

**UNIVERSIDAD DE GUADALAJARA**

**CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E  
INGENIERÍAS**

**INGENIERÍA EN COMPUTACIÓN(INCO)**



**MATERIA:** SEMINARIO DE SOLUCIÓN DE PROBLEMAS DE  
TRADUCTORES DE LENGUAJES II.

**SECCIÓN:** D02.

**DOCENTE:** Lopez Franco, Michel Emanuel.

**ALUMNO:** Izmael Guzmán Murguía.

**CÓDIGO:** 216728179.

**Tarea:** Analizador Sintáctico (Implementación usando Objetos).

**FECHA DE ENTREGA:** Lunes 06 de septiembre de 2022.

## Pruebas:

1. Dada una entrada de cadena que represente una suma es posible determinar un grafo si es que todo sale bien.

The screenshot shows a development environment with Visual Studio Code on the left and a web browser on the right. In VS Code, the file 'valida.js' contains the following code:

```
pruebas > p1.txt
1 int a, Izmael, Guzman, Murguia;
2 a = Izmael + Guzman + Murguia;
```

The web browser displays a compiler interface with the following components:

- LR(1) Item Set Table:**

Item	Lookahead	Action
\$0Izmaeld2+d3Guzmand2+d3Murguad2	\$	r2 = E -> id
\$0Izmaeld2+d3Guzmand2+d3E4	\$	r1 = E -> id + E
\$0Izmaeld2+d3E4	\$	r1 = E -> id + E
\$0E1	\$	r0(accept)

- Parse Tree Diagram:** A tree structure showing the derivation of the expression 'Izmael + Guzman + Murguia'. The root node is 'E', which branches into 'E + E'. The left 'E' branches into 'E + E', and the right 'E' branches into 'Murguia'. The leftmost 'E' branches into 'Izmael' and '+', and the middle 'E' branches into 'Guzman' and '+'. The nodes are labeled with their respective values or symbols.
- Console Log:** Shows the parsing process with messages like 'inserto 2', 'inserto 1', 'Salida:', and 'Valida:'.

2. Además se proporciona la tabla LR(1) numérica y con el string completo para un mejor entendimiento.

The screenshot shows the same development environment as above, but with a different LR(1) item set table and a different parse tree diagram.

- LR(1) Item Set Table:**

Item	Lookahead	Action
\$0Izmaeld2+d3E4	\$	r1 / id
\$0E1	\$	r0

- Parse Tree Diagram:** A tree structure showing the derivation of the expression 'Izmael + Guzman + Murguia'. The root node is 'E', which branches into 'E + E'. The left 'E' branches into 'E + E', and the right 'E' branches into 'Murguia'. The leftmost 'E' branches into 'Izmael' and '+', and the middle 'E' branches into 'Guzman' and '+'. The nodes are labeled with their respective values or symbols.

At the bottom of the browser window, there is a footer with the following text:

Creado por: Izmael Guzman Murguia.  
Materia: Seminario de solución de problemas de traductores de lenguajes II.  
Maestro: Michel Emanuel Lopez Franco.

3. Así mismo se consideran todos los casos, incluso cuando esa expresión es un único id.

The screenshot shows a web-based compiler interface. On the left, a code editor displays the following code:

```
pruebas > p1.txt
1 int a,Izmael,Guzman,Murguia;
2 a = Izmael;
```

On the right, the compiler's output is shown. It includes a table titled "Mini analizador sintactico" with two sections. The first section has columns "Pila", "Entrada", and "Salida". The second section has columns "Pila", "Entrada", and "Salida". The table shows the state of the compiler for the input "Izmael".

Pila	Entrada	Salida
20	02	d2
2002	2	r2 = E -> id
2031	2	r0(accept)

Pila	Entrada	Salida
\$0	Izmael\$	d2
\$0Izmaeld2	\$	r2 = E -> id
\$0E1	\$	r0(accept)

Below the table, a diagram shows the expression "Izmael" being processed. The diagram consists of a single node labeled "E" connected to a node labeled "Izmael".

At the bottom, the text reads: "Creado por: Izmael Guzman Murguia. Materia: Seminario de solucion de problemas de traductores de lenguajes II. Maestro: Michel Emanuel Lopez Franco."

4. Pueden generarse expresiones de diferente tamaño lo que generará un grafo diferente.

The screenshot shows the same web-based compiler interface as before, but with a more complex expression. The code editor displays the following code:

```
pruebas > p1.txt
1 int a,b,c,d;
2 a = a+b;
```

The compiler's output is shown. The table titled "Mini analizador sintactico" shows the state of the compiler for the input "a+b".

