



## Ingeniería en Sistemas de Información

### Sintaxis y Semántica de los Lenguajes:

#### Primer Entrega

#### Trabajo Práctico Integrador

Grupo: N°1

Integrantes
Juan Mercurio Badii
Nazarena Abdel Nour
Tomás Ezequiel Giacoboni
Lucas Nicolás Koch
Lucas Sebastián Battaglia

**Sintaxis y Semántica del Lenguaje****Trabajo Práctico II**

**Objetivo:** Aplicar los conocimientos teóricos en el diseño de un compilador básico de Lenguaje de Programación

Para ello, cada equipo, definirá un lenguaje básico, compuesto mínimamente:

- Uno o dos tipo de dato
- Identificadores con longitud máxima de 16 caracteres.
- Los identificadores deberán tener una convención predefinida
- Mínimo 2 sentencias, debiendo contar con convención de fin de sentencia , y de apertura y cierre.
- Cuatro ( 4) palabras reservadas
- Alfabeto a elección del equipo.

**Primer Entrega: Gramática Léxica**

- a. Diseñar las categorías léxicas correspondiente al lenguaje elegido
- b. Formalizar cada lenguaje regular con Autómatas Finitos y Expresiones regulares

**Fecha de Entrega: Julio 2,2021**

**DESARROLLO**

- Tipo de datos (mínimo dos): **entero, cadena**
- Identificadores con longitud máxima de 16 caracteres
- Los identificadores deberán tener una convención predefinida: **Deben arrancar con letra mayúscula y los caracteres siguientes pueden ser{[A-Z][a-z][0-9]}**
- Mínimo 2 sentencias, debiendo contar con convención de fin de sentencia, y de apertura y cierre. **Sentencia de asignación y sentencia de suma. Fin de sentencia es el punto (.)**
- Cuatro (4) palabras reservadas: **entero, cadena, si, mientras**
- Alfabeto a elección del equipo:  $\Sigma = \{[A-Z][a-z][0-9]\}$

**a) Diseño de las categorías léxicas**

palabraReservada: entero cadena si mientras

identificadores: identificador NoDigito

identificador dígito

noDigito: [a-z] [A-Z]

dígito: [0-9]

carácterPuntuacion: . , ; ( ) { }

operadores: + \* > < == <= >= != & | =

constantes: constanteEntera

constanteCaracter

**b) Expresiones Regulares:**

- palabra reservada = {entero cadena si mientras}
- identificadores: metaER = {[A-Z]([a-z]|[0-9]|[A-Z])<sup>n</sup> / con 0 < n < 16}

$$ER = (A - Z)^+ [(A - Z)(a - z)(0 - 9)]^*$$

- operadores = {[+ \* > < == <= >= != & | =]}
- constante entera = {[0-9]}
- constante carácter = {[a-z][A-Z]}
- caracteres de puntuación = { . , ; ( ) { } }

## DESARROLLO

Alfabeto = { [A-Z][a-z][0-9]-\_ }

### **Tipo de datos:**

Se eligió implementar los tipos de datos de **entero** y **cadena**.

#### Entero:

Teniendo en cuenta que un dígito es:

- <dígito> ::= 0|1|2|3|4|5|6|7|8|9

Lo que definimos como entero sería lo siguiente:

- <entero> ::= <dígito>|<entero><dígito>

**Cadena:**

Teniendo en cuenta que un carácter puede ser cualquier elemento del alfabeto:

- $\langle \text{caracter} \rangle = \text{cualquier elementos del alfabeto}$
- $\langle \text{cadena} \rangle ::= \langle \text{caracter} \rangle | \langle \text{cadena} \rangle \langle \text{caracter} \rangle^*$

**Identificadores:**

Los identificadores deben tener una longitud máxima de 16 y deben iniciar con una letra mayúscula.

Para cumplir esta condición podemos definir la siguiente *metaER*:

- $\{ [A-Z] ([a-z] | [0-9] | [A-Z])^n \} / \text{con } 0 < n < 16 \}$

**Sentencias:**

Se eligió la sentencia de *asignación* y la sentencia de *suma*. Para indicar la finalización de una sentencia se utiliza el punto (.)

- La sentencia de asignación puede ser realizada con todo tipo de datos
- La sentencia de suma solo se puede realizar entre enteros

$\langle \text{asignacion} \rangle ::= \langle \text{identificador} \rangle = \langle \text{digito} \rangle | \langle \text{cadena} \rangle$

$\langle \text{suma} \rangle ::= \langle \text{identificador} \rangle | \langle \text{dígito} \rangle + \langle \text{identificador} \rangle | \langle \text{dígito} \rangle$

