Universidad San Carlos de Guatemala

Facultad de ingeniería

Organización de lenguajes y compiladores 1

Proyecto 2

Juan Pablo Rojas Chinchilla

201900289

Primer semestre 2021

# Manual Técnico:

|  |  |  |
| --- | --- | --- |
| Objeto | Instrucción para realizar | Descripción |
| Asignacion | Asignacion de una variable ya declarada anteriormente en ese entorno. | Esta es una operación que recibe el nombre de la variable que se quiere asignar y el valor que se le asignará. |
| Declaración | Es la declaración de una variable en cierto entorno, esto significa crear un espacio en memoria para utilizar esta variable. | Esta función recibe el nombre que tendrá la variable, además de poder recibir un valor si así lo desea el usuario, este es un atributo opcional, si ingresara el nombre de una variable que ya esta declarada este se ignorará y no podrá agregarse ni asignarse esta variable. |
| Do While | Este es un ciclo el cual repite una serie de instrucciones dadas una vez o más, dependiendo de la condición que se envíe. | Esta operación es un ciclo el cual recibe una condición, si esta es verdad se repetirá este ciclo hasta que esta no sea verdad, en caso de ser mentira este realizará las instrucciones dadas una única vez. |
| Else | Este es la contraparte del if, es la acción que realizara algo si no llegara a realizarse ningún if o else if. | Esta operación es simple, ya que únicamente compara las condiciones de sus if o else if anteriores y dependiendo si alguna de ellas fue verdad o todas fueron falsas realizara sus instrucciones. |
| Expression | Operaciones lógicas, relacionales y aritméticas. | Esta es la generalización de las operaciones que se pueden realizar en este lenguaje, estas reciben uno o dos parámetros dependiendo que tipo de operación sea, este objeto se ejecuta y devuelve un símbolo, el cual será un número, cadena, carácter o el valor de una variable dependiendo como se utilice, este podría considerarse el corazón y cerebro del lenguaje |
| Function\_call | Es la llamada de una función antes declarada. | Esta operación no es más que la llamada a una función o método antes declarado, esto nos sirve para ejecutar una serie de instrucciones que encapsulamos en una función |
| Function | Esta es la declaración de una función o método. | Este es la otra parte de la llamada de funciones, ya que sin esta no podría llamarse a las funciones, esta instrucción lo único que hace es encapsular una serie de instrucciones para luego poder utilizarla únicamente con un llamado, esta puede o no tener parámetros. |
| If | Esta es una instrucción de control. | Esta es la instrucción de control mas básica, esta recibe una condición y una serie de instrucciones, esta ejecutara la serie de instrucciones siempre y cuando la condición sea verdadera, de no serlo no ejecutara estas instrucciones y dependiendo de otras instrucciones hará lo que se le pida. |
| Instrucción | Esta es la generalización de todas las clases antes mencionadas y también las que faltan por mencionar. | Esta es la clase padre de la cual salen todas las demás clases, esta es utilizada para la ejecución de instrucciones, este nos ayuda a llevar un orden y poder generalizar la serie de instrucciones inicial que recibimos, para luego así solo ejecutar el mismo método pero de diferentes clases. |
| List\_if | Esta instrucción no es una como tal, pero es utilizada tanto para if, como para if-else, como para else if. | Esta instrucción es utliizada para almacenar las condiciones de una serie de else if y de un if else, se utiliza también para los if solos únicamente para tener un estándar el cual poder seguir. |
| Parameters\_ins | Esta instrucción nos sirve únicamente para poder mandar y recibir parámetros de una manera ordenada. | Este objeto únicamente nos sirve para almacenar los parámetros y ejecutarlos al momento de asignarlos, es utilizado únicamente en funciones, también algunas sentencias de control lo utilizan ya que pueden ser llamadas en métodos y cambian el valor de algunos parámetros. |
| Print | Esta es una función nativa la cual nos sirve para imprimir cosas en consola. | Esta función nos sirve únicamente para tener una interacción con el programador, esta nos mostrara en un apartado llamado consola lo que el programador nos solicite. |
| Return | Esta es una sentencia de retorno. | Lo que se refiere a una sentencia de retorno es que esta sentencia nos devolverá un valor el cual nosotros le pidamos, este puede ser tanto una variable como un valor constante. |
| While | Este es el ciclo más básico y del cual derivan los demás. | Este ciclo funciona recibiendo una condición y una serie de instrucciones como el do-while, pero la diferencia es que este no ejecutara ninguna vez su serie de instrucciones si la condición enviada resulta ser falsa, esta serie de instrucciones se ejecutara hasta que la condición sea falsa. |
| Switch | Esta responde a una sentencia de control muy parecida al if. | La sentencia switch responde a una sentencia de control en la cual se comparan los valores de cada case con el valor ingresado en el switch, si estos son iguales procederá a hacer las instrucciones enlistadas en el case respectivo |
| For | Este ciclo es una manera “fácil” de usar sentencias cíclicas. | Este ciclo a diferencia de los demás no recibe únicamente una condición e instrucciones, este también recibe una declaración y una actualización, esta nos sirve para llevar un control de las veces que se repetirá este ciclo. |

