



Tecnológico de Monterrey

Propuesta del proyecto final de la clase de compiladores

Integrantes:

Diego Eduardo Contreras Zapata
A00817441

Luis Salomon Flores Ugalde
A00817435

Febrero 25, 2019

Indice

Visión del proyecto	2
Objetivo principal del lenguaje	2
Requerimientos del lenguaje	2
Tokens	2
Diagramas de sintaxis	5
Funciones especiales	13
Tipos de datos	13
Lenguaje y sistema operativo para el desarrollo	14
Bibliografía	14

Visión del proyecto

Llegar a un lenguaje que permita las principales características de un lenguaje de programación moderno, como creación de clases, herencia simple, arreglos, matrices, ciclos, y funciones.

Objetivo principal del lenguaje

Desarrollar un lenguaje orientado a objetos con elementos básicos que se esperan para poder desarrollar código orientado a objetos.

Requerimientos del lenguaje

Tokens

Lista de tokens que se usarán en el lenguaje

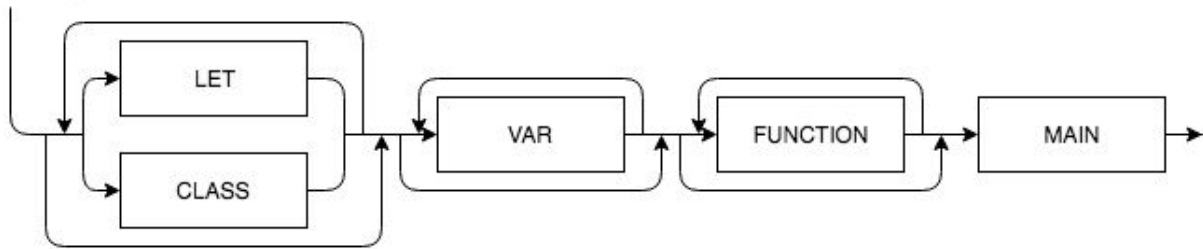
Token	Patron	Lexemas ejemplos
ID	Letra seguida de letras y números	contador, diego, contador2
if	Caracteres i,f	if
else	Caracteres e,l,s,e	else
while	Caracteres w,h,i,l,e	while
input	Caracteres i,n,p,u,t	input
print	Caracteres p,r,i,n,t	print
var	Caracteres v,a,r	var
let	Caracteres l,e,t	let
function	Caracteres f,u,n,c,t,i,o,n	function
class	Caracteres c,l,a,s,s	class
return	Caracteres r,e,t,u,r,n	return
init	Caracteres i,n,i,t	init

main	Caracteres m,a,i,n	main
Int	Caracteres l,n,t	Int
Float	Caracteres F,l,o,a,t	Float
Bool	Caracteres B,o,o,l	Bool
true	Caracteres t,r,u,e	true
false	Caracteres f,a,l,s,e	false
cte_i	Constante numérica entera	10, 20, 3012
cte_f	Constante numérica con punto flotante	10.19, 90.12, 10.0
cte_b	Constante booleana, verdadero o falso.	true, false
cte_s	Cualquier serie de caracteres entre comillas “ ”	“Hello World”, “Hola”
find	Caracteres f,i,n,d	find
append	Caracteres a,p,p,e,n,d	append
size	Caracteres s,i,z,e	size
empty	Caracteres e,m,p,t,y	empty
sort	Caracteres s,o,r,t	sort
pow	Caracteres p,o,w	pow
ceil	Caracteres c,e,i,l	ceil
floor	Caracteres f,l,o,o,r	floor
:	:	:
.	.	.
,	,	,
;	;	;
->	->	->
+	+	+

