**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Lenguajes de Programación**

Avance Final Proyecto PHP



**Elaborado por:**

* Angie Argudo
* Miguel Parra
* Katherine Morales

2020 – 2S

1. **INTRODUCCIÓN**

PHP es un lenguaje de programación de código abierto tipo ‘scriptiping’, lo que significa que también usa secuencias de comandos del lado del servidor. Sus siglas hacen referencia a Hypertext Preprocessor. La sintaxis de PHP está basada en lenguajes como C, Java y Pearls sin embargo, a pesar de que PHP requiere menos tiempo y esfuerzo en ser aprendido, aún no es considerada como una herramienta de aprendizaje de programación.

Este lenguaje es usado especialmente por desarrolladores web ya que puede incrustarse de forma directa en HTML y además sus características de rapidez, flexibilidady pragmatismo proporcionan una ventaja sobre la mayoría de alojamientos web. PHP es el motor de los CMS más populares como WordPress, Joomla, Drupal o Magento, etc. y encontramos potentes frameworks como Lavarel o Symfony [1].

El proyecto desarrollado consolida los conceptos principales que debe tener todo lenguaje de acuerdo a su orden de precedencia. Inicialmente se estableció un analizador léxico que sea capaz de reconocer los tokens necesarios descritos en el Avance 0. El segundo aspecto considerado fue la definición de las gramáticas y su correspondiente analizador, en este caso léxico, utilizando las gramáticas de estructuras que admite el lenguaje. Para finalmente diseñar una interfaz gráfica que analice una entrada de código y se visualice un análisis léxico y sintáctico del mismo.

Para el diseño e implementación utilizó el lenguaje interpretado de Python a través de las IDE de PyCharm y Visual Studio Code. Los analizadores requirieron las librerías de PLY, de las cuáles se utilizaron el módulo lex y el módulo jacc para el analizador léxico y sintáctico respectivamente. La librería Tkinter fue útil para crear la interfaz, de modo que se implementaran los analizadores para realizar las respectivas pruebas que dan origen a nuestro analizador de lenguaje PHP. Por último, las herramientas colaborativas usadas fueron GitHub y Repl.it.

1. **DOCUMENTACIÓN**

**ALCANCE**

Se describe a continuación el alcance de este proyecto con las respectivas reglas.

La etiqueta de apertura que PHP busca al analizar un fichero es: **<?php**.

La etiqueta de cierre es: **?>**.

Estas indican dónde inicia y termina el código a interpretar; el punto y coma (;) marca el final de toda sentencia.

## VARIABLES

|  |  |  |
| --- | --- | --- |
| **IDENTIFICADOR** | **DESCRIPCIÓN** | **EJEMPLO** |
| $nuevaVariable | Se inicializa con un signo de dólar seguido del nombre que es sensible a minúsculas y mayúsculas, el nombre de la variable tiene que empezar con una letra o carácter underscore (No puede iniciar con un número), luego de lo anterior, este puede ser seguido de cualquier cantidad de letras, números o underscore. | $nuevaVariable = 50; |

## CONSTANTE

Una constante es un nombre para un valor simple, cuyo valor no puede variar durante la ejecución.

|  |  |
| --- | --- |
| **IDENTIFICADOR** | **DESCRIPCIÓN** |
| const nombreConstante | Un nombre de constante válido empieza por una letra o guion bajo, seguido por cualquier número de letras, números o guiones bajos. |

## DATOS PRIMITIVOS

|  |  |  |
| --- | --- | --- |
| **IDENTIFICADOR** | **DETALLE** | **EJEMPLO** |
| Boolean | Valores booleanos. | True  False |
| Integer | Valores enteros. | 10 |
| Float | Valores de coma (punto) flotante. | 5.5 |

## DATOS ESTRUCTURADOS

|  |  |  |
| --- | --- | --- |
| **IDENTIFICADOR** | **DETALLE** | **EJEMPLO** |
| String | Valores de cadena. | $cadena= “hola”; |
| Array | Mapa ordenado | $array = array(  "foo" => "bar",  "bar" => "foo",); |
| Iterable | Cualquier array u objeto que implemente la interfaz Traversable. | function foo(iterable $iterable) {  foreach ($iterable as $valor) {  // ...  }  } |
| Resource | Recursos no representables de forma nativa. | $conn = mysqli\_connect(localhost,"root","admin","animals"); |
| NULL | Representa la ausencia de valor para una variable. | $numeroNulo = NULL; |

## SIGNOS

|  |  |
| --- | --- |
| **IDENTIFICADOR** | **DESCRIPCIÓN** |
| [] | Corchetes |
| {} | Llaves |
| () | Paréntesis |
| ; | Punto y coma |
| , | Punto |
| // | Comentarios |