Gramática:

La gramática utilizada en este proyecto debe ser libre de ambigüedades ya que el analizador utilizado es de tipo ascendente.

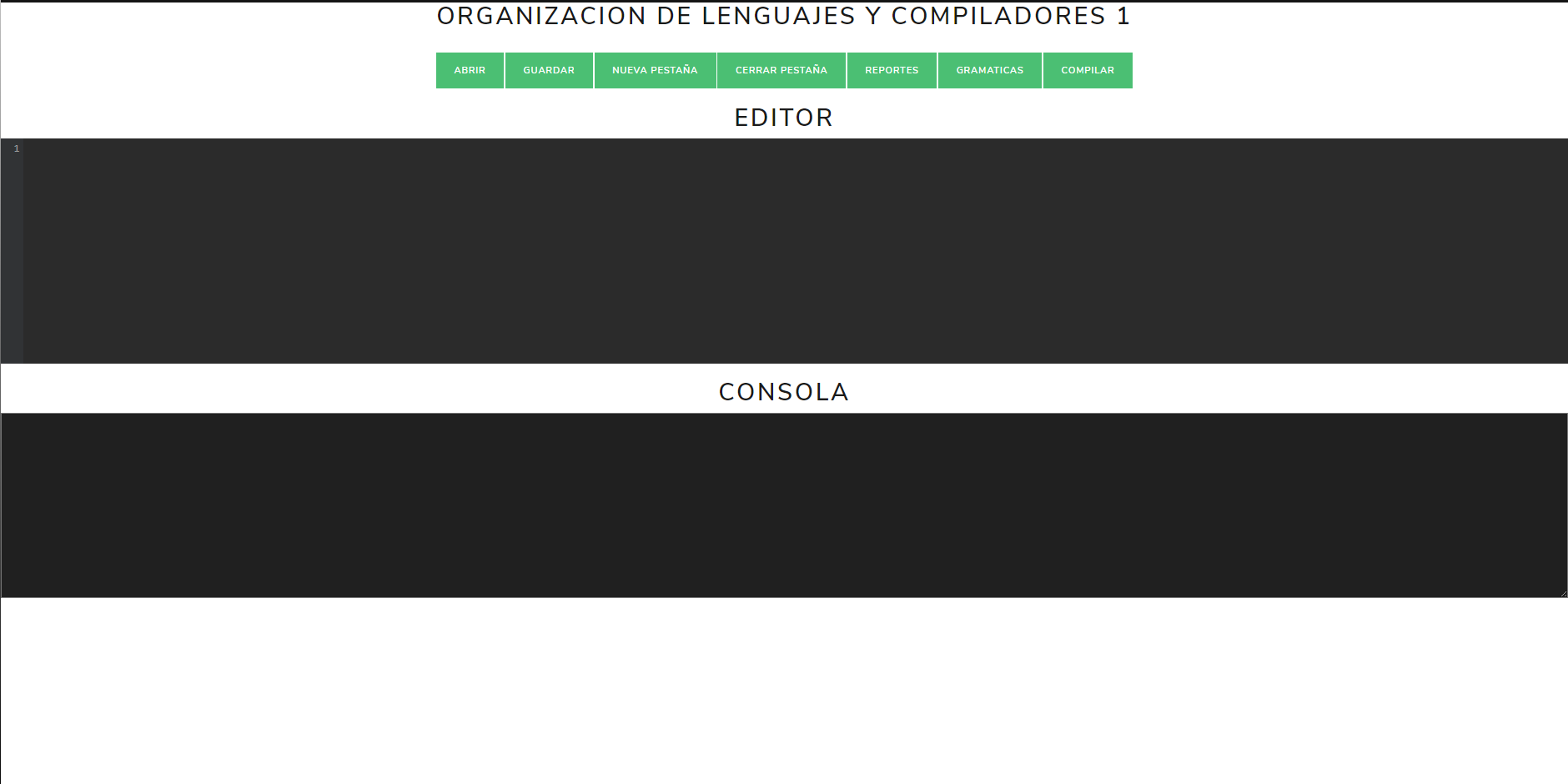
La gramática utilizada fue la siguiente

|  |  |
| --- | --- |
| Palabra reservada | Simbolo de palabra |
| COMA | , |
| P\_COMA | ; |
| DOSPUNTOS | : |
| PARENTESIS\_A | ( |
| PARENTESIS\_C | ) |
| CORCHETE\_A | [ |
| CORCHETE\_C | ] |
| LLAVE\_A | { |
| LLAVE\_C | } |
| IGUALIGUAL | == |
| DIFERENTE | != |
| MENORIGUAL | <= |
| MENOR | < |
| MAYORIGUAL | >= |
| MAYOR | > |
| MAS | + |
| MENOS | - |
| MULTI | \* |
| DIVISION | / |
| POTENCIA | ^ |
| MODULO | % |
| IGUAL | = |
| INT | Int |
| DOUBLE | Double |
| BOOL | Boolean |
| CHAR | Char |
| STRING | String |
| VOID | Void |
| NEW | New |
| LIST | List |
| ADD | Add |
| PRINT | Print |
| TOLOWER | toLower |
| TOUPER | toUpper |
| LENGTH | Length |
| TRUNCATE | Truncate |
| ROUND | Round |
| TYPEOF | typeOf |
| TOSTRING | toString |
| TOCHARARRAY | toCharArray |
| EXEC | Exec |
| IF | If |
| ELSE | Else |
| SWITCH | Switch |
| CASE | Case |
| DEFAULT | Default |
| BREAK | Break |
| CONTINUE | Continue |
| RETURN | Return |
| DO | Do |
| WHILE | While |
| FOR | For |
| OR | || |
| AND | && |
| NOT | ! |
| DECIMAL | [0-9]+("."[0-9]+)\b |
| ENTERO | [0-9]+\b |
| CARACTER | "'"([^']\*)"'" |
| CADENA | ("\"")([^\"\\]|\\.)\*("\"") |
| IDENTIFICADOR | [a-zA-z\_]([0-9a-zA-Z\_])\* |
| TRUE | True |
| FALSE | False |

