

308917²³
Reje.



UNIVERSIDAD PANAMERICANA

ESCUELA DE INGENIERIA
CON ESTUDIOS INCORPORADOS A LA
UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

PROGRAMA PARA RESOLVER CICLOS DE GENERACION
DE POTENCIA POR MEDIO DE VAPOR

T E S I S

QUE PARA OBTENER EL TITULO DE
INGENIERO MECANICO ELECTRICISTA
AREA: MECANICA ELECTRICA

PRESENTA:

EDMUNDO LOPEZ PEÑAFIEL

DIRECTOR: ING. ALFREDO GONZALEZ RUIZ

MEXICO, D. F., 1994

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 está 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.

DEDICATORIAS

A mis padres

. Por todo el cariño y el apoyo que me han dado todos estos años, sinceramente muchas gracias.

A mi hermana Moni

Por tantos momentos de distracción, te quiero.

A mis amigos

Jorge y Alejandro, siempre he podido contar con ustedes, gracias por su amistad.

A mi generación

Aprendimos tantas cosas juntos, jamás los olvidaré.

A mis profesores

. Por el rico cúmulo de experiencias que me brindarán estos 5 años

AGRADECIMIENTOS.

Ingeniero Alfredo González.

Ingeniero Eduardo Lémus.

Ingeniero Jorge A. Llaguno.

Mauricio I. Pérez López.

Sritas Irma y Margarita.

CONTENIDO

Dedicatorias	2
Contenido	3
Introducción	5
Capítulo 1 Antecedentes	
1.1. Historia de la máquina de vapor	7
Capítulo 2 Ciclos de Potencia	
2.1. Ciclos de Potencia	11
2.2. Reversibilidad	12
2.3. Ciclo de Carnot	13
2.4. Ciclo Rankine	14
2.5. Efecto de la presión y temperatura en la eficiencia del ciclo	16
2.6. Ciclo con recalentamiento	19
2.6.1. Eficiencia de Bomba	22
2.6.2. Eficiencia de Turbina	23
2.7. Ciclo Regenerativo	23
2.7.1 Calentador Abierto de Agua	27
2.7.2 Calentador Cerrado de Agua	28
2.8. Ciclo Combinado	31
Notas del Capítulo 2	34
Capítulo 3 Ejemplos de Ciclos	
3.1. Objetivo	35
3.2. Ciclo de Carnot con vapor	35
3.3. Ciclo Rankine	38
3.4. Ciclo con Recalentamiento	40
3.5. Ciclo Regenerativo	43
3.6. Ciclo con Recalentamiento y Regeneración	48
3.7. Ciclo con Recalentamiento, Regeneración y eficiencia de Turbina y Bomba	52
Capítulo 4 Modelos y Simulación	
4.1. Planteamiento del problema	57
4.2. Definición de modelo	57
4.3. Historia del desarrollo de la simulación	58

4.4. Ventajas y desventajas del uso de la computadora	60
4.5. Lenguajes de simulación	60
4.6. Software existente	62
Capítulo 5 Descripción del programa	
5.1. Comportamiento de los elementos que conforman el ciclo	65
5.1.1. Turbina	65
5.1.2. Condensador	66
5.1.3. Bomba	66
5.1.4. Calentador Abierto	66
5.1.5. Calentador Cerrado	67
5.1.6. Trampa	67
5.1.7. Caldera	68
5.2. Estructura general del programa .	68
5.3. Unidades auxiliares del programa	68
5.3.1. Archiv.TPU	69
5.3.2. Elemento.TPU	71
5.3.3. Sammy.TPU	72
5.4. Descripción del programa	74
5.4.1. Iniciar un nuevo ciclo	74
5.4.2. Cargar un reporte existente	80
5.4.3. Editar parámetros	81
5.4.4. Solución Numérica	81
5.4.5. Solución Gráfica	84
5.4.6. Dar Trabajos	85
5.4.7. Potencia de la Turbina	85
5.4.8. Salir	85
Notas del Capítulo 5	86
Conclusiones	87
Bibliografía	89
Anexo I (Instructivo)	90
Anexo II (Programa Principal)	99
Anexo III (Unidad Archiv.TPU)	158
Anexo IV (Unidad Elemento.TPU)	167
Anexo V (Unidad Sammy.TPU)	174

INTRODUCCION

INTRODUCCION.

Una de las áreas de mayor interés para toda ingeniería es la producción, transformación y aprovechamiento de la energía, dentro de ellas la más útil es la eléctrica, generada comercialmente a través de plantas de vapor. Existen otros tipos de plantas para la obtención de electricidad como las hidroeléctricas, que aprovechan alguna caída de agua, las geotérmicas, en donde la energía se manifiesta como fuentes de aguas termales en las cuales el agua excesivamente caliente se extrae del subsuelo y las nucleoeeléctricas, que operan con la energía nuclear producida por la fusión nuclear de ciertos isótopos. En nuestro país la mayor parte de energía se produce por medio de termoeléctricas.

Dichas plantas generadoras de potencia operan mediante la combustión de materiales fósiles como el carbón, pero todas siguiendo el mismo ciclo, claro, con algunas variaciones, pero con el mismo principio.

Mientras más sencillo es el ciclo, más fácil es de analizar, pero es menos eficiente. Una solución para aumentar la eficiencia es agregar varias etapas regenerativas lo cual, efectivamente da una ligera mejora en el aprovechamiento de los recursos, pero desgraciadamente se ha complicado demasiado el análisis termodinámico del ciclo. Se presentan básicamente dos problemas:

1. La solución numérica, ya sea con tablas o con un diagrama de Mollier toma mucho tiempo, los procesos de interpolación y búsqueda de datos son lentos, si se tienen que resolver varios casos estamos hablando de varias horas de trabajo.

2. Dada la gran cantidad de datos que se buscan y que hay que

calcular, hay una gran probabilidad de error, la cual aumenta al irse complicando el ciclo.

Lo expuesto anteriormente no es muy alentador, podemos pasar cerca de una hora resolviendo un ciclo y al final descubrir que hay un error por lo que, aunque no recalculamos todo el ciclo, si habrá necesidad de revisar varias partes.

El objetivo de esta tesis, es plantear una solución para dicho problema. Se propone desarrollar un programa para Computadoras Personales (PC's) que sea capaz de resolver ciclos de potencia; que nos presente los valores de sus propiedades (Presión, Temperatura, Entalpía específica, Entropía específica y Volumen específico); que nos proporcione una respuesta gráfica, un diagrama T-s resulta muy ilustrativo; finalmente, que nos dé el dato de la eficiencia térmica del ciclo, a fin de cuentas, es el más importante de todos. El programa tendrá, sobre todo, un fin didáctico.

Para realizar este trabajo es necesario conocer a fondo la naturaleza de los ciclos y un lenguaje de programación. El proyecto nos da así la oportunidad de aplicar parte de los conocimientos adquiridos en termodinámica y computación.

CAPITULO 1
ANTECEDENTES

CAPITULO 1
ANTECEDENTES

1.1. HISTORIA DE LA MAQUINA DE VAPOR.

La máquina de vapor fue el principal impulsor de la profunda convulsión social y tecnológica denominada Revolución Industrial. Contribuyó a transformar Inglaterra, el país de su nacimiento, que paso de una economía agrícola a otra industrial.

La primera referencia que se tiene data del año 100, donde Herón de Alejandría construyó en el siglo I la primera máquina que utilizaba el vapor para producir un movimiento rotatorio. Se trataba de un recipiente redondo donde se hervía agua; el vapor escapaba por dos tubos y hacía girar la caldera sobre un eje. Herón había inventado una forma de turbina de vapor pero la consideraba un simple juguete.

En 1660, el físico francés Denis Papin construyó un cilindro de bronce de 6 cm., de diámetro en el que vertió unas gotas de agua. Introdujo un émbolo por encima del agua y lo calentó hasta que hirvió el agua, lo que empujó el émbolo hasta arriba, donde quedó detenido por un gancho. Apartó el calor y el vapor se condensó, creando en el cilindro un vacío parcial. Cuando se soltó el gancho, la presión atmosférica hizo descender el émbolo con tal fuerza que pudo elevar casi 30 Kg., de peso.

En 1698 surge el primer dispositivo de vapor capaz de efectuar un trabajo; se trataba de una bomba que usaba 'la fuerza impelente del fuego' para achicar el agua de las minas, entonces muy propensas a inundaciones. La bomba se montaba en la mitad del pozo de la mina. El vapor de una caldera pasaba a un tanque hasta que lo llenaba; se cerraba la válvula del vapor y se rociaba agua para condensarlo. La presión atmosférica extraía el agua de la mina mediante una tubería y la elevaba hasta el tanque al vacío. Luego se abría nuevamente la válvula de vapor y

la presión de éste impulsaba al agua el resto del camino hasta la superficie.

Thomas Newcomen diseñó en 1712 la primera máquina de vapor aprovechable comercialmente, y en el decenio de 1760, cientos de máquinas Newcomen estaban en uso para extraer el agua de la minas. Su máquina consistía en un émbolo que estaba unido mediante una barra a una larga viga que pivotaba en el centro como un columpio. Cuando se llenaba de vapor el cilindro, el émbolo ascendía por el peso de la barra de la bomba que colgaba del otro extremo de la viga. Cuando el émbolo estaba en lo alto, se rociaba agua en el cilindro, lo que condensaba el vapor y creaba un vacío parcial. La presión atmosférica hacía bajar el émbolo en la carrera motriz y elevaba la barra de la bomba.

La válvula de vapor y la espita de agua estaban controladas por la propia máquina, capaz de 14 carreras por minuto; cada una elevaba 45 litros de agua. Esta máquina necesitaba únicamente vapor de baja presión, pues el esfuerzo verdadero lo realizaba la presión atmosférica, de ahí que a esta máquina suele denominarsele: 'máquina atmosférica'.

En 1777, el escocés James Watt comprendió que la máquina de Newcomen podría resultar mucho más eficiente si el cilindro no se enfriara y se calentara constantemente e inventó un condensador separado, que recibía y condensaba el vapor procedente del cilindro. Rodeó éste de un aislamiento para mantenerlo caliente y en vez de confiar en la presión atmosférica para hacerlo bajar, prefirió introducir vapor a baja presión por encima del émbolo. Se trataba pues de una verdadera máquina de vapor y consumía la tercera parte de carbón que la máquina de Newcomen. El francés Nicolás J. Cugnot, independientemente de Watt, construyó en París un pequeño vehículo movido a vapor en 1769,

pero vió que el vehículo no tenía suficiente energía y resultaba inestable.

Hacia 1782, Watt había logrado dos nuevas mejoras. Consiguió que la máquina fuera de doble acción, introduciendo alternativamente el vapor a uno y otro lado del émbolo, utilizando el vapor de forma expansiva; lo introducía en el cilindro durante la primera parte de la carrera, de forma que su expansión impulsará el émbolo. Watt inventó también los dispositivos mecánicos necesarios para imprimir movimiento giratorio a la máquina para que pudiera impulsar la maquinaria de la fábrica.

Ya en 1784, William Murdoch y James Watt, habían construido un pequeño modelo de locomotora para carretera, pero Watt no animó a Murdoch a seguir experimentando. Quedo para Richard Trevithick el construir tanto la primera locomotora de alta presión y doble tracción para carretera (1801) como la primera locomotora que se utilizó con éxito sobre carriles (1804).

Su primera máquina alcanzaba una velocidad de hasta 14 Km/h, velocidad excesiva para la época. La máquina de Trevithick funcionaban con una presión de 3.5 Kg/cm² y su autor prescindió de la pesada viga en favor del acoplamiento directo con un cigüeñal.

La segunda máquina se construyó para una factoría al sur de Gales; se demostró que una locomotora de vapor podía arrastrar mucho más que un caballo y a una velocidad de 8 Km/h.

George Stephenson construyó su primera máquina en 1814; su locomotora fue notable porque era la primera que rodaba sobre el canto de un carril de zapata, y no sobre planchas con rebordes. Stephenson continuó trabajando en el desarrollo y perfeccionamiento de la locomotora; finalmente en 1825 se inauguró el primer ferrocarril público de vapor que discurría a una velocidad de 20 a 25 Km/h.

La máquina de vapor tuvo también gran aplicación en la industria naviera. El marqués Jouffroy d'Abbans construyó el primer barco de vapor práctico en Francia en 1783; la máquina hacía girar unas ruedas de paleta.

En la última mitad del siglo XIX se consiguieron los últimos grandes avances en el diseño de la máquinas de vapor. Dos de ellos permitieron un uso más económico y eficaz del vapor y del combustible. El sistema de recalentamiento, inventado hacia 1860, permite que el vapor de las calderas pierda humedad al elevar su temperatura antes de que actúe sobre los émbolos, con lo que aumenta su eficiencia. En las máquinas de expansión múltiple, también inventadas hacia 1860, se usaba el vapor varias veces, primero en cilindros pequeños a presión elevada y luego en cilindros más grandes a presiones más bajas. En las máquinas de triple y cuádruple expansión el vapor se usaba 3 ó 4 veces.

La innovación más espectacular fue la turbina de vapor en 1883. Su inventor, sir Charles Parsons, matemático e ingeniero inglés, la exhibió ante los monarcas de Europa, los cuales equiparon a sus armadas con dichas turbinas

La máquina de vapor sigue desempeñando un papel importante en los modernos países industriales. La mayoría de la electricidad que se consume en forma de energía y luz está producida por maquinaria movida por el vapor en las centrales eléctricas de origen térmico.

CAPITULO 2
CICLOS DE POTENCIA

2.1. CICLOS DE POTENCIA.

Un ciclo se define como una serie de procesos termodinámicos durante los cuales el fluido de operación puede someterse a cambios que comprendan transiciones de energía y a continuación retornar a su estado original. El objetivo de cualquier ciclo práctico es convertir energía de una forma a otra más útil.

Los ciclos de potencia se clasifican a menudo por el carácter del fluido de trabajo que interviene en el ciclo. Las dos grandes clases de ciclos son el de vapor y el de gas. El primero difiere del segundo en dos aspectos. En el segundo, no hay ningún cambio de fase de la sustancia de trabajo y el trabajo de compresión de los ciclos de gas representan un gran porcentaje de la salida de trabajo útil del mismo. En el ciclo de vapor, la sustancia de trabajo se condensa hasta la fase líquida a la temperatura inferior del ciclo. El líquido se bombea a la presión deseada de salida; como el líquido es, en esencia, incompresible, cabe esperar que el trabajo de bombeo del ciclo de vapor represente un pequeño porcentaje de la salida de trabajo útil del ciclo.

Ejemplos de ciclo de gas son el ciclo Otto, el Diesel y el dual, utilizados en máquinas de combustión interna, mientras que de ciclos de vapor tenemos el Rankine con sus variaciones, utilizados para la generación de energía eléctrica.

En la práctica los ciclos reales difieren de los ideales debido a su irreversibilidad y a otras razones de índole práctica, más el estudio de los ciclos ideales da resultados muy útiles aplicados a los ciclos reales.

2.2. REVERSIBILIDAD.

Antes de empezar a desarrollar la teoría de ciclos termodinámicos, se debe tener claro el concepto de reversibilidad. Se define como un proceso reversible a aquél que se lleva a cabo de manera tal que un sistema y todos sus alrededores puedan regresarse a sus estados iniciales efectuando el proceso inverso.

El proceso reversible es una idealización es decir, algo ficticio. Es un concepto que algunas veces se puede aproximar por los dispositivos actuales pero que nunca se iguala. Por esta razón, en lo sucesivo la palabra ideal aparecerá continuamente.

Cuando un proceso es tal que ni el sistema ni sus alrededores puedan ser devueltos a sus estados iniciales, se dice que el proceso es irreversible. Hasta ahora se ha visto que la irreversibilidad se origina de dos fuentes:

1. Por los efectos de disipación inherentes que resultan de la naturaleza misma de la sustancia. Estos efectos incluyen la fricción de cualquier tipo.
2. Por ausencia de equilibrio mecánico, térmico o químico durante un proceso, es decir, por procesos que no son cuasiestáticos.

La presencia de cualquiera de estos efectos hace que el proceso no sea ideal. Puesto que todos los procesos reales incluyen tales efectos, el proceso reversible es un proceso límite al que tratan de acercarse, en rendimiento, todos los ciclos reales, pero nunca se consiguen realizarlos. Sin embargo, los procesos reversibles son un punto de partida adecuado en el que se basan los cálculos de diseños en ingeniería.

2.3. CICLO DE CARNOT.

Carnot propuso un ciclo reversible compuesto por dos procesos isotérmicos reversibles y dos procesos adiabáticos e isoentrópicos reversibles. Si el fluido de trabajo aparece en las fases líquida y vapor durante distintas partes del ciclo, los diagramas T-S del ciclo serán como se muestran en la figura 1; junto se muestra un esquema del equipo necesario para tal ciclo.

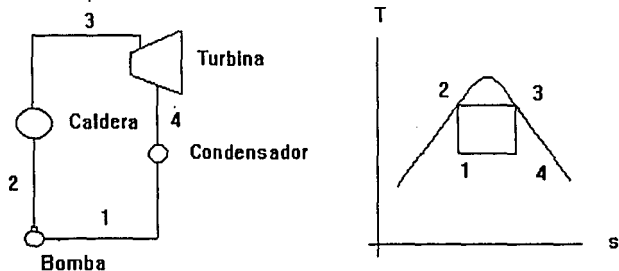


Figura 2.1. Esquema y diagrama T-s del ciclo de Carnot

Para esta tesis supondremos que el fluido de trabajo es agua. En el estado 1 se comprime isoentrópicamente vapor húmedo hasta el líquido saturado del estado 2. A esta presión se suministra energía (calor) de un medio infinito a presión constante hasta que el agua se evapora completamente al vapor saturado del estado 3. Luego se permite que la energía recibida produzca trabajo mediante la expansión adiabática e isoentrópicamente del vapor en una turbina hasta el estado 4. Entonces el vapor húmedo que sale de la turbina se condensa parcialmente a presión constante hasta regresar al estado 1.

El ciclo de Carnot es reversible. Su importancia reside en la siguiente conclusión: Ninguna máquina que opere entre dos temperaturas fijas, una fuente (T_1) y un sumidero (T_2) y que entregue trabajo de manera continua puede ser más eficiente que una máquina reversible que opere entre estos dos límites de temperatura.

Para calcular la eficiencia de un ciclo de Carnot se utiliza la siguiente ecuación (1)

$$h = 1 - (T_2/T_1)$$

Donde T_2 y T_1 deben estar en escalas absolutas

Cabe hacer notar que según esto, la eficiencia depende sólo de las temperaturas, independientemente del fluido que se utilice.

2.4. CICLO RANKINE.

El más sencillo de los ciclos de cambio de fase en uso es el ciclo Rankine. El agua, en forma de líquido saturado en el estado 1 de la figura, se comprime isoentrópicamente hasta la presión de entrada de la caldera, estado 2, por medio de una bomba. El líquido se calienta a presión constante hasta alcanzar el estado de saturación 2'. La adición de energía a presión constante continua a lo largo de 2'-3 con la producción de vapor seco en 3, el que luego experimenta una expansión isoentrópica en una turbina hasta el estado 4 (una porción muy pequeña del trabajo generado por la turbina se utiliza para mover la bomba que comprime el líquido del estado 1 al 2. Después de la expansión, el vapor se condensa con rechazo de energía a presión constante y regresa al estado 1.

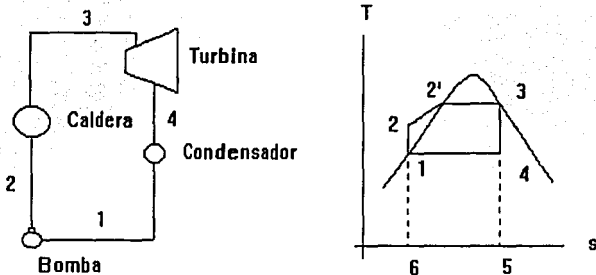


Figura 2.2. Esquema y diagrama T-s de ciclo Rankine sencillo.

De la figura, la entrada de calor está dada por el área 6-2-2'-3-5, mientras que la entrada de trabajo de la turbina es el área 1-2-2'-3-4.

Cabe hacer notar dos aspectos:

1) Que en el ciclo ideal se supone que el líquido suministrado a la bomba está saturado a la presión más baja del ciclo. En un ciclo real, el líquido por lo regular está ligeramente subenfriado para evitar la formación de burbujas de vapor en la bomba, lo que ocasiona un proceso conocido como cavitación, que acarrea daños a la bomba.

2) La transferencia de energía en la fuente (caldera) no ocurre a una temperatura constante, sino a temperaturas que varían entre 2 y 2'; por tanto es de esperar que la eficiencia del ciclo Rankine sea menor que la de un ciclo de Carnot que opere entre los mismos límites altos y bajos de temperatura.

Las transferencias de energía comprendidas en las diversas partes del ciclo Rankine son:

1. Trabajo para mover la bomba

$$w_b = h_2 - h_1$$

Sin embargo, el trabajo de la bomba también puede calcularse a partir de la ecuación relativa al trabajo mecánico de régimen permanente

$$w = \int v dP$$

Dado que el agua es un fluido incompresible, el trabajo de bomba puede determinarse con precisión deseada por la expresión

$$w_b = V_f(P_2 - P_1)$$

Donde V_f es el volumen específico del líquido saturado en el estado 1.

2. La transferencia de calor en la caldera

$$q_c = h_3 - h_2$$

3. El trabajo producido por la turbina

$$w_t = h_3 - h_4$$

En este caso se suponen despreciables las pérdidas por fricción, así como las diferencias de energía potencial y cinética a la entrada y salida de la turbina.

4. La transferencia de calor en la condensación

$$q = h_4 - h_1$$

Finalmente:

$$\eta = (w_t - w_b) / q_c$$

2.5. EFECTOS DE LA PRESION Y TEMPERATURA EN LA EFICIENCIA DEL CICLO.

Consideremos el diagrama T-S mostrado en la figura 3

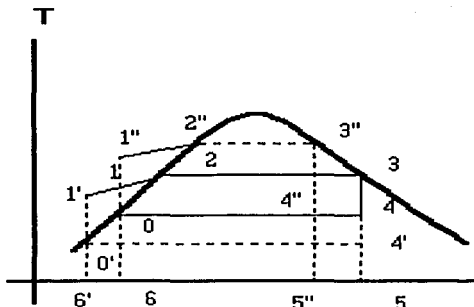


Figura 2.3. Efecto de la presión en la eficiencia del ciclo Rankine.

En esta figura 3, el ciclo 0-1-2-3-4 tiene una presión de escape p_0 que es más alta que la del ciclo 0''-1''-2''-3''-4''. El descenso de la presión de escape ha aumentado la entrada de energía térmica en el área 1-6-6'-1', mientras que el trabajo neto producido ha aumentado en la cantidad 4-4'-0'-0. Por tanto, la entrada de energía como calor es menor que el trabajo extra producido debido a la disminución de presión, esto causa un aumento de eficiencia.

Sin embargo, se puede ver que el vapor después de la expansión hasta el estado 4' tiene más humedad de la que tenía en el estado 4. El flujo de esta humedad por sobre los álabes de la turbina reduce la eficiencia interna de la turbina y puede también erosionarla rápidamente. Para una buena eficiencia y larga vida de la turbina, la humedad del vapor debe estar por debajo de 12%.

Un aumento en la presión de la caldera sube la eficiencia hasta cierto límite. Esto se debe a que a una más alta presión de la caldera

resulta en una más alta temperatura de saturación, a la que se suministra toda la energía térmica necesaria para vaporizar el líquido.

Dentro de la misma figura 3, el ciclo 0-1"-2"-3"-4" representa condiciones a una más alta presión de la caldera que la del ciclo 0-1-2-3-4, a la presión del condensador. El ciclo a más alta presión tiene mayor humedad después de la expansión que el ciclo a baja presión. Por tanto puede que no sea tolerable un aumento en la presión de la caldera sin un dispositivo para limitar la humedad al final de la expansión; un método para disminuir la humedad es dando un sobrecalentamiento al vapor antes de entrar a la turbina.

El proceso de sobrecalentamiento produce una mayor temperatura a la entrada de la turbina sin que aumente la presión máxima del ciclo. Después de que el vapor saturado sale de la caldera, el fluido pasa por otra sección de suministro de calor, en la cual la temperatura aumenta, en teoría, a presión constante. El vapor sale del sobrecalentamiento a temperatura que generalmente está restringida sólo por aspectos metalúrgicos, casi siempre temperaturas en el intervalo de 540 a 600°C ó 1000 a 1100°F.

A continuación se muestra un diagrama T-S con un sobrecalentamiento en la figura 4

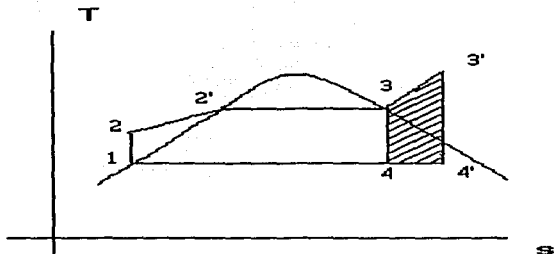


Figura 2.4. Diagrama T-s del ciclo Rankine con sobrecalentamiento.

El área sombreada del diagrama T-S representa la salida adicional de trabajo neto y el área bajo la curva 3-3', representa el calor adicional entregado en la sección de sobrecalentamiento. Según el análisis de la máquina de Carnot, se esperaría que la eficiencia del ciclo hubiese aumentado, y así es, pues la calidad del estado 4' es mayor que la de 4, por lo que el problema de la humedad se mitiga.

2.6. CICLO CON RECALENTAMIENTO.

El ciclo de Rankine simple adolece del hecho de que se agrega calor cuando varía la temperatura y el exceso de humedad durante el proceso de expansión es nocivo para la operación y vida útil de la turbina. El sobrecalentamiento tiende a ayudar a la turbina, pero termodinámicamente no aporta un aumento considerable de la eficiencia térmica del ciclo. Para solucionar esto, se propuso el ciclo con recalentamiento.

En este ciclo, no se deja que el vapor se expanda completamente en un solo paso hasta la presión del condensador. Después de la expansión parcial, el vapor es extraído de la turbina para una expansión adicional hasta la presión de escape. Se puede considerar que la turbina consiste en dos etapas, una de alta y otra de baja presión. Las figuras 5a y 5b muestra el ciclo junto con un diagrama T-S

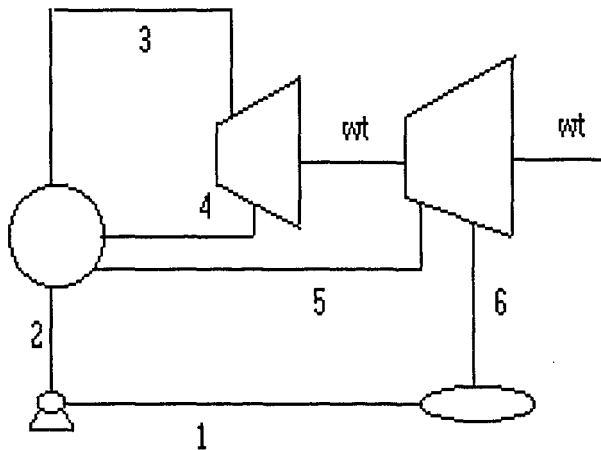


Figura 2.5.1. Esquema de un ciclo Rankine con una etapa de recalentamiento.

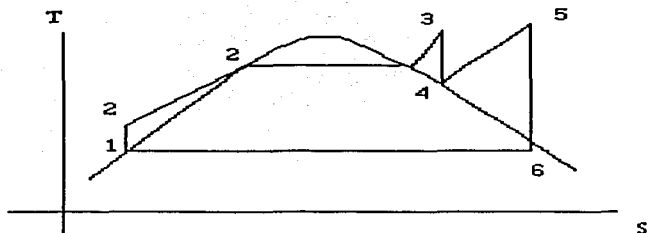


Figura 2.5.1. Diagrama T-S de un ciclo Rankine con una etapa de recalentamiento.

Al calcular la eficiencia térmica del ciclo, se toman en cuenta la salida del trabajo de las dos etapas de la turbina, así como las dos entradas de calor. Con los subíndices de la figura tenemos :

$$\eta = \frac{(h_3 - h_4) + (h_5 - h_6) - W_b}{(h_3 - h_2) + (h_3 - h_4)}$$

En ocasiones el ciclo de recalentamiento se llama ciclo de recalentamiento ideal si en la primera expansión, se expande el vapor de manera isoentrópica hasta la saturación y durante el calentamiento subsecuente, éste se regresa a la misma entalpía que tenía antes de la expansión.

Existe una presión del recalentamiento que maximizará la eficiencia térmica para los valores dados de P_3 , T_3 , T_5 y P_6 de la figura 5. La eficiencia máxima del ciclo ideal con recalentamiento suele ocurrir cuando el cociente P_4/P_3 está en la región de 0.15 a 0.35. En la figura 6 se muestra un diagrama T-S comparando un ciclo real con un ciclo ideal de trabajo para un ciclo con recalentamiento, siendo la línea punteada la parte real. Para tomar en cuenta los efectos no ideales de la turbina y de la bomba, se introducen las eficiencias de estas mismas.

2.6.1. Eficiencia de Bomba.

Se define la eficiencia adiabática de la bomba como: (2)

$$h_B = (h_{1'} - h_0) / (h_1 - h_0)$$

$$h_B = \text{Trabajo teórico bomba} / \text{Trabajo real bomba}$$

Por lo general, la eficiencia de la bomba varía entre el 69 y el 80% . El líquido a alta presión, estado 1, se empuja a la caldera a través de los tubos. Debido a las pérdidas en las tuberías, los codos y válvulas que ofrecen resistencia friccional al flujo del agua, ésta última, que debía estar en un estado de saturación 2' en un proceso libre de pérdidas, está realmente en el estado 2 cuando se inicia la evaporización en la caldera. El vapor producido en el estado 3' se sobrecalienta en los serpentines del sobrecalentador, donde ocurre una caída adicional de presión debido a la fricción. Por tanto, para el vapor en el estado 3, donde está listo para entrar a la turbina, la presión es 5 a 6% menor que la del agua a la salida de la bomba.

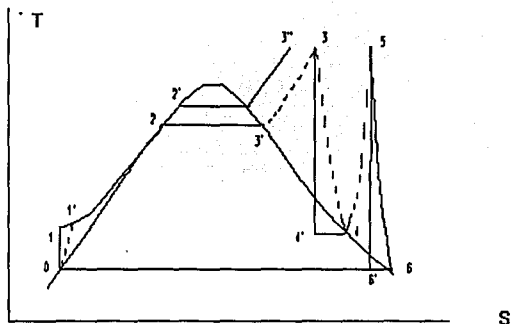


Figura 2.6. Comparación de un ciclo real vs. un ciclo ideal con recalentamiento.

2.6.2. Eficiencia de Turbina.

Se define la eficiencia adiabática de la turbina como: (3)

$$h_T = (h_3 - h_4) / (h_3 - h_4')$$

$$h_T = \text{Trabajo real de la turbina} / \text{Trabajo isoentrópico turbina.}$$

La expansión que ocurre en la turbina no es isoentrópica tal como se muestra en la figura, es precisamente con el valor de la eficiencia como podemos apreciar la desviación del valor real del ideal. Los valores comunes de eficiencia de la turbina varían de 80 a 90%.

Con estas variaciones, la eficiencia térmica total del ciclo real varía entre un 20 y un 35% del ciclo ideal.

2.7. CICLO REGENERATIVO.

Pese a las mejoras logradas, la eficiencia de un ciclo Rankine simple o con recalentamiento, es bastante baja comparada con la

eficiencia de un ciclo de Carnot que opere en los mismo rangos de temperatura. Dado que la mayor pérdida de energía en una planta de potencia se presenta en el condensador, en donde se desecha el calor enfriado, es pertinente considerar métodos de reducir este calor desechado y de mejorar la eficiencia del ciclo.

En los ciclo antes expuestos, el condensado se regresa a la caldera a la temperatura más baja del ciclo. El fluido se calienta hasta la saturación por la mezcla directa en el domo de vapor de la caldera. En vez de pasar por tales procedimientos, a continuación se considera un método de calentamiento de agua de alimentación que, aunque idealizado, conduce a ciertas conclusiones interesantes.

El método más deseable de calentamiento, sería uno que fuera reversible y continuo. Suponiendo que esto fuera posible tendríamos el ciclo mostrado en la figura 7

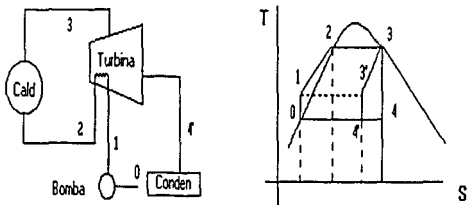


Figura 2.7. Ciclo ideal regenerativo.

El líquido sale de la bomba en el estado 1 y circula por serpentines ubicados alrededor de una turbina regenerativa a través de los cuales se

expande el vapor a partir del estado 3. El agua de alimentación se calienta regenerativamente por el vapor que se expande, para alcanzar la saturación en el estado 2. Entra entonces a la caldera donde se vaporiza para convertirse en el vapor saturado en el estado 3 a la entrada de la turbina. Después de una expansión no adiabática hasta el estado 3', el vapor se expande finalmente de modo isoentrópico de 3'-4', antes de condensarse y para ser recalentado otra vez como agua de alimentación. De modo ideal, la transferencia de energía entre el vapor y el agua es reversible y la energía transferida del vapor al agua (área bajo 3-3') es igual a la energía adquirida por el agua de alimentación (área bajo 1-2). Así ha habido una sola absorción de calor isotérmica de 2-3 a T_3 , y un solo rechazo de calor isotérmico de 4-0 a T_0 ; por tanto, la eficiencia queda:

$$\eta = T_0/T_3$$

El ciclo mostrado anteriormente no puede alcanzarse en la práctica debido a la complejidad de construir una turbina regenerativa, además el vapor de escape de la turbina contendría demasiada humedad. Por consiguiente, el ciclo real difiere del ideal en que hay un sangrado de la turbina de una parte del vapor, después de que ha sufrido una expansión parcial y se mezcla con el agua de alimentación en un intercambiador de calor.

El proceso real se muestra en la figura 8

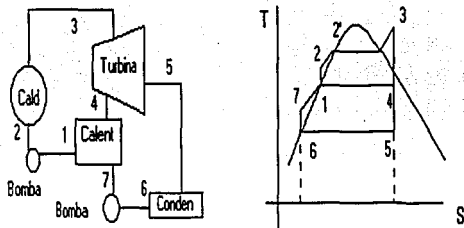


Figura 2.8. Ciclo real regenerativo.

