

25

reg



**UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO**

FACULTAD DE QUÍMICA

**PROGRAMACIÓN
(EJEMPLIFICACIÓN CON TURBO PASCAL)**



EXAMEN PROFESIONALES
FAC. DE QUÍMICA

T E S I S
QUE PARA OBTENER EL TÍTULO DE:
INGENIERO QUÍMICO
P R E S E N T A:
ESTHER CORTÉS SOLÍS



MÉXICO, D.F.

1996

**TESIS CON
FALLA DE ORIGEN**

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

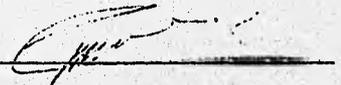
El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

JURADO ASIGNADO

PRESIDENTE: Prof. Caritino Moreno Padilla
VOCAL: Prof. Antonio Francisco Díaz García
SECRETARIO: Prof. David Gutiérrez Garavito
1^{er} SUPLENTE: Profa. Norma Gisela González Mariscal
2^{do} SUPLENTE: Prof. Victor Manuel Chávez Vargas

SITIO DONDE SE DESARROLLÓ EL TEMA: Laboratorio de Computación
Facultad de Química

ASESOR DEL TEMA: Caritino Moreno Padilla



SUSTENTANTE: Esther Cortés Solís



Doy gracias a Dios por todo lo que me ha dado.

Raquelito y Felipe : Su paso por mi vida fue brevísimo...
Gracias por todo.

A mi hermana Male: Por todo el cariño, paciencia y apoyo
con los que me haz acompañado a lo largo de mi vida.

A todos y cada uno de mis hermanos : María Magdalena,
Francisco, Esteban, Tomás, Samuel, Moisés, Raquel y
Violeta por el incansable apoyo que siempre me han dado.

Ebenezer.

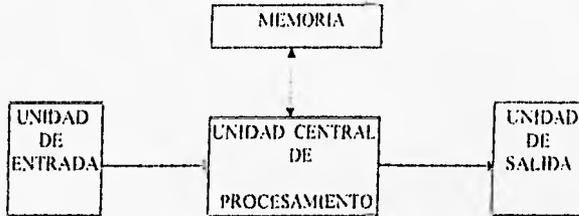
INDICE

	Pg
INTRODUCCIÓN	1
CAPÍTULO I Descripción de programación y desarrollo de algoritmos	5
CAPÍTULO II Descripción de algoritmo y representación mediante diagramas de flujo, representación de procesos secuenciales de elección y de repetición.	14
CAPÍTULO-III Aplicación de la programación en la resolución de problemas diversos.	35
CAPÍTULO IV Codificación en Pascal	63
CONCLUSIONES Y RECOMENDACIONES	80
BIBLIOGRAFÍA	81

INTRODUCCIÓN

LA MEMORIA PRINCIPAL.

Es un conjunto generalmente grande de celdas direccionables (la forma más sencilla de lograr acceder a ellas es organizarlas en forma de vector) donde una computadora almacena toda la información (datos, programas) que utilizará mientras esté encendida. Cualquier instrucción que el procesador efectúe deberá necesariamente residir en la memoria central, ya que es ahí donde la UCP (unidad central de procesamiento que contiene las unidades aritmética, lógica y de control) buscará la siguiente instrucción.



El parámetro más importante en una memoria es su velocidad de acceso, que mide el tiempo transcurrido desde que el procesador central pide la información contenida en una celda cualquiera hasta que ésta puede ser leída (o escrita). Los tiempos de acceso de las memorias de semiconductores se miden en unidades de millonésimas de segundo.

Cada elemento de información que se almacena en la memoria de la computadora está codificado por una combinación única de ceros y unos. Estos ceros y unos se llaman bits (del inglés binary digits = dígitos binarios). Cada bit está representado por un dispositivo electrónico que, en cierto sentido, está activado (uno) o desactivado (cero).

La mayoría de las pequeñas computadoras tienen memorias organizadas en múltiplos de ocho bits, llamados bytes. Normalmente, un byte puede representar un solo carácter (es decir, una letra, un dígito o un símbolo de puntuación); una instrucción puede ocupar uno, dos o tres bytes, y una sola cantidad numérica puede ocupar de dos a ocho bytes, dependiendo de la posición y el tipo del número.

El tamaño de la memoria de una computadora se expresa, generalmente, como múltiplo de $2^{10} = 1.024$ bytes. A este valor se le llama 1K. Las pequeñas computadoras tienen memorias cuyos tamaños van típicamente desde 64 K a 1.024K (1M) bytes.

El nombre genérico de estas memorias (estáticas o dinámicas) es RAM (Random Access Memory, memoria de acceso aleatorio).

Una desventaja de éstas con respecto a las memorias de ferrita es que los circuitos integrados pierden la información que tenían almacenada cuando se interrumpe la alimentación eléctrica. Esto obliga al diseño de memorias no volátiles, en las que se graba

información que ya no se pierde. Estos nuevos tipos de circuitos reciben el nombre genérico de ROM (Read Only Memory, memoria sólo de lectura), y se fabrican en varias configuraciones.

Debido a que la memoria central de una computadora es costosa y escasa, se vuelve necesario tener áreas adicionales de almacenamiento para guardar grandes cantidades de información de manera más económica. Además, la memoria central pierde los datos almacenados al interrumpirse el suministro de corriente eléctrica, por lo que resulta poco práctico utilizarla para almacenamiento permanente de datos.

Estas y otras razones dan lugar a la creación de unidades periféricas de memoria que reciben, en conjunto, el nombre de memoria auxiliar o secundaria (cintas y discos magnéticos).

La información residente en cualquiera de estos medios magnéticos recibe el nombre genérico de archivo. Un archivo está formado por un número variable de registros , generalmente de tamaño fijo, que pueden contener datos (numéricos o alfabéticos) a programas fuente escritos en algún lenguaje de programación.

INTRODUCCIÓN A PASCAL

Pascal es un lenguaje de programación de alto nivel y de propósito general que se ha derivado del Algol-60. Sus instrucciones se componen de expresiones de apariencia algebraica y de ciertas palabras inglesas, como begin , end, read, write, if, then, repeat, while, do. En este aspecto, Pascal se parece a muchos otros lenguajes de alto nivel. Pero Pascal también contiene ciertos rasgos singulares que han sido específicamente diseñados para estimular el uso de la *programación estructurada*.

El nombre Pascal fue elegido en honor de Blaise Pascal (1623-1662), brillante científico y matemático francés entre cuyos logros se encuentra la invención de la primera máquina de calcular mecánica en el mundo.

HISTORIA DEL LENGUAJE PASCAL

Pascal fue originalmente desarrollado a principios de los años 70 por Niklaus Wirth , en la Universidad Técnica de Zurich, Suiza. El propósito original de Wirth fue desarrollar un lenguaje de alto nivel, disciplinado, para enseñar programación estructurada. La definición original del lenguaje debido a Wirth se conoce, a veces, como *Pascal estándar o Pascal estándar según definición de Jensen y Wirth*.

CARACTERÍSTICAS DEL LENGUAJE PASCAL

Una de las características fundamentales del lenguaje Pascal es la capacidad para manejar un conjunto de instrucciones como si fuese una sola de tal manera que al codificar en este lenguaje podemos, con base en los requerimientos del algoritmo, utilizar códigos individuales o conjuntos de códigos.

A un código individual se le denomina instrucción simple y a un conjunto de códigos agrupados como una sola instrucción se les denomina instrucción compuesta o estructura, esto es lo que identifica a Pascal como un lenguaje estructurado.

No debe confundirse la capacidad de manejo de estructura con la capacidad de manejo de módulos o subprogramas. Todos los lenguajes poseen la capacidad de manejar módulos (subprogramas), pero no todos manejan instrucciones compuestas (estructuras).

DESCRIPCIÓN DE UNA ESTRUCTURA

Inicia con el código **begin** continuando con todas las instrucciones simples y finalizando con el código **end**;

```
begin
  -----;
  -----;
end;
```

Pueden existir estructuras dentro de estructuras

```
begin
  -----;
  -----;
  begin
    -----;
    -----;
    -----;
  end;
  begin
    -----;
    -----;
    -----;
  end;
  -----;
  -----;
end.
```

Se dice que un programa codificado en Pascal por sí mismo es una estructura que contiene en su interior un conjunto de estructuras, por el hecho de que inicia con el código begin y finaliza con el código end.

DESCRIPCIÓN DE LA PROGRAMACIÓN MODULAR

La técnica de programación modular, consiste básicamente en *fragmentar* el problema o tarea que se tenga que resolver en un subconjunto de tareas (módulos) que resultan más simples de resolver, las cuales al conjuntarse nos permiten llegar a una solución total por medio de soluciones parciales.

Una vez obtenidas las soluciones parciales, éstas se van integrando en forma secuencial y simultánea para conformar la resolución de la tarea global.

La aplicación de la Programación Modular está prevista en todos los Lenguajes de Programación, lo cuales a través del concepto de subprograma (subrutina) introducen esta forma de fragmentación de una tarea, en este caso una tarea global es seccionada en subtareas, cada subtarea es representada mediante un subprograma, en este caso, un programa denominada programa principal integra todos los módulos, activándolos en la secuencia correspondiente.



Programación Modular: Descomposición de una tarea en subtareas (Módulos), cada subtarea es más simple de analizar que la tarea global. Esto es análogo a la descomposición de procesos en subprocesos en Ingeniería Química.

La programación Modular frecuentemente se confunde con la clasificación de lenguajes estructurados, aún cuando se trata de dos conceptos completamente diferentes. Se puede aplicar la técnica de programación modular y codificar el algoritmo utilizando un lenguaje estructurado.

Un lenguaje estructurado dadas sus características evita la transferencia de control o ramificación excesiva dentro de un programa, ya que pretende agrupar esas transferencias en un conjunto de instrucciones (instrucción compuesta) que son realizadas en un mismo punto del programa sin necesidad de transferirse a otro segmento. Por eso se dice que es aberrante utilizar goto en pascal o C. A esta capacidad de manejar estructuras (instrucciones compuestas) se confunde con la capacidad de manejar módulos.

Programación Modular: La tarea se divide en Subtareas.

Lenguaje Estructurado: Una tarea o una subtarea es codificada utilizando instrucciones compuestas.

CAPÍTULO
I

Programación:

Cuando los objetivos de realización de una tarea están perfectamente definidos será posible proponer un camino del cómo o qué trabajo realizar para llevarla a cabo. Esta propuesta de realización implica hacer un análisis exhaustivo de qué acciones concretas conformarían el trabajo, pensando que éste sea efectuado con la mayor eficacia posible.

El planear estas acciones implica identificar el qué hacer y en qué orden ejecutarlo, a esta planeación se le denomina programación.

PROGRAMAR ES PLANEAR

Una vez especificadas las actividades y el orden en que deben realizarse, éstas pueden ser presentadas mediante una lista numerada para indicar la secuencia en que se realizarán o bien mediante un diagrama en el cual cada actividad se presenta con un símbolo y la secuencia de ejecución se expresa mediante flechas de dirección.

A esta lista secuencial (o diagrama) que constituye la descripción paso a paso del cómo llevar a cabo la tarea se le denomina algoritmo; de tal manera que el algoritmo es el producto de la programación.

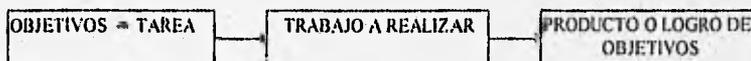
Es recomendable, para llevar a cabo de manera sistemática la programación de una tarea, desarrollar la programación por etapas. Recomendamos las siguientes:

ETAPAS DE PROGRAMACIÓN

1.- Descripción de la tarea:

Definiremos tarea como el propósito de obtener un producto o resolver un problema, lo que implica realizar algún trabajo de el cual depende la calidad del producto o respuesta obtenida.

La necesidad de obtener la solución de un problema o disponer de cierto producto nos conduce a pensar en el trabajo que tendríamos que desarrollar y cómo llevarlo a cabo para alcanzar los objetivos que fueron el origen de la implantación de la tarea.



Una vez comprendidos correctamente los objetivos representativos de la tarea podremos pensar en qué hacer para lograrlos. Si los objetivos son mal interpretados o no son claros, las acciones que se planeen para su logro serán incorrectas o serán demasiadas, lo que significa realizar el trabajo sin producción o premio alguno.

Como se dijo al inicio se trata de entender cuál es el verdadero problema asociado con la tarea, es decir cuál es el objetivo de realizar un trabajo.

2.- Formulación del modelo matemático representativo de la tarea:

Con el conocimiento previo de en qué consiste la tarea a realizar, se trata de representarla mediante una ecuación o conjunto de ecuaciones (ó expresiones lógicas) las cuales, cuando sea posible, se presentarán de manera explícita con respecto a la (s) variable (s) a calcular.

3.- Especificación del requerimiento de datos para resolver el modelo matemático:

Analizando el modelo matemático será posible identificar qué datos se requieren para realizar los cálculos involucrados en dicho modelo y obtener una respuesta al problema.

4.- Desarrollo del algoritmo:

Se trata de expresar mediante un diagrama de flujo o una lista de actividades (pseudocódigos) qué actividades realizar para desarrollar el trabajo correspondiente a la tarea. El nivel de desglose de actividades dependerá de la profundidad del análisis realizado ó de el posible agrupamiento que se efectúe de subconjuntos de actividades que representen subtareas reconocidas, dentro de la tarea global.

Al analizar una tarea el trabajo que represente podrá ser planeado ó descrito ya sea analizando toda la tarea en su conjunto o bien segmentándola en módulos, partes o porciones, a las cuales se les denomina subtareas y nos permite enfocar nuestros esfuerzos de análisis a cada porción en forma aislada con lo cual se facilita la programación; a este tipo de programación se le denomina **programación modular**.

