



Sección	Catedrática	Tutor académico
A-	Inga. Damaris Campos de López	Néstor Alexander Tzunun Mérida
B-	Inga. Zulma Aguirre	José Alberto Velásquez Orozco

ENUNCIADO SEGUNDO PROYECTO

OBJETIVOS GENERAL:

Aplicar los conocimientos adquiridos del curso Lenguajes Formales y de Programación en la creación de soluciones de software.

OBJETIVOS ESPECIFICOS:

- ✓ Esta práctica tiene como objetivo reforzar los conocimientos obtenidos en el curso.
- ✓ Que el estudiante practique el análisis léxico y sintáctico.
- ✓ Que el estudiante realice un analizador léxico a partir de un AFD.
- ✓ Que el estudiante realice un analizador sintáctico a partir de la gramática tipo 2.
- ✓ Aplicar los conceptos aprendidos en clase y laboratorio.

DESCRIPCIÓN:

El proyecto DrawMath que el departamento de Matemáticas de la Facultad de Ingeniería quiere realizar consiste en una aplicación que grafique funciones matemáticas a partir del análisis de un archivo de entrada que contendrá las instrucciones necesarias para generar la imagen solicitada. Se debe realizar análisis léxico del archivo de entrada, esto para verificar que el archivo no contenga caracteres que no pertenecen al lenguaje seguidamente se debe realizar análisis sintáctico con la finalidad de verificar que las instrucciones se encuentren en el orden lógico correcto.

DESCRIPCIÓN DEL LENGUAJE:

El lenguaje se divide en diferentes secciones, permitiendo que sea de fácil edición para cualquier usuario. Las secciones que podemos localizar son:

- ✓ Variables y Constantes: En esta sección es posible definir variables y constantes que pueden utilizarse en cualquier lugar después de su definición.
- ✓ Función: Se definen las funciones que se quieren graficar.
- ✓ Graficas: Se generan las imágenes en la ruta especificada y en la galería de imágenes de la propia aplicación.

Definición de Lenguaje por sección:

El software está representado de manera jerárquica y contiene las siguientes secciones, las cuales vienen dentro de una sentencia.

```

Inicio Math
    //Declaracion de variables
    //Definicion de funciones
    //Creacion de graficas
Fin Math

```

Variables y Constantes: En esta sección se pueden definir todas las variables y constantes que pueden ser utilizadas en cualquier sección del lenguaje. Cada variable y constante contiene los siguientes atributos:

- ✓ **Nombre:** Es el nombre que se le asigna a la variable, el cual debe ser un identificador (Inicia obligatoriamente con una letra y seguido puede venir un conjunto de letras, números o guiones bajos).
- ✓ **Tipo:** es el tipo de la variable, el cual puede ser: cadenas, enteros y decimales.
- ✓ **Valor:** es el valor asignado a la variable según su tipo.

```

Inicio Declaracion Constantes y Variables
    Inicio constante
        nombre=cons1,
        tipo= Entero,
        valor = Multiplicar(4,Multiplicar(5,Potencia(Dividir(20,4),3))) //4*5*(20/4)^3
    Fin constante
    Inicio constante
        tipo= Decimal,
        valor = 20.32,
        nombre=cons2
    Fin constante
    Inicio variable
        nombre = var1,
        tipo = Entero,
        valor = Multiplicar(10*cons1) //10*cons1
    Fin variable
Fin Declaracion Constantes y Variables

```

Precedencia de operadores:

La precedencia de operadores para las operaciones aritméticas es definida por la agrupación en los paréntesis.

