

Instituto Tecnológico de Chetumal
ISIC I7A

Materia:

Lenguajes y Autómatas

Maestro:

May Canche Isaías

Proyecto Antlr4

Equipo:

- Alexis Eduardo Cauich Caballero.
- Kevin Jesús Yam Sánchez.

Contenido

Resumen:	3
Funcionalidades:	4
El print:	4
El input:	4
Definición de variables:	5
Ciclos y Condicionales:	6
Ciclos:	6
Condicionales:	7
Comparaciones Booleanas:	8
Operaciones Matemáticas:	9
Suma:	9
Resta:	9
Multiplicación:	9
División:	9
Potenciación:	10
Raíz:	10
Sobranate:	10
Definición de funciones:	11
Agrupación:	12
Paréntesis:	12
Llaves	12
Corchetes:	12
Manejo de arreglos y matrices:	13
Arreglos	13

Resumen:

Se optó principalmente por una traducción del lenguaje de programación a español, y la simplificación de la sintaxis de esta misma con el objetivo de hacerlo más entendible a la hora de ordenar las operaciones. Es necesario aclarar que es un modelo de lenguaje para programación estructurada.

Funcionalidades:

El print:

Sera remplazado por la palabra clave **Escribe** y lo que venga luego de este será lo considerado para imprimir, delimitado por ()

Escribe ('Hola mundo')

El input:

Sera remplazado por la palabra **Lee**, seguido de () donde se pondrá el mensaje que quiere verse en consola:

- Enteros:

Lee ('Proporcione un numero: ') en entero NumEntero

- Decimales:

Lee ('Proporcione un numero: ') en decimal NumEntero

- Cadenas:

Lee ('Proporcione un numero: ') en palabra NumEntero

Definición de variables:

Los nombres de las variables validas estarán delimitados por la expresión '[a-zA-Z][a-zA-Z0-9]*', es decir serán válidas aquellos nombres que empiezan con una letra (mayúscula o minúscula), seguido de mas letras o números. También se sustituyó el carácter de igualdad = por ->, sin embargo, ambos son válidos.

- Enteros:

Usando la palabra reservada '**entero**' para su definición, seguido de un identificador (ID) y luego '->' para después poner el 'valor':

entero numEntero -> 10

- Flotantes:

Usando la palabra reservada '**decimal**' para su definición, seguido de un identificador (ID) y luego '->' para después poner su 'valor'.

decimal numDecimal -> 5.5

- Cadenas:

Usando la palabra reservada '**palabra**' para su definición, seguido de un identificador (ID) y luego '->' para después poner su 'valor'.

palabra cadena -> 'Hola mundo'

- Boléanos:

Usando la palabra reservada '**esVerdad**' para su definición, seguido de un identificador (ID) y luego '->' para después poner su 'valor', el cual vareara entre SI y NO.

esVerdad verdadero -> si

esVerdad falso -> no

es importante decir que los valores aceptados como valor valido booleano deberán ser minúsculas para ser aceptados.

Ciclos y Condicionales:

Ciclos:

- For:

Estos ciclos serán desencadenados por la palabra reservada 'Repetir' seguido del **número de repeticiones**, luego ira el ciclo a repetir delimitado por 'Hacer', 'Fin'.

Repetir 5 veces

Hacer

Contenido

fin

Manipulación del contador interno del for: usando la palabra reservada 'actual', es posible utilizar el valor interno que usa el for para manejar los ciclos.

Repetir 5 veces

Hacer

entero numero -> actual

Fin

- While:

Estos bucles se basan en la repetición mientras se cumpla una condición, entonces, usando la palabra reservada 'Mientras' seguido de '(condición)' se producirá este ciclo, y el contenido del ciclo estará delimitado por 'Hacer' y 'Fin'

Mientras (numero > 5)

Hacer

contenido

Fin

Condicionales:

- If:

Para el manejo de este condicional se usará la palabra reservada 'SI', seguido de (condición), y luego las palabras 'Hacer' y 'Fin' para delimitar las acciones.

