
MANUAL DE USUARIO

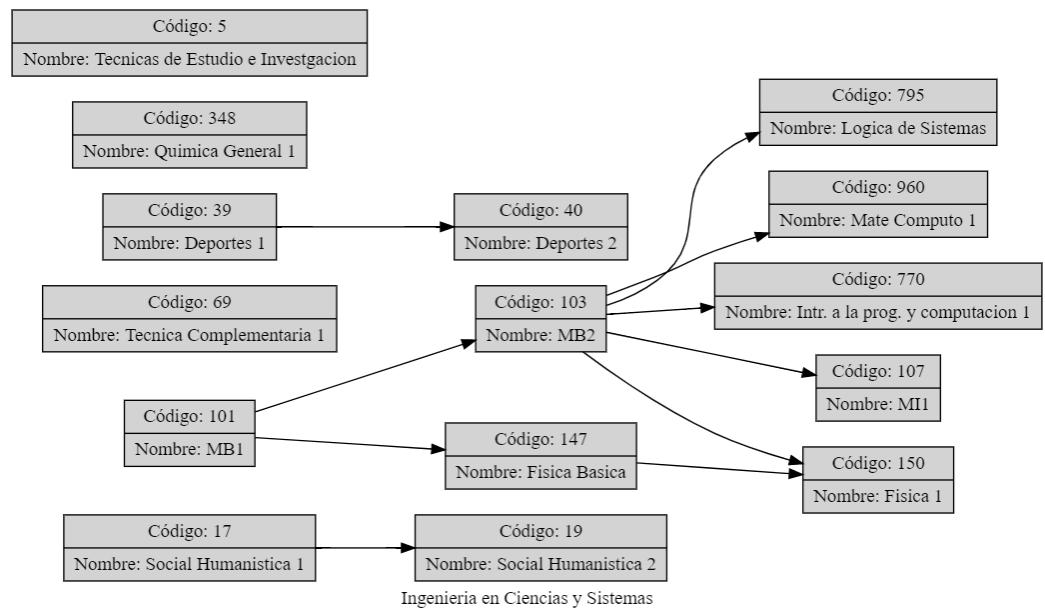
POR JAVIER GIRÓN

30 DE DICIEMBRE DEL 2021

OBJETIVO DEL PROGRAMA

El objetivo principal del programa es poder generar mallas curriculares de los cursos que se ingresen por medio de un lenguaje creado exclusivamente para el programa. El programa analizara las instrucciones que se ingresen en la parte izquierda y si todo esta correcto, esta las ejecutará.

Cabe mencionar que las redes que se generen serán entregadas en un archivo .dot con el nombre que se le asigne en consola con el fin de generar el código necesario para utilizar en graphviz, una vez creada la malla, solo se necesita de un interprete del archivo .dot para poder visualizar nuestra malla curricular.



INTERFAZ

BOTONES



ABRIR

El Botón de "Abrir" permite al usuario poder seleccionar por medio del explorador de archivos, seleccionar el archivo de instrucciones que se desea visualizar en pantalla, cabe mencionar que el formato de archivo que el programa acepta solamente es ".lfp"

ANALIZAR

El Botón de "Analizar" permite al usuario poder iniciar el analizador Léxico para verificar si todo lo que contiene el documento abierto pertenezca al lenguaje que maneja el programa, de igual manera inicia al finalizar el analizador léxico el analizador Sintáctico que verificará que la estructura que contiene el archivo sea el correcto, que se utilicen las palabras reservadas de manera correcta y que tengan la sintaxis correcta. Mas adelante se mostrara la estructura de como se debe hacer el archivo.

REPORTES

El Botón de "Reportes" permite al usuario poder generar tres reportes en formato HTML, estos 3 archivos se dividen de la siguiente manera:

- **Tokens Ingresados**
- **Errores Léxicos**
- **Errores Sintácticos**

Cada uno detalla con precisión la línea donde se encuentra, la columna y su descripción. Esto con el fin de poder agilizar la búsqueda de errores en el programa.

Cabe mencionar que el cuadro de texto izquierdo será en el cuadro en el cual se ingresen las instrucciones y el derecho en el cual se vera reflejado el resultado.

