



**Universidad Michoacana de San
Nicolás de Hidalgo**

**SOFTWARE DE SIMULACIÓN DE
CENTRALES TERMOELÉCTRICAS EN
DIAGRAMAS DE MOLLIER**

TESIS

**Que para obtener el Título de
INGENIERO ELECTRICISTA**

Presenta

ESAÚ GUERRA ZAMORA

Asesor de Tesis

Dr. GILBERTO GONZÁLEZ ÁVALOS

OCTUBRE DE 2009



Agradecimientos

A dios por haberme brindado la paciencia, la fuerza y el valor necesarios para concluir satisfactoriamente esta etapa de mi vida, y con la cual deseo servir a todas aquellas personas que me rodean y que confían en mí.

A mis padres que sin sus consejos y su apoyo hubiera sido muy difícil llegar a ser lo que un día de niño dije ¡Quiero ser ingeniero!, finalmente, gracias a ellos ese día se ha llegado y un sueño mas en mi vida se ha convertido en realidad.

Al Dr. Gilberto González Ávalos primero como maestro y finalmente como asesor quien tuvo la disponibilidad y paciencia de orientarme en el desarrollo de mi tesis.

A todos mis profesores que participaron a lo largo de mi formación académica agradezco su valioso apoyo y sus enseñanzas que fueron fundamentales para cristalizar un sueño.

Sin duda a la Universidad Michoacana la cual fue mi segundo hogar por varios años y sin ella este proyecto no hubiera sido posible.

Dedicatoria

Dedico este trabajo a todas aquellas madrugadas calidas, frías y lluviosas en las cuales tuve que caminar diariamente 3 kilómetros para cumplir un sueño que hoy veo hecho realidad.

A ustedes Papá y Mamá que con sus sacrificios y oraciones he podido culminar una etapa muy importante en nuestras vidas. También te la dedico a ti Papá chucho que aunque ya no hallas podido ver este logro se que te sentirías muy contento de saber que me he recibido como ingeniero.

También dedico este trabajo a mis compañeros de estudio y a todas aquellas personas que confiaron en mí y me apoyaron. A mis ex novias que por la obsesión de llegar a la meta no se pudo consolidar nada.

Resumen

El siguiente tema de tesis trata los beneficios ofrecidos por los sistemas computacionales los cuales en la actualidad son muy utilizados en diferentes áreas de la economía. Para el caso en particular de esta tesis trata los problemas de centrales termoeléctricas los cuales tienen diferentes características y dependiendo de estas el número de variables para determinar su comportamiento es cada vez mayor a medida que aumenta la complejidad del sistema.

Por lo antes descrito se crea un algoritmo (apéndice A) que es capaz de interactuar con un usuario el cual proporcionara datos y este deberá realizar un gran numero de operaciones entre las cuales destacan la búsqueda de datos, comparación de datos, interpolaciones y manejo de ecuaciones para cada variedad de centrales termoeléctricas.

El programa de simulación realizado en esta tesis (apéndice A) se basa en la información del diagrama de Mollier, diagrama que consiste en mostrar el comportamiento del agua en sus diferentes fases a presiones y temperaturas distintas.

Contenido

Agradecimientos	ii
Dedicatoria	iii
Resumen	iV
Lista de figuras	Viii
Lista de Tablas	X
Lista de símbolos y abreviaciones	Xii

Capítulo 1 Introducción

1.1 La generación en México mediante centrales termoeléctricas	1
1.2 Objetivo de la Tesis	3
1.3 Justificación	3
1.4 Metodología	3
1.5 Organización de la tesis	4

Capítulo 2 Antecedentes de las centrales termoeléctricas y diagramas de Mollier

2.1 Centrales termoeléctricas	5
2.1.1 Ciclo convencional	6
2.1.2 Ciclo de recalentamiento	8
2.1.3 Ciclo Regenerativo	9
2.1.4 Consideraciones de diseño	11
2.2 Diagrama de Mollier	11

2.2.1 Breve historia	11
2.2.2 Construcción	14

Capítulo 3 Diseño del software de simulación

3.1 Introducción	16
3.2 La programación del diagrama de Mollier	16
3.3 Búsqueda de información a partir del diagrama	18
3.4 Software de una termoeléctrica	25

Capítulo 4 Pruebas y Resultados

4.1 Introducción	26
4.2 Simulación de una central termoeléctrica convencional caso vapor saturado seco	27
4.2.1 Mediante diagrama de Mollier	29
4.2.2 Usando software de simulación	31
4.3 Prueba de una central termoeléctrica convencional con vapor sobrecalentado	33
4.3.1 Usando software de simulación	37
4.4 Prueba de una central termoeléctrica con ciclo de recalentamiento	38
4.4.1 Usando software de simulación	45
4.5 Prueba de una central termoeléctrica con ciclo de regeneración	46
4.5.1 Usando software de simulación	50
4.6 Prueba de una central termoeléctrica con ciclo de recalentamiento y regeneración	52
4.6.1 Usando software de simulación	55
4.7 Comparación de resultados	57

Capítulo 5 Conclusiones y Recomendaciones

5.1 Conclusiones	58
5.2 Recomendaciones	59
BIBLIOGRAFIA	60
APENDICE A	(CD ANEXO)

Lista de Figuras

Figura 1.1 Ubicación de las centrales termoeléctricas en el país	2
Figura 2.1 Central termoeléctrica convencional	7
Figura 2.2 Central termoeléctrica con una etapa de recalentamiento	8
Figura 2.3 Central termoeléctrica con una etapa regenerativa	10
Figura 2.4 Richard Mollier	12
Figura 2.5 Diagrama de Mollier	13
Figura 2.6 Sección del diagrama de Mollier	14
Figura 3.1 Regiones de operación del diagrama de Mollier	17
Figura 3.2 Secuencia de programación para centrales termoeléctricas	18
Figura 3.3 Secuencia de programación para una central del tipo convencional operando con vapor saturado	19
Figura 3.4 Secuencia de programación para una central del tipo convencional operando con vapor sobrecalentado	20
Figura 3.5 Secuencia de programación para una central de tipo recalentamiento operando con vapor sobrecalentado	21
Figura 3.6 Secuencia de programación para una central de tipo regenerativo operando con vapor sobrecalentado	22
Figura 3.7 Secuencia de programación para una central de tipo mixto operando con vapor sobrecalentado	23
Figura 4.1 Diagrama T – V de una sustancia pura	26
Figura 4.2 Distintas fases de una sustancia pura	27
Figura 4.3 Procesos termoeléctricos de una central convencional	28
Figura 4.4 Datos del diagrama de Mollier	29
Figura 4.5 Datos de entrada al simulador	32
Figura 4.6 Resultados del simulador	32
Figura 4.7 Central termoeléctrica convencional	33
Figura 4.8 Datos del diagrama de Mollier	34
Figura 4.9 Datos de entrada al simulador	37
Figura 4.10 Resultados del simulador	38

Figura 4.11 Central termoeléctrica con recalentamiento	39
Figura 4.12 Datos del diagrama de Mollier	40
Figura 4.13 Datos de entrada al simulador	45
Figura 4.14 Resultados del simulador	46
Figura 4.15 Central termoeléctrica con ciclo de regeneración	47
Figura 4.16 Datos de entrada al simulador	51
Figura 4.17 Resultados del simulador	51
Figura 4.18 Central termoeléctrica mixta	52
Figura 4.19 Datos de entrada al simulador	56
Figura 4.20 Resultados del simulador	56

Lista de Tablas

Tabla 2.1 Comparación de ciclos	10
Tabla 4.1 Interpolación de entalpía y entropía alrededor de una presión de 440.7 lbf/in ²	30
Tabla 4.2 Interpolación de entalpía y entropía alrededor de una entropía de 1.475 btu/lbm°R	30
Tabla 4.3 Interpolación de una entalpía alrededor de una presión de 1.10 lbf/in ²	30
Tabla 4.4 Interpolación de entalpía y entropía alrededor de una presión de 1.10 lbf/in ²	31
Tabla 4.5 Interpolación de entalpías y entropías alrededor de una presión de 734.5 lbf/in ²	34
Tabla 4.6 Interpolación de entalpía y entropía alrededor de una temperatura de 752°F	35
Tabla 4.7 Interpolación de entalpía y entropía alrededor de una entropía de 1.587 btu/lbm°R	35
Tabla 4.8 Interpolación de entalpía alrededor de una presión de 1.10 lbf/in ²	36
Tabla 4.9 Interpolación de entalpías alrededor de una presión de 1.10 lbf/in ²	36
Tabla 4.10 Interpolación de entalpía y entropía alrededor de una presión de 734.5 lbf/in ²	40
Tabla 4.11 Interpolación de entalpía y entropía alrededor de una temperatura de 734.5 lbf/in ²	41
Tabla 4.12 Interpolación de entalpía y entropía alrededor de una presión de 1.58 lbf/in ²	41
Tabla 4.13 Interpolación de entalpía alrededor de una presión de 117.5 lbf/in ²	42
Tabla 4.14 Interpolación de entalpías y entropías alrededor de una presión de 117.5 lbf/in ²	42

Tabla 4.15	Interpolación de entalpía y entropía alrededor de una temperatura de 752 °F	43
Tabla 4.16	Interpolación de entalpías y entropías alrededor de una entropía de 1.81 btu/lbm°R	43
Tabla 4.17	Interpolación de entalpía alrededor de una presión de 1.10 lbf/in ²	44
Tabla 4.18	Interpolación de entalpías alrededor de una presión de 1.10 lbf/in ²	44
Tabla 4.19	Interpolación de entalpías y entropías alrededor de una presión de 734.5 lbf/in ²	48
Tabla 4.20	Interpolación de entalpía y entropía alrededor de una temperatura de 752°F	48
Tabla 4.21	Interpolación de entalpías alrededor de una entropía de 1.58 lbf/in ² btu/lbm°R	49
Tabla 4.22	Interpolación de entalpías y entropías alrededor de una presión de 44.07 y 1.10 lbf/in ²	49
Tabla 4.23	Interpolación para datos de tablas de vapor	49
Tabla 4.24	Interpolación de entalpías alrededor de una entropía de 1.81 btu/lbm°R	54
Tabla 4.25	Interpolación de entalpías alrededor de las presiones de 44.07 y 1.10 lbf/in ²	54
Tabla 4.26	Entalpías del ciclo combinado	55
Tabla 4.27	Comparación de ciclos	57

Lista de Símbolos y Abreviaturas

Gw	Gigawatts
CO ₂	Dióxido de carbono
°F	Grados Fahrenheit
°C	Grados Centigrados
°R	Grados Rankine
X	Calidad
lbf	Libras fuerza
lbm	Libras masa
in ²	Pulgadas cuadradas
btu	Unidad de energía en el sistema ingles
bar	Unidad de presión en el sistema internacional

CAPITULO 1

INTRODUCCION

1.1 La generación en México mediante las centrales termoeléctricas

Una central termoeléctrica es una instalación empleada para la generación de energía eléctrica a partir de la energía liberada en forma de calor, normalmente mediante la combustión de combustibles fósiles como petróleo, gas natural o carbón. Este calor es empleado por un ciclo termodinámico convencional para mover una turbina acoplado a ella un alternador y producir energía eléctrica. Este tipo de generación eléctrica es contaminante ya que genera dióxido de carbono.

Por otro lado también existen centrales termoeléctricas que emplean fisión nuclear del uranio para producir electricidad. Este tipo de instalación recibe el nombre de central nuclear. Con lo antes dicho podemos clasificar a la energía de dos maneras:

- a).- Energía primaria. La energía primaria es aquella que se obtiene directamente de la naturaleza, como lo es el petróleo, el carbón, combustóleo, gas natural, etc.
- b).- Energía secundaria. Este tipo de energía es aquella que ha sido ya industrializada como lo es la gasolina, electricidad, etc.

Actualmente en México existen 23 plantas termoeléctricas, de las cuales el 68.8 % de la electricidad que se genera en México es debido a estos procesos termoeléctricos, el 28.8% es de origen hidroeléctrico, el 2.4% restante a otros tipos de generación, como la geotérmica y la eólica.

En la figura 1.1 se muestra la ubicación de cada una de las centrales termoeléctricas dentro de territorio mexicano.

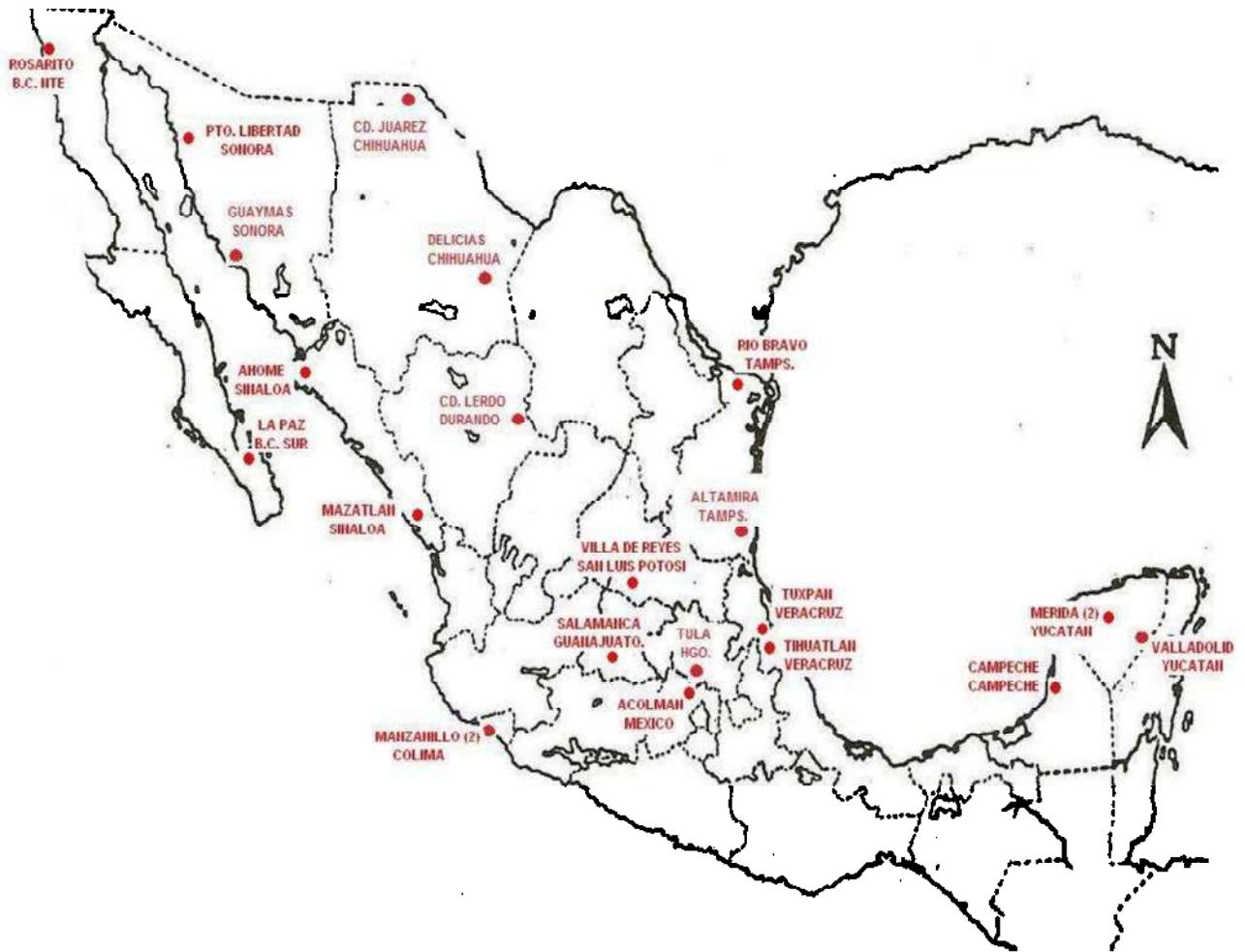


Figura 1.1 Ubicación de las centrales termoeléctricas en el país

Como se puede apreciar en el mapa anterior la mayoría de las centrales termoeléctricas se encuentran ubicadas sobre las costas debido a que es más fácil de tener acceso al agua para la operación de la misma.

Algunas de las ventajas con las que cuentan las centrales termoeléctricas, son su costo, debido a la simplicidad con que estas cuentan y sobre todo las de carbón comparándolas con otras de diferente combustible.

Las centrales de ciclo combinado de gas natural son mucho más eficientes (alcanzan hasta un 50%) que centrales convencionales, aumentando la energía eléctrica generada y por tanto las ganancias con la misma cantidad de combustible, y rebajando las emisiones contaminantes mas arriba de un 20% que las convencionales.

Las grandes desventajas de este tipo de centrales son el uso de combustibles fósiles ya que estos provocan una gran cantidad de gases de efecto invernadero (emisión de

dióxido de carbono CO₂), esto provoca que el tipo de energía sea finita debido a la duración de las reservas.

1.2 Objetivo

El objetivo de esta tesis consiste en desarrollar un software de simulación de centrales termoeléctricas basado en las curvas del comportamiento del agua en diagrama de Mollier que permita obtener parámetros con mayor facilidad y precisión, además de mostrar gráficamente los parámetros.

1.3 Justificación

Hoy en día es necesario realizar cálculos de cualquier tipo con la mayor rapidez y precisión por tanto enfrentamos la necesidad de buscar nuevas formas y técnicas para la obtención de dichos parámetros.

Actualmente los equipos de cómputo son fundamentales para la realizar y ejecutar cualquier tipo de cálculo que se indique por complejo que este resulte. Por tanto para el desarrollo y ejecución de este software será necesaria la utilización de un equipo de cómputo siendo la manera más sencilla, rápida y precisa de obtener los parámetros de una central termoeléctrica.

Debido a las cuantiosas inversiones, el producto nacional del sector de la electricidad, el gas y el agua creció un 5.2% en el primer semestre de 1997. El aprovechamiento de las distintas fuentes existentes permitió una generación de energía eléctrica de 145,628 Gw en 1996, dato que supuso un crecimiento del 4% con relación al año anterior, la capacidad instalada de energía eléctrica del sector estatal aumentó, por su parte, en un 5.3%. De la capacidad total instalada, el 28.8% es de origen hidroeléctrico, el 57.7% de termoeléctrico, el 7.4% de Carbo eléctrico y el 3.7% de núcleo eléctrico, correspondiendo el 2.4% restante a otros tipos de generación, como la geotérmica y la eólica.

