4. DESCRIPCION DE LOS MODULOS.

Se utiliza un seudolenguaje con la finalidad de explicar los procedimientos y funciones.

4.1. ADMINISTRADOR DEL SISTEMA.

Es el modulo de mayor jerarquía que administra al Intérprete.

Input : Parámetro externo del Sistema Operativo.

```
Begin
  Captura el parámetro como archivo de trabajo;
  Blanquea y carga datos de los arreglos o tablas;
  Inicializa la pantalla;
  Carga y muestra datos de pantalla;
  Escoge una opción del menú;
End.
```

Escoger Opciones del menú.

```
Begin
Salida = falso;
Repeat
  leetecla;
  Si  tecla  es  <RETURN>  o <ENTER> enctonces ejecutar un
  proceso;
  Si proceso es Salir entonces salida = verdadero;
  If tecla es flecha then mover cursor;
Until Salida
end;
```

4.2. ANALIZADOR LEXICOGRAFICO.

La función SigTok, busca un token desde la cadena de trabajo.

```
Procedimiento Error(Número del Error);
Begin
  Ir a un lugar de la pantalla;
  Mostrar el Error y su descripción;
  Retornar al menú principal;
end;
```

Función Leecar;
Devuelve un carácter leído desde una tira leída de una línea del archivo del código fuente.

```
Begin
```

Verificar si la variable lógica Flag esta en Falso o

```
  Verdadero;

  Si esta en Verdadero, entonces retorna el carácter ya
  leído, de lo contrario Capturar el carácter de la tira
  leída del fuente;


  Si la tira ya fue leída entonces leer de otra línea del
  fuente;

  Verifica el final del archivo;
end;


Procedimiento ReponeTok;
  Simula el retorno del token al texto. Cambia de sentido
  de la variable lógica FlagTok.

Begin
  FlagTok = true;
end;

Procedimiento Unget;
  Simula el retorno del carácter leído anteriormente.
  Cambia de sentido de la variable lógica Flag.

Begin
  Flag = true;
end;

Función SigTok;
  Devuelve un Token;
Begin
  Si Flagtok = True entonces
        retornar el último token hallado y poner
        Flagtok = False
  Sino hallar el token de la siguiente manera
       lee letra por letra una palabra, hasta hallar un
       espacio en blanco.
  Verificar si la palabra es reservada, si no lo es,
  Instalar la palabra en la Tabla de Símbolos;
end;
```

4.3. ANALIZADOR SINTACTICO.

Siguiendo la sintaxis definida previamente se construyó el procedimiento para el análisis del código fuente.

```
Procedure Gencode;
  Genera las líneas del Código Intermedio,  una tras otra.
  Se dispone  de la  variable Cx,  que apunta  a la última
  línea de código.

Begin
  Cx = Cx+1;
  Reemplaza El operador y los argumentos del código;
End;

Procedure Genvartemp;
  Genera  variables  temporales  para  los  resultados
  intermedios de las operaciones lineales.
  Existe un contador de variables temporales Vtn.

Begin
       Crea variable _T<Vtn> en la tabla de símbolos;
end;

Procedure Parser;
  Es  el  modulo  principal  que contiene todo el resto de
  procedimientos analizadores;

begin
  Abre el archivo fuente;
  Verifica los primeros tokens como SISTEMA y su nombre;
  Analiza el Bloque;
  Cierra el archivo fuente;
end;

Procedimiento Bloque;
  Analiza un bloque. Dispone de apuntadores de niveles que
  se van  actualizando cada  vez que se hacen las llamadas
  recursivas.

Begin

  Actualiza los apuntadores de nivel, contador de bloque y
      recursión.
  Captura Sigtok;
  Si el  token es CONS, se analiza la definición de
       constantes y llama al SigTok;

  Si  el  token  es  VAR,  se  analiza  la  definición  de
  Variables y llama al Sigtok;

  Si  el  token  es  CONTROL,  se analiza la definición de
  Control llama al SigTok;

  Si el Token es  SUBSISTEMA,  analizar  la  definición de
  Subsistemas.
```

Si el Token es FUNCION, analizar la definición de Funciones.