Función: En la sección de funciones, se definen las características de las funciones, dentro de los atributos establecidos están:

- ✓ **Nombre:** Es el nombre que se le asigna a la función el cual debe ser un identificador (Inicia obligatoriamente con una letra y seguido puede venir un conjunto de letras, números o guiones bajos).
- ✓ **Valor:** Es la parte en donde se le asigna a la función, acá es posible hacer uso de las constantes y variables declaradas anteriormente.

```

Inicio Declaracion Funciones
  Inicio Funcion
    nombre = funcion1,
    valor = Suma(Seno(x), var1) //y=Sen(x)+var1
  Fin Funcion
  Inicio Funcion
    nombre = funcion2,
    valor = Suma(RaizCuadrada(Multiplicar(var1,Potencia(x,5))), y) //z=sqr(x^5*var1)+y
  Fin Funcion
Fin Declaracion Funciones

```

Generación de gráficas: En la sección, se establecen todos los valores que definen la posición, tamaño, extensión de las gráficas de las funciones generadas. Contiene los siguientes atributos.

- ✓ **Nombre:** Se define el nombre de la imagen
- ✓ **x_positivo=** Se establece el límite de x positivo, para mostrar en la gráfica
- ✓ **x_negativo=** Se establece el límite de x negativo, para mostrar en la gráfica
- ✓ **y_positivo=** Se establece el límite de y positivo, para mostrar en la gráfica
- ✓ **y_negativo=** Se establece el límite de y negativo, para mostrar en la gráfica
- ✓ **Ancho:** Se define la cantidad en pixeles de la imagen para el ancho de la imagen (pixeles reales)
- ✓ **Largo:** Se define la cantidad en pixeles de la imagen para el largo de la imagen (pixeles reales)
- ✓ **Ruta:** Se define la ruta en donde la imagen será exportada, debe ser capaz de utilizar rutas relativas y absolutas (Al inicio este valor solamente debe ser guardada en la tabla de símbolos, cuando se elija exportar imágenes, se utilizan para guardar las imágenes en el disco en la ruta especificada)
- ✓ **Función:** Se debe escribir la función que será graficada.

```

Inicio Generacion graficas
  Inicio grafica
    nombre = grafical,
    x_positivo= 10,
    x_negativo= 10,
    y_positivo=cons1,
    y_negativo= 2,
    ancho=520,
    largo=520,
    ruta = "C:\users\usuario1\imagenes\",
    funcion= funcion1
  Fin grafica
Fin Generacion graficas

```

Consola:

Se debe mostrar una consola en la aplicación en donde se irá anotando todas las actividades que vaya realizando.

Operaciones Aritméticas:

Suma	Suma(4,5) Suma(cons1, 10) Suma(Var1,cons2)
Resta	Resta(10,6) Resta(10, Cons1) Resta(Cons1, Cons2)
Multiplicación	Multiplicar(10,3) Multiplicar(cons1,3)
Dividir	Dividir(30,3) Dividir(Cons1, Var1)
Potencia (10^2)	Potencia(10,2) Potencia(cons1,3)
Raiz Cuadrada	RaizCuadrada(25) RaizCuadrada(var1)

Funciones Trigonómicas

Seno	Seno(x)
Coseno	Coseno(x)
Tangente	Tangente(y)

Comentarios:

La aplicación acepta comentarios de dos tipos:

- ✓ Una línea: Se inicia con doble diagonal (//) y finaliza con el salto de línea
- ✓ Multilínea: Se inicia con diagonal asterisco (/*) y finaliza con asterisco diagonal (*//)

ENTORNO GRÁFICO:

El entorno gráfico debe tener como mínimo.