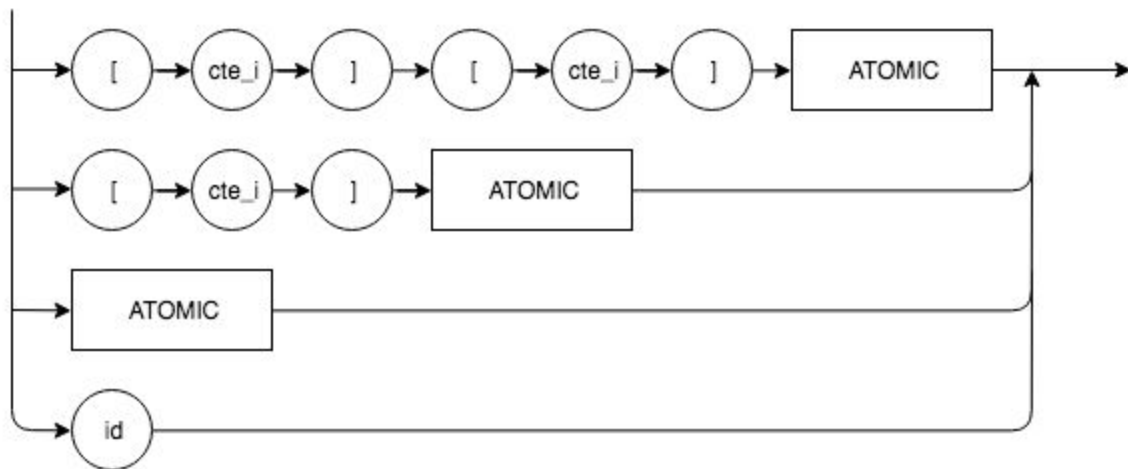
-	-	-
*	*	*
/	/	/
&&	&&	&&
!	!	!
=	=	=
>	>	>
<	<	<
==	==	==
!=	!=	!=
>=	>=	>=
<=	<=	<=
[[[
]]]
{	{	{
}	}	}
(((
)))
//	//	//
/*	/*	/*
*/	*/	*/