Como se mencionó anteriormente un algoritmo es la descripción de qué actividades y en qué secuencia realizarlas para llevar a cabo una tarea, esto es, cómo resolver un problema. Es el método o procedimiento a utilizar para pasar del problema a la solución.

Al realizar el análisis de un número diverso de algoritmos con la finalidad de identificar actividades equivalentes entre ellos, se llegaría a concluir que tendríamos cinco diferentes tipos de actividades como máximo, las cuales corresponden a las actividades que puede realizar el procesador.

Una vez interpretada una tarea y descrito el algoritmo, dicho algoritmo podrá ser expresado mediante un diagrama o una lista de pseudocódigos.

Una lista de pseudocódigos utiliza verbos en modo imperativo para denotar acciones que se espera realice el procesador. Un diagrama de flujo mediante símbolos especiales denota cada acción.

Como método de representación del algoritmo aplicaremos el diagrama de flujo.

Conjuntando las actividades que puede realizar un procesador, un algoritmo es estructurado fundamentalmente con las actividades denominadas actividades básicas, de las cuales se hablará más adelante.

5.- Codificación:

Se trata de la representación del algoritmo (representación de cada actividad de la lista o diagrama) mediante códigos de un lenguaje de computadora (Fortran, Pascal, Basic, C, etc.).

Una vez que se ha aplicado la metodología de la programación, la cual concluye con la codificación, deberá evaluarse y reproducirse la tarea representada por el algoritmo.

La prueba se lleva a cabo activando la computadora para que realice la tarea suministrando a su memoria los códigos de las actividades correspondientes.

La tarea representada en códigos de un lenguaje se denomina programa fuente, el cual es suministrado a la computadora mediante un editor o procesador de palabras que ya puede venir integrado dentro del procesador de lenguaje, el cual puede ser de tipo compilador o de tipo interpretador.

En un sistema de tipo compilador el programa fuente es traducido en su totalidad por el procesador de lenguaje. *Ninguna actividad será activada en tanto la traducción no sea total.*

Cada código de el programa fuente expresado en el lenguaje de codificación que se este utilizando será traducido a otro código que es reconocido directamente por la computadora (códigos en lenguaje binario denominados lenguaje de máquina).

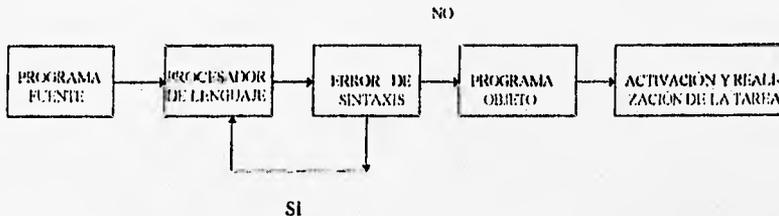
A esta nueva versión obtenida del programa fuente se denomina programa objeto, el cual será finalmente activado para que la computadora realice la serie de actividades efectuando de esta manera la tarea.

En todo este proceso existe probabilidad de introducir errores, por ejemplo los códigos pueden estar mal escritos o mal estructurados en lo referente a su forma de uso; en este caso el procesador de lenguaje no lo aceptará y reconocerá el error en el que se está incurriendo y no podrá crear el programa objeto en tanto no se corrija el error o errores. A estos errores se les denomina errores de sintaxis.

Un procesador de lenguaje contiene una tabla de códigos y las reglas de cómo utilizarlas (reglas de sintaxis) A este conjunto de códigos y sus reglas de sintaxis es a lo que se le denomina lenguaje de programación.

Cada código del programa fuente es filtrado en esta tabla y si no detecta errores lo va traduciendo a lenguaje binario. Si en alguno de los códigos hay error nos lo informa a través de un mensaje de error o código de error.

El programa objeto será producido hasta que ningún código fuente tenga error.



No se debe perder de vista que si no se encuentran errores de sintaxis la tarea se llevará a cabo, pero que el resultado puede no ser el esperado debido a otro tipo de error llamado lógico, el cual es resultado de un mal análisis del problema. De aquí que se hace necesario insistir en que no es posible resolver adecuadamente un problema si no se le ha comprendido completamente.

PROCESADORES DE LENGUAJE

Un procesador de lenguaje es a su vez un programa cuyo propósito es leer los códigos escritos en el lenguaje correspondiente y traducirlos a lenguaje binario obteniendo de esta manera códigos que son directamente interpretados por la computadora.

El lenguaje en el que son escritos los códigos originales, se denomina lenguaje de alto nivel y tal es el caso de Fortran, Pascal, C, etc.

El lenguaje binario se denomina lenguaje de bajo nivel ó lenguaje de máquina.

Un procesador de lenguaje sirve para transcribir códigos de alto nivel a códigos de máquina existiendo un procesador para cada tipo de lenguaje.



CAPÍTULO
II

CLASIFICACIÓN DE LOS PROCESADORES DE LENGUAJE POR SU FORMA DE OPERACIÓN

1.- Sistema compilador:

El sistema compilador traduce todos los códigos del programa fuente creando una imagen en binario que se denomina **programa objeto**; la tarea se realiza con este programa traducido. No se creará el programa objeto si algún código del programa fuente no cumple con su regla de sintaxis.

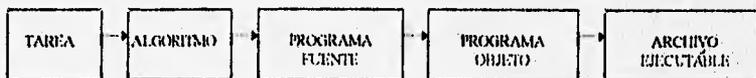
2.- Sistema intérprete:

El sistema intérprete trabaja de la siguiente manera: conforme cada código es traducido es ejecutado de manera tal que aún cuando no conozca si todos los códigos que integran el programa están correctos la tarea se va realizando parcial y simultáneamente.

ARCHIVO EJECUTABLE

A partir del programa objeto se puede obtener, mediante el ensamblado de módulos, un **archivo ejecutable**, el cual será independiente del **procesador de lenguaje** y será activado directamente desde el ambiente de sistema operativo logrando de esta manera producir programas que pueden ser comercialmente distribuidos para su uso.

Las distintas etapas para elaborar un programa que finalmente lleve a cabo una tarea se muestra en el siguiente diagrama (descripción asumiendo que el sistema es compilador):



Del análisis exhaustivo de la tarea se llega a proponer un algoritmo, el cual será la descripción del cómo efectuar dicha tarea.

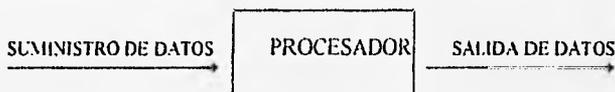
El algoritmo será la base para crear el **programa fuente**, el cual corresponde a la misma descripción de la tarea, pero con los códigos del lenguaje de programación seleccionado.

El **programa fuente** es compilado (es leído por el procesador de lenguaje) para producir el **programa objeto** siempre y cuando no detecte errores de sintaxis.

Cuando la tarea se realiza a través del programa objeto a este evento se le denomina proceso y a la máquina o ser humano que lo lleva a cabo se le denomina procesador.

RELACIÓN PROCESADOR - ALGORITMO

Durante el desarrollo del proceso los datos son transformados mediante operaciones aritméticas o de manejo de texto. El procesador está constituido por circuitos electrónicos diseñados para realizar este tipo de operaciones además de evaluar expresiones lógicas. Se dice que el procesador está constituido por una unidad lógica, una aritmética, y una unidad de control de flujo de información para que interactúen tanto con memoria RAM como con otros dispositivos externos. De tal manera que el procesador se puede asumir como una caja negra a la cual se le suministran datos, los transforma y emite resultados.



ACTIVIDADES QUE SE ASUMEN DEL PROCESADOR

1.- Cálculo de fórmulas o expresiones algebraicas :

Provee la manera de realizar cálculos aritméticos .

2.- Evaluación de expresiones lógicas o de relación de orden :

Permite seleccionar qué actividad realizar cuando se cumple alguna condición del proceso.

3.- Control de flujo de proceso :

Controla que cada actividad se lleve a cabo en el orden preestablecido: Hace posible el modificar la secuencia de cálculo para elegir o repetir actividades .

Como actividades de interacción con el procesador requerimos que exista una manera de suministrarle datos para que les aplique las actividades antes mencionadas y de igual forma , requerimos una manera de que el procesador nos muestre los resultados de la transformación.

Lo anterior nos conduce a exigir de cualquier procesador la capacidad para realizar las siguientes cinco actividades:

CINCO ACTIVIDADES QUE SE REQUIERE REALICE UN PROCESADOR

1.- Aceptar la entrada de datos :

Consiste en suministrar los valores asignados a las variables. A esta actividad se le denomina entrada o lectura de datos.

2.- Mostrar valores contenidos en su memoria :

Consiste en ver el resultado o valor contenido en variables de proceso , cuyos valores son asignados como resultado de los cálculos internos. A esta actividad se le denomina salida o escritura de resultados .

3.- Evaluar expresiones algebraicas :

Consiste en evaluar fórmulas o estructuras algebraicas diversas. A esta actividad se le denomina asignación o reemplazo.

4.- Evaluar expresiones lógicas o de relación de orden :

Consiste en aplicar condiciones a procesos de cálculo para seleccionar cuál es la siguiente operación a realizar. A esta actividad se le denomina sí condicional.

5.- Modificar la secuencia de cálculo :

Todo proceso se realiza en una secuencia natural , lo cual significa que si escribimos una lista de actividades , primero se llevará a cabo la que está en la parte superior de la lista y así sucesivamente de manera descendente , de tal manera que la actividad que se suministra primero queda en la parte más alta de la memoria, lo que es equivalente a escribir las actividades en una lista descendente .

Consiste en modificar la secuencia natural de proceso y lograr que ciertas actividades no se realicen o que un conjunto de actividades se repita un número de veces preespecificado. A esta actividad se le denomina transferencia de control a procesador .



ACTIVIDADES BÁSICAS DE UN PROCESADOR

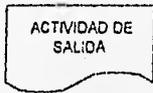
1.- Actividad de entrada, suministro de datos o lectura de datos.

Es la actividad mediante la cual suministraremos datos al proceso. Su símbolo para diagrama de flujo es:



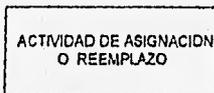
2.- Actividad de salida, escritura o despliegue de valores.

Esta actividad nos permite visualizar el contenido de las memorias asignadas a variables. Su símbolo para diagrama de flujo es:



3.- Actividad de asignación o reemplazo.

Mediante esta actividad asignamos valores a las variables, ya sea como resultado de un cálculo algebraico o como un valor directo especificado. Su símbolo para diagrama de flujo es:



4.- Actividad sí condicional.

Con esta actividad podemos elegir qué siguientes acciones realizar acordes con la secuencia de cálculo requerida. Esta actividad maneja las expresiones de relación de orden y las expresiones lógicas. Su símbolo para diagrama de flujo es:



5.- Actividad transferencia de control.

Todo proceso se lleva a cabo en una secuencia denominada natural, la cual se da de arriba hacia abajo considerando que las actividades están ordenadas en una lista de tal manera que primero se realizará la de la parte superior de la lista y continuará con las siguientes hacia abajo en el orden correspondiente. En un diagrama de flujo se visualiza la misma manera; de aquí surge la recomendación para que un diagrama de flujo se construya en forma vertical siempre que sea posible, para que podamos asociar el orden de ejecución con su nivel de posición en el diagrama, aunque esto no llega a ser un requisito indispensable ya que dentro del diagrama los símbolos se unen con flechas y las flechas especifican el orden de prioridad. Su símbolo en diagrama de flujo es :



En el siguiente capítulo (II) mediante ejemplos descriptivos aplicaremos las diferentes etapas de la programación concluyendo hasta el desarrollo de algoritmos, cuyo producto será el diagrama de flujo

CAPÍTULO
III

PROGRAMACIÓN. APLICACIÓN PARA LA REALIZACIÓN DE TAREAS DIVERSAS.

Generalidades :

Quando se desea realizar un trabajo el cual implica el producir algo que aporte un beneficio, lo primero será identificar cuál es el producto deseado es decir cuál es el objetivo u objetivos a alcanzar una vez que se haya llevado a cabo dicho trabajo. Basándonos en esto haremos las siguientes definiciones :

Tarea : Descripción del trabajo a realizar para el logro de uno o más objetivos.

Algoritmo : Representación de la tarea mediante un conjunto de actividades (acciones) concretas que efectuadas en cierta secuencia permitirán que la tarea sea concluida.

Programa Fuente : Lista de actividades a realizar expresada mediante códigos de un lenguaje de programación.

Lenguaje de Programación : Conjunto de códigos y reglas que para su uso, hacen posible activar una computadora para que realice acciones concretas

Proceso : Ejecución de un algoritmo. Si un proceso se está llevando a cabo implica que la tarea en ese instante se está realizando. El equipo o ser humano que lleve a cabo un proceso se denomina Procesador.

Programación : Planeación de las actividades y la secuencia en que cada una deberá realizarse para que lograr llevar a cabo una tarea.

La función de un analista de procesos consiste en analizar tareas para programar las acciones a realizar. En este capítulo se aplicarán las primeras cuatro etapas de la programación:

- 1.-Tarea.- Análisis de la tarea.
- 2.-Modelo matemático.- Representación de la tarea mediante ecuaciones matemáticas.
- 3.- Requerimiento de datos.- Qué datos requiere el modelo para su solución.

4.- Desarrollo de algoritmos - Aplicaremos el método de representación mediante diagramas de flujo.