1.4 Metodología

Para la realización de esta tesis fue necesario conocer las distintas formas de cálculo de los parámetros de una central termoeléctrica, investigar y documentarse acerca de la

información existente de las propiedades del agua en sus distintos estados físicos.

1.5 Organización de la tesis

En el capítulo 1, se da una breve introducción de cómo es la generación en México mediante las centrales termoeléctricas, además de mencionar la ubicación de cada central termoeléctrica y especificar la cantidad de energía que se genera y de las unidades e inicio de operación de cada central.

En el capítulo 2, se presenta una breve historia de los antecedentes de las centrales termoeléctricas y del origen del diagrama de Mollier. También se aborda el tema de cómo es el funcionamiento de una central termoeléctrica convencional con una etapa de recalentamiento. Así mismo se menciona cual es el funcionamiento del diagrama de Mollier y cuales son sus ventajas y desventajas de este con respecto a las otras formas de obtención de parámetros de las centrales termoeléctricas.

En el capítulo 3, se presenta un software de simulación para la obtención de parámetros de una central termoeléctrica utilizando el diagrama de Mollier.

En el capítulo 4, se muestran algunas pruebas y resultados de algunos problemas propuestos utilizando el software de simulación.

En el capítulo 5, se presentan las conclusiones y recomendaciones generales para trabajos futuros. Finalmente se incluyen los diferentes tipos de bibliografía utilizados para el desarrollo de esta tesis.

CAPITULO 2

Antecedentes de las centrales termoeléctricas y diagramas de Mollier

2.1 Centrales termoeléctricas

La primera central termoeléctrica nace en Nueva York en (1882) construida con la primera estación generadora, inventada por Edison. El principio de funcionamiento de una central térmica se basa en el intercambio de energía calórica en energía mecánica y luego en energía eléctrica.

Las primeras centrales que se construyeron eran máquinas de vapor a pistón, similares en su funcionamiento a una locomotora y que movían al generador (una de estas se conserva todavía, en la escuela Otto Krause y se pone en funcionamiento una vez al año). Luego se reemplazó por una turbina de vapor, con la que se calienta agua en una caldera que produce vapor a presión, el cual se aplica sobre los álabes de la turbina que convierte energía potencial (presión) en energía cinética que acciona el generador. A este tipo de centrales termoeléctricas se les denominan clásicas debido a que emplean la combustión del carbón, petróleo (fuelóleo) o gas natural para generar energía eléctrica.

Son consideradas las centrales más económicas y más rentables, por lo que su utilización está muy extendida en el mundo económicamente avanzado y en el mundo en vías de desarrollo, a pesar de que estén siendo criticadas debido a su elevado impacto ambiental.

El funcionamiento de todas las centrales térmicas, o termoeléctricas, es semejante. El combustible se almacena en parques adyacentes, desde donde se suministra a la central, pasando a la caldera, en la que se provoca la combustión. Esta se emplea para calentar el agua que se tiene en la caldera, y producir vapor. Este con una alta presión, hace girar los alabes de las turbinas, cuyo eje del rotor gira solidariamente con el de un generador que produce energía eléctrica; esta energía se transporta mediante líneas de alta tensión a los centros de consumo.

Por su parte, el vapor es enfriado en un condensador y convertido otra vez en agua, que vuelve a los tubos de la caldera, comenzando de nuevo el ciclo. A este ciclo también se le conoce como ciclo Rankine y es uno de los ciclos fundamentales de las maquinas de vapor. Puesto que es el ciclo de mayor eficiencia. En este ciclo los procesos de calentamiento y enfriamiento se efectúan a presión constante.

El agua en circulación que refrigera el condensador expulsa el calor extraído a la atmósfera a través de las torres de refrigeración, grandes estructuras que identifican este tipo de centrales; parte del calor extraído pasa a un río próximo, lago o al mar.

Las torres de refrigeración son enormes cilindros contraídos a media altura (hiperboloides), que emiten constantemente, vapor de agua (que se forma durante el ciclo) no contaminante, a la atmósfera. Para minimizar los efectos contaminantes de la combustión sobre el entorno, la central dispone de una chimenea de gran altura (llegan a los 300 metros) y de unos precipitaderos que retienen las cenizas y otros volátiles de la combustión. Las cenizas se recuperan para su aprovechamiento en procesos de metalurgia y en el campo de la construcción, donde se mezclan con el cemento.

2.1.1. Ciclo convencional

Una central termoeléctrica convencional tiene como objetivo producir energía eléctrica a base de una serie de conversiones de energía las cuales se realizan dentro de la central, a partir de una energía disponible, para el caso de las centrales termoeléctricas convencionales utilizan como combustible el carbón, gas, combustóleo, etc. Que es la energía química disponible a convertir en energía eléctrica. En la figura 2.1 se muestra un esquema de los componentes en donde se efectúan las conversiones de energía en este tipo de centrales.

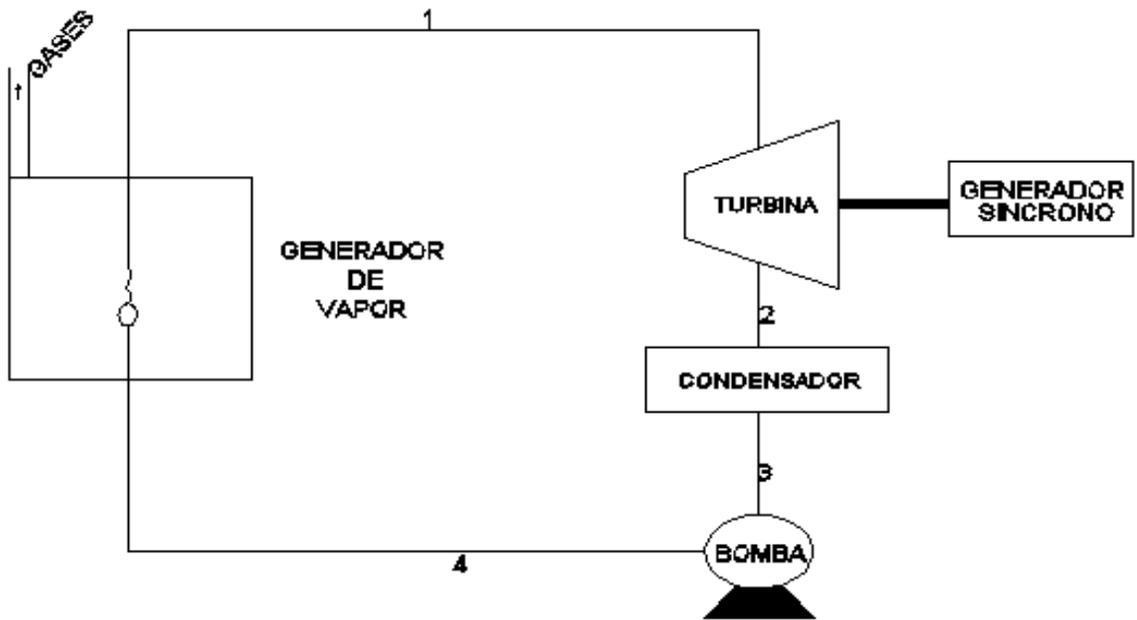


Figura 2.1 Central termoeléctrica convencional

De la figura 2.1 se observan los principales componentes de una central termoeléctrica en donde se realizan las distintas transformaciones energéticas. Estas transformaciones inician a partir de la energía que libera el combustible debido a sus reacciones químicas que en este caso es la combustión.

Al producirse la combustión ya se tiene la primera transformación de energía, es decir, que la energía química contenida por el combustible se transforma en calor (energía calorífica) en la flama y en los gases calientes de la combustión.

La combustión se realiza en el hogar de un generador de vapor con la finalidad producir vapor en base al calentamiento del agua por lo que ya se tiene otra transformación de la energía.

La energía del vapor será transformada en la siguiente etapa en trabajo mecánico por medio de una turbina de vapor y por tanto se obtiene una transformación más de energía. Finalmente, si a la turbina se acopla un generador síncrono éste convierte la energía mecánica entregada por la turbina de vapor a energía eléctrica que es la última transformación de energía y por la cual se llega al objetivo deseado.

Todas las transformaciones de energía mencionadas anteriormente son efectuadas dentro de una central termoeléctrica que cuenta con el equipo necesario para efectuarlas.

El equipo de una central termoeléctrica es importante pero de acuerdo a su participación en el logro del objetivo, del tamaño y de su costo estos se clasifican de la siguiente manera:

- a) Generador de vapor
- b) Turbina y condensador
- c) Generador eléctrico

Al resto del equipo que conforman la central se le clasifica como auxiliar entre los cuales están: Bombas, Ventiladores, Calentadores, Tanque, Compresores, Enfriadores, etc.

Existen algunas variantes de las centrales termoeléctricas como:

- 1.- Ciclo de recalentamiento
- 2.- Ciclo regenerativo
- 3.- Ciclo combinado o mixto

2.1.2. Ciclo de recalentamiento

En esta variación lo que conseguimos es un aumento en el rendimiento total. Esto es debido a que el vapor que ya trabajó en la turbina se vuelve a calentar a presión constante y por tanto se elimina la humedad dentro del generador de vapor. En la figura 2.2 se muestra el diagrama de una central termoeléctrica con una etapa de recalentamiento.

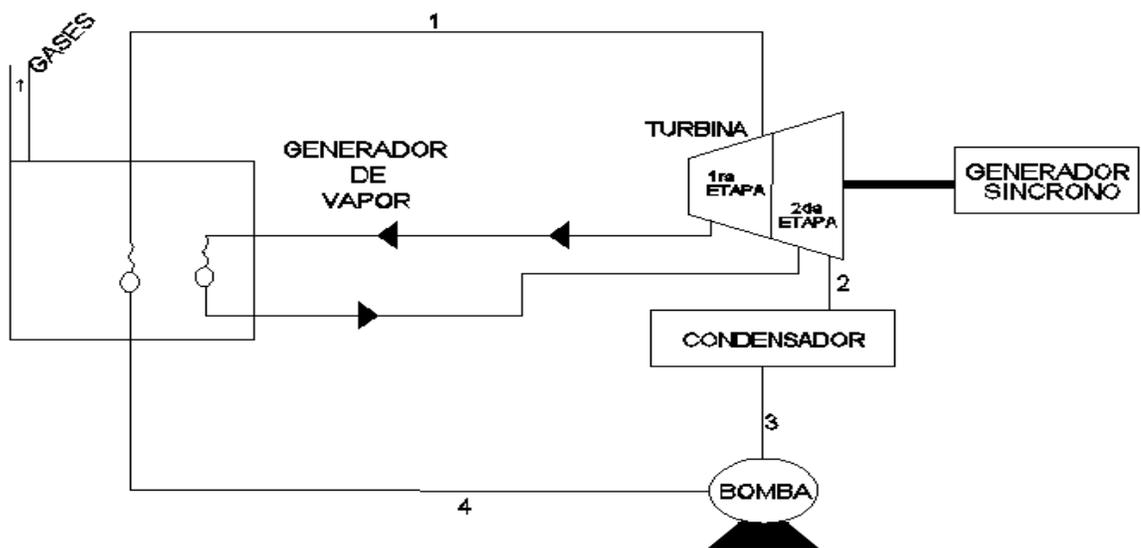


Figura 2.2 Central termoeléctrica con una etapa de recalentamiento

De la figura 2.2 se muestra que después del recalentamiento se vuelve a expandir en la segunda etapa de la turbina hasta la presión del condensador. A la primera etapa de la turbina se le denomina turbina de alta presión y a la segunda etapa de la turbina de baja presión. En esta variación de ciclo Rankine, encontramos ventajas respecto al ciclo convencional como un aumento de vapor en la turbina y aumentar la potencia de esta. Pero por otro lado también encontraremos inconvenientes como una instalación más compleja y un aumento en su costo.

2.1.3. Ciclo regenerativo

En esta variación de centrales se introduce un nuevo elemento al ciclo, un calentador abierto. Este elemento consiste en un intercambiador de calor por contacto directo en el cual se mezclan dos corrientes de agua para dar una corriente de temperatura intermedia. De las dos corrientes que entran al calentador una proviene de una extracción de vapor de la turbina y la otra del condensador. En la figura 2.3 se muestra el esquemático de una central termoeléctrica con una etapa regenerativa.

En esta etapa de la central termoeléctrica (figura 2.3) el calor se transfiere directamente de los gases calientes al agua, a través de un elemento intermedio almacenador de calor, que consiste en un tambor conteniendo paquetes de laminillas llamadas canastas. El objetivo de este ciclo es el extraer parte del vapor que ha sido usado en la turbina pero a una presión superior a la que opera el condensador y usarlo para calentar el agua que alimenta a la caldera, de esta manera se recupera parte de la energía que libera el vapor extraído al condensarse.

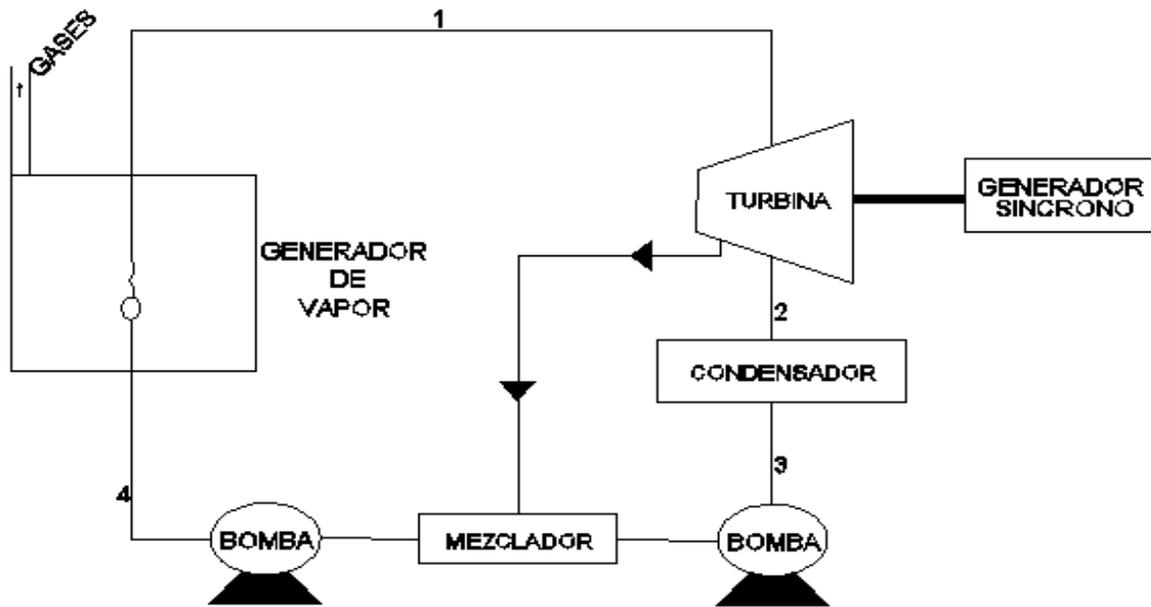


Figura 2.3 Central termoeléctrica con una etapa de regenerativa

En esta variación del ciclo Rankine, encontramos ventajas respecto al ciclo convencional como un aumento del rendimiento y una reducción de aporte de calor a la caldera. Pero por otro lado encontramos también inconvenientes como una reducción de la potencia de la turbina y un aumento de la complejidad de la instalación, ya que añadiremos a la instalación una bomba más y un mezclador de flujos.

En la tabla 2.1 se muestra el desempeño de cada tipo de central con las mismas características de operación.

Tabla 2.1 Comparación de ciclos

CICLO	CONDICION DE ENTRADA	CONDICION DE SALIDA	EFICIENCIA
CON CONDENSADOR	50 bar 400 °C	0.075 bar	37%
RECALENTAMIENTO 1 PASO	50 bar 400 °C	0.075 bar	38%
REGENERATIVO 1 PASO	50 bar 400 °C	0.075 bar	39.6%

2.1.4. Consideraciones de diseño

Los principales motivos por los cuales se puede incrementar la eficiencia de una central termoeléctrica son el aumento de la presión en la turbina y el aumento de la temperatura inicial del vapor. Pero sin embargo estas modificaciones tienen repercusiones negativas en los equipos.

Para el caso de aumentar la presión las repercusiones son las siguientes:

- Aumento de potencia de la bomba
- Modificación del costo de la caldera
- Incremento del costo de la turbina
- Modificación de confiabilidad del sistema

Para el caso de aumentar la temperatura inicial del vapor se tienen las siguientes repercusiones:

- Reducción del esfuerzo admisible de los materiales
- Corrosión tanto en las tuberías como en las paredes del generador de vapor así como en la turbina.

De lo anterior se concluye que si se desea mejorar la eficiencia de una central depende en un gran porcentaje del desarrollo tecnológico de los materiales.

2.2 Diagrama de Mollier

2.2.1. Breve historia

El diagrama de Mollier es una representación gráfica de las propiedades termodinámicas de los materiales y los estados con "entalpía" en una de las coordenadas.

El diagrama lleva este nombre en honor a Richard Mollier (1863-1935) fue profesor de la universidad de Dresde, Alemania, y fue pionero en la representación gráfica de la relación de la temperatura, presión, entalpía, entropía y el volumen de vapor y aire húmedo que desde entonces ha contribuido a la enseñanza de la termodinámica a muchas generaciones de ingenieros. Su diagrama de entalpía –entropía de vapor se publicó por primera vez en 1904. Continuo su labor de actualizar sus diagramas termodinámicos y mesas de vapor para tener en cuenta los nuevos avances técnicos.

EL profesor Mollier se activa en el ámbito de la gasificación y combustión de los procesos en relación con las máquinas de vapor, motores de combustión y un equipo de refrigeración.

En 1923 en una conferencia de termodinámica, celebrada en los Ángeles, El profesor Mollier fue otorgado con el gran honor de nombrar a todos los diagramas termodinámicos que tengan la entalpía en una de sus coordenadas como “diagrama de Mollier”.



