



Manual Técnico 2022

Estudiante

Nombre: Andy Ezequiel Sanic Tiul

Carnet: 202006699

Contenido

| | |
|---|----|
| INTRODUCCIÓN | 3 |
| Información destacada | 3 |
| Objetivos | 3 |
| REQUERIMIENTOS | 4 |
| INSTALACIÓN Y CONFIGURACIÓN | 5 |
| PROGRAMA..... | 6 |
| Analizador.py | 7 |
| Error.py | 8 |
| main.py | 8 |
| ANEXOS..... | 13 |
| Creación de árbol..... | 13 |
| Creación de árbol..... | 13 |
| Tabla de transiciones | 14 |
| Método del árbol | 15 |
| Calcular siguientes..... | 16 |
| Construir tabla de transiciones | 17 |
| Optimizar tabla | 17 |
| Autómata..... | 18 |
| Gramática tipo 2 o libre de contexto..... | 19 |

INTRODUCCIÓN

El siguiente trabajo describe los aspectos técnicos informáticos del sistema de información. El documento familiariza al personal técnico especializado encargado de las actividades de mantenimiento, revisión, solución de problemas, instalación y configuración del sistema.

Información destacada

Este manual es con el fin de orientar y referenciar a la información necesaria para orientar al personal, planteamiento del análisis programación e instalación del sistema.

Este manual es referido a personal con conocimientos en programación avanzada, sobre el uso de autómatas, método árbol y construcción de este.

Objetivos

Instruir el uso correcto del Sistema de Información, para el acceso oportuno y continuamente adecuado, descripción de los archivos relevantes del sistema, con los que se podrá orientas en la configuración y soporte de este

REQUERIMIENTOS

El sistema puede ser instalado en cualquier sistema operativo con los siguientes requerimientos:

- Visual Estudio Code v 1.65.1 Librerías:
- PrettyTable
- Tkinter
- Pandas
- Webbrowser
- 2 GB RAM

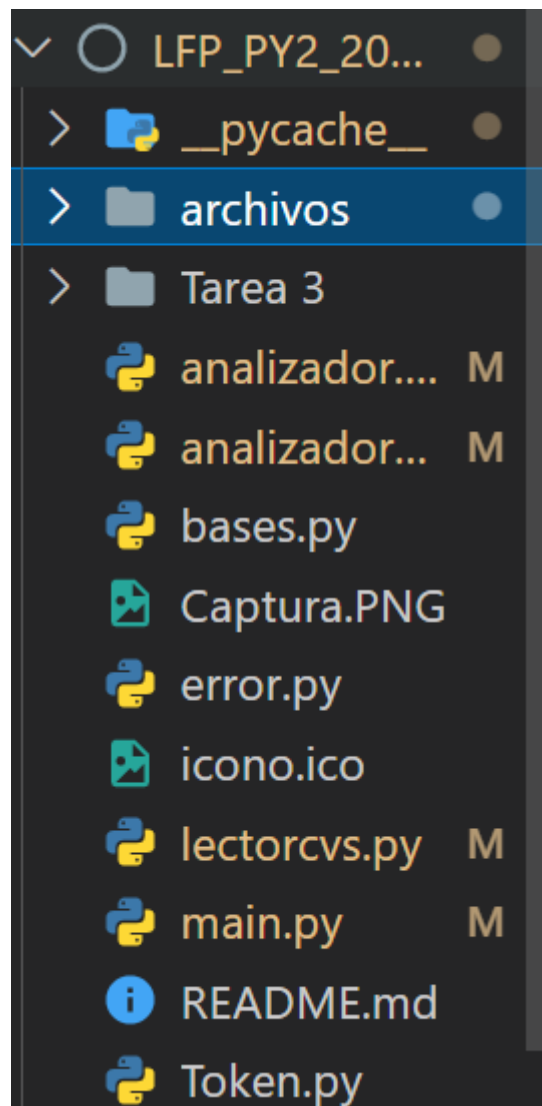
INSTALACIÓN Y CONFIGURACIÓN

Existen muchos métodos de instalación, sin embargo, a continuación, se describe de forma sencilla y segura la instalación, la que genera un link simbólico.

1. Descomprima el rar, para su uso directo.
2. Instale las librerías, especificadas, (PrettyTable, Tkinter, Webbrowser, Pandas).