## OPERADORES ARITMÉTICOS

|  |  |
| --- | --- |
| **IDENTIFICADOR** | **DESCRIPCIÓN** |
| = | Igual |
| + | Suma |
| - | Diferencia |
| \* | Multiplicación |
| / | División, devuelve valor flotante a menos que los dos sean integers o strings que se conviertan a integers. |
| % | Módulo, el resultado tendrá el signo del dividendo. |
| \*\* | Potenciación |

## OPERADORES LÓGICOS

|  |  |
| --- | --- |
| **IDENTIFICADOR** | **DESCRIPCIÓN** |
| > | Mayor que |
| >= | Mayor o igual que |
| < | Menor |
| <= | Menor o igual que |
| == | Comparación igual |
| === | Idéntico |
| !=  <> | Diferente |
| !== | No idéntico |
| => | Flecha |
| <=> | Nave espacial |
| ?? | Fusion null |

## PALABRAS RESERVADAS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| do | break | echo | or | array |
| const | else | static | xor | key |
| if | elseif | var | not | current |
| for | True | global | function | next |
| while | False | and | return |  |

## ESTRUCTURAS DE CONTROL

|  |  |  |
| --- | --- | --- |
| **IDENTIFICADOR** | **DESCRIPCIÓN** | **EJEMPLO** |
| if | La estructura de control if permite la ejecución condicional de fragmentos de código. | <?php  If ($x > $y){  Echo "$x es mayor que $y”;  } |
| while | La estructura de control while permite que se ejecuten las sentencias dentro del while siempre y cuando se evalúe como true el valor de la expresión que se encuentra en los paréntesis, se comprueba cada vez al inicio del loop, y la ejecución no se detendrá hasta que finalice la iteración, si la expresión while se evalúa como false, las sentencias no se ejecutarán ni siquiera una vez. | <?php  $i = 1;  while($i <= 10){  Echo $i;  $i++;  } |
| for | La estructura de control for crea ciclos de repetición en código, estos se basan en la estructura condicional que llevan entre los paréntesis. | <?php  for ($i = 1; $i <= 10; $i++) {  echo $i;  } |

MÉTODO

|  |  |  |
| --- | --- | --- |
| **IDENTIFICADOR** | **DESCRIPCIÓN** | **EJEMPLO** |
| echo | Echo no es una función, es una construcción del lenguaje y es usada para mostrar información al usuario, al no ser una función este no requiere de paréntesis al enviar los datos y no retorna nada. | <?php echo "<h2>PHP is Fun!</h2>"; echo "Hello world!<br>"; echo "I'm about to learn PHP!<br>"; echo "This","string ", "was ", "made ", "with multiple parameters."; ?> |

FUNCIONES

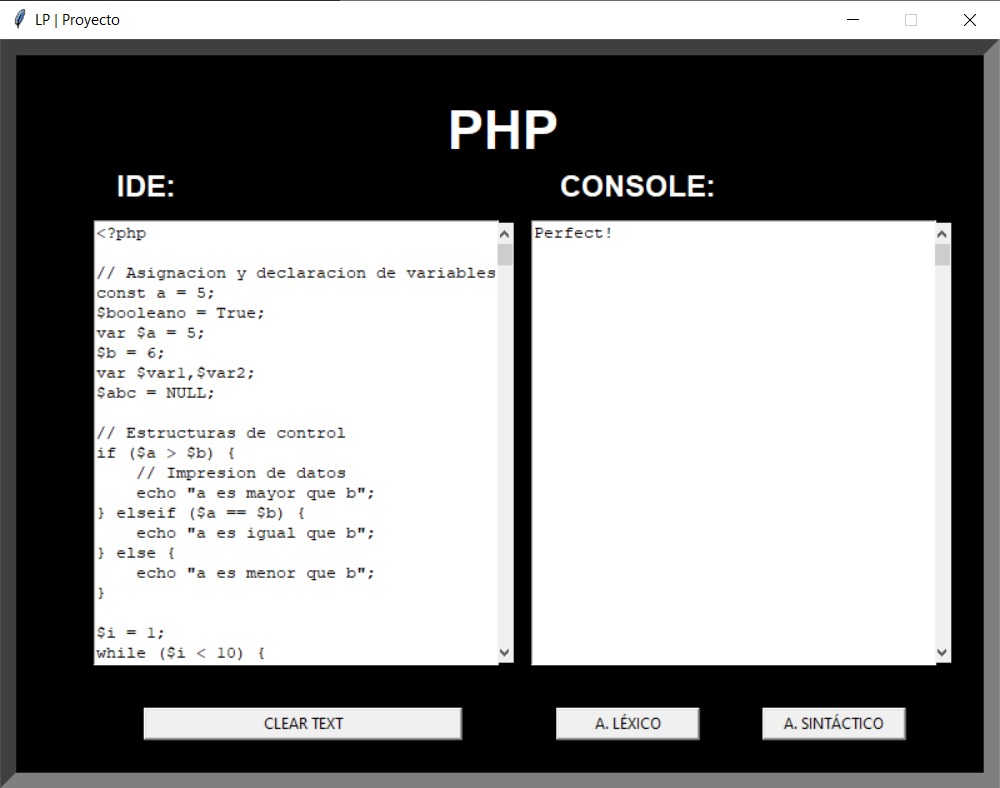
|  |  |
| --- | --- |
| **IDENTIFICADOR** | **EJEMPLO** |
| function | <?php  function primera\_funcion() {  echo 'Hola mundo';  }  segunda\_funcion();  ?> |

