

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERIA
ORGANIZACIÓN DE LENGUAJES Y COMPILADORES 1

MANUAL TECNICO

ERICK ABDUL CHACON BARILLAS

201807169

SECCION: C

TUTOR: MYNOR OCTAVIO PILO TUY

Type Wise

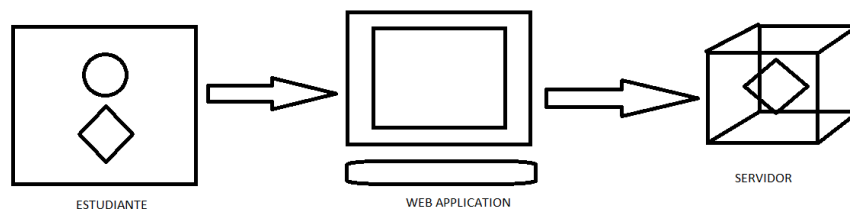
Type Wise es un lenguaje de programación que consiste en un lenguaje de programación para que los aspirantes del curso Introducción a la Programación y Computación 1 puedan familiarizarse y aprender a programar como son las generalidades de un lenguaje de programación. Cabe destacar que este programa será utilizado para generar sus primeras practicas de laboratorio del curso antes mencionado. Y del ámbito de la computación.

Requerimientos mínimos del sistema:

- Windows 7, 8, 10.
- Graphviz-2.49.3 (64-bit).
- Nodejs v18.15
- Jison 0.4.18

Funcionamiento:

El programa cuenta con una arquitectura cliente-servidor, esto con el objetivo de que pueda separar los servicios administrados por el intérprete, de la aplicación cliente que se mostrara al usuario final (estudiante de programación).



Entorno de trabajo:

El programa cuenta con los siguientes entornos de trabajo.

1. Editor: proporciona funcionalidades, características y herramientas que serán de utilidad al usuario. La función principal del editor será el ingreso del código fuente que será analizado. En este se podrán abrir diferentes archivos al mismo tiempo y deberá de mostrar la línea actual.
2. Funcionalidades: el programa tiene las siguientes funcionalidades:
 - Crear archivos: El editor deberá ser capaz de crear archivos en blanco.
 - Abrir archivos: El editor deberá abrir archivos .tw.
 - Guardar archivo: El editor deberá guardar el estado del archivo en el que se estará trabajando.
 - Eliminar pestaña: Permitirá cerrar la pestaña actual.
3. Características: El programa cuenta con múltiples pestañas con la finalidad de ver y abrir los archivos de prueba en la aplicación.
4. Herramientas: El programa cuenta con lo siguiente
 - Ejecutar: hace el llamado al interprete, el cual se hace cargo de realizar los análisis léxico y sintáctico. Este también

genera un árbol de sintaxis abstracta (AST) se genera una imagen del árbol al momento de realizar el análisis, también cuenta con el reporte de la tabla de símbolos esta a diferencia de los otros reportes muestra todo el contenido ya sea variables, métodos, funciones, entornos, etc.

5. Área de consola: En esta área se mostrarán los resultados, mensajes y todo lo que sea indicado dentro del programa.

Descripción del Lenguaje TypeWise:

Case insensitive	El lenguaje no distingue entre mayúsculas y minúsculas.
Comentarios	Comentarios de una línea: las cuales deberán comenzar con // y terminan con un salto de línea. Comentarios multilínea: las cuales deberán comenzar con /* y terminar con */.
Tipos de Datos	Los tipos de datos que soportara el programa en concepto del tipo de variable se definen como: Entero, Booleano, Cadena, Carácter, Decimal (Double).
Aritmética	Suma, Resta, Multiplicación, División, Potencia, Módulo, Negación Unaria.
Relacionales	Igualación, Diferenciación, Menor que, Menor o igual, Mayor que, Mayor o igual que.
Lógicos	OR, AND, NOT

Ternario	Este operador se le considera como un atajo para la instrucción if.
Declaración y Asignación	Una variable deberá de ser declarada antes de poder ser utilizada. Todas las variables tendrán un tipo de dato y un nombre de identificador. Las variables podrán ser declaradas global y local.
Casteos	El lenguaje aceptara los siguientes casteos: <ol style="list-style-type: none"> 1. Int a Double 2. Double a Int 3. Int a String 4. Int a Char 5. Double a String 6. Char a Int 7. Char a Double
Incremento y Decremento	Los incrementos y decrementos nos ayudan a realizar la suma o resta continua de un valor de uno en uno.
Vectores	Los vectores son una estructura de datos de tamaño fijo que pueden almacenar valores de forma limitada y los valores que pueden almacenar son de un único tipo: int, double, boolean, char y string.
Listas	Las listas son una estructura de datos que pueden crecer de forma iterativa y pueden

	almacenar hasta N elementos de un solo tipo: int , double, boolean, char y string.
Sentencia if	Esta sentencia ejecuta las instrucciones solo si se cumple una condición. Si la condición es falsa, se omiten las sentencias dentro de la sentencia.
Sentencia Switch	Esta sentencia es una estructura utilizada para agilizar la toma de decisiones múltiples, trabaja de la misma manera que lo harían sucesivos if.
Sentencia While	Esta sentencia es cíclica que ejecuta una secuencia de instrucciones mientras la condición de ejecución se mantenga verdadera.
Sentencia For	Esta sentencia es cíclica que nos permite ejecutar N cantidad de veces la secuencia de instrucciones que se encuentra dentro de ella.
Sentencia Do-While	Esta sentencia es cíclica que ejecuta al menos una vez el conjunto de instrucciones que se encuentran dentro de ella y que se sigue ejecutando mientras la condición sea verdadera.
Break	La sentencia break hace que se salga del ciclo inmediatamente, es decir que el código que se encuentre después del break en la misma

	iteración no se ejecutara y este se saldrá del ciclo.
Continue	La sentencia continue puede detener la ejecución de la iteración actual y saltar a la siguiente. La sentencia continue siempre debe de estar dentro del ciclo, de lo contrario será un error.
Return	La sentencia return finaliza la ejecución de un metodo o función y puede especificar un valor para ser devuelto a quien llama la función.
Funciones	Las funciones serán declaradas definiendo primero su tipo, luego un identificador para la función, seguido de una lista de parámetros dentro de paréntesis.
Métodos	Los métodos serán declarados haciendo uso de la palabra reservada "void" al inicio, luego se indicará el identificador del método, seguido de una lista de parámetros dentro de paréntesis.
Llamadas a funciones	La llamada a una función especifica la relación entre los parámetros reales y los formales y ejecuta la función.