PROGRAMA

Estructura raíz



A continuación, describimos los directorios y archivos más importantes:

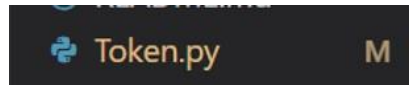
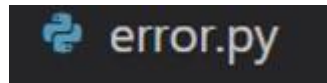
Analizador.py



Aquí se hace el autómata, y se desarrolla el árbol generando la estructura del contenido a programar.

```
from error import Error
from Token import constructor
from prettytable import PrettyTable
import os
from tkinter import messagebox
...
class AnalizadorLexico():
    def __init__(self):
        self.ListaTokens = []
        self.ListaTokens2= []
        self.ListaErrores = []
        self.linea = 1
        self.columna= 0
        self.lexema = "" # strings
        self.estado = 1 #
        self.i = 0 #? con este vamos recorriendo nuestras listas y guardando
    # while=?
    def limpieza2(self):...
    def agregar_token(self, caracter, tipo, linea, columna ):...
    def error_append(self, caracter, linea, columna, tipo):...
    #!=====
    #? Entrada de excepciones, simbolos, caracteres, opciones, etc
    def Estado0(self, caracter):...
    #!=====
    #? Estados para ingreso de cadenas, indicadores y reservadas, excepciones de cadenas
    #* Verificado completo
    def Estado1(self, caracter :str):...
    #!=====
    #? Estados de digitos enteros
    def Estado2(self, caracter :str):...
    #!=====
    #? Estados de digitos enteros
    def Analizar(self, caracter):...
        You, hace 2 semanas * cambios y entrada de analizador ...
    def imprimirTokens(self):...
    def imprimirTokens2(self):...
    def imprimirErrores(self):...
    def limpiarTokens(self):
        self.ListaTokens = []
    def limpiarerror(self):
        self.ListaErrores = []
    def guardar(self, name: str, cadena: str, abrir: bool = True ): #? es una libreria por
    def busqueda(self, codigo):...
    def Tabla_tokens(self, cadena, nombre_tab):...
```

Error.py



Se genera nuestras estructuras, constructor y clases correspondientes.

```

analizador.py M  error.py  X  main.py M
error.py > Error > __init__ > [?] tipo
You, ayer | 1 author (You)
class Error:
    def __init__(self, descripcion:str, linea, columna, tipo):
        self.descripcion = descripcion #? lexema
        self.linea = linea
        self.tipo = tipo
        self.columna = columna
    def imprimir_error(self):
        print(self.descripcion, self.linea, self.columna)

```

```

analizador.py M  Token.py M  X  main.py M
Token.py > constructor
1  class constructor:
2      '''Clase token'''
3      def __init__(self, descripcion : str, linea, columna : int, tipo) -> None:
4
5          self.descripcion = descripcion #?lexema
6          self.linea = linea
7          self.columna = columna
8          self.tipo = tipo
9      def imprimir_data(self):
10         print(self.descripcion, self.linea, self.columna, self.tipo)
11
12
13
14     def get_Descripción_Token (self):
15         return self.descripcion
16     def get_linea_Token(self):
17         return self.linea
18     def get_columna_Token(self):
19         return self.columna
20     def get_tipo_Token (self):
21         return self.tipo
22

```

main.py

Aquí importamos nuestras funciones, para ejecutar nuestra aplicación.


```

import tkinter as tk
from tkinter import *
from tkinter import messagebox
from tkinter import ttk
from tkinter.filedialog import askopenfile
from tkinter import filedialog, Tk
import sys
from tkinter.scrolledtext import ScrolledText
from analizador import AnalizadorLexico
import webbrowser

lexico = AnalizadorLexico ()

def ventana_creacion():
    #!ttk son los componentes
    #? creamo nuestro objeto
    ventana = tk.Tk()
    #? modificamos el tamaño de la ventana
    ventana.geometry('650x550')
    #? titulo de la ventana
    ventana.title("Interfaz Gráfica")
    #? configuramos nuestro icono de nuestra app
    ventana.iconbitmap("icono.ico")
    ventana.resizable(0,0) #? evitamos que cambie de tamaño
    ventana.config(background= "#6898FD")

sintactico = AnalizadorSintactico(1)
def ventana_creacion():
def manualusuario():
    ruta_datos = ruta.path.dirname(uso_.path.abspath(__file__)) + "\\archivos\\Manual Usuario"
    webbrowser.open_new(ruta_datos)
    print(ruta_datos)
def manualtecnico():
    ruta_datos = ruta.path.dirname(uso_.path.abspath(__file__)) + "\\archivos\\Manual Tecnico"
    webbrowser.open_new(ruta_datos)
    print(ruta_datos)
def lmp (): ...
#? entradas de respuesta
def preguntas(): ...