ESTRUCTURA DEL DOCUMENTO



La idea principal del programa es poder identificar lo que se le esta ingresando por consola, sin embargo estas instrucciones deben seguir un orden y cada instrucción tiene su palabra reservada que ejecutará cada una de las instrucciones.

A continuación te mostraremos cada una de sus palabras reservadas y el orden que deben cumplir para que se elabore de manera correcta el documento.

```
Proyecto_2LFP >  pruebas.lfp
1  nombre_de_red('Ingenieria en Ciencias y Sistemas');
2  crearcurso(1, 17, 'Social Humanistica 1', [12]);
3  crearcurso(1, 101, 'MB1', [2]);
4  crearcurso(1, 102, 'MB2', [22]);
5  consola('Cursos creados');
6  consola('Cursos creados');
7  consolaln('Cursos de la red Ingenieria en Ciencias y Sistemas');
8  consolaln('hola mundo');
9  cursoPorCodigo(1);
10
11 crearcurso(1, 17, 'Social Humanistica 1', [12]);
12 crearcurso(1, 101, 'MB1', []);
13 crearcurso(2, 103, 'MB2', [101]);
14 crearcurso(2, 147, 'Fisica Basica', [101]);
15 crearcurso(3, 150, 'Fisica 1', [103,147]);
```

PALABRAS RESERVADAS

Para ejecutar cada instrucción, el programa solicitará palabras específicas que las ejecutará dependiendo de cuales sean, estas palabras y su estructura son las siguientes:

NOMBRE_DE_RED

Permite configurar el nombre de la carrera o red de estudios con la que se estará trabajando.

Ejemplo:

```
nombre_de_red('Ingenieria en Ciencias y Sistemas');
```

CREARCURSO

Permite crear muchos cursos que pertenecen a la red de estudios con la palabra reservada crearcursro seguida de los siguientes parámetros encerrados entre paréntesis y separados por comas, al finalizar la instrucción lleva un punto y coma:

- Semestre: Número de semestre a la cual pertenece el curso, es un número entero.
- Código: Código del curso que se está agregando, es un número entero.
- Nombre del curso: Nombre del curso que se está agregando, es una cadena de texto.
- Prerrequisitos: Arreglo de códigos de cursos prerrequisitos para llevar el curso que se está agregando, se encierran entre corchetes y se separan por comas si hubiera mas de un curso prerrequisito, de no tener prerrequisito los corchetes estarán vacíos, estos codigos de cursos son números enteros.

Ejemplo:

```
crearcursro(1, 17, 'Social Humanistica 1', [12]);
```

```
crearcursro(1, 101, 'MB1', [2]);
```

```
crearcursro(1, 102, 'MB2', [22]);
```

PALABRAS RESERVADAS

CONSOLA

Imprime por consola el valor dado por la cadena.

Ejemplo:

```
consola('Cursos creados');
```

CONSOLALN

Imprime en consola el valor dado por la cadena y realiza un salto de linea.

Ejemplo:

```
consolaln('Cursos de la red Ingenieria en Ciencias y Sistemas');
```

CURSOPORSEMESTRE

Imprime por consola la información de los cursos dados por el parámetro, en este caso imprimiría la información de todos los cursos de primer semestre.

Ejemplo:

```
cursoPorSemestre(1);
```

CURSOPORCODIGO

Imprime por consola la información del curso dado por el parámetro, en este caso imprimiría la información del curso con código 150 (Fisica 1).

Ejemplo:

```
cursoPorCodigo(150);
```

PALABRAS RESERVADAS

CURSO POR NOMBRE

Imprime por consola la información del curso dado por el parámetro, en este caso imprimiría la información del curso con nombre Técnica Complementaria 1.

Ejemplo:

```
cursoPorNombre("MB1");
```

CURSOS PRERREQUISITOS

Imprime por consola los cursos prerrequisitos del curso dado por el parámetro, en este caso imprimiría la información del curso con código 150 (Fisica 1).

Ejemplo:

curtosPrerrequisitos(150);

CURSOS POST REQUISITOS

Imprime por consola los cursos postrrequisitos del curso dado por el parámetro, en este caso imprimiría la información del curso con código 103 (MB2).

Ejemplo:

cursosPostrequisitos(150);

GENERARRED

Guarda el grafo de la red de estudios con el nombre dado.

Ejemplo:

```
generarRed('Ingenieria en Ciencias y Sistemas');
```