Los problemas presentados los clasificaremos de acuerdo con los tres tipos de procesos posibles.

a) Secuenciales: Durante su desarrollo no se modifica la secuencia natural de ejecución, es decir, todas las actividades están ordenadas de acuerdo con su posición dentro de la lista, la actividad inicial será la de la parte superior de la lista, ésta se ejecutará primero, y continuará con la actividad inferior siguiente hasta llegar a la última en una secuencia operacional de arriba hacia abajo; secuencia que se denomina secuencia natural de proceso. Un proceso secuencial se efectúa de acuerdo con la secuencia natural sin modificación alguna.

Esquema de un proceso secuencial :

Actividad1;
Actividad2;
Actividad3;
Actividad4;
Actividad5;

Actividad_última;

El proceso se inicia con Actividad1, sigue este orden hasta llegar a Actividad_última.

Ejemplo : Calcular la suma de tres números .

Lista de actividades

- 1.- Suministro o entrada de los valores A,B y C
- 2.- Calcular $D = A+B-C$
- 3.- Escribir resultado, valor de D.

El proceso inicia con la actividad uno, la cual avanzará en el momento en que se suministren los tres valores.

Continuará con la actividad 2 realizando la suma y asignando esta suma a la variable D.

Finalmente escribirá el valor de D.

Esto es secuencia natural de proceso, pero existen muchos procesos que, como éste, no modifican su secuencia de cálculo.

b) De elección: Presenta opciones de actividades a realizar.

Por lo general se disponen de varios conjuntos de actividades de los cuales deberá seleccionarse y llevarse a cabo uno sólo. Para la selección se aplican criterios o condiciones las cuales hacen posible elegir un conjunto de acciones e ignorar a restantes.

Si un proceso incluye varios conjuntos de actividades seleccionables, implica que posiblemente uno sólo de ellos esté acorde con la secuencia natural de proceso y los restantes involucrarán modificación de secuencia, o posiblemente todos estén fuera de secuencia.

Esquema de un proceso de elección:

Actividad1;

Actividad2;

.

.

Condición de Proceso : Si se cumple realiza Conjunto1
Si no se cumple realiza Conjunto2

.

.

Actividad_última;

Una condición puede enlazar a otra condición y de esta manera incluir varios conjuntos de actividades. Esto hace posible el definir Ramificación de Procesos como una actividad complementaria.

Ejemplo : Se disponen de dos ecuaciones que representan el comportamiento de Y con respecto a X.

$$\begin{array}{ll} X \leq 5 & Y = 2X + 10 \\ X > 5 & Y = 15 \end{array}$$

La condición $X > 5$ permitirá elegir entre las expresiones a utilizar para evaluar Y.

e) De repetición: Procesos que involucren repetición de un conjunto de actividades secuenciales. Esta repetición implica modificar la secuencia de cálculo para que el control de proceso retorne a la primera actividad del conjunto.

Esquema de un proceso de repetición :

Actividad1;

Actividad2;

Proceso de repetición

Conjunto1;

Actividad_última;

Conjunto1 : Podrá estar integrado por una o más actividades.

Análisis de Tareas :

A continuación, se desarrollarán problemas sencillos y diversos, aplicando la metodología de programación y el proceso secuencial paso por paso:

PROBLEMA No.1 "Realizar la suma de dos números"

Aplicamos la metodología de la programación:

1.- Análisis de la tarea:

El objetivo claro y primordial que tenemos que alcanzar es lograr la suma de dos números cualesquiera que éstos sean.

2.- Modelo matemático:

De el álgebra elemental sabemos que : $a + b = c$, dicho modelo será aplicable siempre y cuando no se nos indique que estamos realizando sumas entre números que no pertenecen al conjunto de los números reales.

3.- Requerimiento de datos:

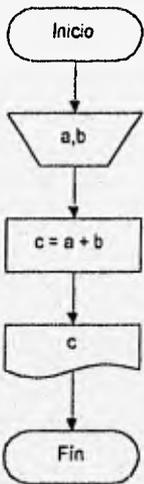
Para satisfacer adecuadamente el modelo matemático propuesto en el punto anterior, se hace necesario conocer por lo menos 2 números al mismo tiempo, los cuales bien pueden ser a y b.

4.- Desarrollo de algoritmo:

La manera más simple de expresar un algoritmo es el diagrama de flujo, pero para llegar a él, primero que nada se realizará el planteamiento de cómo y en qué orden se van a ejecutar los pasos necesarios a seguir para llevarla a cabo, los cuales serán:

- 1.- Es necesario suministrar los valores de a y b respectivamente.
- 2.- Una vez suministrados los valores de a y b, le asignamos la suma de a y b a la variable c y en ese momento será posible satisfacer el modelo matemático, y por ende conocer el valor de la variable c.
- 3.- Como el valor de c se ha generado, el paso siguiente es conocer cuál es ese valor, esto significa desplegarlo.

5.-Diagrama de flujo.



PROBLEMA No.2 "Realizar la multiplicación de tres números"

Aplicamos la metodología de la programación:

1.- Análisis de la tarea:

El objetivo claro y primordial que tenemos que alcanzar es lograr la multiplicación de tres números cualesquiera que éstos sean.

2.- Modelo matemático:

De el álgebra elemental sabemos que : $a \times b \times c = d$, dicho modelo será aplicable siempre y cuando no se nos indique que estamos realizando multiplicaciones entre números que no pertenecen al conjunto de los números reales.

3.- Requerimiento de datos:

Para satisfacer adecuadamente el modelo matemático propuesto en el punto anterior, se hace necesario conocer por lo menos 3 números al mismo tiempo, los cuales bien pueden ser a , b y c.

4.- Desarrollo de algoritmo:

La manera más simple de expresar un algoritmo es el diagrama de flujo, pero para llegar a él, primero que nada se realizará el planteamiento de cómo y en qué orden se van a ejecutar los pasos necesarios a seguir para llevarla a cabo, los cuales serán:

1.- Suministrar los valores de a, b y c respectivamente.

2.- Una vez suministrados los valores de a , b y c le asignamos a la variable d la multiplicación de a,b y c, por lo cual será posible satisfacer el modelo matemático, y por ende conocer el valor de la variable d.

3.- Como el valor de d se ha generado, el paso siguiente es conocer cuál es ese valor, esto significa desplegarlo.

5.- Diagrama de flujo



PROBLEMA No.3 "Realizar el cálculo del volumen de un rectángulo"

Aplicamos la metodología de la programación:

1.- Análisis de la tarea:

El objetivo claro y primordial que tenemos que alcanzar es calcular el volumen de un rectángulo.

2.- Modelo matemático:

De la geometría elemental sabemos que : $V = l \times a \times e$.

Donde:

V: volumen ,
l: longitud,
a: ancho,
e: espesor.

3.- Requerimiento de datos:

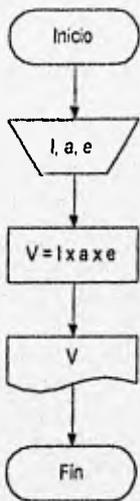
Para satisfacer adecuadamente el modelo matemático propuesto en el punto anterior, se hace necesario conocer por lo menos 3 variables al mismo tiempo, los cuales bien pueden ser l, a y e.

4.- Desarrollo de algoritmo:

La manera más simple de expresar un algoritmo es el diagrama de flujo, pero para llegar a él, primero que nada se realizará el planteamiento de cómo y en qué orden se van a ejecutar los pasos necesarios a seguir para llevarla a cabo, los cuales serán:

- 1.- Suministrar los valores de l , a y e respectivamente.
- 2.- Una vez suministrados los valores de l , a y e se realizará la asignación a la variable V de la multiplicación de l , a y e , siendo así posible satisfacer el modelo matemático, y por ende conocer el valor de la variable V .
- 3.- Como el valor de V se ha generado, el paso siguiente es conocer cuál es ese valor, esto significa desplegarlo.

5.- Diagrama de flujo



PROBLEMA No.4 "Realizar el cálculo de un área simple"

Aplicamos la metodología de la programación:

1.- Análisis de la tarea:

El objetivo claro y primordial que tenemos que alcanzar es calcular un área, la cual puede ser de un triángulo.

2.- Modelo matemático:

De la geometría elemental sabemos que: $At = \frac{B \times h}{2}$

Donde:

At: área de un triángulo ,

B: base ,

h: altura,

3.- Requerimiento de datos:

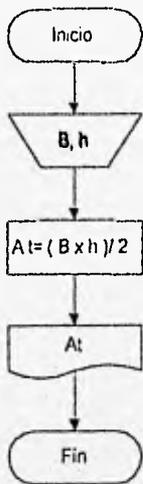
Para satisfacer adecuadamente el modelo matemático propuesto en el punto anterior, se hace necesario conocer por lo menos 2 variables al mismo tiempo, los cuales bien pueden ser B y h.

4.- Desarrollo de algoritmo:

La manera más simple de expresar un algoritmo es el diagrama de flujo, pero para llegar a él, primero que nada se realizará el planteamiento de cómo y en qué orden se van a ejecutar los pasos necesarios a seguir para llevarla a cabo, los cuales serán:

- 1.- Suministrar los valores de B y h respectivamente.
- 2.- Una vez suministrados los valores de B , y h será posible satisfacer el modelo matemático al asignar a la variable At la multiplicación de las variables B y h dividida entre dos, y por ende conocer el valor de la variable At.
- 3.- Como el valor de At se ha generado, el paso siguiente es conocer cuál es ese valor, esto significa desplegarlo.

5.- Diagrama de flujo.



PROBLEMA No.5 "Realizar el cálculo de un área simple"

Aplicamos la metodología de la programación:

1.- Análisis de la tarea:

El objetivo claro y primordial que tenemos que alcanzar es calcular un área, la cual puede ser de un círculo.

2.- Modelo matemático:

De la geometría elemental sabemos que : $A_c = \frac{\pi \times D^2}{4}$

Donde:

A_c: área de un círculo,

$\pi = 3.1416$,

D: diámetro.

3.- Requerimiento de datos:

Para satisfacer adecuadamente el modelo matemático propuesto en el punto anterior, se hace necesario conocer al menos el valor de 1 variable, la cual es D.

4.- Desarrollo de algoritmo:

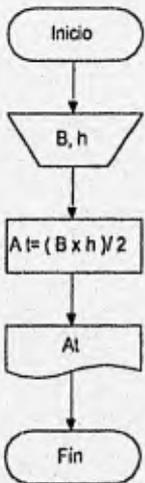
La manera más simple de expresar un algoritmo es el diagrama de flujo, pero para llegar a él, primero que nada se realizará el planteamiento de cómo y en qué orden se van a ejecutar los pasos necesarios a seguir para llevarla a cabo, los cuales serán:

1.- Suministrar el valor de D.

2.- Una vez suministrado el valor de D, será posible satisfacer el modelo matemático al realizar las asignaciones correspondientes, y por ende es posible conocer el valor de la variable Ac.

3.- Como el valor de Ac se ha generado, el paso siguiente es conocer cuál es ese valor, esto significa desplegarlo.

5.- Diagrama de flujo.



PROBLEMA No.6

"Realizar el cálculo de diversas propiedades termodinámicas"

Aplicamos la metodología de la programación:

1.- Análisis de la tarea:

El objetivo claro y primordial que tenemos que alcanzar es calcular diversas propiedades termodinámicas, las cuales pueden ser volumen, presión, temperatura y número de moles.

2.- Modelo matemático:

De la termodinámica elemental sabemos que $PV = nRT$, a partir de la cual sabemos que $P = \frac{nRT}{V}$, $V = \frac{nRT}{P}$, $T = \frac{PV}{nR}$, $n = \frac{PV}{RT}$

Donde:

P: Presión del sistema,

V: Volumen del sistema,

T: Temperatura del sistema,

n: Número de moles totales del sistema,

R: Constante universal de los gases ideales, la cual es igual a 0.082 atm-l/mol k

3.- Requerimiento de datos:

Para satisfacer adecuadamente el modelo matemático propuesto en el punto anterior, se hace necesario conocer al menos el valor de 3 variables para cada modelo matemático que se desee resolver, las cuales deben ser 3 de las siguientes 4: n, T, V, P.

4.- Desarrollo de algoritmo:

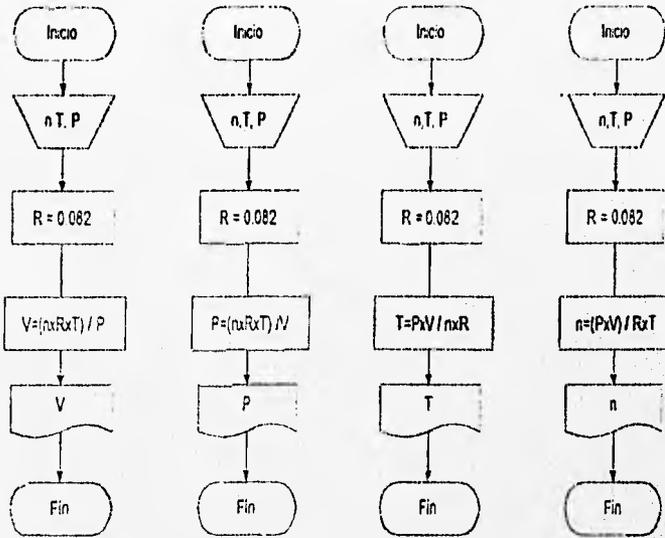
La manera más simple de expresar un algoritmo es el diagrama de flujo, pero para llegar a él, primero que nada se realizará el planteamiento de cómo y en qué orden se van a ejecutar los pasos necesarios a seguir para llevar a cabo las diversas tareas:

1.- Suministrar los valores de n, T, P y V respectivamente según el modelo que se desee resolver.

2.- Una vez suministrado el valor de las variables respectivas, así como las asignaciones pertinentes será posible satisfacer el modelo matemático, y por ende conocer el valor de la variable incógnita.

3.- Como el valor de la variable incógnita se ha generado, el paso siguiente es conocer cuál es ese valor, esto significa desplegarlo.

5.- Diagramas de flujo.