Si el Token es INICIO, se ingresa a analizar una Proposición o sentencia. Actualizar apuntadores. Cuando se termine de analizar la Sentencia, verificar el Token FIN.

Si no es ninguno de los tokens anteriores, error.
end;

Procedimiento Sentencia;
Analiza una Proposición o Sentencia.
Begin
Captura el SigTok;

Si es el Token GRAFICA, analizar la proposición para graficar. Generar un línea de código intermedio para escribir en el archivo intermedio los datos referenciados por un arreglo llamado **Tabwrit;**

Si el Token es una variable o una función, entonces analizar el lado derecho para su evaluación invocando al procedimiento **EXPR** de Expresiones. En el caso de que la variable sea de Nivel (en la metodología de Dinámica de Sistemas) se generan líneas especiales de Código Intermedio para la integración respectiva.

Si el Token es una referencia de un SUBSISTEMA, entonces se analiza sus parámetros y se escriben líneas para el paso de parámetros.

Si el Token es de Proposición Compuesta, como INICIO .. FIN; se actualizan los apuntadores y se ingresa a analizar nuevamente la **Sentencia.**

Si el Token SI, se analiza la proposición de bifurcación SI..ENTONCES. Primero, analizar la Expresión Lógica, luego ENTONCES y Proposición adjunta. Si existiera el SINO analizarlo.

Si el Token es MIENTRAS, Analizar la proposición de iteraciones, que contiene un contador de iteraciones y un llamado al procedimiento **Sentencia.**

Si no fuera ninguno de lo anteriores, reponetok;

end;

4.3.1. EJEMPLO.

Enseguida se muestra un ejemplo de la traducción del código fuente hacia el código intermedio, mostrandose la Tabla de Símbolos que se genera.

Ejemplo de Código Fuente:

```
Sistema prueba1;
var   a,b,c : aux;
control  dt = 1;
         desde_t = 1;
         hasta_t = 10;
pon_gráfica(tiempo,1);
funcion cubo(r:aux):aux;
inicio
      cubo = r*r*r;
fin;
inicio
      a = tiempo;
      b = cubo(a)+a;
      si (a <= 5) entonces
         c = tiempo + b
      sino
         c = b - tiempo;
      gráfica(a,0);
      gráfica(b,1);
      gráfica(c,2);
fin.
```

La Tabla de Símbolos que aparece antes de ejecutarse el código intermedio, se halla en la siguiente forma:

Tabla de Símbolos :

Núm	LEXEMA	OBJETO	NIVEL	ATRIBUTO	VALOR
...					
74	PRUEBA1	9	1	0	0.00
75	A	0	1	VARIABLE	0.00
76	B	0	1	VARIABLE	0.00
77	C	0	1	VARIABLE	0.00
78	10	0	1	VALOR	10.00
79	CUBO	FUNCION	11	FUNCION	1.00
80	R	0	11	VARIABLE	0.00
81	_T1	0	11	VAR.AUX.	0.00
82	_T2	0	11	VAR.AUX.	0.00
83	_T1	0	1	VAR.AUX.	0.00

84	_T2	0	1	VAR.AUX.	0.00
85	5	0	1	VALOR	0.00
86	_T3	0	1	VAR.AUX.	0.00
87	_T4	0	1	VAR.AUX.	0.00
88	_T5	0	1	VAR.AUX	0.00
89	0	0	1	VALOR	0.00
90	2	0	1	VALOR	2.00

El Código Intermedio generado es :

Núm	OPERADOR	OPERANDO 1	OPERANDO 2	OPERANDO 3	Descripción
1	ASIG	9	0	41	Cond. Iniciales
2	ASIG	10	0	41	
3	ASIG	11	0	78	
4	RET	0	0	0	
5	MULTIPLICA	80	80	81	Función Cubo
6	MULTIPLICA	81	80	82	
7	ASIG	79	0	82	
8	RET	0	0	0	Fin de función
9	ASIG	75	0	39	Inicio Ejecución
10	ASIG	80	0	75	
11	CALL	79	0	0	Llamada a Cubo
12	ASIG	83	0	79	
13	SUMA	83	75	84	
14	ASIG	76	0	84	
15	VLE	75	85	86	
16	JGE	86	41	18	
17	JMP	0	0	21	Salto a 21
18	SUMA	39	76	87	
19	ASIG	77	0	87	
20	JMP	0	0	23	Salto a 23
21	RESTA	76	39	88	
22	ASIG	77	0	88	
23	WRAR	1	1	0	Escribe en
24	WRAR	2	1	0	Archivo
25	WRAR	3	1	0	
26	WRAR	4	1	0	
27	WRAR	0	0	0	
28	RET	0	0	0	Fin de código