|  |  |
| --- | --- |
| Instrucción | Producciones |
| Init | Instrucciones EOF |
| Instrucciones | Instrucciones instrucción  Instrucción |
| Sentencias | LLAVE\_A instrucciones LLAVE\_C |
| Instrucción | declaracion\_variables  asignación\_variables  actualización  print  else\_if  while  do\_while  function  function\_call  EXEC instrucción  RETURN expresión  Error |
| Expression | MENOS expression  PARENTESIS\_A expression PARENTESIS\_C  Expression MAS expression  Expression MENOS expression  Expression MULTI expression  Expression DIVISION expression  Expression POTENCIA expression  Expression MODULO expression  Expression AND expression  Expression OR expression  NOT expression  Expression MAYOR expression  Expression MENOR expression  Expression MAYORIGUAL expression  Expression MENORIGUAL expression  Expression IGUALIGUAL expression  Expression DIFERENTE expression  Expression MAS MAS  Expression MENOS MENOS  ENTERO  DECIMAL  CADENA  CARACTER  FALSE  TRUE  IDENTIFICADOR  Function\_call |
| Declaracion\_variables | Tipo IDENTIFICADOR P\_COMA  Tipo IDENTIFICADOR IGUAL expression P\_COMA |
| Asignacion\_variables | IDENTIFICADOR IGAULA expression P\_COMA |
| While | WHILE PARENTESIS\_A expression PARENTESIS\_C LLAVE\_A instrucciones LLAVE\_C |
| Do\_while | DO LLAVE\_A instruciones LLAVE\_C WHILE PARENTESIS\_A expression PARENTESIS\_C P\_COMA |
| If | IF PARENTESIS\_A expression PARENTESIS\_C LLAVE\_A instrucciones LLAVE\_C  LLAVE\_A instrucciones LLAVE\_C |
| Else\_if | else\_if ELSE if  if |
| Function | tipo IDENTIFICADOR PARENTESIS\_A parametros PARENTESIS\_C LLAVE\_A instrucciones LLAVE\_C  tipo IDENTIFICADOR PARENTESIS\_A PARENTESIS\_C LLAVE\_A instrucciones LLAVE\_C |
| Function\_call | IDENTIFICADOR PARENTESIS\_A PARENTESIS\_C  IDENTIFICADOR PARENTESIS\_A params\_value PARENTESIS\_C |
| Parametros | parametros COMA tipo IDENTIFICADOR  tipo IDENTIFICADOR |
| Params\_value | params\_value COMA expression  expression |
| Tipo | INT  DOUBLE  BOOL  CHAR  STRING  VOID |
| Print\_ | PRINT PARENTESIS\_A expression PARENTESIS\_C |
| Switch | SWITCH PARENTESIS\_A expression PARENTESIS\_C LLAVE\_A cases LLAVE\_C |
| Cases | Cases case  Case |
| Case | Case expression DOSPUNTOS instrucciones BREAK |
| For | FOR PARENTESIS\_A declaracion P\_COMA expression P\_COMA expression PARENTESIS\_C LLAVE\_A instrucciones LLAVE\_C |

# Manual de usuario:

Pantalla principal:



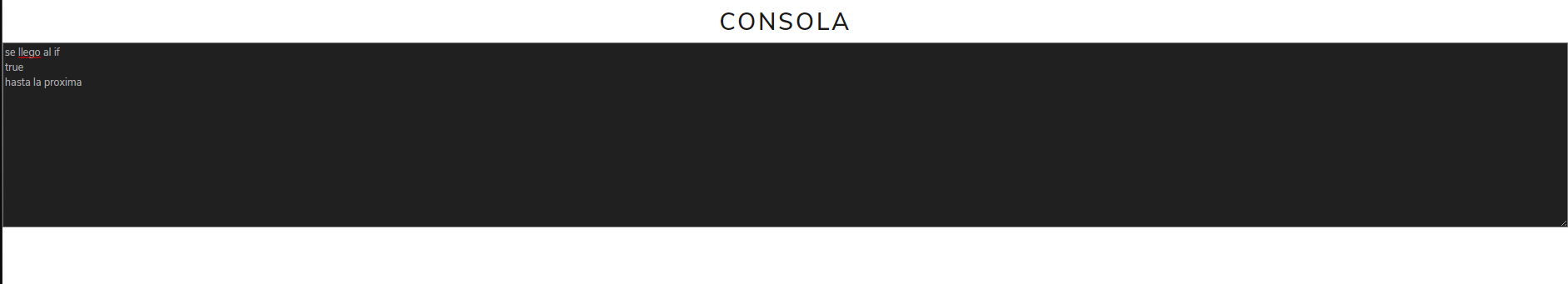
En esta pantalla podemos acceder a todas las funcionalidades de nuestra aplicación con un simple click.

Editor de código:

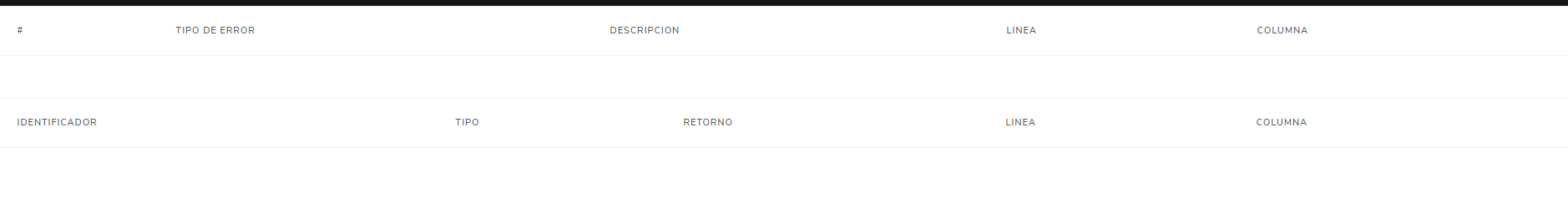


En nuestro editor de código es el lugar donde podemos desarrollar nuestro código para que después podamos mostrarlo en consola.