PROBLEMA No.7

“Realizar operaciones elementales con cantidades fraccionarias (quebrados).”

Aplicamos la metodología de la programación:

1.- Análisis de la tarea:

Se trata de realizar diversas operaciones con cantidades fraccionarias, conocidas como quebrados, para lo cual será necesario aplicar las reglas correspondientes a el tipo de operación aritmética que se llevará a cabo.

2.- Modelo matemático:

El modelo matemático tiene diversas representaciones de acuerdo al tipo de operación que se desee llevar a cabo. Dichos modelos de acuerdo con la aritmética son:

1.- SUMA:

$$\frac{a}{b} + \frac{c}{d} = \frac{ad + cb}{bd}$$

2.- RESTA:

$$\frac{a}{b} - \frac{c}{d} = \frac{ad - cb}{bd}$$

3.- MULTIPLICACIÓN:

$$\frac{a}{b} \times \frac{c}{d} = \frac{ac}{bd}$$

4.- DIVISIÓN:

$$\frac{a}{b} \div \frac{c}{d} = \frac{ad}{bc}$$

3.- Requerimiento de datos:

Para poder satisfacer el modelo matemático respectivo a cada operación, será necesario suministrar el valor de las fracciones sobre las cuales se desee operar. Las variables bien pueden ser a, b, c y d .

4.- Desarrollo de algoritmo:

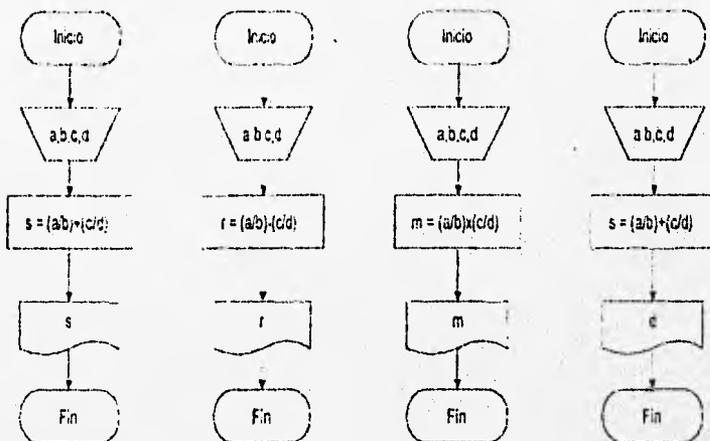
La manera más simple de expresar un algoritmo es el diagrama de flujo, pero para llegar a él, primero que nada se realizará el planteamiento de cómo y en qué orden se van a ejecutar los pasos necesarios a seguir para llevar a cabo las diversas tareas:

1.- Suministrar los valores de a, b, c y d respectivamente según el modelo que se desee resolver.

2.- Una vez suministrado el valor de las variables respectivas, y realizadas las asignaciones correspondientes, será posible satisfacer el modelo matemático, y por ende conocer el valor de la variable incógnita o resultado de la operación.

3.- Como el valor de la variable incógnita se ha generado, el paso siguiente es conocer cuál es ese valor, esto significa desplegarlo.

5.- Diagrama (s) de flujo



PROBLEMA No.8

“Resolver un sistema de dos ecuaciones algebraicas lineales, por sustitución”

Aplicamos la metodología de la programación:

1.- Análisis de la tarea:

Se trata de resolver dos ecuaciones algebraicas lineales, y de ese modo obtener los valores de aquellas incógnitas presentes en dichas ecuaciones.

2.- Modelo matemático:

El modelo matemático tiene diversas representaciones de acuerdo con el sistema que se quiera resolver, pero de el álgebra lineal obtenemos un modelo general:

$$x + y = 0$$

$$2x - y = 4$$

3.- Requerimiento de datos:

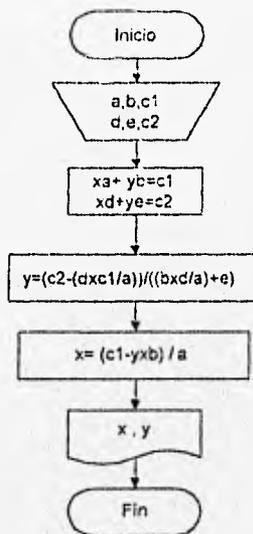
Para satisfacer adecuadamente el modelo matemático propuesto en el punto anterior, se hace necesario manipular las ecuaciones respectivas para acceder al método de resolución conocido como sustitución.

4.- Desarrollo de algoritmo:

La manera más simple de expresar un algoritmo es el diagrama de flujo, pero para llegar a él, primero que nada se realizará el planteamiento de cómo y en qué orden se van a ejecutar los pasos necesarios a seguir para llevar a cabo las diversas tareas:

- 1.- Despejar parcialmente alguna de las variable, por ejemplo x .
- 2.- Una vez generada la pseudoecuación respectiva, será posible satisfacer la siguiente ecuación matemática, y por ende conocer el valor de la variable incógnita
- 3.- Como el valor de la variable incógnita se ha generado, el paso siguiente es conocer cuál es ese valor, esto significa desplegarlo.

5.- Diagrama de flujo



Análisis de Tareas :

A continuación, se desarrollarán problemas sencillos y diversos, aplicando la metodología de programación y el proceso de elección paso por paso:

PROBLEMA No.1 "Resolución de la ecuación de segundo grado "

Aplicamos la metodología de la programación:

1.- Análisis de la tarea:

El objetivo claro y primordial que tenemos que alcanzar es resolver una ecuación de segundo grado, para lo cual tendremos que encontrar las dos raíces que satisfagan el sistema.

2.- Modelo matemático:

Del álgebra elemental sabemos que :

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Donde:

a,b,c son las constantes de la ecuación cuadrática.

3.- Requerimiento de datos:

Para satisfacer adecuadamente el modelo matemático propuesto en el punto anterior, se hace necesario conocer al menos el valor de las 3 constantes para el modelo matemático que se desee resolver.

4.- Desarrollo de algoritmo:

La manera más simple de expresar un algoritmo es el diagrama de flujo, pero para llegar a él, primero que nada se realizará el planteamiento de cómo y en qué orden se van a ejecutar los pasos necesarios a seguir para llevar a cabo las diversas tareas:

1.- Suministrar los valores de a , b y c para resolver el modelo .

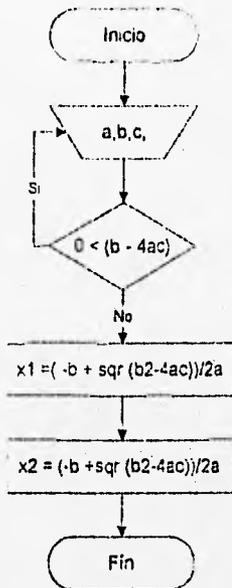
2.- Se realizará una prueba con estos valores para determinar si existen raíces imaginarias. ¿Es $(b^2 - 4ac) < 0$?

Si se cumple que $(b^2 - 4ac)$ es menor que cero, se desplegará un mensaje de que existe al menos una raíz imaginaria y cambiará la ejecución de la secuencia natural de proceso, ya sea para salir del programa o bien para regresar al inicio e intentarlo con otro sistema (ecuación).

3.- Si no existen raíces negativas, será posible satisfacer el modelo matemático, y por ende conocer el valor de la variable incógnita x o de las raíces, lo cual se llevará a cabo mediante la secuencia natural de proceso.

4.- Como el valor de la variable incógnita o raíces se han generado, el paso siguiente es conocer cuál es ese valor, esto significa desplegarlo.

5 - Diagrama de flujo



PROBLEMA No.2

“Evaluación del factor de fricción. Selección entre flujo laminar y flujo turbulento”

Aplicamos la metodología de la programación:

1.- Análisis de la tarea:

El objetivo claro y primordial que tenemos que alcanzar es evaluar el factor de fricción, para lo cual tendremos que empezar por seleccionar entre los dos tipos de flujo que se pueden presentar: laminar y turbulento.

2.- Modelo matemático:

En este problema en particular, no existe solo un modelo con sus respectivas ecuaciones, sino que existen al menos dos modelos y sus respectivas ecuaciones, para lo cual se hace necesario especificar cuales son las ecuaciones que corresponden al flujo laminar y cuales al flujo turbulento.

La forma más simple que tendremos de averiguar de qué tipo de flujo se está hablando es mediante el número dimensional de Reynolds, el cual dice que:

$$N_{Re} = \frac{D \times v \times \delta}{\mu}$$

Donde:

D : Diámetro interno
 v : Velocidad del fluido
 δ : Densidad
 μ : Viscosidad

Existen valores preestablecidos para determinar el régimen de flujo al que se está haciendo referencia, los cuales son:

Laminar $9 \times 10^3 < N_{Re} < 0 \quad f = \frac{64}{N_{Re}}$

Turbulento $N_{Re} > 1 \times 10^4$

3.- Requerimiento de datos:

Para satisfacer los modelos matemáticos que se plantean, es necesario realizar antes que nada la inspección respectiva al régimen de flujo en que se esté trabajando. Dicha inspección se puede realizar por medio de el número de Reynolds, para lo cual se requerirá conocer V, μ, δ, D . Una vez suministrados estos valores, se procede a satisfacer el modelo matemático (N_{Re}), con el cual se decidirá cuál es el modelo matemático que se tendrá que aplicar (factor de fricción) y por ende se resolverá un modelo matemático distinto (régimen laminar o turbulento).

4.- Desarrollo de algoritmo:

La manera más simple de expresar un algoritmo es el diagrama de flujo, pero para llegar a él, primero que nada se realizará el planteamiento de cómo y en qué orden se van a ejecutar los pasos necesarios a seguir para llevar a cabo las diversas tareas:

1.- Suministrar los valores de V, D, δ, μ y resolver para el Número de Reynolds (N_{Re}).

2.- Se realizará una prueba con este valor para determinar el régimen de flujo en el que se esta trabajando. ¿Es $9 \times 10^3 < N_{Re} < 0$?

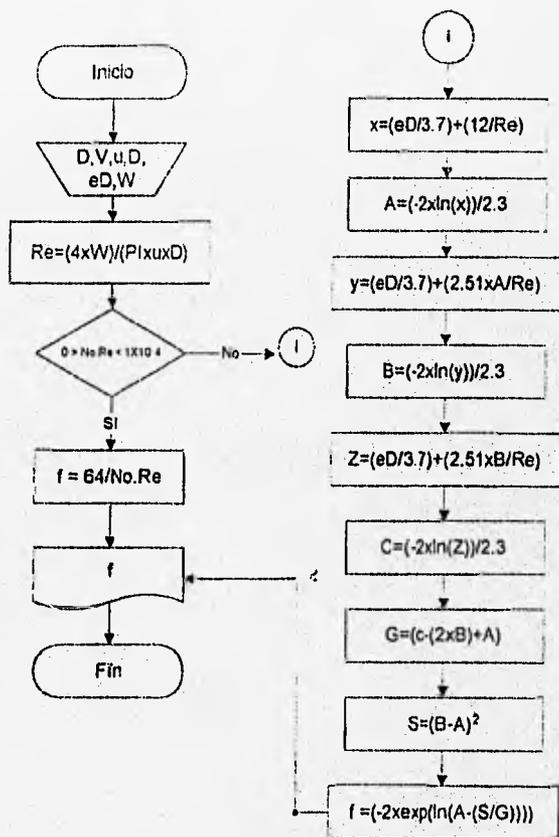
Si se cumple que N_{Re} es menor que 9×10^3 , se desplegará un mensaje que diga el régimen de flujo es laminar, la ejecución del programa se llevará a cabo en modo secuencial (secuencia natural de proceso). Si se cumple que N_{Re} es mayor o igual a

$\times 10^4$ la ejecución del programa cambiará de secuencia, para evaluar el factor de fricción para flujo turbulento, desplegando también un mensaje de que diga el régimen de flujo de es turbulento.

3.- Si no existe régimen turbulento, será posible satisfacer el modelo matemático, y por ende conocer el valor de la variable incógnita f , no sin antes conocer el valor del Re , lo cual se llevará a cabo mediante la secuencia natural de proceso.

4.- Como el valor de la variable incógnita f se han generado, el paso siguiente es conocer cuál es ese valor, esto significa desplegarlo.

5.- Diagrama de flujo.



Programación y su aplicación en la resolución de problemas diversos

Toda tarea o trabajo a realizar, por compleja que sea, puede ser representada mediante un algoritmo tal que máximo podrán ser identificadas 5 actividades diferentes; esto permite simplificar el desarrollo de algoritmos ya que el analista de procesos se ocupa de identificar dentro del conjunto de las 5 actividades bases, cuáles están presentes en su tarea.

El desarrollo de la Programación ha permitido que se describan completamente las 5 actividades a las cuales les denominaremos **actividades básicas**, con las cuales se podrá proponer el procedimiento para efectuar una tarea.

Tarea:

Desarrollo de algoritmo :

Identificación de qué actividades básicas y en qué secuencia se deberán realizar.

Presentación de este algoritmo mediante una Lista de Pseudocódigos o Diagrama de Flujo,

Esta última será la que aplicaremos en este trabajo por lo que es conveniente recordar la simbología asociada a cada actividad. (Capítulo I)

En la resolución de problemas es frecuente el uso combinado del **Si Condicional** y la **Transferencia de Control**, ya sea para representar procesos de repetición o procesos de elección múltiple, en este caso todos los Lenguajes de Programación permiten el uso de actividades que corresponden a esta combinación para diversos propósitos.

A esta combinación le denominaremos **Actividades Derivadas** y se reconocen dos :

1.- Circuitos de Cálculo.

2.- Ramificación de Procesos.

