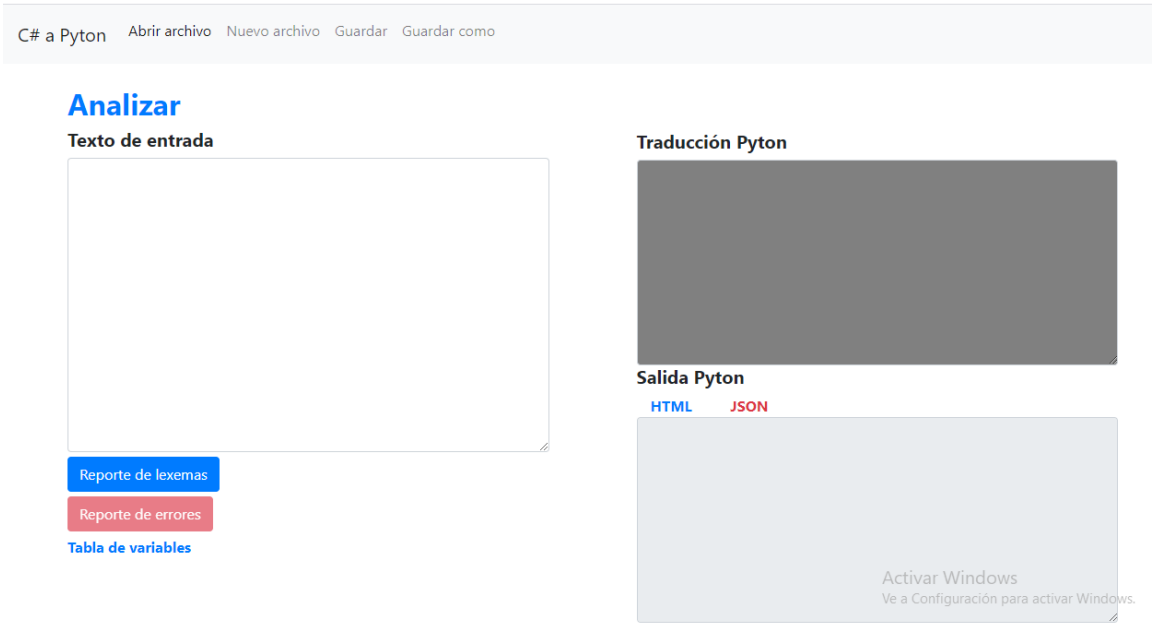
# MANUAL DE USUARIO

C# a Python es una aplicación que permite al usuario traducir de forma efectiva una entrada en formato C# y traducirlo a su semejante en Python. Para ello se dispone de la pantalla de inicio, donde de desarrolla todo el ciclo de ejecución.

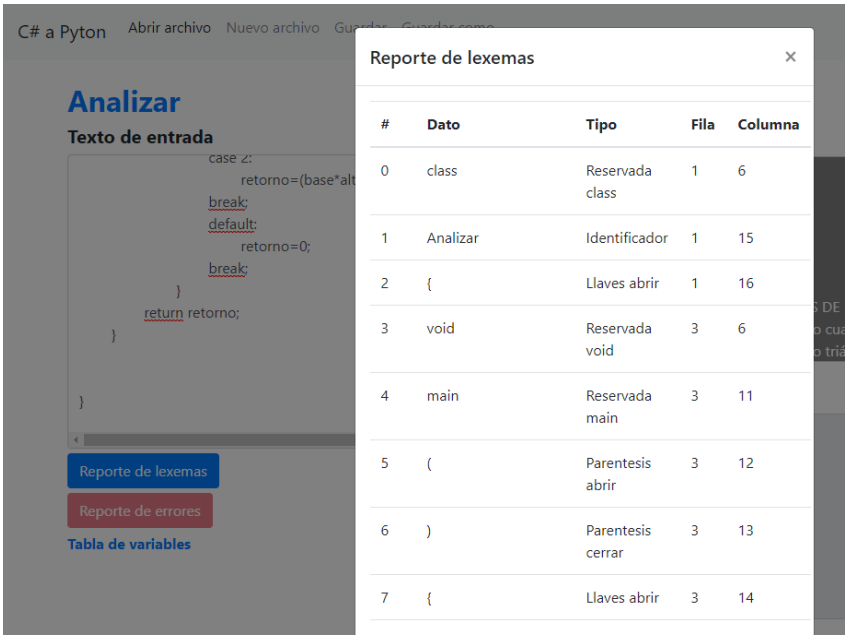



**Pantalla de inicio**

## Ciclo de ejecución

Se puede dar inicio ya sea abriendo un archivo con extención ***.cs*** o ingresando de forma manual la entrada en el texarea ubicado en la sección de **Texto de entrada**.