Figura 2.4 Richard Mollier (1863-1935)

Los diagramas de Mollier, figura 2.5, se utilizan habitualmente en el diseño de los trabajos relacionados con las plantas de energía fósil o nuclear, como; compresores, turbinas de vapor, sistemas de refrigeración, aire acondicionado, y equipo de trabajo para visualizar los ciclos termodinámicos de los sistemas.

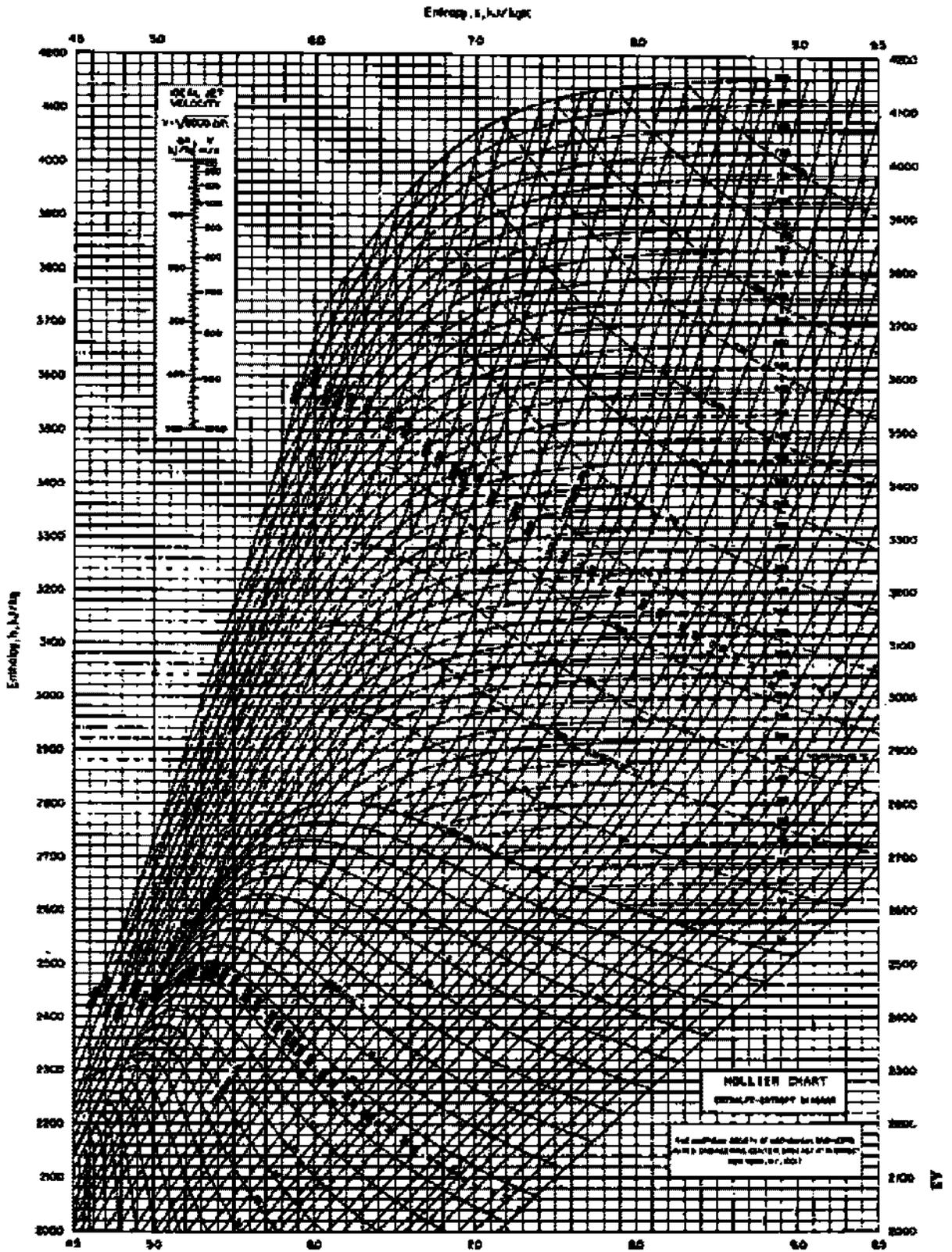


Figura 2.5 Diagrama de Mollier

2.2.2. Construcción

El diagrama de Mollier es una representación de las propiedades del agua y vapor de agua. Se usa un sistema principal de coordenadas H-S (entalpía - entropía).

En la figura 2.6 la línea de saturación (borde de la campana de cambio de fase) es una línea de importancia. Separa la zona de líquido saturado de la zona de vapor sobrecalentado. Dentro de la campana de cambio de fase las isóbaras se confunden con las isotermas. Es decir, si la condensación es a presión constante, también será a temperatura constante. Una propiedad importante de estas líneas de condensación es que son rectas.

El punto de origen del diagrama de Mollier (coordenadas 0) es a 1 atm. De presión y 0°C de temperatura. Allí se fija a la entropía y la entalpía con valor 0 (cero). Los diagramas de Mollier usuales sólo representan una porción del espacio completo H - S. Esta representación se limita a las temperaturas y presiones más usuales y en general se excluye la zona de líquido saturado o subsaturado.

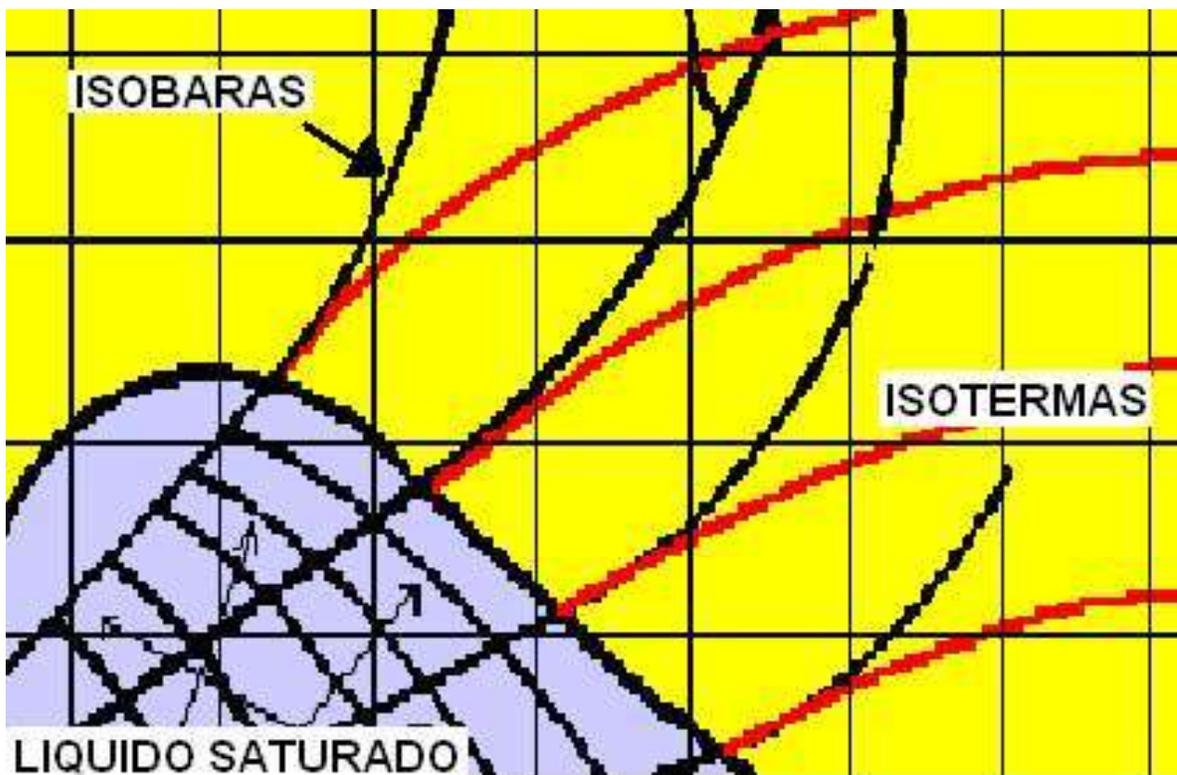


Figura 2.6 Sección de un diagrama de Mollier

De la figura 2.6 observamos que dentro de la campana de cambio de fase se debe

tener claro que las isotermas y las isobáras se confunden en una línea única que llamamos rectas de condensación. En el diagrama se lee directamente la presión. Para leer la temperatura es necesario subir por la recta de condensación y leer la temperatura en $X = 1$ (línea de vapor saturado).

Las otras líneas de importancia en esta zona son las líneas de igual título. Definiremos a la calidad X del vapor como:

$$x = \frac{\text{Masa Vapor saturado}}{\text{liq. + vap. satur.}}$$

No se debe olvidar que un líquido está saturado cuando está en equilibrio con su fase vapor. Asimismo el vapor está saturado cuando está en equilibrio con la fase líquida. Por lo tanto, el concepto de calidad representa la fracción de vapor saturado que existe en una masa unitaria de líquido y vapor saturado. El concepto de calidad no tiene sentido fuera de la campana de cambio de fase, en la zona de vapor sobrecalentado se separan las isóbaras de las isotermas, además de la información obvia que se extrae directamente del diagrama (entalpía y entropía en un punto), se puede además obtener información adicional.

CAPITULO 3

Diseño del software de simulación

3.1 Introducción

Con la aparición de las computadoras desaparecen las secuencias de posiciones de llaves mecánicas que debían desconectarse para obtener una acción determinada. Actualmente para obtener cualquier tipo de acción solamente es necesario desarrollar un programa basado en una serie de instrucciones y estas enviárselas a un ordenador (computador) en un lenguaje entendido por él, para decirle exactamente lo que queremos que haga. Si el ordenador no entiende alguna instrucción generalmente este lo comunicará mediante mensajes visualizados en la pantalla.

Para el desarrollo de este software se utilizará el lenguaje de programación java, el cual es un lenguaje que permite crear programas para uso personal, para una empresa, para aplicaciones distribuidas a través de Internet aplicaciones de bases de datos y otros muchos más.

Actualmente en el mercado existe una gran diversidad de herramientas de programación como C++, forte de sun, matlab, y otros muchos más, pero la forma de ver el alcance de java sobre los demás es desarrollando directamente a través del kit de desarrollo de java. Se trata de un paquete basado en un conjunto de herramientas de órdenes en línea para editar, compilar, ejecutar y depurar programas en java.

3.2 La programación del diagrama de Mollier

Para realizar la programación del diagrama de Mollier es necesario obtener los datos de las distintas regiones en las cuales opera. En la figura 3.1 se muestra cada una de las regiones que componen el diagrama de Mollier.

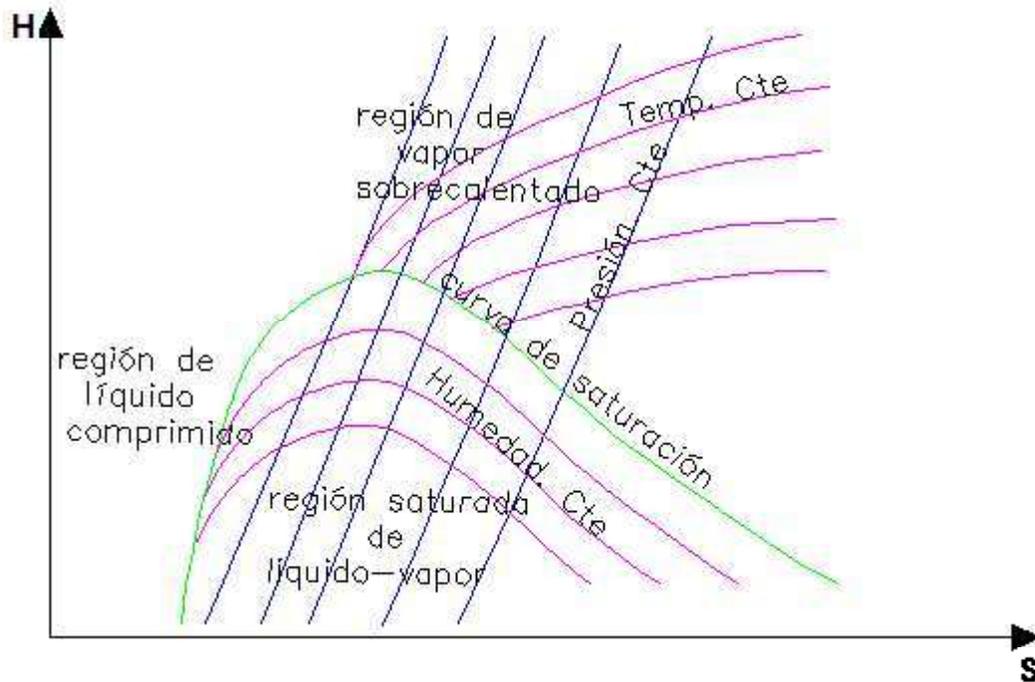


Figura 3.1 Regiones de operación del diagrama de Mollier

Para programar la región de vapor sobrecalentado es necesario obtener los datos de esta región los cuales son almacenados y ordenados en dos matrices en donde una de ellas llamada entalpía. Esta almacena todas las entalpías de cada una de las presiones a distintas temperaturas. Con las mismas características son almacenados en otra matriz denominada entropía la cual contiene todos los datos de las entropías para cada una de las presiones y sus temperaturas correspondientes. Estos valores de entalpía y entropía se determinan mediante los cruces de cada una de las presiones a diferentes temperaturas.

Al igual que en la región de vapor sobrecalentado en la región saturada de líquido - vapor los datos se almacenan en dos matrices con la misma denominación que en la región de sobrecalentado, en las cuales la entalpía y entropía son determinadas por los cruces de las presiones y los porcentajes de humedad que corresponden para cada presión. Para los datos de la región de saturación que es el punto donde la sustancia química (agua para este caso) se encuentra en su punto de equilibrio es decir, la cantidad de vapor y agua son iguales, por tanto para la obtención de estos datos es necesario obtenerlos de las tablas termodinámicas de vapor saturado y seco las cuales definen el valor exacto de las entalpías tanto para el porcentaje de agua como para el

de vapor, estos valores son ordenados y almacenados en dos vectores los cuales tienen la información de las entalpías para cada una de las presiones y sus temperaturas de ebullición correspondientes.

Para la región de líquido comprimido la forma en la que se adquieren los datos es tomando los valores de la entalpía como de la entropía en los puntos donde se efectúan los cruces de las presiones con distintos valores de temperatura, al igual que en la región de vapor saturado.

3.3 Búsqueda de información a partir del diagrama

La búsqueda de información en el software de simulación de una central termoeléctrica se realiza a partir de los datos de entrada en el simulador (datos iniciales de operación de la central). En la figura 3.2 se muestra la secuencia de programación para cada tipo de central que se menciono en capítulos anteriores.

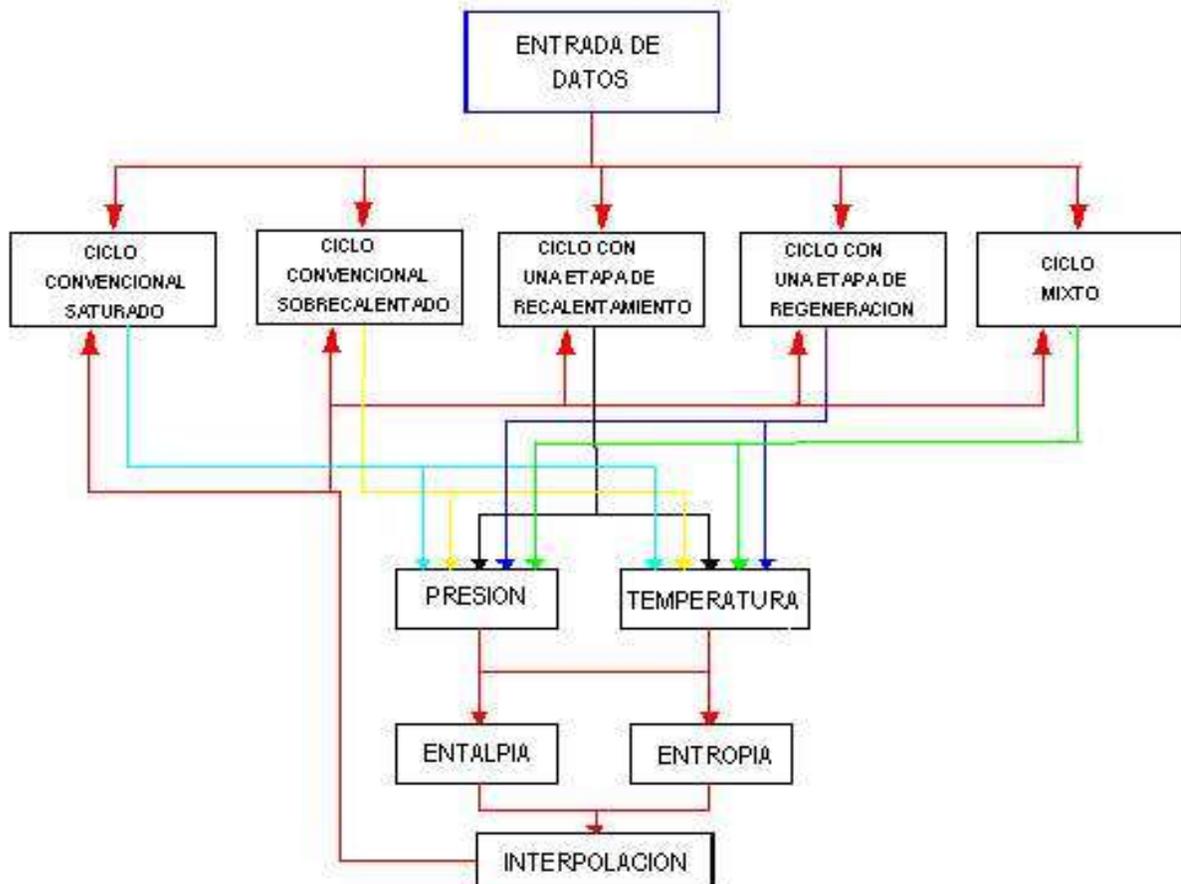


Figura 3.2 Secuencia de programación para centrales termoeléctricas

En la figura 3.2 las líneas de color rojo representan los datos que son comunes para las distintas modificaciones del ciclo Rankine las líneas de color violeta, verde, amarillo y azul son colores que corresponden a cada tipo de central es decir, cuando los datos de entrada indican que la central es de tipo regenerativo y opera en la región de vapor sobrecalentado entonces, el bloque que entra en funcionamiento es el “ciclo con una etapa de regeneración” y es este el que envía datos de presión y temperatura a los bloques de mismo nombre y estos definen con precisión el tipo de dato. Posteriormente son enviados a los bloques de entalpía y entropía y finalmente al bloque de interpolación el cual regresa un dato bien definido al bloque de regeneración y este almacena el dato y envía los demás datos de la entrada a los bloques correspondientes para así desplegar un valor final.