4.4. EJECUTANDO EL CODIGO INTERMEDIO.

Una vez obtenida la tabla de Código Intermedio, se debe ejecutar línea tras línea.

```
Procedimiento Ejecuta Código Intermedio;
  Ejecuta la Tabla de código intermedio, desde un punto
  inicial.

Begin
  Si se han definido gráficos, definir el archivo de
  resultados intermedios;
  Capturar los datos del horizonte temporal;
  Mientras no se haya cumplido el tiempo final hacer
       Inicia la pila de control;
       Operar el código desde un punto inicial;
       Recalcular el tiempo actual;
     Cerrar el archivo intermedio.
end;

Procedimiento Opera Código ( Número );
  Con este procedimiento se ejecuta cada orden del Código
  Intermedio, utilizando las facilidades del Compilador de
  base.

Begin
  Iniciar a ejecutar el código intermedio en el lugar
  ubicado por el Número del parámetro;


  Controlar los operadores que apuntan a lugares de la
  Tabla de Símbolos;

  Ejecutar cada acción definida en el Código Intermedio.
  Actualizar el apuntador de la pila de control en caso de
  llamadas y retornos de subrutinas;

  En el caso de funciones matemáticas se calculan y se
  colocan los resultados en los lugares de la Tabla de
  Símbolos referenciados por los operadores respectivos.
end;
```

4.5. UNIDAD GRAFICA.

Con esta unidad se generan las gráficas en la pantalla de la computadora.

```
Procedimiento Inicializa Gráficos;
  Con esta opción se cambia al modo de gráficos y se
  actualiza los parámetros de la ventana, según sea la
  tarjeta que disponga la computadora.
```

```
Begin
  Detecta y inicializa los parámetros gráficos;
end;

Funciones para convertir números a cadenas para su
impresión;

Procedimiento Gráfico_general;
  Crea la ventana presentación de gráficos.
Begin
  Calcula el tamaño de la letra;
  Traza las líneas y los títulos;
end;


Procedimiento Gráfica;
Begin
  Inicializa Gráficos;
  Gráfico general;
  Traza las líneas con los datos que se hallan en el
  archivo intermedio;
end;
```

5. PERSPECTIVAS DE DESARROLLO FUTURO.

En el futuro se pueden desarrollar los siguientes módulos :

- Cambiar las estructuras de las tablas a estructuras de punteros para un más efectivo manejo de la memoria.
- Aumento de las funciones del Intérprete con los módulos para administrar tablas y retardos en la transmisión de materia e información.
- Escribir el código para un Editor de Textos integrado al sistema.
- Utilización de técnicas de Inteligencia Artificial en un módulo de busqueda de las mejores soluciones de ciertas variables en el tiempo, bajo un objetivo determinado previamente.

6. CONCLUSIONES Y RECOMENDACIONES.

1. Se ha construido un intérprete denominado Qenqo para simulaciones con modelos basados en Dinámica de Sistemas. Este muestra los resultados gráficos basados en una tabla generada por el ejecutor de su Código Intermedio. Este Código es generado por un Analizador Sintáctico basado en un lenguaje diseñado para éste propósito.

2. El intérprete tiene un ambiente integrado para la selección del archivo de código fuente, interpretar este y ejecutar el código intermedio. Además existen facilidades para el Análisis de Sensibilidad a una variable y el cambio de las condiciones iniciales de las variables de nivel o de control.

3. Sería conveniente aumentar las capacidades del intérprete en lo referente al análisis de retardos y el manejo de tablas. Asimismo, sería útil aplicar técnicas de Inteligencia Artificial para hallar las mejores soluciones (cual sería el mejor camino) en la busqueda un objetivo en el futuro determinado por el modelador.