El vapor en el estado 3 entra en la turbina y se expande hasta el estado 4, de donde hay un sangrado de una porción y se alimenta a un intercambiador de calor que llamaremos calentador de agua de enfriamiento para calentar el agua que procede de la bomba. El vapor sobrante, se expande hasta la presión del condensador, a la salida del condensador, se bombea a través del calentador de agua de enfriamiento para que el flujo se junte con el del sangrado de la turbina y alcanzar la saturación en el estado 1, correspondiente a la presión del vapor sangrado. El vapor alcanza el estado 3 en el sobrecalentado que viene después de la caldera antes de entrar a la turbina. Por tanto, el agua de alimentación, sólo recibe energía de los alrededores durante 2-2'-3 del ciclo. La temperatura termodinámica media de este suministro de energía se encuentra entre T_3 y T_2 , mientras que un ciclo sin calentamiento de agua de alimentación, tendría su temperatura media entre T_2' y T_3 .

Con todo esto, el ciclo no equivale al sistema ideal antes planteado, puesto que la transferencia de calor ocurre de modo irreversible. Además, para alcanzar el estado ideal, serían necesarios un número infinito de

calentadores de agua con sangría de vapor en todos los puntos de la turbina.

En las instalaciones de grandes centrales se utilizan desde 1 hasta 12 calentadores para el agua de alimentación.

En la siguientes páginas se presentan unas gráficas que comparan las eficiencias de los distintos ciclos, apreciándose la superioridad del ciclo regenerativo. (4)

Comúnmente se usan dos tipos de calentadores de agua: el calentador de agua abierto o de contacto, y el calentador de agua cerrado o de superficie.

2.7.1. Calentador abierto de agua.

El vapor se pone directamente en contacto con el agua subenfriada y se mezcla con la misma. Las porciones de agua y vapor están ordenadas de tal manera que el agua de alimentación se calienta hasta el estado de saturación correspondiente a la presión de la bomba.

Suponiendo que el flujo total de vapor sea m y que la fracción m_a sea el sangrado de la turbina, entonces tenemos el siguiente balance de energía el calentador:

$$(m-m_a)h_7 + m_a h_4 = mh_1$$

El sistema es altamente sensible a fluctuaciones de la carga de la planta. Además, si se utilizan varios calentadores de agua de alimentación, es necesario tener una bomba asociada por cada calentador, de modo que se mantenga la presión correcta igual a la del vapor en dicha etapa. Esto resulta bastante molesto y costoso de disponer. Por consiguiente, se acostumbra usar sólo un calentador de alimentación. Las características de transferencia de calor son excelentes, puesto que es

posible calentar el agua de alimentación a la misma temperatura que el vapor, si el vapor no está sobrecalentado.

2.7.2. Calentador cerrado de agua.

Como su nombre lo indica, las dos corrientes no se mezclan directamente. El agua de alimentación que sale del condensador circula en tubos que atraviesan el calentador; mientras que el vapor que sale de la turbina, entra al calentador y se condensa en el exterior de los tubos en los que fluye el agua de alimentación. Para una transferencia de calor efectiva, es necesario que haya una diferencia de 1 a 3°C de temperatura entre el vapor que se condensa y el agua que sale del calentador. Esta diferencia de temperatura se conoce como diferencial terminal de temperatura.

A continuación se muestra una representación esquemática de un calentador cerrado de agua.

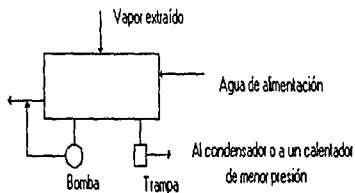


Figura 2.9. Esquema de calentador cerrado.

En este tipo de calentadores se tiene dos opciones diferentes para disponer del vapor sangrado después de la condensación del mismo.

1. Se utiliza una bomba, la cual toma el vapor condensado y lo bombea a la salida del calentador de agua de alimentación.

Efecto de Temperatura de sangrado en la Eficiencia del Ciclo.

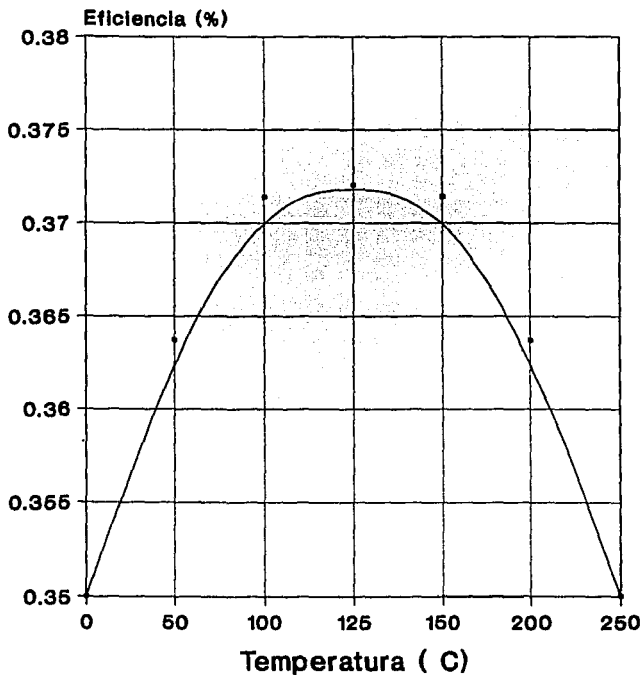


Figura 2.10

Presión de la caldera vs Eficiencia.

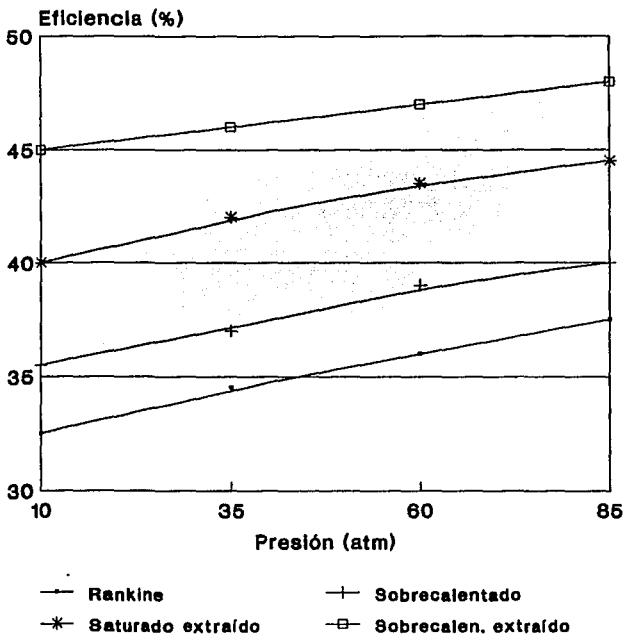


Figura 2.11

2. Se utiliza una trampa que sólo permita que fluya el líquido a otra región de menor presión, que podría ser el mismo condensador u otro calentador de agua de alimentación.

Independientemente de la opción que se seleccione, se debe cumplir el siguiente balance de masa y energía:

$$(mDH)_{\text{extraído}} = (mDH)_{\text{agua de alimentación}}$$

Una ventaja de un calentador cerrado, es que las presiones del vapor extraído y la del agua de alimentación pueden ser muy diferentes. Por su tipo de construcción, los calentadores cerrados operan a presiones mayores que los de tipo abierto.

Las modernas plantas de potencia de vapor generalmente utilizan varios calentadores de agua de alimentación abiertos y cerrados. Aunque aumenta la eficiencia térmica al crecer el número de calentadores, por desgracia también aumenta el costo.

2.8. CICLO COMBINADO.

Ocasionalmente se han usado modificaciones de los ciclos de vapor básicos, en interés de la mayor economía de la operación; cualquier planta de fuerza que esté usando recalentamiento, también puede emplear etapas regenerativas. La combinación de los dos ciclos de vapor conservará las ventajas de cada uno de ellos, y no se producirán desventajas adicionales; pese a esto, la energía disipada es, en ambos casos, una gran fracción de la entrada de energía. Un ciclo generador de potencia combinado se basa en la conexión o acoplamiento de dos ciclos de potencia distinto de tal manera que el calor disipado en uno se use parcial o totalmente como fuente de calor del otro. Algunas combinaciones

de los ciclos de vapor con la turbina de gas son técnicas interesantes, y posiblemente se perfeccionen bastante, esto es, combinar una turbina de gas (ciclo Brayton) con un ciclo de turbina de vapor de agua (ciclo Rankine).

En un ciclo de turbina de gas, la corriente de escape que sale de la turbina es relativamente caliente, un método para aprovechar esta energía de alta temperatura es por medio de regeneración; una posibilidad alternativa es utilizar la corriente de escape de la turbina de gas como la fuente de energía en la caldera de un ciclo común de potencia con vapor de agua. Sin embargo, el recalentamiento en la sección de la turbina del ciclo de vapor rara vez será posible.

El vapor de agua tiene propiedades físicas no muy convenientes en sus dos extremos de expansión, pero ninguna otra sustancia común se adapta mejor a ambos extremos que el vapor de agua. El principal problema del vapor de agua es su temperatura crítica. Al usar una caldera, el calor se suministra al fluido a temperatura constante y relativamente baja. Esta entrega de calor a baja temperatura podría mejorarse parcialmente si se utilizará una sustancia a una temperatura crítica mucho más elevada.

Se ha descubierto que las propiedades de los metales alcalinos son apropiadas para los ciclos Rankine de altas temperaturas.. Los más promisorios son el potasio y el sodio con sus mezclas. Dichos metales tienen propiedades termodinámicas convenientes como coeficientes de transferencia de calor razonablemente altos. En consecuencia el tamaño de cualquier intercambiador de calor se reduce considerablemente. Este ciclo se ha propuesto para la generación de energía eléctrica en naves espaciales.

El estudio y uso de estos metales alcalinos ha renovado el interés por los ciclos Binarios. Dichos ciclos son aquéllos en los cuales el calor extraído durante el proceso de eliminación de un ciclo de potencia se usa como la entrada de calor para otro ciclo.

El ciclo vapor-mercurio es el único ciclo binario de vapor operado en escala comercial hasta el presente. Las ventajas del mercurio como fluido de un ciclo de vapor son éstas:

1. Tiene presión de vapor moderada a altas temperaturas, por ejemplo a 537.78°C su presión de saturación es menor de 13.788 bar.
2. El líquido tiene una alta densidad, que es una propiedad conveniente para asegurar la separación del vapor del líquido en la caldera, el poder alimentar el líquido a la caldera con sólo presión hidrostática en vez de bombeo.
3. Baja entalpía específica que resulta en velocidades bajas en las toberas de la turbina, permitiendo que se usen turbinas sencillas y de pocas etapas.

Contra estas ventajas debe considerarse el alto costo del mercurio, la limitación aparente de su abastecimiento, sus cualidades tóxicas y su penetración, dado que puede introducirse en las juntas o grietas que serían impermeables en el agua o al vapor.

NOTAS DEL CAPITULO 2.

(1) cfr Granet Irving, Termodinámica, México, Ed Prentice Hall, 1988, (3ra edición), p 158.

(2) cfr Kadambi V. Conversión de Energía, México, Ed. Limusa, 1984, (1ra edición), tomo 2, p 215.

(3) cfr Idem 216

(4) cfr Idem 224.

CAPITULO 3
EJEMPLOS DE CICLOS DE POTENCIA

3.1. OBJETIVO.

Dentro de este capítulo, veremos las soluciones de diversos problemas de ciclos de potencia. Dichos problemas van desde el caso más sencillo de un ciclo de Carnot, hasta el de un ciclo con regeneración y recalentamiento.

El objetivo de presentar estos ejemplos, es el comenzar a justificar la elaboración del programa, ya que como se verá, sobre todo en los últimos casos, el gran número de cálculos necesarios para obtener todos los puntos de un ciclo, llegan a complicar mucho el diseño de las condiciones de operación de la planta.

Estos problemas pueden resolverse por dos métodos:

1. Utilizando un diagrama de Mollier.
2. Utilizando Tablas de propiedades de las sustancias.

Dado que el programa utilizará tablas, todos los problemas están resueltos con este método.

Todos los ejercicios fueron tomado de la sección de problemas del libro: WARK Kenneth Termodinámica, México, Editorial Mc Graw Hill, 1988, (4 edición).

A continuación se presentan dichos casos.

3.2. CICLO DE CARNOT CON VAPOR.

Una máquina de Carnot contiene 0.10 Kg de agua. Al principio del proceso de suministro de calor, el fluido es un líquido saturado y al final de este proceso es un vapor saturado. El suministro de calor ocurre a 120 bar y la disipación de calor es a 0.3 bar. Obténgase:

- a) La calidad final de la expansión adiabática y al final de la disipación isotérmica de calor.

- b) La eficiencia térmica.
 c) El calor suministrado por ciclo, en kilojoules.
 d) Salida neta de trabajo por ciclo.

Primeramente obtenemos todos los puntos del sistema

-Punto 2:

$$P_2 = 120 \text{ bar}$$

$$h_2 = h_f = 1494.3 \text{ KJ/Kg}$$

-Punto 3:

$$P_3 = 120 \text{ bar}$$

$$h_3 = h_g = 2684.9 \text{ KJ/Kg}, \quad s_3 = 5.4924 \text{ KJ/Kg}^\circ\text{C}$$

-Punto 4:

$$P_3 = 0.3 \text{ bar}$$

$$s_3 = s_4 = 5.4924 \text{ KJ/Kg}^\circ\text{C}, \quad s_g = 7.7686 \text{ KJ/Kg}^\circ\text{C}$$

Como $s_3 < s_g$, estamos en la región de la campana.

$$s_f = 0.9439 \text{ KJ/Kg}^\circ\text{C}$$

Obtenemos la calidad de la mezcla con la siguiente ecuación

$$s_4 = s_f + x(s_{fg})$$

$$x = (5.4924 - 0.9439) / (7.7686 - 0.9439) = 0.66647$$

$$h_4 = h_f + x(h_{fg}) \quad \text{donde } h_f = 289.23 \text{ KJ/Kg y } h_{fg} = 2336.1 \text{ KJ/Kg}$$

$$h_4 = 289.23 + 0.66647(2336.1) = 1845.04 \text{ KJ/Kg}$$

-Punto 1:

$$P_1 = 0.3 \text{ bar}$$

$$s_1 = s_2 = 3.4962 \text{ KJ/Kg}^\circ\text{C},$$

$$s_g = 7.7686 \text{ KJ/Kg}^\circ\text{C}, \quad s_f = 0.9439 \text{ KJ/Kg}^\circ\text{C}$$

$$x = (3.4962 - 0.9439) / (7.7686 - 0.9439) = 0.3739$$

$$h_1 = h_f + x(h_{fg}) \quad \text{donde } h_f = 289.23 \text{ KJ/Kg y } h_{fg} = 2336.1 \text{ KJ/Kg}$$

$$h_1 = 289.23 + 0.3739(2336.1) = 1162.88 \text{ KJ/Kg}$$

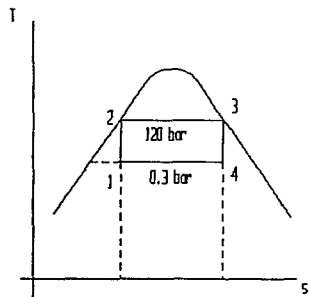
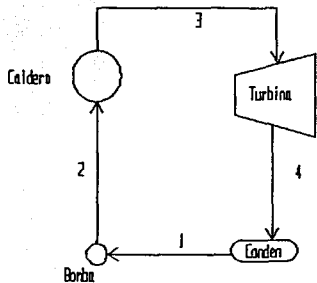


Figura 3.1 Ciclo de Carnot

a)

La calidad al final de la expansión isoentrópica es la que ya calculamos, x_1
 $= 0.66647$

La calidad al final de la disipación de calor también ya esta calculada, $x_2 =$
 0.3739

b)

Para hallar la eficiencia usamos la ecuación vista en el capítulo 1

$$\eta = 1 - T_1/T_2 = 1 - (69.1 + 273)/(324.8 + 273) = 0.4277$$

c)

Para el calor suministrado: $q = h_3 - h_2$

$$q = 2684.9 - 1491.3 = 1193.6 \text{ KJ/kg}$$

$$Q = q \cdot m = 1193.6 \text{ KJ/Kg} \cdot 0.1 \text{ Kg} = 119.36 \text{ KJ}$$

d)

Para el trabajo neto

$$w_{\text{neto}} = h_3 - h_4 - (h_2 - h_1)$$

$$w_{\text{neto}} = 2684.9 - 1845.04 - (1491.3 - 1162.88) = 511.44 \text{ KJ/Kg}$$

$$w_{\text{neto}} = 511.44 \text{ KJ/Kg} \cdot 0.1 \text{ Kg} = 51.1 \text{ KJ}$$

3.3. CICLO RANKINE.

Un ciclo Rankine tiene una presión en el escape de 0.08 bar y una temperatura a la entrada de la turbina de 520°C. Encuentrese:

- El contenido de humedad a la salida de la turbina.
- La eficiencia térmica del ciclo para una presión a la entrada de la turbina de 120 bar.

-Punto 3:

$$\text{De tablas } h_3 = 3401.8 \text{ KJ/Kg} \quad s_3 = 6.5555 \text{ kJ/Kg}^\circ\text{C}$$

-Punto 4:

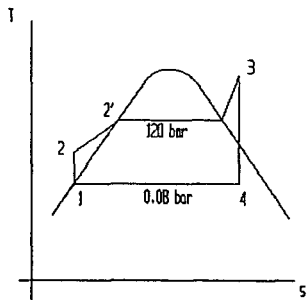
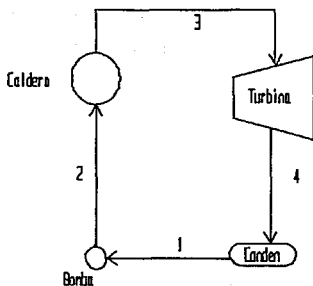


Figura 3.2 Ciclo Rankine

$$P_4 = 0.08 \text{ bar}$$

$$s_3 = s_4 = 6.5555 \text{ KJ/Kg}^\circ\text{C}, s_g = 8.2287 \text{ KJ/Kg}^\circ\text{C} \text{ y } s_f = 0.5926 \text{ KJ/Kg}^\circ\text{C}$$

$$h_f = 173.88 \text{ KJ/Kg}, h_{fg} = 2403.1 \text{ KJ/Kg}$$

$$x = (s_4 - s_f) / (s_g - s_f) = 0.7808$$

$$h_4 = h_f + x \cdot (h_{fg}) = 173.88 + 0.7808 \cdot (2403.1) = 2048.298 \text{ KJ/Kg}$$

-Punto 1:

$$\text{De tablas } h_1 = h_f = 173.88 \text{ KJ/Kg} \text{ y } v_f = 1.0084 \cdot 10^{-3} \text{ m}^3/\text{Kg}$$

$$\text{Para el trabajo de bomba } w = v_f \cdot (P_2 - P_1)$$

$$w = 1.0084 \cdot (120 - 0.08) / 10 = 12.092 \text{ KJ/Kg}$$

Dividiendo todo entre 10, obtenemos ya las unidades convenientes.

-Punto 2:

$$h_2 = h_1 + w = 173.88 + 12.092 = 185.972 \text{ KJ/Kg}$$

a)

En el punto 4 obtuvimos la calidad de vapor de la mezcla, por tanto, el porcentaje de humedad está dado por:

$$\% \text{ humedad} = 1 - 0.7808 = 0.2191$$

b)

Para la eficiencia térmica, necesitaremos el trabajo neto y el calor suministrado.

$$q = h_3 - h_2 = 3401.8 - 185.97 = 3215.828 \text{ KJ/Kg}$$

$$w = w_{\text{turbina}} - w_{\text{bomba}} = (h_3 - h_4) - w_{\text{bomba}} = 1341.41 \text{ KJ/Kg}$$

$$\eta = w/q = 1341.41 / 3215.83 = 0.4171\%$$

3.4. CICLO CON RECALENTAMIENTO.

Un ciclo ideal de Rankine genera vapor a 140 bar y 560°C y condensa vapor a 0.06 bar. Se tiene además un recalentamiento a 10 bar y 540°C. La salida neta de potencia es de 20 MW. Obténgase:

- a) La entrada de calor en KJ/Kg.
 b) La eficiencia térmica.
 c) El flujo de vapor en kg/h.
 d) El flujo de agua de enfriamiento que requiere el condensador si el agua sufre un aumento en su temperatura de 10°C.

-Punto 3

$$\text{De tablas } h_3 = 3486 \text{ KJ/Kg} \quad s_3 = 6.5941 \text{ KJ/Kg}^\circ\text{C}$$

-Punto 4

$$P_4 = 10 \text{ bar}$$

$$s_4 = s_3 = 6.5941 \text{ KJ/Kg}^\circ\text{C}$$

$s_g = 6.5863 \text{ KJ/Kg}^\circ\text{C}$, Como $s_4 > s_g$, estamos en la región de vapor sobrecalentado.

$$\text{Interpolando } h_4 = 2781.62 \text{ KJ/Kg}$$

-Punto 5

$$P_5 = 10 \text{ bar}, T_5 = 540^\circ\text{C}$$

$$\text{Interpolando } h_5 = 3565.6 \text{ KJ/Kg y } s_5 = 7.8720 \text{ KJ/Kg}^\circ\text{C}$$

-Punto 6

$$P_6 = 0.06 \text{ bar}$$

$$s_6 = s_5 = 7.8720 \text{ KJ/Kg}^\circ\text{C}, s_g = 8.3304 \text{ KJ/Kg}^\circ\text{C}, s_f = 0.5210 \text{ KJ/Kg}^\circ\text{C}$$

$$x = (7.8720 - 0.5210) / (8.3304 - 0.5210) = 0.9413$$

$$h_6 = h_f + xh_g = 151.53 + .9413(2415.9) = 2425.62 \text{ KJ/Kg}$$

-Punto 1

$$P_1 = 0.06 \text{ bar}$$

$$h_1 = h_f = 151.53 \text{ KJ/Kg y } v_1 = 1.0064 \text{ cm}^3/\text{g}$$

$$w_{\text{bomba}} = v_1(P_2 - P_1) = 1.0064 \cdot (140 - 0.06) / 10 = 14.083 \text{ KJ/Kg}$$

-Punto 2

$$h_2 = h_1 + w_{\text{bomba}} = 151.53 + 14.083 = 165.61 \text{ KJ/Kg}$$

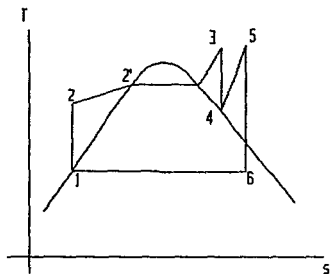
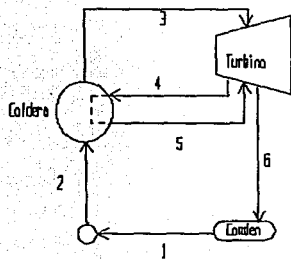


Figura 3.3 Ciclo con Recalentamiento

a)

$$q_{\text{sum}} = h_3 - h_2 = 3486 - 165.61 = 3320.38$$

b)

$$w_{\text{neto}} = (h_3 - h_4) + (h_5 - h_6) - w_{\text{bomba}}$$

$$w_{\text{neto}} = 3486 - 2781.62 + 3565.6 - 2425.62 - 14.083 = 1830.27 \text{ KJ/Kg}$$

$$\eta = w_{\text{neto}} / (q_{\text{sum}} + q_{\text{rec}})$$

$$\eta = 1830.27 / (3320.38 + (h_5 - h_4)) = 0.4459$$

c)

$$m = W/w = 20 \cdot 10^6 / 1830.27 = 10.92 \text{ Kg/s} = 39338.45 \text{ Kg/h}$$

d)

$$Q = m C_p (D T)$$

$$Q = (h_6 - h_1) \cdot m_{\text{vapor}} = m_{\text{agua}} \cdot C_p \cdot (D T)$$

a Temperatura ambiente el C_p del agua es 4.1818 KJ/Kg°C

$$(2425.62 - 151.53) \cdot 10.92 = m(4.1818)(10)$$

$$m = 593.83 \text{ Kg/s} = 2137812 \text{ Kg/h}$$

3.5. CICLO REGENERATIVO.

El vapor entra a la turbina de un ciclo regenerativo ideal a 40 bar y 500°C. Se extrae vapor a 7 y 3 bar que entra a dos calentadores abiertos de agua de alimentación que están conectados en serie. Después del condensador, que opera a 0.06 bar y después de cada calentador, se utilizan bombas apropiadas. Obténgase :

a) La fracción total de flujo que a los calentadores de 3 y 7 bar.

b) La salida de trabajo de la turbina y el trabajo total de bombeo en KJ/Kg del flujo total.

c) La eficiencia térmica.

-Punto 3

De tablas $h_3 = 3445.3 \text{ KJ/Kg}$ y $s_3 = 7.0901 \text{ KJ/Kg°C}$

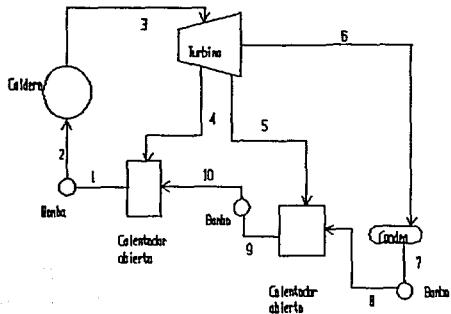
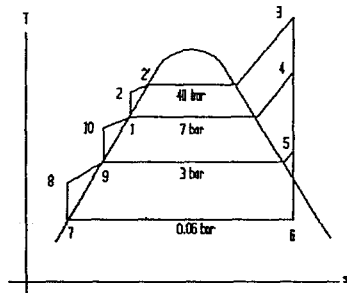


Figura 3.4 Ciclo Regenerativo



-Punto 4

$$P_4 = 7 \text{ bar}$$

$s_4 = s_3 = 7.0901 \text{ KJ/Kg}^\circ\text{C}$. Estamos en la región de vapor sobrecalentado.

Interpolando $h_4 = 2946.06 \text{ KJ/Kg}$

-Punto 5

$$P_5 = 3 \text{ bar}$$

$s_5 = s_3 = 7.0901 \text{ KJ/Kg}^\circ\text{C}$. Estamos en la región de vapor sobrecalentado.

Interpolando $h_5 = 2766.54 \text{ KJ/Kg}^\circ\text{C}$

-Punto 6

$$P_6 = 0.06 \text{ bar}$$

$s_6 = s_3 = 7.0901 \text{ KJ/Kg}^\circ\text{C}$, $s_g = 8.3304 \text{ KJ/Kg}^\circ\text{C}$, $s_f = 0.5210$

$$x = (s_6 - s_f) / (s_g - s_f) = (7.0901 - 0.521) / (8.3304 - 0.521) = 0.841$$

$$h_6 = h_f + x \cdot h_{fg} = 151.53 + 0.841 \cdot (2415.9) = 2183.3 \text{ KJ/Kg}$$

-Punto 7

$$P_7 = 0.06 \text{ bar}$$

$$h_7 = h_f = 151.53 \text{ KJ/Kg} \text{ y } v_7 = 1.0064 \text{ cm}^3/\text{g}$$

$$w_{\text{bomba}} = v_7(P_8 - P_7) = 1.0064 \cdot (3 - 0.06) / 10 = 0.2958 \text{ KJ/Kg}$$

-Punto 8

$$h_8 = h_7 + w_{\text{bomba}} = 151.53 + 0.2958 = 151.82 \text{ KJ/Kg}$$

-Punto 9

$$P_9 = 3 \text{ bar}$$

$$h_9 = h_f = 561.47 \text{ KJ/Kg} \text{ y } v_9 = 1.0732 \text{ cm}^3/\text{g}$$

$$w_{\text{bomba}} = v_9(P_{10} - P_9) = 1.0732 \cdot (7 - 3) / 10 = 0.4298 \text{ KJ/Kg}$$

-Punto 10

$$h_{10} = h_9 + w_{\text{bomba}} = 561.47 + 0.4298 = 561.89 \text{ KJ/Kg}$$

-Punto 1

$$P_1 = 7 \text{ bar}$$

$$h_1 = h_f = 697.22 \text{ KJ/Kg y } v_1 = 1.1080 \text{ cm}^3/\text{g}$$

$$w_{\text{bomba}} = v_1(P_2 - P_1) = 1.1080 \cdot (40 - 7) / 10 = 3.6564 \text{ KJ/Kg}$$

-Punto.2

$$h_2 = h_1 + w_{\text{bomba}} = 697.22 + 3.6564 = 700.87 \text{ KJ/Kg}$$

Viendo los flujos de los calentadores :

Planteamos las siguientes ecuaciones de conservación de masa:

$$1... m_9 = m_5 + m_7 \quad \text{donde } m_7 = m_6 = m_8$$

$$2... m_4 + m_{10} = m_1$$

$$3... m_1 = m_4 + m_5 + m_6 \quad \text{por tanto:}$$

$$4... m_1 = m_4 + m_9$$

Planteamos las ecuaciones de conservación de la energía:

$$5... m_{10}h_{10} + m_4h_4 = m_1h_1$$

$$6... m_9h_9 = m_5h_5 + m_7h_7$$

Dividiendo 4 y 5 entre m_1 obtenemos:

$$4'... y_{10} + y_4 = 1$$

$$5'... y_{10}h_{10} + y_4h_4 = h_1 \quad 5'... 561.89 \cdot y_{10} + 2946.06 \cdot y_4 = 697.22$$

Resolviendo las ecuaciones simultáneas

$$y_{10} = 0.9432 \quad y_4 = 0.05675$$

Dividiendo 1 y 6 entre m_1 obtenemos:

$$1'... y_9 = y_5 + y_7$$

$$2'... y_9h_9 = y_5h_5 + y_7h_7 \quad 2'... 0.9432 \cdot 561.47 = 2766.54 \cdot y_5 + 151.53 \cdot y_7$$

Resolviendo las ecuaciones simultáneas

$$y_5 = 0.1478 \quad y_7 = 0.795$$

a)

$$y_{3\text{bar}} = 0.147, \quad y_{7\text{bar}} = 0.056$$

$$b) \quad w_{\text{turbina}} = h_3 - h_4 + (1 - y_7)(h_4 - h_5) + (1 - y_4 - y_7)(h_5 - h_6)$$

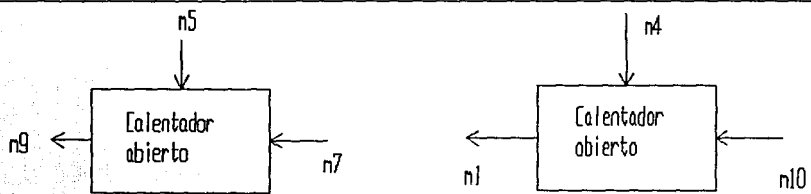


Figura 3.5.1 Flujos en el ciclo regenerativo.

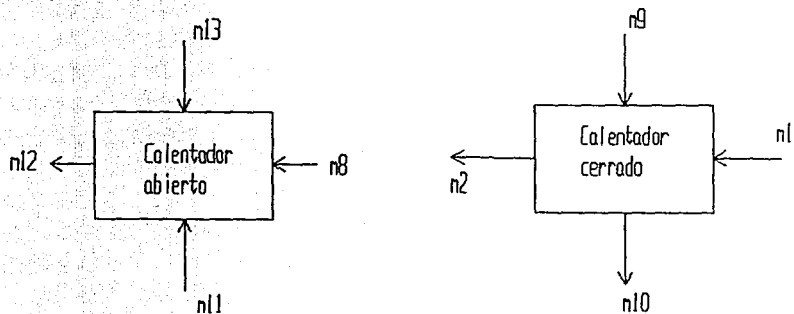


Fig. 3.5.2 Ciclo regenerativo y recalentamiento.

$$w_{\text{turbina}} = 3445.3 - 2946.06 + (1 - 0.0567) \cdot (2946.06 - 2766.54) + (1 - 0.0567 - 0.1478) \cdot (2766.54 - 2183.3) = 1132.54 \text{ KJ/Kg}$$

$$w_{\text{bomba}} = w_{\text{bomba1-2}} + (1 - \gamma_7) w_{\text{bomba10-9}} + (1 - \gamma_7 - \gamma_4) w_{\text{bomba7-8}}$$

$$w_{\text{bomba}} = 3.6564 + (1 - 0.0567) \cdot 0.4292 + (1 - 0.0567 - 0.1478) \cdot 0.2958 = 4.25 \text{ KJ/Kg}$$

c)

$$w_{\text{neto}} = 1132.54 - 4.25 = 1128.29 \text{ KJ/Kg}$$

$$q_{\text{sum}} = h_3 - h_2 = 3445.3 - 700.87 = 2744.43 \text{ KJ/Kg}$$

$$\eta = 1128.29 / 2744.43 = 0.411$$

3.6. CICLO CON RECALENTAMIENTO Y REGENERACION.

Un ciclo ideal con recalentamiento y regeneración opera con las condiciones de entrada a la turbina de 140 bar y 600°C, con un recalentamiento a 7 bar hasta 500°C. Un calentador cerrado de agua de alimentación trabaja a 15 bar y el condensado del calentador cerrado se capta en un calentador abierto de agua de alimentación que opera a 3 bar. La presión en el condensador es de 0.06 bar. Encuentre:

- La eficiencia térmica del ciclo
- El flujo necesario en el generador de vapor para obtener una potencia de salida de la turbina de 100 000 KW.