En la figura 3.3 se explica gráficamente como se llega al valor final cuando al bloque de entrada de datos (figura 3.2) se aplican los datos que especifican que la planta es del tipo convencional y que opera con vapor saturado y seco.

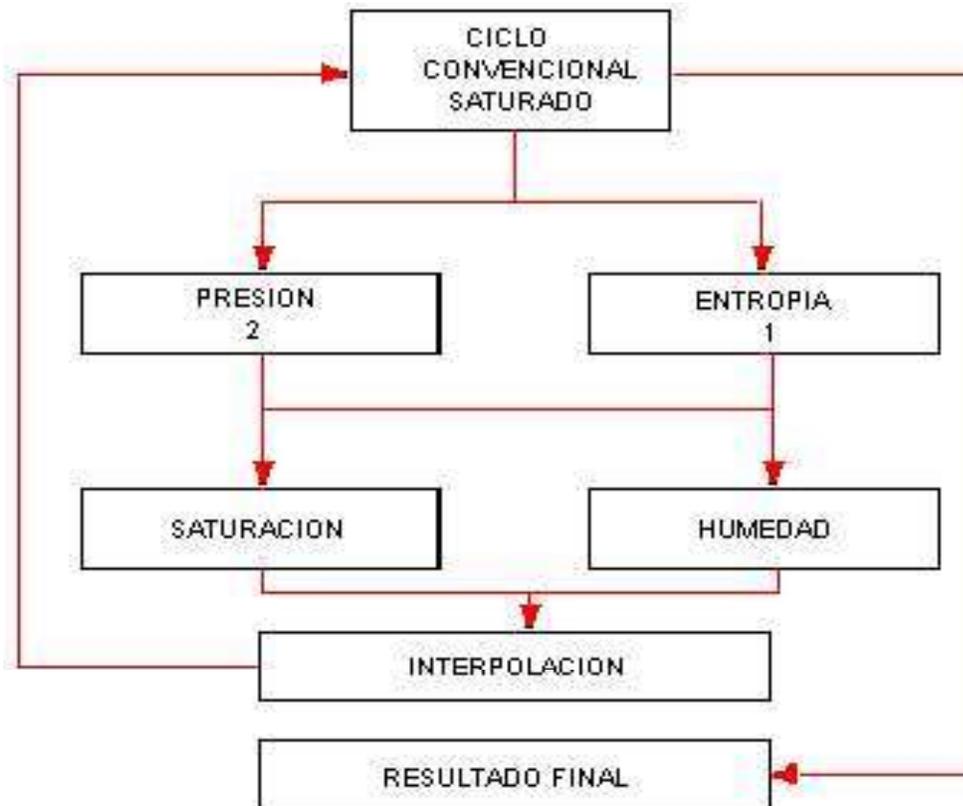


Figura 3.3 Secuencia de programación para una central de tipo convencional Operando con vapor saturado y seco

En la figura anterior se muestra como se determinan los parámetros a la salida de la

turbina en donde se aplica la presión 2 (presión a la salida de la turbina) y la entropía 1 que fue determinada en el primer proceso y almacenada por el bloque convencional saturado (figura 3.2), con estos dos parámetros definidos se obtienen los datos de la curva de saturación y la cantidad de humedad que sale de la turbina finalmente, estos resultados son interpolados y regresados al bloque convencional saturado para que este procese todos los datos almacenados y posteriormente sea mostrado un resultado final.

Para la simulación de una central termoeléctrica convencional operando con vapor sobrecalentado el procedimiento para determinar los parámetros de entrada es descrito en la figura 3.2 usando el bloque de convencional saturado y para los parámetros a la salida de la turbina se describen en la figura 3.3 ya que el procedimiento es el mismo que en un ciclo convencional saturado ver figura 3.4.

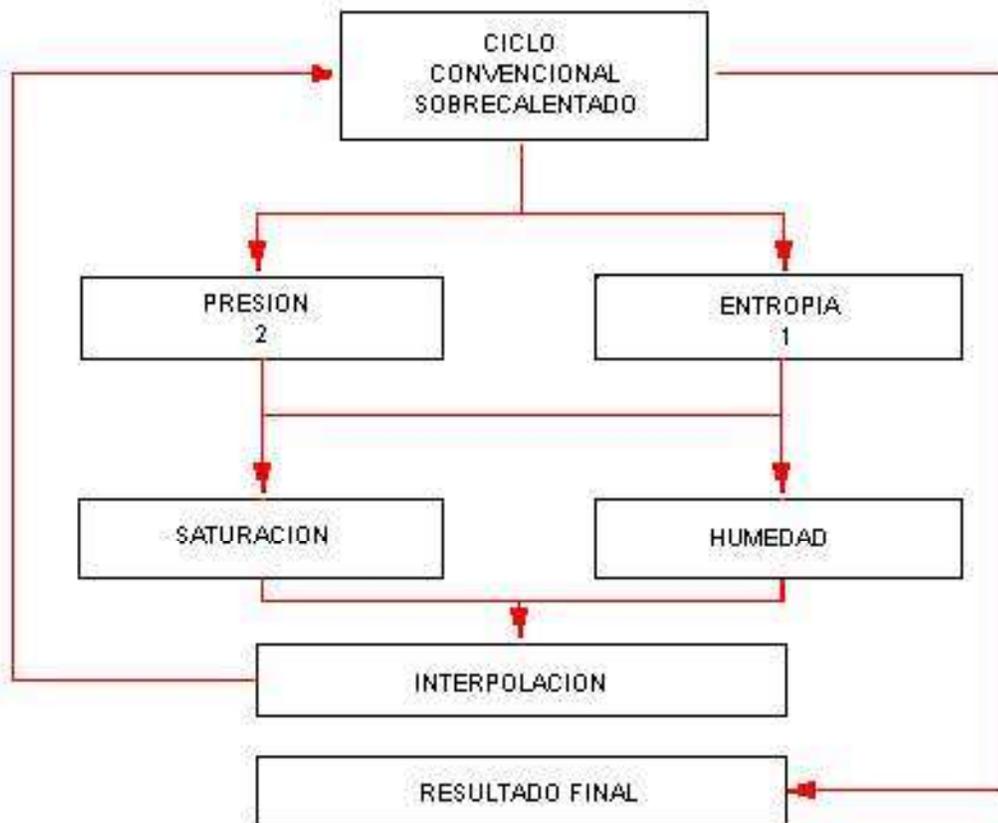


Figura 3.4 Secuencia de programación para una central de tipo convencional operando con vapor sobrecalentado

Para un ciclo con recalentamiento los parámetros iniciales se calculan como se describe

en la figura 3.2 solamente que se utiliza el bloque de recalentamiento y su funcionamiento se explica en la figura 3.5

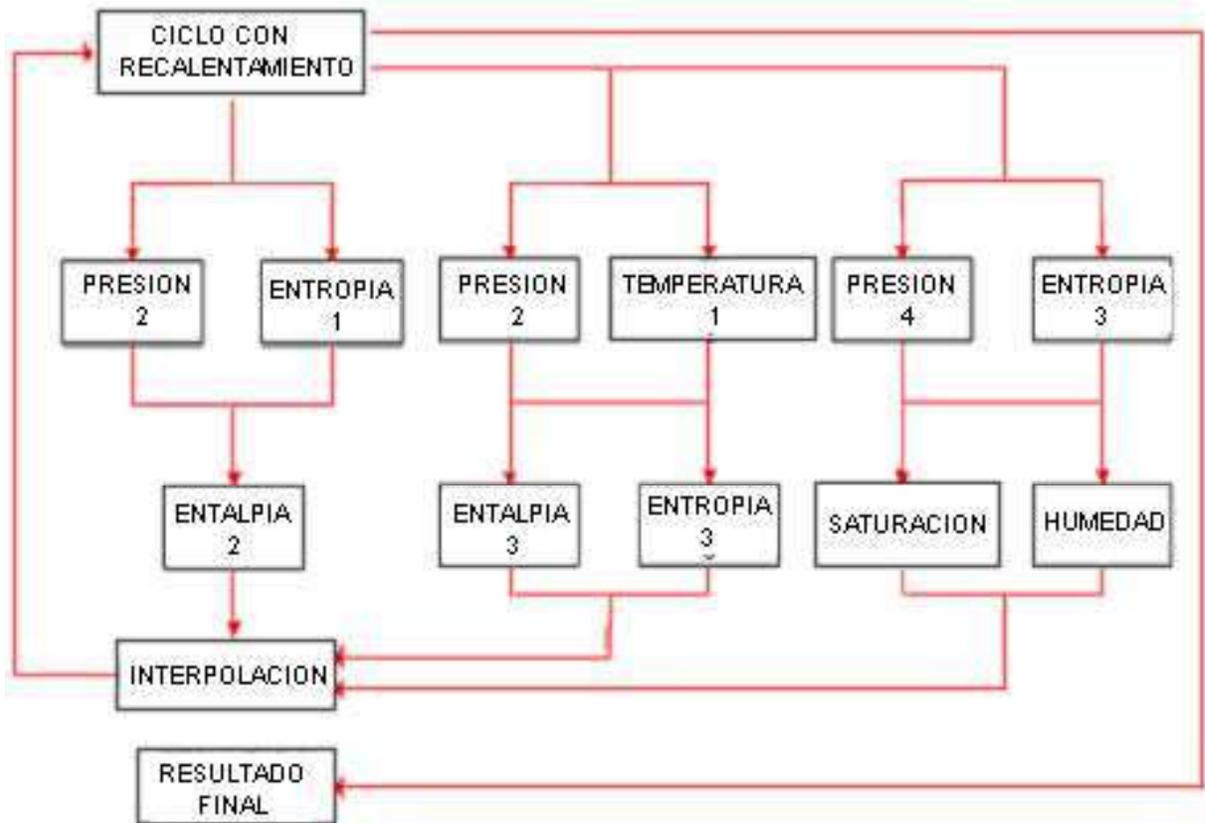


Figura 3.5 Secuencia de programación para una central de tipo recalentamiento Operando con vapor sobrecalentado

En la figura 3.5 una vez que se a almacenado los primeros datos (datos de la entrada de la turbina) en el bloque de recalentamiento este envía los datos de presión 2 (presión a la entrada del recalentamiento) y el dato de entropía 1 (entropía que se determina en la entrada de la turbina) y estos dos datos se envían al bloque de entalpía 2 y luego al bloque de interpolación para después ser almacenado en el bloque de recalentamiento como la entalpía a la entrada del recalentamiento. Enseguida el bloque de recalentamiento selecciona los datos de la presión a la entrada del recalentamiento y la temperatura del vapor que entrega el generador de vapor, estos datos son enviados a los bloques de presión 2 y temperatura 1 para que con ambos datos se determine el valor de la entalpía 3 y entropía 3 y que son enviados al bloque de interpolación y el resultado es regresado al bloque de recalentamiento y finalmente se tienen los parámetros a la salida del recalentamiento o a la entrada de la turbina en la segunda

etapa. Posteriormente el bloque de recalentamiento manda los datos a los bloques de presión 4 y entropía 3 los cuales son necesarios para determinar los datos de la entalpía a la entrada del condensador y la entalpía de la parte líquida que entra al condensador, estas entalpías se encuentran en los bloques de saturación y humedad los cuales mandan su resultado al bloque de interpolación y este finalmente lo regresa al bloque de recalentamiento y este envía todos los datos almacenados de las diferentes interpolaciones al bloque de resultado final para que este procese todos los datos y muestre en pantalla el resultado de la simulación.

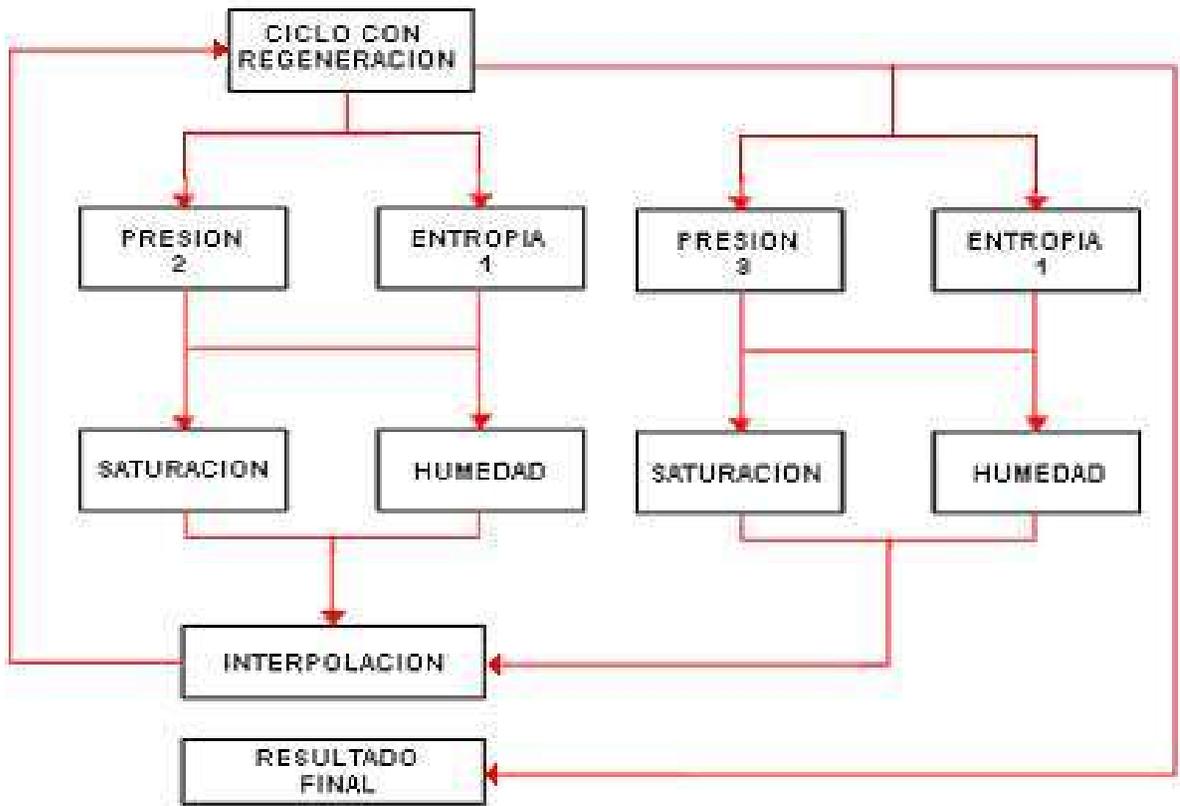


Figura 3.6 Secuencia de programación para una central de tipo regenerativo
Operando con vapor sobrecalentado

De la misma manera que en los demás procesos en el ciclo regenerativo figura 3.6 los parámetros obtenidos a la entrada de la turbina son descritos en la figura 3.2 y el resultado es almacenado en el bloque de regeneración. Una vez determinados los primeros parámetros el bloque de regeneración envía los datos de presión 2 y entropía 1 (dato obtenido de la primera de la figura 3.2) para determinar los valores de la entalpía a la salida de la extracción de la turbina o entrada al calentador tanto de la

mezcla de vapor y agua como la entalpía del líquido datos que son enviados a una interpolación para posteriormente ser almacenados en el bloque de regeneración. Para determinar la entalpía a la salida de la turbina o entrada al condensador el bloque de regeneración selecciona los datos en esta etapa del ciclo que son la presión 3 (presión a la salida de la turbina o entrada al condensador) así como la entropía 1 que es el mismo valor de la entrada de la turbina o salida del generador de vapor y son enviados a los bloques de presión 3 y entropía 1 datos que son utilizados para encontrar los datos de la entalpía de la mezcla de vapor y agua y la entalpía del líquido que se encuentra en este punto del ciclo y que son enviados la bloque de interpolación y posteriormente enviados al bloque de regeneración y finalmente este envía todos los datos almacenados al bloque de resultado final el cual se encarga de procesar la información para mostrar un resultado final.

La figura 3.7 muestra la secuencia de programación de un ciclo Rankine combinado donde la programación de este ciclo inicia cuando se aplican los datos con los que opera la central y este proceso se describe en la figura 3.2 donde son calculados los parámetros de entalpía y entropía a la entrada de la turbina.