En caso de necesitar la cláusula **else**, se implementará la palabra 'SINO', delimitando sus acciones con 'Hacer' y 'Fin'.

```
SI (numero > 5)
```

```
  Hacer
```

```
    acciones
```

```
  Fin
```

```
SINO
```

```
  Hacer
```

```
    acciones
```

```
  Fin
```

Comparaciones Booleanas

Estas no se cambiarán y seguirán usándose las comparaciones normales.

<: menor-que

>: mayor-que

<=: menor-igual-que

>=: mayor-igual-que

==: igual-que

!=: diferente-que

Operaciones Matemáticas:

Se busca cambiar los operadores matemáticos con su 'traducción' para ayudarlos a adentrarse a la lógica detrás de las operaciones matemáticas, utilizando un lenguaje que usan de forma cotidiana. Sin embargo, no se planea borrar la posibilidad de usar los operadores con tal que no pueda resultar perjudicial para los niños en su transición académica.

Suma:

Se busca implementar la palabra reservada **mas** para sustituir el operador lógico **+**.

entero numero1 -> 5 **mas** 5

Resta:

Se busca implementar la palabra reservada **menos** para sustituir el operador lógico **-**.

entero numero1 -> 10 **menos** 5

Multiplicación:

Se busca implementar la palabra reservada **por** para sustituir el operador lógico *****.

entero numero1 -> 5 **por** 2

División:

Se busca implementar la palabra reservada **entre** para sustituir el operador lógico **/**.

entero numero1 -> 10 **entre** 5

Potenciación:

Se busca implementar la palabra reservada **potencia** para sustituir el operador lógico ^.

entero numero1 -> 2 **potencia** 2

Raíz:

Se busca implementar la palabra reservada **raiz** para sustituir el operador lógico √.

decimal numero1 -> 25 **raiz** 5

Sobrante:

Se busca implementar la palabra reservada **sobrante** para sustituir el operador lógico %.

decimal numero1 -> 25 **raiz** 5

Definición de funciones:

Para la definición de funciones se usa la palabra reservada **receta** para la construcción de la función, seguido de su **identificador** y sus **ingredientes** (sus argumentos) seguido de '**Hacer**' y '**Fin**' para englobar las acciones dentro de la función.

```
Receta funcion (entero numero1, entero numero2)
```

```
Hacer
```

```
<acciones>
```

```
Fin
```

Es posible especificar un return, el cual se sustituirá por la palabra reservada **regresar**.

```
Receta suma (entero numero1, entero numero2)
```

```
Hacer
```

```
entero resultado -> numero1 mas numero2
```

```
regresar resultado
```

```
Fin
```

Para el uso de las funciones se conserva el modo de llamado estándar en otros lenguajes:

```
funcion (numero1,7)
```

```
entero resSuma = Suma (2,4)
```

Agrupación:

Paréntesis:

El uso de paréntesis se mantiene en su mayoría, en operaciones matemáticas (4 mas 5) por 3, o para definir los argumentos de una función, al igual que en la llamada de esta misma., y en su uso para los ciclos Mientras y el condicional SI.

Llaves

Las llaves fueron sustituidas por las palabras Hacer y Fin en su uso como determinador del contenido de una función.

Corchetes

Los corchetes se mantienen en sus usos en arreglos.

Manejo de arreglos:

Arreglos

La sintaxis es similar a las de las variables:

Arreglo entero arrNum -> [1,2,3,4,5]

Arreglo entero arrNum -> []

el acceso a las variables:

arrNum [1] #retornara 2

no se cambió la existencia del puesto 0 en un arreglo, pues es un concepto importante en programación y podría ser perjudicial.

Operaciones:

- Insertar:

- Insertar al final:

Arreglo entero arrNum -> [1,2,3]

arrNum. Agregar (4)

[1,2,3,4] # Salida

- Inserta en una posición específica:

Arreglo entero arrNum -> [1,2,3]

arrNum. Agregar-en(4, 1)

[1,4,2,3] # Salida

- Eliminar:

- Eliminar específico:

Arreglo entero arrNum -> [1,2,3]

arrNum. Elimina(2) #Elimina el primer 2

[1,3] # Salida

- Elimina la posición específica:

Arreglo entero arrNum -> [1,2,3]

arrNum.Elimina-en (0)

[2,3] # Salida