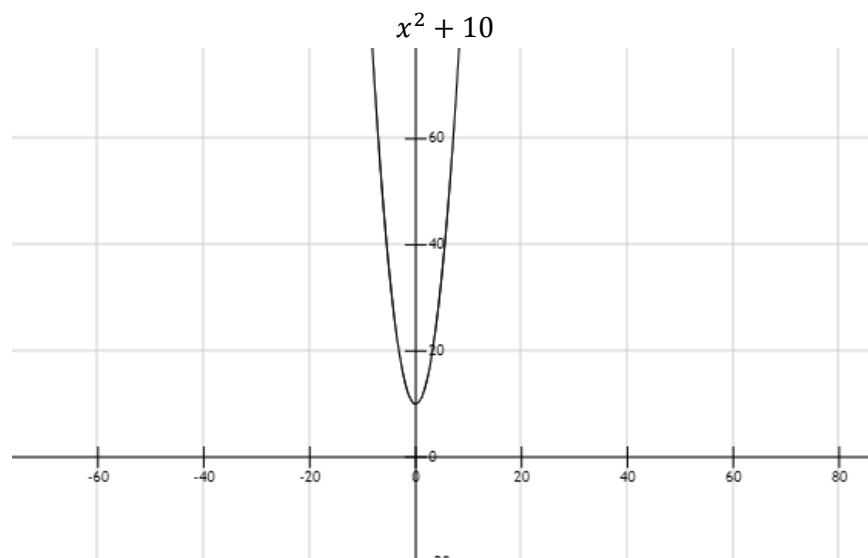
- Barra de Menú
 - Archivo
 - Abrir: La interfaz debe tener la opción de abrir archivos con extensión .draw
 - Guardar: La aplicación debe tener la opción de guardar el texto contenido en el editor. Si no ha sido guardado nunca, debe desplegar una opción para ingresar el nombre del nuevo archivo y la ubicación en la que desea guardar.
 - Guardar como: Esta opción permite guardar el archivo de entrada con otro nombre, se debe preguntar el nombre del nuevo archivo y la ubicación en la que desea guardar.
 - Análisis Léxico y Sintáctico: Debe realizar el análisis léxico y sintáctico de la entrada que se encuentra actualmente en el editor, debe generar todas las gráficas de las funciones en una galería de imágenes dentro de la aplicación.
 - Galería de funciones: Se debe mostrar en otro formulario todas las imágenes de las

funciones generadas, además debe tener la información de la función dada, y debe ser capaz de navegar entre imágenes.

- Exportar imágenes: Se deben exportar las imágenes de las funciones que se han generado en la ruta especificada en el archivo de entrada.
- Reportes de Errores: Esta opción debe desplegar los reportes correspondientes de las salidas del analizador, en esta se deben contener todos los errores léxicos y sintácticos, deben ser presentados en formato html.
- Ayuda
 - Acerca de: Debe mostrar en una ventana desplegable los datos del estudiante y del curso.
 - Manual de Usuario: Debe mostrar el manual de Usuario de la Aplicación
 - Manual Técnico: Debe mostrar el manual Técnico de la aplicación, incluyendo:
 - Para el analizador léxicos, el listado de componentes léxicos y su respectivo patrón (expresión regular), el DFA que corresponde por el método del árbol.
 - Para el analizador sintáctico, la gramática tipo 2.
- Editor de Texto: En esta área se definirán las instrucciones que recibirá para la creación de las gráficas de las funciones.
- Consola: En esta parte deberá mostrar la salida de las operaciones realizadas de asignación, creación de constantes, variables, funciones, entre otras.

Salida:

La aplicación debe ser capaz de generar las imágenes en el siguiente formato:



Reportes

El reporte de la Tabla de Símbolos se debe mostrar en una página HTML, la información en una tabla como se muestra a continuación:

#	Lexema	Fila	Columna	Id Token	Token
1	Valor	12	02	11	Reservada valor
2	5	12	12	12	Número
3	,	12	14	13	Final de Instrucción

El archivo de Errores, si existieran, se debe mostrar en una página HTML con la información en una tabla como se muestra a continuación:

ERRORES LÉXICOS

#	Fila	Columna	Carácter	Descripción
1	05	10		No pertenece al lenguaje
2	08	30	@	No pertenece al lenguaje
3	10	05	%	Desconocido

ERRORES SINTÁCTICOS

#	Fila	Columna	Descripción
1	05	10	Se espera el carácter '('
2	08	30	Se espera el carácter '='
3	10	05	Desconocido

Ejemplo Archivo de Entrada:

Inicio Math

 //Declaracion de variables y constastes

 Inicio Declaracion Constantes y Variables

 Inicio constante

 nombre=cons1,

 tipo= Entero,

 valor = Multiplicar(4,Multiplicar(5,Potencia(Dividir(20,4),3))) //4*5*(20/4)^3

 Fin constante

 Inicio constante

 tipo= Decimal,

 valor = 20.32,

 nombre=cons2

 Fin constante

 Inicio variable

 nombre = var1,

 tipo = Entero,

 valor = Multiplicar(10*cons1) //10*cons1

 Fin variable

 Inicio variable

 nombre = var2,

 tipo = Decimal,

 valor = Multiplicar(5.36*cons2) //5.36*cons2

```

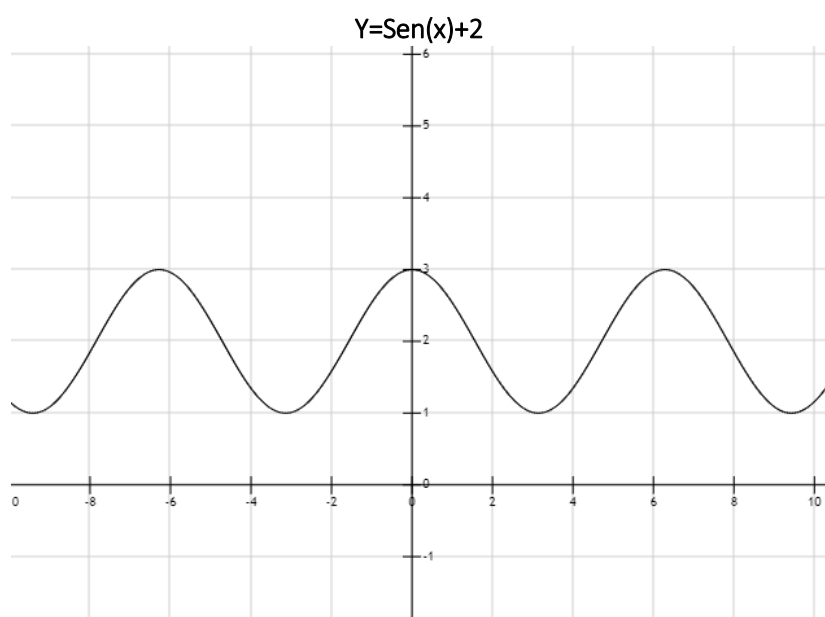
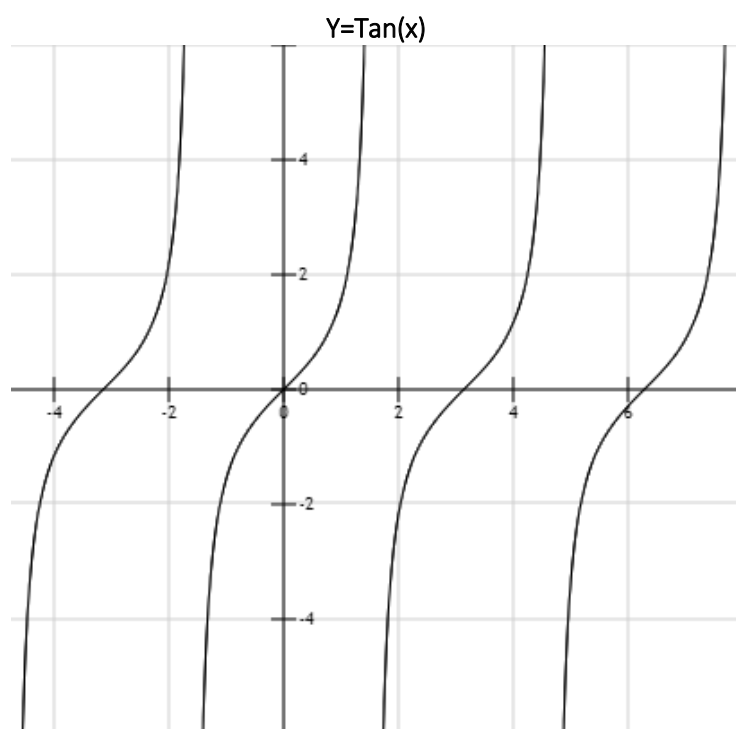
        Fin variable
Fin Declaracion Constantes y Variables

/*Definicion de
funciones */
Inicio Declaracion Funciones
    Inicio Funcion
        nombre = funcion1,
        valor = Suma(Seno(x), var1) //y=Sen(x)+var1
    Fin Funcion
    Inicio Funcion
        nombre = funcion2,
        valor = Suma(RaizCuadrada(Multiplicar(var1,Potencia(x,5))), y)
//z=sqr(x^5*var1)+y
    Fin Funcion
    Inicio Funcion
        nombre = funcion3,
        valor = Tangente(x) //y=Tan(x)
    Fin Funcion
Fin Declaracion Funciones

//Creacion de graficas
Inicio Generacion graficas
    Inicio grafica
        nombre = grafica1,
        x_positivo= 10,
        x_negativo= 10,
        y_positivo=cons1,
        y_negativo= 2,
        ancho=520,
        largo=520,
        ruta = "C:\users\usuario1\imagenes\",
        funcion= funcion3
    Fin grafica
    Inicio grafica
        funcion= funcion1,
        nombre = grafica1,
        x_positivo= 10,
        x_negativo= 10,
        y_positivo=6
        y_negativo= 2,
        ancho=520,
        largo=520,
        ruta = "C:\users\usuario1\imagenes\"
    Fin grafica
Fin Generacion graficas
Fin Math