-Punto 3

$$P_3 = 140 \text{ bar} \quad T_3 = 600 \text{ °C}$$

$$\text{Interpolando } h_3 = 3591.1 \text{ KJ/Kg} \quad s_3 = 6.7172 \text{ KJ/Kg°C}$$

-Punto 4

$$P_4 = 7 \text{ bar} \quad s_4 = s_3 = 6.7172 \text{ KJ/Kg°C}$$

$$\text{Interpolando } h_4 = 2767.59 \text{ KJ/Kg}$$

-Punto 9

$$P_9 = 15 \text{ bar}$$

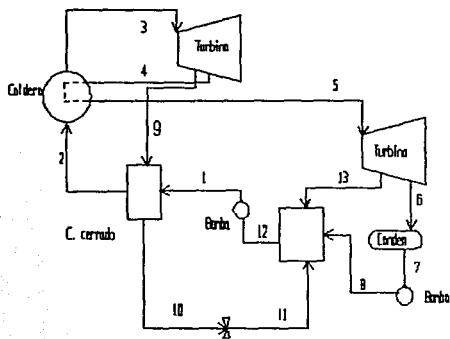
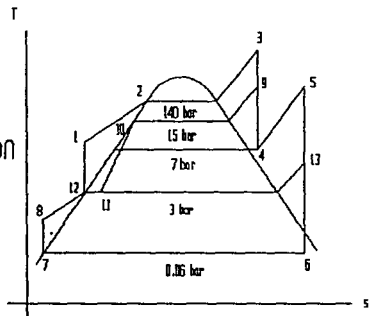


Figura 3.6 Ciclo con regeneración y recalentamiento



$$s_g = s_3 = 6.7172 \text{ KJ/Kg}^\circ\text{C}$$

$$\text{Interpolando } h_g = 2928.28 \text{ KJ/Kg}^\circ\text{C}$$

-Punto 5

$$P_5 = 7 \text{ bar } T_5 = 500 \text{ }^\circ\text{C}$$

$$\text{Interpolando } h_5 = 3481.7 \text{ KJ/Kg } s_5 = 7.9299 \text{ KJ/Kg}^\circ\text{C}$$

-Punto 13

$$P_{13} = 3 \text{ bar}$$

$$s_{13} = s_5 = 7.9299 \text{ KJ/Kg}^\circ\text{C}$$

$$\text{Interpolando } h_{13} = 3207.72 \text{ KJ/Kg}$$

-Punto 6

$$P_6 = 0.06 \text{ bar}$$

$$s_6 = s_5 = 7.9299 \text{ KJ/Kg}^\circ\text{C}, s_g = 8.3304 \text{ KJ/Kg}^\circ\text{C}, s_f = 0.521 \text{ KJ/Kg}^\circ\text{C}$$

$$x = (s_6 - s_f) / (s_g - s_f) = (7.9299 - 0.521) / (8.3304 - 0.521) = 0.9487$$

$$h_6 = h_f + x \cdot h_{fg} = 151.53 + 0.9487 \cdot (2415.9) = 2443.49 \text{ KJ/Kg}$$

-Punto 7

$$P_7 = 0.06 \text{ bar}$$

$$h_7 = h_f = 151.53 \text{ KJ/Kg } y v_7 = 1.0064 \text{ cm}^3/\text{g}$$

$$w_{\text{bomba}} = v_7 \cdot (P_8 - P_7) = 1.0064 \cdot (3 - 0.06) / 10 = 0.2958 \text{ KJ/Kg}$$

-Punto 8

$$h_8 = h_7 + w_{\text{bomba}} = 151.53 + 0.2958 = 151.82 \text{ KJ/Kg}$$

-Punto 12

$$P_{12} = 3 \text{ bar}$$

$$h_{12} = h_f = 561.47 \text{ KJ/Kg } y v_{12} = 1.0732 \text{ cm}^3/\text{g}$$

$$w_{\text{bomba}} = v_{12} \cdot (P_1 - P_{12}) = 1.0732 \cdot (140 - 3) / 10 = 14.702 \text{ KJ/Kg}$$

-Punto 1

$$h_1 = h_{12} + w_{\text{bomba}} = 561.47 + 14.702 = 576.17 \text{ KJ/Kg}$$

-Punto 10

$$P_{10} = 15 \text{ bar} \quad T_{10} = 198.3 \text{ }^\circ\text{C}$$

$$h_{10} = h_f = 844.84 \text{ KJ/Kg}$$

-Punto 11

En la mayoría de las aplicaciones, los dispositivos de estrangulamiento están aislados térmicamente o la transferencia térmica es insignificante. En este caso, para un proceso de estrangulamiento, el cambio de entalpía es cero, por tanto

$$h_{11} = h_{10} = 844.84 \text{ KJ/Kg}$$

-Punto 2

$$P_2 = 140 \text{ bar} \quad T_2 = T_{10} = 198.3 \text{ }^\circ\text{C}$$

Como $T_2 < T$ dentro de la campana ($336.8 \text{ }^\circ\text{C}$) para la presión de 140 bar, indica que estamos en la región de líquido comprimido.

$$\text{De tablas } h_2 = 851.04 \text{ KJ/Kg}$$

Viendo los flujos de los calentadores:

Para el calentador cerrado planteamos :

$$1... m_g = m_{10} = m_{11}$$

$$2... m_1 = m_2 = m_{12}$$

$$3... m_g h_g - m_{10} h_{10} = m_1 h_1 + m_2 h_2$$

Dividiendo 3 entre m_1

$$3'... y_g (h_g - h_{10}) = (h_2 - h_1), \quad 3'... y_g (2928.28 - 844.84) = (851.04 - 576.17)$$

$$y_g = 0.1319$$

Para el calentador abierto plantamos:

$$4... m_{12} = m_{13} + m_8 + m_{11} \quad 5... m_3 = m_9 + m_4$$

$$6... m_4 = m_{13} + m_6 \quad 7... m_{12} h_{12} = m_{13} h_{13} + m_8 h_8 + m_{11} h_{11}$$

Dividiendo 4 y 7 entre m_{12} (ver ecuación 2)

$$4'... 1 = y_{13} + y_8 + y_{11}$$

$$5'... h_{12} = y_{13} h_{13} + y_8 h_8 + y_{11} h_{11}$$

$$5'... 561.47 = 3207.72 \cdot y_{13} + 151.52 \cdot y_8 + 1.1319 \cdot (844.84)$$

$$y_{13} = 0.1041 \quad y_8 = 0.7639$$

a)

$$q_{\text{sum}} = 1(h_3 - h_2) + (1 - y_9)(h_5 - h_4)$$

$$q_{\text{sum}} = (3591.1 - 851.04) + (1 - 0.1319) \cdot (3481.7 - 2767.59) = 3359.98 \text{ KJ/Kg}$$

$$w_{\text{tur}} = 1(h_3 - h_9) + (1 - y_9)(h_9 - h_4) + (1 - y_9)(h_5 - h_{13}) + (1 - y_9 - y_{13})(h_{13} - h_6)$$

$$w_{\text{tur}} = (3591.1 - 2928.28) + (1 - 0.1319) \cdot (2928.28 - 2767.59) +$$

$$(1 - 0.1319) \cdot (3481.7 - 3207.72) + (0.7639) \cdot (3207.72 - 2443.5)$$

$$w_{\text{tur}} = 1623.94 \text{ KJ/Kg}$$

$$w_{\text{bomb}} = (1 - y_9 - y_{13})(h_8 - h_7) + (h_1 - h_{12}) = (0.7639) \cdot (0.2958) + 14.702 =$$

$$14.9279 \text{ KJ/Kg}$$

$$w_{\text{neto}} = 1623.94 - 14.9279 = 1609.017 \text{ KJ/Kg}$$

$$\eta = 1609.017 / 3359.98 = 0.478$$

b)

$$qm = Q \quad m = Q/q = 100\,000 / 1484.85 = 67.34 \text{ Kg/s.}$$

3.7. CICLO CON RECALENTAMIENTO, REGENERACION Y EFICIENCIA DE TURBINA Y BOMBA.

Resolver el problema anterior considerando ahora una eficiencia de turbina del 90% y una eficiencia de bombas de 75%

El esquema del ciclo se muestra en la siguiente hoja.

-Punto 3

$$P_3 = 140 \text{ bar} \quad T_3 = 600 \text{ }^\circ\text{C}$$

$$\text{Interpolando } h_3 = 3591.1 \text{ KJ/Kg} \quad s_3 = 6.7172 \text{ KJ/Kg}^\circ\text{C}$$

-Punto 4

$$P_4 = 7 \text{ bar}$$

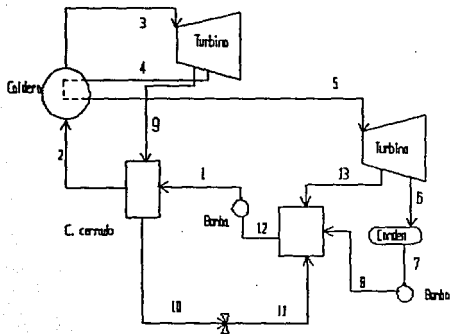
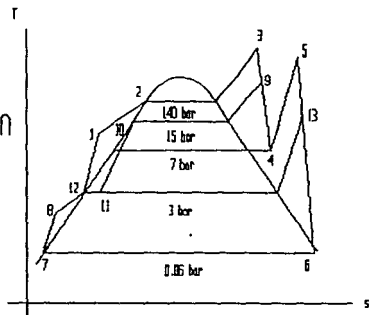


Figura 3.7 Ciclo con Regeneración y Recalentamiento



$$s_4' = s_3 = 6.7172 \text{ KJ/Kg}^\circ\text{C}$$

$$\text{Interpolando } h_4' = 2767.59 \text{ KJ/Kg}$$

$$h_4 = h_3 - h_{tur} \cdot (h_3 - h_4') = 3591.1 - 0.9 \cdot (3591.1 - 2767.59) = 2849.94 \text{ KJ/Kg}$$

-Punto 9

$$P_9 = 15 \text{ bar}$$

$$s_9' = s_3 = 6.7172 \text{ KJ/Kg}^\circ\text{C}$$

$$\text{Interpolando } h_9' = 2928.28 \text{ KJ/Kg}^\circ\text{C}$$

$$h_9 = 3591.1 - 0.9 \cdot (3591.1 - 2928.28) = 2994.562 \text{ KJ/Kg}$$

-Punto 5

$$P_5 = 7 \text{ bar } T_5 = 500^\circ\text{C}$$

$$\text{Interpolando } h_5 = 3481.7 \text{ KJ/Kg } s_5 = 7.9299 \text{ KJ/Kg}^\circ\text{C} \text{ -Punto 13}$$

$$P_{13} = 3 \text{ bar}$$

$$s_{13}' = s_5 = 7.9299 \text{ KJ/Kg}^\circ\text{C}$$

$$\text{Interpolando } h_{13}' = 3207.72 \text{ KJ/Kg}$$

$$h_{13} = 3481.7 - 0.9 \cdot (3481.7 - 3207.72) = 3235.118 \text{ KJ/Kg}$$

-Punto 6

$$P_6 = 0.06 \text{ bar}$$

$$s_6 = s_5 = 7.9299 \text{ KJ/Kg}^\circ\text{C}, s_g = 8.3304 \text{ KJ/Kg}^\circ\text{C}, s_f = 0.521 \text{ KJ/Kg}^\circ\text{C}$$

$$x = (s_6 - s_f) / (s_g - s_f) = (7.9299 - 0.521) / (8.3304 - 0.521) = 0.9487$$

$$h_6 = h_f + x \cdot h_{fg} = 151.53 + 0.9487 \cdot (2415.9) = 2443.49 \text{ KJ/Kg}$$

$$h_6 = 3481.7 - 0.9 \cdot (3481.7 - 2443.49) = 2547.31 \text{ KJ/Kg}$$

-Punto 7

$$P_7 = 0.06 \text{ bar}$$

$$h_7 = h_f = 151.53 \text{ KJ/Kg } y v_7 = 1.0064 \text{ cm}^3/\text{g}$$

$$w_{\text{bomba}} = v_7 \cdot (P_8 - P_7) = 1.0064 \cdot (3 - 0.06) / 10 = 0.2958 \text{ KJ/Kg}$$

-Punto 8

$$h_8' = h_7 + w_{\text{bomba}} = 151.53 + 0.2958 = 151.82 \text{ KJ/Kg}$$

$$h_8 = \eta_{\text{bomba}}(h_8 - h_7) + h_7 = 0.75(151.82 - 151.53) + 151.53 = 151.7475$$

-Punto 12

$$P_{12} = 3 \text{ bar}$$

$$h_{12} = h_f = 561.47 \text{ KJ/Kg y } v_{12} = 1.0732 \text{ cm}^3/\text{g}$$

$$w_{\text{bomba}} = v_{12}(P_1 - P_{12}) = 1.0732(140 - 3)/10 = 14.702 \text{ KJ/Kg}$$

-Punto 1

$$h_{1'} = h_{12} + w_{\text{bomba}} = 561.47 + 14.702 = 576.17 \text{ KJ/Kg}$$

$$h_1 = 0.75(576.17 - 561.47) + 561.47 = 572.72$$

-Punto 10

$$P_{10} = 15 \text{ bar } T_{10} = 198.3 \text{ }^\circ\text{C}$$

$$h_{10} = h_f = 844.84 \text{ KJ/Kg}$$

-Punto 11

$$h_{11} = h_{10} = 844.84 \text{ KJ/Kg}$$

-Punto 2

$$P_2 = 140 \text{ bar } T_2 = T_{10} = 198.3 \text{ }^\circ\text{C}$$

Como $T_2 < T$ dentro de la campana ($336.8 \text{ }^\circ\text{C}$) para la presión de 140 bar, indica que estamos en la región de líquido comprimido.

$$\text{De tablas } h_2 = 851.04 \text{ KJ/Kg}$$

Viendo los flujos de los calentadores:

Para el calentador cerrado planteamos :

$$1... m_9 = m_{10} = m_{11}$$

$$2... m_1 = m_2 = m_{12}$$

$$3... m_9 h_9 - m_{10} h_{10} = m_1 h_1 + m_2 h_2$$

Dividiendo 3 entre m_1

$$3'... y_g(h_9 - h_{10}) = (h_2 - h_1), \quad 3'... y_g(2994.56 - 844.84)/(851.04 - 572.72)$$

$$y_g = 0.1294$$

Para el calentador abierto plantamos:

$$4... m_{12} = m_{13} + m_8 + m_{11} \quad 5... m_3 = m_9 + m_4$$

$$6... m_4 = m_{13} + m_6 \quad 7... m_{12}h_{12} = m_{13}h_{13} + m_8h_8 + m_{11}h_{11}$$

Dividiendo 4 y 7 entre m_{12} (ver ecuación 2)

$$4'... 1 = y_{13} + y_8 + y_{11} \quad 4'... 0.8706 = y_{13} + y_8$$

$$5'... h_{12} = y_{13}h_{13} + y_8h_8 + y_{11}h_{11} \quad 5'...$$

$$561.47 = 3235.12 \cdot y_{13} + 151.74 \cdot y_8 + 1294 \cdot (844.84)$$

$$y_{13} = 0.1037 \quad y_8 = 0.7668$$

a)

$$q_{sum} = 1(h_3 - h_2) + (1 - y_9)(h_5 - h_4)$$

$$q_{sum} = (3591.1 - 851.04) + (1 - 0.1319) \cdot (3481.7 - 2894.94) = 3290.07 \text{ KJ/Kg}$$

$$w_{tur} = 1(h_3 - h_9) + (1 - y_9)(h_9 - h_4) + (1 - y_9)(h_5 - h_{13}) + (1 - y_9 - y_{13})(h_{13} - h_6)$$

$$w_{tur} = (3591.1 - 2994.56) + (1 - 0.1294) \cdot (2994.56 - 2849.94) +$$

$$(1 - 0.1294) \cdot (3481.7 - 3235.12) + (0.7668) \cdot (3235.1 - 2547.3)$$

$$w_{tur} = 1464.52 \text{ KJ/Kg}$$

$$w_{bomb} = (1 - y_9 - y_{13})(h_8 - h_7) + (h_1 - h_{12}) = (0.7668) \cdot (0.21) + 14.702 = 11.41$$

KJ/Kg

$$w_{neto} = 1464.52 - 11.41 = 1453.109 \text{ KJ/Kg}$$

$$\eta = 1453.109 / 3290.07 = 0.441$$

CAPITULO 4
MODELOS Y SIMULACION

4.1. PLANTEAMIENTO DEL PROBLEMA.

Podemos observar, del capítulo anterior, como se va complicando la solución de los problemas mientras más elementos se aumenten; la eficiencia del ciclo aumenta muy poco, pero la dificultad del análisis del mismo si se incrementa considerablemente.

Necesitamos poder plantear, resolver y analizar los datos de manera rápida, sencilla y efectiva. Dicha solución implica conocer las propiedades de los principales puntos que conforman el ciclo; tener una representación gráfica, como un diagrama Temperatura-Entropía (T-s) y, finalmente, calcular el trabajo neto del ciclo, así como el calor suministrado pero, sobre todo, su eficiencia térmica.

La propuesta de esta tesis es la elaboración de un programa para computadora, que con los datos básicos de entrada, sea capaz de proporcionarnos toda la información necesaria. En pocas palabras, hacer un modelo del sistema físico estudiado.

4.2. DEFINICION DE MODELO.

Un modelo es una representación de un objeto, sistema o idea, de forma diferente a la de la identidad misma. Usualmente su propósito es ayudar a explicar, entender o mejorar un sistema. Un modelo de un objeto puede ser una réplica exacta de éste o una abstracción de las propiedades dominantes del objeto.

Mucha gente piensa que el uso de modelos es algo reciente; sin embargo, la modelación no es nueva; la conceptualización y desarrollo de modelos han tenido un papel muy importante en la actividad intelectual de

la humanidad, desde que el hombre empezó a tratar de entender y manipular su medio. Este siempre ha usado la idea de los modelos para tratar de representar ideas y objetos; de hecho, el progreso y la historia de la ciencia y la ingeniería se reflejan con mayor precisión en el progreso de la habilidad del hombre para desarrollar modelos de fenómenos naturales, ideas y objetos.

Los modelos pueden tomar muchas formas, pero una de las más útiles, y ciertamente la que más se usa, es la matemática, la cual expresa por medio de un conjunto de ecuaciones las características esenciales del sistema o del fenómeno de estudio. En el caso de esta tesis, debido al gran número de cálculos realizados, las matemáticas son pieza fundamental dentro del modelo.

Finalmente, el uso de modelos hace posible la experimentación controlada en situaciones en que los experimentos directos serían imprácticos o prohibitivos por su costo. En nuestro caso, el modelo nos ayudará como una herramienta didáctica en el estudio de ciclos de vapor. Al resolver varios ciclos, el usuario podrá ver las distintas eficiencias en cada caso y seleccionar el ciclo que considere más adecuado.

4.3. HISTORIA DEL DESARROLLO DE LA SIMULACION.

Varias generaciones de paquetes de simulación se han desarrollado a partir de la creación de la primera computadora electrónica; con cada generación se han podido modelar sistemas cada vez más complejos.

Gracias al desarrollo de lenguajes de alto nivel como el FORTRAN, se ha tenido la base necesaria para hacer modelos de sistemas biológicos, físicos, mecánicos, eléctricos, electrónicos etc. Durante esta

etapa, los modelos se representaban por medio de ecuaciones diferenciales.

Durante los años 80, las compañías que desarrollaban los paquetes de simulación decidieron reescribir su **software** para estaciones de trabajo (**workstations**) aprovechando las ventajas y la capacidad gráfica inherentes en esta plataforma.

Posteriormente los modelos comenzaron a especificarse mediante diagramas de bloques, cada bloque era un conjunto de procesos unidos. Los procesos eran normalmente definidos en notación matemática de transformada de Laplace, ecuaciones diferenciales, matrices etc.

Estos bloques a su vez podían juntarse con otros para obtener unos "superbloques", lo cual permitió el desarrollo de un gran número de subsistemas, y con esto, un gran número de librerías para cada grupo de usuarios.

En la Programación Orientada a Objetos (OOP) se permite definir elementos con características y cualidades propias; las funciones se vuelven más generales, más sencillas de modificar y más compactas. Este fue el siguiente nivel de abstracción.

Hasta ahora, el último paso de la simulación consiste en trabajar junto con el **software** para crear modelos. Usar la información capturado durante el diseño y simular el proceso para hacer pruebas y prototipos.

Pese a los éxitos obtenidos, la evolución continúa con el fin de ayudar a los usuarios a desarrollar mejores modelos, más eficientes y más económicos.

4.4. VENTAJAS Y DESVENTAJAS DEL USO DE LA COMPUTADORA.

A continuación se presenta una pequeña lista con los atributos asociados a las computadoras:

- 1) Solamente una, o un número limitado de operaciones se llevan a cabo simultáneamente.
- 2) La precisión es relativamente independiente de la calidad de los componentes de la computadora y está principalmente determinado por el tamaño de los registros de memoria, el número de dígitos significativos que se usan, y las técnicas numéricas seleccionadas.
- 3) Los tiempos de solución pueden ser algo prolongados y están determinados por la complejidad del problema.
- 4) Operaciones aritméticas son posibles (sumas, restas, multiplicaciones y divisiones); sin embargo operaciones más complejas como diferenciales e integrales deben hacerse a través de métodos numéricos.
- 5) Los datos numéricos y no numéricos se memorizan indefinidamente.

4.5. LENGUAJES DE SIMULACION.

Un lenguaje de programación de computadora es un conjunto de símbolos reconocibles por una computadora, mediante el cual un programador le da instrucciones a ésta con referencia a las operaciones que lleva a cabo.

Dentro de los lenguajes para el desarrollo de modelos, podemos distinguir dos grandes grupos:

- Lenguajes de propósito general.
- Lenguajes de propósito especial.

Los lenguajes de propósito general como COBOL, FORTRAN, BASIC, PASCAL etc., se diseñan para resolver una extensa variedad de problemas. Por otra parte, los lenguajes de propósito general, tales como GPSS, BHSL, Dynamo etc., se diseñan para satisfacer o resolver una clase o tipo particular de problemas.

Desgraciadamente se han desarrollado tantos lenguajes de programación de ambos grupos, que es casi imposible decir que lenguaje se ajusta mejor o casi se ajusta a una aplicación particular; en consecuencia, el procedimiento usual es adoptar un lenguaje conocido por el analista, no porque sea el mejor sino porque se conoce. Cualquier lenguaje algorítmico general puede expresar el modelo deseado; sin embargo uno de los lenguajes especializados puede tener ventajas muy diferentes en cuestión de facilidad, eficiencia y efectividad.

El programa que se describirá en el siguiente capítulo fue desarrollado en un lenguaje de propósito general: PASCAL de Borland, versión 6.0, esto debido a que era el lenguaje que conocía, y que sabía que contaba con las herramientas necesaria para desarrollar un buen proyecto. No era conveniente buscar y tratar de aprender otro lenguaje si se podía trabajar con PASCAL. Además de que el programa no necesita una máquina especial para correrse, puede hacerse en cualquier computadora personal, en algunas más rápido que en otras, pero en todas debe funcionar.

El trabajar con un lenguaje como PASCAL presenta algunas ventajas y desventajas como las siguientes:

- 1) Número mínimo de restricciones impuestas al formato de salida.
- 2) Da bastante flexibilidad para trabajar en el diseño, implementación y uso del modelo. Sin embargo, esta flexibilidad se obtiene a expensas de

mayor dificultad de programación.

3) El tiempo de programación es largo pues hay que programar todas las subrutinas y la utilería que necesite el programa.

Algunas veces los lenguajes de programación general como, PASCAL, son la única alternativa disponible para el analista que no tiene acceso a una computadora de gran capacidad, debido a que la mayoría de los lenguajes de propósito especial existentes se escribieron para usarse en máquinas de gran capacidad.

Para finalizar este capítulo, únicamente mencionaremos que además de las técnicas para los lenguajes de computadoras digitales, como los descritos anteriormente, existen otras dos:

1. Las técnicas para computadoras analógicas electrónicas, las cuales representan las variables de un problema por medio de cantidades físicas que se generan o se controlan fácilmente como voltajes eléctricos. Desarrolla una solución mediante operaciones simultáneas, lo cual las hace mucho más rápidas que las digitales en caso de resolver sistemas de ecuaciones diferenciales.

2. Las técnicas de computadoras híbridas; dicha computadora puede ser uno de varios sistemas posibles, pero el más común es la combinación de una computadora digital de propósito general, una computadora analógica y una interfase o enlace.

4.6. SOFTWARE EXISTENTE.

Actualmente existen varios paquetes, diseñados por diversas compañías, que son capaces de simular la operación de cualquier planta energética. A continuación se presentan algunos de esos programas existentes en el mercado.

GATE/CYCLE.

Diseña y analiza cualquier sistema de plantas de potencia, incluyendo las de combustibles fósiles, las nucleares y las que utilizan ciclos combinados. Calcula datos que no puedan ser medidos. Cuentan con una librería muy amplia de los distintas marcas de los equipos de que consta el ciclo y que existen en el mercado (Turbinas de gas y de vapor, Condensadores, Evaporadores, Intercambiadores de Calor etc). Permite a su vez, estudiar el comportamiento térmico de cada elemento del ciclo.

Es fabricado por la compañía Enter y tiene un costo aproximado de 6,000 dolares.

PC - BOILER.

Simulación del domo de una caldera con combustible fósil. Opera en sistema británico o en SI. Cuenta con una excelente documentación y está provista con varios ejemplo. Puede correrse en una IBM PC o compatible con coprocesador matemático y tarjeta de gráficos EGA o VGA.

Fue desarrollado por la compañía SIMTECH Services y tenía un costo aproximado de 3,995 dólares en 1991.

MMS - B&W.

Contiene una intensa librería de componentes de plantas fósiles y nucleares, esto permite hacer diversas simulaciones para el control y el perfeccionamiento de dichas plantas. Puede correrse en una IBM AT, PS/2 o compatible, pero necesita un compilador especial.

Este paquete fue desarrollado por B&W Nuclear Service Company Engineering & Plant Service Division. El costo del equipo es según requerimientos.

CAPITULO 5
DESCRIPCION DEL PROGRAMA

CETTRAN.

Simulador para plantas de potencia de combustible fósil o nuclear. Necesita para poder correrse una VAX o una DEC

El simulador fue desarrollado por la ABB Power Automation Inc. El costo del programa es negociable.

EASY5x.

Paquete para simular sistemas mecánicos, hidráulicos, químicos, eléctricos, neumáticos, térmicos etc. Cuenta además con una interfase gráfica lo cual da una excelente resolución durante la construcción y el análisis de sistema a simular. Requiere una IBM RS6000 UNIX Workstation.

El paquete fue fabricado por la compañía Boing y tenía un costo aproximado de 5,688 dólares en 1991.

5.1. COMPORTAMIENTO DE LOS ELEMENTOS QUE CONFORMAN EL CICLO.

El primer punto necesario para la realización de este proyecto, fue el conocer la naturaleza de los elementos que conforman el ciclo, refiriéndonos principalmente al estado termodinámico en que entra y sale el fluido de trabajo de dichos elementos.

5.1.1 Turbina.

En la entrada tenemos vapor sobrecalentado, puede darse el caso de tener una mezcla de líquido y vapor, pero no es lo común por ser ciclos poco eficientes; la excepción es el ciclo de Carnot, pero éste, como ya se mencionó anteriormente, es un ciclo ideal.

La Turbina puede tener varias salidas o sangrados, ya sea a Calentadores Abiertos o Cerrados, al Condensador o para alguna etapa de recalentamiento. Para todos los casos, con excepción de la salida al Condensador, se desea tener vapor sobrecalentado; para este último se desea tener una mezcla líquido vapor, pero muy cerca de la región de vapor saturado.

En el caso ideal, la entropía permanece constante durante todas las salidas de la turbina. Si existe recalentamiento, la entropía es constante desde la salida de la Caldera hasta la etapa de recalentamiento, cambia a un nuevo valor y vuelve a ser constante desde el recalentamiento hasta la salida final al Condensador.

5.1.2. Condensador.

La entrada al condensador es generalmente a temperaturas entre 25°C y 45°C, con el fluido dentro de la región de la campana. Para dichas temperaturas tenemos rangos de presión entre 0.03 y 0.1 Bars.(1)

A la salida, debemos tener el fluido como líquido saturado.

El Condensador trabaja a presión constante.

5.1.3. Bombas.

Las bombas se colocan siempre a la salida del Condensador o de Calentadores, para casos prácticos se considera que el agua sale como líquido saturado, de ahí, que para la entrada a las bombas se toman estos valores.

Al observar un diagrama T-s, vemos que los puntos a la salida de las bombas se encuentran en la región de líquido comprimido.

Las bombas dan un incremento de una presión baja (P_1) a una alta (P_2).

5.1.4. Calentador Abierto.

A la entrada del calentador abierto podemos tener los 3 estados termodinámicos distintos estudiados en estos ciclos. Se tiene una entrada de vapor sobrecalentado que viene de la Turbina; podemos tener también la entrada del flujo que viene de una Bomba como líquido comprimido, y, en algunos casos, la entrada de flujos que vengan de una Trampa de estrangulación que viene como una mezcla de líquido y vapor.

A la salida de estos elementos tenemos siempre, idealmente, líquido saturado.

Los Calentadores Abiertos trabajan a una presión única.

5.1.5. Calentadores Cerrados.

Los Calentadores Cerrados tienen dos entradas y dos salidas de flujos. Una entrada viene de un sangrado de la Turbina como vapor sobrecalentado; la otra entrada viene del resto del sistema, que puede ser el flujo saliente de una bomba o de otro calentador cerrado, en ambos casos tenemos líquido comprimido .

En cuanto a las salidas, la que viene del sangrado de la Turbina puede mandarse a una Trampa de estrangulamiento o a otra Bomba, en ambos casos el agua sale como líquido saturado. La otra salida es la que viene del sistema, en este punto, el agua sale como líquido comprimido.

En los Calentadores Cerrados ambas salidas tienen idéntica temperatura; mientras que las presiones de entrada sí son distintas.

5.1.6. Trampas.

Las Trampas de estrangulamiento se colocan en la salida de los Calentadores Cerrados que viene del sangrado de la Turbina; después del intercambio de calor, que en ellos ocurre, el agua entra a la Trampa como líquido saturado.

Las Trampas regresan el agua al Condensador o a un Calentador de menor presión; para los casos aquí aplicados, los flujos de las trampas se regresarán a un Calentador Abierto. Los puntos, en esta parte, se encuentran nuevamente dentro de la campana.(2)

A la salida de los sistemas de estrangulamiento, pese a la reducción de presión, se mantiene la misma entalpía.

5.1.7. Caldera.

En la Caldera, el fluido entra como líquido comprimido y sale a la Turbina como vapor sobrecalentado.

Todo el proceso ocurre a presión constante.

5.2. ESTRUCTURA GENERAL DEL PROGRAMA.

El proyecto que se presenta en esta tesis consta de:

1. Un programa principal llamado CPOTEN.EXE, el cual controla todas las rutinas utilizadas. Este programa será examinado detenidamente a lo largo de este capítulo.

2. Seis unidades, tres estándares de PASCAL (CRT, PRINTER y GRAPH), y tres elaboradas especialmente para este caso; dichas unidades serán descritas posteriormente.

3. Tres archivos que contienen las tablas de vapor de agua; dichos archivos son los siguientes:

- a. VAP_SCAL.UP Contiene las tablas de vapor sobrecalentado.
- b. CAMPANA.UP Contiene las tablas de la campana.
- c. LIQ_COMP.UP Contiene las tablas de líquido comprimido.

Las tablas fueron tomadas de: KEENAN Joseph, KEYES Frederick, HILL Philip, MOORE Joan, Steam Tables, USA, Ed. John Wiley & Sons Inc, 1969.

5.3. UNIDADES AUXILIARES DEL PROGRAMA.

El programa principal utiliza tres unidades para la solución de los ciclos. Dichas rutinas son usadas durante la corrida del programa, de ahí

que el usuario nunca las utiliza directamente. A continuación se describen brevemente cada una de ellas.

5.3.1. Archiv.TPU

Esta unidad es la más importante de todas, pues es la encargada de la búsqueda de temperatura, presión, volumen específico, entalpía específica y entropía específica, en los archivos de las tablas; además calcula calidades y propiedades de la sustancia dentro de la campana.

La unidad consta de 4 funciones:

1. FUNCTION Busca_campana. La cual busca valores en la región de la campana y entrega el valor numérico de la propiedad deseada. Para utilizar estas tablas necesitamos conocer un dato de entrada, la función necesita 3 parámetros:

- a) Clave del dato conocido.
- b) Clave del dato buscado.
- c) Valor del dato conocido.

Las claves de a) y b) son las siguientes:

p = presión.

t = temperatura.

hf = entalpía de líquido saturado.

hg = entalpía de vapor saturado.

sf = entropía de líquido saturado.

sg = entropía de vapor saturado.

vf = volumen de líquido saturado.

vg = volumen de vapor saturado.

2. FUNCTION Busca_dato. Dicha función busca valores en la región de vapor sobrecalentado. Para poder usar estas tablas necesitamos conocer dos datos de entrada. La rutina ahora necesita 4 parámetros:

- a) Valor de la presión de entrada.
- b) Clave del dato conocido.
- c) Clave del dato buscado.
- d) Valor del dato conocido.

Las claves para esta función son:

- p = presión.
- t = temperatura.
- h = entalpía.
- s = entropía.
- v = volumen.

3. FUNCTION Busca_liqs. Esta función busca los datos en la región de líquido comprimido, el uso de las tablas es similar al caso anterior, de ahí que utiliza los mismos parámetros y las mismas claves.

4. FUNCTION Calidad. Puede hacer dos cálculos distintos:

- a) La calidad del vapor dentro de la región de la campana.
- b) Hallar el valor de volumen, entalpía o entropía dentro de la campana dada una calidad.

Para ambos casos se necesitan 4 parámetros:

- a) Número Clave para Saber qué se va a Calcular
 - 1. Calcula calidad.
 - 2. Calcula el valor de una propiedad.
- b) Valor Conocido

Para el caso de número clave = 1 es h, s o v.

Para el caso de número clave = 2 es la calidad.

- c) El valor de la propiedad como líquido saturado.
- d) El valor de la propiedad como vapor saturado.

5.3.2. Elemento.TPU

En esta unidad se definen:

1. Los 4 tipos de listas que se usarán durante la corrida del programa. Dichas listas son:

- a) Lista de puntos. Guarda la información de número, presión, temperatura, entalpía, entropía y volumen de cada punto del ciclo.
- b) Lista de flujos. Almacena las fracciones del flujo que salen a los Calentadores y al Condensador, en el caso de ciclos Regenerativos.
- c) Lista de Calentadores cerrados. Guarda la información de las entalpías de entrada y salida de los Calentadores cerrados, para poder hacer los balances térmicos necesarios.
- d) Lista de presión. Aquí se almacena la información de presiones de entrada y salida de los objetos Bombas, Trampas, Calentadores Abiertos y Calentadores Cerrados.

Los tres primeros casos son listas de Registros, mientras que la última es una lista de Objetos.

2. Los objetos Presión, Turbina y Condensador.

- a) Objeto Presión. Se utiliza este objeto para capturar los datos de presiones de entrada y salida para Bombas, Trampas, Calentadores Cerrados y Calentadores Abiertos. Contiene además los procedimientos para inicializarse, para calcular las condiciones de líquido saturado y para reportar la información que posee.

El objeto se inicializa con las presiones de entrada y salida.

b) Objeto Turbina. Contiene la presión y temperatura de entrada a la Turbina, así como la entropía. Además de los procedimientos para inicializarse y reportar su información, tiene las rutinas para calcular entalpía, volumen y temperatura para cualquier presión de salida, ya sea como vapor sobrecalentado o como mezcla de líquido vapor. El objeto además calcula, por sí mismo, el valor de su entropía constante.

El objeto se inicializa con la presión y la temperatura de entrada.

c) Objeto Condensador. Guarda la información de presión y temperatura de entrada, la cual toma como datos, y obtiene los datos de entalpía, entropía y volumen de salida. Sus procedimientos son los de inicializarse, cálculo de las propiedades de salida y reportar la información que posee.

El objeto se inicializa con la presión de entrada.

3. Los procedimientos necesarios para crear listas de Calentadores cerrados, colocar datos en dicha lista y reportar los valores de entalpías de estas listas.

5.3.3. Sammy.TPU

En esta unidad tenemos las rutinas y los objetos gráficos necesarios para dibujar los ciclos. Esta unidad consta de:

1. Definición de los objetos para dibujar Turbinas, Bombas, Trampas, Calentadores, Condensador y Calderas.

Se define primeramente un objeto punto, con las propiedades de dibujarse, borrarse y moverse. Este objeto será el padre de todas las demás figuras y estará representado por una pequeña cruz.