Dentro de los **Circuitos de Cálculo** la mayoría de los Lenguajes de Programación manejan tres tipos de Circuitos :

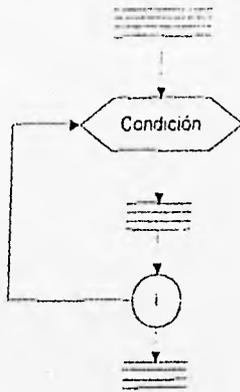
1.1 Circuito de Conteo.

1.2 Circuito de Convergencia con condición al inicio.

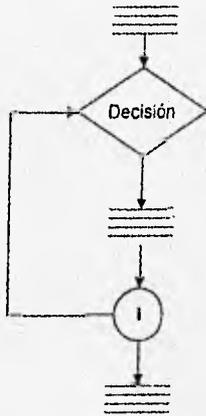
1.3 Circuito de Convergencia con condición al final.

La simbología respectiva de los tres tipos de circuitos es:

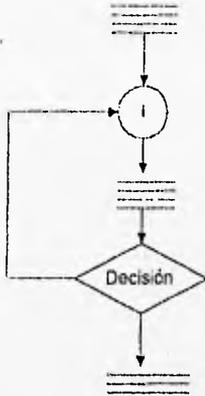
1.1.- Circuito de Conteo.



1.2.- Circuito de Convergencia con Condición al Inicio.



1.3.- Circuito de Convergencia con Condición al Final.



La Ramificación de Procesos consiste en la aplicación múltiple del Si Condicional para crear una estructura de selección múltiple. Existen muchas decisiones para las cuales se hallan muchas alternativas (más de dos)

Toda tarea podrá ser representada con las 5 actividades básicas o si así se desea, incluir actividades derivadas, recordando que en realidad son combinaciones de dos actividades básicas.

Tarea: Algoritmo representado por un diagrama de flujo.

3.1.- Aplicación de la Programación : Resolución de Problemas

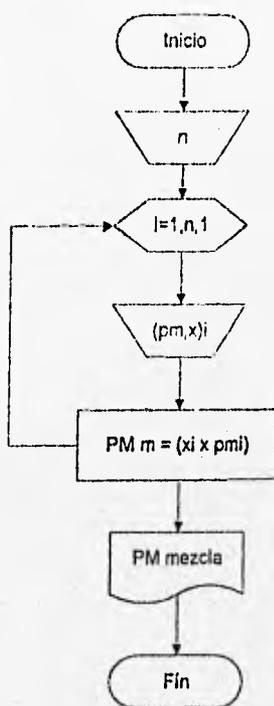
I.- Tarea # 1 : Calcular el Peso Molecular de la mezcla compuesta por 30% en mol de CO₂, 40 % en Mol de O₂ y 30% en mol de H₂.

II.- Modelo Matemático :

$$P M_{mezcla} = \sum_{i=1}^n (x_i \times p m_i)$$

III.- Requerimiento de Datos : Se requiere conocer el peso molecular de los diferentes compuestos, así como su fracción mol.

IV.- Desarrollo de Algoritmo : se representará mediante un diagrama de flujo



Comentarios sobre el algoritmo:

Debido a que no sabemos la cantidad de componentes que forman la mezcla, es necesario realizar un circuito de cálculo en función de n (número de componentes que forman la mezcla), para así poder evaluar el peso molecular de la mezcla.

V.- Codificación : Se traducirá el algoritmo a códigos del Lenguaje de Programación a utilizar, esto se hará en el siguiente capítulo(IV).

I.-Tarea # 2 : Calcular la presión absoluta y la presión de vacío de un sistema dado.

II.-Modelo matemático:

$$P_a = P_m + P_b$$

$$P_v = P_b - P_a$$

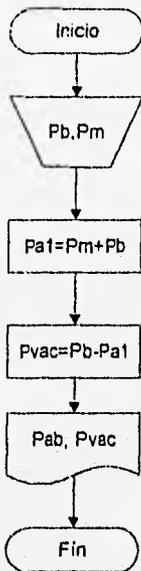
Donde:

P_a : Presión absoluta, P_v : Presión de vacío.

P_b : Presión manométrica,

III.- Requerimiento de datos: Es necesario suministrar el valor de la presión manométrica y de la presión absoluta.

IV.- Desarrollo del algoritmo: Se representará mediante el diagrama de flujo :



Comentarios sobre el algoritmo:

El algoritmo se desarrolla de manera secuencial, ya que no es necesario llevar a cabo cálculos alternos.

V.- Codificación : Se traducirá el algoritmo a códigos del Lenguaje de Programación a utilizar, esto se hará en el siguiente capítulo(IV).

I.- Tarea # 3 : Calcular el trabajo de compresión realizado por una turbina , así como la presión final. En dicho proceso no existe fricción.

II.- Modelo matemático:

$$C = P \times V^{1.3}$$

$$w = \int P_1 dV$$

$$w = \frac{-P_1 V_1}{0.3} \times \left(\frac{V_1}{V_2} \right)^{0.3} - 1$$

$$P_2 = P_1 \times \left(\frac{V_1}{V_2} \right)^{1.3}$$

Donde:

C: Constante de compresión,

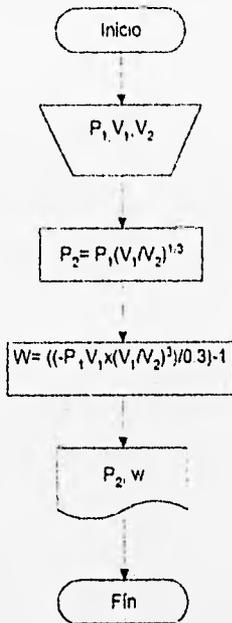
P_1 y P_2 : Presión inicial y final respectivamente,

V_1 y V_2 : Volúmenes inicial y final respectivamente,

w : Trabajo de compresión.

III.- Requerimiento de datos: Se requiere conocer el Volumen inicial y final, así como la Presión inicial.

IV.- Desarrollo del algoritmo: se representará mediante un diagrama de flujo:



Comentarios sobre el algoritmo:

El algoritmo se desarrolla de manera secuencial, ya que no es necesario llevar a cabo cálculos atemos.

V.- Codificación : Se traducirá el algoritmo a códigos del Lenguaje de Programación a utilizar, esto se hará en el siguiente capítulo(IV).

I.- Tarea # 4 : Calcular el trabajo de expansión que realiza un gas dentro de un globo.

II.- Modelo matemático:

$$V = \frac{\pi \times d^3}{6} \qquad C = \frac{P_1}{P_2}$$

$$w = \int_{V_1}^{V_2} P dV \qquad w = \frac{P_1 \times \pi \times d_1^3}{8} \times \left[\left(\frac{d_2}{d_1} \right)^4 - 1 \right]$$

Donde:

V: Volumen del globo,

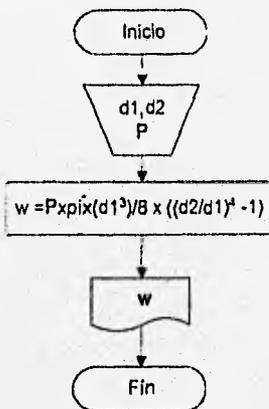
d_1 y d_2 : Son el diámetro inicial y final del globo respectivamente,

P: Presión del sistema

w: Trabajo de expansión

III.- Requerimiento de datos: Se requiere conocer d_1 , d_2 y la presión del sistema.

IV.- Desarrollo del algoritmo: se representará mediante un diagrama de flujo:



Comentarios sobre el algoritmo:

El algoritmo se desarrolla de manera secuencial, ya que no es necesario llevar a cabo cálculos alternos.

V.- Codificación : Se traducirá el algoritmo a códigos del Lenguaje de Programación a utilizar, esto se hará en el siguiente capítulo(IV).

I.- Tarea # 5 : Estimar la potencia necesaria para impulsar un compresor centrífugo. El proceso no tiene fricción .

II.- Modelo matemático.

$$C^* = p v^{k+1}$$

$$w = \frac{-1.4 P_1}{0.4 \times \delta_1} \left[\left(\frac{P_2}{P_1} \right)^{\frac{k}{k+1}} - 1 \right]$$

$$w = \int_{V_1}^{V_2} p dV \quad W = m \times w$$

Donde:

C: Constante de compresibilidad,

P_1 y P_2 : Presión inicial y final respectivamente,

δ : Densidad del fluido (aire),

w : Trabajo realizado por el sistema,

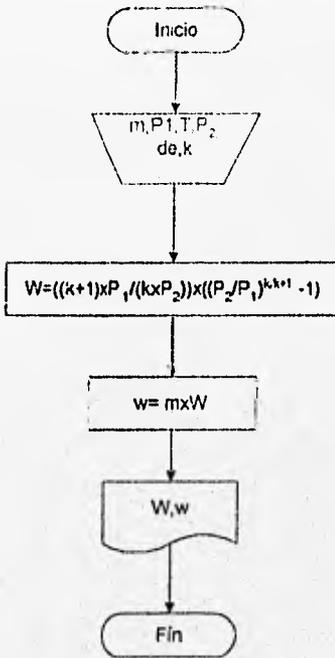
W : Potencia requerida por el sistema.

k : Relación de compresibilidad

III.- Requerimiento de datos:

Se requiere conocer la masa de aire a manejar, así como su densidad, las presiones de entrada y salida.

IV.- Desarrollo del algoritmo: se representará mediante un diagrama de flujo:



Comentarios sobre el algoritmo:

El algoritmo se desarrolla de manera secuencial, ya que no es necesario llevar a cabo cálculos alternos.

V.- Codificación : Se traducirá el algoritmo a códigos del Lenguaje de Programación a utilizar, esto se hará en el siguiente capítulo(IV).

I.-Tarea #6 : Calcular la velocidad de salida de aire que fluye por una tobera.

II.- Modelo matemático:

$$C = \rho v^{1.4}$$

$$V_2^2 = -2 \int v dV = -2 * \frac{1.4}{.4} P_1 V_1 \left[\left(\frac{P_2}{P_1} \right)^{\frac{.4}{1.4}} - 1 \right]$$

Donde:

V_1 : Velocidad inicial,

V_2 : Velocidad final,

v: Volumen,

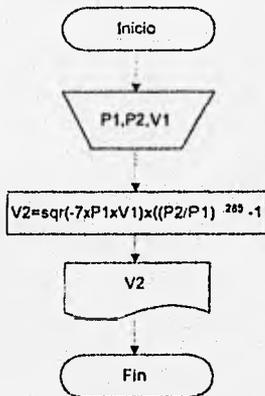
P_1 : Presión inicial,

P_2 : Presión final

III.- Requerimiento de datos:

Se requiere suministrar la presión inicial, final y la velocidad inicial.

IV.- Desarrollo del algoritmo : Se representará mediante un diagrama de flujo



Comentarios sobre el algoritmo:

El algoritmo se desarrolla de manera secuencial, ya que no es necesario llevar a cabo cálculos alternos.

V.- Codificación : Se traducirá el algoritmo a códigos del Lenguaje de Programación a utilizar, esto se hará en el siguiente capítulo(IV).

I.- Tarea #7: Estimar la potencia requerida para impulsar una bomba que conduce aceite de un nivel a otro.

II.- Modelo matemático :

$$\begin{aligned}
 Pa_1 &= Pb_1 - Pva_1 & G &= \delta \times Q \\
 Pa_2 &= Pa_1 + Pm_2 & w &= \frac{P_{a2} - P_{a1}}{\delta} - \frac{V_2^2 - V_1^2}{2} - g\Delta z \\
 V_1 &= \frac{Q}{S_1} & S &= \frac{\pi}{4} \times d^2 \\
 V_2 &= \frac{Q}{S_2} & W &= G \times w
 \end{aligned}$$

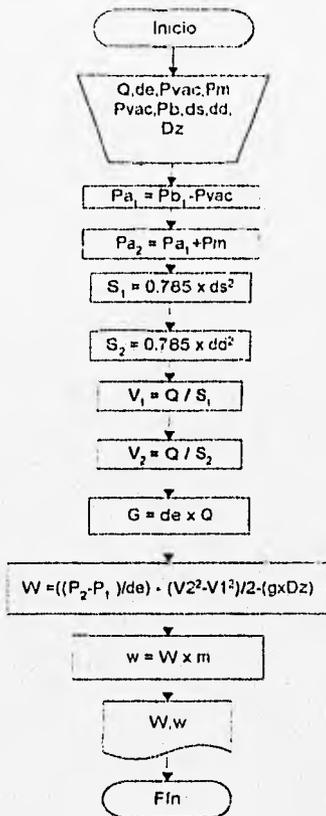
Donde:

Pa_1 y Pa_2 : Presiones absolutas inicial y final respectivamente,
 Pva_1 : Presión de vacío al inicio,
 Pm_2 : Presión manométrica,
 V_1 y V_2 : Son las velocidades inicial y final respectivamente,
 Q : Es el gasto volumétrico que maneja el sistema,
 S_1 y S_2 : Sección transversal a la succión y a la descarga respectivamente,
 Δz : Diferencia de alturas entre la succión y la descarga,
 g : Constante de gravitación universal,
 G : Gasto másico,
 δ : Densidad del fluido,
 w : Trabajo que se requiere realizar,
 W : Potencia que necesita el sistema.

III.- Requerimiento de datos:

Se requiere conocer el flujo a manejar, la densidad del fluido, la presión de vacío, la presión manométrica, la presión barométrica, el diámetro de succión y el diámetro de descarga, la diferencia de alturas entre la succión y la descarga.

IV.- Desarrollo del algoritmo: se representará mediante un diagrama de flujo:



Comentarios sobre el algoritmo:

El algoritmo se desarrolla de manera secuencial, ya que no es necesario llevar a cabo cálculos alternos.

V.- Codificación : Se traducirá el algoritmo a códigos del Lenguaje de Programación a utilizar, esto se hará en el siguiente capítulo(IV).