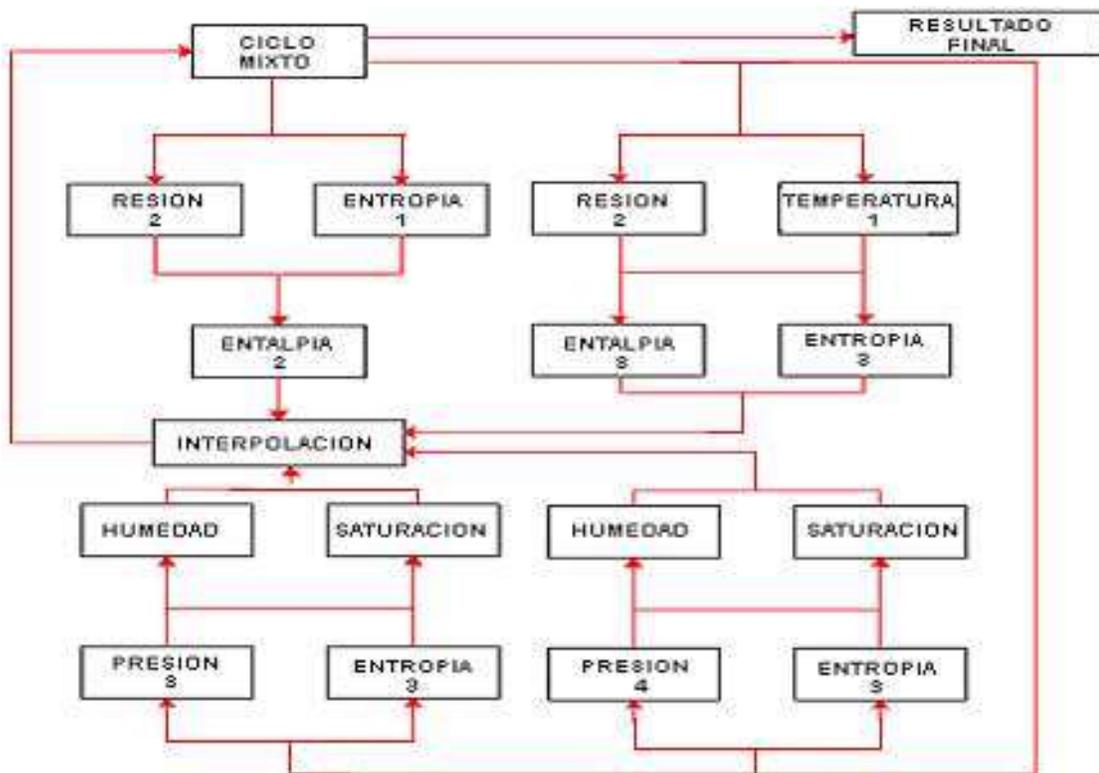


Figura 3.7 Secuencia de programación para una central de tipo mixto Operando con vapor sobrecalentado

En la figura 3.7 muestra los diferentes procesos que tiene el bloque de ciclo mixto el cual tiene la función de almacenar y seleccionar los tipos de datos que le son enviados del bloque entrada de datos figura 3.2 y posteriormente enviarlos a los bloques correspondientes para su procesamiento y así determinar un valor final.

Para el caso específico de la figura 3.7 los primeros valores que se determinan son la entalpía 2 que es el valor que se tiene a la entrada del recalentamiento y es determinado por los bloques de la presión 2 y la entropía 1 y donde el bloque de entalpía 2 envía a interpolar el dato correspondiente para después ser almacenado en el bloque de ciclo mixto. Para el cálculo de la entalpía a la salida de la etapa de recalentamiento o entrada de la turbina en la segunda etapa, el bloque de ciclo mixto envía los datos de presión 2 y temperatura 1 a los bloques de entalpía 3 y entropía 3 los cuales determinan los valores que serán enviados al bloque de interpolación y posteriormente enviados al bloque de ciclo mixto. Una vez realizado este ciclo de operaciones nuevamente el ciclo mixto envía los datos de presión 3 que es la presión con la que opera el calentador y el valor de la entropía 3 que es el valor de entrada a la segunda etapa de la turbina con estos valores los bloques de presión 3 y entropía 3 seleccionan los datos correspondientes para enviarlos a los bloques de humedad y saturación que son los que definen los valores de la entalpía de la mezcla líquido – vapor. La entalpía es posteriormente enviada al bloque de interpolación para después ser enviada al bloque de ciclo mixto y ahí almacenar el valor.

La penúltima ejecución del ciclo mixto es enviar los datos de la salida de la última etapa de la turbina o bien la entrada del condensador, y son los datos de presión 4 y entropía 3 los cuales son enviados a los bloques de presión 4 y entropía 3 que son los bloques que definen con exactitud el tipo de valor. Este valor es usado en los bloques de humedad y saturación que tienen la finalidad de encontrar los valores de la entalpía de la mezcla de agua-vapor y la entalpía de la cantidad de líquido que expulsa la turbina. Una vez encontrados los valores de las entalpías estos se envían al bloque de interpolación. Ya realizada esta operación se envían los datos al bloque de ciclo mixto. La última acción de este bloque es enviar todos los datos almacenados al bloque de resultado final en donde se realizan las operaciones correspondientes dependiendo de la fórmula que define la eficiencia del ciclo Rankine.

3.4 Software de una termoeléctrica

El diseño del software diseñado en esta tesis para una central termoeléctrica se basa en la construcción del diagrama de Mollier, las regiones donde opera el vapor y en las distintas modificaciones del ciclo Rankine. Para que el software pueda determinar el comportamiento de cada tipo de central es necesario tener datos iniciales que el usuario deberá aplicar y en base a estos se realizará una serie de búsquedas de parámetros que son necesarios para realizar las operaciones y llegar a un resultado final.

El código del software para una central termoeléctrica se muestra en el apéndice A de esta tesis. El código mostrado está hecho en el lenguaje de programación java, lenguaje que está disponible para el desarrollo de programas de uso general.

CAPITULO 4

Pruebas y Resultados

4.1 Introducción

Una sustancia que tiene una composición química fija recibe el nombre de sustancia pura. El agua, el nitrógeno y el helio y el dióxido de carbono son sustancias puras. Una mezcla de dos o más fases de una sustancia pura sigue siendo sustancia pura, siempre que la composición química de las fases sea la misma.

Existen varias situaciones prácticas en que dos fases de una sustancia pura coexisten en equilibrio como se puede ver en la figura 4.2b. El agua existe como una mezcla de líquido y vapor en la caldera y en el condensador de una central termoeléctrica. El refrigerante pasa de líquido a vapor en el congelador de un refrigerador.

Y puesto que el agua es una sustancia que puede existir en su fase líquida a distintas presiones y temperaturas. A presión atmosférica el agua se puede encontrar como líquido en un amplio rango de temperaturas como se puede ver en la figura 4.2a, si la temperatura es elevada existirá como vapor como se muestra en la figura 4.2c. Si se modifica la presión también tendrá que modificarse la temperatura. De tal modo que para una determinada fase tendremos diferentes presiones y temperaturas. En la figura 4.1 se muestra el diagrama T – V de una sustancia pura.

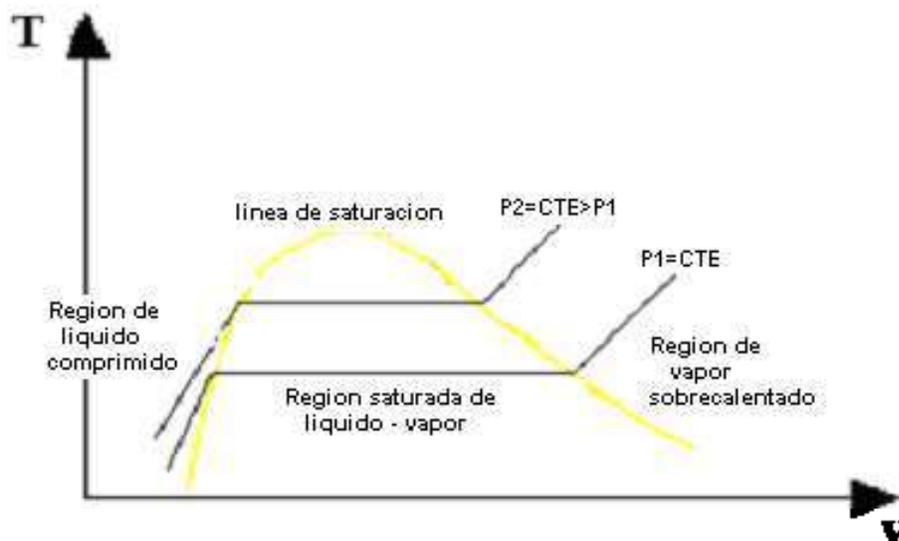


Figura 4.1 Diagrama T – V de una sustancia pura

Las condiciones de presión y temperatura en la que las dos fases coexisten en equilibrio se llaman condiciones de saturación y a la presión y a la temperatura correspondiente, se les denomina presión de saturación y temperatura de saturación respectivamente.

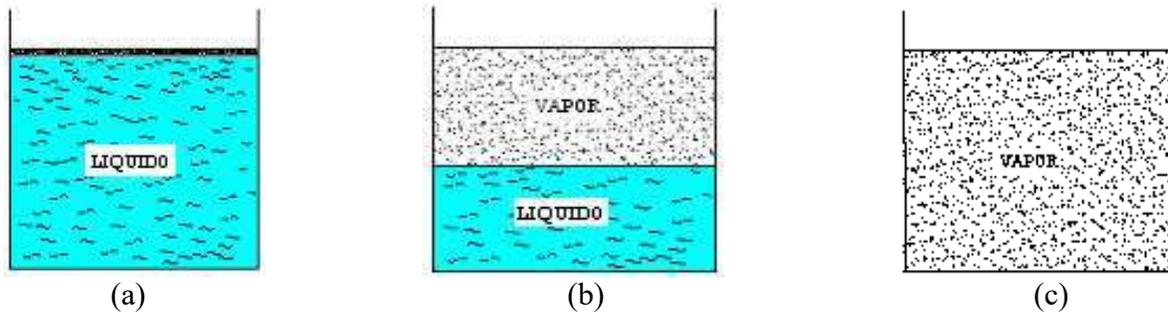


Figura 4.2 Distintas fases de una sustancia pura

En el recipiente mostrado de la figura 4.2(a) se tiene aproximadamente una temperatura de 20°C por tanto el agua se encuentra en su fase líquida. En el recipiente (b) se aplica calor al sistema hasta que exista el cambio de fase.

En el recipiente (c) se ha seguido aplicando calor. Por tanto se tendrá solamente vapor. La temperatura del vapor trata de estar en equilibrio con la fuente que le proporciona calor debido a la ley cero de la termodinámica.

Cuando un sistema se encuentra bajo las condiciones de saturación se conoce como fase saturada. Llamándole líquido subenfriado cuando la temperatura sea menor que la de saturación correspondiente a la presión dada, líquido comprimido cuando la presión sea mayor a la de saturación correspondiente a la temperatura dada en la figura 4.1.

Si existe como vapor será vapor saturado si esta bajo las condiciones de saturación, vapor sobrecalentado o recalentado cuando la temperatura es mayor a la de saturación correspondiente la presión dada figura 4.1.

4.2 Simulación de una central termoeléctrica convencional caso vapor saturado seco

Los componentes asociados a una central termoeléctrica convencional (caldera, turbina, condensador, bomba), figura 4.3, son dispositivos de flujo estable por tanto es posible analizar los cuatro procesos que conforman al ciclo Rankine como procesos de flujo estable.

Los cambios de la energía cinética y potencial son pequeños con respecto al trabajo y a la transferencia de calor, por tanto casi siempre se ignoran. De este modo la ecuación de energía de flujo estable por unidad de masa de vapor es,

$$(q_{entra} - q_{sale}) + (w_{entra} - w_{sale}) = h_o - h_i \quad (4.1)$$

Para la simulación de este tipo de central es necesario aplicar los conceptos termodinámicos que caracterizan al sistema como son el isobárico (presión cte.), el isentrópico (entropía cte.), el isotérmico (temperatura cte.), en la figura 4.3 se muestran procesos termodinámicos de los componentes de una central convencional.

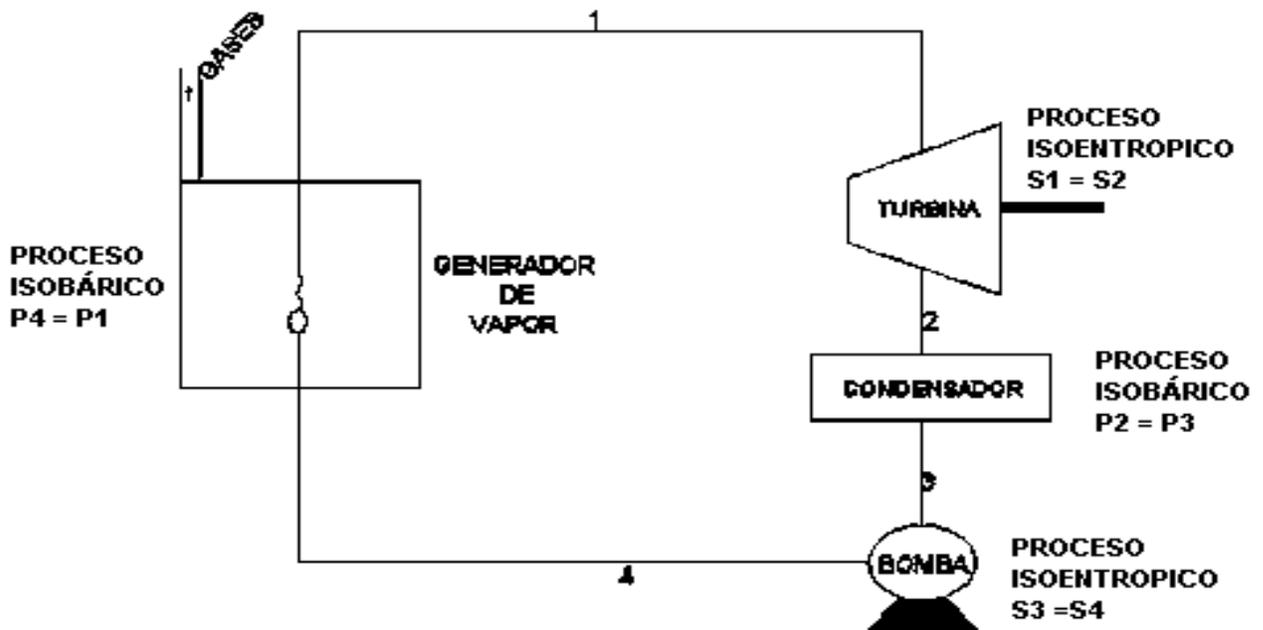


Figura 4.3 Procesos termoeléctricos de central convencional

De la figura 4.3 se puede calcular el trabajo realizado por la turbina y la eficiencia del sistema usando el diagrama de Mollier y de manera alternativa utilizando software de simulación.

Para obtener el trabajo desarrollado por la turbina se hace uso de la ecuación (4.2). Y la eficiencia del sistema, es dada por la ecuación 4.3.

$$W_{tur} = h_1 - h_2 \quad (4.2)$$

$$e = \left(\frac{h_1 - h_2}{h_1 - h_4} \right) \quad (4.3)$$

Los valores iniciales para el sistema son los siguientes:

$$P_1 = 440.7 \text{ lbf / in}^2$$

$$P_2 = 1.10 \text{ lbf / in}^2$$

Siendo el vapor que entra a la turbina de vapor saturado seco

4.2.1. Mediante el diagrama de Mollier

A partir de diagrama de Mollier de la figura 4.4 y de la interpolación del diagrama de Mollier se obtienen las entalpías h_1 y h_2 , con estos dos parámetros y en base a la ecuación (4.2) obtenemos el trabajo realizado por la turbina en la tabla 4.1 se interpola para obtener h_1 y s_1 .

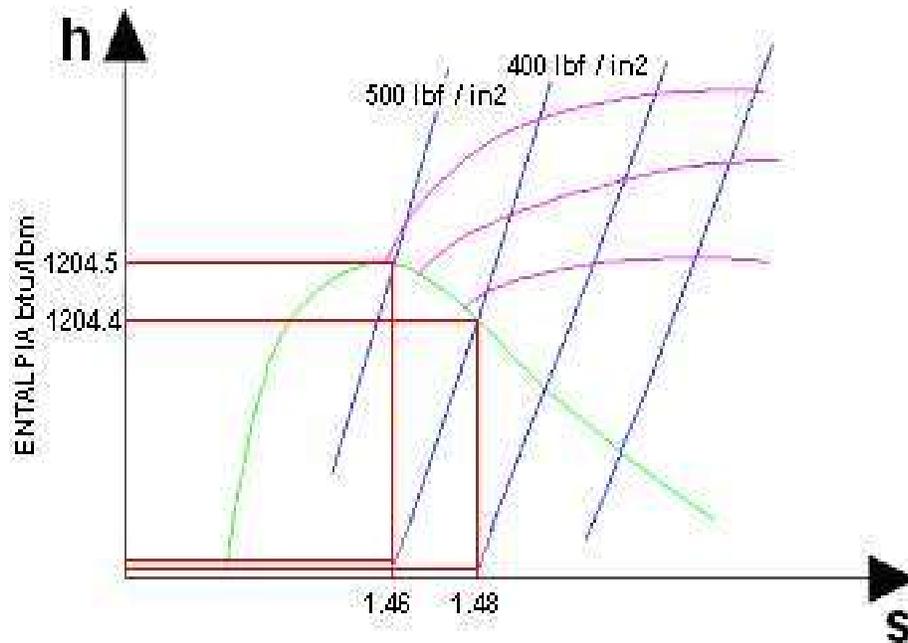


Figura 4.4 Diagrama de Mollier

Tabla 4.1 interpolación de entalpía y entropía alrededor de una presión de 440.7 lbf/in²

		h1	s1
Presión original	440.7		
Presión menor	400.0	1204.5	1.48
Presión mayor	500.0	1204.4	1.46
	-100.0	0.1	0.021
	40.7		
		1204.45	1.475

Por tanto $h_1 = 1204.45$ btu/lbm

Para el cálculo de h_2 se realiza mediante una interpolación doble mostrada en la tabla 4.2 y 4.3 utilizando s_1 y la presión 2

Interpolando para obtener h_2

Primera interpolación con s_1

Tabla 4.2 Interpolación de entalpía y entropía alrededor de una entropía de 1.475

		h_{1.5}	S₁
Entropía original	1.475		
Entropía menor	1.46	838	820
Entropía mayor	1.48	845	828
	-0.02	-7	-8
	0.01		
		841.5	824

Segunda interpolación con P2

Tabla 4.3 Interpolación de entalpía alrededor de una presión de 1.10 lbf/in²

		h2
Presión original	1.10	
Presión menor	1.0	824
Presión mayor	1.5	841.5
	-0.5	-17.5
	0.1	
		827.5

Por tanto $h_2 = 827.5$ btu/lbm

Aplicando ec. (4.2)

$$W_{tur} = h_1 - h_2$$

$$W_{tur} = 1204.45 - 827.5 = 376.95 \text{ btu/lbm}$$

Para el caso de la eficiencia del sistema se requieren de las entalpías h_3 y h_4 las cuales se obtienen con la presión 2 y utilizando las tablas de vapor de agua como $P_2 = 1.10$ interpolamos con los valores más cercanos como se muestra en la tabla 4.4.