Consola

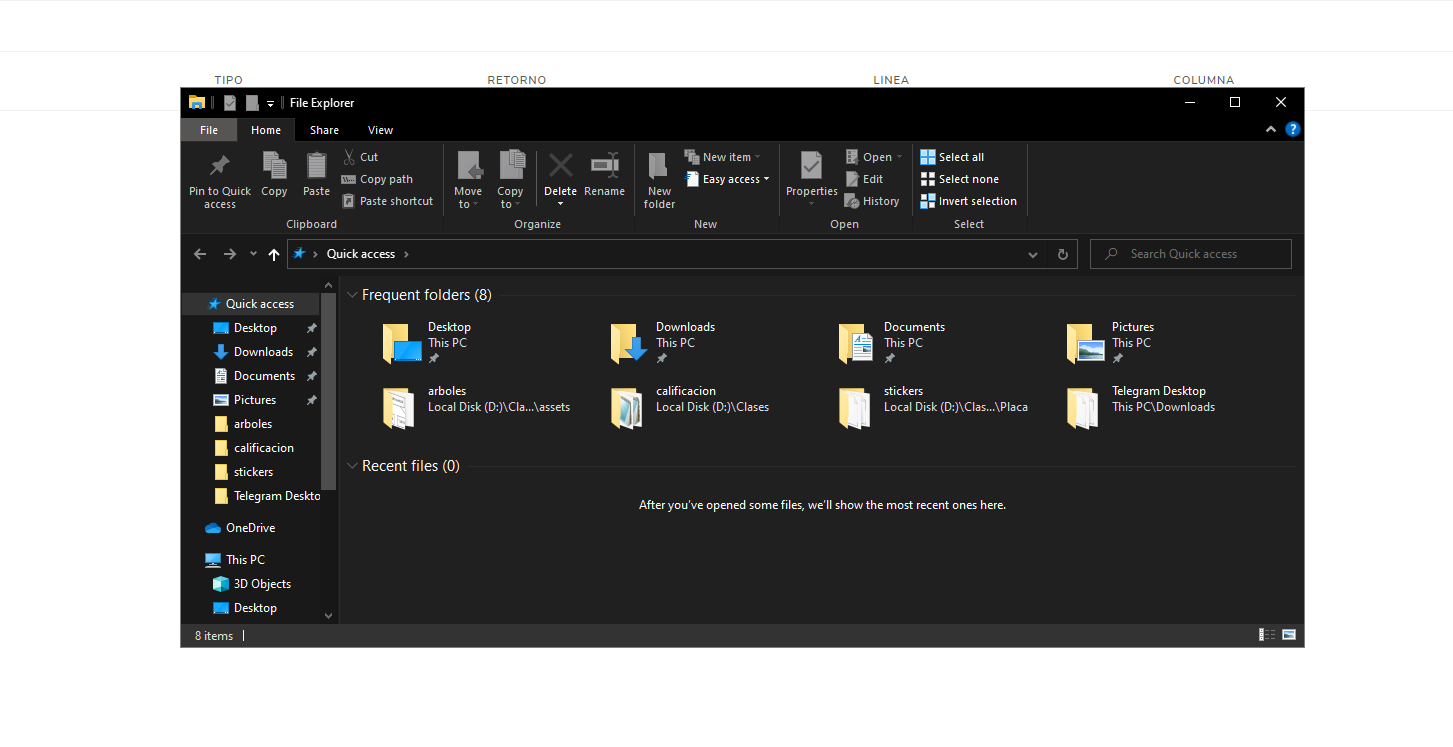
En nuestra consola se mostraran todo tipo de errores y las funciones print que el desarrollador haya definido.

