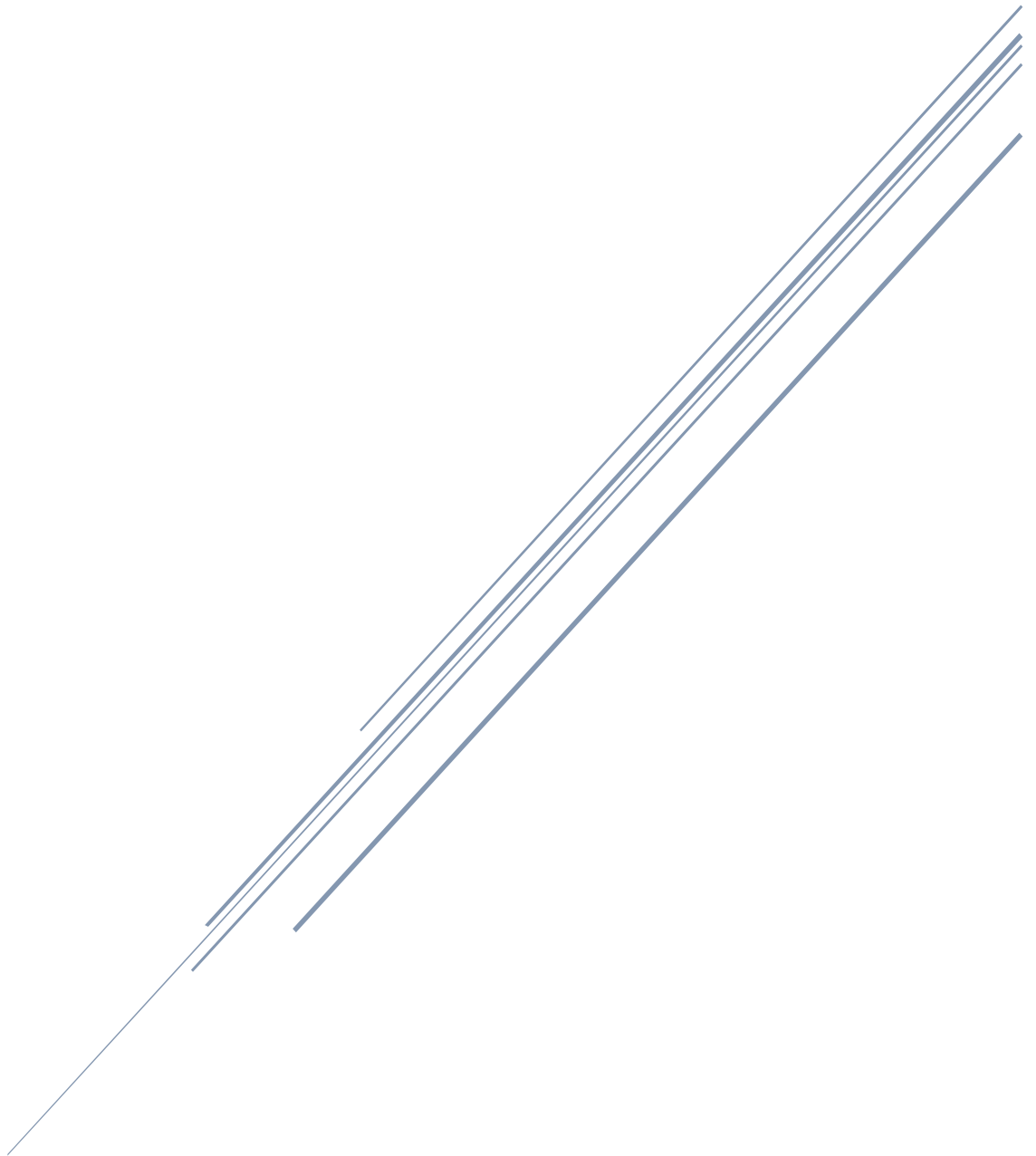


MANUAL TECNICO.

typeSty.



Ariel Rubelce Macario Coronado
Universidad San Carlos de Guatemala.

DESCRIPCION GENERAL DE LA SOLUCION.

TypeSty es una solución para los que están empezando a programar, ya que es amigable y fácil de aprender, también se pueden visualizar los reportes, como la tabla de símbolos y la tabla de errores, también podemos ver el flujo del código por medio de un árbol de análisis sintáctico. Este funciona como un interprete.

REQUERIMIENTOS MINIMOS DEL ENTORNO DE DESARROLLO.

Para del desarrollo debemos de contar con Node Js, en su versión estable, contar con npm y contar con el conocimiento de Javascript, contar un espacio de almacenamiento para el entorno.

DICCIONARIO DE CLASES.

Ámbito: Clase que nos ayuda a manejar los ámbitos.

Analizador: Es aquella clase que nos ayuda a compilar los archivos Analizador.json

Método: Es que nos ayuda a manejar los métodos creados a la hora de copilar.

Símbolo: Clase que nos ayuda a manejar los símbolos creados.

TablaSímbolos: Clase que define un conjunto, que tiene como parámetros, un nombre y un rango.

DICCIONARIO DE METODOS.

addSimbolo: Agregar símbolos a la tabla de símbolos.

Actualizar: Actualizar un símbolo en la tabla de símbolos.

Aritmética: Realiza una operación entre dos operando.

Asignacion: Se asigna un valor a una variable que esta en la tabla de símbolos.

Bloque: Donde hacemos la llamada de las instrucciones.

CicloDoWhile: Creamos el Ciclo Do While.

CicloWhile: Creamos el Ciclo While.

DecMetodo: Declaramos un método en la tabla de métodos.

DecParametro: Declaramos los parámetros de un método.

Declaración: Declaramos una variable y se guarda en la tabla de símbolos.

Exec: Creamos el exec que dará vida a nuestra programa.

existeSimbolo: Verificar si existe el símbolo en la tabla de símbolos.

existeSimboloAmbitoActual: Verificar si existe el símbolo en el ámbito actual.

For: Declaramos el For una sentencia cíclica.

Global: Donde vamos hacer 3 pasadas y para verificar si no existe mas de un exec.

getEntornoString: Para obtener el ámbito donde se esta trabajando.

getSimbolo: Retornar Símbolo de la tabla de simbolos.

graficarSentencia: Graficar el sentencia enviada por la instrucción.

graficarNuevaOpBinaria: Graficar una operación binaria.

graficarExec: Graficamos el exec.

graficarLlamadaFuncMetodo: Graficamos el método.

graficarPrint: Graficamos el print.

graficarDecMetodo: Graficamos la declaración de un método.

graficarDecVar: Graficamos la declaración de una variable.

graficarAsgVar: Graficamos la asignación de un variable.

graficarWhile: Graficamos el While.

graficarDoWhile: Graficamos el Do While.

graficarFor: Graficamos el For.

Instrucción: Aquí se hace la ejecución de todas las instrucciones.

Lógica: Realizamos una operación Lógica.

Operación: Donde verificamos que tipo de operadores realizan la operación.

Relacional: Realizamos una operación Relacional.

Sentencialf: Sentencia de control, este es la función if.

SentencialfElse: Sentencia If pero el detalle es que acá hacemos uso del if.

SentencialfElseIF: Sentencia If pero el detalle es que podemos tener varios else fi y también un else.

SentenciaSwthCase: Hacemos uso de la sentencia de control Switch default.

SentenciaSwitchD: Hacemos uso de la sentencia de control Switch Case.

Print: podemos realizar un print