```

```

def lmp (): ...
#? entradas de respuesta
def preguntas(): ...

#? limpieza
def analizar(): ...

def tokens(): ...
def errores(): ...

def limpiarTok(): ...
def limpiarerr(): ...

def salir(): ...
def abrir(): ...
#!rutas dinamicas
def rutars(entrada): ...

```

```

# #? creacion de mensajes
if __name__ == '__main__':
    ventana_creacion()
    lexico.Tabla_tokens()
    print(len(lexico.ListaErrores))
    sintactico.imprimirErrores()

```

| Funcion | Descripción | Paquete | Clase |
|-------------------------|---------------------------------|---------|---------|
| def ventana_creacion(): | crea la ventana de nuestra app | tkinter | ninguna |
| def manualusuario() | ingresa los manuales | os | ninguna |
| def manualtecnico(): | Entrada de todos nuestros datos | os | ninguna |
| def preguntar(): | Entrada de todos nuestros datos | python | ninguna |
| def analizar (): | Entrada de todos nuestros datos | python | ninguna |
| def tokens (): | Entrada de todos nuestros datos | python | ninguna |
| def errores (): | Entrada de todos nuestros datos | python | ninguna |
| def limpiarTok(): | Entrada de todos nuestros datos | python | ninguna |
| def limpiarerr(): | Entrada de todos nuestros datos | python | ninguna |
| def salir() | Entrada de todos nuestros datos | python | ninguna |
| def abrir () | Entrada de todos nuestros datos | python | ninguna |
| def rutars () | Entrada de todos nuestros datos | python | ninguna |

| Funcion | Descripción | Paquete | Clase |
|---|---------------------------------|---------|------------------|
| def limpieza2(): | limpia tokens 2 | tkinter | AnalizadorLexico |
| def agregar_token() | agrega tokens | os | AnalizadorLexico |
| def error_apend(): | Entrada de todos nuestros datos | os | AnalizadorLexico |
| def Estado0(): | Entrada de nuestros tokens | python | AnalizadorLexico |
| def Estado1(), def Estado2(), def Estado3(), def Estado4(), def Estado5(), def Estado6(), def Estado7(), def Estado8(), def Estado9(), def Estado10() | analizador de nuestra entrada | python | AnalizadorLexico |
| def Analizar | analizador de nuestra entrada | | |
| def imprimirTokens | imprime tokens | python | AnalizadorLexico |
| def imprimirErrores | imprime errores | python | AnalizadorLexico |
| def limpiaTokens | limpia tokens 2 | python | AnalizadorLexico |
| def limpiarerror | limpia errores | python | AnalizadorLexico |
| def guardar, def busqueda, def Tabla_tokens | guarda y reliaza nuestras data | python | AnalizadorLexico |
| def Tabla_tokens | tokens | python | AnalizadorLexico |

| Funcion | Descripción | Paquete | Clase |
|--------------------|-------------------|---------|-------------|
| resultado_partido | resultado partido | tkinter | ImportarCSV |
| resultados_jornada | resultado jornada | os | ImportarCSV |
| resultados_goles | resultado goles | os | ImportarCSV |
| resultados_tabla | resultado tabla | python | ImportarCSV |
| resultado_equipo | resultado equipo | python | ImportarCSV |
| resultado_top | resultado top | | ImportarCSV |

| | | | |
|--|--------------------------------|--------|-------------|
| def imprimirTokens | resultado imprimir tokens | python | ImportarCSV |
| def imprimirErrores | resultado errores | python | ImportarCSV |
| def limpiaTokens | | python | ImportarCSV |
| def limpiarerror | limpia errores | python | ImportarCSV |
| def guardar, def busqueda, def Tabla_tokens | guarda y reliaza nuestras data | python | ImportarCSV |
| def Tabla_tokens | tokens | python | ImportarCSV |