Para analizar la entrada, es necesario hacer click sobre **Analizar** que, al finalizar el proceso, inmediatamente desplegará diferentes alertas de acuerdo a la situación de la entrada analizada.




**Ejemplo de despliegue reportes en ventana auxiliar**

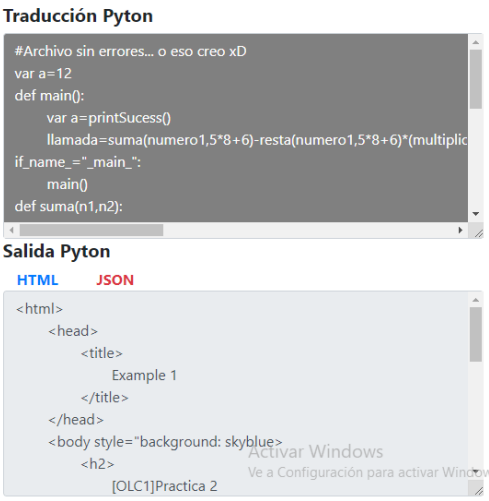
**Reporte de lexemas:** al presionar el botón de *Reporte de lexemas* se desplegará una ventana auxiliar cuyo contenido es una tabla con la descripción de todos los lexemas identificados en el análisis léxico.

**Reporte de errores**: al presionar el botón de *Reporte de errores* se desplegará de igual forma una ventana auxiliar con una tabla uniendo tanto los errores léxicos y sintácticos en caso de que estos se encuentren en la entrada.

## Traducción y salidas

**Traducción Python:** en esta sección es donde se muestra la traducción de las instrucciones especificadas en la entrada C#.

**Salida Python:** sección que muestra en el formato especificado, las instrucciones HTML que se encuentren dentro de las sentencias Console.Write delimitadas por comillas simples.




**Ejemplo de salidas tras el análisis de entrada**

## Barra de navegación

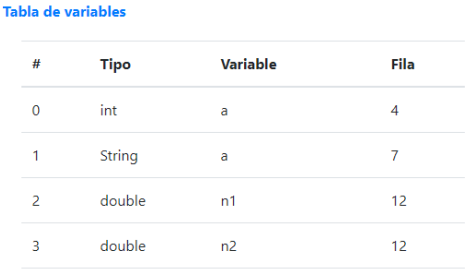
**Abrir archivo:** permite explorar dentro de los archivos y seleccionar una entrada a analizar.

**Guardar:** guarda el estado actual del archivo que se está trabajando. En caso de que la entrada no sea producto de *Abrir archivo* solicitara un nombre bajo el cual guardarlo.

**Guardar como:** despliega un mensaje de texto donde se solicita el nombre del archivo que se desea guardar.

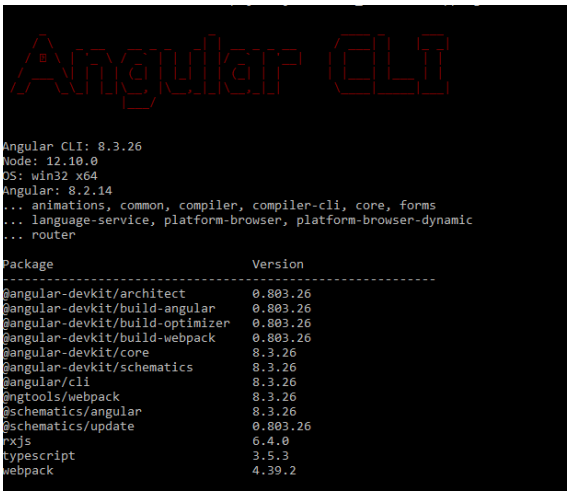
## Tabla de Variables

Muestra un listado de todas las variables declaradas con su respectivo tipado dentro del archivo de entrada y su ubicación.



# MANUAL TÉCNICO

La presente aplicación fue montada en angular bajo la versión especificada en la imagen de la derecha, utilizando Typescript, lenguaje del cual se auxilia principalmente angular. Así mismo, fue necesaria la instalación de jQuery para bootstrap para ciertas animaciones y despliegue de información, bootstrap para la estética general y sweetAlert(incluido sweetAlert2) para las alertas, por lo que de disponer de lo especificado con anterioridad la aplicación debe de correr sin problema alguno.




**LÓGICA**

Se hizo uso de un analizador léxico para la distinción de todos los lexemas definidos del lenguaje origen, C#, por lo que la implementación de un AFD fue necesario. En lo que compete al análisis del orden de los lexemas, el análisis sintáctico se auxilia de una gramática de tipo LL1 que permite recorrer y consumir los elementos obtenidos del analizador léxico para deducir si dichos respetan la grámatica antes mencionada. Cabe decir también que se dispone de un método de recuperación de tipo *Modo Pánico*, cuya lógica es consumir tokens hasta encontrar uno denominado *clave* escogido a conveniencia para que se pueda seguir el análisis a partir de él.