Tabla 4.4 Interpolación de entalpía y entropía alrededor de una presión de 1.10 lbf/in^2

		h_{f2}	S_{fg2}
Presión original	1.10		
Presión menor	1.0	69.70	1036.3
Presión mayor	1.5	81.80	1029.2
	-0.5	-12.1	7.1
	0.10		
		72.12	1034.88

Por tanto $h_3 = h_{f2} = 72.12 \text{ btu/lbm}$

Para obtener h_4 consideramos que $h_4 = h_3$ por ser líquido saturado .

Por tanto $h_4 = 72.12 \text{ btu/lbm}$

Aplicando la ecuación (4.3) obtenemos la eficiencia del sistema

$$e = \left(\frac{1204.45 - 827.5}{1204.45 - 72.12} \right) = 0.332 = 33.20 \%$$

4.2.2. Usando software de simulación

Con los parámetros iniciales de la central termoeléctrica aplicados al programa de simulación, el cual tiene la capacidad de determinar los valores de entalpía y entropía, así mismo como realizar interpolaciones de datos que no se encuentran bien definidos por el diagrama de Mollier.

Al aplicar datos al programa de simulación, se presenta una pantalla como la mostrada en la siguiente figura.

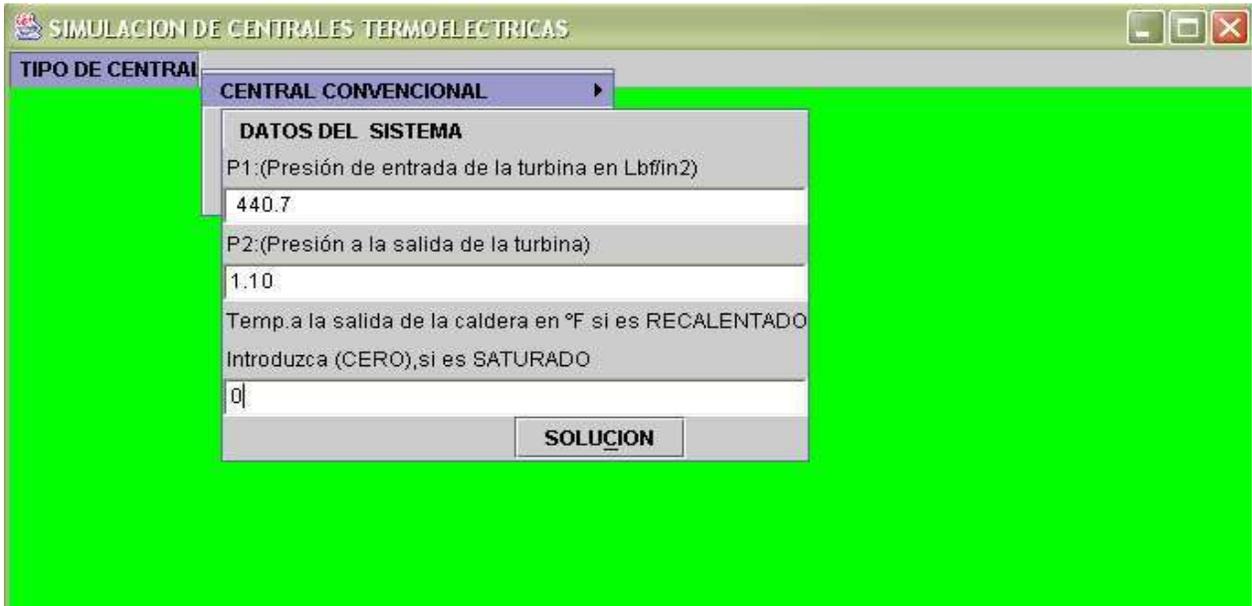


Figura 4.5 Datos de entrada al simulador

Una vez que se introdujeron los datos, el sistema realiza los cálculos correspondientes para mostrarnos una imagen como la mostrada en la figura 4.6 con los resultados que corresponden.

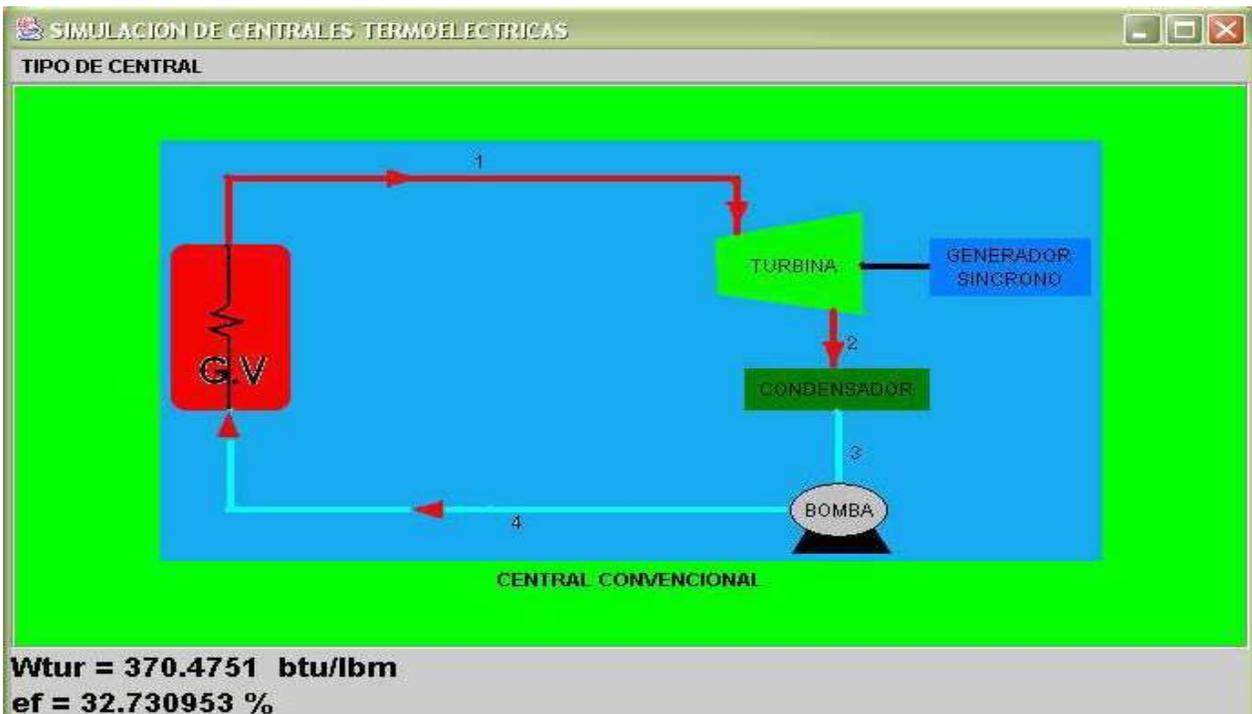


Figura 4.6 Resultados del simulador

4.3 Prueba de una termoelectrica con vapor sobrecalentado

Para aumentar la eficiencia de un ciclo de potencia es necesario aumentar la temperatura del fluido de trabajo de la caldera y disminuir la temperatura de entrada del condensador. Otra forma de cómo incrementar la eficiencia es aumentando la presión a la entrada del generador de vapor lo que eleva automáticamente la temperatura de ebullición. Esto a su vez incrementa la temperatura promedio a la que se añade calor al vapor y de ese modo se incrementa la eficiencia térmica del ciclo.

Para ejemplificar lo antes descrito calculemos la eficiencia del ciclo Rankine convencional mostrado en la figura 4.7 con una presión de 734.5 lbf / in^2 y una temperatura de 752°F a la entrada de la turbina y 1.10 lbf / in^2 a la entrada de condensador.

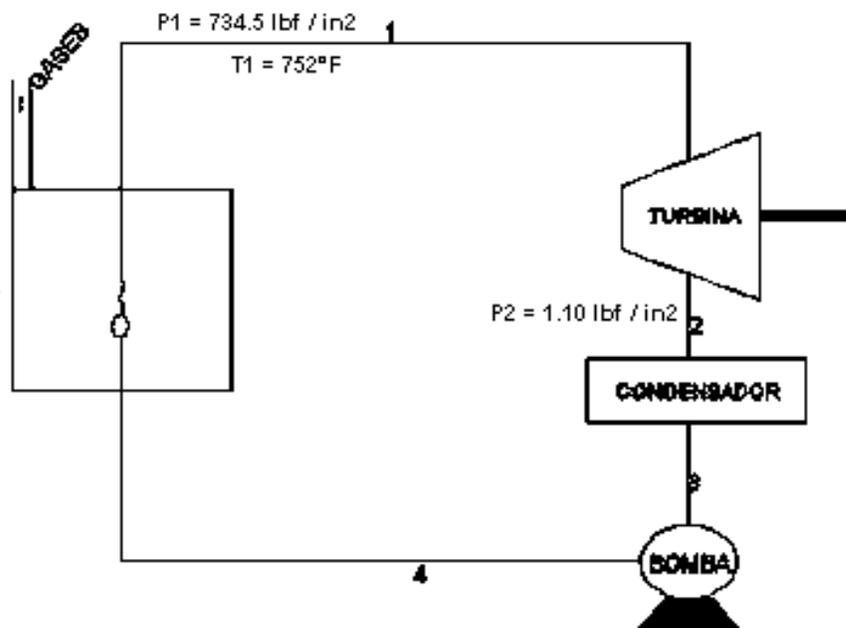


Figura 4.7 Central termoelectrica convencional

Con los datos establecidos con anterioridad y con el uso del diagrama de Mollier, figura 4.8, obtenemos los siguientes parámetros mostrados en la tabla 4.5.

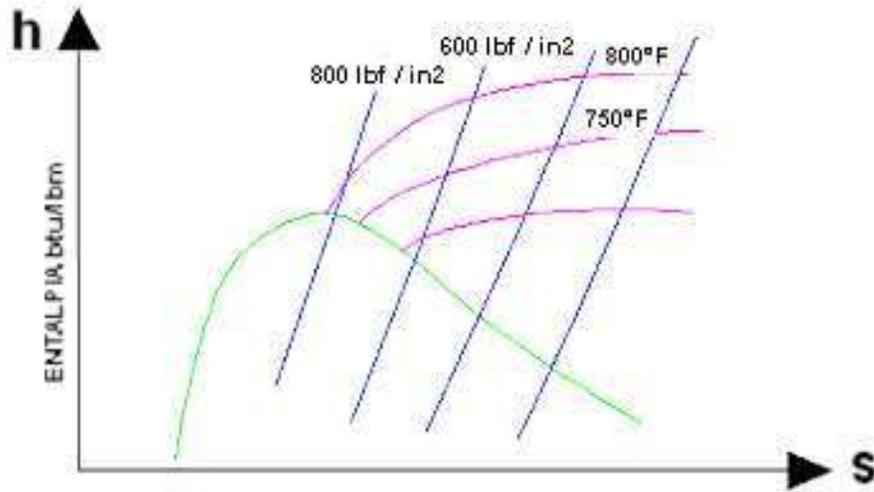


Figura 4.8 Diagrama de Mollier

En la tabla 4.5 y 4.6 se interpolan los resultados del diagrama de Mollier para obtener la entalpía h_1 y s_1 .

Primera interpolación con P_1 .

Tabla 4.5 Interpolación de entalpías y entropías alrededor de una presión de 734.5 lbf/in^2

		h_{800}	h_{750}	S_{800}	S_{750}
Presión original	734.5				
Presión mayor	800	1400	1370	1.60	1.575
Presión menor	600	1408	1380	1.64	1.61
	200	-8	-10	-0.04	-0.035
	-65.5				
		1402.62	1373.27	1.613	1.586

Segunda interpolación con T1

Tabla 4.6 Interpolación de Entalpía y entropía alrededor de una temperatura de 752°F

		H₁	S₁
Temperatura original	752		
Temperatura mayor	800	1402.6	1.613
Temperatura menor	750	1373.27	1.586
	50	29.33	0.027
	2		
		1374.44	1.587

Por tanto $h_1=1374.44$ btu/lbm

$$S_1=1.587 \text{ btu/lbm}^\circ\text{R}$$

Para el cálculo de h_2 , se realiza mediante una interpolación doble y para ello, se hace uso del diagrama de Mollier obteniéndose los mostrados en las tablas 4.7 y 4.8.

Primera interpolación con s_1 .

Tabla 4.7 Interpolación de Entalpía y entropía alrededor de una entropía de 1.587

		h₁	h_{1.5}
Entropía original	1.587		
Entropía menor	1.58	881.0	900.0
Entropía mayor	1.60	891.0	914.0
	-0.02	-10	-14
	0.007		
		884.5	904.9

Segunda interpolación con P2

Tabla 4.8 Interpolación de Entalpía alrededor de una presión de 1.10 lbf/in²

		h₂
Presión original	1.10	
Presión menor	1.0	884.5
Presión mayor	1.5	904.9
	-0.5	-20.4
	0.10	
		888.58

Por tanto $h_2 = 888.58$ btu/lbm

aplicando ec.(4.2)

$$W_{tur} = h_1 - h_2$$

$$W_{tur} = 1374.44 - 888.58 = 485.86 \text{ btu/lbm}$$

Para el caso de la eficiencia del sistema se requieren de las entalpías h_3 y h_4 las cuales se obtienen con la presión 2 y utilizando las tablas de vapor de agua como $P_2 = 1.10$. En la tabla 4.9 interpolamos con los valores mas cercanos.

Tabla 4.9 Interpolación de Entalpías alrededor de una presión de 1.10 lbf/in²

		h_{f2}	h_{fg2}
Presión original	1.10		
Presión menor	1.0	69.70	1036.3
Presión mayor	1.5	81.80	1029.2
	-0.5	-12.1	7.1
	0.10		
		72.12	1034.88

Por tanto $h_3 = h_{f2} = 72.12$ btu/lbm

Para obtener h_4 consideramos que $h_4 = h_3$ por ser líquido saturado.

Por tanto $h_4 = 72.12$ btu/lbm

Aplicando la ecuación (4.3) obtenemos la eficiencia del sistema

$$e = \left(\frac{1374.44 - 888.58}{1374.44 - 72.12} \right) = 0.373 = 37.3 \%$$

4.3.1. Usando software de simulación

Con los parámetros iniciales con que opera el sistema, se aplican al software de simulación para que este pueda realizar las operaciones necesarias para así poder determinar la eficiencia. Al aplicar los datos al programa de simulación, se muestra una pantalla como la mostrada en la figura 4.9.

The screenshot shows a software interface for simulating thermal power plants. The title bar reads "SIMULACION DE CENTRALES TERMOELECTRICAS". A dropdown menu labeled "TIPO DE CENTRAL" is open, showing "CENTRAL CONVENCIONAL" selected. Below this is a "DATOS DEL SISTEMA" form with the following fields and values:

Variable	Value
P1: (Presión de entrada de la turbina en Lb/in ²)	734.5
P2: (Presión a la salida de la turbina)	1.10
Temp. a la salida de la caldera en °F si es RECALENTADO Introduzca (CERO), si es SATURADO	752

A "SOLUCION" button is located at the bottom of the form.

Figura 4.9 Datos de entrada al simulador

Una vez introducidos los valores iniciales al sistema se nos muestra una imagen como la mostrada en la figura 4.10.

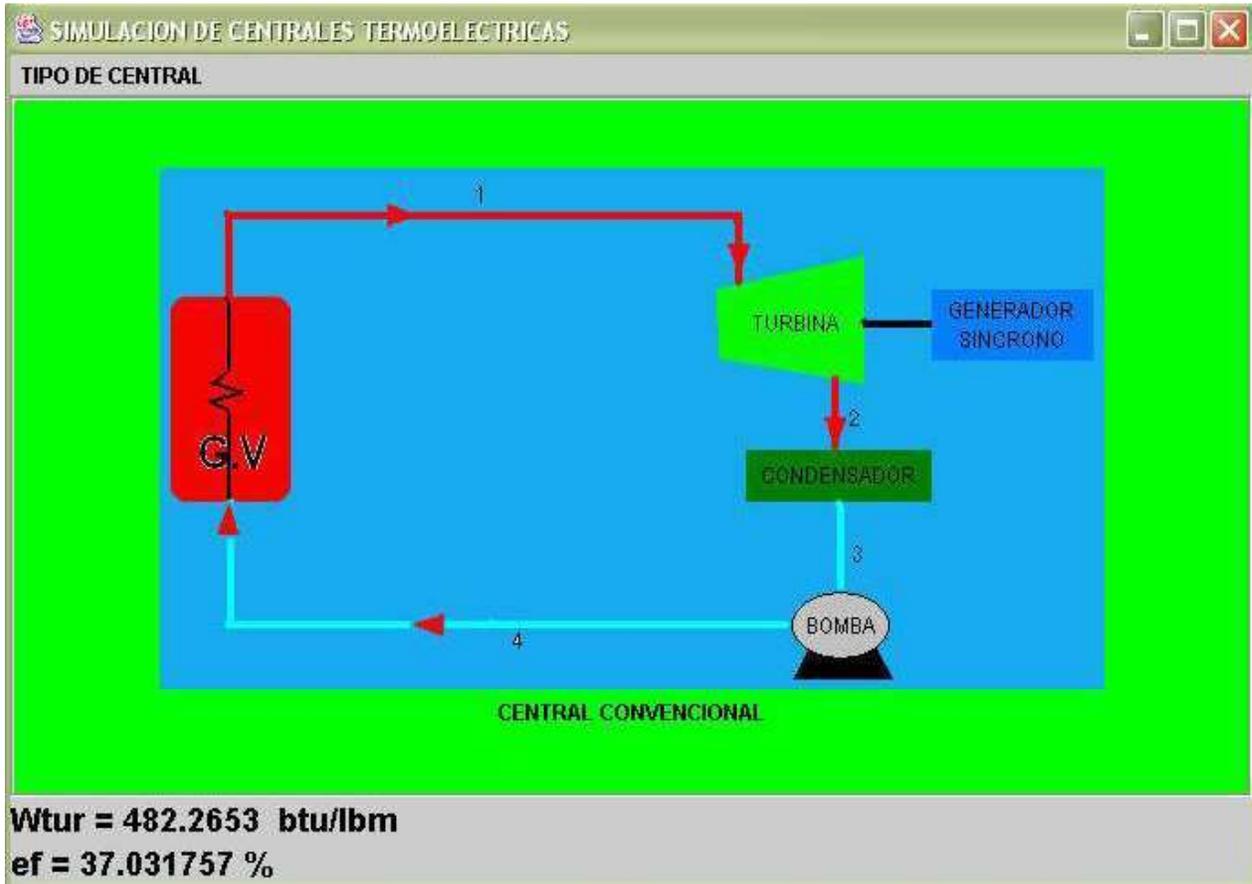


Figura 4.10 Resultados del simulador

4.4 Prueba de una termoeléctrica con ciclo de recalentamiento

El recalentamiento consiste en la adición de calor al vapor que se ha extraído después de haber pasado por una etapa de la turbina para eliminar la humedad, este recalentamiento se realiza a presión constante.