1. **RESULTADOS**

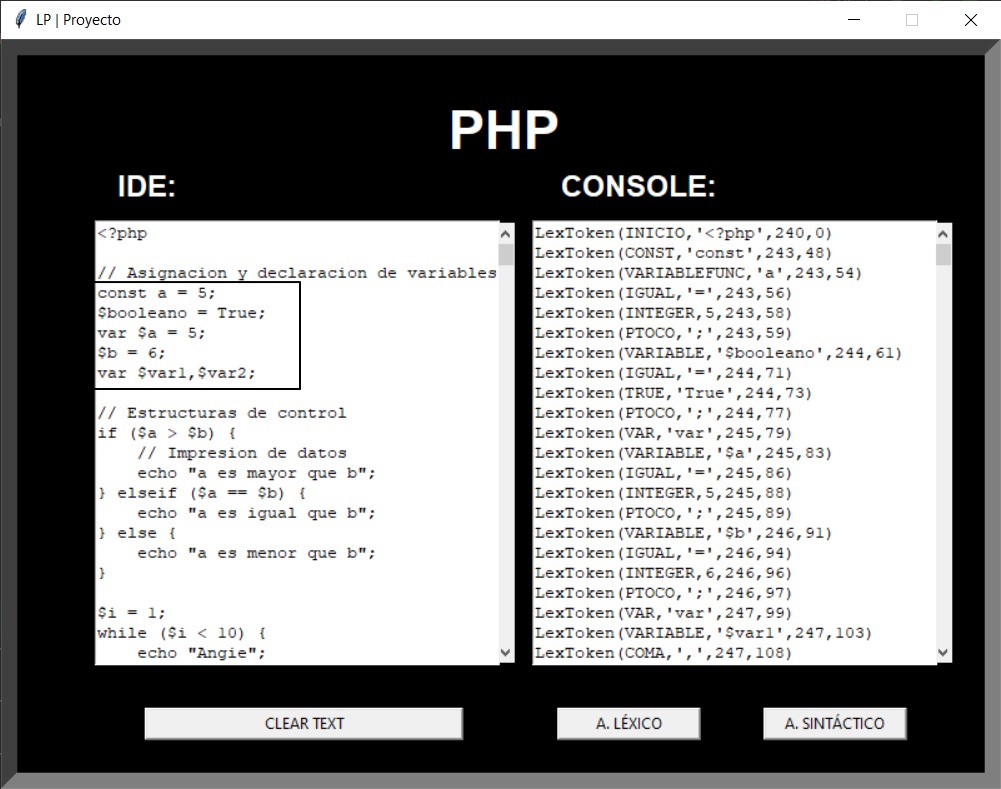
**ALGORITMO DE PRUEBA**

**RESULTADOS GENERADOS**

Sintáctico sin error.



Léxico sin error.



Con error.

Se obtuvo una interfaz que permite ingresar un código y a través del analizador léxico y analizador sintáctico desarrollado, se comprueba correctamente las reglas de asignación, comparación, funciones y métodos descritos en este documento del lenguaje de programación PHP.

El proyecto cumplió con el 80% de la propuesta inicial y se propuso la adición de otro porcentaje para funciones. En principio se incluía la totalidad de palabras reservadas del lenguaje, con ello algunos métodos especiales de datos estructurados, de lectura de datos, así como algunos de impresión. Se redujo la cantidad de datos estructurados debido a que incrementaría la extensión del proyecto. Es por ello que se delimitó el proyecto de acuerdo las reglas ya mencionadas.

Respecto a la sintaxis, la facilidad de lectura y estructura de la información es adecuada. Para la función asignación se puede asignar variables usando el ámbito o no, se intentó utilizar el empty para esto, pero no funcionó, de la misma forma para declaración, con lo cual la calidad del proyecto puede mejorarse respecto a estos aspectos.

1. **CONCLUSIONES**

* del LP asignado

A pesar de que el lenguaje

* del proyecto
* de las herramientas utilizadas.