Diagramas de sintaxis

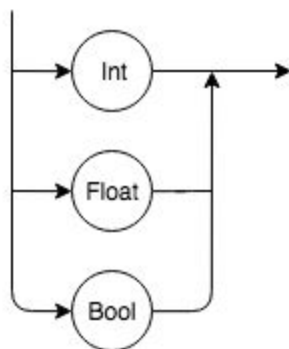
< PROGRAM >



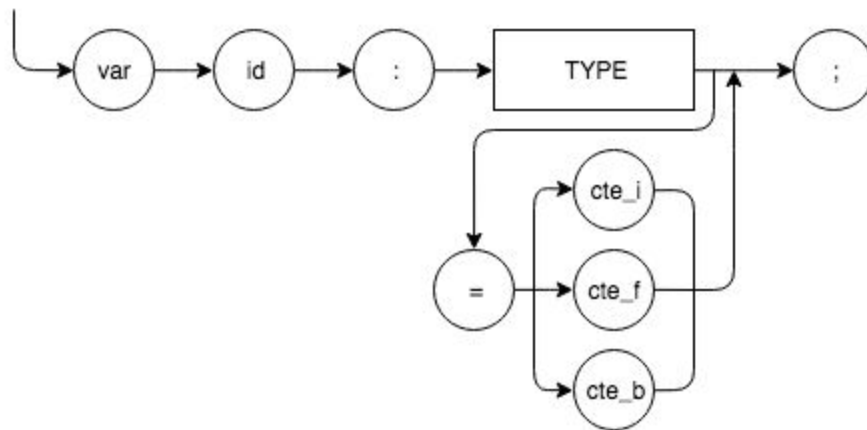
< TYPE >



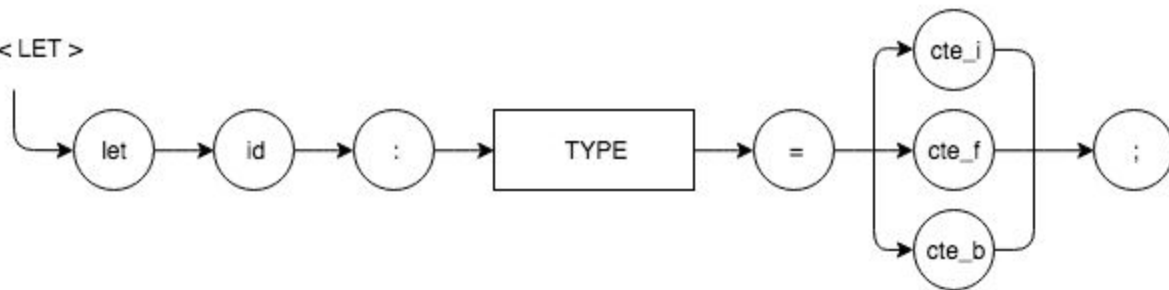
< ATOMIC >



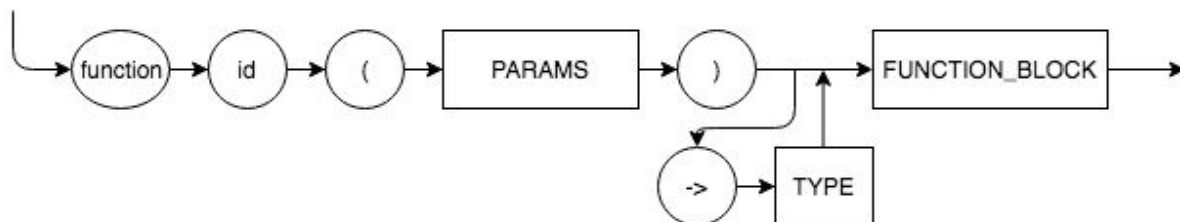
< VAR >



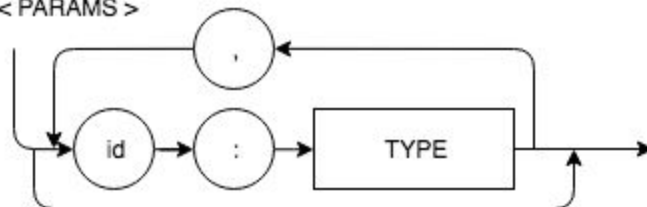
< LET >