Reportes



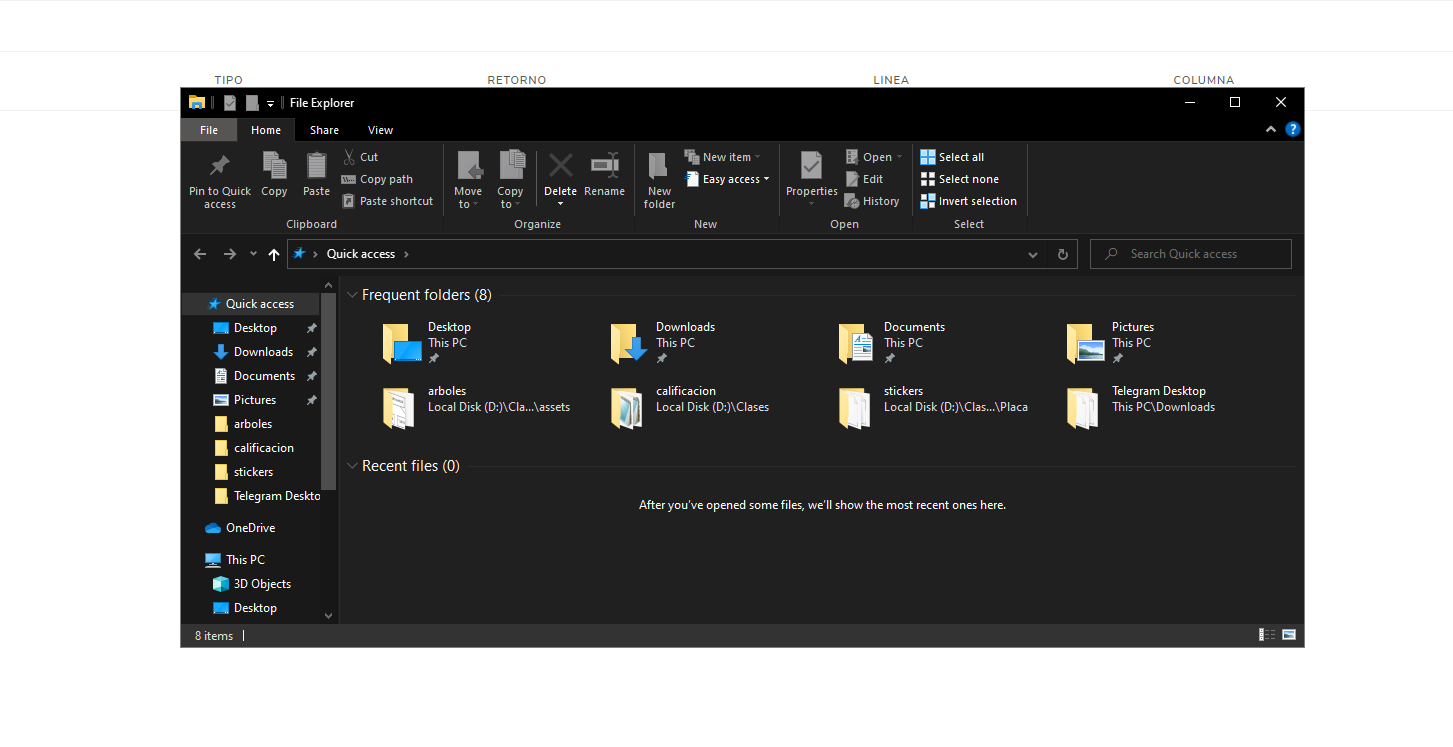
En el apartado de reportes contaremos con una vista para poder ver nuestros símbolos, errores y nuestro ast en caso de que existan.