Pila	Entrada	Salida
\$0ad2+d3bd2	\$	r2 = E -> id
\$0ad2+d3E4	\$	r1 = E -> id + E
\$0E1	\$	r0(accept)

Pila	Entrada	Salida
\$0ad2+d3bd2	\$	r2 = E -> id
\$0ad2+d3E4	\$	r1 = E -> id + E
\$0E1	\$	r0(accept)

Below the table, a diagram shows the expression "a+b" being processed. The diagram consists of three nodes: "a", "b", and "E". The node "a" is connected to the node "E", and the node "b" is connected to the node "E". The node "E" is also connected to the node "E".

At the bottom, the text reads: "Creado por: Izmael Guzman Murguia. Materia: Seminario de solucion de problemas de traductores de lenguajes II. Maestro: Michel Emanuel Lopez Franco."

## 5. Expresión de suma con tres variables.

The screenshot shows a compiler interface with two main panes. The left pane, titled 'pruebas > p1.txt', contains the following code:

```
1 int a,b,c,d;  
2 a = a+b+c;
```

The right pane, titled 'Compilador — Mozilla Firefox', displays the output of the compiler. It shows a table with three rows of memory addresses and their corresponding values:

\$0ad2+d3bd2+d3E4	\$	r1 = E -> id + E
\$0ad2+d3E4	\$	r1 = E -> id + E
\$0E1	\$	r0(accept)

Below the table is a parse tree diagram for the expression 'a+b+c'. The root node is 'E', which has three children: 'a', 'E', and 'c'. The middle 'E' node has two children: 'b' and 'E'. The innermost 'E' node has one child: 'c'. The diagram is highlighted with a green border.

At the bottom of the right pane, there is a footer with the following text:

Creado por: Izmael Guzman Murguía.  
Materia: Seminario de solución de  
problemas de traductores de lenguajes II.  
Maestro: Michel Emanuel Lopez Franco.

The footer also includes the logo of CUCEI and the logo of the Universidad de Guadalajara.

## 6. Podemos generar expresiones muy largas.

The screenshot shows the same compiler interface as in the previous image, but with a much longer expression. The left pane, titled 'pruebas > p1.txt', contains the following code:

```
1 int a,b,c,d,e,f,g,h,izmael;  
2 a = a+b+c+d+e+f+g+h+izmael;
```

The right pane, titled 'Compilador — Mozilla Firefox', displays the output of the compiler. It shows a table with three rows of memory addresses and their corresponding values:

\$0ad2+d3E4		
\$0E1		

Below the table is a parse tree diagram for the expression 'a+b+c+d+e+f+g+h+izmael'. The root node is 'E', which has many children, each representing a variable or a sub-expression. The diagram is highlighted with a green border.

At the bottom of the right pane, there is a footer with the following text:

Creado por: Izmael Guzman Murguía.  
Materia: Seminario de solución de  
problemas de traductores de lenguajes II.  
Maestro: Michel Emanuel Lopez Franco.

The footer also includes the logo of CUCEI and the logo of the Universidad de Guadalajara.

7. Se genera de igual manera la tabla con enteros y con la misma cadena.

pruebas > p1.txt

```
1 int a,b,c,d,e,f,g,h,izmael;
2 a = a+b+c+d+e+f+g+h+izmael;
```

Pila	Entrada	Salida
\$0	a+b+c+d+e+f+g+h+izmael	d2
\$0ad2	+b+c+d+e+f+g+h+izmael	d3
\$0ad2+d3	+c+d+e+f+g+h+izmael	d2
\$0ad2+d3bd2	+c+d+e+f+g+h+izmael	d3
\$0ad2+d3bd2+d3	+d+e+f+g+h+izmael	d2
\$0ad2+d3bd2+d3d2	+d+e+f+g+h+izmael	d3
\$0ad2+d3bd2+d3d2+d3	+e+f+g+h+izmael	d2
\$0ad2+d3bd2+d3d2+d3d2	+e+f+g+h+izmael	d3
\$0ad2+d3bd2+d3d2+d3d2+d3	+f+g+h+izmael	d2
\$0ad2+d3bd2+d3d2+d3d2+d3d2	+f+g+h+izmael	d3
\$0ad2+d3bd2+d3d2+d3d2+d3d2+d3	+g+h+izmael	d2
\$0ad2+d3bd2+d3d2+d3d2+d3d2+d3d2	+g+h+izmael	d3
\$0ad2+d3bd2+d3d2+d3d2+d3d2+d3d2+d3	+h+izmael	d2
\$0ad2+d3bd2+d3d2+d3d2+d3d2+d3d2+d3d2	+h+izmael	d3
\$0ad2+d3bd2+d3d2+d3d2+d3d2+d3d2+d3d2+d3	+izmael	d2
\$0ad2+d3bd2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2	+izmael	d3
\$0ad2+d3bd2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3	izmael	d2
\$0ad2+d3bd2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3		r2 = E -> id
\$0ad2+d3bd2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2		r1 = E -> id + E
\$0ad2+d3bd2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3		r1 = E -> id + E
\$0ad2+d3bd2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2		r1 = E -> id + E
\$0ad2+d3bd2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3		r1 = E -> id + E
\$0ad2+d3bd2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2		r1 = E -> id + E
\$0ad2+d3bd2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3		r1 = E -> id + E
\$0ad2+d3bd2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2		r1 = E -> id + E
\$0ad2+d3bd2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3d2+d3		r0(accept)