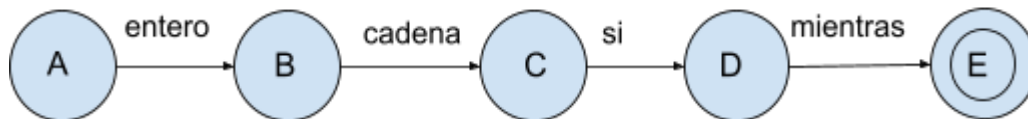
**Palabras Reservadas:**

- **entero:** crea una variable de tipo número entero.
- **cadena:** crea una variable de tipo cadena, es decir una secuencia de caracteres entre " ".
- **si:** es una estructura condicional basada en el "if" del lenguaje C
- **mientras:** es una estructura cíclica, que puede ejecutar una o más líneas de código, basada en el "while" del lenguaje C

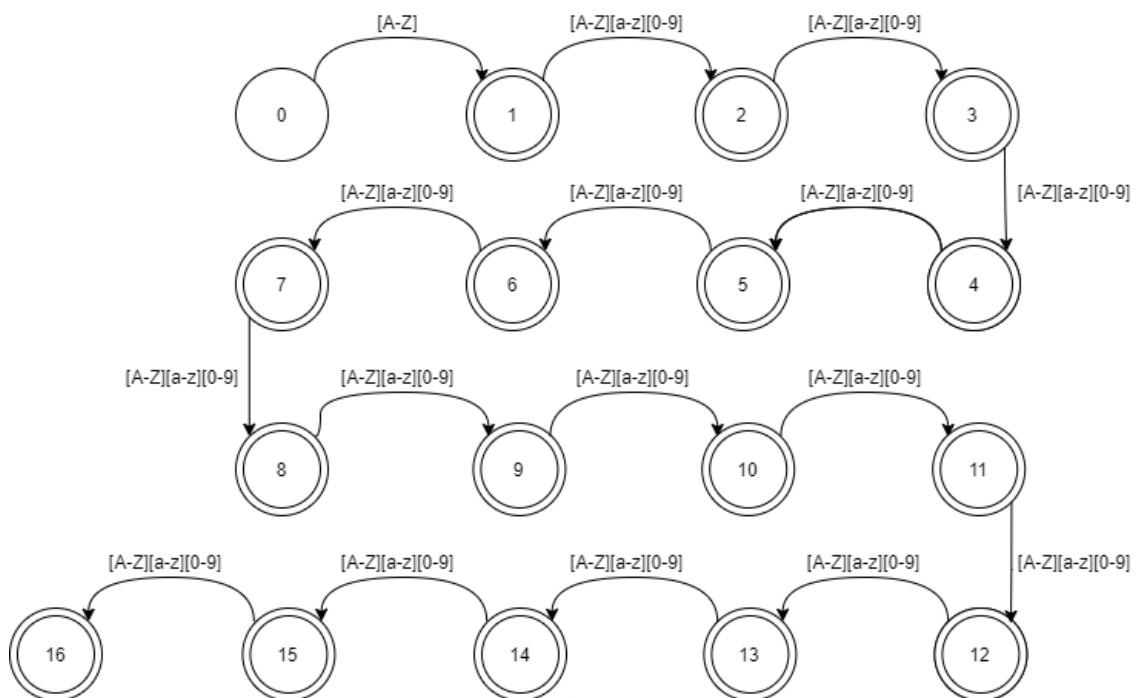
$ER = \{ \text{entero} + \text{cadena} + \text{si} + \text{mientras} \}$

## AUTÓMATAS FINITOS

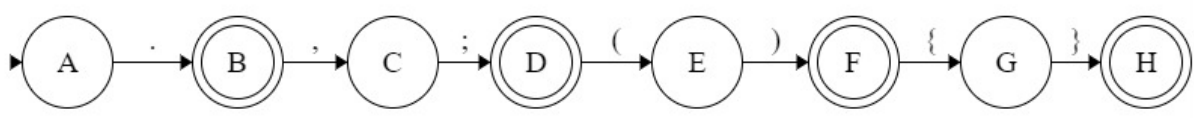
Palabras reservadas:

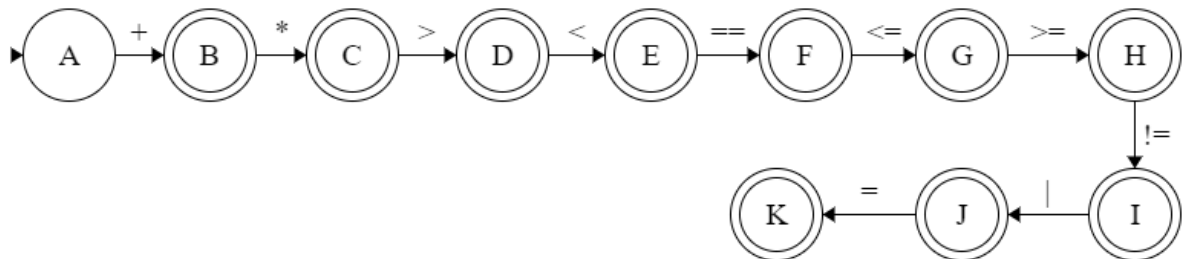


Identificadores:



Caracteres de puntuación:



**Operadores:****Constante enteras:**