```

Ejemplo Salida



Entregables

- Manual de Usuario
- Manual Técnico, debe incluir lo que se pide a continuación, lo cual debe coincidir con la programación de la respectiva fase:
 - Para el analizador léxicos, el listado de componentes léxicos y su respectivo patrón (expresión regular), el DFA que corresponde por el método del árbol.
 - Para el analizador sintáctico, la gramática tipo 2.
- Código Fuente, se verificará que corresponda al DFA (análisis léxico) y a la gramática tipo 2 (análisis sintáctico).
- Ejecutable de la Aplicación

Documentación a entregar de forma física el día de la calificación:

- Hoja de calificación (Original y una copia)

Notas importantes:

- La práctica se debe desarrollar de forma individual.
- Esta práctica se deberá desarrollar utilizando C# con Visual Studio 2012 en adelante.
- No se pueden usar ninguna librería o frameworks para realizar el análisis léxico y/o sintáctico.
- Para la creación de las gráficas se recomienda utilizar la Clase Graphics (System.drawing)
- El proceso de obtener tokens, se debe hacer a través de la implementación del autómata finito determinista desarrollado por el propio estudiante a través del método del árbol.
- No se puede agregar o quitar algún símbolo en el archivo de entrada. El proyecto deberá funcionar con los archivos de prueba que se disponga para la calificación, sin modificación.
- La calificación de la práctica será personal y durará como máximo 30 minutos, en un horario que posteriormente será establecido. Se debe tomar en cuenta que durante la calificación no podrán estar terceras personas alrededor, de lo contrario no se calificará la práctica.
- No se dará prórroga para la entrega del proyecto.
- **Copia parcial o total de la práctica tendrá una nota de 0 puntos, y se notificará a la Escuela de Sistemas para que se apliquen las sanciones correspondientes.**
- En el caso de no cumplir con alguna de las indicaciones antes mencionadas, NO se calificará el proyecto; por lo cual, se tendrá una nota de cero puntos.

Fecha de Entrega: 23 de octubre de 2017