Manual de Usuario

Generador de Código Embebido y visualizador de HTML

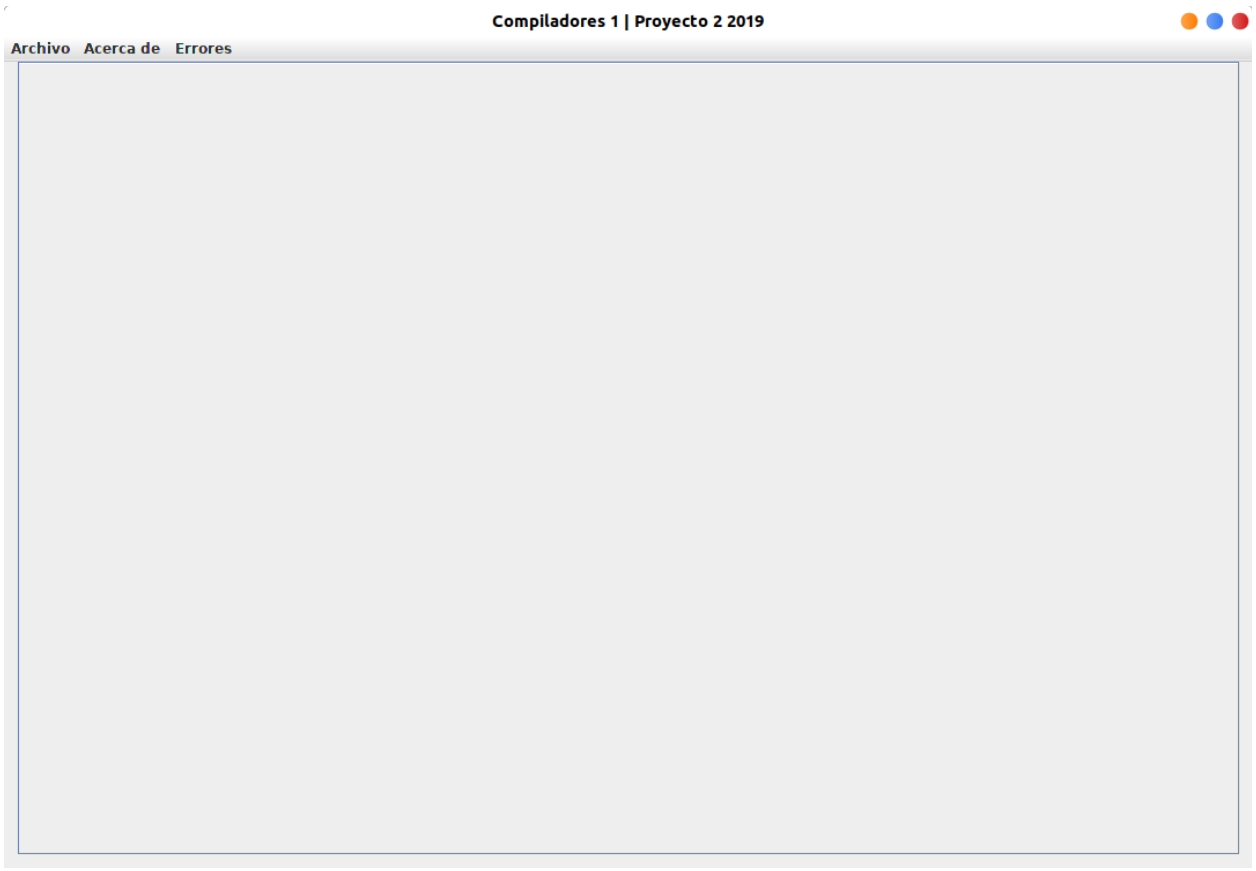
Compiladores 1

Jonathan Bryant Daniel Chiroy Rivera
201631722

Índice

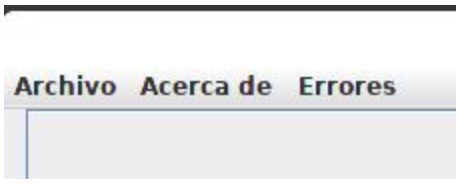
Pantalla Principal

Al ingresar al programa, la ventana principal es la siguiente:



Menú de la Aplicación

El programa le brinda al usuario la posibilidad de importar archivos con contenido en texto plano desde el Menú en la pestaña de Archivo, además puede guardar los cambios realizados. Adicional a ellos, se encuentra la información del autor y el generador de archivos XML con los errores.