La Turbina será un trapecio, los calentadores y la caldera serán rectángulos, y las bombas y trampas círculos.

2. Procedimiento `Dibuja_def`. Da un esquema muy sencillo para ciclos de Carnot y Rankine con y sin recalentamiento, pero sin regeneración. Originalmente se había diseñado de tal manera de tener un dibujo más completo, pero por limitaciones de la memoria de PASCAL, se quedaron los esquemas más simples.

En este procedimiento los parámetros que se dan son dos variables lógicas (`f` y `calen`)

- a) Si `f` es `TRUE`, el ciclo es de Carnot, en caso contrario, el ciclo es Rankine.
- b) Si `calen` es `TRUE`, existe recalentamiento.

3. Procedimiento `Dibuja_simul`. Esta rutina es la encargada de construir la planta para resolver ciclos Rankine regenerativos; aquí es donde se utilizan todos los objetos anteriormente mencionados. Después de colocar los dibujos de los diversos elementos, se procede a unirlos por medio de líneas.

Este procedimiento necesita de varios parámetros para ejecutarse, a continuación se mencionan:

- a) Número de presiones de salida por arriba de la de recalentamiento, si es que éste existe.
- b) Número de Calentadores Cerrados.
- c) Número de Calentadores Abiertos.
- d) Número de Bombas.
- e) Número de Trampas de estrangulación.
- f) Si existe recalentamiento.
- g) Número de puntos. Esta variable entra al procedimiento con el valor de cero, y al terminar el dibujo, regresa el número de puntos de los que consta el ciclo.

4. Procedimiento Óptimo. Al igual que para el caso de los ciclos Rankine y Carnot, este proceso presenta un esquemas sencillo del caso de un ciclo óptimo. Este ciclo consta un número predefinido de elementos y de puntos, su justificación se explicará cuando se analice el programa en el siguiente punto de la tesis.

5.4. DESCRIPCION DEL PROGRAMA.

Para finalizar este capítulo, analizaremos el programa. A partir del menú principal iremos viendo todas las opciones, las llamadas a rutinas auxiliares y los mensajes de error que pueden llegar a aparecer.

El menú principal esta formado por 8 opciones:

- a. Iniciar un Nuevo Ciclo.
- b. Cargar un Reporte Existente.
- c. Editar Parámetros.
- d. Solución Numérica.
- e. Solución Gráfica.
- f. Dar Trabajos.
- g. Potencia de Turbina.
- h. Salir.

5.4.1. Iniciar un nuevo ciclo.

Al seleccionar esta opción, se llama primeramente a la rutina Portada, en la cual se escoge uno de los 4 ciclos presentados que el programa puede resolver. Estos ciclos son:

1. Ciclo de Carnot.
2. Ciclo Rankine sencillo.

3. Ciclo Rankine regenerativo.

4. Ciclo Rankine óptimo.

La rutina esta ciclada hasta que el usuario este seguro del ciclo que desee resolver.

Dependiendo del ciclo que se desea resolver, pasamos a la parte de captura de datos. Todas las rutinas son similares en el sentido de que primero se capturan datos, después vemos un esquema del ciclo y, finalmente, regresamos al menú principal. Las variantes radican en los datos que se preguntan para realizar los cálculos del ciclo.

1. Ciclo de Carnot. La pantalla de captura es muy sencilla, únicamente se preguntan las presiones de entrada a la Turbina y al Condensador. El programa calcula posteriormente las temperaturas de entrada al Condensador y a la Turbina para poder inicializar los objetos.

2. Ciclo Rankine sencillo. En este caso se preguntan la presión y la temperatura de entrada a la Turbina, si existe recalentamiento, en caso afirmativo la presión y temperatura de recalentamiento, y por último, la presión de entrada al Condensador. Finalmente el programa calcula la temperatura del Condensador.

3. Ciclo Rankine regenerativo. Esta rutina consta de 4 pantallas. En la primera se pregunta lo mismo que en el Rankine sencillo. En la segunda se pide el número de Calentadores Cerrados, Calentadores Abiertos, Bombas y Trampas de estrangulamiento que se usarán. La tercera pantalla captura las presiones para cada uno de estos elementos; dichos datos son guardados en 4 variables distintas del tipo Objeto PresiónPtr, ya antes descrito, estas listas son: Basecc, Baseca, Baseb, Baset. Para las

Bombas y Trampas la presión 1 es la presión de entrada y la presión 2 es la presión de salida; en los Calentadores Cerrados P1 y P2 son iguales, pues como ya se menciono, trabajan a presión única, así que sólo se captura el valor de P1, y el programa automáticamente inicializa el valor de P2; para los calentadores Cerrados, la presión 1 es la presión a la que llega el vapor de la Turbina, mientras que la presión 2 es la que viene del ciclo.

Finalmente, la última pantalla nos presenta las siguientes opciones:

1. Editar C. cerrados.
2. Editar C. abiertos.
3. Editar Bombas.
4. Editar Trampas.
5. Mostrar C. cerrados.
6. Mostrar C. abiertos.
7. Mostrar Bombas.
8. Mostrar Trampas.
9. Seguir adelante.

Las cuatro primeras opciones son para editar algún dato, las cuatro siguientes son únicamente para ver información pero no se pueden alterar nada, la última opción es para pasar a la construcción del esquema del ciclo. Un dato importante que se calcula en este momento es el número de salidas de presión mayores a la de recalentamiento en la variable sumpto; si no existe recalentamiento, el valor de sumpto es cero.

La opción de ciclo Rankine regenerativo es la única que no trae un dibujo predefinido, sino que, con los datos anteriormente ingresados, el usuario hará su propio diagrama. Los dibujos salen de la esquina superior izquierda de la pantalla, la Turbina y el Condensador tienen sus formas

características; las bombas y trampas son ambas círculos, así que para diferenciarlas las bombas traen abajo la leyenda BOMB, mientras que las trampas dicen TRAM; la Caldera y los Calentadores son rectángulos, aquí la diferencia radica en que la Caldera es más grande y su base es uno de los lados cortos, mientras que los Calentadores están apoyados en su lado mayor, para diferenciar a los Calentadores Cerrados de los Abiertos los primeros tiene una leyenda que dice CC, mientras que la de los segundos es CA.

Después de colocar los objetos a lo largo de la pantalla, la rutina une y asigna números a la Turbina con todas sus entradas salidas, en el caso de recalentamiento la línea que une a la Turbina con la Caldera tendrá 3 números. La asignación de números es de mayor a menor presión de sangrado, de ahí que la primera salida de la Caldera a la Turbina es siempre el número 1.

La última parte de esta rutina gráfica es acabar de unir los demás elementos; con la tecla F1, aparece el cursor en la parte superior izquierda de la pantalla para unir dos objetos, se mueve uno con las flechas de teclado y al llegar al objeto deseado se teclee ENTER, inmediatamente aparece otro cursor donde estaba originalmente el primero para que se dirija al segundo objeto, al llegar a éste, se aprieta ENTER nuevamente, automáticamente quedan unidos ambos objetos y se les asigna un número de punto dentro del ciclo. La unión de estos puntos debe hacerse de menores a mayores presión, y dejando siempre las salidas de Calentadores Cerrados que vienen de la Turbina hasta el último.

Cuando se acaba de hacer el diagrama con F3 se regresa al menú principal.

4. Ciclo Rankine Optimo. Primeramente mencionaremos las

consideraciones tomadas para dicho ciclo.

Los ciclos más óptimos son aquéllos que cuentan con una combinación de recalentamiento y regeneración; la combinación de los dos ciclos conservará las ventajas de cada uno de ellos y no se producirán desventajas adicionales.(3)

El número de calentadores comúnmente usados en una planta media, varía entre uno y cinco, observando una gráfica de número de etapas de extracción vs. Reducción en consumo de calor en % (4), vemos que el incremento de la eficiencia del ciclo con 4 etapas de extracción se vuelve asintótico, es decir, ya no se aprecia un incremento de eficiencia que justifique el costo de otra etapa de extracción. Lo común es utilizar únicamente un Calentador Abierto, mientras que los demás deberán ser cerrados.(5)

La manera de determinar las presiones de extracción para el ciclo óptimo es la siguiente:

De la presión de entrada a la Turbina se lee su valor de temperatura dentro de la campana, se hace lo mismo con la presión del calentador. Posteriormente se divide el intervalo de temperaturas entre el número de etapas de extracción que se desean, asignándoseles la presión correspondiente, siempre dentro de la campana, a las temperaturas obtenidas.(6)

Finalmente se optó por usar Trampas de estrangulación a la salida de los Calentadores Cerrados por ser más económicas que las bombas, y por no causar un trabajo que influya en el cálculo de trabajo neto del ciclo.

Tenemos así :

1 Calentador Abierto.

3 Calentadores Cerrados.

2 Bombas.

3 Trampas de estrangulación.

Se pregunta la presión y la temperatura de entra a la Turbina, se supone que existe recalentamiento, por lo que se pide también la presión y temperatura de recalentamiento, y por último, la presión de entrada al Condensador.

El programa calcula inmediatamente la temperatura del Condensador, las presiones de sangrado para los Calentadores y el Condensador, posteriormente presenta los valores asignados a cada uno en 4 pantallas distintas, la primera para las Bombas, la segunda para las Trampas, la tercera para los Calentadores Cerrados y la cuarta par Calentadores Abiertos. Por último se presenta el esquema del ciclo.

Çabe destacar además:

- Existen límites para las entradas de presiones y temperaturas, dichos límites se refieren principalmente a las restricciones metalúrgicas de los elementos y se presentan en la parte superior de las pantallas de captura.
- Al final de cada pantalla, se presenta la opción de editar los datos oprimiendo la tecla E, en caso de hacerse esto, se llama primeramente al procedimiento MueveXY para poder ubicarnos en cualquier punto de la pantalla, seleccionamos la cifra que deseemos editar oprimiendo ENTER, así llamamos a la rutina de edición, la cual nos permite cambiar datos. Al finalizar la edición oprimimos ESC, terminando esta rutina con un mensaje de Edición terminada.
- Al finalizar cualquiera de los cuatro ciclos antes mencionados, se inicializan los objetos TurbinaVar, CondensadorVar y TurbinaBaja, si este último es necesario. Dicha inicialización es después de que se editaron los datos que fueron necesarios.

- Tanto las rutinas de captura y edición están validadas de tal forma que no sea posible ingresar datos que no sean numéricos, es decir, no se pueden escribir espacios vacíos, letras o caracteres extraños.
- La última instrucción de este ciclo es inicializar los parámetros con sus valores predefinidos, de esto se hablará posteriormente.

5.4.2. Cargar reporte existente.

Opción para ver reportes previamente creados. Todo reporte constan de dos archivos con idéntico nombre, pero distinta extensión. Los archivos *.DAT tienen los datos de eficiencia térmica, número de calentadores etc.; los archivos *.CPT tienen la lista de los puntos con sus propiedades.

Se pregunta el nombre del archivo que se va a cargar, si no existe el archivo, manda un mensaje de error y regresa al menú principal; si existe el archivo, lee primero el *.DAT, posteriormente lee el *.CPT colocando los datos en la pantalla.

Después de ver los resultados, se pregunta si se desea sacar una impresión del reporte.

Finalmente se pregunta si se desea ver el diagrama T-s. Si se responde afirmativamente, se corre la rutina que dibuja la campana, posteriormente la encargada de colocar los puntos del ciclo en el dibujo.

Cabe destacar:

- Sólo se presentan reportes, la información presentada no se puede modificar.
- En caso de que se de el nombre de un reporte que no existe, se manda un mensaje de error y se regresa al menú principal.

5.4.3. Editar parámetros.

El programa maneja cuatro parámetros, dos para el cálculo de los puntos del ciclo, y dos para la potencia de la Turbina. Estos parámetros son:

1. Eficiencia de la Turbina. Su valor predefinido es 100%
2. Eficiencia de las Bombas. Su valor predefinido es 100%
3. Cp del agua de alimentación. Su valor predefinido es de 4.1818 KJ/Kg°C.
4. Incremento de temperatura en el agua del condensador. Su valor predeterminado es 8°C.

Cabe destacar:

- Resulta muy ilustrativo resolver un ciclo con las condiciones ideales, ver su eficiencia térmica y compararlo después con un ciclo real para comparar resultados. Por la manera como está estructurado el programa esto es muy fácil de hacer, ya que siempre que se corra la rutina de resultados, se recalculan los valores con los nuevos parámetros.
- Siempre se debe correr esta rutina después de inicializar un ciclo y antes de ver la solución numérica.
- Siempre que se corra un ciclo nuevo, los valores de los parámetros se inicializan.

5.4.4. Solución numérica.

La rutina consta de 5 partes:

1. Primeramente revisa si la lista PuntosPtr, que es la que guarda la información de los puntos, esta vacía, si no la limpia con el procedimiento Vacía_puntos.
2. Dependiendo del ciclo escogido, es la rutina que llama.

- Procedure Carnot. Resuelve ciclos ideales. No se puede incluir en otro procedimiento por la naturaleza especial como se comporta este ciclo.
- Procedure Rankine sencillo. Resuelve ciclos sin etapas regenerativas, puede resolverse también con la opción de Rankine regenerativo, pero esta rutina es mucho más rápida.
- Procedure Rankine regenerativo. A diferencia de los otros tres ciclos, que ya tienen sus procedimientos predefinidos, en este vamos a ir calculando todo por partes.

Primero, se crea una lista vacía con el total de puntos de que va a constar el ciclo. Si existe recalentamiento, se calculan los puntos de éste, de la rutina del dibujo de esquema, ya sabe el programa a qué puntos corresponde este recalentamiento. Enseguida calcula todas las propiedades de los sangrados de la Turbina empezando con los de mayor presión hasta llegar al Condensador. Se pasa después a calcular los puntos a las salidas de los Calentadores Abiertos, si es que existen, de mayores a menores presiones. Se hace lo mismo con las salidas de los Calentadores Cerrados, primero las que vienen del ciclo y después las que vienen del sangrado de la Turbina. Posteriormente se calculan las salidas de todas las bombas y los llamados puntos finales (-1 y -2); el por qué de dichos puntos se mencionará en la parte de la solución gráfica. La rutina acaba calculando los puntos de salida de las Trampas cuando éstas existen.

Esta misma rutina se utiliza para los ciclos óptimos, con todos los datos de entrada ya definidos durante la rutina captura.

3. Se llama al procedimiento Despliegue para que presente los resultados obtenidos.

4. Se calcula el trabajo de la Turbina, trabajo de las Bombas, el trabajo neto, el calor suministrado y la eficiencia térmica por medio del `procedure Eficiencia`.

Para los casos regenerativos y óptimo, es necesario calcular antes las fracciones de flujo que van a cada elemento. Esto se hace mediante el procedimiento `Flujos`, el cual hace un balance energético en los calentadores, primero en los Abiertos y después en los Cerrados.

Se guarda la información de cada flujo en una lista del tipo `FlujosPtr`.

Si el balance no se puede hacer por el sentido de los flujos, manda un mensaje de error, diciendo que el ciclo es imposible y avisa que para continuar, tomará el valor absoluto de los flujos.

Para los cálculos de trabajos y calores se usan las ecuaciones vistas en los ejemplos del capítulo 2.

5. Pregunta si se desea generar un reporte. En caso afirmativo, se crean dos archivos con la siguiente información:

a) *.DAT: nombre del ciclo, número de Calentadores Cerrados, Abiertos, Bombas y Trampas, trabajo neto, calor suministrado, eficiencia térmica, eficiencia de la Turbina, eficiencia de las Bombas y si existe o no recalentamiento.

b) *.CPT: Número del punto, presión, temperatura, entalpía, entropía, volumen y cómo están conectados los elementos.

En caso de existir algún error en el disco, como el que esté protegido contra escritura, el programa manda un mensaje de error y pregunta si se desea reintentarlo, en caso afirmativo trata de volver a crear el reporte en la misma unidad, en caso negativo, nos regresa al menú principal.

Cabe destacar:

- Independientemente del ciclo que se resuelva, todos los resultados se almacenan en una lista del mismo tipo, además todos utilizan las rutinas: Cabeza_puntos, para calcular el punto 1; Añade_punto, para sumar puntos a la lista; Vacía_punto, para colocar los datos en la lista.
- Aunque el problema se halle ya resuelto, siempre que se llama esta rutina, se recalcula el ciclo, para poder hacer variaciones de los parámetros.
- Correr siempre esta rutina antes de querer ver soluciones gráficas, trabajos o potencia de turbina, pues estos tres procedimientos necesitan forzosamente los datos que proporciona la rutina de Solución numérica.
- En caso de que exista algún error con los datos de entrada (que algún punto quede fuera de las condiciones de operación) aparece un mensaje de error indicando si está fuera de rango superior o inferior de la parte de líquido comprimido o de la de vapor sobrecalentado.

5.4.5. Solución Gráfica.

La rutina reconoce primeramente qué tipo de monitor se utiliza y qué tarjeta de video tiene la máquina en la que se está corriendo. En base a esto calcula la escala y las dimensiones del gráfico.

Presenta el diagrama T-s del ciclo.

Aunque los puntos -1 y -2 no son necesarios para hallar la eficiencia del ciclo, sí lo son para hacer un buen dibujo, pues dan los puntos donde se 'cierra' la gráfica. El punto -1 tiene las propiedades de líquido saturado a la presión de entrada de la Turbina, mientras que el pto -2 da el vapor saturado a la misma presión.

Según la teoría, las Trampas son líneas rectas que caen dentro de la región de la campana, por esto no están en la misma lista de los demás

puntos.

Los puntos a la izquierda de la campana son muy parecidos, por tanto siempre se ven encimados al momento de construir el diagrama.

Un punto interesante a observar es el comportamiento del ciclo cuando la eficiencia de Turbina y Bombas es distinta a 100%.

5.4.6. Dar trabajos.

Presenta los resultados obtenidos en la rutina Eficiencia obtenidos en la rutina Solución Numérica. Los datos aquí obtenidos se calculan durante la rutina de Solución Numérica.

5.4.7. Potencia de Turbina.

Para un flujo dado de vapor de agua en Kg/s, calcula la potencia de la Turbina y el flujo de agua de enfriamiento necesario para mantener el incremento de temperatura definido en los parámetros.

Como ya se mencionó antes, es necesario correr primero la rutina de Solución Numérica.

Para los resultados que se encuentran en esta rutina se consideran también las fracciones de flujo.

5.4.8. Salir.

Rutina para finalizar el programa y regresamos al sistema operativo.

NOTAS DEL CAPITULO 5.

(1) cfr. WARK Kenneth, Termodinámica, México, Ed. McGraw Hill, 1988, 4^a edición, p. 684.

(2) cfr. KADAMBI V. Conversión de energía, México, Ed. Limusa, 1984, 1^a edición, p. 227.

(3) cfr. MORSE T. Frederick. Centrales Eléctricas, México, Ed. CECSA, 1981, 9^a edición, p.344

(4) cfr. ibídem.

(5) cfr. KADAMBI V. op.cit, nota (2), p. 225.

(6) cfr. MORSE T. Frederick. op.cit, nota(3), p. 334

CONCLUSIONES

CONCLUSIONES.

Al final de este trabajo vemos que el uso de la herramienta computacional para resolver ciclos de generación de potencia por medio de vapor, entrega buenos resultados. Se deben destacar los siguientes aspectos:

Con el programa solucionamos los dos problemas mencionados en la introducción; resolvemos el ciclo en poco tiempo y con bastante exactitud. En base a esto, podemos desarrollar, resolver y comparar varios ciclos en poco tiempo, con la seguridad de que nuestros resultados son correctos.

El programa cumple con la función que se tenía planteada como material de apoyo para el aprendizaje y manejo de ciclos de potencia de vapor, únicamente debe tenerse cuidado en el manejo del mismo, pues es necesario tener las bases teóricas necesarias, pues en caso contrario se corre el peligro de aprender a manejar el programa pero no entender el proceso físico que se está estudiando y por tanto tampoco interpretar correctamente los resultados que arroja el mismo.

El programa además cuenta con las ventajas de poder instalarse en cualquier computadora personal con las características descritas en el instructivo (apéndice 1) y ser de manejo sencillo. Los programas descritos al final del capítulo 4 son mucho más poderosos que el trabajo aquí presentado, pero éstos, por su alto precio y por el equipo que pueden llegar a necesitar, sólo justifican su gasto en trabajos de investigación o desarrollo de una planta.

Una importante función del programa es el poder analizar si las condiciones de diseño (presiones y temperaturas de entrada) son

correctas, puede darse el caso de escogerse presiones de sangrado para Calentadores dentro de la región de la campana, lo cual no es conveniente para nuestro equipo.

Otra importante aportación es el poder comparar de manera rápida la eficiencia de los ciclos reales contra los ideales, lo cual permite ir desarrollando condiciones óptimas, termodinámicamente hablando, pues estos fueron los únicos aspectos que se consideraron, los aspectos económicos para la selección de una planta no fueron estudiados en este trabajo.

Dentro de las desventajas debe decirse que en PASCAL 6 podemos llegar a tener limitantes, principalmente por el uso de memoria, aunque para el trabajo que se realizó, el paquete fue suficiente para cubrir las expectativas planteadas originalmente, si se debe recomendar que si se desea continuar trabajando sobre temas de este tipo, se use otro lenguaje de programación más poderoso.

BIBLIOGRAFIA

BIBLIOGRAFIA.

MORSE T. Frederick, Centrales Eléctricas, México, Ed. CECSA, 1981, (9ª edición).

KADAMBI V., PRASAD Manohar, Conversión de la Energía, (Trad. Carlos Alberto García Ferrer), México, Ed. Limusa, 1984.

SOCIETY FOR COMPUTER SIMULATION, Directory of Simulation Software, USA, 1991, volumen 2

PALMER D. Scott, Domine Turbo Pascal6, México, Ed. Ventura, 1992.

GROGONO Peter, Programación en Pascal, México, Ed. Addison Wesley Iberoamericana, 1986.

BORLAND Cía. Object-Oriented Programming Guide, USA, 1985, Borland Cía.

SHANNON E. Robert, Simulación de Sistemas, México, Ed. Trillas, 1988.

GRANET Irving, Termodinámica, (Trad. Luis Rafael León C.), México, Ed. Prentice Hall, 1988, (3ª edición).

WARK Kenneth, Termodinámica, (Trad. Pablo A. Lonngi Ayala), México, Ed. McGraw Hill, 1988, (4ª edición).

ADDIEGO Joe, "The New Technology of Simulation", Automotive Engineering, USA, No 11, Noviembre 1992, volumen 100.

ANEXO 1
INSTRUCTIVO

INSTRUCTIVO.

Para poder ejecutar el programa se necesita una computadora personal AT con al menos un procesador 80286, sistema operativo MSDOS versión 4.0 y 640K en memoria RAM a 12Mhz. No importa el tipo de monitor, ya que el programa reconoce el uso de hardware de gráficas que tiene su computadora y escoge el controlador adecuado. Se recomienda usar un monitor VGA para poder apreciar los colores usados en las distintas pantallas, además de por su buena resolución.

Para iniciar el programa teclee CPOTEN desde sistema operativo. Aparece un menú principal con 8 opciones distintas:

1. Iniciar un nuevo ciclo.
2. Cargar un reporte existente.
3. Editar parámetros.
4. Solución Numérica.
5. Solución Gráfica.
6. Dar trabajos.
7. Potencia de Turbina.
8. Salir.

Para seleccionar alguna de ellas, basta moverse por medio de las flechas del teclado y oprimir <ENTER>. A continuación analizaremos cada una de las opciones antes mencionadas.

1. Iniciar un nuevo ciclo.

Esta opción inicializa todos los parámetros y prepara al programa

para comenzar un ciclo nuevo.

La primera pantalla que aparece es para preguntar el ciclo que se desea resolver, igual que en el menú principal, la selección se hace mediante las flechas del teclado y oprimiendo <ENTER> en la deseada, hay 4 opciones:

- a. Ciclo de Carnot.
- b. Ciclo Rankine Sencillo.
- c. Ciclo Rankine Regenerativo.
- d. Ciclo Rankine óptimo.

a. Ciclo de Carnot.

Pregunta las presiones de entrada a la Turbina y al Condensador; muestra el esquema del ciclo y regresara al menú principal.

b. Ciclo Rankine Sencillo.

Pregunta la presión y temperatura de entrada a la Turbina, si existe recalentamiento, en caso afirmativo, la presión y temperatura de recalentamiento y la presión del Condensador; a continuación enseña el esquema del ciclo y regresa al menú principal.

c. Ciclo Rankine Regenerativo.

Consta de 4 pantallas, en la primera se pregunta los mismos datos que en el caso del Rankine sencillo, en la segunda el número de Calentadores Cerrados y Abiertos, el número de Bombas y el de Trampas, en la tercera pantalla se piden las presiones de entrada y salida a las que operan los equipos antes mencionados, en la última pantalla se da la

opción de ver o editar algún dato de la pantalla anterior.

Este es el único procedimiento que no tiene esquema predefinido, sino que hay que construirlo. Los elementos van apareciendo en el ángulo superior izquierdo de la pantalla, se mueven por medio de las flechas y cuando se desea que se quede en una determinada posición, se presiona <ENTER>, e inmediatamente sale el nuevo elemento de la misma esquina. Una vez colocados todos los componentes, se procede a unirlos, el programa enlaza a la turbina con todos sus sangrados y les asigna un número en el ciclo, dichos números van de mayor a menor según la presión de salida, por tanto el número 1 corresponde siempre a la entrada a la Turbina; una línea de la Turbina a la Caldera con 3 números indica recalentamiento.

Para juntar los elementos basta con teclear <F1> siempre que se desee un punto, sale el cursor, también del ángulo superior izquierdo, se mueve con las flechas y se deja en el primer elemento tecleando <ENTER>, al instante aparece un nuevo cursor en la esquina izquierda, el cual se debe mover hasta el segundo objeto, una vez que este último ha sido ubicado, también con <ENTER>, los dos elementos quedan unidos por una línea y con un número asignado. La unión de elementos debe hacerse de los de menor a mayor presión, y dejando al final las salidas de los Calentadores Cerrados que vienen de la Turbina y van a una Trampa que regresará el flujo al Calentador Abierto o que van a una Bomba que mandará el flujo hacia adelante.

Después de hacer el dibujo, regresamos al menú principal.

d. Ciclo Rankine Optimo.

Pregunta la presión y temperatura de entrada a la Turbina, la presión y temperatura de recalentamiento y la presión del Condensador; posteriormente enseña el valor que se ha calculado para las presiones de entrada y salida a los Calentadores, Bombas y Trampas; a continuación enseña el esquema del ciclo y regresa al menú principal.

En todas las pantallas, a excepción de la última del ciclo Rankine Regenerativo y de las presiones calculadas en el ciclo Optimo, se pueden editar los datos que se están viendo oprimiendo la tecla <E>, a continuación nos movemos con las flechas hasta la cifra que deseamos cambiar, tecleamos <ENTER>, la cifra desaparece y estamos listos para ingresar el nuevo valor, para finalizar la edición tecleamos <ESC>, aparece un mensaje de 'Edición Terminada', ya podemos pasar a la siguiente pantalla.

Toda la rutina de captura esta validada para que no se pueda escribir letras o dejar espacios vacíos cuando se pida una cifra, si hay un error al ingresar una cantidad aparece un mensaje de 'Error en la cifra', y se espera a que se anote el valor correcto; no se puede continuar hasta que no se dé una cantidad dentro del rango establecido en la ventana superior. El programa no avanza a la siguiente preguntas si se tratan de ingresar valores por arriba de los límites permitidos.

2. Cargar reporte existente.

Mediante esta opción podemos ver reportes de ciclos que ya fueron creados y resueltos.

Se pregunta el nombre del archivo que se desea ver, si se da el nombre de un archivo que no ha sido creado o que no se encuentra en el

disco del que esta leyendo, el programa manda el mensaje de error 'No existe el archivo' y regresa al menú principal.

Si el archivo existe, se presenta primero la siguiente información: nombre del Ciclo, número de Calentadores Cerrados, de Calentadores Abiertos, de Bombas y de Trampas, trabajo neto, calor suministrado, eficiencia térmica, eficiencia de la Turbina y eficiencia de las Bombas. A continuación aparecen los datos de todos los puntos del ciclo, número, presión, temperatura, entalpía, entropía, volumen específico y conexión.

Posteriormente se pregunta si se desea imprimir el reporte, en caso afirmativo se manda la información a la impresora, en caso de no estar lista ésta, nos manda al siguiente paso del programa.

Finalmente pregunta si se desea ver el diagrama T-s del ciclo; en caso negativo, nos regresa al menú principal.

Cabe destacar los reportes aquí presentados sólo dan información, no pueden ser modificados

3. Editar parámetros.

La rutina nos presenta una pantalla con los valores predefinidos de los parámetros. Estos con sus respectivos valores son:

- a. Eficiencia de la Turbina = 100%
- b. Eficiencia de las Bombas = 100%
- c. Cp del agua = 4.1818 KJ/Kg°C.
- d. Incremento de temperatura en agua de condensado = 8°C.

La manera de editar datos es similar a la de las pantallas de captura de datos, mediante las teclas <E> para iniciar la edición y <ESC> para

terminaría.

Siempre que se inicie un nuevo ciclo, los valores de los parámetros se inicializan nuevamente.

4. Solución Numérica.

Esta rutina presenta una lista con el número de cada punto según el esquema visto en la parte de captura, su presión, temperatura, entalpía, entropía, volumen y conexión.

Puede ocurrir que mientras se esté ejecutando esta parte aparezcan varios mensajes de error, éstos pueden ser:

* 'Flujo de C. A. negativo, Ciclo Termodinámico imposible, se tomará el valor absoluto del flujo para seguir.' Este mensaje sólo cuando se trabaja con ciclos óptimos o regenerativos. Indica que al hacer el balance energético en el Calentador Abierto, un flujo dio un valor negativo, por tanto el ciclo no es factible de llevarse a la realidad. Se avisa que el ciclo continuará pero tomando el valor absoluto de dicho flujo, de todos modos debe tenerse en mente que, aunque se tengan buenos resultados, el Ciclo no es real.

* 'Fuera de rango superior (vapor sobrecal).' Indica que se trató de hacer una interpolación por arriba del límite superior de la tabla de vapor sobrecalentado, puede presentarse si se escogen muy altas temperaturas o muy altas presiones. Estos mensajes, como los 3 siguientes, aparecen en la pantalla pero no permanecen mucho tiempo. Si lo ve durante la corrida del programa aunque se entregue una solución Numérica, los resultados no son reales, pues algunos puntos tendrán valores falsos; estos puntos serán los del lado derecho de la campana.

* 'Fuera de rango inferior (vapor sobrecal).' El mismo caso del anterior, pero ahora se trató de interpolar con valores por debajo del límite inferior de la tabla de vapor sobrecalentado. Este mensaje puede aparecer porque al escoger una presión de sangrado para un Calentador que caiga dentro de la región de la campana.

* 'Fuera del rango superior (liq compr).' Se trató de buscar un dato, para una presión dada, por arriba del límite superior de la tabla de líquido comprimido.

* 'Fuera del rango inferior (liq compr).' Se trató de buscar un dato, para una presión dada, por abajo del límite inferior de la tabla de líquido comprimido.

Después manda un mensaje preguntando si se desea generar un reporte para este ciclo, en caso afirmativo, se pide un nombre para el archivo con un máximo de 8 caracteres, se crean 2 archivos para cada reporte (nombre.DAT y nombre.CPT) el programa asigna las extensiones necesarias.

El archivo nombre.DAT contiene los datos de nombre del Ciclo, número de Calentadores Cerrados, de Calentadores Abiertos, de Bombas y de Trampas, trabajo neto, calor suministrado, eficiencia térmica, eficiencia de la Turbina y eficiencia de las Bombas.

El archivo nombre.CPT guarda la lista de puntos.

Siempre que se llame a esta rutina se recalcula el valor de todos los puntos, esto presenta la ventaja de que podemos mantener los datos de un mismo ciclo e ir variando el valor de los parámetros para observar su comportamiento.

5. Solución Gráfica.

Aquí se presenta el diagrama T-s para el ciclo que se ha resuelto, por esto es necesario ejecutar primero la Solución Numérica antes de pasar a esta rutina.

El programa autoescala la gráfica para que ocupe el total de la pantalla, dibuja la región de la campana y a continuación coloca todos los puntos del ciclo con su respectivo número para unirlos finalmente.

Igual que en la opción anterior, el diagrama siempre se construye con los datos que tiene en memoria, así que si se modifican los parámetros antes de correr la opción de la Solución Numérica y cambia el valor de los puntos, también cambiará el diagrama T-s.

6. Dar Trabajos.

Esta rutina presenta la información del trabajo de la Turbina, el trabajo total de bombas, el trabajo neto, el calor suministrado y la eficiencia térmica del ciclo.

Para correr esta rutina es necesario correr previamente la Solución Numérica, puede correrse antes o después de la Solución Gráfica, ya que no están relacionadas directamente.

7. Potencia de la Turbina.

Todo el ciclo se resuelve con propiedades específicas, para poder saber la potencia de la Turbina es necesario ingresar un dato de flujo de vapor de agua, de esta manera es ya fácil calcular tanto la potencia de la Turbina, como el suministro de agua de enfriamiento necesario para mantener el incremento de temperatura definido en los parámetros.

Durante esta parte del programa solamente se Pregunta el flujo de vapor en Kg/s y entrega los resultados de los dos cálculos antes mencionados.

8. Salir.

Esta opción da por terminado el programa y nos regresa al Sistema Operativo.

ANEXO 2

PROGRAMA PRINCIPAL

A continuación se presenta el listado del programa desarrollado en la tesis.