## Anéxo

### Gramática libre de contexto

**<INICIO>**::=class <Identificador> { <CONTENIDO> <CONTENIDOP>}

**<CONTENIDO>**::= <DECISION><CONTENIDO>

|<METODOS_VOID><CONTENIDO>

|<COMENTARIO BLOQUE><CONTENIDO>

| <COMENTARIO SIMPLE><CONTENIDO>

| ε

**<METODOS_VOID>**::=void <VOID>{<VOIDP>}

**<VOID>**::=<CUERPOMETODO>

| main()

**<VOIDP>**::=<INSTRUCCIONES> <RETURN>

**<DECISION_FUERA>**::=<tipo><identificador><DA>

**<DA>**::= (<PARAMETROS>{<INSTRUCCIONES> return <LOGICO>; <TIPADOP>}

|<LISTA><DECLARARP>;

**<TIPADO>**::=<INSTRUCCIONES><RETURNP>

**<CUERPOMETODO>**::=<identificador>(<PARAMETROS>

**<RETURN>**::=return;<VOIDP>

| ε

**<RETURNP>**::=return <LOGICO> ;<TIPADO>

| ε

**<PARAMETROS>::=**<P> <PP>)

| )

**<P>**::= <tipo> <identificador>

**<PP>**::= , <P><PP>

| ε

**<INSTRUCCIONES>**::= <COMENTARIO BLOQUE><INST>

| <COMENTARIO SIMPLE><INST>

| <DECLARAR><INST>

| <DECISION_DENTRO><INST>

| <IF><INST>

|<WHILE><INST>

| <FOR><INST>

| <SWITCH><INST>

| <IMPRIMIR><INST>

ε

**<SECPARAMETROS>**::=<SP> <SPP>

**<SP>**::= <LOGICO>

**<SPP>**::= , <SP><SPP>

| ε

**<LOGICO>::=**<L><LP>

**<L>::=**<NOT><EXPRESION><LP>

**<NOT>**::=! <NOT>

| ε

**<LP>::=** && <L>

| || <L>

| ε

**<EXPRESION>**::=<E> <COMPARACION>

**<COMPARACION>**::= == <E>

| != <E>

| > <E>

| < <E>

| <= <E>

| >= <E>

| ε

**<E>**::=<T><EP>

**<T>**::=<NOT><F><TP>

**<F>**::= <dato>

" <dato> "

' <dato> '

<identificador><O>

(<EXPRESION>)

**<O>**::= ( <OP>

| ε

**<OP>**::= )

|<SECPARAMETROS>)

**<TP>**::=*<F><TP>

| /<F><TP>

| ε

**<EP>**::=+<T><EP>

| -<T><EP>

|++<EPP>

|--<EPP>

| ε

**<EPP>**::=+<T><EP>

| -<T><EP>

| ε

**<IMPRIMIR>**::=Console.WriteLine ( <SALIDA> ;

**<SALIDA>**::=<LOGICO> )

| )

**<DECISION_DENTRO>**::=<identificador><DD> ;

**<DD>**::= == <LOGICO>

| (<OP>

**<DECLARAR>**::=<tipo> <CUERPO>

**<CUERPO>**::=<identificador> <LISTA> <DECLARARP>

**<LISTA>** ::= , <identificador> <LISTA>

| ε

**<DECLARARP>::=** = <LOGICO><DECLARARPP>

| ε

**<DECLARARPP>**::= ,<CUERPO>

| ε

**<SWITCH>**::= switch ( <LOGICO> ) { <CASE> }

**<CASE>**::= case <LOGICO> : <BREAK> <CASEP>

|<DEFAULT>

**<BREAK>::=<INSTRUCCIONES> <BREAKP>**

**<BREAKP>::=break;**<BREAK>

| ε

**<CASEP>**::= case <LOGICO> : <BREAK> <CASEP>

| ε

**<DEFAULT>**::= default : <INSTRUCCIONES> break ; <BREAK>

| ε

**<FOR>::=** for ( <FORDEC> ; <LOGICO> ; <LOGICO> ) { <BUCLE> }

**<FORDEC>::=** <DECLARAR>

| <DECISION_DENTRO>

**<BUCLE>**::= <INSTRUCCIONES><FORPP>

**<BUCLEP>**::=break;<FORP>

| continue;<FORP>

| ε

**<WHILE>**::= while( <LOGICO> ) { <BUCLE> }

**<IF>**::= if ( <LOGICO> ) { <INSTRUCCIONES> } <ELSE_IF>

**<ELSE_IF>::=else <ELSEP_IFP>**

| ε

**<ELSE_IFP>::=if (<LOGICO>) {<INSTRUCIONES} <ELSE_IF>**

**| {** <INSTRUCCIONES> }