Pestañas

En esta sección del programa tenemos la sección donde podemos escribir nuestros comandos de código embebido junto con nuestro código HTML y una segunda ventana donde se desplegará nuestro html ya interpretado.

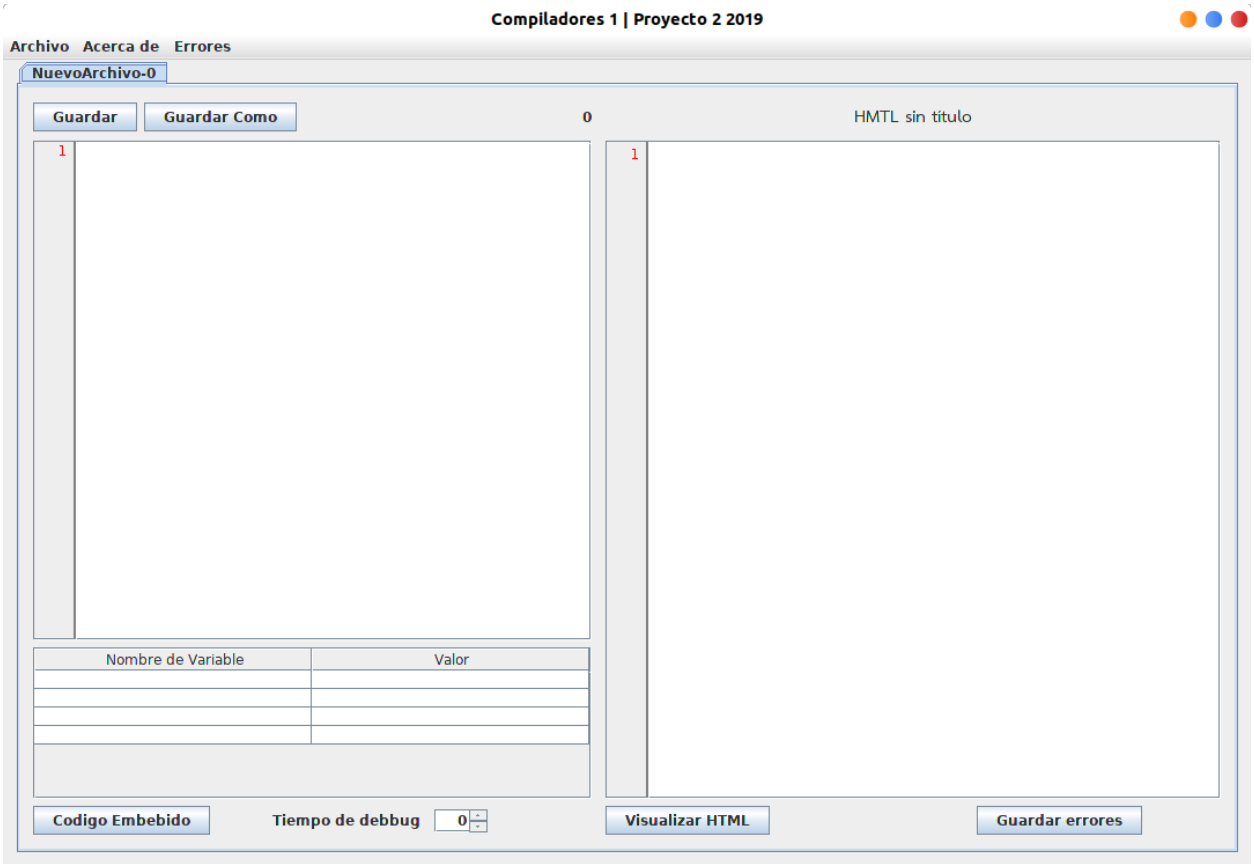
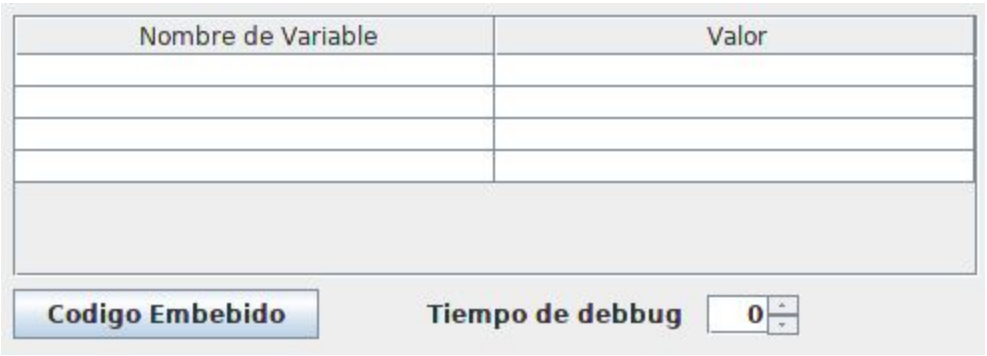


Tabla de Símbolos

En esta parte del programa podemos seleccionar el tiempo (de 0 a 10 segundos) que deseamos en que se pause nuestro programa para que podamos apreciar que valores van tomando nuestras variables.



