

# **COMPILADORES**

## **PROYECTO-PRIMERA PARTE**

### **CONSTRUCCIÓN DE UN ANALIZADOR LEXICOGRÁFICO**

#### **I. Objetivo general**

Aplicar las principales técnicas de diseño de lenguajes de programación y de sus compiladores.

#### **II. Objetivos específicos**

- Diseñar un lenguaje de programación a manera de aplicar los criterios básicos a considerar en su utilización.
- Construir un analizador lexicográfico para un lenguaje de programación, que tome en cuenta los aspectos básicos pero fundamentales involucrados en esta fase de compilación.
- Que el estudiante enfoque el problema de la construcción de compiladores y de los ambientes en que estos trabajan, usando principios de la programación de sistemas y de la ingeniería de software.
- Que el estudiante encuentre un sitio natural para la aplicación del conocimiento adquirido en otras áreas de las ciencias de la computación, tales como: programación, estructura de datos, sistemas operativos y teoría matemática de la computación.

#### **II. Descripción**

Se diseñará un lenguaje de programación no robusto pero que cuente con las herramientas esenciales de programación. Este diseño podrá hacerse de manera independiente o en base a los lenguajes de programación existentes.

Luego, se construirá un analizador lexicográfico para dicho lenguaje de programación, así como una interfaz con el usuario, que se continuará desarrollando hasta el final del proyecto, pensando en la creación de un entorno de desarrollo liviano.

#### **IV. Requerimientos**

El lenguaje de programación a diseñar, deberá contar para este primer nivel del proyecto con:

- Manejo de variables.
- Manejo de al menos 5 tipos de datos.
- Al menos 3 instrucciones de selección.
- Al menos 3 instrucciones iterativas.
- Instrucciones de entrada y salida de datos.
- Creación de sub-algoritmos.

- Paso de parámetros por valor y por referencia.
- Manejo de arreglos.
- Manejo básico de archivos (apertura y volcado).
- Al menos 10 funciones predefinidas.

Este lenguaje deberá ser presentado al profesor de la materia, por los integrantes del grupo, de manera informal en la semana del **29 de agosto al 2 de septiembre**, de forma que puedan hacerse sugerencias sobre el diseño a vista de las próximas fases del proyecto.

Una vez diseñado el lenguaje de programación, se deberá:

- a. Especificar las expresiones regulares que definen los elementos del lenguaje, especialmente la de los identificadores.
- b. Especificar expresiones regulares para números enteros y números reales
- c. Indicar como se construyen comentarios en el lenguaje.

Posteriormente, el grupo procederá a construir el analizador lexicográfico para estas componentes de léxico, usando las técnicas vistas en clase, con los siguientes requerimientos específicos:

1. Se debe sustituir la búsqueda lineal del analizador lexicográfico, por una búsqueda binaria.
2. Se deben omitir los comentarios bien formados (los comentarios no son tokenizados)
3. Se escribirá además una interfaz para el usuario con las siguientes funciones básicas:
  - abrir un archivo
  - crear y/o modificar texto (editar)
  - copiar texto
  - pegar texto
  - guardar texto
  - ejecutar el analizador lexicográfico sobre el archivo abierto.

La salida de la ejecución debe ser el programa fuente (texto) tokenizado. Esta salida se debe mostrar en una ventana “hija” dentro de la interfaz, y debe tener el siguiente formato:

```
lexeme -> token
lexeme -> token
.....
lexeme -> token
```

Por ejemplo, para PL0, teniendo un código fuente como:

```
var a;
a := 2.
```

La salida en la interfaz debe ser:

```
var -> vartok
a   -> identificador
;   -> puntoycoma
a   -> identificador
:=  -> asignacion
2   -> numero
.   -> punto
```

La interfaz para el usuario debe ofrecer la posibilidad de configurar los siguientes parámetros para el analizador lexicográfico:

- Tamaño máximo del buffer de líneas (MAXLINEA)
- Máximo número de dígitos en los enteros (MAXDIGIT)
- Máxima longitud de los identificadores (MAXID)

Estos parámetros serán almacenados en un archivo de texto denominado “param.txt”, con el formato:

```
parámetro;valor
parámetro;valor
.....
parámetro;valor
```

Este archivo de parámetros debe ser leído por el analizador lexicográfico para tomar en cuenta estos valores.