PROGRAM TESIS;

USES crt, graph, sammy, archiv, elementos, printer;

TYPE

```
Str2 = string[2];
Str5 = string[5];
Str8 = string[8];
Str11 = string[11];
Str12 = string[12];
```

Infor = RECORD

```
cicl,ccer,cabi,bomb,tram      : integer;
wneto,qsum,efic,etabom,etatur : real;
rec                          : boolean;
END;
```

VAR

```
opcic,rutina                : Integer;
nlineas,sumpto,sha         : Integer;
cc,ca,nb,nt                : Integer;
recal,cadena,adel         : Boolean;
gdat                       : Char;
basecc,baseb,baset,baseca : IntegerPtr;
eft,efb,cp,pc,deltat      : Real;
pmax,pmin,tmax,tmin       : Real;
pcond,maxdt,tt,tb,qr,eta,wn : Real;
nomArch                    : Str12;
nomcic                     : String[20];
TurbinaVar,TurbinaBaja     : Turbina;
CondensadorVar             : Condensador;
head_p,guia_t,tail_p,cola_t : PuntoPtr;
FormaPto,lista             : PuntoPtr;
varPto                     : Punto;
Varinfo                    : Infor;
ap_cc,ul_cc                : c_cerradoPtr;
h_flujo,t_flujo            : FlujoPtr;
```

Archivo : FILE OF Punto;
Datext : FILE OF Infor;

```
PROCEDURE Opven_bis(cf1,ct1,cf2,ct2: integer);
BEGIN
  ClrScr;
  Window(1,1,80,25);
  TextBackground(cf1);
  TextColor(ct1);
  ClrScr;
  Window(1,1,80,5);
  TextBackground(cf2);
  TextColor(ct2);
  ClrScr
END;
```

```
PROCEDURE Clven_bis(cf1,ct1: integer);
BEGIN
  Window(1,6,80,25);
  TextBackground(cf1);
  TextColor(ct1)
END;
```

```
PROCEDURE MuevaXY;
VAR
  c : Char;
```

```
BEGIN
  REPEAT
    c := ReadKey;
    CASE c OF
      #72 : Gotoxy(WhereX,WhereY-1);
      #75 : Gotoxy(WhereX-1,WhereY);
      #77 : Gotoxy(WhereX+1,WhereY);
      #80 : Gotoxy(WhereX,WhereY+1)
    END;
  UNTIL c = #13
END;
```

```
PROCEDURE Mueva_marca(x,ymin,ymax: integer;marca: char);
VAR
  c : Char;
  yaux : Integer;
BEGIN
```



```

yaux := WhereY;
REPEAT
  c := ReadKey;
  CASE c OF
    #72 : BEGIN
      IF yaux <> ymin THEN
        BEGIN
          Gotoxy(x,yaux);
          Write(' ');
          yaux := yaux-1;
          Gotoxy(x,yaux);
          Write(marca)
        END
      END;
    #80 : BEGIN
      IF yaux <> ymax THEN
        BEGIN
          Gotoxy(x,yaux);
          Write(' ');
          yaux := yaux+1;
          Gotoxy(x,yaux);
          Write(marca)
        END
      END
    END;
  UNTIL c = #13
END;

```

```

PROCEDURE llu_Str(str: String; x,y: integer);
VAR
  xaux : Integer;

BEGIN
  TextBackground(1);
  TextColor(15);
  GotoXY(x,y); Write(str);
  TextBackground(7);
  Textcolor(1)
END;

```

```

PROCEDURE Mensaje_editor(VAR c : char);
BEGIN
  REPEAT
    Gotoxy(25,17); Write('1. < RET > para continuar o ');
    Gotoxy(25,18); Write('2. < E > para editar alg-n dato y < ESC > para');
    Gotoxy(25,19); Write(' Terminar edici_n ');
    c := ReadKey
  UNTIL c in ['e','E','#13]

```

END;

PROCEDURE llu_chr(VAR c: char; x,y,cf1,ct1,cf2,ct2: integer);

VAR

xaux : Integer;

sigue : Boolean;

BEGIN

xaux := x;

TextBackground(cf2);

TextColor(ct2);

REPEAT

Gotoxy(x,y); Write(' ');

Gotoxy(x,y); c := ReadKey;

Write(c);

UNTIL c in ['s','S','n','N'];

TextBackground(cf1);

TextColor(ct1)

END;

PROCEDURE Portada;

VAR

sn : Char;

BEGIN

Opven_bis(1,14,4,14);

Gotoxy(6,2); Write('UNIVERSIDAD PANAMERICANA ');

Gotoxy(6,3); Write('Escuela de Ingenier_a ');

Gotoxy(45,2); Write('Soluci_n de Ciclos de vapor ');

Gotoxy(45,3); Write('Utilice ',#24,' y ',#25,' para moverse ');

Clven_bis(1,14);

REPEAT

ClrScr;

Gotoxy(10,3); Write('Sistema de Unidades : ');

Gotoxy(50,3); Write('Metrico ');

Gotoxy(10,13); Write(' Usted selecciono : ');

pmax:= 220.9; pmin:= 0.04; tmax:= 800; tmin:= 28.96; pcond:=0.4 ; maxdt:= 18;

Gotoxy(10,8); Write('Ciclo a Resolver : ');

Gotoxy(32,8); Write('1. Ciclo de Carnot.....[_] ');

Gotoxy(32,9); Write('2. Ciclo Rankine sencillo.....[] ');

Gotoxy(32,10); Write('3. Ciclo Rankine regenerativo...[] ');

Gotoxy(32,11); Write('4. Ciclo Rankine optimo.....[] ');

Gotoxy(66,8);

Mueva_marca(66,8,11,#219);

opcic := WhereY;

CASE opcic OF

8 : BEGIN

Gotoxy(40,14); Write(' Ciclo de Carnot ')

```

    END;
9: BEGIN
    Gotoxy(40,14); Write(' Ciclo Rankine sencillo ');
    END;
10: BEGIN
    Gotoxy(40,14); Write(' Ciclo Rankine regenerativo ');
    END;
11: BEGIN
    Gotoxy(40,14); Write(' Ciclo Rankine Optimo ');
    END
    END;
Gotoxy(40,19); Write(' Estan correctos tus datos S/N ');
ilu_chr(sn,WhereX,WhereY,7,1,1,15);
UNTIL sn in ['s','S']
END;

```

```

PROCEDURE ilu_cant(x,y,cf1,ct1,cf2,ct2:integer; var s: str8; var n:real);
VAR
code      : Integer;
sigue, fin : Boolean;

BEGIN
fin:=FALSE;
REPEAT
TextBackground(cf2);
TextColor(ct2);
REPEAT
Gotoxy(x,y); Write(' ');
Gotoxy(x,y); ReadLn(s);
IF s = " THEN sigue := false
ELSE sigue := true;
UNTIL sigue = true;
VAL(s,n,code);
IF code <> 0 THEN
BEGIN
Gotoxy(x,y+1); Write(' Error en la cifra ');
ReadLn;
TextBackground(cf1);
TextColor(ct1);
Gotoxy(x,y+1); Write(' ');
TextBackground(cf2);TextColor(ct2);
fin := FALSE
END;
IF code = 0 THEN
BEGIN
Gotoxy(x+9,y);
TextBackground(cf1);
TextColor(ct1);
fin := TRUE

```

```

    END;
    UNTIL fin = TRUE
END;

```

```

PROCEDURE Resalta(x,y,cf1,ct1,cf2,ct2:integer; n:real);
BEGIN
    TextBackground(cf2);
    TextColor(ct2);
    Gotoxy(x,y); Write(' ');
    Gotoxy(x,y); Write(n:5:4);
    TextBackground(cf1);
    Textcolor(ct1)
END;

```

```

PROCEDURE ilu_ent(var i: integer; var s: str8; x,y,cf1,ct1,cf2,ct2: integer);
VAR
    code : Integer;
    sigue,fin : Boolean;

BEGIN
    REPEAT
        i:=0;
        TextBackground(cf2);
        TextColor(ct2);
        REPEAT
            GotoXY(x,y); Write(' ');
            GotoXY(x,y); ReadLn(s);
            IF s = " THEN sigue := FALSE
                ELSE sigue := TRUE;
            UNTIL sigue = TRUE;
        VAL(s,i,code);
        IF code <> 0 THEN
            BEGIN
                GotoXY(x,y+1); Write(' Error en la cifra ');
                fin := FALSE;
                ReadLn;
                TextBackground(cf1);
                TextColor(ct1);
                GotoXY(x,y+1); Write(' ');
                TextBackground(cf2); Textcolor(ct2)
            END;
            IF code = 0 THEN fin := TRUE;
            Gotoxy(x+3,y);
            TextBackground(cf1);
            Textcolor(ct1);
            UNTIL fin = TRUE
        END;

```

```
PROCEDURE Cap(x,y: integer; dat: Str2; VAR n: real);
```

```
VAR
```

```
  mensaje      : String[40];
```

```
  nombre       : Str8;
```

```
  codigo       : Char;
```

```
  top          : Boolean;
```

```
  valida,minimo : Real;
```

```
BEGIN
```

```
  IF dat = 'pt' THEN
```

```
    BEGIN
```

```
      mensaje := 'Presión de entrada a la turbina.....';
```

```
      top := FALSE;
```

```
      valida := pmax;
```

```
      minimo := pmin;
```

```
      codigo := 'p'
```

```
    END;
```

```
  IF dat = 'pc' THEN
```

```
    BEGIN
```

```
      mensaje:='Presión de entrada al condensador.....';
```

```
      top := FALSE;
```

```
      valida := pcond;
```

```
      minimo := pmin;
```

```
      codigo := 'p'
```

```
    END;
```

```
  IF dat = 'pr' THEN
```

```
    BEGIN
```

```
      mensaje := 'Presión de entrada de recalentamiento.';
```

```
      top := FALSE;
```

```
      valida := pmax;
```

```
      minimo := pmin;
```

```
      codigo := 'p'
```

```
    END;
```

```
  IF dat = 'tt' THEN
```

```
    BEGIN
```

```
      mensaje := 'Temperatura de entrada a la turbina...';
```

```
      top := TRUE;
```

```
      valida := tmax;
```

```
      minimo := tmin;
```

```
      codigo := 't'
```

```
    END;
```

```
  IF dat = 'tr' THEN
```

```
    BEGIN
```

```
      mensaje := 'Temperatura de recalentamiento.....';
```

```
      top := TRUE;
```

```
      valida := tmax;
```

```
      minimo := tmin;
```

```
      codigo := 't'
```

```
    END;
```

```

GotoXY(x,y); Write(mensaje);
GotoXY(x+57,y);
  IF top=FALSE THEN Write('Bar')
  ELSE Write("°C");
  REPEAT
    ilu_cant(x+40,y,7,1,1,15,nombre,n);
  UNTIL (n <= valida) and (n >= minimo);
END;
```

```
PROCEDURE Editor(x,y,pro: integer; dat: char; VAR n: real; VAR f: boolean);
```

```

VAR
otro,u           : Char;
c                : Str8;
cf1,ct1,cf2,ct2 : Integer;
condicion,minimo : Real;
```

```

BEGIN
CASE pro OF
  1 : BEGIN
    - cf1 := 7;
      ct1 := 1;
      cf2 := 1;
      ct2 := 15;
    END;
  2 : BEGIN
    cf1 := 9;
    ct1 := 14;
    cf2 := 3;
    ct2 := 14;
    END;
END;
END;
IF dat = 'p' THEN
BEGIN
  condicion := pmax;
  minimo := pmin;
  u:='p'
END;
IF dat = 'c' THEN
BEGIN
  condicion := pcond;
  minimo := pmin;
  u := 'p'
END;
IF dat = 'e' THEN
BEGIN
  condicion := 100;
  minimo := 1;
  u:='e'
END;
```

```

IF dat = 'l' THEN
  BEGIN
    condicion := tmax;
    minimo := tmin;
    u:='l'
  END;
IF dat = 'd' THEN
  BEGIN
    condicion := maxdt;
    minimo := 1;
    u := 'e'
  END;
REPEAT
  ilu_cant(x,y,cf1,ct1,cf2,ct2,c,n);
UNTIL (n <= condicion) and (n >= minimo);
REPEAT;
  otro := ReadKey;
UNTIL otro in [#27,#72,#75,#77,#80];
  IF otro = #27 THEN f := TRUE
  ELSE f := FALSE
END;

```

```

PROCEDURE Ed_integer(y: integer; VAR i: integer; VAR f: boolean);
VAR
  otro : Char;
  c : Str8;

BEGIN
  ilu_ent(i,c,48,y,7,1,1,15);
  REPEAT
    otro := ReadKey;
  UNTIL otro in [#27,#72,#75,#77,#80];
  IF otro = #27 THEN f := TRUE
END;

```

```

PROCEDURE Init_presion;
BEGIN
  basecc := NIL;
  baseca := NIL;
  baseb := NIL;
  baset := NIL
END;

```

```

PROCEDURE Lista_def(VAR base :PresionPtr; i : Integer);
VAR
  tail : PresionPtr;
  cont : Integer;

```

```

BEGIN
  cont := 0;
  WHILE cont <> I DO
    BEGIN
      cont := cont+1;
      IF base = NIL THEN
        BEGIN
          NEW(base);
          base^.init(0,0);
          base^.numera(cont);
          base^.apunta(NIL);
          tail := base
        END
      ELSE
        BEGIN
          NEW(tail^.apunt);
          IF base = tail THEN base^.apunt := tail^.apunt;
          tail := tail^.apunt;
          tail^.init(0,0);
          tail^.numera(cont);
          tail^.apunta(NIL);
        END
      END
    END
  END;

```

```

PROCEDURE Llana_listadef(VAR base: PresionPtr; I :integer; pent,
psal :real);
VAR
  siguePre : PresionPtr;

```

```

BEGIN
  siguePre := base;
  WHILE siguePre <> NIL DO
    BEGIN
      IF I = siguePre^.numero THEN
        BEGIN
          siguePre^.init(pent,psal);
          siguePre:=NIL;
        END
      ELSE
        siguePre := siguePre^.apunt
      END
    END
  END;

```

```

PROCEDURE Forma_lista(numero,v :integer; VAR i,iaux: integer; VAR base:
presionptr);
VAR

```



```

tail      : PresionPtr;
cif       : Str8;
mensaje   : String[15];
lector,lect2 : Real;
ban       : Boolean;

```

```
BEGIN
```

```
  ban := FALSE;
```

```
  CASE v OF
```

```
    1 : mensaje := 'Bomba';
```

```
    2 : mensaje := 'Trampa';
```

```
    3 : mensaje := 'C. Cerrado';
```

```
    4 : mensaje := 'C. Abierto'
```

```
  END;
```

```
  IF v in [1] THEN
```

```
    BEGIN
```

```
      TextColor(20);
```

```
      GotoXY(10,17); Write('ADVERTENCIA ');
```

```
      TextColor(9);
```

```
      GotoXY(10,18);
```

```
      Write('Recuerde poner presiones de menores a MAYORES para ,mensaje, ');
```

```
      TextColor(1)
```

```
    END;
```

```
  IF v in [2,3,4] THEN
```

```
    BEGIN
```

```
      TextColor(20);
```

```
      GotoXY(10,17); Write('ADVERTENCIA ');
```

```
      TextColor(15);
```

```
      GotoXY(10,18);
```

```
      Write('Recuerde poner presiones de MAYORES a menores para ',mensaje);
```

```
      TextColor(1);
```

```
    END;
```

```
  REPEAT
```

```
    IF numero = 0 THEN ban := TRUE
```

```
    ELSE
```

```
      BEGIN
```

```
        IF laux = numero THEN ban := TRUE;
```

```
        IF base = NIL THEN
```

```
          BEGIN
```

```
            IF v = 3 THEN
```

```
              BEGIN
```

```
                GotoXY(10,2);
```

```
                Write('P1 = Presión de salida Turbina. P2 = Presión alta')
```

```
              END
```

```
            ELSE
```

```
              BEGIN
```

```
                Gotoxy(10,2);
```

```
                Write('P1 = Presión de entrada. P2 = Presión de salida')
```

```
              END;
```

```
            NEW(base);
```

```

REPEAT
  GotoXY(6,2+i); Write('P 1 ',mensaje,' ',iaux);
  GotoXY(36,2+i); Write('Bar');
  Ilu_cant(25,2+i,7,1,1,15,cif,lector);
UNTIL (lector <= pmax) and (lector >= pmin);
REPEAT
  IF v = 4 THEN
    BEGIN
      GotoXY(42,2+i); Write('P 2 ',mensaje,' ',iaux);
      GotoXY(72,2+i); Write('Bar');
      Resalta(61,2+i,7,1,1,15,lector);
      lect2 := lector
    END
  ELSE
    BEGIN
      GotoXY(42,2+i); Write('P 2 ',mensaje,' ',iaux);
      GotoXY(72,2+i); Write('Bar');
      Ilu_cant(61,2+i,7,1,1,15,cif,lect2)
    END;
  UNTIL (lect2 <= pmax) AND (lect2 >= pmin);
  base^.init(lector,lect2);
  base^.numera(iaux);
  base^.apunta(NIL);
  tail := base
END
ELSE
  BEGIN
    NEW(tail^.apunt);
    IF base = tail THEN base^.apunta(tail^.apunt);
    REPEAT
      GotoXY(6,2+i); Write('P 1 ',mensaje,' ',iaux);
      GotoXY(36,2+i); Write('Bar');
      Ilu_cant(25,2+i,7,1,1,15,cif,lector);
    UNTIL (lector <= pmax) AND (lector >= pmin);
    REPEAT
      IF v = 4 THEN
        BEGIN
          GotoXY(42,2+i); Write('P 2 ',mensaje,' ',iaux);
          GotoXY(72,2+i); Write('Bar');
          Resalta(61,2+i,7,1,1,15,lector);
          lect2 := lector
        END
      ELSE
        BEGIN
          GotoXY(42,2+i); Write('P 2 ',mensaje,' ',iaux);
          GotoXY(72,2+i); Write('Bar');
          Ilu_cant(61,2+i,7,1,1,15,cif,lect2)
        END;
      UNTIL (lect2 <= pmax) AND (lect2 >= pmin);
      tail := tail^.apunt;

```

```

tail^.numera(iaux);
tail^.init(lector,lect2);
tail^.apunta(NIL)
END;
lector := 0;
i := i+1;
iaux := iaux+1;
IF i >= 15 THEN
  BEGIN
    Clrscr;
    i := 1
  END
END;
UNTIL ban = TRUE;
i := i+1;
iaux := 1
END;

```

PROCEDURE Muestra_lista(v : Integer; VAR base : PresionPtr);

VAR

```

a,b      : Real;
j,m      : Integer;
SiguePre : PresionPtr;
mensaje  : String[40];
s        : Str8;

```

BEGIN

CASE v OF

```

  1 : mensaje := 'Presiones de las BOMBAS';
  2 : mensaje := 'Presiones de las TRAMPAS';
  3 : mensaje := 'Presiones de los C. CERRADOS';
  4 : mensaje := 'Presiones de los C. ABIERTOS'

```

END;

j := 1;

SiguePre := base;

ClrScr;

IF siguePre = NIL THEN

BEGIN

llw_Str(' LISTA VACIA, no hay datos a editar ',15,5);

END

ELSE

BEGIN

WHILE siguePre <> NIL DO

BEGIN

a := siguePre^.get_pin;

b := siguePre^.get_pout;

m := siguePre^.numero;

GotoXY(25,2); Write(mensaje, ' |Bar');

GotoXY(2,3+j); Write(m);

```

        GotoXY(5,3+j); Write('P1 = ');
        Resalta(10,3+j,7,1,1,15,a);
        GotoXY(45,3+j); Write('P2 = ');
        Resalta(50,3+j,7,1,1,15,b);
        siguePre := siguePre^.apunt;
        j := j+1
    END
END;
Gotoxy(10,18);
IF v in [1,2] THEN Write('Recuerde poner presiones de menores a MAYORES');
IF v in [3,4] THEN Write('Recuerde poner presiones de MAYORES a menores')
END;

```

PROCEDURE Edita_lista(y,no: integer; VAR valor : real; VAR base : presionptr; VAR f: boolean);

VAR

```

i           : Integer;
fijo       : Real;
otro      : Char;
especial  : Boolean;
siguePre  : PresionPtr;

```

BEGIN

```

siguePre := base;
IF base = baseca THEN especial := TRUE
ELSE especial := False;
f := FALSE;
WHILE siguePre <> NIL DO
    BEGIN
        i := siguePre^.numero;
        IF i = (y-3) THEN
            BEGIN
                CASE no OF
                    1 : BEGIN
                            IF especial = FALSE THEN
                                BEGIN
                                    fijo := siguePre^.get_pout;
                                    siguePre^.init(valor,fijo)
                                END
                            ELSE
                                BEGIN
                                    siguePre^.init(valor,valor)
                                END
                            END;
                    2 : BEGIN
                            IF especial = False THEN
                                BEGIN
                                    fijo := siguePre^.get_pin;
                                    siguePre^.init(fijo,valor)
                                END
                            END;
                END
            END
        END;
    END

```

```

        END
      ELSE
        BEGIN
          siguePre^.init(valor,valor)
        END
      END
    END
  END;
  siguePre := NIL
END
ELSE
  siguePre := siguePre^.apunt
END;
REPEAT
  otro := ReadKey;
UNTIL otro in [#27,#72,#75,#77,#80];
IF otro = #27 THEN f:=TRUE
ELSE f := FALSE
END;
END;

```

```

PROCEDURE Ed_final(n : integer; VAR base : presionptr);

```

```

VAR

```

```

j,k,o   : Integer;

```

```

cifra   : Str8;

```

```

cambio  : Real;

```

```

flag    : Boolean;

```

```

BEGIN

```

```

  REPEAT

```

```

    Muestra_lista(n,base);

```

```

    IF base = NIL THEN flag := TRUE

```

```

    ELSE

```

```

      BEGIN

```

```

        GotoXY(30,15); Write('<ENTER> para editar dato ');

```

```

        GotoXY(30,16); Write('<ESC> para finalizar edici_n ');

```

```

        GotoXY(10,4);

```

```

        MuevaXY;

```

```

        IF WhereX = 10 THEN j := 1;

```

```

        IF WhereX = 50 THEN j := 2;

```

```

        k := WhereX;

```

```

        o := WhereY;

```

```

        REPEAT

```

```

          ilu_cant(k,o,7,1,1,15,cifra,cambio);

```

```

        UNTIL (cambio <= pmax) and (cambio >= pmin);

```

```

        Edita_lista(WhereY,j,cambio,base,flag);

```

```

      END;

```

```

    UNTIL flag = TRUE;

```

```

  CirScr;

```

```

  Muestra_lista(n,base);

```

```

  Ilu_Str(' Edici_n Terminada ',30,15);

```

```

ReadLn
END;

```

```

PROCEDURE Vacia_presion(cabeza : PresionPtr);
VAR
  tempr1, tempr2 : PresionPtr;
BEGIN
  tempr1 := cabeza;
  WHILE tempr1 <> NIL DO
    BEGIN
      tempr2 := tempr1;
      tempr1 := tempr1^.apunt;
      DISPOSE(tempr2)
    END
  END;
END;

```

```

PROCEDURE Initg;
VAR
  grdriver, grmode : Integer;
BEGIN
  grdriver := detect;
  initgraph(grdriver, grmode, 'c:\tp\bgi');
  IF graphresult <> grOk then halt(1)
END;

```

```

FUNCTION Checa_maximo(p_r : real; base : PresionPtr):Integer;
VAR
  inf : Integer;
  siguePre : PresionPtr;
BEGIN
  inf := 0;
  siguePre := base;
  WHILE siguePre <> NIL DO
    BEGIN
      IF siguePre^.get_pin > p_r THEN
        inf := inf+1;
        siguePre := siguePre^.apunt
      END;
    checa_maximo := inf
  END;
END;

```

```

PROCEDURE Captura;
TYPE

```

```
Str20 = string(20);
```

```
VAR
ciclo          : Str20;
cifra         : Str8;
pt,pc,pr,tr,tt,tc : Real;
cambio,toq    : Real;
oprec,ed      : Char;
flag,ban      : Boolean;
j,k,lugar     : Integer;
divtem,divpr,su,pcao : Real;
siguePre      : PresionPtr;
```

```
BEGIN
```

```
  ClrScr;
  cc := 0; ca := 0; nb := 0; nt := 0;
  recal := FALSE;
  portada;
```

```
  ClrScr;
```

```
  CASE opcic OF
```

```
    8 : ciclo := 'CARNOT';
    9 : ciclo := 'RANKINE';
    10 : ciclo := 'REGENERATIVO';
    11 : ciclo := 'OPTIMO'
```

```
  END;
```

```
  Opven_bis(7,1,1,15);
```

```
  ClrScr;
```

```
  GotoXY(5,1); Write('M_trico');
```

```
  GotoXY(5,2); Write('CiCLO : ',ciclo);
```

```
  GotoXY(34,2); Write('UTILICE LAS FLECHAS PARA EDITAR CANTIDADES ');
```

```
  GotoXY(5,4);
```

```
  Write('Pmax = ',pmax:4:1,' Tmax = ',tmax:3:0,' Pmax condensador = pcond:3:3);
```

```
  GotoXY(5,5); Write('Pmin = ',pmin:4:3,' Tmin = ',tmin:4:2);
```

```
  Clven_bis(7,1);
```

```
  CASE opcic OF
```

```
    8 : BEGIN
```

```
      Cap(8,6,'pt',pt);
```

```
      tt := Busca_campana('p','t',pt);
```

```
      Cap(8,8,'pc',pc);
```

```
      tc := Busca_campana('p','t',pc);
```

```
      Mensaje_editor(ed);
```

```
      IF ed in ['e','E'] THEN
```

```
        BEGIN
```

```
          flag := FALSE;
```

```
          REPEAT
```

```
            MueveXY;
```

```
            CASE WhereY OF
```

```
              6 : editor(48,6,1,'p',pt,flag);
```

```
              8 : editor(48,8,1,'c',pc,flag)
```

```

        END;
        UNTIL flag = TRUE;
        tt := Busca_campana('p','t,pt);
        tc := Busca_campana('p','t,pc);
        llu_Str(' Edición terminada ',10,14);
        ReadLn
    END;
    Initt;
    Dibuja_def(TRUE,FALSE);
    Closegraph
END;

```

```

9 : BEGIN
    recal := FALSE;
    Cap(8,3,'pt,pt);
    Cap(8,5,'t,tt);
    GotoXY(8,6); Write('+Existe Recalentamiento? s/n .....');
    llu_chr(ed,48,6,7,1,1,15);
    IF ed in ['s','S'] THEN
        BEGIN
            recal := TRUE;
            Cap(8,7,'pr,pr);
            Cap(8,9,'tr,tr)
        END;
    Cap(8,11,'pc,pc);
    tc := Busca_campana('p','t,pc);
    Mensaje_editor(ed);
    IF ed in ['e','E'] THEN
        BEGIN
            flag := FALSE;
            REPEAT
                MuevøXY;
                CASE WhereY OF
                    3 : editor(48,3,1,'p',pt,flag);
                    5 : editor(48,5,1,'t',tt,flag);
                    7 : editor(48,7,1,'p',pr,flag);
                    9 : editor(48,9,1,'t',tr,flag);
                    11 : editor(48,11,1,'c',pc,flag)
                END;
                tc := Busca_campana('p','t,pc);
            UNTIL flag = TRUE;
            llu_Str(' Edición terminada ',10,14);
            ReadLn
        END;
    Initt;
    Dibuja_def(FALSE,recal);
    Closegraph
END;

```

```

10 : BEGIN

```



```

Cap(8,3,'pt',pt);
Cap(8,5,'tt',tt);
GotoXY(8,8);Write(' Existe Recalentamiento? s/n .....');
llu_chr(ed,48,6,7,1,1,15);
IF ed in ['s','S'] THEN
  BEGIN
    recal := TRUE;
    Cap(8,7,'pr',pr);
    Cap(8,9,'tr',tr)
  END;
Cap(8,11,'pc',pc);
tc := Busca_campana('p','t',pc);
Mensaje_editor(ed);
IF ed in ['e','E'] THEN
  BEGIN
    flag := FALSE;
    REPEAT
      Muevexy;
      CASE WhereY OF
        3 : editor(48,3,1,'p',pt,flag);
        5 : editor(48,5,1,'t',tt,flag);
        7 : editor(48,7,1,'p',pr,flag);
        9 : editor(48,9,1,'t',tr,flag);
        11 : editor(48,11,1,'c',pc,flag)
      END;
      tc := Busca_campana('p','t',pc);
    UNTIL flag=TRUE;
    llu_Str(' Edición terminada ',10,14);
    ReadLn
  END;
ClrScr; {segunda pantalla}
GotoXY(8,3); Write('Número de calentadores cerrados.....');
llu_ent(cc,cifra,48,3,7,1,1,15);
GotoXY(8,5); Write('Número de calentadores abiertos.....');
llu_ent(ca,cifra,48,5,7,1,1,15);
  REPEAT
    GotoXY(8,7); Write('Número de bombas.....');
    llu_ent(nb,cifra,48,7,7,1,1,15);
  UNTIL nb>0;
GotoXY(8,9); Write('Número de trampas de estrangulamiento..');
llu_ent(nt,cifra,48,9,7,1,1,15);
Mensaje_editor(ed);
IF ed in ['e','E'] THEN
  BEGIN
    flag := FALSE;
    REPEAT
      Muevexy;
      CASE WhereY OF
        3 : ed_integer(3,cc,flag);
        5 : ed_integer(5,ca,flag);

```

```

7 : ed_integer(7,nb,flag);
9 : ed_integer(9,nt,flag);
END;
UNTIL flag = TRUE;
Ilu_Str(' Edición terminada ',10,14);
ReadLn
END;
ClrScr; {tercera pantalla}
j := 1; k := 1;
ban := FALSE;
IF basecc <> NIL THEN Vacía_presion(basecc);
IF baseca <> NIL THEN Vacía_presion(baseca);
IF baseb <> NIL THEN Vacía_presion(baseb);
IF baset <> NIL THEN Vacía_presion(baset);
Init_presion;
Forma_lista(cc,3,j,k,basecc);
Forma_lista(ca,4,j,k,baseca);
Forma_lista(nb,1,j,k,baseb);
Forma_lista(nt,2,j,k,baset);
ClrScr; {cuarta pantalla}
REPEAT
ClrScr;
GotoXY(20,3); Write('Editar C. Cerrados.....[ _ ] ');
GotoXY(20,4); Write('Editar C. Abiertos.....[ ] ');
GotoXY(20,5); Write('Editar Bombas.....[ ] ');
GotoXY(20,6); Write('Editar Trampas.....[ ] ');
GotoXY(20,7); Write('Mostrar los C. Cerrados.....[ ] ');
GotoXY(20,8); Write('Mostrar los C. Abiertos.....[ ] ');
GotoXY(20,9); Write('Mostrar las Bombas.....[ ] ');
GotoXY(20,10); Write('Mostrar las Trampas.....[ ] ');
GotoXY(20,11); Write('Seguir adelante.....[ ] ');
GotoXY(50,3);
Mueve_marca(50,3,11,#219);
lugar:=WhereY;
CASE lugar OF
3 : Ed_final(3,basecc);
4 : Ed_final(4,baseca);
5 : Ed_final(1,baseb);
6 : Ed_final(2,baset);
7 : BEGIN
Muestra_lista(3,basecc);
ReadLn
END;
8 : BEGIN
Muestra_lista(4,baseca);
ReadLn
END;
9 : BEGIN
Muestra_lista(1,baseb);
ReadLn

```

```

        END;
    10 : BEGIN
        Muestra_lista(2,baset);
        Readln
    END
END;
UNTIL lugar = 11;
sumpto := 0;
IF Recal = TRUE THEN sumpto := Checa_maximo(pr,basecc)
ELSE sumpto := 0;
Initg;
Dibuja_simul(sumpto,cc,ca,nb,nt,recal,nlineas);
Closegraph
END; {end de 10 del case}

11: BEGIN
divtem := 0;
sumpto := 0;
Cap(8,3,'pt',pt);
Cap(8,5,'tt',tt);
Cap(8,7,'pr',pr);
Cap(8,9,'tr',tr);
Cap(8,11,'pc',pc);
tc := Busca_campana('p','t',pc);
Mensaje_editor(ed);
IF ed in ['e','E'] THEN
BEGIN
    flag := FALSE;
    REPEAT
        MueveXY;
        CASE WhereY OF
            3 : editor(48,3,1,'p',pt,flag);
            5 : editor(48,5,1,'t',tt,flag);
            7 : editor(48,7,1,'p',pr,flag);
            9 : editor(48,9,1,'t',tr,flag);
            11 : editor(48,11,1,'c',pc,flag)
        END;
        tc := Busca_campana('p','t',pc);
    UNTIL flag = TRUE;
    llv_Str(' Edición terminada ',10,14);
    Readln
END;
cc := 3; ca := 1; nb := 2; nt := 3;
su := (Busca_campana('p','t',pt)-tc)/5;
Init_presion;
divtem := tc+su;
divpr := Busca_campana('t','p',divtem);
divpr := ROUND(divpr);
pcao := divpr;
Lista_def(baset,3); Lista_def(basecc,3);

```

```

Lista_def(baseb,2); Lista_def(baseca,1);
Llena_listadef(baseca,1,divpr,divpr);
Llena_listadef(baseb,1,pc,divpr);
Llena_listadef(baseb,2,divpr,pt);
divtem := Busca_campana('p','t',pt);
FOR j := 1 to cc DO
  BEGIN
    divtem := divtem-su;
    divpr := Busca_campana('t','p',divtem);
    divpr := ROUND(divpr);
    IF divpr > pr THEN sumpto := sumpto+1;
    Llena_listadef(basecc,j,divpr,pt)
  END;
divtem := Busca_campana('p','t',pt);
FOR j := 1 to nt DO
  BEGIN
    divtem := divtem-su;
    divpr := Busca_campana('t','p',divtem);
    divpr := ROUND(divpr);
    Llena_listadef(baset,j,divpr,pcao)
  END;
nlineas := 21;
Recal := TRUE;
Muestra_lista(1,baseb); Readln;
Muestra_lista(2,baset); Readln;
Muestra_lista(3,basecc); Readln;
Muestra_lista(4,baseca); Readln;
Initg;
optimo(sumpto);
Closegraph;
END; {end del 11 del case}

END; {end case general}
TurbinaVar.init(pt,tt);
TurbinaVar.entropia(pt,tt);
CondensadorVar.init(pc,tc);
CondensadorVar.salidas(pc);
IF recal = TRUE THEN
  BEGIN
    TurbinaBaja.init(pr,tr);
    TurbinaBaja.entropia(pr,tr)
  END;
IF cc > 0 THEN
  BEGIN
    Init_cc(ap_cc,ul_cc);
    siguePre := basecc;
    WHILE siguePre <> NIL DO
      BEGIN
        Crea_cc(siguePre^.get_pin,siguePre^.get_pout,ap_cc,ul_cc);
        siguePre := siguePre^.apunt
      END;
    END;
  END;

```

```

        END
    END;
END; {end del procedure}

```

```

PROCEDURE Init_param;
BEGIN
    eft := 1;
    efb := 1;
    cp := 4.1818;
    deltat := 8;
END;

```

```

PROCEDURE Parametros;
VAR
    cifra      : Str8;
    ed,otro    : Char;
    flag       : Boolean;