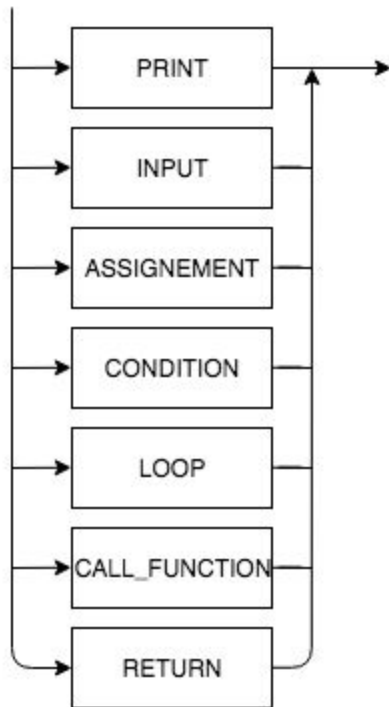
< FUNCTION >



< PARAMS >



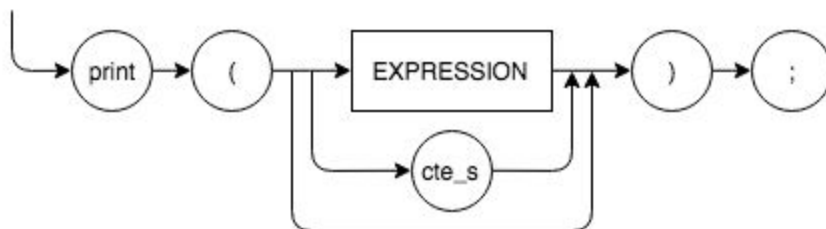
< STATUTE >



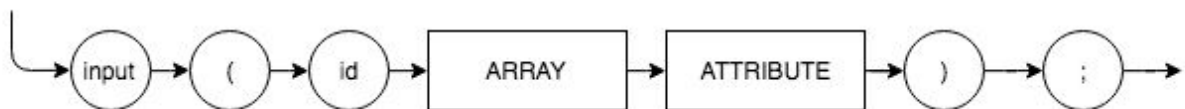
< ASSIGNEMENT >



< PRINT >



< INPUT >



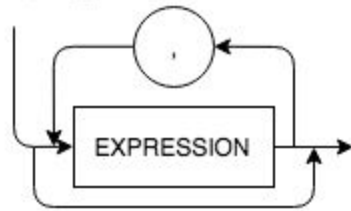
< LOOP >



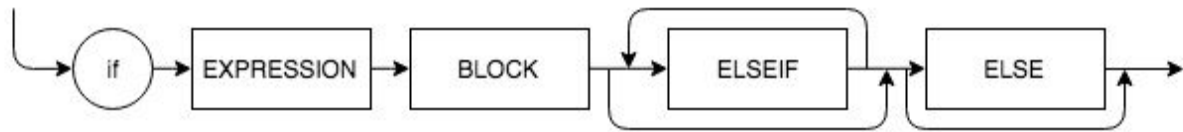
< CALL_FUNCTION >



