



# MANUAL DE USUARIO

Compiladores 1

carlos tenes

## Introducción

Muchas veces en programación es necesario transcribir una publicación dado a un cambio de plataforma o mejoras en otro lenguaje, de ahí nace la necesidad de un transpilador, en el siguiente programa web se incluye un transpilador sencillo de C# a Python usando conceptos web.

## Capturas

Compiladores

Code here!

```
void mifuncion(int a, int b){
    Console.WriteLine("hola mundo");
    Console.WriteLine("hola mundo");
    //mi casa
    Console.WriteLine("<html>");
    Console.WriteLine("lalalala");
    Console.WriteLine("<b2>ooooo</b2>");
    Console.WriteLine("</html>");
}

class MyProgram
{
    void main ()
    {
        int a = 0;
        a=((a-b)*fibonacci(78/a-b*g));
        //un comentario
        /*otro comentario*/
    }
}
```

Compile

Guardar en el PC | Seleccionar archivo | No se eligió archivo

Compiladores

HTML/JSON | Errores | Variables | AST | Python

METODO

CLASS

IF

```
[
  {
    "metodo": {
      "nombre": "mifuncion",
      "parametros": [
        {
          "tipo": "a",
          "tipo": "b"
        }
      ],
      "instrucciones": [
        {
          "print": {
            "valor": {
              "tipo": "V_CADENA",
              "valor": "hola mundo"
            }
          }
        }
      ]
    }
  },
  {
    "print": {
      "valor": {
        "tipo": "V_CADENA",
        "valor": "hola mundo"
      }
    }
  }
]
```

```
def miFuncion(a,b):  
    Print ("hola mundo")  
    Print ("hola mundo")  
    #mi casa  
    Print ('<html>')  
    Print ('jajajaja')  
    Print ('<h2>ooooo</h2>')  
    Print ('</html>')  
  
class MyProgram:  
    def main():  
        a = 0  
        a = ((a - b * c) - fibonacci(78 / a - b * c))  
        #un comentario  
        """ otro comentario """  
  
if ( sueldo>3000 and variable ):  
    Print ("Esta persona debe abonar impuestos")  
  
if __name__=="main"  
    main()
```

[Guardar en el PC](#)

## Tecnologías usadas

- Node Js para el backend y deploy de la aplicación.
- JISON para el análisis del archivo de entrada.
- Express js para el manejo de rutas en el servidor.
- Herramientas del análisis sintáctico ascendente.

Expresión regular

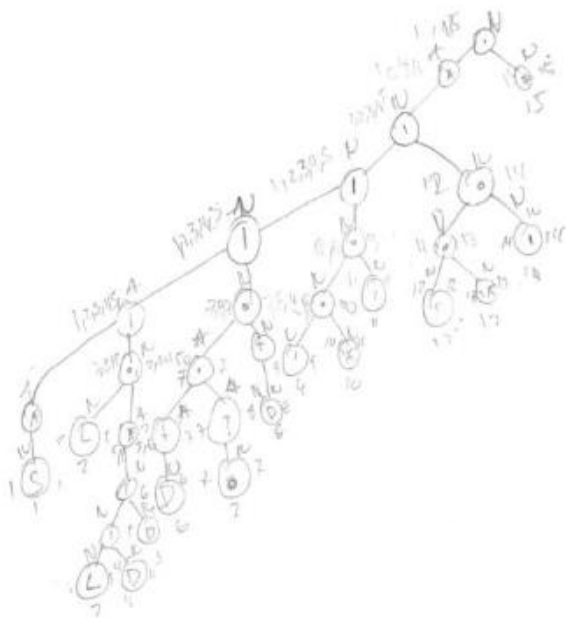
$(L(L|'|D)^*|D^+?D^+|[\"']\cdot*[\"']|[']00/00/0000['])^*$   
 $S^+)^*$

