



UNIVERSIDAD NACIONAL AUTÓNOMA DE  
MÉXICO

FACULTAD DE INGENIERÍA

**Proyecto 2**

**Analizador Sintáctico**

PRESENTA

**Navarrete Zamora Aldo Yael**

PROFESORA

**Laura Sandoval Montaña**

ASIGNATURA

**Compiladores**

## 1. Descripción del problema

Valor	Palabra reservada	Equivalencia en C
0	alternative	case
1	big	long
2	evaluate	if
3	instead	else
4	large	double
5	loop	while
6	make	do
7	number	int
8	other	default
9	real	float
10	repeat	for
11	select	switch
12	small	short
13	step	continue
14	stop	break
15	symbol	char
16	throw	return

Valor	Op. relacional
0	<
1	>
2	<=
3	>=
4	==
5	!=

Figura 1: Catálogos para las palabras reservadas y los operadores relacionales.

## 2. Conjuntos de selección para cada producción

## 3. Ejecución del programa

El programa se encuentra comentado, con una descripción breve de lo que hace cada una de las funciones. La sangría está muy cuidada y se ve bastante presentable y legible el código. Para correr el programa, debemos tener instalado flex en nuestro equipo y seguir los siguientes pasos una vez que lo tengamos instalado.

1. Abrir en una ventana de terminal la carpeta donde se encuentran todos los archivos.
2. Una vez dentro de la carpeta correremos el comando `flex lexical-analyzer.l`
3. `gcc lex.yy.c -ly -ll -o lexical-analyzer.out` para compilarlo mediante gcc y tener un archivo de salida ejecutable para un entorno LINUX.
4. Finalmente, `./lexical-analyzer.out inputFile.txt` es importante destacar que, el archivo *inputFile.txt* es el archivo de entrada, **en caso de no teclear ningún archivo de entrada el programa puede manejar este tipo de problema mandando un mensaje de que es necesaria esa entrada por consola.**

Una vez ejecutados los pasos anteriores, podremos observar **dos archivos de texto para la salida:**

1. El archivo de texto para la generación de los tokens.

2. El archivo de texto para la la impresión de errores léxicos.

Este último archivo reportará qué expresiones regulares no están definidas por el lenguaje desarrollado.

## 4. Conclusiones

"Hola

"

— **Navarrete Zamora Aldo Yael**, Estudiante de Ingeniería en Computación