GOLang y JavaScript

***Tema: “Lenguajes de programación”***

**Integrantes**:

* Boido, Tomás
* Choque Mamani, Ariel
* Lannert, Nicolás
* Tamborini Criscueli, Agustín Ariel

**Materia**: [Sintaxis](mailto:tisaempresa@gmail.com) y Semántica del Lenguaje **Año**: 2019

**GoLang**

1) Generalidades del lenguaje, características principales y algo de historia.

2) Gramática léxica. Token y comentarios.

3) Literales de cadena y array.

1. Generalidades:

* Es un lenguaje de programación fuertemente tipado.
* Es de programación concurrente y compilado.
* Se creó en el 2009 y desarrollado por Google.
* Es de código abierto.
* Cuenta con su propio recolector de basura.

Ventajas:

* + Es un lenguaje simple de entender.
  + Soporta miles de conexiones simultáneas.
  + Su velocidad y rendimiento.

1. En GoLang podemos realizar comentarios de dos formas distintas:

* A través del carácter “//”, es el comentario de una línea
* Y para realizar comentarios de línea múltiple se utiliza la secuencia “/\*” para iniciar y “\*/” para terminar.

**Identificadores**

Los identificadores en este lenguaje son una secuencia de una o más letras y dígitos. El primer carácter de un identificador debe ser una letra.

identificador = letra { letra | dígito\_unicode } .

**Constantes**

Un entero literal es una secuencia de dígitos que representa una constante entera.

Un número de coma flotante literal es una representación decimal de una constante de coma flotante. Tiene una parte entera, un punto decimal, una parte fraccional y una parte exponente.

Un imaginario literal es una representación decimal de la parte imaginaria de una constante compleja. Siempre termina con i.

**Las palabras reservadas**

break default func interface select

case defer go map struct

chan else goto package switch

const fallthrough if range type

continue for import return var

**Operadores y delimitadores**:

+ & += &= && == != ( )

- | -= |= || < <= [ ]

\* ^ \*= ^= <- > >= { }

/ << /= <<= ++ = := , ;

% >> %= >>= -- ! ... . :

&^ &^=

**EXTRA BNF:**

Producción = nombre\_producción "=" [ Expresión ] "." .

Expresión = Alternativa { "|" Alternativa } .

Alternativa = Término { Término } .

Término = nombre\_producción | símbolo [ "…" símbolo ] | Grupo | Opción

| Repetición .

Grupo = "(" Expresión ")" .

Opción = "[" Expresión "]" .

Repetición = "{" Expresión "}" .

1. Los literales de cadena son cadenas constantes a partir de una secuencia de concatenación de cadenas.

Estas se pueden hacer de 2 formas:

* Cadenas literales crudas: son secuencia de caracteres entre comillas inversas como `abc`. Esto permite que con sola esas secuencias puedas escribir string de múltiples línes.
* Cadenas literales interpretadas: son secuencia de comillas dobles.

Los arrays en GoLang se declaran de esta manera: “var a [5] int;”

Este ejemplo indica que es un arreglo de 5 posiciones de tipo entero.

Y para obtener su longitud es de esta forma “len(a);”

**JAVASCRIPT**

**Características del lenguaje**

JavaScript, es uno de los más potentes e importantes lenguajes de programación en la actualidad, por tres enfoques claros: es útil, práctico y está disponible en cualquier navegador web. Nació como un lenguaje sencillo destinado a añadir algunas características interactivas a las páginas web.

El poder de JavaScript está disponible principalmente en lado Front End (diseño y aspecto visual de un sitio web), agregando mayor interactividad a la web.

1. Es liviano.
2. Multiplataforma, ya que se puede utilizar en Windows, Linux o Mac o en el navegador de tu preferencia.
3. Es Imperativo y estructurado, mediante un conjunto de instrucciones indica al computador qué tarea debe realizar.
4. Las variables no requieren declarar a qué tipo de dato pertenecen.
5. Prototipado, debido a que usa prototipos en vez de clases para el uso de herencia.
6. Orientado a objetos y eventos.
7. Es Interpretado, no se compila para poder ejecutarse.
8. Es dinámico, responde a eventos en tiempo real.

**Identificadores**

Un identificador en JavaScript debe comenzar con una letra, un guión bajo o un símbolo de dólar ($). La letra puede ser mayúscula o minúscula, sin embargo, cabe destacar que el lenguaje es sensible a mayúsculas y no será lo mismo “var a” que “var A”. Luego puede continuar con cualquiera de los mismos y también con números.

identificador = primerCaracter restoCaracteres

primerCaracter = (letraMayuscula|letraMinuscula|\_|$)

restoCaracteres = (primerCaracter|digitoDecimal|digitoUnicode)\*

**Palabras Reservadas**

A continuación brindaremos las palabras reservadas que se encuentran en el motor EcmaScript 6, versión actual con la que se maneja JavaScript desde 2015.

break - case - catch - continue - const - debugger - default - delete - do - else - export - extends - finally - for - function - if - import - in - instanceof - let - new - return - super - switch - this - throw - try - typeof - var - void - while - whit - yield

**Signos de Puntuación**

A continuación brindaremos los signos de puntuación que utiliza JavaScript:

( ) { } ; : , [ ] .

**Operadores**

Dentro de JavaScript podemos encontrar diversos tipos de operadores como:

Operadores de asignación: = += -= \*= /= %= \*\*= <<= >>= >>>= &= ^= |=

Operadores de comparación: == != === !== > >= < <=

Operadores aritméticos: % ++ -- - + \* / \*\*

Operadores lógicos: && || !

Estos son únicamente los principales operadores y los más utilizados. Sin embargo, el lenguaje cuenta con aún mayor cantidad de estos, así que les brindamos el link correspondiente para que puedan conocerlos: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Expressions_and_Operators#Bitwise>

**Constantes**

Una constante es un tipo de variable diferente a las variables definidas normalmente con var o let, ya que el valor de estas no podrá ser modificado a través de reasignaciones una vez que se ha creado. La forma de definir una constante en JavaScript es: const variable = ...;

**Comentarios**

Podemos acceder a comentar una línea de la siguiente forma: //

O bien podemos hacer un comentario multilínea de la siguiente forma: /\* \*/

**Arrays**

El objeto Array de JavaScript es un objeto global que es usado en la construcción de arreglos, que son objetos tipo lista de alto nivel. Su prototipo tiene métodos para realizar operaciones de recorrido y mutación. Ni la longitud ni los tipos de los elementos del array son fijos. Dado que la longitud de un array puede cambiar en cualquier momento, y los datos pueden almacenarse en ubicaciones no contiguas en él.

SINTAXIS

[elemento0, elemento1, ..., elementoN]

new Array(elemento0, elemento1[, ...[, elementoN]])

new Array(longitudDelArray)

**Strings**

Es un objeto que representa una serie de caracteres dentro de una cadena. El objeto String encapsula el tipo de dato de la cadena primitiva con diferentes métodos.

SINTAXIS

new String(cadena)

let/var/const variable = “Cadena de Texto”

let/var/const variable = ‘Cadena de Texto’

let/var/const variable = `Cadena de Texto ${variable}`