```

```

BEGIN
    eft := eft*100; efb := efb*100;
    Opven_bis(9,14,3,14);
    GotoXY(6,2); Write('UNIVERSIDAD PANAMERICANA ');
    GotoXY(6,3); Write('PARAMETROS DEL PROGRAMA ');
    GotoXY(45,2); Write('Sistema M_trico');
    GotoXY(45,3); Write('Utilice '#24,' y '#25,' para moverse ');
    Clven_bis(9,14);
    GotoXY(10,3); Write('Eficiencia de la turbina.....');
    GotoXY(10,5); Write('Eficiencia de las bombas.....');
    GotoXY(10,7); Write('Cp del agua de enfriamiento.....');
    GotoXY(10,9); Write('Delta T del agua del condensador.....');
    Resalta(55,3,9,14,3,14,efb);
    Resalta(55,5,9,14,3,14,efb);
    Resalta(55,7,9,14,3,14,cp);
    Resalta(55,9,9,14,3,14,deltat);
    GotoXY(67,3); Write('%');
    Gotoxy(67,5); Write('%');
    Gotoxy(67,9); Write('KJ/Kg_C');
    GotoXY(67,7); Write('°C');
    Mensaje_editor(ed);
    IF ed in ['e','E'] THEN
        BEGIN
            flag := FALSE;
            REPEAT
                cifra := "";
                MueveXY;
                CASE WhereY OF
                    3 : editor(55,3,2,'e',eft,flag);
                    5 : editor(55,5,2,'e',efb,flag);
                    7 : editor(55,9,2,'d',cp,flag);

```

```

        9 : editor(55,11,2,'d',dellat,flag)
    END;
    UNTIL flag = TRUE;
    Jlu_Str(' Edición terminada ',10,14);
    Readln
    END;
    efb := efb/100;
    eft := eft/100
END;

```

PROCEDURE Inicia_lista;

```

BEGIN
    head_p := NIL;
    tail_p := NIL;
    guia_t := NIL;
    cola_t := NIL;
END;

```

PROCEDURE Cabeza_puntos;

VAR

checap, checas : Real;

BEGIN

```

    checap := TurbinaVar.get_pin;
    checas := TurbinaVar.get_s;
    NEW(head_p);
    WITH head_p^ DO
        BEGIN
            num := 1;
            p := TurbinaVar.get_pin;
            t := TurbinaVar.get_tin;
            s := TurbinaVar.get_s;
            h := TurbinaVar.get_hov(1, checap, checas);
            v := TurbinaVar.get_hov(2, checap, checas);
            enlace := 'CALD_TUR';
            apunt := NIL
        END;
    tail_p := head_p
END;

```

PROCEDURE Anade_punto(l : integer);

BEGIN

```

    NEW(tail_p^.apunt);
    IF head_p = tail_p THEN head_p^.apunt := tail_p^.apunt;
    tail_p := tail_p^.apunt;

```

```

tail_p^.num := i;
tail_p^.apunt := NIL
END;

```

```

PROCEDURE Vacia_datos(i: Integer; pr,tm,hp,sp,ve: real; lig: Str8);
VAR
next_p : PuntoPtr;

```

```

BEGIN
next_p := head_p;
WHILE next_p <> NIL DO
BEGIN
IF next_p^.num = i THEN
BEGIN
next_p^.p := pr;
next_p^.t := tm;
next_p^.h := hp;
next_p^.s := sp;
next_p^.v := ve;
next_p^.enlace := lig;
next_p := NIL
END
ELSE
next_p := next_p^.apunt
END
END;

```

```

PROCEDURE P_condensado(k: INTEGER; pcond,scond: REAL);
VAR
h1,p1,hreal,sreal,vap,vreal : Real;

```

```

BEGIN
hreal := 0;
sreal := 0;
vreal := 0;
vap := 0;
IF recal = FALSE THEN
BEGIN
p1 := TurbinaVar.get_pin;
h1 := TurbinaVar.get_hov(1,p1,scond)
END
ELSE
BEGIN
p1 := TurbinaBaja.get_pin;
h1 := TurbinaBaja.get_hov(1,p1,scond)
END;
hreal := h1-ef*(h1-TurbinaVar.get_hov(1,pcond,scond));
IF eft <> 1 THEN

```

```

BEGIN
vap :=
Calidad(1,hreal,Busca_campana('p','hf',pcond),Busca_campana('p','hg',pcond));
sreal :=
Calidad(2,vap,Busca_campana('p','sf',pcond),Busca_campana('p','sg',pcond));
vreal :=
Calidad(2,vap,Busca_campana('p','sf',pcond),Busca_campana('p','sg',pcond))
END
ELSE
BEGIN
sreal := scond;
vreal := TurbinaVar.get_hov(2,pcond,scond)
END;
Vacia_datos(k,pcond,CondensadorVar.get_tin,hreal,sreal,vreal,'TUR_COND');
Vacia_datos(k+1,pcond,CondensadorVar.get_tin,CondensadorVar.get_hvs(1),
CondensadorVar.get_hvs(3),CondensadorVar.get_hvs(2),'COND_BOM')
END;

```

```

PROCEDURE P_finales(pmx : REAL);
BEGIN
Anade_punto(-1);
Vacia_datos(-1,pmx,Busca_campana('p','t',pmx),Busca_campana('p','hf',pmx),
Busca_campana('p','sf',pmx),Busca_campana('p','vf',pmx),'CALDERA');
Anade_punto(-2);
Vacia_datos(2,pmx,Busca_campana('p','t',pmx),Busca_campana('p','hg',pmx),
Busca_campana('p','sg',pmx),Busca_campana('p','vg',pmx),'CALDERA')
END;

```

```

FUNCTION Get_hn(i: integer): REAL;
VAR
siguePre : PuntoPtr;

```

```

BEGIN
siguePre := head_p;
WHILE siguePre <> NIL DO
BEGIN
IF siguePre^.num = i THEN
BEGIN
get_hn := siguePre^.h;
siguePre := NIL
END
ELSE
siguePre := siguePre^.apunt
END
END;

```

```

PROCEDURE Carnot;

```



```

VAR
prest,presec,tempt,tempc,st,sc : Real;
BEGIN
  Inicia_lista;
  prest := TurbinaVar.get_pin;
  presec := CondensadorVar.get_pin;
  tempt := TurbinaVar.get_tin;
  tempc := CondensadorVar.get_tin;
  st := TurbinaVar.get_s;
  sc := Busca_campana('p','sf',prest);
  Cabeza_puntos;
  Anade_punto(2);

```

```

Vacia_datos(2,presec,tempc,TurbinaVar.get_hov(1,presec,st),TurbinaVar.get_s,TurbinaV
ar.get_hov(2,presec,st),'TUR_COND');
  Anade_punto(3);
  Vacia_datos(3,presec,tempc,TurbinaVar.get_hov(1,presec,sc),sc,TurbinaVar.
get_hov(2,presec,sc),'COND_BOM');
  Anade_punto(4);
  Vacia_datos(4,prest,tempt,Busca_campana('p','hf',prest),Busca_campana('p','sf',prest),
Busca_campana('p','vf',prest),'BOM_CALD')
END;

```

PROCEDURE Rankine;

```

VAR
ptb,pcd,pbaj,sbaj,s1,w,h1,hreal,sreal,vreal : Real;

```

```

BEGIN
  hreal := 0;
  sreal := 0;
  ptb := TurbinaVar.get_pin;
  pcd := CondensadorVar.get_pin;
  s1 := TurbinaVar.get_s;
  w := (ptb-pcd)*Busca_campana('p','vf',pcd)/10;
  h1 := CondensadorVar.get_hvs(1)+w;
  Inicia_lista;
  Cabeza_puntos;
  Anade_punto(2);
  Anade_punto(3);
  Anade_punto(4);
  IF recal = FALSE THEN
    BEGIN
      P_condensado(2,pcd,s1);
      IF efb <> 1 THEN
        hreal := h1-efb*(h1-Get_hn(3))
      ELSE
        hreal := h1;
      Vacia_datos(4,ptb,Busca_liqs(ptb,'h','t',hreal),hreal,
        Busca_liqs(ptb,'h','s',hreal),Busca_liqs(ptb,'h','v',hreal),'BOM_CALD');
    END;
  END;

```

```

P_finales(ptb);
END
ELSE
BEGIN
  Anade_punto(5);
  pbaj := TurbinaBaja.get_pin;
  sbaj := TurbinaBaja.get_s;
  hreal := Get_hn(1)-eft*(Get_hn(1)-TurbinaVar.get_hov(1,pbaj,s1));
  IF eft <> 1 THEN
    BEGIN
      sreal := Busca_dato(pbaj,'h','s',hreal);
      vreal := Busca_dato(pbaj,'h','v',hreal);
    END
  ELSE
    BEGIN
      sreal := s1;
      vreal := TurbinaBaja.get_hov(2,pbaj,s1)
    END;
  Vacia_datos(2,pbaj,TurbinaBaja.get_hov(3,pbaj,s1),hreal,sreal,
    vreal,'TUR_CALD');
  Vacia_datos(3,pbaj,TurbinaBaja.get_tin,TurbinaBaja.
get_hov(1,pbaj,sbaj),sbaj,TurbinaBaja.get_hov(2,pbaj,sbaj),'CALD_TUR');
  P_condensado(4,pcd,sbaj);
  Anade_punto(6);
  IF efb <> 1 THEN
    hreal := h1-efb*(h1-Get_hn(5))
  ELSE
    hreal := h1;
  Vacia_datos(6,ptb,Busca_liqs(ptb,'h','t',hreal),hreal,
    Busca_liqs(ptb,'h','s',hreal),Busca_liqs(ptb,'h','v',hreal),'BOM_CALD');
  P_finales(ptb)
END
END;

```

```

PROCEDURE Llenar(base: PresionPtr; cual : integer; VAR cuento: integer);
VAR
  ecorr,sm,tq,hreal,hayu,sreal,vreal : Real;
  siguePre          : PresionPtr;
  liga              : Str8;

BEGIN
  hreal := 0;
  sreal := 0;
  vreal := 0;
  hayu := 0;
  IF base = basecc THEN liga := 'TUR_CC';
  IF base = baseca THEN liga := 'TUR_CA';
  siguePre := base;

```

```

WHILE siguePre <> NIL DO
  BEGIN
    ecorr := siguePre^.get_pin;
    IF recal = FALSE THEN
      BEGIN
        sm := TurbinaVar.get_s;
        hreal := Get_hn(1)-eft*(Get_hn(1-TurbinaVar.get_hov(1,ecorr,sm)));
        IF eft <> 1 THEN
          BEGIN
            sreal := Busca_dato(ecorr,'h','s',hreal);
            vreal := Busca_dato(ecorr,'h','v',hreal)
          END
        ELSE
          BEGIN
            sreal := sm;
            vreal := TurbinaVar.get_hov(2,ecorr,sm)
          END;
        Vacia_datos(cuento,ecorr,TurbinaVar.get_hov(3,ecorr,sm)
          ,hreal,sreal,vreal,liga);
        cuento := cuento+1;
        IF base = basecc THEN
          BEGIN
            Suma_dato(ecorr,TurbinaVar.get_hov(1,ecorr,sm),'h1',ap_cc)
          END
        END
      ELSE
        BEGIN
          tq := TurbinaBaja.get_pin;
          IF ecorr >= TurbinaBaja.get_pin THEN
            BEGIN
              sm := TurbinaVar.get_s;
              hayu := Get_hn(1)
            END
          ELSE
            BEGIN
              sm := TurbinaBaja.get_s;
              hayu := TurbinaBaja.get_hov(1,tq,sm)
            END;
          IF cuento = cual THEN
            BEGIN
              cuento := cuento+2
            END;
          hreal := hayu-eft*(hayu-TurbinaVar.get_hov(1,ecorr,sm));
          IF eft <> 1 THEN
            BEGIN
              sreal := Busca_dato(ecorr,'h','s',hreal);
              vreal := Busca_dato(ecorr,'h','v',hreal)
            END
          ELSE
            BEGIN

```

```

    sreal := sm;
    vreal := TurbinaVar.get_hcv(2,ecorr,sm)
END;

```

```

Vacia_datos(cuento,ecorr,TurbinaVar.get_hov(3,ecorr,sm),hreal,sreal,vreal,liga);
    cuento := cuento+1;
    IF base = basecc THEN
        BEGIN
            Suma_dato(ecorr,TurbinaVar.get_hov(1,ecorr,sm),'h1',ap_cc)
        END
    END;
    siguePre := siguePre^.apunt
END
END;

```

```

PROCEDURE s_ccerrado(VAR i : Integer; j: Integer; tsc,pscc,paux : Real);
VAR
    numerar : String;

BEGIN
    Str(j,numerar);
    Vacia_datos(i,pscc,tsc,Busca_liqs(pscc,'t','h',tsc),Busca_liqs(pscc,
        't','s',tsc),Busca_liqs(pscc,'t','v',tsc),'OUT CC'+numerar);
    Suma_dato(paux,Busca_liqs(pscc,'t','h',tsc),'h4',ap_cc)
END;

```

```

PROCEDURE Lista_trampa(nupt: integer; pr,tm,hp,sp,ve : real; lig: Str8);

```

```

BEGIN
    IF guia_t = NIL THEN
        BEGIN
            NEW(guia_t);
            guia_t^.num := nupt;
            guia_t^.p := pr;
            guia_t^.t := tm;
            guia_t^.h := hp;
            guia_t^.s := sp;
            guia_t^.v := ve;
            guia_t^.enlace := lig;
            guia_t^.apunt := NIL;
            cola_t := guia_t
        END
    ELSE
        BEGIN
            NEW(cola_t^.apunt);
            IF guia_t = cola_t THEN guia_t^.apunt := cola_t^.apunt;
            cola_t := cola_t^.apunt;
            cola_t^.num := nupt;

```

```

cola_t^.p := pr;
cola_t^.t := tm;
cola_t^.h := hp;
cola_t^.s := sp;
cola_t^.v := ve;
cola_t^.enlace := lig;
cola_t^.apunt := NIL

```

```

END

```

```

END;

```

```

PROCEDURE s_trampa(i : integer; ptram, htram : real);

```

```

VAR

```

```

pvapor, aux1, aux2, vaux, saux : Real;

```

```

BEGIN

```

```

aux1 := Busca_campana('p', 'hf', ptram);

```

```

aux2 := Busca_campana('p', 'hg', ptram);

```

```

pvapor := Calidad(1, htram, aux1, aux2);

```

```

saux :=

```

```

Calidad(2, pvapor, Busca_campana('p', 'sf', ptram), Busca_campana('p', 'sg', ptram));

```

```

vaux :=

```

```

Calidad(2, pvapor, Busca_campana('p', 'vf', ptram), Busca_campana('p', 'vg', ptram));

```

```

Lista_trampa(i, ptram, Busca_campana('p', 't', ptram), htram, saux, vaux, 'TRAM_CA');

```

```

END;

```

```

PROCEDURE Vacia_puntos(cabeza : PuntoPtr);

```

```

VAR

```

```

temptr1, temptr2 : PuntoPtr;

```

```

BEGIN

```

```

temptr1 := cabeza;

```

```

WHILE temptr1 <> NIL DO

```

```

BEGIN

```

```

temptr2 := temptr1;

```

```

temptr1 := temptr1^.apunt;

```

```

DISPOSE(temptr2)

```

```

END

```

```

END;

```

```

PROCEDURE Regenerativo;

```

```

VAR

```

```

i, j, z, k, fm, ica, m : Integer;

```

```

informe : Str8;

```

```

ptb, pcd, pbaj, pcol, s1, sbaj : Real;

```

```

hliq, psal, tigual, hreal, vreal, sreal : Real;

```

```

siguePre, Apbomb : PresionPtr;

```

```

ayuda : PuntoPtr;

```



```

IF eft <> 1 THEN
  BEGIN
    sreal := Busca_dato(pbj,'h','s',hreal);
    vreal := Busca_dato(pbj,'h','v',hreal)
  END
ELSE
  BEGIN
    sreal := s1;
    vreal := TurbinaBaja.get_hov(2,pbj,s1)
  END;
  Vacia_datos(j,pbj,TurbinaBaja.get_hov(3,pbj,s1),hreal,
    sreal,vreal,'TUR_CALD');
  Vacia_datos(j+1,pbj,TurbinaBaja.get_tin,TurbinaBaja.get_hov
    (1,pbj,sbj),sbaj,TurbinaBaja.get_hov(2,pbj,sbj),'CALD_TUR');
  j := 2
END
END;
END;
IF cc > 0 THEN Llenar(basecc,z,j);
IF ca > 0 THEN Llenar(baseca,z,j);
  IF recal = TRUE THEN s1 := sbaj
  ELSE s1 := s1;
  P_condensado(j,pcd,s1);
  j := j+1;
IF ca > 0 THEN
  BEGIN
    z := j+2*ca;
    siguePre := baseca;
    WHILE siguePre <> NIL DO
      BEGIN
        pcol := siguePre^.get_pin;
        siguePre^.liqsat(pcol);
        Vacia_datos(z,pcol,Busca_campana('p','t',pcol),siguePre^.
          get_hf,siguePre^.get_sf,siguePre^.get_vf,'CA_BOM');
        z := z-2;
        siguePre := siguePre^.apunt
      END
    END;
  END;
z := 0;
IF cc > 0 THEN
  BEGIN
    lca := cc;
    m := cc;
    z := mlineas-2*nt;
    siguePre := basecc;
    WHILE siguePre <> NIL DO
      BEGIN
        pcol := siguePre^.get_pout;
        psal := siguePre^.get_pin;
        siguePre^.liqsat(psal);
        tigual := Busca_campana('p','t',psal);

```

```

IF nt > 0 THEN
  BEGIN
    s_ccerrado(z,m,tigual,pcol,psal);
    Vacia_datos(z+2*cc-1,psal,tigual,siguePre^.get_hf,
      siguePre^.get_sf,siguePre^.get_vf,'CC_BOM');
    Suma_dato(psal,siguePre^.get_hf,'h2',ap_cc);
    m := m-1;
    z := z-1
  END
ELSE
  BEGIN
    fm := z-2*ica;
    s_ccerrado(fm,m,tigual,pcol,psal);
    Vacia_datos(z-1,psal,tigual,siguePre^.get_hf,
      siguePre^.get_sf,siguePre^.get_vf,'CC_BOM');
    Suma_dato(psal,siguePre^.get_hf,'h2',ap_cc);
    z := z-2;
    ica := ica-2;
    m := m-1
  END;
  siguePre := siguePre^.apunt
END
END;
k := j+2*ca+2*cc;
j := k+1;
siguePre := baseb; informe := "";
hreal := 0; sreal := 0; vreal := 0;
  WHILE siguePre <> NIL DO
    BEGIN
      pcol := siguePre^.get_pin;
      psal := siguePre^.get_pout;
      hliq := Busca_campana('p','vf',pcol)*(psal-pcol)/10;
      IF cc > 0 THEN
        BEGIN
          Apbomb := basecc;
          WHILE Apbomb <> NIL DO
            BEGIN
              IF Apbomb^.get_pout = psal THEN
                BEGIN
                  informe := 'BOMB_CC';
                  Apbomb := NIL
                END
              ELSE
                Apbomb := Apbomb^.apunt
            END
          END
        END;
      IF informe = " THEN
        BEGIN
          Apbomb := baseca;
          WHILE Apbomb <> NIL DO

```



```

BEGIN
  IF Apbomb^.get_pin = psal THEN
    BEGIN
      informe := 'BOMB_CA';
      Apbomb := NIL
    END
  ELSE
    Apbomb := Apbomb^.apunt;
  END
END;
IF informe = " THEN informe := 'BOM_CALD';
ayuda := head_p;
k := 0;
WHILE ayuda <> NIL DO
  BEGIN
    k := k+1;
    IF ayuda^.p <> 0 THEN ayuda:=ayuda^.apunt
    ELSE
      BEGIN
        hliq := hliq+Busca_campana('p','hf',pcol);
        hreal := Get_hn(k-1)+efb*(hliq-Get_hn(k-1));
        Vacia_datos(k,psal,Busca_liqs(psal,'h','f',hreal),
        hreal,Busca_liqs(psal,'h','s',hreal),Busca_liqs(psal,'h','v',hreal),informe);
        informe := "";
        ayuda := NIL
      END
    END;
    siguePre := siguePre^.apunt;
  END;
P_finales(TurbinaVar.get_pin);
j := nlineas-2*nt+1;
IF nt > 0 THEN
  BEGIN
    siguePre := baset;
    WHILE siguePre <> NIL DO
      BEGIN
        pcol := siguePre^.get_pin;
        hliq := Busca_campana('p','hf',pcol);
        siguePre^.liqsat(pcol);
        Lista_trampa(j,pcol,Busca_campana('p','t',pcol),
        siguePre^.get_hf,siguePre^.get_sf,siguePre^.get_vf,'CC_TRAM');
        j := j+1;
        s_trampa(j,siguePre^.get_pout,hliq);
        siguePre := siguePre^.apunt;
        j := j+1
      END
    END;
  END;
IF cc > 0 THEN
  BEGIN
    j := 0;
  END;

```

```

calentador := ap_cc;
WHILE calentador <> NIL DO
  BEGIN
    pcol := calentador^.pcic;
    pnec := calentador^.ptur;
    hguia := calentador^.h4;
    ayuda := head_p;
    WHILE ayuda <> NIL DO
      BEGIN
        IF (ayuda^.p = pcol) AND (ayuda^.h = hguia) THEN
          BEGIN
            j := ayuda^.num-1;
            Suma_dato(pnec,Get_hn(j),'h3'.ap_cc);
            ayuda := NIL;
          END
        ELSE
          ayuda := ayuda^.apunt
        END;
      calentador := calentador^.apunt
    END
  END
END;

```

```

PROCEDURE Despliega(VAR i : integer; resultado : PuntoPtr);
VAR
  next_p : PuntoPtr;

```

```

BEGIN
  GotoXY(5,2); Write('Pto');
  GotoXY(12,2); Write('P (Bar)');
  GotoXY(23,2); Write('T (C)');
  GotoXY(34,2); Write('h (KJ/Kg)');
  GotoXY(48,2); Write('s (KJ/Kg_C)');
  GotoXY(60,2); Write('v (cm^3/g)');
  next_p := resultado;
  WHILE next_p <> NIL DO
    BEGIN
      GotoXY(5,i); Write(next_p^.num);
      GotoXY(70,i); Write(next_p^.enlace);
      GotoXY(12,i); Write(next_p^.p:5:2);
      GotoXY(23,i); Write(next_p^.t:5:2);
      GotoXY(34,i); Write(next_p^.h:5:4);
      GotoXY(48,i); Write(next_p^.s:5:4);
      GotoXY(58,i); Write(next_p^.v:5:4);
      i := i+1;
      IF i = 16 THEN
        BEGIN
          Readln;
          ClrScr;
        END
      END
    END
  END

```

```

        l := 4
      END;
      next_p := next_p^.apunt
    END;
  END;

```

```

PROCEDURE Dibuja_campana(temmax, smax : real);

```

```

VAR

```

```

  xmax, ymax, cont : Integer;
  sv, tv, ou       : String;
  ux, uy, n, facx, facy : Real;

```

```

BEGIN

```

```

  xmax := GetMaxX;
  ymax := GetMaxY;
  ux := GetMaxX/100;
  uy := GetMaxY/75;
  facx := (xmax-20*ux)/smax;
  facy := (ymax-10*uy)/temmax;
  n := 28.96;
  SetBkColor(blue);
  REPEAT
    PutPixel(Round(xmax-90*ux+facx*busca_campana('t','sf',n)),
             Round(ymax-5*uy-facy*n), white);
    PutPixel(Round(xmax-90*ux+facx*busca_campana('t','sg',n)),
             Round(ymax-5*uy-facy*n), white);
    n := n+3;
  UNTIL n >= 374.1;
  Line(Round(xmax-90*ux), Round(ymax-5*uy),
       Round(xmax-90*ux), Round(ymax-70*uy));
  FOR cont := 1 TO 4 DO
    BEGIN
      Str(cont*temmax/4:3:1, sv);
      OutTextXY(Round(xmax-95*ux), Round(ymax-16.25*cont*uy-5*uy), sv)
    END;
  Line(Round(xmax-90*ux), Round(ymax-5*uy),
       Round(xmax-9*ux), Round(ymax-5*uy));
  FOR cont := 1 TO 4 DO
    BEGIN
      Str(cont*smax/4:3:2, sv);
      OutTextXY(Round(xmax+10*ux-20*(5-cont)*ux), Round(ymax-5*uy), 'R');
      OutTextXY(Round(xmax+10*ux-20*(5-cont)*ux), Round(ymax-3*uy), sv)
    END;
  OutTextXY(Round(xmax-93*ux), Round(ymax-72*uy), ' T ');
  OutTextXY(Round(xmax-7*ux), Round(ymax-5*uy), ' s ')
END;

```

```

PROCEDURE Poner_campana(punt : PuntoPtr);

```

```

VAR
mostrar,pp                               : PuntoPtr;
xmax,ymax,a,b                             : Integer;
ux,uy,facx,facy,smax,temmax              : Real;
mpres,npres                               : PresionPtr;
sv,tv,ou                                  : String;

```

```
BEGIN
```

```

Dibuja_campana(TurbinaVar.get_tin+30,9);
xmax := GetMaxX;
ymax := GetMaxY;
ux := GetMaxX/100;
uy := GetMaxY/75;
temmax := TurbinaVar.get_tin+30; smax:=9;
facx := (xmax-20*ux)/smax; facy:=(ymax-10*uy)/temmax;
mostrar := punt;
pp := mostrar^.apunt;
a := Round(xmax-90*ux+facx*mostrar^.s);
b := Round(ymax-5*uy-facy*mostrar^.t);
  WHILE mostrar <> NIL DO
    BEGIN
      Putpixel(a,b,yellow);
      Str(pp^.num,ou);
      Circle(Round(xmax-90*ux+facx*pp^.s),Round(ymax-5*uy-
facy*pp^.t),Round(0.3*ux));
      SetColor(14);
      Line(a,b,Round(xmax-90*ux+facx*pp^.s),Round(ymax-5*uy-facy*pp^.t));
      OutTextXY(Round(xmax-91*ux+facx*pp^.s),Round(ymax-4*uy-facy*pp^.t),ou);
      a := Round(xmax-90*ux+facx*pp^.s);
      b := Round(ymax-5*uy-facy*pp^.t);
      mostrar := mostrar^.apunt;
      pp := pp^.apunt;
      IF pp = NIL THEN pp := punt
    END;
    pp := NIL;
    IF nt > 0 THEN
      BEGIN
        mostrar := guia_t;
        WHILE mostrar <> NIL DO
          BEGIN
            Str(mostrar^.num,ou);
            Circle(Round(xmax-90*ux+facx*mostrar^.s),
              Round(ymax-5*uy-facy*mostrar^.t),Round(0.3*ux));
            OutTextXY(Round(xmax-91*ux+facx*mostrar^.s),
              Round(ymax-4*uy-facy*mostrar^.t),ou);
            a := Round(xmax-90*ux+facx*mostrar^.s);
            b := Round(ymax-5*uy-facy*mostrar^.t);
            mostrar := mostrar^.apunt;
            Str(mostrar^.num,ou);
            Circle(Round(xmax-90*ux+facx*mostrar^.s),

```

```

        Round(ymax-5*uy-facy*mostrar^.t),Round(0.3*ux));
Line(a,b,Round(xmax-90*ux+facx*mostrar^.s),
    Round(ymax-5*uy-facy*mostrar^.t));
OutTextXY(Round(xmax-91*ux+facx*mostrar^.s),
    Round(ymax-4*uy-facy*mostrar^.t),ou);
    mostrar := mostrar^.apunt
END
END;
END;

```

```

PROCEDURE Campana_archivo(nombre: Str12);
VAR
xmax,ymax,a,b,c,d,j,ayu1,ayu2,fin1,fin2 : Integer;
s,saux                                     : String;
ultimo                                     : Boolean;
ux,uy,temmax,smax,facx,facy              : Real;

BEGIN
ASSIGN(Archivo,nombre);
RESET(Archivo);
Read(Archivo,VarPto);
Initg;
    xmax := GetMaxX;
    ymax := GetMaxY;
    ux := GetMaxX/100;
    uy := GetMaxY/75;
    temmax := VarPto.t+30; smax := 9;
    facx := (xmax-20*ux)/smax; facy := (ymax-10*uy)/temmax;
    a := Round(xmax-90*ux+facx*VarPto.s); ayu1 := a;
    b := Round(ymax-5*uy-facy*VarPto.t); ayu2 := b;
    ultimo := FALSE;
    Str(VarPto.num,s);
    Dibuja_campana(VarPto.t+30,9);
    WHILE NOT EOF(Archivo) DO
        BEGIN
            Read(Archivo,VarPto);
            IF ultimo = FALSE THEN
                BEGIN
                    c := Round(xmax-90*ux+facx*VarPto.s);
                    d := Round(ymax-5*uy-facy*VarPto.t);
                    Circle(Round(xmax-90*ux+facx*VarPto.s),Round(ymax-5*uy-facy*
                        VarPto.t),Round(0.3*ux));
                    OuttextXY(a+3,b-6,s);
                    Str(VarPto.num,s);
                    saux := s;
                    LINE(a,b,c,d);
                    a := c; fin1 := a;
                    b := d; fin2 := b;
                    IF VarPto.num = -2 THEN ultimo := TRUE

```

```

END
ELSE
BEGIN
  Str(VarPto.num,s);
  a := Round(xmax-90*ux+facx*VarPto.s);
  b := Round(ymax-5*uy-facy*VarPto.t);
  Circle(Round(xmax-90*ux+facx*VarPto.s),Round(ymax-5*uy-facy*
    VarPto.t),Round(0.3*ux));
  OuttextXY(a+3,b-6,s);
  Read(Archivo,VarPto);
  Str(VarPto.num,s);
  c := Round(xmax-90*ux+facx*VarPto.s);
  d := Round(ymax-5*uy-facy*VarPto.t);
  Circle(Round(xmax-90*ux+facx*VarPto.s),Round(ymax-5*uy-facy*
    VarPto.t),Round(0.3*ux));
  OuttextXY(c+3,d-6,s);
  LINE(a,b,c,d)
END
END;
OuttextXY(fin1+3,fin2-6,saux);
Circle(Round(xmax-90*ux+facx*ayu1),Round(ymax-5*uy-
facy*ayu2),Round(0.3*ux));
LINE(fin1,fin2,ayu1,ayu2);
Readln;
Closegraph;
CLOSE(Archivo);
END;

```

PROCEDURE Add_flujo(cont: integer; flux,pturbina : real);

```

BEGIN
  IF h_flujo = NIL THEN
    BEGIN
      NEW(h_flujo);
      h_flujo^.n := cont;
      h_flujo^.yn := flux;
      h_flujo^.p := pturbina;
      h_flujo^.apunt := NIL;
      t_flujo := h_flujo
    END
  ELSE
    BEGIN
      NEW(t_flujo^.apunt);
      IF h_flujo = t_flujo THEN h_flujo^.apunt:=t_flujo^.apunt;
      t_flujo := t_flujo^.apunt;
      t_flujo^.n := cont;
      t_flujo^.yn := flux;
      t_flujo^.p := pturbina;
      t_flujo^.apunt := NIL
    END
  END

```

```

END
END;

```

```

PROCEDURE M_flujos;

```

```

VAR

```

```

    siguefl : FlujoPtr;

```

```

BEGIN

```

```

    siguefl := h_flujo;

```

```

    WHILE siguefl <> NIL DO

```

```

        BEGIN

```

```

            GotoXY(4,20+siguefl^.n); Write(siguefl^.p,' ');

```

```

            GotoXY(10,20+siguefl^.n); WriteLn(siguefl^.yn:5:4);

```

```

            siguefl := siguefl^.apunt

```

```

        END;

```

```

    ReadLn

```

```

END;

```

```

FUNCTION Suma_flujos: REAL;

```

```

VAR

```

```

    siguefl : FlujoPtr;

```

```

    contar : Real;

```

```

BEGIN

```

```

    siguefl := h_flujo;

```

```

    contar := 0;

```

```

    WHILE siguefl <> NIL DO

```

```

        BEGIN

```

```

            contar := contar+siguefl^.yn;

```

```

            siguefl := siguefl^.apunt

```

```

        END;

```

```

    Suma_flujos := contar

```

```

END;

```

```

FUNCTION Get_flujo(pturbina : Real): Real;

```

```

VAR

```

```

    siguefl : FlujoPtr;

```

```

    aux : Real;

```

```

BEGIN

```

```

    aux := 0;

```

```

    siguefl := h_flujo;

```

```

    WHILE siguefl <> NIL DO

```

```

        BEGIN

```

```

            IF siguefl^.p = pturbina THEN

```

```

                BEGIN

```

```

                    aux := siguefl^.yn;

```

```

        siguefl := NIL
    END
    ELSE siguefl := siguefl^.apunt
END;
IF aux = 0 THEN aux := 1;
Get_flujo := aux
END;

```

```

FUNCTION Acumfl(i:integer) : REAL;
VAR
siguefl : FlujoPtr;
contar : Real;
j      : Integer;

BEGIN
j := 0;
contar := 1;
siguefl := h_flujo;
    WHILE siguefl <> NIL DO
        BEGIN
            IF j = i THEN siguefl := NIL
            ELSE
                BEGIN
                    contar := contar-siguefl^.yn;
                    j := j+1;
                    siguefl := siguefl^.apunt
                END
            END;
        END;
    Acumfl := contar
END;

```

```

FUNCTION Fcorr: REAL;
VAR
siguefl : FlujoPtr;
contar : Real;
i,j      : Integer;

BEGIN
siguefl := h_flujo;
i := ABS(cc-nt);
j := 0;
contar := 1;
    WHILE siguefl <> NIL DO
        BEGIN
            IF j = i THEN siguefl := NIL
            ELSE
                BEGIN
                    contar := 1-siguefl^.yn;

```



```

        j := j+1;
        siguefl := siguefl^.apunt
    END
END;
fcorr := contar
END;

```

```

PROCEDURE Vacia_flujos(cabeza : FlujoPtr);
VAR
    tempr1, tempr2 : FlujoPtr;

```

```

BEGIN
    tempr1 := cabeza;
    WHILE tempr1 <> NIL DO
        BEGIN
            tempr2 := tempr1;
            tempr1 := tempr1^.apunt;
            DISPOSE(tempr2)
        END
    END
END;

```

```

PROCEDURE Flujos;

```

```

VAR
    siguePre, ayuda          : PresionPtr;
    siguePto                 : PuntoPtr;
    siguecc                  : c_cerradoPtr;
    i                        : Integer;
    t,yx,yob,yot,b,factb,factt : Real;
    hcic,htur,hot,m,hsum,hcom : Real;
    st                       : Str5;

```

```

BEGIN

```

```

    Init_flujo(h_flujo,t_flujo);
    factb := 1;
    yob := 1;
    hcic := 0;
    htur := 0;
    factt := 1;
    i := 0;
    IF cc > 0 THEN
        BEGIN
            siguecc := ap_cc;
            st := "";
            WHILE siguecc <> NIL DO
                BEGIN
                    i := i+1;
                    b := siguecc^.ptur;
                    siguePre := baseb;