Donde

$L = \text{letras} \rightarrow a-z A-Z$

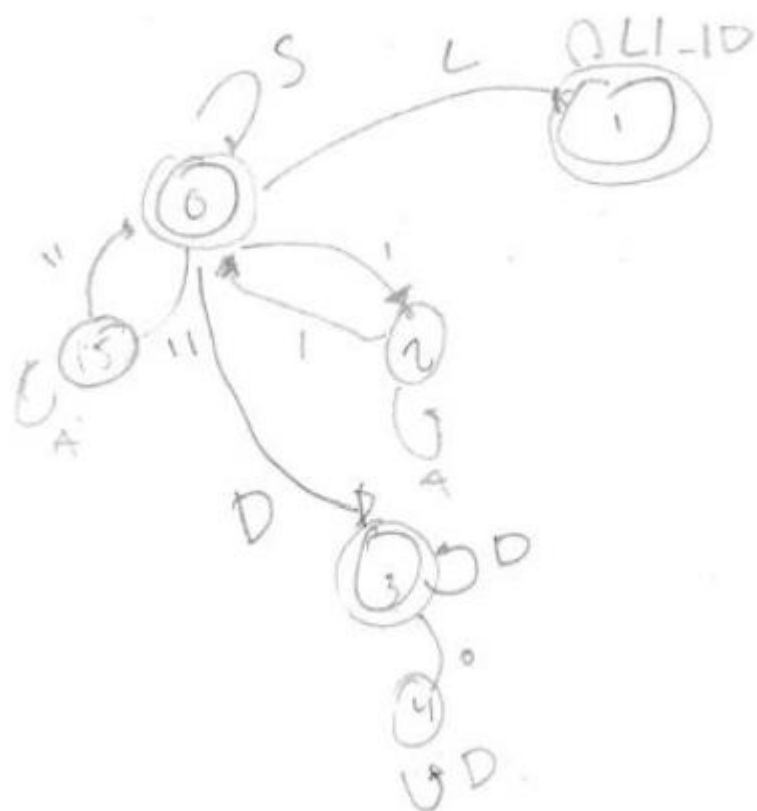
$D = \text{digitos} \rightarrow [0-9]$

$S = \text{Símbolos} \{ (, ), >, <, =, !, *, \cdot, , \}$



Follow

- 1 1, 2, 3, 4, 12
- 2 12, 3, 4, 12
- 3 1, 2, 3, 4, 12
- 4 1, 2, 3, 4, 12
- 5 8, 9, 12
- 6 8, 9, 12
- 7 8, 9, 12
- 8 8, 9
- 9 8, 9
- 10 10, 11
- 11 10, 11, 12, 13
- 12 13, 14, 15
- 13 14, 15
- 14 14, 15
- 15 —



## Gramática

### instr\_methods

-> instr\_methods instr\_meth

| instr\_meth

### instr\_meth

-> CLASS IDENTIFICADOR { instr\_methods }

| VOID IDENTIFICADOR ( params ) { instr\_methods }

| typo\_var IDENTIFICADOR ( params ) { instr\_methods }

| VOID MAIN ( ) { instr\_methods }

| IF ( asignacion ) { instr\_methods }

| ELSE IF ( asignacion ) { instr\_methods }

| ELSE { instr\_methods }

| WHILE ( asignacion ) { instr\_methods }

| DO { instr\_methods } WHILE ( asignacion ) ;

| CONSOLE PUNTO WRITE ( asignacion ) ;

| FOR ( var\_for ; asignacion ; asignacion\_icr ) { instr\_methods }

| typo\_var lista\_v IGUAL asignacion ;

| typo\_var lista\_v ;

| BREAK ;

| RETURN asignacion\_ret ;

| IDENTIFICADOR sms ;

| IDENTIFICADOR ( params2 ) ;

| IDENTIFICADOR IGUAL asignacion ;

| SWITCH ( asignacion ) { sw\_op }

| CONTINUE ;

| epsilon

asignacion\_ret

-> asignacion

| epsilon

asignacion\_icr

: IDENTIFICADOR sms

sms

: + +

| - -

lista\_v

-> lista\_v , IDENTIFICADOR

| IDENTIFICADOR

sw\_op

-> sw\_op casos

| casos

casos

- > CASE asignacion : instr\_methods
- | DEFAULT : instr\_methods

var\_for

- > typo\_var IDENTIFICADOR IGUAL asignacion
- | IDENTIFICADOR IGUAL asignacion

params2

- > params2 , asignacion
- | asignacion
- | epsilon

params

- > params , typo\_var IDENTIFICADOR
- | typo\_var IDENTIFICADOR
- | epsilon

typo\_var

- > INT
- | DOUBLE



- | STRING
- | CHAR
- | BOOLEAN

asignacion

- > asignacion symb asignacion
- | valx

valx

- > ENTERO
- | DECIMAL
- | IDENTIFICADOR
- | TRUE
- | FALSE
- | CADENA
- | ( asignacion )
- | unar\_op

unar\_op

- > MENOS valx
- | NOT valx

symb

- > AND

| OR

| NOT IGUAL

| MAS

| MENOS

| POR

| DIV

| MAYOR

| MENOR

| MAYOR IGUAL

| MENOR IGUAL

| IGUAL IGUAL