< CALL_PARAMS >



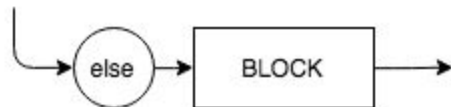
< CONDITION >



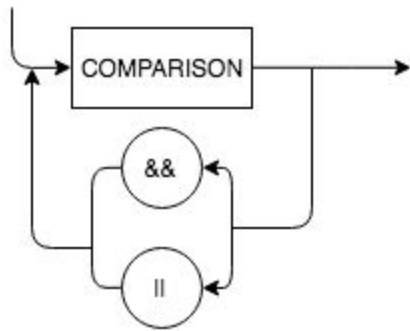
< ELSEIF >



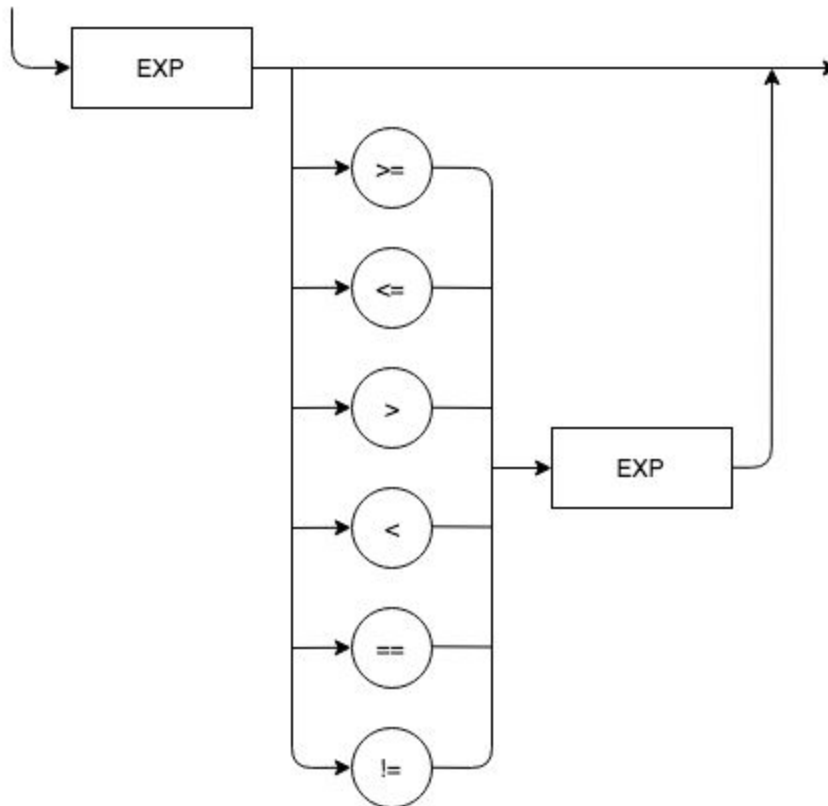
< ELSE >

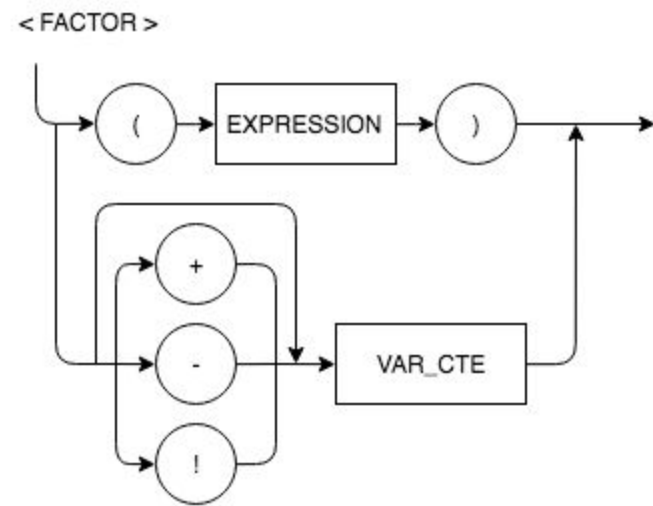
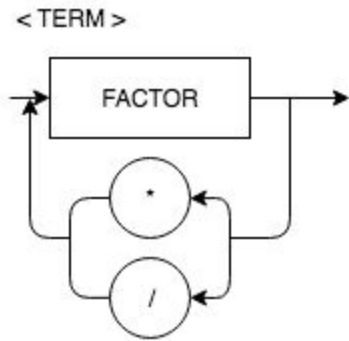
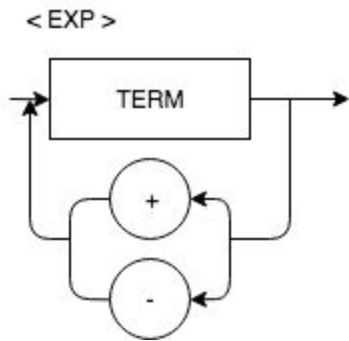