I.- Tarea # 8 : Determinar la razón de cambio con respecto al tiempo de la energía interna de un acumulador de carga eléctrica.

II.- Modelo matemático:

$$Q - W = \frac{dU}{dt} \quad W = -V * I$$

Donde :

Q: Calor disipado,

W: Trabajo realizado

V: Voltaje,

I: Intensidad de corriente,

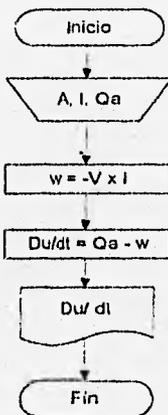
dU : Diferencial de energía interna,

dt : Diferencial de tiempo.

III.- Requerimiento de datos:

Se requiere conocer el voltaje, la intensidad de corriente y el calor disipado.

IV.- Desarrollo del algoritmo: Se representará mediante un diagrama de flujo.



Comentarios sobre el algoritmo:

El algoritmo se desarrolla de manera secuencial, ya que no es necesario llevar a cabo cálculos alternos.

V.- Codificación : Se traducirá el algoritmo a códigos del Lenguaje de Programación a utilizar, esto se hará en el siguiente capítulo(IV).

I.- Tarea # 9: Calcular la potencia que desarrolla un compresor que opera adiabáticamente.

II.- Modelo matemático :

$$w = h_1 - h_2$$
$$W = m \times w$$

Donde:

h_1 y h_2 : Entalpía del fluido a la succión y a la descarga respectivamente.

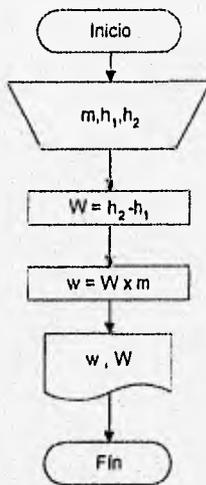
w: Trabajo realizado por el sistema,

W: Potencia realizada por el sistema.

III.- Requerimiento de datos:

Se requiere conocer la entalpía de succión y de descarga respectivamente, así como la masa de fluido a manejar.

IV.- Desarrollo de algoritmo: Se representará mediante un diagrama de flujo



Comentarios sobre el algoritmo:

El algoritmo se desarrolla de manera secuencial, ya que no es necesario llevar a cabo cálculos eternos.

V.- Codificación : Se traducirá el algoritmo a códigos del Lenguaje de Programación a utilizar, esto se hará en el siguiente capítulo(IV), en el cual se describe el Lenguaje Pascal.

I - Tarea # 10: Determinar la velocidad de descarga del aire que circula por una tobera convergente mediante un proceso adiabático.

II.-Modelo matemático:

$$C = pV^{1.4}$$

$$V_2 = 2(h_1 - h_2)$$

Donde:

h_1 : Entalpía a la entrada de la tobera,

h_2 : Entalpía a la salida de la tobera,

C : Constante de compresión,

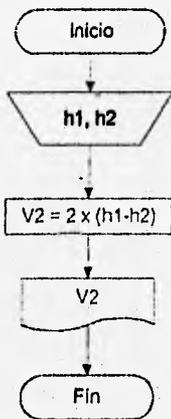
P : Presión,

V : Volumen

III.-Requerimiento de datos:

Se requiere suministrar los valores de h_1 y h_2 , que son la entalpía a la entrada y a la salida de la tobera respectivamente.

IV.- Desarrollo del algoritmo: Se representará mediante un diagrama de flujo.



Comentarios sobre el algoritmo:

El algoritmo se desarrolla de manera secuencial, ya que no es necesario llevar a cabo cálculos alternos.

V.- Codificación : Se traducirá el algoritmo a códigos del Lenguaje de Programación a utilizar, esto se hará en el siguiente capítulo(IV),

I.-Tarea # 11: Determinar para una turbina con una válvula de estrangulamiento:

- Cuál es la temperatura del vapor a la descarga de la válvula,
- El título del vapor a la descarga de la válvula si es que éste es húmedo.

II.- Modelo matemático:

$$q = \frac{Q}{m} \quad , \quad w = \frac{W}{m} \quad ,$$

$$\Delta ec = \frac{V_2^2 - V_1^2}{2} \quad , \quad \Delta ep = g \left(\frac{z_2 - z_1}{1000} \right)$$

$$\Delta h = q - w - \Delta ec - \Delta ep$$

$$h_2 = \Delta h - h_1 \quad , \quad h_3 = h_2 \quad , \quad V_{hum} = h_3 - \frac{hf_3}{hf_{g,3}}$$

Donde:

q : Calor por unidad de masa,

Q: Calor perdido,

w: Trabajo,

W: Potencia,

m : Masa de vapor,

V_1 y V_2 : Velocidades de succión y de descarga respectivamente,

Δec : Diferencial de energía cinética,

Δep : Diferencial de energía potencial,

V_{hum} : Vapor húmedo,

Δh : Balance de entalpías (energía),

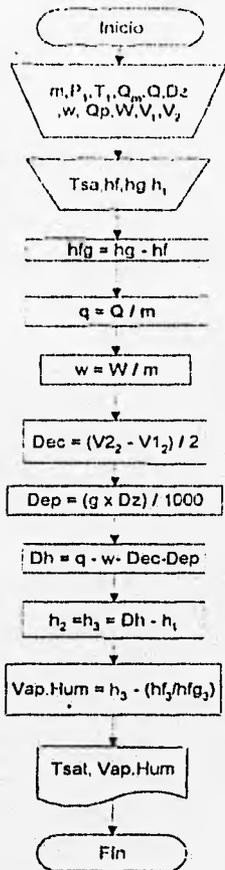
h_1, h_2, h_3 : Entalpía en la entrada, salida y válvula respectivamente.

Z_1, Z_2 : Diferencia de alturas entre la descarga y la succión.

III.- Requerimiento de datos:

Se requiere suministrar al modelo la masa de vapor que maneja la turbina, el flujo de calor perdido, velocidades de succión y de descarga, diferencia de alturas entre la succión y la descarga, la potencia que desarrolla la turbina, y, aunque si bien no es necesario, suministrar la temperatura de saturación, se hace necesario conocer el balance de energía para esa temperatura, (suministro de las entalpías correspondientes).

IV.- Desarrollo del algoritmo: Se representará mediante un diagrama de flujo.



Comentarios sobre el algoritmo:

El algoritmo se desarrolla de manera secuencial, ya que no es necesario llevar a cabo cálculos alternos.

V.- Codificación : Se traducirá el algoritmo a códigos del Lenguaje de Programación a utilizar, esto se hará en el siguiente capítulo(IV).

I.-Tarea # 12 : Calcular para una turbina de gas estacionaria, que trabaja isobáricamente el flujo de calor que debe suministrarse a la turbina, el gasto de gas necesario para suministrar el calor requerido, así como el flujo másico de combustible requerido.

II.- Modelo matemático:

$$v_1 = \frac{RT_1}{P_1} \quad \text{y} \quad v_2 = \frac{RT_2}{P_2}$$

$$m = \frac{A_1 V_1}{v_1} \quad V_2 = \frac{m v_2}{A_2} \quad \Delta h = Cp(T_2 - T_1)$$

$$Q = W + m \left[\Delta h + \frac{V_2^2 - V_1^2}{2} + g \Delta z \right] \quad C_{gas} = \frac{Q}{3900} \quad m_{gas} = \frac{C_{gas}}{v}$$

$$CE_{gas} = \frac{m_{gas}}{W} \quad \text{Relación} = \frac{m_{gas}}{m}$$

Donde:

A_1 y A_2 : Sección transversal de succión y de descarga respectivamente,

v_1 y v_2 : Son los volúmenes a la succión y a la descarga respectivamente,

V_1 y V_2 : Son las velocidades de succión y descarga respectivamente,

m : Es el flujo másico de aire,

Δh : Es la diferencia de entalpías entre la descarga y la succión,

Q : Es el flujo de calor que debe suministrarse a la turbina,

C_{gas} : Caudal de gas natural,

m_{gas} : Flujo másico de gas natural

CE : Consumo específico de gas,

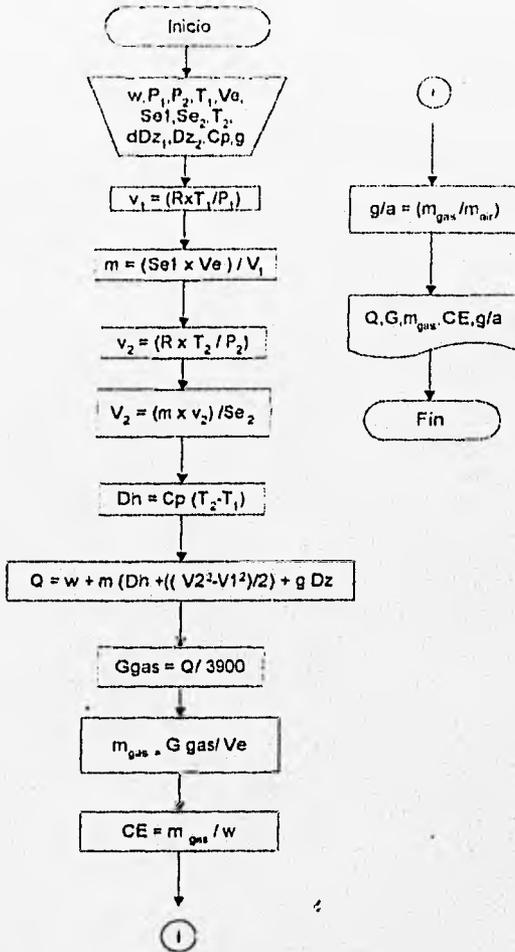
Relación: Relación que existe entre el combustible y el aire

III.- Requerimiento de datos:

Se requiere conocer la potencia, la presión y temperatura de la descarga y la succión,

la velocidad con la que está entrando el aire, la diferencia de alturas que existe entre la descarga y la succión, la sección transversal de succión y de descarga y la capacidad calorífica del aire.

IV.- Desarrollo del algoritmo : Se representará mediante un diagrama de flujo.



Comentarios sobre el algoritmo:

El algoritmo se desarrolla de manera secuencial, ya que no es necesario llevar a cabo cálculos alternos.

V.- Codificación : Se traducirá el algoritmo a códigos del Lenguaje de Programación a utilizar, esto se hará en el siguiente capítulo(IV).

I.- Tarea # 13: Determinar la temperatura de descarga del aire que sale de un compresor que opera adiabáticamente, así como la potencia necesaria para impulsar el compresor.

II.- Modelo matemático :

$$T_2 = T_1 \left(\frac{P_2}{P_1} \right)^{\frac{k-1}{k}} \quad -w = Cp(T_2 - T_1)$$

$$v = \frac{RT}{P} \quad m = \frac{G}{v} \quad - = \times$$

Donde :

T_1 y T_2 : Temperaturas de succión y descarga respectivamente,

P_1 y P_2 : Presión de succión y descarga respectivamente

v : Volumen del aire,

w : Trabajo realizado,

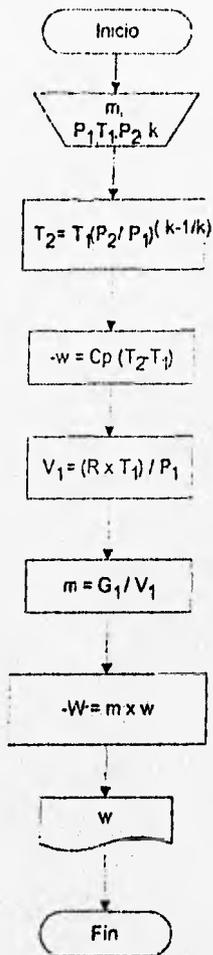
W : Potencia requerida,

m : Masa de aire

III.- Requerimiento de datos:

Se requiere conocer el gasto de aire que maneja el compresor, así como las temperaturas y presiones iniciales y finales, así como el valor de k (para un proceso adiabático).

IV.- Desarrollo del algoritmo : Se representará mediante un diagrama de flujo.



Comentarios sobre el algoritmo:

El algoritmo se desarrolla de manera secuencial, ya que no es necesario llevar a cabo cálculos atemos.

V.- Codificación : Se traducirá el algoritmo a códigos del Lenguajes de Programación a utilizar, esto se hará en el siguiente capítulo(IV).

I.- Tarea # 14 : Evaluar el coeficiente de fugacidad para el gas metano.

II.- Modelo matemático :

$$\ln \frac{f}{P} = \int_p^P \frac{C-1}{P} dP$$

Donde:

f: fugacidad,

P : Presión del sistema,

C: factor de compresibilidad,

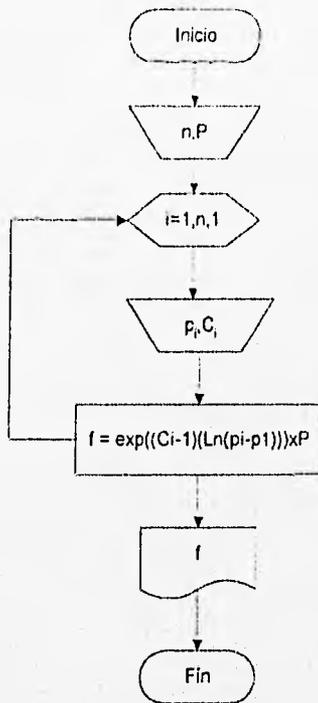
p: intervalo de presión a la cual se evaluará la fugacidad.

P₁ : presión de referencia (1 atm).

II.- Requerimiento de datos :

Se requiere conocer la presión del sistema, y el factor de compresibilidad para el rango respectivo de presión, así como el incremento de presión .

IV.- Desarrollo del algoritmo : Se representará mediante un diagrama de flujo.



