

## Segundo proyecto

Dado el éxito de su analizador léxico los inversionistas decidieron hacer crecer el proyecto. El analizador léxico debe de modificarse para poder reconocer la siguiente tabla:

Patrón	Tipo
<b>funcion</b>	<b>Palabra Reservada</b>
<b>principal</b>	<b>Palabra Reservada</b>
<b>retornar</b>	<b>Palabra Reservada</b>
<b>vacio</b>	<b>Palabra Reservada</b>
variable	Palabra Reservada
entero	Palabra Reservada
decimal	Palabra Reservada
booleano	Palabra Reservada
cadena	Palabra Reservada
<b>carácter</b>	<b>Palabra Reservada</b>
si	Palabra Reservada
sino	Palabra Reservada
mientras	Palabra Reservada
<b>para</b>	<b>Palabra Reservada</b>
hacer	Palabra Reservada
<b>imprimir</b>	<b>Palabra Reservada</b>
VERDADERO	Boolean
FALSO	Boolean



+	Operador
-	Operador
*	Operador
/	Operador
%	Operador
=	Operador
==	Operador
<	Operador
>	Operador
>=	Operador
<=	Operador
(	Agrupación
)	Agrupación
{	Agrupación
}	Agrupación
"	Signo
;	Signo
Número seguido de uno o más números	Número entero
Uno o más números seguido de punto seguido de uno o más números	Número flotante
letra seguido de cero o más letras o números	Identificador
<b>comillas seguidas de cero ó más letras o símbolos, terminando con comillas.</b>	<b>cadena</b>
<b>comilla simple seguida de una letra o símbolo terminando con una comilla simple</b>	<b>carácter</b>



	<b>comentario</b>
<b>diagonal asterisco seguido de cualquier carácter terminando con asterisco diagonal</b>	<b>comentario</b>

## Gramática

El proyecto debe de reconocer una gramática que tiene la siguiente estructura:

```
funcion principal{  
    variable entera x = 5;  
}
```

La función principal es obligatoria y es donde el programa inicia puede tener muchas instrucciones dentro de ella, incluyendo llamada a funciones. Se pueden realizar llamadas a funciones pero estas deben de definirse antes de usarse, por ejemplo:

```
funcion f1( entero x){  
    imprimir ("el número es " + x);  
}  
  
funcion principal{  
    variable entera x = 5;  
    f1(x);  
}
```

Las instrucciones se resumen a continuación:

- imprimir (); Dentro de los paréntesis puede venir la concatenación de variables, strings o char. La concatenación se realiza con signos +. Ejemplo imprimir (a + b + "hola mundo");
- la asignación de una variable se puede realizar como en el lenguaje de programación java. Con la diferencia que se utilizan las palabras reservadas del lenguaje. Se pueden definir de todos los tipos mencionados arriba. Ejemplo: variable entera x = 5;
- Operaciones matemáticas. Para esto pueden utilizarse los paréntesis para agrupar operadores. ejemplo:  $x = (x * x / (5 + 5)) / (1 + 2)$ ;
- para los casos del if, for, while, la gramática es la misma de java solo que con el uso de palabras reservadas del lenguaje.

## Tabla de símbolos

Debe llevar el control de la tabla de símbolos. Utilizar de manera adecuada el uso de tabla de símbolos para el manejo del ámbito de las variables.

## Ámbito

El ámbito para este lenguaje es dinámico. Este estará definido por el uso de las llaves, de la misma manera como funciona java.

## Recursividad

El lenguaje puede hacer llamadas recursivas.

## Intérprete

El programa debe de mostrar en un apartado la salida del programa que se está ejecutando.

## Consideraciones importantes

Es importante el manejo de errores léxicos, sintácticos y manejar errores determinado por la tabla de símbolos. Para los errores es importante brindar toda la información necesaria respecto a los errores. Línea, columna, token y que es lo que se esperaba para poder corregir el código.

Además es importante que el programa sea intuitivo y que se pueda visualizar en todo momento la salida del programa, así como la tabla de símbolos. Es su decisión si se ejecutará instrucción por instrucción, línea por línea. No se permite que se ejecute todo de un solo y ver los resultados al final. Es importante ver el proceso.

## Restricciones generales

No hay restricciones en cuanto al lenguaje a utilizar. Tendrá mucho valor el orden y programar cada una de las partes del intérprete de una manera adecuada. La mantenibilidad del código será importante, así como las buenas prácticas de programación.

## Documentación

Debe adjuntar toda la documentación que permita entender la forma en que su programa esta funcionando. Diagramas, etc, también incluir todos los requerimientos necesarios para que su código funcione. Se ejecutará en la máquina del profesor así que es importante definir los requisitos.

## Fechas importantes

- Lunes 4 de noviembre debe de subir el enlace git de su repositorio. Este enlace no lo podrá cambiar. Para la calificación se descargara el repositorio del enlace que suba, si cambia el enlace habrá penalización.
- El martes 19 de noviembre a la media noche se descargan los repositorios, debe tomar en cuenta que su repositorio debe ser público. El miércoles 20 de noviembre empezará la calificación.