< EXPRESSION >

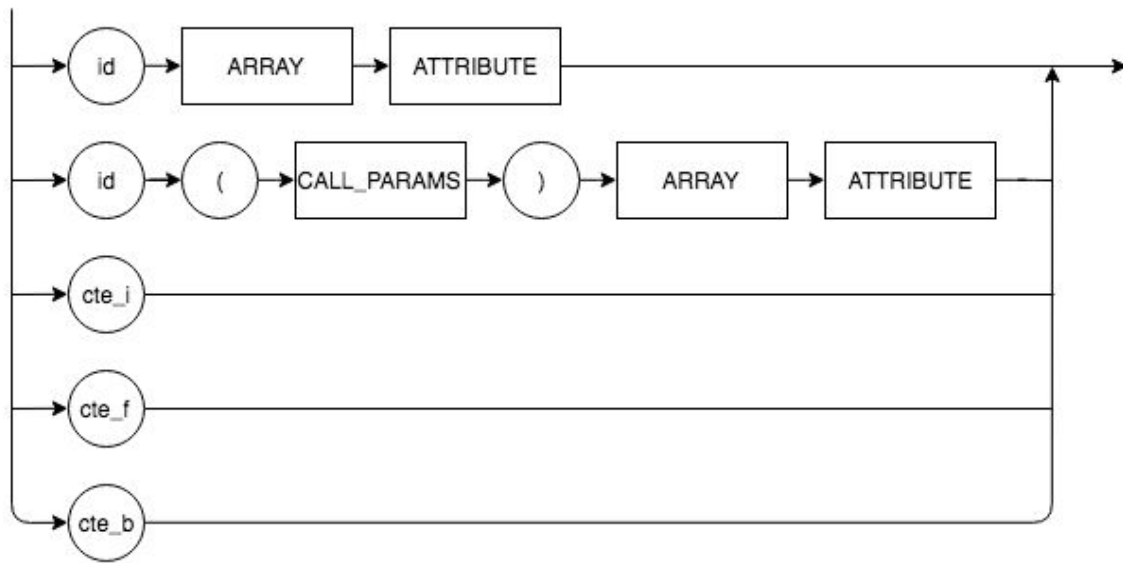


< COMPARISON >

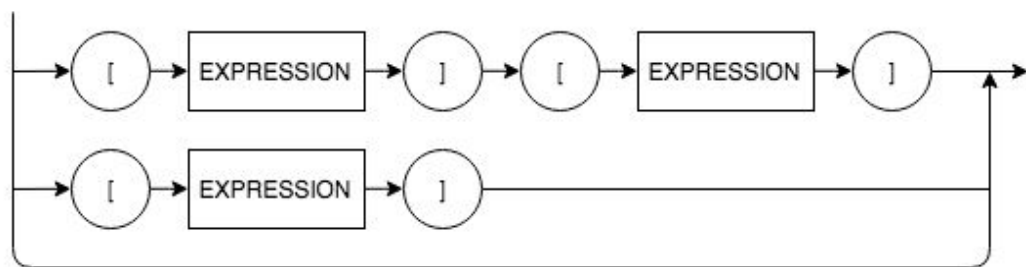




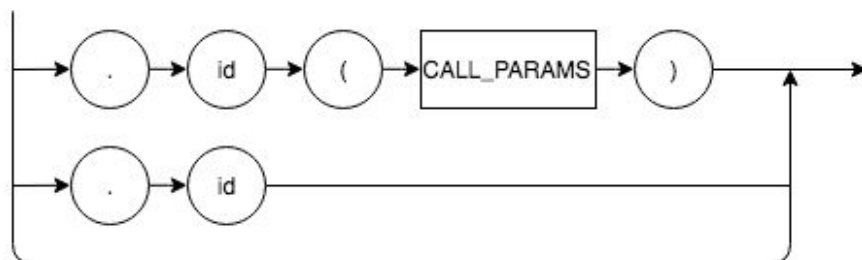
< VAR_CTE >



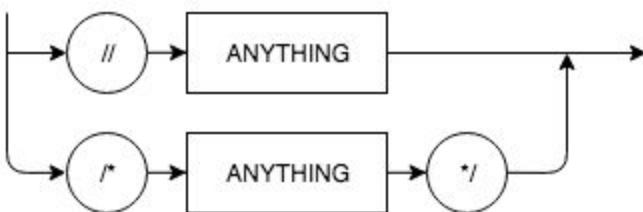
< ARRAY >



< ATTRIBUTE >



< COMMENT >



Características semánticas principales

Los valores de tipo **Int**, podrán tener operaciones con otros valores de tipo **Int** y con valores **Float**.

Los valores de tipo **Float**, podrán tener operaciones con otros valores de tipo **Int** y **Float**.

Los valores de tipo **Bool**, solo podrán interactuar con otros valores de tipo **Bool**.

Los valores de tipo **constante String**, no podrán tener operaciones con ningún valor.

Las variables tienen que ser declaradas en el cuerpo principal de una función. Dentro de los bloques de estatutos `if` o `while` no es posible declarar nuevas variables para usarlas dentro de una función.

Funciones especiales

Cast

`Float ()` : Regresa el valor de tipo `int` como un `float`.

`Int ()` : Regresa el valor tipo `float` como un `int`.

Arreglos

`find (x)` : Regresa la posición del arreglo donde se encuentra `x`, `-1` si no la encuentra.

`append (x)` : Inserta un elemento al final del arreglo.

`size()` : Regresa el tamaño del arreglo.

`empty()` : Regresa un booleano si un arreglo está vacío.

`sort()` : Ordena el arreglo de menor a mayor.

Matemáticas

`pow(x,y)` : Eleva el número `x` a la `y` potencia.

`ceil(x)` : Redondea para arriba el número `x`.

`floor(x)` : Redondea para abajo el número `x`.

Tipos de datos

El lenguaje soportará los siguientes tipos de datos:

- **Int:** Números enteros
- **Float:** Números con punto flotante
- **Bool:** Variable de verdadero o falso
- **Constantes string:** Cualquier serie de caracteres entre comillas

Lenguaje y sistema operativo para el desarrollo

El proyecto será desarrollado en el sistema operativo macOS, y con el lenguaje de python usando las librerías de lex y yacc en PLY.

Bibliografía

<https://www.dabeaz.com/ply/ply.html>

Diagramas hechos en: <https://www.draw.io/>