(string); // metodo donde se ira moviendo el apuntador de toda la cadena
    void validacionCadena(string); // metodo donde se procesara y validara el simbolo de entrada
    void estadosFinales(); // metodo donde se regresaran los estados del automata
};
```