```

```

WHILE siguePre <> NIL DO
  BEGIN
    IF siguePre^.get_pin = b THEN
      BEGIN
        st := 'bomba';
        siguePre := NIL;
      END
    ELSE siguePre := siguePre^.apunt
    END;
  IF st = 'bomba' THEN
    BEGIN
      yx := (siguecc^.h4-siguecc^.h3)*factb/
        (siguecc^.h4-siguecc^.h3+siguecc^.h1-siguecc^.h2);
      factb := yob-yx;
      yob := factb
    END;
    IF st = " THEN
      BEGIN
        yx := (siguecc^.h4-siguecc^.h3)*fact/
          (siguecc^.h1-siguecc^.h2);
        IF factb <> 1 THEN fact := factb;
        END;
        Add_flujo(i,yx,b);
        siguecc := siguecc^.apunt
      END
    END;
  IF ca > 0 THEN
    BEGIN
      siguePre := baseca;
      WHILE siguePre <> NIL DO
        BEGIN
          i := i+1;
          b := siguePre^.get_pin;
          siguePto := head_p;
          WHILE siguePto <> NIL DO
            BEGIN
              IF siguePto^.p = b THEN
                BEGIN
                  t := Busca_campana('p',t,b);
                  IF siguePto^.t < t THEN
                    BEGIN
                      hcic := siguePto^.h;
                      siguePto := siguePto^.apunt;
                    END;
                  IF siguePto^.t > t THEN
                    BEGIN
                      htur := siguePto^.h;
                      siguePto := siguePto^.apunt;
                    END;
                  IF siguePto^.t = t THEN

```

```

        BEGIN
            hot := siguePto^.h;
            siguePto := siguePto^.apunt;
        END
    END
ELSE
    siguePto := siguePto^.apunt;
END;
hsum := 0;
IF nt > 0 THEN
    BEGIN
        siguePto := guia_t;
        WHILE siguePto <> NIL DO
            BEGIN
                t := siguePto^.p;
                hcom := siguePto^.h;
                IF t = b THEN
                    BEGIN
                        ayuda := baset;
                        WHILE ayuda <> NIL DO
                            BEGIN
                                IF (ayuda^.get_pout = t) AND
                                    (ayuda^.get_hf = hcom) THEN
                                    BEGIN
                                        m := ayuda^.get_pin;
                                        ayuda := NIL
                                    END
                                ELSE ayuda := ayuda^.apunt
                                END;
                                hsum := hsum+get_flujo(m)*siguePto^.h;
                                siguePto := siguePto^.apunt
                            END
                        ELSE
                            siguePto := siguePto^.apunt
                        END
                    END
                END
            END
        END;
    END;
yot := 1-Suma_flujos;
IF nt = 0 THEN yx := (siguePre^.get_hf-hcic)*yo/(htur-hcic);
IF nt > 0 THEN
    BEGIN
        yx := (siguePre^.get_hf*Fcorr-hsum-hcic*yot)/(htur-hcic);
        IF yx < 0 THEN
            BEGIN
                ClrScr;
                Ilu_Str(' Flujo del C.A. negativo, Ciclo Termodinamico Imposible',10,5);
                Ilu_Str(' Se tomará el valor absoluto del flujo para seguir',10,6);
                yx := ABS(yx);
                Readln
            END
        END
    END;
END;

```

```

        Add_flujo(i,yx,b);
        siguePre := siguePre^.apunt
    END
END
END;

```

```

PROCEDURE Eficiencia;

```

```

VAR

```

```

pncop,i,mismp,j      : Integer;
siguePre,ayudaPre   : PresionPtr;
siguePto            : PuntoPtr;
sigueecc            : c_cerradoPtr;
facf,m,n,hpar,loc,g : Real;
indic               : Boolean;

```

```

BEGIN

```

```

    pncop := opcic;
    IF pncop = 11 THEN pncop := 10;
    CASE pncop OF

```

```

    8 : BEGIN

```

```

        tt := get_hn(1)-get_hn(2);
        tb := get_hn(4)-get_hn(3);
        wn := tt-tb;
        qr := get_hn(1)-get_hn(4);
        eta := wn/qr

```

```

    END;

```

```

    9 : BEGIN

```

```

        IF recal = FALSE THEN
        BEGIN
            tt := get_hn(1)-get_hn(2);
            tb := get_hn(4)-get_hn(3);
            wn := tt-tb;
            qr := get_hn(1)-get_hn(4);
            eta := wn/qr

```

```

        END

```

```

        ELSE

```

```

        BEGIN

```

```

            tt := get_hn(1)-get_hn(2)+get_hn(3)-get_hn(4);
            tb := get_hn(6)-get_hn(5);
            wn := tt-tb;
            qr := get_hn(1)-get_hn(6)+get_hn(3)-get_hn(2);
            eta := wn/qr

```

```

        END

```

```

    END;

```

```

    10 : BEGIN

```

```

Flujos;
IF recal = FALSE THEN
  BEGIN
    siguePto := head_p;
    i := 0;
    factl := 1;
    tt := 0;
    WHILE siguePto <> NIL DO
      BEGIN
        i := i+1;
        tt := tt+(get_hn(i)-get_hn(i+1))*factl;
        IF i = cc+ca+1 THEN siguePto := NIL
          ELSE
            BEGIN
              siguePto := siguePto^.apunt;
              factl := factl-Get_flujo(siguePto^.p)
            END
          END;
        i := nlineas-2*nt;
        qr := get_hn(1)-get_hn(i);
      END
    {recal=t} ELSE
      BEGIN
        tt := 0;
        m := TurbinaBaja.get_pin;
        n := TurbinaBaja.get_tin;
        j := 0; i := 0; factl := 1;
        siguePre := basecc;
        WHILE siguePre <> NIL DO
          BEGIN
            IF siguePre^.get_pin = m THEN
              j := j+1;
              siguePre := siguePre^.apunt
            END;
            siguePto := head_p;
            WHILE siguePto <> NIL DO
              BEGIN
                IF (siguePto^.p = m) AND (siguePto^.t = n) THEN
                  BEGIN
                    mismp := siguePto^.num-1;
                    hpar := Get_hn(mismp);
                    siguePto := NIL
                  END
                ELSE siguePto := siguePto^.apunt
              END;
            {mismp nos da pto salida al recalentamiento }
            IF j = 0 THEN
              BEGIN
                siguePto := head_p;
                WHILE siguePto <> NIL DO

```

```

BEGIN
  i := i+1;
  tt := tt+(get_hn(i)-get_hn(i+1))*facfl;
  IF i+1 = mismp THEN
    BEGIN
      i := 1+i;
      siguePto := siguePto^.apunt
    END;
  IF i = cc+ca+3 THEN siguePto := NIL
  ELSE
    BEGIN
      siguePto := siguePto^.apunt;
      IF Get_flujo(siguePto^.p) <> 1 THEN
        facfl := facfl-Get_flujo(siguePto^.p)
      ELSE facfl := facfl
    END
  END
END
ELSE
BEGIN
  siguePto := head_p;
  j := 1;
  WHILE siguePto <> NIL DO
    BEGIN
      i := i+1;
      tt := tt+(Get_hn(i)-Get_hn(i+j))*facfl;
      IF i+1 = mismp THEN
        BEGIN
          i := 1+i;
          siguePto := siguePto^.apunt;
          loc := siguePto^.h;
          IF loc = hpar THEN
            BEGIN
              j := 2;
              siguePto := siguePto^.apunt
            END
          ELSE j := 1
        END;
      IF i = mismp+1 THEN
        BEGIN
          i := i+1;
          j := 1;
          siguePto := siguePto^.apunt
        END;
      IF i = cc+ca+3 THEN siguePto := NIL
      ELSE
        BEGIN
          IF Get_flujo(siguePto^.p) <> 1 THEN
            facfl := facfl-Get_flujo(siguePto^.p)
          ELSE facfl := facfl;
        END
      END
    END
  END

```

```

        siguePto := siguePto^.apunt
    END
END
END;
i := niineas-2*nt;
mismp := 0;
siguePre := basecc;
WHILE siguePre <> NIL DO
    BEGIN
        IF siguePre^.get_pin = TurbinaBaja.get_pin
            THEN mismp := mismp+1;
            siguePre := siguePre^.apunt
        END;
    qr := get_hn(1)-get_hn(i);
    siguePto := head_p;
    WHILE siguePto <> NIL DO
        BEGIN
            IF (siguePto^.p = m) AND (siguePto^.t = n) THEN
                BEGIN
                    i := siguePto^.num-3+mismp;
                    facfl := Acumfl(i);
                    qr := qr+(get_hn(siguePto^.num)-get_hn(siguePto^.num-1))*facfl;
                    siguePto := NIL
                END
            ELSE siguePto := siguePto^.apunt
            END
        END;
    siguePre := baseb;
    tb := 0;
    g := 0;
    facfl := 1-suma_flujos;
    g := facfl;
    WHILE siguePre <> NIL DO
        BEGIN
            n := siguePre^.get_pin;
            siguePre^.liqsat(n);
            IF cc > 0 THEN
                BEGIN
                    siguecc:=ap_cc;
                    WHILE siguecc <> NIL DO
                        BEGIN
                            IF siguecc^.ptur = n THEN
                                BEGIN
                                    facfl := Get_flujo(n);
                                    siguecc := NIL
                                END
                            ELSE siguecc := siguecc^.apunt
                            END
                        END
                    END
                END;
            IF nt > 0 THEN

```

```

BEGIN
  ayudaPre := baset;
  WHILE ayudaPre <> NIL DO
    BEGIN
      IF ayudaPre^.get_pout = n THEN
        BEGIN
          factl := factl+Get_flujo(ayudaPre^.get_pin);
          g := factl;
          ayudaPre := ayudaPre^.apunt;
        END
      ELSE ayudaPre := ayudaPre^.apunt
      END
    END;
    tb := tb+((siguePre^.get_pout-siguePre^.get_pin)*
siguePre^.get_vf/10)*factl;
    m := Get_flujo(siguePre^.get_pout);
    IF m = 1 THEN factl := g
    ELSE factl := factl+m;
    g := factl;
    siguePre := siguePre^.apunt;
  END;
  wn := tt-tb;
  eta := wn/qr;
END
END
END;

```

PROCEDURE Potencias;

```

VAR
nomb           : Str8;
mensaje,up     : Str5;
i              : Integer;
valor,pot,fagua,k : Real;
siguePto      : PuntoPtr;

```

BEGIN-

```

  ClrScr;
  pot := 0;
  IF opcic in [8,9] THEN k := 1;
  IF opcic in [10,11] THEN k := 1-Suma_flujos;
  GotoXY(15,3); Write(' Flujo de vapor (Kg/s) ');
  lfu_cant(50,3,7,1,1,15,nomb,valor);
  pot := valor*wn;
  siguePto := head_p;
  i := 0;
  WHILE siguePto <> NIL DO
    BEGIN
      i := i+1;
      IF siguePto^.p = CondensadorVar.get_pin THEN

```



```

BEGIN
- siguePto := NIL
END
ELSE
siguePto := siguePto^.apunt
END;
fagua := k*valor*(Get_hn(i)-Get_hn(i+1))/(cp*deltat);
GotoXY(15,7); Write('Potencia de Turbina ');
GotoXY(15,8); Write('Flujo de agua de enfriamiento ');
Resalta(50,7,7,1,1,15,pot);
Resalta(50,8,7,1,1,15,fagua);
GotoXY(65,7); Write('KW');
GotoXY(65,8); Write('Kg/s');
ReadLn;
END;

```

BEGIN { PRINCIPAL}

```

ClrScr;
Inicia_lista;
Init_presion;
init_flujo(h_flujo,t_flujo);

REPEAT
ClrScr;
Opven_bis(7,1,1,14);
GotoXY(6,2); Write('UNIVERSIDAD PANAMERICANA ');
GotoXY(6,3); Write('Escuela de Ingenier_a ');
GotoXY(45,2); Write('Soluci_n de Ciclos de vapor ');
GotoXY(45,3); Write('Utilice ',#24,' y ',#25,' para moverse ');
Clven_bis(7,1);
GotoXY(20,3); Write('Iniciar un nuevo ciclo.....[ _ ] ');
GotoXY(20,4); Write('Cargar reporte existente....[ ] ');
GotoXY(20,5); Write('Editar Parámetros.....[ ] ');
GotoXY(20,6); Write('Soluci_n Num_rica.....[ ] ');
GotoXY(20,7); Write('Soluci_n Gráfica.....[ ] ');
GotoXY(20,8); Write('Dar Trabajos.....[ ] ');
GotoXY(20,9); Write('Potencia de Turbina.....[ ] ');
GotoXY(20,10); Write('Seguir adelante.....[ ] ');
GotoXY(40,12); WriteLn(MemAvail);
GotoXY(50,3);
Mueve_marca(50,3,10,#219);
rutina := WhereY;

CASE rutina OF

3 : BEGIN

```

```

captura;
init_param
END;

```

```

4 : BEGIN

```

```

    sha := 10;
    nomArch := "";
    ClrScr;
    GotoXY(3,2); Write(' Nombre de archivo ');
    llu_Str('CPT',32,2);
    llu_Str(' ',24,2);
    TextBackGround(1);
    TextColor(15);
    GotoXY(24,2);
    Readln(nomArch);
    TextColor(1);
    TextBackGround(7);
    {GotoXY(24,2); Readln(nomArch);}
    {$!}
    ASSIGN(Datext,nomArch+'.DAT');
    RESET(Datext);
    IF IORresult = 0 THEN
        BEGIN
            RESET(Datext);
            Read(Datext,varInfo);
            GotoXY(5,3); Write('C. Cerrados = ',varInfo.ccer);
            GotoXY(25,3); Write('C. Abiertos = ',varInfo.cabi);
            GotoXY(45,3); Write('Bombas = ',varInfo.bomb);
            GotoXY(60,3); Write('Trampas = ',varInfo.tram);
            GotoXY(5,4);
            Write('SISTEMA METRICO');
            GotoXY(5,8);
            Write('Pto P (bar)   T (C)   h (KJ/Kg)   s (KJ/Kg_C)   v (cm^3/g) ');
            GotoXY(25,4);
            CASE varInfo.cicl OF
                8 : nomcic := 'CICLO DE CARNOT';
                9 : nomcic := 'CICLO RANKINE';
                10 : nomcic := 'CICLO REGENERATIVO';
                11 : nomcic := 'CICLO OPTIMO'
            END;
            Write(nomcic);
            GotoXY(5,5); Write('W neto = ',VarInfo.wneto:5:3);
            GotoXY(25,5); Write('Q sum = ',VarInfo.qsum:5:3);
            GotoXY(45,5); Write('Eficiencia = ',VarInfo.efic:5:4);
            GotoXY(5,6); Write('Ef. Turbina = ',VarInfo.etatur:3:2);
            GotoXY(25,6); Write('Ef. Bombas = ',VarInfo.etabom:3:2);
            GotoXY(45,6);
            IF recal = TRUE THEN
                Write('EXISTE RECALENTAMIENTO')
            ELSE Write("");

```

```

CLOSE(Datext);
ASSIGN(Archivo,nomArch+'.CPT');
RESET(Archivo);
  WHILE NOT EOF(Archivo) DO
    BEGIN
      Read(Archivo,varPto);
      GotoXY(5,sha); Write(varPto.num);
      GotoXY(10,sha); Write(varPto.p:5:2);
      GotoXY(21,sha); Write(varPto.t:5:2);
      GotoXY(34,sha); Write(varPto.h:5:4);
      GotoXY(48,sha); Write(varPto.s:5:4);
      GotoXY(58,sha); Write(varPto.v:5:4);
      GotoXY(70,sha); Write(varPto.enlace);
      sha := sha+1;
      IF sha = 18 THEN
        BEGIN
          Readln;
          Clrscr;
          GotoXY(5,2);
          sha := 4;
        Write('Pto P T h s v ')
        END;
      END;
      Writeln; Writeln;
      Write(' Deseas imprimir el reporte s/n ');
      IF UPCASE(gdat) = 'S' THEN
        BEGIN
          Writeln(ist,'Archivo '+nomArch+' SISTEMA METRICO');
          Writeln(ist,nomcic);
          Writeln(ist,'C. cerrados ',varinfo.ccer,' C. abiertos ',varinfo.cabi);
          Writeln(ist,'Bombas ',varinfo.bomb,' Trampas ',varinfo.tram);
          Writeln(ist,'W neto (KJ/Kg) ',varinfo.wneto:5:3);
          Writeln(ist,'Q suministrado (KJ/Kg) ',varinfo.qsum:5:3);
          Writeln(ist,'Eficiencia ',varinfo.wneto:5:3);
          Writeln(ist,' ');
          Write('P T h s v ');
          Writeln(ist,' ');
          ASSIGN(Archivo,nomArch+'.CPT');
          RESET(Archivo);
          WHILE NOT EOF (Archivo) DO
            BEGIN
              Read(Archivo,varPto);
              Writeln(ist,varPto.num,' ',varPto.p:5:2,' ',varPto.t:5:2,' ',varPto.h:5:2,' ',
                varPto.s:5:4,' ',varPto.v:5:4);
              Writeln(ist,varPto.enlace);
            END
          END
          CLOSE(Archivo);
          Writeln; Writeln;
          Write(' Deseas ver el diagrama T-s s/n ');
        }grafico}

```

```

        llu_chr(gdat,WhereX,WhereY,7,1,1,15);
        IF UPCASE(gdat) = 'S' THEN
            BEGIN
                Campana_Archivo(nomArch+'.CPT')
            END;
        END
    ELSE
        BEGIN
            llu_Str(' No existe el archivo ',10,7);
            Readln
        END;
    }$!+}
END;
-
5 : BEGIN
    parametros
END;
6 : BEGIN
    nomArch:=";
    IF head_p <> NIL THEN
        BEGIN
            Vacia_puntos(head_p)
        END;
    IF guia_t <> NIL THEN
        BEGIN
            Vacia_puntos(guia_t);
        END;
    Inicia_lista;
    IF opcic = 8 THEN Camot;
    IF opcic = 9 THEN Rankine;
    IF opcic = 10 THEN Regenerativo;
    IF opcic = 11 THEN Regenerativo;
    ClrScr;
    sha := 4;
    Despliege(sha,head_p);
    IF nt>0 THEN
        Despliege(sha,guia_t);
    Readln;
    Eficiencia;
    Write(' Deseas generar un reporte s/n ');
    llu_chr(gdat,WhereX+4,WhereY,7,1,1,15);
    IF UPCASE(gdat) = 'S' THEN
        BEGIN
            WriteLn;
            Write(' Nombre del archivo (Max 8 caracteres) ');
            llu_Str('.CPT',WhereX+8,WhereY);
            llu_Str(' ',WhereX-12,WhereY);
            GotoXY(WhereX-8,WhereY);
            TextBackGround(1);

```

```

TextColor(15);
Readln(nomArch);
TextColor(1);
TextBackground(7);
{$I-}
  cadena := FALSE;
  adel := TRUE;
  REPEAT
    MKDIR('ELPEMC');
    IF IOResult = 0 THEN
      BEGIN
        RMDIR('ELPEMC');
        cadena := FALSE;
        adel := TRUE
      END
    ELSE
      BEGIN
        Ilu_Str(' Error en la unidad ',32,16);
        Ilu_Str(' Deseas Reintentar s/n ? ',32,17);
        Ilu_Chr(cin,59,17,7,1,1,15);
        IF UPCASE(cin) = 'S' THEN cadena := TRUE;
        ELSE
          BEGIN
            cadena := FALSE;
            adel := FALSE;
          END;
        END
      UNTIL cadena = FALSE;
    {$I+}
    IF adel = TRUE THEN BEGIN
      ASSIGN(Archivo,nomArch+'.CPT');
      REWRITE(Archivo);
      FormaPto := head_p;
      WHILE FormaPto <> NIL DO
        BEGIN
          varPto.num := FormaPto^.num;
          varPto.p := FormaPto^.p;
          varPto.t := FormaPto^.t;
          varPto.h := FormaPto^.h;
          varPto.s := FormaPto^.s;
          varPto.v := FormaPto^.v;
          varPto.enlace := FormaPto^.enlace;
          Write(Archivo,varPto);
          FormaPto := FormaPto^.apunt
        END;
      IF nt > 0 THEN
        BEGIN
          FormaPto := guia_t;
          WHILE FormaPto <> NIL DO
            BEGIN

```

```

varPto.num := FormaPto^.num;
varPto.p := FormaPto^.p;
varPto.t := FormaPto^.t;
varPto.h := FormaPto^.h;
varPto.s := FormaPto^.s;
varPto.v := FormaPto^.v;
varPto.enlace := FormaPto^.enlace;
Write(Archivo,varPto);
FormaPto := FormaPto^.apunt
END
END;
CLOSE(Archivo);
ASSIGN(Datext,nomArch+'.DAT');
REWRITE(Datext);
varInfo.cicl := opcic;
varInfo.ccer := cc;
varInfo.cabi := ca;
varInfo.bomb := nb;
varInfo.tram := nt;
varInfo.wneto := wn;
varInfo.qsum := qr;
varInfo.efic := eta;
varInfo.etabom := efb;
varInfo.etatur := eft;
varInfo.rec := recal;
Write(Datext,varInfo);
CLOSE(Datext)
END
END
END;

7 : BEGIN
  initg;
  Poner_campana(head_p);
  ReadIn;
  Closegraph
END;

8 : BEGIN
  ClrScr;
  GotoXY(20,4); Write('Tt = ',tt:5:3,' KJ/Kg');
  GotoXY(20,5); Write('Tb = ',tb:5:3,' KJ/Kg');
  GotoXY(20,6); Write('Tn = ',wn:5:3,' KJ/Kg');
  GotoXY(20,7); Write('Qs = ',qr:5:3,' KJ/Kg');
  GotoXY(20,8); Write('Ef = ',eta:5:4);
  ReadLn;
END;

9 : BEGIN
  Potencias

```

ANEXO 3

UNIDAD ARCHIV.TPU

END
END;

UNTIL rutina=10;
ClrScr;

END.

UNIT ARCHIV;

INTERFACE

TYPE

```
Str2 = String[2];
```

```
Registro1 = RECORD
```

```
  h,s,v,p,t : real;
```

```
END;
```

```
Registro2 = RECORD
```

```
  hf,hg,sf,sg,vf,vg,p,t : real;
```

```
END;
```

VAR

```
Archivo1 : FILE of Registro1;
```

```
Archivo2 : FILE of Registro2;
```

```
Valor1   : Registro1;
```

```
Valor2   : Registro2;
```

```
Function Busca_campana(letra1,letra2: Str2; VAR x: real): REAL;
```

```
Function Busca_dato(VAR pres: real; let1,let2: char; VAR val: real):REAL;
```

```
Function Busca_liqs(VAR pres: real; let1,let2: char; VAR val: real):REAL;
```

```
Function Calidad(dato: integer; un,uf,ug: real): REAL;
```

IMPLEMENTATION

```
FUNCTION Busca_campana(letra1,letra2: str2; VAR x: real): REAL;
```

```
VAR
```

```
  x2,y2 : real;
```

```
  x1,y1 : real;
```

```
  i     : integer;
```

```
  flat  : boolean;
```

```
  dlr   : integer;
```

```
PROCEDURE asigna_c(ch1 :str2; VAR n:real);
```

```
BEGIN
```

```
  Read(archivo2,valor2);
```

```
  WITH valor2 DO
```

```

BEGIN
  IF ch1='p' THEN n:=p;
  IF ch1='t' THEN n:=t;
  IF ch1='hf' THEN n:=hf;
  IF ch1='hg' THEN n:=hg;
  IF ch1='sf' THEN n:=sf;
  IF ch1='sg' THEN n:=sg;
  IF ch1='vf' THEN n:=vf;
  IF ch1='vg' THEN n:=vg;
END
END; .

```

```

PROCEDURE corre_rutina;
BEGIN
  i:=0;
  flat:=false;
  REPEAT
    Seek(archivo2,i);
    WITH valor2 DO
      BEGIN
        Read(archivo2,valor2);
        IF letra1='p' THEN
          BEGIN
            x2:=p;
            dirr:=1
          END;
        IF letra1='t' THEN
          BEGIN
            x2:=t;
            dirr:=1
          END;
        IF letra1='hf' THEN
          BEGIN
            x2:=hf;
            dirr:=1
          END;
        IF letra1='hg' THEN
          BEGIN
            x2:=hg;
            dirr:=3
          END;
        IF letra1='sf' THEN
          BEGIN
            x2:=sf;
            dirr:=1
          END;
        IF letra1='sg' THEN
          BEGIN
            x2:=sg;

```

```

    dirr:=2
  END;
  IF letra1='vf' THEN
    BEGIN
      x2:=vf;
      dirr:=1
    END;
  IF letra1='vg' THEN
    BEGIN
      x2:=vg;
      dirr:=2
    END;
  Seek(archivo2,i);
  asigna_c(letra2,y2)
END;
IF x = x2 THEN
BEGIN
  Busca_campana:=y2;
  flat:=true
END;
IF x > x2 THEN
BEGIN
CASE dirr OF
  1 : BEGIN
    i:=i+1;
    IF i>49 THEN
      BEGIN
        write('Error por arriba');
        Readln;
      END
    END;
  2 : BEGIN
    Seek(archivo2,i-1);
    asigna_c(letra1,x1);
    Seek(archivo2,i-1);
    asigna_c(letra2,y1);
    Busca_campana:=((y2-y1)/(x2-x1))*(x-x1)+y1;
    flat:=true
  . END
END
END;
IF x < x2 THEN
BEGIN
CASE dirr OF
  1 : BEGIN
    Seek(archivo2,i-1);
    asigna_c(letra1,x1);
    Seek(archivo2,i-1);
    asigna_c(letra2,y1);
    Busca_campana:=((y2-y1)/(x2-x1))*(x-x1)+y1;

```

```

        flat:=true
    END;
2 : BEGIN
    i:=i+1;
    IF i>49 THEN
        BEGIN
            write('Error por abajo');
            Readln
        END
    END
END
END
UNTIL flat=true
END;

```

```

BEGIN
    ASSIGN(archivo2,'campana.up');
    RESET(archivo2);
    corre_rutina;
    CLOSE(archivo2);
END;

```

```

FUNCTION Busca_dato(VAR pres: real; let1,let2: char; VAR val: real):real;

```

```

VAR
x1,y1 : real;
x2,y2 : real;
y      : real;
flap   : boolean;

```

```

PROCEDURE asigna(letra1 :char;VAR n :real);

```

```

BEGIN
    Read(archivo1,valor1);
    WITH valor1 DO
        BEGIN
            IF letra1='p' THEN n:=p;
            IF letra1='l' THEN n:=l;
            IF letra1='h' THEN n:=h;
            IF letra1='s' THEN n:=s;
            IF letra1='v' THEN n:=v
        END
    END;

```

```

FUNCTION y_interp(x2,y2,x1,y1,x: real):real;
BEGIN
    y_interp:=((y2-y1)/(x2-x1))*(x-x1)+y1

```

END;

PROCEDURE interpola(i: integer;x2c,y2c: real;letra1,letra2: char);

BEGIN

IF x2c = val THEN

BEGIN

y:=y2c;

flap:=true

END;

IF val > x2c THEN

BEGIN

flap:=false

END;

IF val < x2c THEN

BEGIN

Seek(archivo1,i-1);

asigna(letra1,x1);

Seek(archivo1,i-1);

asigna(letra2,y1);

y:=y_interp(x2c,y2c,x1,y1,val);

flap:=true

END

END;

PROCEDURE checa(pr: real; u: integer; c1,c2:char);

VAR

frang : boolean;

BEGIN

frang:=false;

Seek(archivo1,u);

asigna(c1,x2);

IF val < x2 THEN

BEGIN

WriteLn(' Fuera del rango inferior (vapor sobrecal) ');

frang:=true;

y:=0;

END

ELSE frang:=false;

IF frang=false THEN

BEGIN

REPEAT

Seek(archivo1,u);

IF valor1.p<>pr THEN

BEGIN

WriteLn(' Fuera de rango superior (vapor sobrecal) ');

```

        flap:=true
    END
ELSE
    BEGIN
        Seek(archivo1,u);
        asigna(c1,x2);
        Seek(archivo1,u);
        asigna(c2,y2);
        interpola(u,x2,y2,c1,c2);
        u:=u+1
    END;
UNTIL flap=true
END
END;

```

PROCEDURE calculos(p: real; ch1,ch2: char; x: real);

```

VAR
j   : integer;
k   : integer;
p1  : real;
p2  : real;
xa1 : real;
xa2 : real;
ya1 : real;
ya2 : real;
band : boolean;
frang : boolean;

BEGIN
j:=-1;
IF p=0.06 THEN j:=0;
IF p=0.35 THEN j:=12;
IF p=0.7 THEN j:=24;
IF p=1 THEN j:=36;
IF p=1.5 THEN j:=48;
IF p=3 THEN j:=60;
IF p=5 THEN j:=71;
IF p=7 THEN j:=83;
IF p=10 THEN j:=95;
IF p=15 THEN j:=107;
IF p=20 THEN j:=119;
IF p=30 THEN j:=131;
IF p=40 THEN j:=143;
IF p=60 THEN j:=155;
IF p=80 THEN j:=167;
IF p=100 THEN j:=179;
IF p=120 THEN j:=191;
IF p=140 THEN j:=202;
IF p=160 THEN j:=213;

```

```

IF p=180 THEN j:=224;
IF p=200 THEN j:=235;
IF p=240 THEN j:=246;
IF j in [0,12,24,36,48,60,71,83,95,107,119,131,143,155,167,179,191,
        202,213,224,235,246] THEN
  checa(p,j,ch1,ch2)
ELSE
BEGIN
  j:=0;
  band:=false;
  REPEAT
  Seek(archivo1,j);
  Read(archivo1,valor1);
  IF valor1.p < p THEN
    BEGIN
      j:=j+12;
      band:=false
    END;
  IF valor1.p > p THEN
    BEGIN
      Seek(archivo1,j);
      Read(archivo1,valor1);
      p2:=valor1.p;
      checa(p2,j,ch1,ch2);
      xa2:=x;
      ya2:=y;
      j:=j-12;
      Seek(archivo1,j);
      Read(archivo1,valor1);
      p1:=valor1.p;
      checa(p1,j,ch1,ch2);
      xa1:=x;
      ya1:=y;
      band:=true
    END;
  UNTIL band=true;
  y:=y_interp(p2,ya2,p1,ya1,p)
END
END;

BEGIN
  ASSIGN(archivo1,'vap_scal.up');
  RESET(archivo1);
  calculos(pres,let1,let2,val);
  Busca_dato:=y;
  CLOSE(archivo1)
END;

```

```
FUNCTION Busca_llqs(VAR pres: real; let1,let2: char; VAR val: real):real;
```

```
VAR
```

```
x1,y1 : real;
x2,y2 : real;
y      : real;
flap   : boolean;
```

```
PROCEDURE asigna(letra1 :char;VAR n :real);
```

```
BEGIN
```

```
  Read(archivo1,valor1);
```

```
  WITH valor1 DO
```

```
    BEGIN
```

```
      IF letra1='p' THEN n:=p;
```

```
      IF letra1='t' THEN n:=t;
```

```
      IF letra1='h' THEN n:=h;
```

```
      IF letra1='s' THEN n:=s;
```

```
      IF letra1='v' THEN n:=v
```

```
    END
```

```
END;
```

```
FUNCTION y_interp(x2,y2,x1,y1,x: real):real;
```

```
BEGIN
```

```
  y_interp:=((y2-y1)/(x2-x1))*(x-x1)+y1
```

```
END;
```

```
PROCEDURE interpola(i: integer;x2c,y2c: real;letra1,letra2: char);
```

```
BEGIN
```

```
  IF x2c = val THEN
```

```
    BEGIN
```

```
      y:=y2c;
```

```
      flap:=true
```

```
    END;
```

```
  IF val > x2c THEN
```

```
    BEGIN
```

```
      flap:=false
```

```
    END;
```

```
  IF val < x2c THEN
```

```
    BEGIN
```

```
      Seek(archivo1,i-1);
```

```
      asigna(letra1,x1);
```

```
      Seek(archivo1,i-1);
```

```
      asigna(letra2,y1);
```

```
      y:=y_interp(x2c,y2c,x1,y1,val);
```

```
      flap:=true
```

```
    END
```

```
END;
```



```

PROCEDURE checa(pr: real; u: integer; c1,c2:char);
VAR
  frang : boolean;
  mundo : boolean;

BEGIN
  mundo:=FALSE;
  REPEAT
    Seek(archivo1,u);
    IF u in [0,8,17,26,35,44,54,64,69] THEN mundo:=TRUE
    ELSE u:=u-1;
  UNTIL mundo=TRUE;
  Seek(archivo1,u);
  asigna(c1,x2);
  IF val < x2 THEN
    BEGIN
      WriteLn(' Fuera del rango inferior (liq compr) ');
      frang:=true;
      y:=0
    END
  ELSE frang:=false;
  IF frang=false THEN
    BEGIN
      REPEAT
        Seek(archivo1,u);
        IF valor1.p<>pr THEN
          BEGIN
            WriteLn(' Fuera de rango superior (liq compr) ');
            flap:=true
          END
        ELSE
          BEGIN
            Seek(archivo1,u);
            asigna(c1,x2);
            Seek(archivo1,u);
            asigna(c2,y2);
            interpola(u,x2,y2,c1,c2);
            u:=u+1
          END;
      UNTIL flap=true;
    END
  END;
END;

```

```

PROCEDURE calculos(p: real; ch1,ch2: char; x: real);
VAR
  j : integer;
  k : integer;

```

```

p1   : real;
p2   : real;
xa1  : real;
xa2  : real;
ya1  : real;
ya2  : real;
band : boolean;
frang : boolean;

```

```
BEGIN
```

```

IF p=0 THEN j:=0;
IF p=25 THEN j:=8;
IF p=50 THEN j:=17;
IF p=75 THEN j:=26;
IF p=100 THEN j:=35;
IF p=150 THEN j:=44;
IF p=200 THEN j:=54;
IF p=250 THEN j:=64;
IF p=300 THEN j:=69
ELSE j:=3;
IF j in [0,8,17,26,35,44,54,64,69] THEN
  checa(p,j,ch1,ch2)

```

```
ELSE
```

```
  BEGIN
```

```
    j:=0;
```

```
    band:=false;
```

```
    REPEAT
```

```
      Seek(archivo1,j);
```

```
      Read(archivo1,valor1);
```

```
      IF valor1.p < p THEN
```

```
        BEGIN
```

```
          j:=j+9;
```

```
          band:=false
```

```
        END;
```

```
      IF valor1.p > p THEN
```

```
        BEGIN
```

```
          Seek(archivo1,j);
```

```
          Read(archivo1,valor1);
```

```
          p2:=valor1.p;
```

```
          checa(p2,j,ch1,ch2);
```

```
          xa2:=x;
```

```
          ya2:=y;
```

```
          j:=j-9;
```

```
          Seek(archivo1,j);
```

```
          Read(archivo1,valor1);
```

```
          p1:=valor1.p;
```

```
          checa(p1,j,ch1,ch2);
```

```
          xa1:=x;
```

```
          ya1:=y;
```

```
          band:=true
```

```
        END;  
        UNTIL band=true;  
        y:=y_interp(p2,ya2,p1,ya1,p)  
    END  
END;
```

```
BEGIN  
    ASSIGN(archivo1,'liq_comp.up');  
    RESET(archivo1);  
    calculos(pres,let1,let2,val);  
    Busca_liqs:=y;  
    CLOSE(archivo1);  
END;
```

```
FUNCTION Calidad(dato: integer, un,uf,ug: real): REAL;  
BEGIN  
    CASE dato OF  
        1 : Calidad:=((un-uf)/(ug-uf));  
        2 : Calidad:=uf+un*(ug-uf)  
    END  
END;  
  
END.
```