| Funcion | Descripción | Paquete | Clase |
|----------------|---------------------|---------|----------------------|
| def limpieza | limpia errores | tkinter | AnalizadorSintactico |
| agregarErro | agrega errores | python | AnalizadorSintactico |
| sacarToken | saca tokens | python | AnalizadorSintactico |
| analizar | analiza | python | AnalizadorSintactico |
| observarToken | observa tokens | python | AnalizadorSintactico |
| S | estado de inicio | python | AnalizadorSintactico |
| INICIO | inicio | python | AnalizadorSintactico |
| RESULT | result | python | AnalizadorSintactico |
| INS | bandera -f | python | AnalizadorSintactico |
| GOL | gol | python | AnalizadorSintactico |
| JORNA | jornada | python | AnalizadorSintactico |
| TABL | tabla | python | AnalizadorSintactico |
| PARTIDO | partido | python | AnalizadorSintactico |
| INS1 | bandera -ji | python | AnalizadorSintactico |
| INS2 | bandera -jf | python | AnalizadorSintactico |
| TOP | top | python | AnalizadorSintactico |
| CONDICION2 | SUPERIOR O INFERIOR | python | AnalizadorSintactico |
| ADIOS | adios | python | AnalizadorSintactico |
| imprimirErrore | imprime errores | python | AnalizadorSintactico |

ANEXOS

Creación de árbol



Creación de árbol

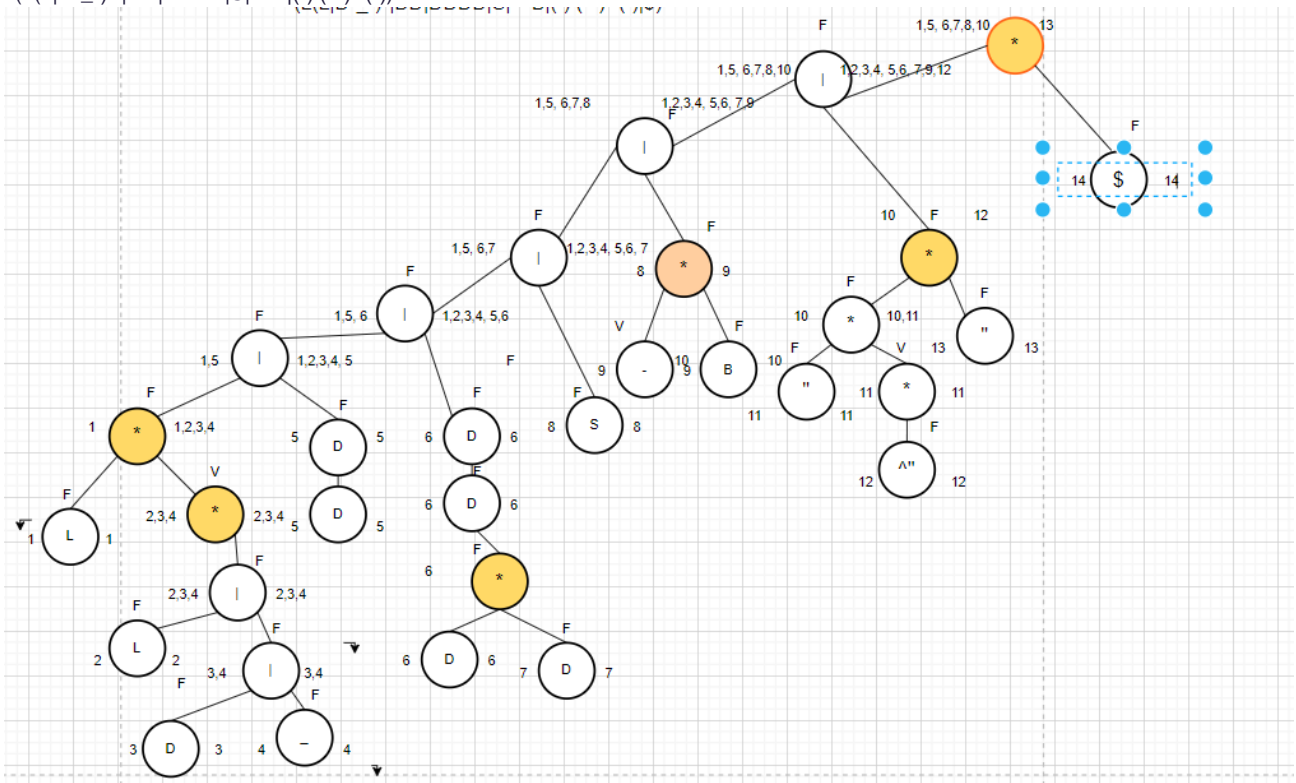
| |
|----------------------------|
| $L = \{A-Z, \tilde{N}\}$ |
| $L_m = \{a-z, \tilde{n}\}$ |
| $D = \{0-9\}$ |
| $S = \{<>- \}$ |
| $B = \{-f, -ji, -jf, -n\}$ |