Abrir archivos



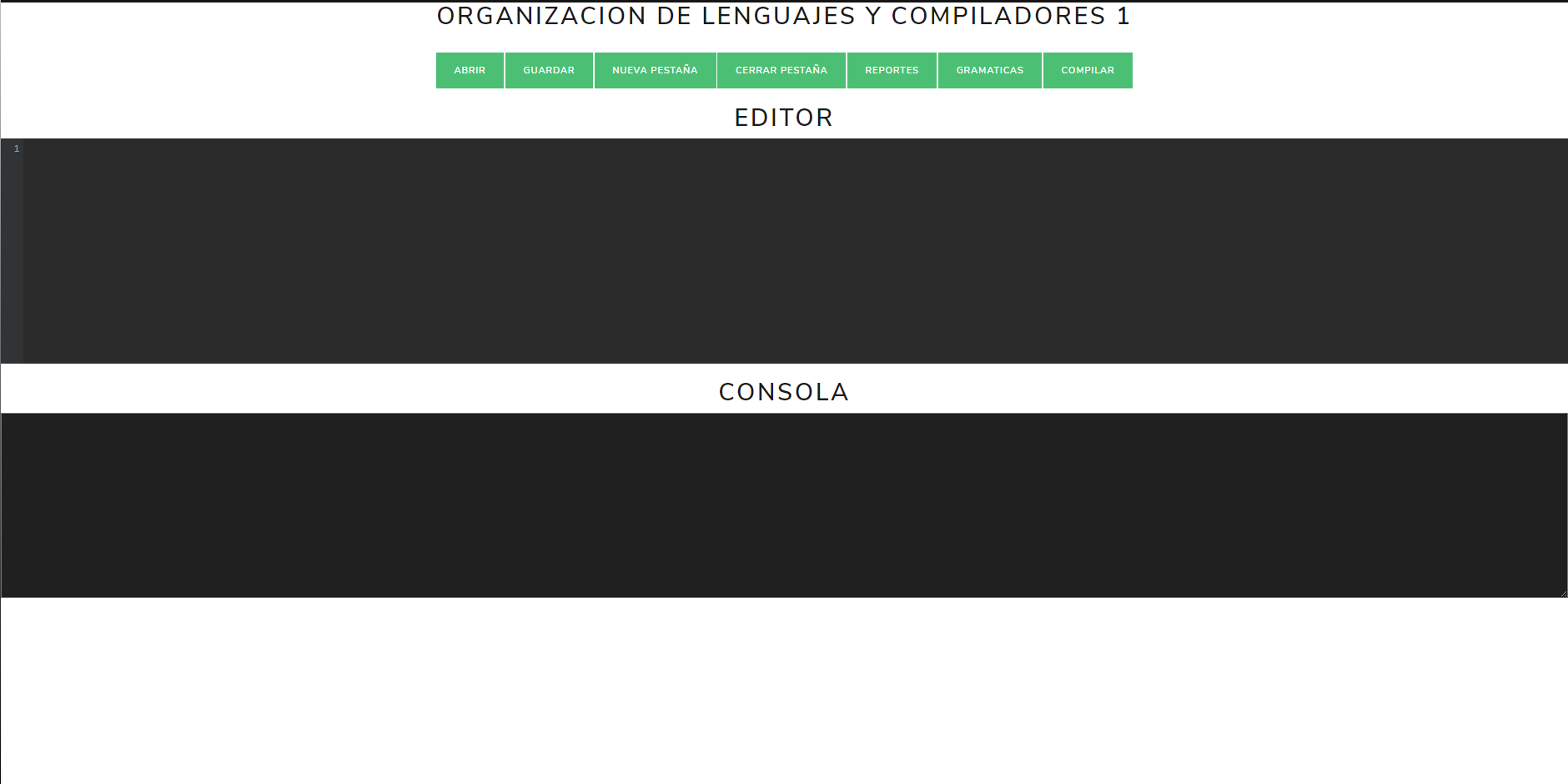
Esta es una funcionalidad a la cual podemos acceder desde nuestra pantalla principal, esta nos permitirá abrir un archivo y mostrarlo en nuestro editor de código.

Guardar archivos



Esta funcinalidad también se encuentra en la pantalla principal, y nos permitirá guardar nuestro archivo si es que hicimos un cambio en este.

Nueva pestaña



Esta funcionalidad también se encuentra en nuestra pantalla principal y esta nos permitirá abrir otra pestaña en la cual podremos seguir desarrollando nuestro código.

Gramáticas

En este apartado se mostrara la gramática que se mostró arriba en el apartado de gramática de este mismo documento.

Compilar

Este es el ultimo comando pero el mas importante ya que este nos podrá compilar nuestro código y así ver el resultado de nuestras creaciones.