Comentarios sobre el algoritmo :

Debido a que el cálculo de la fugacidad se llevará a cabo dentro de un rango de presiones, ha sido necesario resolver la tarea mediante un circuito de conteo.

V.- Codificación : Se traducirá el algoritmo a códigos del Lenguajes de Programación a utilizar, esto se hará en el siguiente capítulo(IV).

I.- Tarea # 15 : Determinar el volúmen específico del vapor de agua a una presión y temperatura dadas, haciendo uso de la ecuación de van der Waals.

II.- Modelo matemático :

$$v^3 - \left(b + \frac{RT_c}{P_c} \right) v^2 + \frac{a}{P_c} v - \frac{ab}{P_c} = 0$$

Donde :

v : Es el volumen específico,

v_c : Es el volumen específico que se tiene que estimar,

R : Constante universal de los gases,

a y b : Constantes del gas,

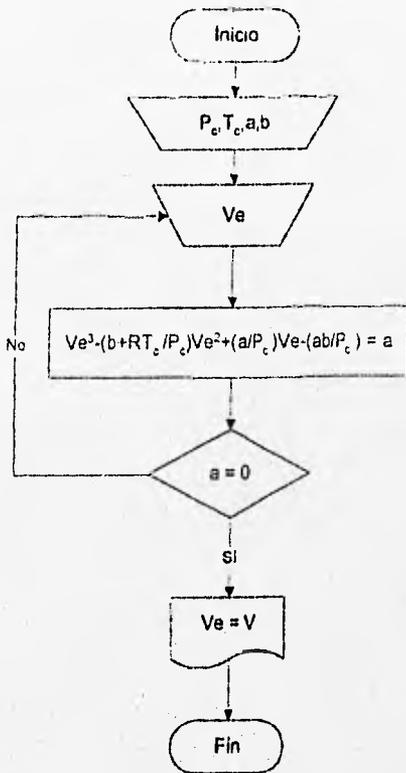
T_c : Temperatura crítica,

P_c : Presión crítica

III.- Requerimiento de datos:

Se requiere conocer un volúmen específico como estimado inicial, la presión y temperatura críticas, así como los valores de las constantes a, b y R.

IV.- Desarrollo del algoritmo : Se representará mediante un diagrama de flujo.



Comentarios sobre el algoritmo :

Para resolver esta tarea, ha sido necesario recurrir a un circuito de cálculo, que tiene la particularidad de tener condición al final, esto es, el circuito de cálculo no se rompe hasta que a no sea aproximadamente cero.

De esa manera se cumplirá que el V_e , será igual al volumen molar real.

V.- Codificación : Se traducirá el algoritmo a códigos del Lenguajes de Programación a utilizar, esto se hará en el siguiente capítulo(IV), en el cual se describe el Lenguaje Pascal.

I.- Tarea # 16 : Calcular la temperatura de flama adiabática teórica del CO al quemarse a presión constante con 100% de exceso de aire cuando los reactivos entran a 200 °F.

II.- Modelo matemático:



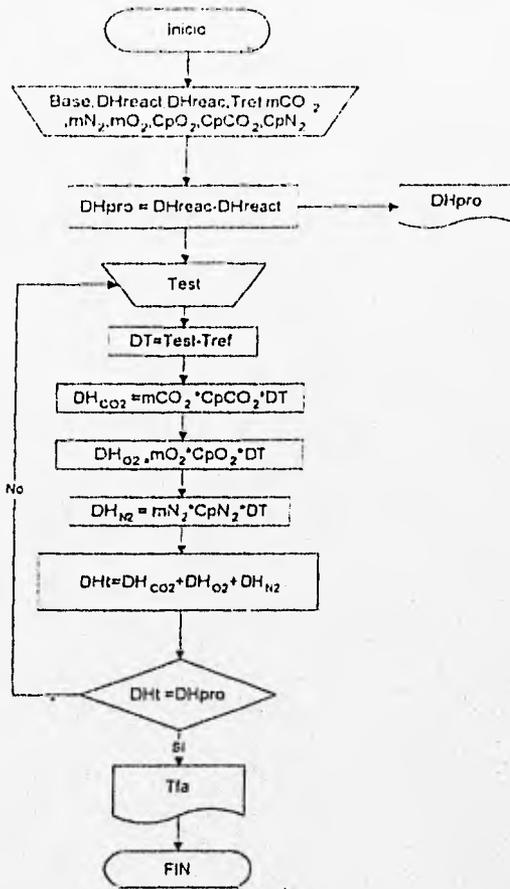
H : Entalpia, ya sea de productos o de reactivos,
 S_R : Calor de reacción,
 C_p : Capacidad calorífica,
 $\frac{dT}{dx}$: Gradiente de temperatura.

Entran		Balance de materia:	
Componente	moles	Componente	Salen moles
CO _(g)	1.00	CO _(g)	1.00
O _{2 (req)}	0.50	O _{2(g)}	0.50
O _{2 (exc)}	0.50	N _{2(g)}	3.76
O _{2 (tot)}	1.00		
N ₂	3.76		
Aire	4.76		
Donde:			

III.- Requerimiento de datos:

Se requiere conocer la base de cálculo o masa con la que se trabaja, esto es, el balance de materia, las capacidades caloríficas de los productos y de los reactivos, así como su gradiente de temperatura. También es necesario conocer el S_R de reacción.

V.- Desarrollo del algoritmo : Se representará mediante un diagrama de flujo.



Comentarios sobre el algoritmo :

Para poder realizar el cálculo de la temperatura de flama adiabática, ha sido necesario recurrir a un circuito de convergencia, para el cual existe una condición al final, si no se cumple que Dht sea igual a DHpro, entonces el circuito de cálculo no se rompe. En este caso, se escogió darle dos salidas, con la finalidad de que el usuario conozca el valor de DHpro, y pueda decidir qué valor de Test va a utilizar.

V.- Codificación : Se traducirá el algoritmo a códigos del Lenguajes de Programación a utilizar, esto se hará en el siguiente capítulo(IV).

CAPÍTULO
IV

EJEMPLO # 1

```
program pemopro;
uses crt;
var n,i:byte;
    pm,x,pmm,pmt: array [1..10] of real;
    o:char;
begin
clrscr;
gotoxy(10,10);write('Este programa te permite calcular el peso molecular ');
gotoxy(20,12);write('promedio de una mezcla gaseosa');
gotoxy(18,15);write('Cu ntos componentes tiene la mezcla ');read(n);
for i:=1 to n do
begin
gotoxy(15,17);Write('Suministra el peso molecular del componente[',i,'] : ');read(pm[i]);
gotoxy(15,19);Write('Suministra la fracción en mol del componente[',i,'] : ');read(x[i]);
clrscr;
pmm[i]:=pm[i]*x[i];
pmm[i]:=pmm[i]-pmm[i-1];
end;
clrscr;
gotoxy(15,15);writeln('El peso molecular promedio de la mezcla es ',pmm[i]:4:3);
O:=Readkey;
end.
```

EJEMPLO # 2

```
program dosnuno;
uses crt;
var Pb,Pm,Pa,Pv:real;
O:char;
begin
  clrscr;
  gotoxy(10,15);write('Suministra el valor de la presión barométrica ');
  read(Pb);
  gotoxy(10,20);write('Suministra el valor de la presión manométrica ');
  read(Pm);
  Pa:=Pm-Pb;
  Pv:=Pb-Pa;
  clrscr;
  gotoxy(15,15);writeln('La presión absoluta del sistema es : ',Pa:4:3);
  gotoxy(15,20);writeln('La presión de vacío del sistema es : ',Pv:4:3);
  O:=Readkey;
end.
```

EJEMPLO # 3

```
program tresuno;
uses crt;
var m,P,T,V1,V2,P2,P1,w,j,l:real;

O:char;
begin
clrscr;
gotoxy(10,10);write ('Suministra el valor el valor de la masa ');
read(m);
gotoxy(10,15);write ('Suministra el valor de la presión inicial ');
read(P1);
gotoxy(10,20);write ('Suministra el valor del volumen inicial ');
read(V1);
gotoxy(10,25);write ('Suministra el valor del volumen final ');
read(V2);
gotoxy(10,30);write ('Suministra el valor de la temperatura ');
read(T);


$$P2:=P1-(1.30*\ln(V1/V2));$$



$$w:=(-P1*V1/0.3)-(0.3*\ln(V1/V2));$$

clrscr;
gotoxy(15,15);writeln ('La presión final del sistema es : ',P2);
gotoxy(15,20);writeln ('El trabajo que realiza el sistema es : ',w:4:3);
O:=Readkey;
end.
```

EJEMPLO # 4

```
program trescuatro;
uses crt;
var d1,d2,P,a,w:real;
o:char;
begin
clrscr;
gotoxy(10,5);write('Este programa te permite calcular el trabajo de ');
gotoxy(10,6);write('expansión que realiza un gas dentro de un globo ');
gotoxy(20,7);write('debido al calentamiento');
gotoxy(10,10);write('Cuánto mide el diámetro inicial ? ');
read(d1);
gotoxy(10,15);write('Cuánto mide el diámetro final ? ');
read(d2);
gotoxy(10,20);write('Cuál es la presión del sistema ');
read(P);
w:=(P*P1*d1*d1*d1/8)*(((d2*d2*d2*d2)/(d1*d1*d1*d1))-1);
clrscr;
gotoxy(15,20);write('El trabajo que realiza el sistema es = ',w:3:3);
o:=readkey;
end.
```

EJEMPLO # 5

```
program trescis;
uses crt;
var
m,P1,P2,de,k,w,Wp,a: real;
O: char;
begin
clrscr;
gotoxy(10,5);write('Este programa te permite estimar la potencia ');
gotoxy(10,6);write('necesaria para impulsar un compresor centrifugo');
gotoxy(10,10);write('Suministra la masa de aire ');read(m);
gotoxy(10,12);write('Suministra la densidad a la temperatura de trabajo');read(de);
gotoxy(10,13);write('Suministra la presión inicial');read(P1);
gotoxy(10,14);write('Suministra la presión final');read(P2);
gotoxy(10,15);write('Suministra el factor de compresibilidad k ');read(k);
clrscr;
a:=k+1;
w:=-a*P1/(k*de))+exp((k/a)*ln(P2/P1-1));
Wp:=m*w;
gotoxy(10,10);write('El trabajo desarrollado por la turbina es = ',w:4:3);
gotoxy(10,12);write('La potencia necesaria para impulsar al compresor es = ',Wp:4:3);
O:=Readkey;
end.
```

EJEMPLO # 6

```
program tresiete;
uses crt;
var P2,P1,v,VV,Ve:real;
o:char;
begin
clrscr;
gotoxy(10,5);write('Este programa te permite calcular la velocidad final que');
gotoxy(10,6);write('alcanza el aire que fluye a través de una tobera');
gotoxy(10,8);write('Suministra la presión de entrada');read(P1);
gotoxy(10,10);write('Suministra la presión de salida');read(P2);
gotoxy(10,11);write('Suministra el volumen específico del aire');read(v);
VV:=((0.7*P1*v)+exp((0.286*ln(P2/P1)-1)));
Ve:=exp(0.5*ln(VV));
clrscr;
gotoxy(10,10);write('La velocidad final del aire en la tobera es ',Ve:3:3);
o:=Readkey;
clrscr;
end.
```

EJEMPLO # 7

```

program trescho;
uses crt;
var
Pa1,Pa2,Pb1,Pvac,Pm,Q,S1,S2,ds,dd,V1,V2,G,dc,gc,Dz,w,Wp:Real;
o:char;

begin
clrscr;
gotoxy(10,5);write('Este programa te permite estimar la potencia necesaria');
gotoxy(9,6);write('para impulsar una bomba que bombea un aceite combustible');
gotoxy(10,8);write('Suministra el flujo volumétrico que maneja la bomba ');
read(Q);
gotoxy(10,9);write('Suministra la densidad del fluido ');read(dc);
gotoxy(10,10);write('Suministra la presión de vacío en la succión ');read(Pvac);
gotoxy(10,11);write('Suministra la presión barométrica en la succión ');read(Pb1);
gotoxy(10,12);write('Suministra la presión manométrica en la descarga ');read(Pm);
gotoxy(10,13);write('Suministra el diámetro de succión ');read(ds);
gotoxy(10,14);write('Suministra el diámetro de descarga ');read(dd);
gotoxy(10,15);write('Suministra la diferencia de alturas entre la succión');read(Dz);
gotoxy(10,25);write('y la descarga ');
clrscr;
gc:=9.081;
Pa1:=Pb1-Pvac;
Pa2:=Pa1+Pm;
S1:=(0.785*ds*ds);S2:=(0.785*dd*dd);
V1:=Q/S1;V2:=Q/S2;
G:=dc*Q;
w:=(((Pa2-Pa1)/dc)-((V2*V2)-(V1*V1))/2-(gc*Dz));
wp:=G*w;
gotoxy(10,15);write(' El trabajo que realiza la bomba es = ',wp:4:3);
gotoxy(10,16);write(' La potencia que se requiere es = ',wp:4:3);
o:=Readkey;
end.