Creado por: Izmael Guzman Murguía.  
Materia: Seminario de solución de problemas de traductores de lenguajes II.  
Maestro: Michel Emmanuel Lopez Franco.

8. Expresiones largas.

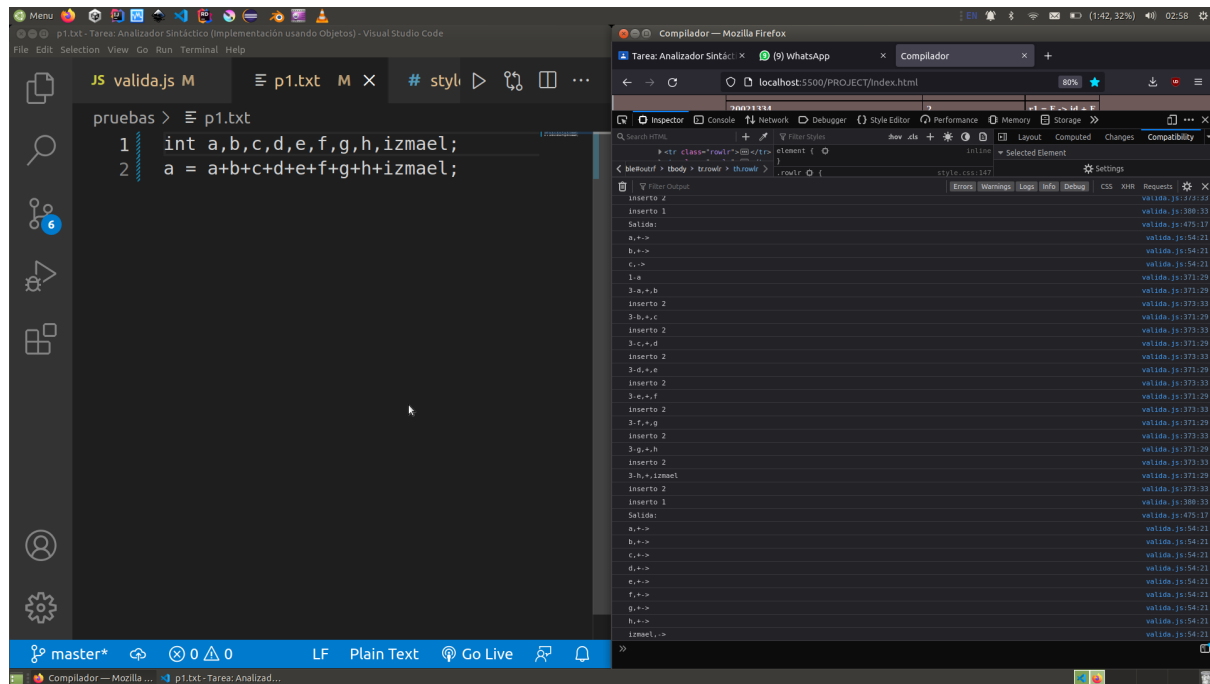
pruebas > p1.txt

```
1 int a,b,c,d,e,f,g,h,izmael;
2 a = a+b+c+d+e+f+g+h+izmael;
```

Pila	Entrada	Salida
\$0ad2+d3bd2+d3E4		\$
\$0ad2+d3E4		\$
\$0E1		\$

Creado por: Izmael Guzman Murguía.  
Materia: Seminario de solución de problemas de traductores de lenguajes II.  
Maestro: Michel Emmanuel Lopez Franco.

9. La información del grafo y nodos también se puede ver en la terminal,



**Conclusión:**

En esta actividad se fortaleció sobre todo la parte de el desarrollo autodidacta en el sentido de que fue necesario investigar cómo representar los nodos creados en JavaScript, lo anterior no era requerido para la actividad, sin embargo resulta útil el conocer como realizar este procedimiento, fue todo un reto lograrlo ya que se presentaron algunas dificultades al incluir el .js que permitiera ejecutar o crear el grafo con sus aristas y nodos, en resumen la actividad fue interesante y sobre todo dejó cosas nuevas.