

Universidad de San Carlos de Guatemala  
División de Ciencias de la Ingeniería  
Ingeniería en Ciencias y Sistemas  
Organización de Lenguajes y Compiladores 2  
Ing. Moisés Granados



# MANUAL TÉCNICO

Proyecto #2



Carné 201830221  
Willians Alberto Orozco López



# Definición de tokens utilizados

Estado **INICIAL**

Token	Lexema	Salta a Estado
%%PY	PYTHON	PYTHON
%%JAVA	JAVA	JAVA
%%PROGRAMA	PROGRAMA	INICIAL
#include	INCLUDE	INICIAL
void	VOID	INICIAL
main	MAIN	INICIAL
(	PARENTESIS_A	INICIAL
)	PARENTESIS_C	INICIAL
{	LLAVE_A	INICIAL
}	LLAVE_C	INICIAL
int	INT	INICIAL
char	CHAR	INICIAL
float	FLOAT	INICIAL
[	CORCHETE_A	INICIAL
]	CORCHETE_C	INICIAL
;	PUNTO_COMA	INICIAL
const	CONSTANTE	INICIAL
+	SUMA	INICIAL
-	RESTA	INICIAL
*	MULTIPLICACION	INICIAL
/	DIVISION	INICIAL
%	MOD	INICIAL





=	ASIGNAR	INICIAL
//	--	COMENTARIO_LINEA
/*	--	COMENTARIO_BLOQUE
<	MENOR	INICIAL
>	MAYOR	INICIAL
==	IGUAL_IGUAL	INICIAL
!=	DIFERENTE	INICIAL
if	IF	INICIAL
else	ELSE	INICIAL
&&	AND	INICIAL
	OR	INICIAL
!	NOT	INICIAL
switch	SWITCH	INICIAL
case	CASE	INICIAL
break	BREAK	INICIAL
:	DOS_PUNTOS	INICIAL
for	FOR	INICIAL
++	MAS	INICIAL
--	MENOS	INICIAL
while	WHILE	INICIAL
do	DO	INICIAL
continue	CONTINUE	INICIAL
break	BREAK	INICIAL

#### Estado **PYTHON**

Token	Lexema	Salta a Estado
%%JAVA	JAVA	JAVA





print	PRINT	PYTHON
println	PRINTLN	PYTHON
input()	INPUT	PYTHON
def	DEF	PYTHON
if	IF	PYTHON
return	RETURN	PYTHON
=	ASIGNAR	PYTHON
==	IGUAL_IGUAL	PYTHON
%	MOD	PYTHON
+	SUMA	PYTHON
-	RESTA	PYTHON
*	MULTIPLICACION	PYTHON
^	POTENCIA	PYTHON
/	DIVISION	PYTHON
continue	CONTINUE	PYTHON
break	BREAK	PYTHON
for	FOR	PYTHON
while	WHILE	PYTHON
do	DO	PYTHON
(	PAR_A	PYTHON
)	PAR_C	PYTHON
,	COMA	PYTHON
<	MENOR	PYTHON
>	MAYOR	PYTHON
>=	MAYOR_IGUAL	PYTHON
<=	MENOR_IGUAL	PYTHON
else	ELSE	PYTHON





:	DOS_PUNTOS	PYTHON
in	IN	PYTHON
range	RANGE	PYTHON
!=	DIFERENTE	PYTHON
and, &	AND	PYTHON
or,	OR	PYTHON
~	NOT	PYTHON
<<	ZERO	PYTHON
>>	SIGNED	PYTHON
true	TRUE	PYTHON
false	FALSE	PYTHON

#### Estado **JAVA**

Token	Lexema	Salta a Estado
%%PROGRAMA	PROGRAMA	INICIAL
public	PUBLIC	JAVA
private	PRIVATE	JAVA
class	CLASS	JAVA
extends	EXTENDS	JAVA
(	PAR_A	JAVA
)	PAR_C	JAVA
{	COR_A	JAVA
}	COR_C	JAVA
void	VOID	JAVA
int	INT	JAVA
String	STRING	JAVA
float	FLOAT	JAVA





char	CHAR	JAVA
boolean	BOOLEAN	JAVA
if	IF	JAVA
>	MAYOR	JAVA
<	MENOR	JAVA
>=	MAYOR_IGUAL	JAVA
<=	MENOR_IGUAL	JAVA
==	IGUAL_IGUAL	JAVA
!=	DIFERENTE	JAVA
&&	AND	JAVA
	OR	JAVA
!	NOT	JAVA
,	COMA	JAVA
:	DOS_PUNTOS	JAVA
else	ELSE	JAVA
print	PRINT	JAVA
println	PRINTLN	JAVA
intinput	INTINPUT	JAVA
floatinput	FLOATINPUT	JAVA
charinput	CHARINPUT	JAVA
while	WHILE	JAVA
do	DO	JAVA
for	FOR	JAVA
=	ASIGNAR	JAVA
;	PUNTO_COMA	JAVA
++	MAS	JAVA
--	MENOS	JAVA





+=	MAS_ASIGNAR	JAVA
-=	MENOS_ASIGNAR	JAVA
true	TRUE	JAVA
false	FALSE	JAVA
return	RETURN	JAVA
switch	SWITCH	JAVA
case	CASE	JAVA
break	BREAK	JAVA
continue	CONTINUE	JAVA

## Gramática

### Código Python

#### Funciones

funciones	funciones funcion
	funcion
funcion	"def" IDENTIFICADOR "(" lista_parametros ")" ":" SALTO INDENT sentencias DEDENT

#### Lista parametros

lista_parametros	lambda
	parametros
parametros	parametro parametrop
parametrop	lambda
	", " parametros





## Sentencias

sentencias	while
	if
	for
	asignar
	print
	"continue" SALTO
	"break" SALTO
	"return" valor SALTO

### While

while	"while" expresion ":" SALTO INDENT sentencias DEDENT
-------	--

### For

for	"for" IDENTIFICADOR "in" "range" "(" rango ")"
-----	--

### Asignar

while	"while" expresion ":" SALTO INDENT sentencias DEDENT
-------	--

### If

if	"if" expresion ":" SALTO INDENT sentencias DEDENT
	"if" expresion ":" SALTO INDENT sentencias DEDENT if_cola

### If cola

if_cola	"else" ":" SALTO INDENT sentencias DEDENT
	elif ":" "else" SALTO INDENT sentencias DEDENT







	elif
--	------

## Elif

elif	"elif" expresion ":" SALTO INDENT sentencias DEDENT
	"elif" expresion ":" SALTO INDENT sentencias DEDENT elif