**DEBE DE ENTREGAR DEBIDAMENTE PARAMETRIZADO SU PRODUCTO, CON VALORES POR DEFECTO.**

**LA LLAMADA AL SCANNER DEBE SER AUTOMÁTICA, NO DEBE CONFIGURARSE NINGÚN TIPO DE RUTA PARA ELLO.**

**LA INTERFAZ DEBERÁ LLAMAR AL EJECUTABLE DEL SCANNER, NO SE TRATA DE MEZCLAR AMBOS CÓDIGOS O QUE UNO COMPILE AL OTRO.**

Una opción del menú de la interfaz debe ser la “Ayuda”. En esta opción se accederá a un archivo llamado “ayuda.pdf”, que contendrá la siguiente información para el usuario:

a. Descripción del diseño del lenguaje y de sus posibles aplicaciones. Debe hablarse de sus tipos de datos, instrucciones, etc.

- b. Palabras reservadas
- c. Expresión regular para los identificadores
- d. Expresiones regulares para números enteros y números reales
- e. Lista de operadores y caracteres especiales
- f. Forma de construcción de comentarios en el lenguaje
- g. Un ejemplo de programa para escribir “Hola mundo” en el lenguaje
- h. Cualquier otra observación (o limitación –o aclaración) que considere necesaria

Se requiere que el analizador lexicográfico sea escrito en C. **Ningún otro lenguaje es elegible**

La interfaz puede ser escrita en cualquier lenguaje, siempre y cuando el grupo se asegure de que será ejecutable en la computadora del profesor de la materia.

## **V. Entregables**

Un CD con una portada especificando el nombre completo de los miembros del grupo, el nombre del grupo y el nombre del lenguaje.

El CD contendrá:

### 1. En la raíz:

- Un archivo ejecutable llamado IDE.EXE (o su equivalente) que arranca la interfaz con el usuario
- Dos programas ejemplos en el lenguaje asignado llamados: ejemplo1.txt y ejemplo2.txt
- Los archivos:”param.txt” y “ayuda.pdf”

### 2. En una carpeta denominada “scanner”:

- Una subcarpeta llamada “fuentes” conteniendo el código fuente completo correspondiente al scanner. Puede contener todas las subcarpetas que sean necesarias.
- Una subcarpeta llamada “objeto” conteniendo el código objeto correspondiente al scanner. Puede contener todas las subcarpetas que sean necesarias.

### 3. En una carpeta denominada “ide”:

- Una subcarpeta llamada “fuentes” conteniendo el código fuente completo correspondiente a la interfaz con el usuario. Puede contener todas las subcarpetas que sean necesarias.
- Una subcarpeta llamada “objeto” conteniendo el código objeto correspondiente a la interfaz con el usuario. Puede contener todas las subcarpetas que sean necesarias.

**POR FAVOR: No incluya nada más en el medio. Revise que su medio no esté defectuoso y que esté libre de virus.**

## **VI. Notas importantes**

- Las únicas plataformas que se pueden presuponer disponibles son Windows 7, Windows 8 y Ubuntu 14.04.
- El producto será probado por el profesor desde la interfaz. NO SUPONGA QUE LOS ARCHIVOS CON LOS QUE SE PROBARA EL PROGRAMA ESTAN EN ALGUN SITIO PREDETERMINADO. NO SUPONGA QUE EL PROGRAMA EJECUTABLE DEBE ESTAR EN ALGUN SITIO PREDETERMINADO.
- Lo que usualmente se hace es copiar el contenido del CD a disco duro y probarlo desde ese lugar.

El CD se entregará el día **20 de septiembre de 2016**. La entrega será en el aula de la materia a las 6:30pm. Se exigirá puntualidad.

**Por ninguna razón se recibirá una segunda entrega del CD, ni se recibirá el CD más allá de la fecha y hora asignada.**