Tabla de transiciones

| Nombre | Patron | ER | Ejemplos |
|-------------------|---|--------------|---|
| Cadena | Expresiones inician con una letra, seguidas de letras, puede seguir de un dígito, o un espacio. | (") (^)* (") | *año *pais_5 *argentina5 *Valencia |
| numero | Expresión compuesta de 1 número seguida de 1 número | DD | *22 *55 *14 *04 |
| dígito | Expresión inicia con un número seguida de 3 dígitos | (DDDD) | *2022 *2001 *1948 *5201 |
| Palabra reservada | Inicia con una letra, seguida de más letras | P | *equipo * condicion |
| Identificador | Inicia con una letra seguida de más letras, definidas por un espacio | L(L D" _")* | *reporte * año1 *vision_3 |
| Simbolos | Expresión compuesta por un símbolo | S | *< *> *_ |
| Bandera | Expresión compuesta un símbolo, seguido de una bandera | "-"B | *-f *-ji *-jf *-n |

Método del árbol

$(L(L|D|_*)^*|DD|DDDD|S|"-B|(") (^{^})^* (^{^}))$



Calcular siguientes

| Valor | Hoja | Siguientes |
|-------|------|------------|
| L | 1 | 2,3,4,13 |
| L | 2 | 2,3,4,13 |
| D | 3 | 2,3,4,13 |
| " " | 4 | 2,3,4,13 |
| D | 5 | 5,13 |
| D | 6 | 6,13 |
| S | 7 | 7,13 |
| - | 8 | 8,9 |
| B | 9 | 9,13 |
| " | 10 | 11,12 |
| ^" | 11 | 11,12 |
| " | 12 | 13 |
| \$ | 13 | _____ |