El sobrecalentamiento tiende a ayudar a la turbina pero termodinámicamente no aporta aumento considerable a la eficiencia térmica del ciclo. En la figura 4.11 se muestran los procesos termodinámicos de una central termoeléctrica con una etapa de recalentamiento los cuales son utilizados para obtener los parámetros del ciclo.

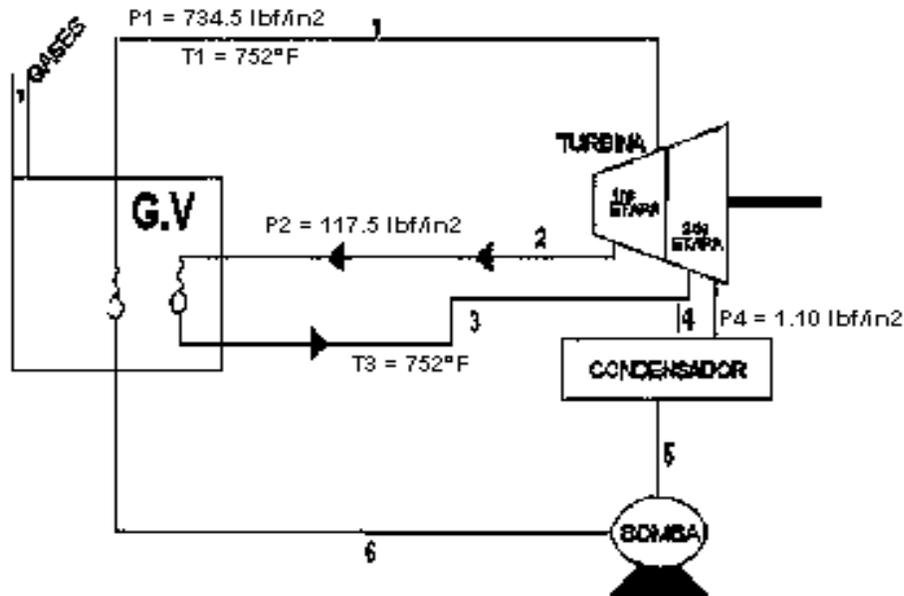


Figura 4.11 Central termoeléctrica con recalentamiento

Con los datos mostrados en la figura 4.11 se calcula el trabajo realizado por la turbina y la eficiencia del sistema usando el diagrama de Mollier y de manera alternativa utilizando software de simulación.

Para obtener el trabajo desarrollado por la turbina se tiene la ecuación (4.4). Del sistema mostrado en la figura 4.11, y la eficiencia que proporciona el sistema con la ecuación (4.5). La solución de este sistema se puede realizar mediante el diagrama de Mollier y también por el software de simulación.

$$W_{tur} = (h_1 - h_2) + (h_3 - h_4) \quad (4.4)$$

$$e = \left(\frac{(h_1 - h_2) + (h_3 - h_4)}{(h_1 - h_4) + (h_3 - h_2)} \right) \quad (4.5)$$

Observando la figura 4.11 se tienen los siguientes valores del sistema:

$$P1 = 734.5 \text{ lbf/in}^2$$

$$P2 = 117.5 \text{ lbf/in}^2$$

$$P4 = 1.10 \text{ lbf/in}^2$$

$$T1 = 752 \text{ lbf/in}^2$$

Con el diagrama de Mollier (figura 4.12) y los datos iniciales del sistema se obtienen los parámetros mostrados en las tablas 4.10 y 4.11.

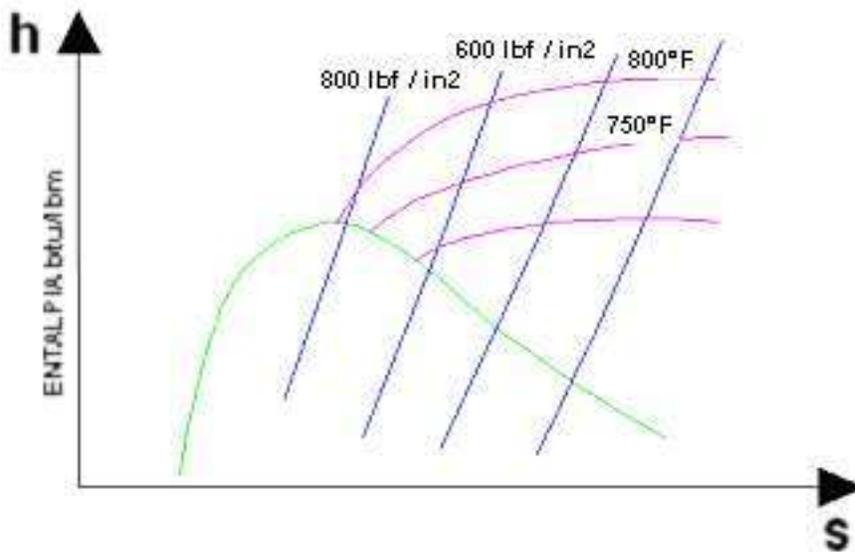


Figura 4.12 Diagrama de Mollier

Los resultados obtenidos del diagrama de Mollier se interpolan en las tablas 4.10 y 4.11 para obtener h_1 y s_1 .

Primera interpolación con P_1 .

Tabla 4.10 Interpolación de Entalpías y entropías alrededor de una presión de 734.5 lbf/in^2

		h_{800}	h_{750}	S_{800}	S_{750}
Presión original	734.5				
Presión mayor	800	1400	1370	1.60	1.575
Presión menor	600	1408	1380	1.64	1.61
	200	-8	-10	-0.04	-0.035
	-65.5				
		1402.62	1373.27	1.613	1.586

Segunda interpolación con T1

Tabla 4.11 Interpolación de Entalpía y entropía alrededor de una temperatura de 752°F

		h₁	S₁
Temperatura original	752		
Temperatura mayor	800	1402.6	1.613
Temperatura menor	750	1373.27	1.586
	-50	29.35	0.027
	-48		
		1374.44	1.587

Por tanto, $h_1 = 1374.44$ btu/lbm

$$S_1 = 1.587 \text{ btu/lbm}^\circ\text{R}$$

Para el cálculo de h_2 se realiza con una interpolación doble usando las tablas 4.12 y 4.13 utilizando S_1 y la presión 2

Interpolando para obtener h_2

Primera interpolación con S_1

Tabla 4.12 Interpolación de Entalpía y entropía alrededor de una entropía de 1.58 lbf/in²

		h₁₀₀	S₁₅₀
Entropía original	1.58		
Entropía menor	1.58	1170.0	1202.0
Entropía mayor	1.60	1188.0	1219.0
	-0.02	-18	-17
	0.007		
		1176.30	1207.95

Segunda interpolación con P2

Tabla 4.13 Interpolación de Entalpía alrededor de una presión de 117.52 lbf/in²

		h₂
Presión original	117.52	
Presión menor	100	1176.30
Presión mayor	150	1207.95
	-50	-31.95
	17.52	
		1187.49

Por tanto, $h_2 = 1187.49$ btu/lbm

Para calcular el valor de h_3 y S_3 se utilizan los valores P_2 y T_1 , que son los valores que se obtienen una vez que se a recalentado el vapor que se extrajo después de haber realizado un trabajo en la turbina. Por tanto con la presión $P_2 = 117.5$ lbf/in² y $T_1 = 752^\circ\text{F}$ en las tablas 4.14 y 4.15 interpolamos los valores obtenidos del diagrama de Mollier.

Primera interpolación con P2

Tabla 4.14 Interpolación de Entalpías y entropías alrededor de una presión de 117.5 lbf/in²

		h₈₀₀	h₇₅₀	S₈₀₀	S₇₅₀
Presión original	117.5				
Presión menor	100	1429	1403	1.78	1.82
Presión mayor	150	1426	1400	1.80	1.84
	-50	3	3	-0.02	-0.02
	17.5				
		1427.90	1401.95	1.78	1.82

Segunda interpolación con T1

Tabla 4.15 Interpolación de Entalpía y entropía alrededor de una temperatura de 752°F

		h3	S3
Temperatura original	752		
Temperatura mayor	800	1427.90	1.78
Temperatura menor	750	1401.95	1.82
	50	25.95	-0.04
	-48		
		1402.98	1.81

Por tanto $h_3 = 1402.98$ btu/lbm

$$S_3 = 1.81 \text{ btu/lbm}^\circ\text{R}$$

Para calcular h_4 se realiza mediante una interpolación doble realizada en las tablas 4.16 y 4.17 utilizando s_3 y la presión 4

Primera interpolación con s_3

Tabla 4.16 Interpolación de Entalpías y entropías alrededor de Una entropía de 1.81 btu/lbm°R

		h₁	h_{1.5}
Entropía original	1.81		
Entropía menor	1.80	1004.0	1029.0
Entropía mayor	1.82	1017.0	1040.0
	-0.02	-10	-11
	0.01		
		1009.0	1034.5

Segunda interpolación con P4

Tabla 4.17 Interpolación de Entalpía alrededor de una presión de 1.10 lbf/in²

		h4
Presión original	1.10	
Presión menor	1.0	1009.0
Presión mayor	1.5	1034.5
	-0.5	-25.5
	0.10	
		1014.1

Por tanto, $h_4 = 1014.1$ btu/lbm

Para determinar el trabajo realizado por la turbina se aplica la ecuación (4.4)

$$W_{tur} = (1374.44 - 1187.49) + (1402.98 - 1014.1) = 575.83 \text{ btu/lbm}$$

Para determinar la eficiencia del sistema se requieren de las entalpías h_5 y h_6 las cuales se obtienen con la presión 2 y utilizando las tablas de vapor de agua como $P_4 = 1.10$ interpolamos en la tabla 4.18 con los valores más cercanos.

Tabla 4.18 Interpolación de Entalpías alrededor de una presión de 1.10 lbf/in²

		h_{f4}	h_{fg4}
Presión original	1.10		
Presión menor	1.0	69.70	1036.3
Presión mayor	1.5	81.80	1029.2
	-0.5	-12.1	7.1
	0.10		
		72.12	1034.88

Por tanto, $h_5 = h_{f4} = 72.12$ btu/lbm

Para obtener h_6 consideramos que $h_6 = h_5$ por ser líquido saturado.

Por tanto, $h_6 = 72.12$ btu/lbm

Aplicando la ecuación (4.5) obtenemos la eficiencia del sistema

$$e = \frac{(1374.4 - 1187.49) + (1402.98 - 1014.1)}{(1374.44 - 72.12) + (1402.98 - 1187.49)} = 0.3793 = 37.93\%$$

4.4.1. Usando software de simulación

La incorporación de una etapa de recalentamiento en un ciclo Rankine convencional implica introducir un dato más al simulador el cual es la presión 2, dicha presión es la que entrega la turbina a la salida de su primera etapa por tanto los valores que se aplican al simulador son los que se describen en la figura 4.13.

The screenshot shows a software window titled "SIMULACIÓN DE CENTRALES TERMOELECTRICAS". On the left, there is a vertical menu labeled "TIPO DE CENTRAL" with two options: "CENTRAL CONVENCIONAL" and "CENTRAL CON RECALENTAMIENTO". The "CENTRAL CON RECALENTAMIENTO" option is currently selected. Below this menu is a form titled "DATOS DEL SISTEMA" with the following fields and values:

Variable	Value
P1:(Presión de entrada de la turbina en Lbf/in ²)	734.5
P2:(Presión a la entrada del recalentamiento)	117.5
P3:(Presión a la salida de la turbina)	1.10
Temp.a la salida de la caldera en °F si es RECALENTADO Introduzca (CERO),si es SATURADO	752

At the bottom of the form is a button labeled "SOLUCION".

Figura 4.13 Datos de entrada al simulador

Una vez introducidos los valores iniciales al software, se muestra una imagen con sus resultados correspondientes como se muestra en la figura 4.14

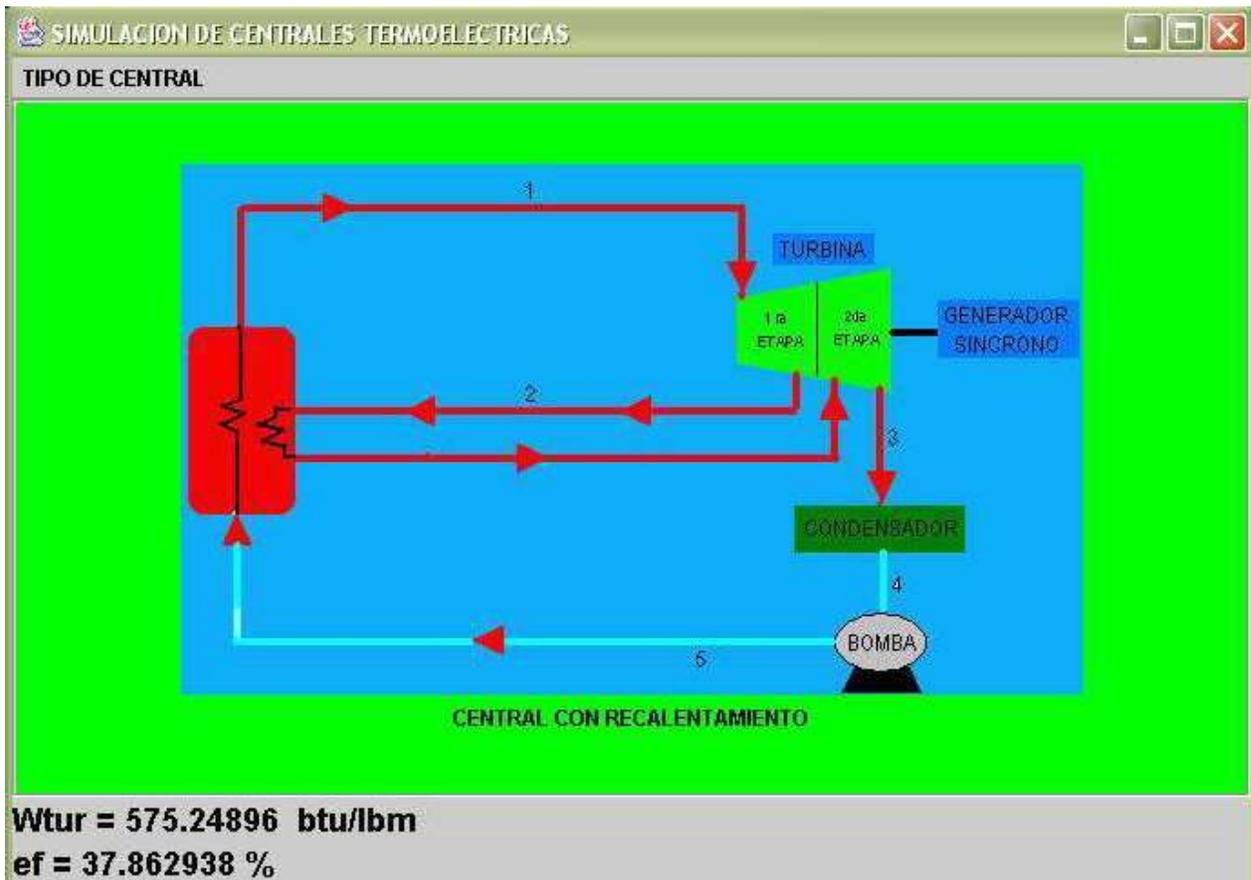


Figura 4.14 Resultados del simulador

4.5 Prueba de una termoeléctrica con ciclo de regenerativo

La regeneración no sólo mejora la eficiencia del ciclo si no también proporciona un medio conveniente de airear el agua de alimentación (eliminando el aire que se filtra en el condensador) para evitar la corrosión en la caldera. También ayuda a controlar la gran tasa de flujo volumétrico del vapor en las etapas finales de la turbina (debido a los grandes volúmenes específicos a bajas presiones). Por consiguiente la regeneración se emplea en todas las modernas plantas de energía de vapor desde su introducción a principios de la década de los veinte.