ANEXO 4

UNIDAD ELEMENTO.TPU

UNIT ELEMENTOS;

INTERFACE

USES crt, archiv;

TYPE

Str2 = String[2];

PuntoPtr = ^Punto;

Punto = RECORD

p,t,v,h,s : real;

num : integer;

enlace : String[8];

apunt : PuntoPtr;

END;

c_cerradoPtr = ^c_cerrado;

c_cerrado = RECORD

h1,h2,h3,h4 : real;

ptur,pcic : real;

apunt : c_cerradoPtr;

END;

FlujoPtr = ^Flujo;

Flujo = RECORD

yn,p : real;

n : integer;

apunt : FlujoPtr

END;

PresionPtr = ^Presion;

Presion = object

p_in,p_out : real;

h_f,s_f,v_f : real;

numero : integer;

apunt : PresionPtr;

Procedure init(pin,pout : real);

Procedure apunta(next : PresionPtr);

Procedure numera(i : integer);

Procedure liqsat(pin : real);

Function get_pin : REAL;

Function get_pout : REAL;

Function get_numero : INTEGER;

Function get_hf : REAL;

Function get_sf : REAL;

Function get_vf : REAL;

END;

Turbina = object

```

p_in,t_in : real;
s_tur      : real;
n          : integer;
Procedure init(pin,t: real);
Procedure salidas(nsal: integer);
Procedure entropia(ps,ts: real);
Function get_pin: REAL;
Function get_tin: REAL;
Function get_s: REAL;
Function get_hov(clave: integer; ps,ss : real): REAL;
end;

```

```

caldera = object
p_in, t_out: real;
recal      : boolean;
procedure init(pin,tout: real; resp: char);
function get_pin: real;
function get_tout: real;
end;

```

```

Condensador = object
p_in,t_in : real;
h_out,s_out,v_out : real;
Procedure init(pin,t: real);
Function get_pin: REAL;
Function get_tin: REAL;
Procedure salidas(pin : real);
Function get_hvs(clave: integer): REAL;
end;

```

```

Procedure Init_cc(VAR head_cc, cola_cc : c_cerradoPtr);
Procedure Crea_cc(pi,po : real; VAR head_cc, cola_cc : c_cerradoPtr);
Procedure Suma_dato(p,x : real; letra : Str2; VAR head_cc : c_cerradoPtr);
Procedure Get_cc(VAR head_cc : c_cerradoPtr);
Procedure Init_flujo(VAR flujo_h, flujo_t : FlujoPtr);

```

IMPLEMENTATION

```

Procedure Presion.init(pin,pout: real);
BEGIN
p_in:=pin;
p_out:=pout;
END;

```

```

Procedure Presion.apunta(next : PresionPtr);
BEGIN
apunt:=next;
END;

```

```

Procedure Presion.numera(i : integer);
BEGIN

```

```
numero:=i;  
END;
```

```
Procedure Presion.liqsat(pin : real);  
BEGIN  
  h_f:=Busca_campana('p','hf',pin);  
  s_f:=Busca_campana('p','sf',pin);  
  v_f:=Busca_campana('p','vf',pin);  
END;
```

```
Función Presion.get_pin: REAL;  
BEGIN  
  get_pin:=p_in;  
END;
```

```
Function Presion.get_pout: REAL;  
BEGIN  
  get_pout:=p_out;  
END;
```

```
Function Presion.get_numero: INTEGER;  
BEGIN  
  get_numero:=numero;  
END;
```

```
Function Presion.get_hf : REAL;  
BEGIN  
  get_hf:=h_f  
END;
```

```
Función Presion.get_sf : REAL;  
BEGIN  
  get_sf:=s_f;  
END;
```

```
Function Presion.get_vf : REAL;  
BEGIN  
  get_vf:=v_f;  
END;
```

```
Procedure Turbina.init(pin,t: real);  
BEGIN  
  p_in:=pin;  
  t_in:=t;  
END;
```

```
Procedure Turbina.salidas(nsal: integer);  
BEGIN  
  n:=nsal;  
END;
```

```

Procedure Turbina.entropia(ps,ts: real);
VAR
  comp : real;

BEGIN
  comp:=busca_campana('p','t',ps);
  IF ts > comp THEN s_tur:=Busca_dato(ps,'t','s',ts);
  IF ts = comp THEN s_tur:=Busca_campana('p','sg',ps);
END;

Function turbina.get_pin : real;
BEGIN
  get_pin:=p_in;
END;

Function turbina.get_tin : real;
BEGIN
  get_tin:=t_in;
END;

Function turbina.get_s : real;
BEGIN
  get_s:=s_tur;
END;

Function Turbina.get_hov(clave: integer; ps,ss: real): real;
VAR
  aux1,aux2,vapor : real;

BEGIN
  aux1:=Busca_campana('p','sg',ps);
  IF ss > aux1 THEN
    BEGIN
      CASE clave OF
        1 : get_hov:=Busca_dato(ps,'s','h',ss);
        2 : get_hov:=Busca_dato(ps,'s','v',ss);
        3 : get_hov:=Busca_dato(ps,'s','t',ss);
      END
    END
  ELSE
    BEGIN
      aux2:=Busca_campana('p','sf',ps);
      vapor:=Calidad(1,ss,aux2,aux1);
      CASE clave OF
        1 : BEGIN
            aux1:=Busca_campana('p','hf',ps);
            aux2:=Busca_campana('p','hg',ps);
            get_hov:=Calidad(2,vapor,aux1,aux2);
          END;
      END;
    END;
  END;

```



```

2 : BEGIN
    aux1:=Busca_campana('p','vf',ps);
    aux2:=Busca_campana('p','vg',ps);
    get_hov:=Calidad(2,vapor,aux1,aux2);
    END;
3 : get_hov:=Busca_campana('p','t',ps);
    END
    END
END;

```

```

Procedure caldera.init(pin,tout: real; resp: char);
BEGIN
    p_in:=pin;
    t_out:=tout;
    IF uppercase(resp) = 's' THEN recal:=true
    ELSE recal:=false;
END;

```

```

Function caldera.get_pin : real;
BEGIN
    get_pin:=p_in;
END;

```

```

Function caldera.get_tout: real;
BEGIN
    get_tout:=t_out;
END;

```

```

Procedure condensador.init(pin,t: real);
BEGIN
    p_in:=pin;
    t_in:=t;
END;

```

```

Function condensador.get_pin: real;
BEGIN
    get_pin:=p_in;
END;

```

```

Function condensador.get_tin: real;
BEGIN
    get_tin:=t_in;
END;

```

```

Procedure condensador.salidas(pin : real);
BEGIN
    h_out:=Busca_campana('p','hf',pin);
    v_out:=Busca_campana('p','vf',pin);
    s_out:=Busca_campana('p','sf',pin);
END;

```

```
Function condensador.get_hvs(clave: integer) : real;
```

```
BEGIN
```

```
  CASE clave OF
```

```
    1 : get_hvs:=h_out;
```

```
    2 : get_hvs:=v_out;
```

```
    3 : get_hvs:=s_out;
```

```
  END;
```

```
END;
```

```
PROCEDURE Init_cc(VAR head_cc,cola_cc : c_cerradoPtr);
```

```
BEGIN
```

```
  head_cc:=NIL;
```

```
  cola_cc:=NIL;
```

```
END;
```

```
PROCEDURE Crea_cc(pi,po : real; VAR head_cc,cola_cc : c_cerradoPtr);
```

```
BEGIN
```

```
  IF head_cc = NIL THEN
```

```
    BEGIN
```

```
      NEW(head_cc);
```

```
      head_cc^.h1:=0;
```

```
      head_cc^.h2:=0;
```

```
      head_cc^.h3:=0;
```

```
      head_cc^.h4:=0;
```

```
      head_cc^.ptur:=pi;
```

```
      head_cc^.pcic:=po;
```

```
      head_cc^.apunt:=NIL;
```

```
      cola_cc:=head_cc;
```

```
    END
```

```
  ELSE
```

```
    BEGIN
```

```
      NEW(cola_cc^.apunt);
```

```
      IF head_cc = cola_cc THEN head_cc^.apunt:=cola_cc^.apunt;
```

```
      cola_cc:=cola_cc^.apunt;
```

```
      cola_cc^.h1:=0;
```

```
      cola_cc^.h2:=0;
```

```
      cola_cc^.h3:=0;
```

```
      cola_cc^.h4:=0;
```

```
      cola_cc^.ptur:=pi;
```

```
      cola_cc^.pcic:=po;
```

```
      cola_cc^.apunt:=NIL;
```

```
    END
```

```
END;
```

```
Procadure Suma_dato(p,x : real; letra : Str2; VAR head_cc : c_cerradoPtr);
```

```
VAR
```

```
  siguacc : c_cerradoPtr;
```

```
BEGIN
```

```

siguecc:=head_cc;
WHILE siguecc <> NIL DO
  BEGIN
    IF p = siguecc^.ptur THEN
      BEGIN
        IF letra='h1' THEN siguecc^.h1:=x;
        IF letra='h2' THEN siguecc^.h2:=x;
        IF letra='h3' THEN siguecc^.h3:=x;
        IF letra='h4' THEN siguecc^.h4:=x;
        siguecc:=NIL;
      END
    ELSE
      siguecc:=siguecc^.apunt;
    END;
  END;

Procedure Get_cc(VAR head_cc : c_cerradoPtr);
VAR
siguecc : c_cerradoPtr;
BEGIN
  ClrScr;
  siguecc:=head_cc;
  IF siguecc = NIL THEN Writeln('Lista vacia')
  ELSE
    BEGIN
      WHILE siguecc <> NIL DO
        BEGIN
          Writeln(siguecc^.ptur);
          Writeln(siguecc^.pcic);
          Writeln(siguecc^.h1);
          Writeln(siguecc^.h2);
          Writeln(siguecc^.h3);
          Writeln(siguecc^.h4);
          siguecc:=siguecc^.apunt;
          Readln;
        END;
      END;
    END;
  END;

PROCEDURE Init_flujo(VAR flujo_h,flujo_t : FlujoPtr);
BEGIN
  flujo_h:=NIL;
  flujo_t:=NIL;
END;
END.

```

ANEXO 5

UNIDAD SAMMY.TPU

UNIT sammy;**INTERFACE**

USES Graph,crt;

TYPE

```
str15 = string[15];
apunt = ^point;
```

```
location = object
  x,y : integer;
  procedure init(initx, inity : integer);
  function getx : integer;
  function gety : integer;
end;
```

```
pointPtr = ^point;
```

```
point = object(location)
  visible : boolean;
  next : apunt;
  constructor init(initx, inity : integer);
  destructor done; virtual;
  procedure show; virtual;
  procedure hide; virtual;
  procedure moveto(newx, newy : integer);
  procedure drag(dragby : integer); virtual;
end;
```

```
circuloPtr = ^circulo;
```

```
circulo = object(point)
  radius : integer;
  nombre : string;
  constructor init(initx,inity,initradius: integer, nom: string);
  procedure show; virtual;
  procedure hide; virtual;
end;
```

```
rectanguloPtr = ^rectangulo;
```

```
rectangulo = object(point)
  xf,yf : integer;
  quees : string;
  apunt : rectanguloPtr;
  constructor init(initx1,inity1,sumx,sumy: integer, nombre: string);
  procedure show; virtual;
  procedure hide; virtual;
```

```

end;

dturbinaPtr = ^dturbina;

dturbina = object(point)
  lado : integer;
  constructor init(initx,inity,inc : integer);
  procedure show; virtual;
  procedure hide; virtual;
end;

dcondensadorPtr = ^dcondensador;

dcondensador = object(point)
  lado : integer;
  constructor init(initx,inity,inc:integer);
  procedure show; virtual;
  procedure hide; virtual;
end;

Procedure dibuja_def(f,calen : boolean);
Procedure dibuja_simul(sp,ncc,nca,nnb,nnr : integer; calen : boolean;
  VAR nl : integer);
Procedure optimo(sobre : integer);

```

IMPLEMENTATION

```

Procedure location.init(initx, inity : integer);
BEGIN
  x:=initx;
  y:=inity;
END;

Function location.getx : integer;
BEGIN
  getx:=x;
END;

Function location.gety : integer;
BEGIN
  gety:=y;
END;

Constructor point.init(initx, inity : integer);
BEGIN
  location.init(initx,inity);
  next:=NIL;
  visible:=false;
END;

Destructor point.done;

```

```
BEGIN
hide;
END;
```

```
Procedure point.show;
BEGIN
visible:=true;
putpixel(x,y,Getcolor);
putpixel(x+2,y,Getcolor);
putpixel(x-2,y,Getcolor);
putpixel(x,y+2,Getcolor);
putpixel(x,y-2,Getcolor);
END;
```

```
Procedure point.hide;
BEGIN
visible:=false;
putpixel(x,y,GetBkcolor);
putpixel(x+2,y,GetBkcolor);
putpixel(x-2,y,GetBkcolor);
putpixel(x,y+2,GetBkcolor);
putpixel(x,y-2,GetBkcolor);
END;
```

```
Procedure point.moveto(newx, newy : integer);
BEGIN
hide;
x:=newx;
y:=newy;
show;
END;
```

```
function getdelta(var deltax : integer;
var deltay : integer): boolean;
```

```
VAR
keychar : char;
quit : boolean;
BEGIN
deltax:=0;
deltay:=0;
getdelta:=true;
REPEAT
keychar:=readkey;
quit:=true;
CASE ord(keychar) of
0 : BEGIN
keychar:=readkey;
CASE ord(keychar) of
72: deltay:=-1;
80: deltay:=1;
```

```

    75: deltax:=-1;
    77: deltax:=1;
    ELSE quit:=false;
    END;
  END;
  13 : getdelta:=false;
  ELSE quit:=false;
  END;
  UNTIL quit;
END;

```

Procedure point.drag(dragby : integer);

```

VAR
  deltax, deltax : integer;
  figurax, figuray: integer;
BEGIN
  show;
  figurax:=getx;
  figuray:=gety;
  WHILE getdelta(deltax,deltay) DO
    BEGIN
      figurax:=figurax+(deltax*dragby);
      figuray:=figuray+(deltay*dragby);
      moveto(figurax,figuray);
    END;
  END;

```

Constructor circulo.init(initx,inity,initradius :integer; nom: string);

```

BEGIN
  point.init(initx,inity);
  radius:=initradius;
  nombre:=nom
END;

```

Procedure circulo.show;

```

BEGIN
  visible:=true;
  putpixel(x,y,Getcolor);
  putpixel(x+2,y,Getcolor);
  putpixel(x-2,y,Getcolor);
  putpixel(x,y+2,Getcolor);
  putpixel(x,y-2,Getcolor);
  graph.circle(x,y,radius);
  graph.outtextxy(x-15,y+10,nombre);
END;

```

Procedure circulo.hide;

```

VAR
  tempcolor : word;
BEGIN

```



```

tempcolor:=graph.getcolor;
graph.setcolor(getbkcolor);
visible:=false;
putpixel(x,y,GetBkcolor);
putpixel(x+2,y,GetBkcolor);
putpixel(x-2,y,GetBkcolor);
putpixel(x,y+2,GetBkcolor);
putpixel(x,y-2,GetBkcolor);
graph.circle(x,y,radius);
graph.outtextxy(x-15,y+10,nombre);
graph.setcolor(tempcolor);
END;

```

Constructor `rectangulo.init`(initx1,inity1,sumx,sumy: integer;
nombre :string);

```

BEGIN
point.init(initx1,inity1);
xf:=sumx;
yf:=sumy;
quees:=nombre
END;

```

Procedure `rectangulo.show`;

```

BEGIN
visible:=true;
putpixel(x,y,Getcolor);
putpixel(x+2,y,Getcolor);
putpixel(x-2,y,Getcolor);
putpixel(x,y+2,Getcolor);
putpixel(x,y-2,Getcolor);
graph.rectangle(x-Round(xf/2),y-Round(yf/2),x+Round(xf/2),y+Round(yf/2));
graph.outtextxy(x+Round(0.2*xf),y+Round(0.7*yf),quees);
END;

```

Procedure `rectangulo.hide`;

```

VAR
tempcolor : word;
BEGIN
tempcolor:=graph.getcolor;
graph.setcolor(getbkcolor);
visible:=false;
putpixel(x,y,GetBkcolor);
putpixel(x+2,y,GetBkcolor);
putpixel(x-2,y,GetBkcolor);
putpixel(x,y+2,GetBkcolor);
putpixel(x,y-2,GetBkcolor);
graph.rectangle(x-Round(xf/2),y-Round(yf/2),x+Round(xf/2),y+Round(yf/2));
graph.outtextxy(x+Round(0.2*xf),y+Round(0.7*yf),quees);
graph.setcolor(tempcolor);
END;

```

Constructor dturbina.init(inix,inity,inc: Integer);

```
BEGIN
  point.init(inix,inity);
  lado:=inc;
END;
```

Procedure dturbina.show;

```
BEGIN
  visible:=true;
  putpixel(x,y,Getcolor);
  putpixel(x+2,y,Getcolor);
  putpixel(x-2,y,Getcolor);
  putpixel(x,y+2,Getcolor);
  putpixel(x,y-2,Getcolor);
  graph.line(Round(x-lado),Round(y-lado/2),Round(x-lado),Round(y+lado/2));
  graph.line(Round(x+lado),Round(y-1.5*lado),Round(x+lado),Round(y+1.5*lado));
  graph.line(Round(x-lado),Round(y-lado/2),Round(x+lado),Round(y-1.5*lado));
  graph.line(Round(x-lado),Round(y+lado/2),Round(x+lado),Round(y+1.5*lado));
END;
```

Procedure dturbina.hide;

```
VAR
  tempcolor : word;
BEGIN
  tempcolor:=graph.getcolor;
  graph.setcolor(getbkcolor);
  visible:=false;
  putpixel(x,y,GetBkcolor);
  putpixel(x+2,y,GetBkcolor);
  putpixel(x-2,y,GetBkcolor);
  putpixel(x,y+2,GetBkcolor);
  putpixel(x,y-2,GetBkcolor);
  graph.line(Round(x-lado),Round(y-lado/2),Round(x-lado),Round(y+lado/2));
  graph.line(Round(x+lado),Round(y-1.5*lado),Round(x+lado),Round(y+1.5*lado));
  graph.line(Round(x-lado),Round(y-lado/2),Round(x+lado),Round(y-1.5*lado));
  graph.line(Round(x-lado),Round(y+lado/2),Round(x+lado),Round(y+1.5*lado));
  graph.setcolor(tempcolor);
END;
```

Constructor dcondensador.init(inix,inity,inc: Integer);

```
BEGIN
  point.init(inix,inity);
  lado:=inc;
END;
```

Procedure dcondensador.show;

```
BEGIN
  visible:=true;
  putpixel(x,y,Getcolor);
```

```

putpixel(x+2,y,Getcolor);
putpixel(x-2,y,Getcolor);
putpixel(x,y+2,Getcolor);
putpixel(x,y-2,Getcolor);
graph.arc(x-2*lado,y,90,270,10);
graph.arc(x+2*lado,y,270,90,10);
graph.line(x-2*lado,y+lado,x+2*lado,y+lado);
graph.line(x-2*lado,y-lado,x+2*lado,y-lado);
END;

```

Procedure dcondensador.hide;

```

VAR
tempcolor: word;
BEGIN
tempcolor:=graph.getcolor;
graph.setcolor(getbkcolor);
visible:=false;
putpixel(x,y,GetBkcolor);
putpixel(x+2,y,GetBkcolor);
putpixel(x-2,y,GetBkcolor);
putpixel(x,y+2,GetBkcolor);
putpixel(x,y-2,GetBkcolor);
graph.arc(x-2*lado,y,90,270,10);
graph.arc(x+2*lado,y,270,90,10);
graph.line(x-2*lado,y+lado,x+2*lado,y+lado);
graph.line(x-2*lado,y-lado,x+2*lado,y-lado);
graph.setcolor(tempcolor);
END;

```

Procedure dibuja_def(f,calen : boolean);

{f para saber que ciclo t=carnot, f=rankine; calen para saber si hay recalentamiento}

```

VAR
xf   : integer;
yf   : integer;
ux,uy : real;
cicl  : string[10];
t_c,h,o,q : char;

BEGIN
xf:=GetMaxX;
yf:=GetMaxY;
ux:=xf/100; uy:=yf/75;
IF f = true THEN cicl:='CARNOT'
ELSE cicl:='RANKINE';
Rectangle(0,0,xf,yf);
OutTextXY(xf-Round(xf/3),Round(yf/20),'CICLO '+cicl);
OutTextXY(Round(xf/8),Round(yf/20),'SISTEMA METRICO');

```

```

Outtextxy(Round(xf/2),Round(yf/1.2), '< RET > para continuar ');
OuttextXY(Round(8*ux),Round(21*uy),'CALDERA');
OuttextXY(Round(8*ux),Round(44*uy),'BOMBA');
OuttextXY(Round(65*ux),Round(21*uy),'TURBINA');
OuttextXY(Round(65*ux),Round(44*uy),'CONDENSADOR');
IF calen=true THEN
  BEGIN
    t_c:=3; h:=4; o:=5; q:=6;
    Outtextxy(Round(40*ux),Round(22*uy),'1 2');
  END
ELSE
  BEGIN t_c:=1; h:=2; o:=3; q:=4; END;
Line(Round(18*ux),Round(21*uy),Round(63*ux),Round(21*uy));
Line(Round(68*ux),Round(23*uy),Round(68*ux),Round(43*uy));
Line(Round(16*ux),Round(44*uy),Round(63*ux),Round(44*uy));
Line(Round(10*ux),Round(23*uy),Round(10*ux),Round(43*uy));
Outtextxy(Round(40*ux),Round(18*uy),q);
Outtextxy(Round(xf/2.5),Round(yf/1.63),h);
Outtextxy(Round(xf/8),Round(yf/2.14),o);
Outtextxy(Round(70*ux),Round(33*uy),t_c);
Readln;
END;
```

```

Procedure dibuja_simul(sp,ncc,nca,nnb,nnt : integer; calen : boolean;
  VAR nl : integer);
```

```

VAR
  xmax,ymax      : integer;
  contador,orig  : integer;
  xo,xf,yo,yf    : integer;
  ntur           : String[7];
  npto,ch1,ch2   : String;
  c              : char;
  ux,uy          : real;
  p1,p2         : Point;
  acirc          : Circulo;
  acald         : Rectangulo;
  headcc,headca : RectanguloPtr;
  tailcc,tailca  : RectanguloPtr;
  auxlta        : RectanguloPtr;
  aturb         : Dturbina;
  acond         : Dcondensador;
```

```

Procedure Init_lispto;
BEGIN
  headcc:=NIL;
  headca:=NIL;
  tailcc:=NIL;
  tailca:=NIL;
END;
```

```
Procedure crea_lispto(VAR cab,cola : rectanguloPtr; numero : Integer;
mens : String);
```

```
VAR
```

```
i : integer;
```

```
BEGIN
```

```
  i:=0;
```

```
  REPEAT
```

```
    IF cab = NIL THEN
```

```
      BEGIN
```

```
        NEW(cab);
```

```
        cab^.init(Round(5*ux),Round(8*uy),Round(7*ux),Round(3*uy),mens);
```

```
        cab^.show;
```

```
        cab^.drag(10);
```

```
        cab^.apunt:=NIL;
```

```
        cola:=cab;
```

```
        i:=i+1;
```

```
      END
```

```
    ELSE
```

```
      BEGIN
```

```
        NEW(cola^.apunt);
```

```
        IF cab = cola THEN
```

```
          cab^.apunt:=cola^.apunt;
```

```
          cola:=cola^.apunt;
```

```
          cola^.init(Round(5*ux),Round(8*uy),Round(7*ux),Round(3*uy),mens);
```

```
          cola^.show;
```

```
          cola^.drag(10);
```

```
          cola^.apunt:=NIL;
```

```
          i:=i+1;
```

```
      END
```

```
    UNTIL i=numero;
```

```
  END;
```

```
Procedure poner_num(VAR k: integer; sps: integer; cabeza: rectanguloPtr;
rc: boolean);
```

```
VAR
```

```
coox,j,seguidor : integer;
```

```
BEGIN
```

```
  auxlta:=cabeza;
```

```
  IF rc = FALSE THEN
```

```
    BEGIN
```

```
      j:=k;
```

```
      WHILE auxlta <> NIL DO
```

```
        BEGIN
```

```
          Str(j,npto);
```

```
          Line(aturb.getx,aturb.gety,auxlta^.getx,auxlta^.gety);
```

```
          OutTextXY(Round((aturb.getx+auxlta^.getx)/2),Round((
```

```

        aturb.gety+auxlta^.gety)/2+0.5*ux),npto);
j:=j-1;
auxlta:=auxlta^.apunt
END;
END
ELSE
BEGIN
IF spe = 0 THEN
BEGIN
j:=k;
WHILE auxlta <> NIL DO
BEGIN
Str(j,npto);
Line(aturb.getx,aturb.gety,auxlta^.getx,auxlta^.gety);
OutTextXY(Round((aturb.getx+auxlta^.getx)/2),Round(
aturb.gety+auxlta^.gety)/2+0.5*ux),npto);
auxlta:=auxlta^.apunt;
j:=j-1;
END
END;
IF spe > 0 THEN
BEGIN
j:=k;
coox:=1+sp;
seguidor:=sp;
WHILE auxlta <> NIL DO
BEGIN
Str(coox,npto);
Line(aturb.getx,aturb.gety,auxlta^.getx,auxlta^.gety);
OutTextXY(Round((aturb.getx+auxlta^.getx)/2),Round(
aturb.gety+auxlta^.gety)/2+0.5*ux),npto);
auxlta:=auxlta^.apunt;
seguidor:=seguidor-1;
IF seguidor = 0 THEN
BEGIN
coox:=j;
j:=j-1;
END
ELSE coox:=coox-1;
END
END
END
END;
BEGIN {principal}
Init_lispto;
xmax:=GetMaxX; ymax:=GetMaxY;
ux:=xmax/100; uy:=ymax/75;
IF calen = TRUE THEN
BEGIN

```

```

IF sp > 0 THEN
  BEGIN
    orig:=2+sp;
    Str(orig,ch1);
    Str(orig+1,ch2);
    ntur:='1 '+ch1+' '+ch2;
    orig:=3;
  END;
IF sp = 0 THEN
  BEGIN
    orig:=3;
    ntur:='1 2 3'
  END
ELSE
  BEGIN
    ntur:='1';
    orig:=1;
  END;
OutTextXY(Round(xmax-40*ux),Round(ymax-72*uy),'CICLO REGENERATIVO');
OutTextXY(Round(xmax-40*ux),Round(ymax-68*uy),'SISTEMA METRICO');
acond.init(Round(5*ux),Round(8*uy),Round(1.5*ux));
acond.show;
acond.drag(10);
aturb.init(Round(6*ux),Round(9*uy),Round(3*uy));
aturb.show;
aturb.drag(10);
IF nca > 0 THEN crea_lispto(headca,tailca,nca,'CA');
IF ncc > 0 THEN crea_lispto(headcc,tailcc,ncc,'CC');
FOR contador:=1 to nnb DO
  BEGIN
    acirc.init(Round(5*ux),Round(8*uy),Round(1.5*ux),'BOMB');
    acirc.show;
    acirc.drag(10);
  END;
IF nnt > 0 THEN
  BEGIN
    FOR contador:=1 to nnt DO
      BEGIN
        acirc.init(Round(5*ux),Round(8*uy),Round(1.5*ux),'TRAM');
        acirc.show;
        acirc.drag(10);
      END
    END;
acald.init(Round(5*ux),Round(8*uy),Round(5*ux),Round(8*uy),'CAL');
acald.show;
acald.drag(10);
Line(aturb.getx,aturb.gety,acald.getx,acald.gety);
OutTextXY(Round((aturb.getx+acald.getx)/2),
  Round((aturb.gety+acald.gety)/2+uy),ntur);

```

```

orig:=orig+ncc;
Poner_num(orig,sp,headcc,calen); orig:=orig+nca; sp:=0;
Poner_num(orig,sp,headca,calen);
Str(orig+1,npto);
Line(aturb.getx,aturb.gety,acond.getx,acond.gety);
OutTextXY(Round((aturb.getx+acond.getx)/2-ux),
           Round((aturb.gety+acond.gety)/2),npto);
Outtextxy(Round(xmax-95*ux),Round(ymax-9*uy),'Defina los puntos de menor a
mayor presi_n ');
Outtextxy(Round(xmax-95*ux),Round(ymax-7*uy),'<F1> para iniciar punto ');
Outtextxy(Round(xmax-95*ux),Round(ymax-5*uy),'<F2> para borrar el +ltimo punto');
Outtextxy(Round(xmax-95*ux),Round(ymax-3*uy),'<F3> para finalizar el dibujo');
contador:=orig+1;
REPEAT
  REPEAT
    c:=Readkey;
  UNTIL c in [#59,#60,#61];
  IF c = #59 THEN
    BEGIN
      contador:=contador+1;
      SetColor(14);
      Str(contador,npto);
      p1.init(Round(5*ux),Round(8*uy));
      p1.show;
      p1.drag(10);
      p2.init(Round(5*ux),Round(8*uy));
      p2.show;
      p2.drag(10);
      Line(p1.getx,p1.gety,p2.getx,p2.gety);
      Outtextxy(Round((p1.getx+p2.getx)/2+0.5*ux),Round((p1.gety+p2.gety)/
        2+0.5*uy),npto);
    END;
  IF c = #60 THEN
    BEGIN
      contador:=contador-1;
      p1.hide;
      p2.hide;
      SetColor(Getbkcolor);
      Line(p1.getx,p1.gety,p2.getx,p2.gety);
      Outtextxy(Round((p1.getx+p2.getx)/2+0.5*ux),Round((p1.gety+p2.gety)/
        2+0.5*uy),npto);
    END;
  UNTIL c = #61;
  n!:=contador;
END; {fin del procedure}

```

Procedure optimo(sobre : integer);

VAR

xmax,ymax,i,j : integer;

ux,uy : real;

ch : String;

BEGIN

l:=sobre+2; Str(i,ch);

j:=i+1;

xmax:=GetMaxX; ymax:=GetMaxY;

ux:=xmax/100; uy:=ymax/75;

OuttextXY(Round(80*ux),Round(10*uy),'TURBINA');

OuttextXY(Round(80*ux),Round(55*uy),'CONDENSADOR');

OuttextXY(Round(10*ux),Round(10*uy),'CALDERA');

OuttextXY(Round(10*ux),Round(35*uy),'C.C1.');

OuttextXY(Round(25*ux),Round(35*uy),'C.C2.');

OuttextXY(Round(40*ux),Round(35*uy),'C.C3.');

OuttextXY(Round(65*ux),Round(35*uy),'C.A.');

OuttextXY(Round(65*ux),Round(55*uy),'BOMBA');

OuttextXY(Round(52*ux),Round(35*uy),'BOMBA');

OuttextXY(Round(10*ux),Round(55*uy),'TRAMPA 1');

OuttextXY(Round(25*ux),Round(55*uy),'TRAMPA 2');

OuttextXY(Round(40*ux),Round(55*uy),'TRAMPA 3');

Line(Round(78*ux),Round(10*uy),Round(20*ux),Round(10*uy));

OuttextXY(Round(47*ux),Round(9*uy),'1');

OuttextXY(Round(50*ux),Round(9*uy),ch); Str(j,ch);

OuttextXY(Round(53*ux),Round(9*uy),ch);

Line(Round(82*ux),Round(12*uy),Round(82*ux),Round(53*uy));

OuttextXY(Round(83*ux),Round(32*uy),'8');

Line(Round(17*ux),Round(36*uy),Round(23*ux),Round(36*uy));

OuttextXY(Round(20*ux),Round(34*uy),'14');

Line(Round(32*ux),Round(38*uy),Round(38*ux),Round(36*uy));

OuttextXY(Round(34*ux),Round(34*uy),'13');

Line(Round(47*ux),Round(38*uy),Round(50*ux),Round(36*uy));

OuttextXY(Round(48*ux),Round(34*uy),'12');

Line(Round(60*ux),Round(36*uy),Round(63*ux),Round(36*uy));

OuttextXY(Round(61*ux),Round(34*uy),'11');

j:=2;

IF j=i THEN j:=j+2; Str(j,ch); j:=j+1;

Line(Round(78*ux),Round(10*uy),Round(13*ux),Round(33*uy));

OuttextXY(Round(46*ux),Round(21*uy),ch);

IF j=i THEN j:=j+2; Str(j,ch); j:=j+1;

Line(Round(78*ux),Round(10*uy),Round(27*ux),Round(33*uy));

OuttextXY(Round(53*ux),Round(21*uy),ch);

IF j=i THEN j:=j+2; Str(j,ch); j:=j+1;

Line(Round(78*ux),Round(10*uy),Round(42*ux),Round(33*uy));

OuttextXY(Round(60*ux),Round(21*uy),ch);

IF j=i THEN j:=j+2; Str(j,ch); j:=j+1;

Line(Round(78*ux),Round(10*uy),Round(67*ux),Round(33*uy));

OuttextXY(Round(73*ux),Round(21*uy),ch);

Line(Round(78*ux),Round(55*uy),Round(72*ux),Round(55*uy));

OuttextXY(Round(75*ux),Round(56*uy),'9');

Line(Round(66*ux),Round(53*uy),Round(66*ux),Round(37*uy));

OuttextXY(Round(68*ux),Round(45*uy),'10');

```
Line(Round(12*ux),Round(53*uy),Round(12*ux),Round(37*uy));  
OuttextXY(Round(13*ux),Round(45*uy),'16');  
Line(Round(27*ux),Round(53*uy),Round(27*ux),Round(37*uy));  
OuttextXY(Round(28*ux),Round(45*uy),'18');  
Line(Round(42*ux),Round(53*uy),Round(42*ux),Round(37*uy));  
OuttextXY(Round(43*ux),Round(39*uy),'20');  
Line(Round(12*ux),Round(12*uy),Round(12*ux),Round(33*uy));  
OuttextXY(Round(13*ux),Round(23*uy),'15');  
Line(Round(12*ux),Round(53*uy),Round(66*ux),Round(37*uy));  
OuttextXY(Round(40*ux),Round(45*uy),'17');  
Line(Round(27*ux),Round(53*uy),Round(66*ux),Round(37*uy));  
OuttextXY(Round(47*ux),Round(45*uy),'19');  
Line(Round(42*ux),Round(53*uy),Round(66*ux),Round(37*uy));  
OuttextXY(Round(54*ux),Round(45*uy),'21');  
Readln;  
END;
```

END. {fin de la unidad}