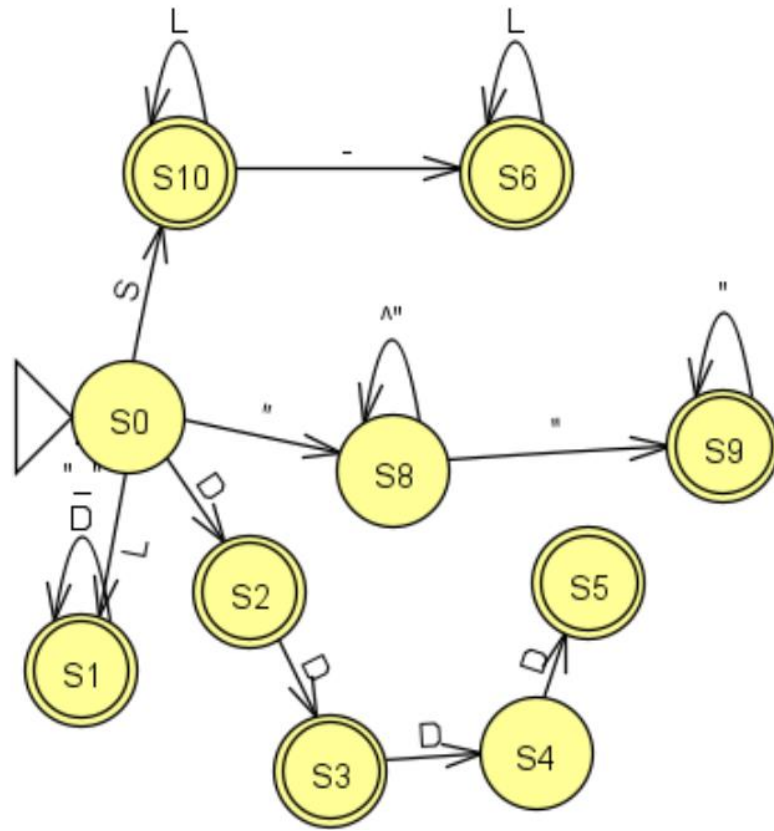
Construir tabla de transiciones

| | ESTADO | VALORES | SIGUIENTES |
|--------------|--------|-----------------------------|---|
| INICIO | S0 | L,D,D,S,-," 1,5,6,7,8,10 | L: {2,3,4,13} = S1 D: {5,13}= S2 D: {6,13} = S3 S:{7,13}=S4 -: {8,9} = S5 ": {11, 12} = S6 |
| ACEPTACIÓN | S1 | L,L,D,"_" \$ 2,3,4,13 | L: {2,3,4,13}=S1 D: {2,3,4,13}=S1 "_" :{2,3,4,13}=S1 |
| ACEPTACIÓN | S2 | D \$ 5, 13 | D:{5,13} = S2 |
| ACEPTACIÓN | S3 | D, \$ 6, 13 | D:{6,13}:S3 |
| FINALIZACIÓN | S4 | \$ 13 | _____ |
| ACEPTACIÓN | S5 | S, \$ 7, 13 | S: {7,13}=S4 |
| ACEPTACIÓN | S6 | "-",B, \$ 8, 9, 13 | "-": {8,9}=S5 B:{9,13}=S5 |
| ACEPTACIÓN | S7 | ^", * 10, 11 | ^": {11,12}=S6 ": {11,12}=S6 |

Optimizar tabla

| ESTADOS | Σ | | | | | | | | | | | |
|---------|----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | L | D | D2 | D3 | D4 | "_" | "-" | B | " | ^" | \$ | S |
| S0 | S1 | S2 | S3 | _____ | _____ | _____ | S5 | _____ | S6 | _____ | S4 | S5 |
| S1 | S1 | S1 | _____ | _____ | _____ | S1 | _____ | _____ | _____ | _____ | _____ | _____ |
| S2 | _____ | S2 | _____ | S2 | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ |
| S3 | _____ | S3 | S3 | S3 | S3 | _____ | _____ | _____ | _____ | _____ | _____ | _____ |
| S4 | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ |
| S5 | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | S5 |
| S6 | _____ | _____ | _____ | _____ | _____ | S6 | _____ | S6 | _____ | _____ | _____ | _____ |
| S7 | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | S7 | S7 | _____ | _____ |

Autómata



Gramática tipo 2 o libre de contexto

Terminales = {TK_result, TK_vs, TK_jorn, TK_temp, TK_gol, TK_tabla, TK_part, Tk_top, tk_cadena, tk_s, tk_num, tk_smay, tk_smen, tk_b, tk_año, TK_local, TK_visitante, TK_total, tk_id, Tk_adios, tkb, tkb1, tkb2, tkb3, TK_inf, TK_sup}

No terminales {<INICIO>, <E>, <RESULT>, <JORNA>, <GOL>, <TEMPORADA>, <TABLA>, <PARTIDO>, <TOP>, <ADIOS>, <CONDICION>, <INS>, <INS1>, <INS2>, <INS3>, <CONDICION2> }

Producciones:

<INICIO>::= <RESULT> | <JORNA> | <GOL> | <TEMPORADA> | <PARTIDOS> | <TOP> | <ADIOS>

RESULTADO equipo VS equipo TEMPORADA <YYYY-YYYY>
<RESULT>::= TK_result tk_cadena tk_VS tk_cadena TK_temp tk_smen tk_año tk_s tk_año tk_smay

<JORNA>::= TK_jorn tk_num TK_temp tk_smen tk_año tk_s tk_año tk_smay
<INS>
<INS>::= tkb tk_id | ϵ

<GOL>::= TK_gol <CONDICION> tk_cadena TK_temp tk_smen tk_año tk_s tk_año tk_smay

<CONDICION>::=tk_local | tk_visitante | tk_total

<TABLA>::= TK_tabla TK_temp tk_smen tk_año tk_s tk_año tk_smay INS

<PARTIDOS>::= TK_part tk_cadena TK_temp tk_smen tk_año tk_s tk_año tk_smay INS INS1 INS2
<INS1>::= tkb1 tk_cadena | ϵ
<INS2>::= tkb2 tk_cadena | ϵ

<TOP> TK_top <CONDICION2> TK_temp tk_smen tk_año tk_s tk_año tk_smay INS3
<CONDICION2>::= TK_sup | TK_inf
<INS3>::= tkb3 tk_cadena | ϵ

<ADIOS>::= Tk_adios