¿Cómo generar código embebido?

Instrucciones

Las instrucciones que deberán ser reconocidas por el intérprete serán las siguientes:

Definición de variables

Todas las variables utilizadas dentro del código embebido deben ser previamente declaradas. La declaración de las variables puede hacerse en cualquier parte siguiendo la siguiente sintaxis.

VAR id_variable: tipo_variable ;

Id_variable Es un identificador único para la variable (no puede declararse más de una vez la misma variable) que consiste en una cadena que debe empezar con una letra y puede seguir con letras, dígitos o guion bajo ("_") sin tener una longitud específica.

Tipo_variable Identifica el tipo de datos que va a manejar la variable y puede ser de tres tipos:

- ➔ INTEGER Para números enteros
- ➔ BOOLEAN Para los valores booleanos que pueden ser TRUE y FALSE
- ➔ STRING Para cadenas de caracteres

Operaciones aritméticas

Las operaciones involucran los símbolos aritméticos: "+", "-", "*" y "/". Hay de dos tipos, numéricas que involucran números enteros, variables (de tipo numérico) y todos los cuatro símbolos aritméticos y operaciones con texto que involucran variables (de tipo texto), texto y el operador "+" y equivale a la concatenación.

Los resultados de las operaciones pueden ser asignados a variables o utilizados dentro de operaciones relacionales.

Operaciones relacionales

Consisten en operaciones de comparación, el resultado es de tipo booleano y se utilizan los siguientes símbolos:

- > Mayor que
- < Menor que
- >= Mayor o igual que
- <= Menor o igual que
- = Igual que
- <> Diferente que

Operaciones lógicas

Son operaciones cuyos resultados son verdadero o falso (TRUE o FALSE) utilizando los operadores AND y OR. Se debe tomar en cuenta que las variables a considerar deben ser tipo booleano o bien las constantes de TRUE o FALSE. El operador AND tiene mayor prioridad que OR

Tanto las operaciones lógicas como las relacionales devuelven valores booleanos por lo que pueden ser asignadas a variables de tipo booleanas.

Condicional

Tiene la sintaxis siguiente:

IF condicion THEN bloque1 ELSE bloque2

Donde condicion es un valor booleano (o una expresión que devuelva un valor booleano) y bloque1 y bloque2 representan un bloque de instrucciones. Al igual que cualquier condicional, en caso de que el valor de condicion sea TRUE se ejecuta bloque1, en caso contrario se ejecuta bloque2.

Ciclos

Pueden ser de dos formas:

WHILE condicion bloque
Que ejecuta el bloque de instrucciones mientras el valor de condición sea TRUE.

FOR variable := ordinal1 TO ordinal2 bloque
Donde variable es una variable de tipo numérico previamente definida y ordinal1 y ordinal2 son dos números enteros donde ordinal2 es siempre mayor o igual que ordinal1. Para esta instrucción, se ejecuta el bloque de instrucciones siempre que ordinal1 sea menor o igual que ordinal2, aumentando en uno el valor de ordinal1 en uno por cada vez que se ejecuta el bloque.

Escritura

Con esta instrucción, se le indicará al intérprete que escriba al archivo de salida. La sintaxis es la siguiente:

PRINT(token[,token[...,token]]);

Donde token puede ser un número, texto, o variable. Todo lo que sea tipo texto o numérico debe ir entre comillas simples ('); en el caso de que sean variables, solamente se coloca el nombre de la misma. Si se desea escribir más de un token se deben separar por comas.

Bloque

El bloque que se menciona en las instrucciones del código embebido tiene la forma siguiente:

BEGIN [instrucciones] END; | instrucción

Donde instrucciones puede ser una o más de una instrucción e instrucción es cualquier instrucción reconocida en este proyecto. Los corchetes son parte de la estructura de bloque cuando se usa BEGIN y END.

Asignaciones

Para asignar valores a las variables se debe usar el operador := (dos puntos seguido de igual) y el valor a asignar puede ser cualquier expresión media vez devuelva el mismo tipo de valor que el de la variable (análisis semántico). La asignación finaliza con punto y coma (;)