```

EJEMPLO # 8

```
program tresonce;
uses
crt;
var
A,I,Qa,w,V,DD:Real;
o:char;
begin
clrscr;
gotoxy(10,10);write('Este programa te permite calcular el cambio ');
gotoxy(10,11);write('de la energía interna en un acumulador de carga');
gotoxy(15,12);write('eléctrico con respecto al tiempo');
gotoxy(10,15);write('Suministra la intensidad de corriente ');read(I);
gotoxy(10,16);write('Suministra el voltaje ');read(V);
gotoxy(10,17);write('Suministra el calor disipado ');read(Qa);
clrscr;
w:=-V*I;
DD:=-Qa-w;
gotoxy(10,15);write('La razón de cambio de energía interna DU/Dt : ',DD:4:3);
o:=Readkey;
end.
```

EJEMPLO # 9

```
program trescatorce;
uses crt;
var
h1,h2,w,Wp,m:Real;
o:char;
begin
clrscr;
gotoxy(10,5);write('Este programa te permite calcular la potencia ');
gotoxy(10,6);write('requerida por una turbina que opera adiabáticamente ');
gotoxy(10,10);write('Suministra la masa de aire ');read(m);
gotoxy(10,12);write('Suministra el valor de la entalpia en la succión ');read(h1);
gotoxy(10,14);write('Suministra el valor de la entalpia en la descarga ');read(h2);
clrscr;
w:= h1-h2;
Wp:= m*w;
gotoxy(10,10);write(' El trabajo desarrollado por la turbina es = ',w:4:3);
gotoxy(10,12);write(' La potencia requerida por el sistema es = ',Wp:4:3);
o:=Readkey;
end.
```

EJEMPLO # 10

```
program tredienueve;
uses
crt;
var
h1,h2,V2,Vs:Real;
o:char;
begin
clrscr;
gotoxy(10,5);write('Este programa te permite calcular la velocidad del aire ');
gotoxy(10,6);write('a la descarga de una tobera convergente, donde se lleva ');
gotoxy(20,7);write('a cabo una expansión adiabática');

gotoxy(10,10);write('Suministra la entalpía del aire a la entrada de la tobera ');read(h1);
gotoxy(10,12);write('Suministra la entalpía del aire a la salida de la tobera ');read(h2);
V2:=2*(h1-h2);
Vs:=exp(0.5*ln(V2));
clrscr;
gotoxy(10,10);write('La velocidad a la descarga de la tobera es = ',Vs:4:3);
o:=Readkey;
clrscr;
```

EJEMPLO # 11

```

program cuatcuar;
uses
  crt;
const
  g=9.81;
var
  m,Q,Dz,w,Qp,Wp,V1,V2,Tsa,hf,hg,h1,hfg,h3,Dec,Dep,Vap,Dh:Real;
  o:Char;
begin
  clrscr;
  gotoxy(10,5);write('Este programa te permite calcular la temperatura del vapor ');
  gotoxy(10,6);write('que sale de una válvula de estrangulamiento y el título del vapor ');
  clrscr;
  gotoxy(10,10);write(' Suministra la masa de vapor ');read(m);
  gotoxy(10,11);write(' Suministra la presión a la entrada de la válvula ');
  gotoxy(10,12);write('Suministra la temperatura a la entrada de la válvula ');
  gotoxy(10,13);write('Suministra la entalpía a las condiciones de entrada ');read(h1);
  gotoxy(10,14);write('Suministra la velocidad de entrada a la válvula ');read(V1);
  gotoxy(10,15);write('Suministra la velocidad de salida a la válvula ');read(V2);
  gotoxy(10,16);write('Suministra la temperatura de saturación ');read(Tsa);
  gotoxy(10,17);write('Suministra la entalpía de líquido saturado a la salida ');read(hf);
  gotoxy(10,18);write('Suministra la entalpía de vapor saturado seco a la salida ');read(hg);
  gotoxy(10,19);write('Suministra el calor perdido ');read(Qp);
  gotoxy(10,20);write('Suministra la potencia que desarrolla la turbina ');read(Wp);
  gotoxy(10,21);write('Suministra la diferencia de alturas entre la salida y la entrada ');read(Dz);

  clrscr;
  hfg:=hg-hf;
  q:=Qp/m; w:=Wp/m;
  Dec:=((V2*V2)-(V1*V1));
  Dep:=(g*Dz)/1000;
  Dh:=q-w-Dec-Dep;
  h3:=Dh-h1;
  Vap:=(h3-(hf/hfg));
  if h3<hg then

  gotoxy(10,10);write('La temperatura de salida es igual a la temperatura de saturación ');
  gotoxy(10,11);write(' El título del vapor es ',Vap:4:3);
  o:=Readkey;
end.

```

EJEMPLO # 12

```

program cinceinc;
uses crt;
const R=0.287e3;g=9.8;
var
Wp,P1,T1,Ve1,Ve2,Se1,Se2,P2,T2,Dz1,Dz2,Cp,v1,v2,m,Dh,Q,Ggas,mgas,CE,ga :Real;
o:Char;
begin
clrscr;
gotoxy(10,10);write('Este programa te permite calcular el flujo de calor ');
gotoxy(10,11);write('que es necesario suministrar a una turbina de gas ');
gotoxy(10,12);write('estacionario que trabaja isobáricamente, así como el gasto');
gotoxy(10,13);write('de gas necesario y el combustible requerido');
gotoxy(10,14);write('Suministra la potencia que desarrolla la turbina ');read(Wp);
gotoxy(10,15);write('Suministra la presión de entrada ');read(P1);
gotoxy(10,16);write('Suministra la temperatura de entrada ');read(T1);
gotoxy(10,17);write('Suministra la velocidad de entrada ');read(Ve1);
gotoxy(10,18);write('Suministra la sección transversal de entrada ');read(Se1);
gotoxy(10,19);write('Suministra la diferencia de altura a la entrada ');read(Dz1);
gotoxy(10,20);write('Suministra la presión de salida ');read(P2);
gotoxy(10,21);write('Suministra la temperatura de salida ');read(T2);
gotoxy(10,22);write('Suministra la sección transversal de salida ');read(Se2);
gotoxy(10,23);write('Suministra la diferencia de altura a la salida ');read(Dz2);
gotoxy(10,24);write('Suministra la capacidad calorífica ');read(Cp);
clrscr;
v1:=(R*T1)/P1;
m:=(Se1*Ve1)/v1;
v2:=(R*T2)/P2;
Ve2:=(m*v2)/Se2;
Dh:=Cp*(T2-T1);
Q:=(Wp+m*Dh+((Ve2*Ve2*m)-(Ve1*Ve1*m))/2 -(g*Dz2-Dz1*g));
Ggas:=Q/3900;
mgas:=Ggas/Ve1;
CE:=mgas/Wp;
ga:=mgas/m;
gotoxy(10,10);write('El flujo de calor que es necesario suministrar es = ',Q:4:3);
gotoxy(10,12);write('El gasto de gas necesario para suministrar el calor es = ',mgas:4:3);
gotoxy(10,13);write('El consumo específico de gas es = ',CE:4:3);
gotoxy(10,14);write('La relación combustible aires es = ',ga:4:3);
O:=Readkey;
end.

```

EJEMPLO # 13

```
program cineseis;
uses crt;
const R=0.287e3;
var T1,T2,P1,P2,v,m,G,Cp,Wp,w:real;
o:char;
begin
clrscr;
gotoxy(10,10);write('Este programa te permite calcular la temperatura de salida');
gotoxy(10,11);write('del aire que pasa a través de una compresión adiabática');
gotoxy(10,12);write('as; como la potencia necesaria para impulsar el compresor');
gotoxy(10,15);write('Suministra la masa de aire a comprimir');read(m);
gotoxy(10,16);write('Suministra la presión inicial');read(P1);
gotoxy(10,17);write('Suministra la temperatura inicial');read(T1);
gotoxy(10,18);write('Suministra la presión final');read(P2);
gotoxy(10,19);write('Suministra la capacidad calorífica del aire');read(Cp);
T2:=T1-exp(0.2857*ln(P2/P1));
w:=-Cp*(T2-T1);
v:=R*T1/P1;
G:=(m/60*v);
Wp:=w*m;
clrscr;
gotoxy(10,10);write('La temperatura a la salida del compresor es = ',T2:5:3);
gotoxy(10,11);write('La potencia requerida para impulsar el compresor es = ',Wp:5:3);
o:=Readkey;
end.
```

EJEMPLO # 14

```
program integral;
uses crt;
const p1=1;
var n,i:Byte;
Ps,a:Real;
c,p,f,Fs: array [1..50] of real;
o:Char;
begin
clrscr;
gotoxy(10,10);write(' Este programa te permite calcular el coeficiente');
gotoxy(7,11);write('de fugacidad para el gas metano a diferentes presiones');
gotoxy(10,13);write('Suministra la presión del sistema ');read(Ps);
gotoxy(10,14);write('Con cuántos valores de presión vas a trabajar ? ');read(n);

for i:=1 to n do
begin
gotoxy(10,16);write('Suministra el valor de la presión[' ,i, ' ] : ');read(p[i]);
gotoxy(10,17);write('Suministra el valor del coeficiente de compresibilidad[' ,i, ' ] : ');read(c[i]);
clrscr;
f[i]:=exp((c[i]-1)*ln(p[i]-p1));
Fs[i]:=f[i]-f[i-1];
end;
a:=Ps*Fs[i];
gotoxy(10,18);write(' La fuga cida d del gas metano es = ',a:4:3);
o:=Readkey;
end.
```

EJEMPLO # 15

```
program van;
uses crt;
var Pc,Tc,a,b,c,Ve,R:Real;
o:char;
begin

write('Suministra el valor de la constante a ');read(a);
write('Suministra el valor de la constante b ');read(b);
write('Suministra el valor de la constante R ');read(R);
write('Suministra la presion critica ');read(Pc);
write('Suministra la temperatura critica ');read(Tc);
repeat
write('Suministra el volumen molar estimado ');read(Ve);
c:=(Ve*Ve*Ve)-((b*(R*Tc/Pc)*Ve*Ve)+(a/Pc)*Ve-(a*b/Pc));
until c<=0.001;
write('El volumen molar estimado es igual al volumen molar real ',Ve:4:3);
o:=Readkey;
end.
```

EJEMPLO # 16

```
program adiabatic;
uses crt;
var DHpro, DHreac, DHreact, Test, Dt, Tref, DHco2, mCo2, cpCO2,
    DHO2, mO2, cpO2, DHn2, mN2, cpN2, DHT, Tfa: Real;
o: char;
begin
  clrscr;
  write('Suministra el DH de reacción '); read(DHreac);
  write('Suministra el DH de reactivos '); read(DHreact);
  write('Suministra la temperatura de referencia '); read(Tref);
  write('Suministra la masa del CO2 '); read(mCO2);
  write('Suministra la capacidad calorífica del CO2 '); read(cpCO2);
  write('Suministra la masa del O2 '); read(mO2);
  write('Suministra la capacidad calorífica del O2 '); read(cpO2);
  write('Suministra la masa del N2 '); read(mN2);
  write('Suministra la capacidad calorífica del N2 '); read(cpN2);
  repeat
    write('Suministra la temperatura de flama adiabática estimada '); read(Test);
    DT:=Test-Tref;
    DHCO2:=mCO2*cpCO2*DT;
    DHO2:=mO2*cpO2*DT;
    DHN2:=mN2*cpN2*DT;
    DHT:=DHCO2-DHO2-DHN2;
  until DHT=DHpro;
  write('La temperatura de flama adiabática real es = ', Test:4:3);
  o:=Readkey;
end.
```

Conclusiones :

Después de realizar un análisis integral de la Programación y su aplicación en la resolución de problemas, llegamos a la conclusión de que la programación la aplicamos en la vida diaria de manera no sistemática y con frecuencia desordenada. Esta tesis promueve el aplicar la programación en cierta secuencia siguiendo las etapas del Método Científico.

Programar una tarea implica un conocimiento previo y amplio sobre el fenómeno o problema a resolver, lo cual hace que la programación sea un pilar o sostén importante dentro de los procesos de enseñanza y de aprendizaje.

Así, dentro del campo de la Ingeniería Química al enseñar alguna de las operaciones unitarias, por ejemplo Transferencia de Calor, implica un dominio completo del fenómeno, lo que permite establecer una serie de pasos para calcular o redimensionar equipos relacionados. Esto significa programar, de tal manera que mediante un algoritmo que exprese el comportamiento fenomenológico, se logre dimensionar o simular el medio en que se desarrollará la transferencia.

En lo referente al aprendizaje, el lograr que un alumno desarrolle su algoritmo de cálculo para un equipo lleva implícito el forzarlo a que adquiera un conocimiento previo de las Operaciones Unitarias en conjunción con la Termodinámica y Fenómenos de Transporte relacionado. El lograr describir un camino para cierto cálculo exige un conocimiento integral del cómo influyen el medio y sus alrededores en la evaluación de la propiedad que desee efectuarse.

Recomendaciones :

Para un Ingeniero Químico :

Programar \Leftrightarrow Conocimiento de la Naturaleza

Programar \Leftrightarrow Conocimiento de las Ciencias Básicas
(Física , Química , Matemática)

ESTRUCTURA DE LA TESIS
NO DEBE SALIR

BIBLIOGRAFIA

- Carnahan, Brice,**
"Applied numerical methods",
John Wiley & sons. Inc.,
U.S.A 1969.
- Castellan, Gilbert,W.,**
"Fisicoquímica",
Fondo Educativo Interamericano, S.A,
México, 1974.
- Grogono, Peter,**
"Programación en Pascal",
Ed.Addison-Wesley Iberoamericana,
México, 1987.
- Himmelblau, M,David,**
"Principios y cálculos básicos de
la ingeniería química",
Cia.Editorial Continental, S.A.,
México, 1976.
- Koffman, Elliot B.,**
"Pascal",
Introducción al lenguaje y
resolución de problemas con
programación estructurada.
Ed.Fondo Educativo Interamericano,
México, 1985.
- Levine, Gutiérrez Guillermo,**
"Introducción a la computación y
a la programación estructurada "
McGraw-Hill,
México, 1990.
- Manrique, José,**
"Termodinámica",
De. Harla, S.A. de C.V.,
México, 1976.
- Yester, Michael,**
"Using Turbo Pascal 6.0 "
2nd Edition,
U.S.A, 1993.