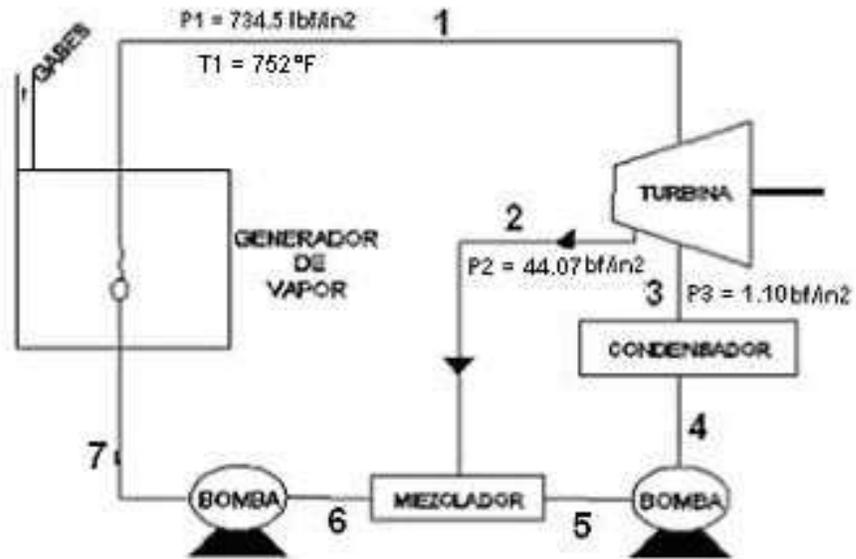


Figura 4.15 Central termoeléctrica con ciclo regenerativo

En la figura 4.15 se muestra un esquema de una central termoeléctrica con una etapa de regeneración para la cual se deberá determinar la cantidad de trabajo que desarrolla la turbina así como la eficiencia del ciclo. Estos parámetros serán determinados por las ecuaciones (4.6) y (4.7), y resolviendo mediante diagrama de Mollier y de manera alternativa aplicando software de simulación.

$$W_{tur} = (h_1 - h_2) + \left(1 - \frac{h_6 - h_5}{h_2 - h_5}\right) (h_2 - h_3) \quad (4.6)$$

$$e = \frac{(h_1 - h_2) + \left(1 - \frac{h_6 - h_5}{h_2 - h_5}\right) (h_2 - h_3)}{(h_1 - h_7)} \quad (4.7)$$

Los valores iniciales del sistema (figura 4.15) son:

$$P_1 = 734.5 \text{ lbf/in}^2$$

$$P_2 = 44.07 \text{ lbf/in}^2$$

$$P_3 = 1.10 \text{ lbf/in}^2$$

$$T_1 = 752^\circ\text{F}$$

En la tabla 4.19 y 4.20 se interpolan los resultados del diagrama de Mollier para obtener la entalpía h_1 y s_1 .

Primera interpolación con P1.

Tabla 4.19 Interpolación de Entalpías y entropías alrededor de una presión de 734.5 lbf/in²

		h₈₀₀	h₇₅₀	S₈₀₀	S₇₅₀
Presión original	734.5				
Presión mayor	800	1400	1370	1.60	1.575
Presión menor	600	1408	1380	1.64	1.61
	200	-8	-10	-0.04	-0.035
	-65.5				
		1402.62	1373.27	1.613	1.586

Segunda interpolación con T1.

Tabla 4.20 Interpolación de Entalpía y entropía alrededor de una temperatura de 752°F lbf/in²

		h₁	S₁
Temperatura original	752		
Temperatura mayor	800	1402.6	1.613
Temperatura menor	750	1373.27	1.586
	-50	29.35	0.027
	-48		
		1374.44	1.587

Por tanto $h_1=1374.44$ btu/lbm

$S_1=1.587$ btu/lbm^{°R}

Para determinar h_2 y h_3 se realiza mediante una interpolación doble, tablas 4.21 y 4.22 respectivamente utilizando S_1 , la presión 2 y la presión 3.

Primera interpolación con S1.

Tabla 4.21 Interpolación de Entalpías alrededor de una entropía de 1.587 btu/lbm°R

		h₁	h_{1.5}	h₄₀	h₅₀
Entropía original	1.587				
Entropía menor	1.58	881.0	900.0	1100.00	1120.00
Entropía mayor	1.60	891.0	914.0	1115.00	1130.00
	-0.02	-10	-14	-15	-10
	0.007				
		884.5	904.9	1105.25	1123.5

Segunda interpolación con P3

Tabla 4.22 Interpolación de Entalpías y entropías alrededor de las presiones de 44.07 y 1.10 lbf/in² respectivamente

	P2	h2	P3	h3
Presión original	44.07		1.10	
Presión menor	40	1105.25	1.0	884.5
Presión mayor	50	1123.5	1.5	904.9
	-10	-18.25	-0.5	-20.4
	4.07		0.10	
		1112.67		888.58

El resultado de la interpolación indica que $h_2 = 1112.67$ y $h_3 = 888.58$ btu/lbm. Para conocer el valor de h_4 y h_6 se utilizan las tablas de vapor de agua, y como $P_2 = 44.07$ lbf/in² y $P_3 = 1.10$ lbf/in², se usa la tabla 4.23 para mostrar la interpolación necesaria.

Tabla 4.23 Interpolación de Entalpías alrededor de las presiones de 44.07 y 1.10 lbf/in² respectivamente

	P2	h_{f2}	h_{fg2}	P3	h_{f3}	h_{fg3}
Presión original	44.07			1.10		
Presión menor	40	236.03	933.7	1.0	69.70	1036.3
Presión mayor	45	243.36	928.6	1.5	81.80	1029.2
	-5	-7.33	5.1	-0.5	-12.1	7.1
	4.07			0.10		
		241.99	929.54		72.12	1034.88

Por consiguiente del resultado de la interpolación nos indica que $h_{f2} = 241.99$ y $h_{f3} = 72.12$ y como $h_7 = h_6 = h_{f2}$ debido que la cantidad de vapor extraído de la turbina al mezclarse con el líquido que proviene del condensador sale del calentador como un líquido saturado. Por las mismas condiciones tenemos que $h_5 = h_4 = h_{f3}$ por tanto tenemos que:

$$h_7 = 241.99 \text{ btu/lbm}$$

$$h_4 = 72.12 \text{ btu/lbm}$$

Sustituyendo en las ecuaciones (4.6) y (4.7)

$$W_{tur} = (1374.44 - 1112.67) + \left(1 - \frac{241.99 - 72.12}{1112.67 - 72.12}\right)(1112.67 - 888.58) =$$

$$W_{tur} = 449.26 \text{ btu/lbm}$$

$$e = \frac{(1374.44 - 1112.67) + \left(1 - \frac{241.99 - 72.12}{1112.67 - 72.12}\right)(1112.67 - 888.58)}{(1374.44 - 241.99)} =$$

$$e = 0.396 = 39.6 \%$$

4.5.1 Usando software de simulación

Los datos de entrada que se aplican al ciclo con recalentamiento son aplicados a este ciclo regenerativo como son la presión 1, presión 2, presión 3 y temperatura 1. Con estos cuatro datos aplicados al software de simulación se realizan las operaciones necesarias para así poder determinar la eficiencia.

Aplicando datos al programa de simulación, se presenta una pantalla como se ilustra en la figura 4.16

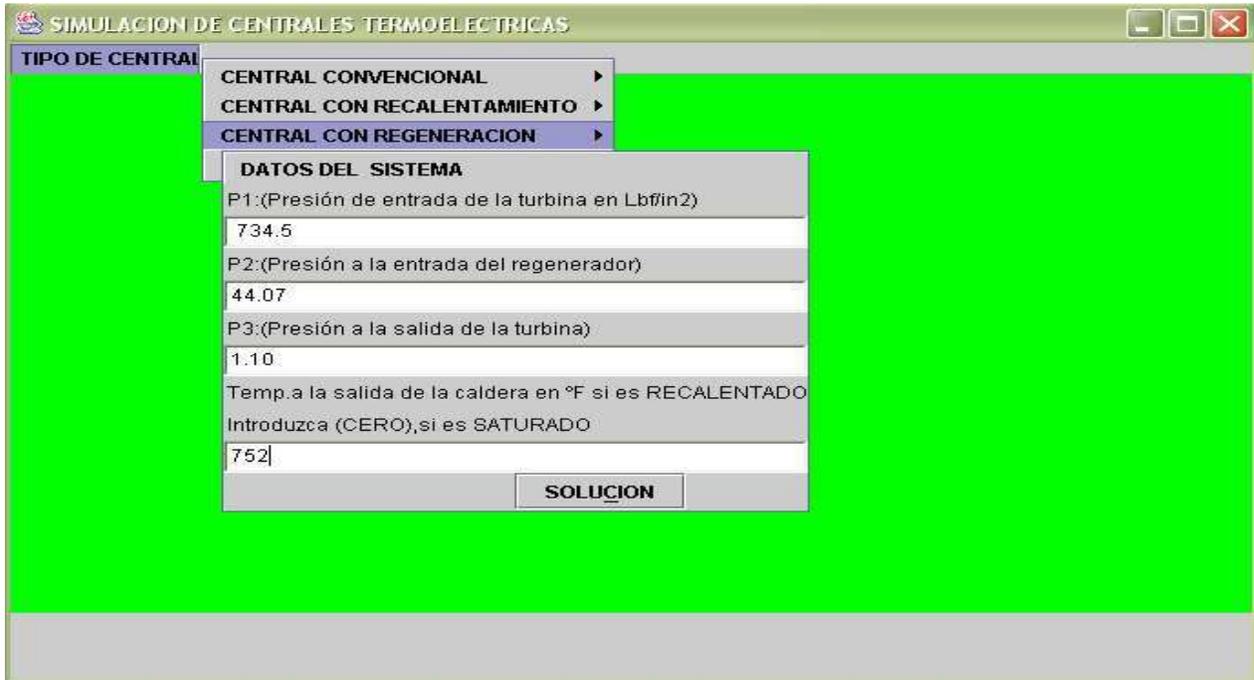


Figura 4.16 Datos de entrada al simulador

El simulador entrega directamente los resultados del problema los cuales se muestran en la figura 4.17

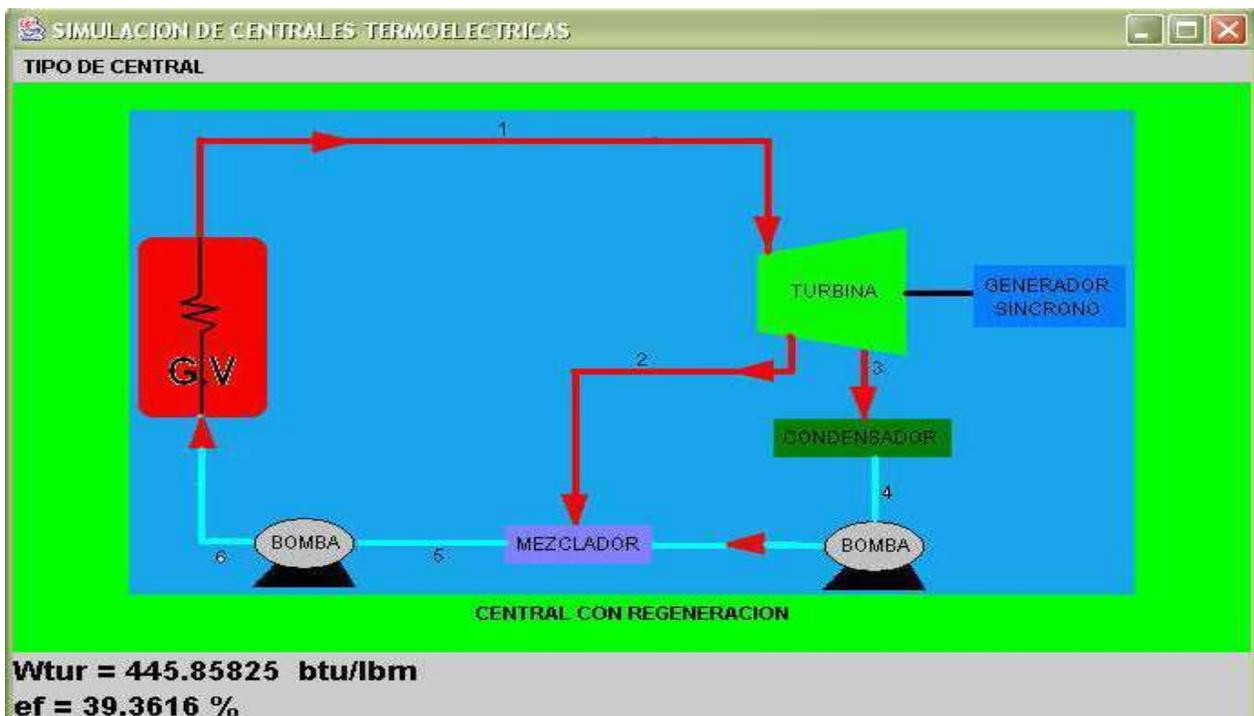


Figura 4.17 Resultados del simulador

4.6 Prueba de una central termoeléctrica con ciclo de recalentamiento y regeneración

Generalmente las centrales termoeléctricas de alta y mediana capacidad tienen ciclos de recalentamiento y de regeneración combinados es decir, que para un ciclo Rankine convencional se adiciona una etapa de recalentamiento y una etapa de regeneración cuya finalidad de estas es la de incrementar la eficiencia del sistema.

Normalmente las centrales modernas tienen hasta unas seis o siete etapas de pasos regenerativos y de recalentamiento para lograr eficiencias termodinámicas del orden del 40%. La contraparte de este tipo de modificaciones es el costo tanto en su construcción como en el mantenimiento. Para esto, en cada caso en particular se debe hacer un balance económico y en base a este determinar el número de extracciones y recalentamientos que se pueden realizar.

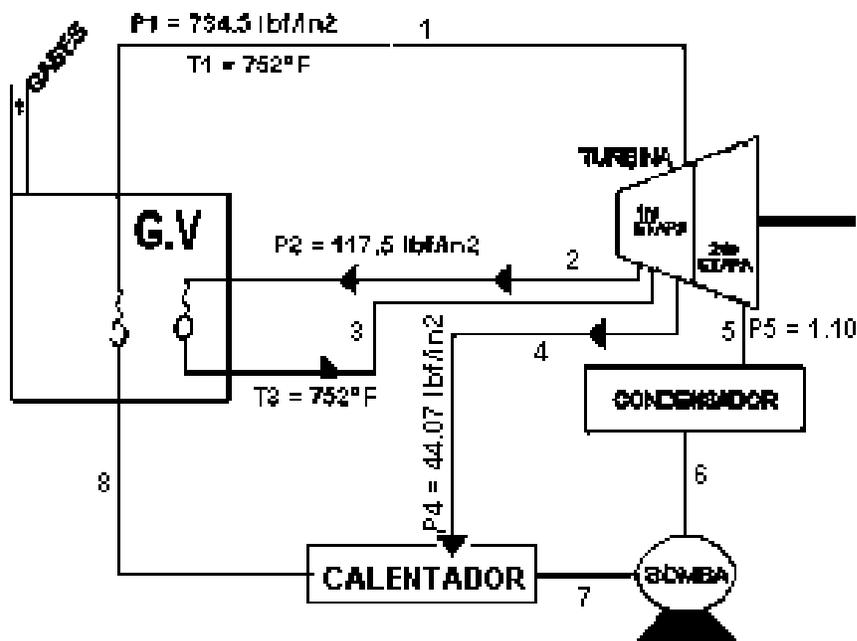


Figura 4.18 Central termoeléctrica mixta

En la figura 4.18 se muestra un arreglo donde se aprecia un ciclo Rankine mixto con una etapa de recalentamiento y una de regeneración. Se analizará el trabajo realizado por la turbina y la eficiencia que tiene el ciclo con estas dos modificaciones operando simultáneamente.

Las ecuaciones necesarias para obtener estos dos parámetros son las siguientes:

$$W_{tur} = (h1 - h2) + (h3 - h4) + \left(1 - \frac{h8 - h7}{h4 - h7}\right)(h4 - h5) \quad (4.8)$$

$$e = \frac{(h1 - h2) + (h3 - h4) + \left(1 - \frac{h8 - h7}{h4 - h7}\right)(h4 - h5)}{(h1 - h8) + (h3 - h2)} \quad (4.9)$$

Los valores iniciales del sistema (figura 4.18) son:

$$P1 = 734.5 \text{ lbf/in}^2$$

$$P2 = P3 = 117.5 \text{ lbf/in}^2$$

$$P4 = 44.07 \text{ lbf/in}^2$$

$$P5 = 1.10 \text{ lbf/in}^2$$

$$T1 = 752^\circ\text{F}$$

De los resultados obtenidos del diagrama de Mollier y de la interpolación de estos en las tablas 4.19 y 4.20 se obtienen $h1 = 1374.44$ y $S1 = 1.587$, que son los valores de entrada la turbina. Para determinar el valor de $h2$ se utiliza la presión ($P2$) y la entropía ($S1$), en la tablas 4.12 y 4.13 se realiza la interpolación de los datos para determinar la entalpía de la primera extracción de la turbina para enviarla a una etapa de recalentamiento y se define como $h2$ y la cual tiene un valor de 1187.49 btu/lbm.

Para determinar el valor de $h3$ la interpolación de datos es realizada en las tablas 4.14 y 4.15 donde el resultado obtenido es, $h3 = 1402.98$ btu/lbm y $S3 = 1.81$. Para el caso de la entalpía 4 y 5 ($h4$ y $h5$), la interpolación se realiza en las tablas 4.24 y 4.25, donde para dicha interpolación se utilizan los datos de presión de la extracción. La presión del condensador ($P5$) y la entropía $S3$. Por tanto las entalpías del sistema se resumen en la tabla 4.24.

Primera interpolación con S3.

Tabla 4.24 Interpolación de Entalpías alrededor de la entropía de 1.81 btu/lbm°R

		h₁	h_{1.5}	h₄₀	h₅₀
Entropía original	1.81				
Entropía menor	1.80	1004.0	1029.0	1272.00	1295.00
Entropía mayor	1.82	1017.0	1040.0	1290.00	1315.00
	-0.02	-10	-11	-18	-20
	0.01				
		1009.0	1034.5	1281.00	1305.00

Segunda interpolación con P3.

Tabla 4.25 Interpolación de Entalpías alrededor de las presiones de 44.07 y 1.10 lbf/in² respectivamente

	P4	h4	P5	h5
Presión original	44.07		1.10	
Presión menor	40	1281.00	1.0	1009.00
Presión mayor	50	1305.00	1.5	1034.50
	-10	-18.25	-0.5	-25.50
	4.07		0.10	
		1288.42		1014.10

Entonces $h_4 = 1288.42$ y $h_5 = 1014.10$ btu/lbm. Los valores de las entalpías h_8 , h_7 , h_6 se calculan en la tabla 4.23 ya que los valores de las presiones son iguales por consiguiente $h_{f4} = h_8$ y $h_{f5} = h_6 = h_7$, la razón por la que se realiza la igualdad es debido que a la salida del condensador como del calentador es líquido saturado. En la tabla 4.26 se resumen todos los valores de la entalpía del sistema.

Tabla 4.26 entalpías del ciclo combinado

H1	1374.44 btu/lbm
H2	1187.49 btu/lbm
H3	1402.98 btu/lbm
H4	1288.42 btu/lbm
H5	1014.10 btu/lbm
H6 = h_{f5}	72.12 btu/lbm
H7 = h_{f5}	72.12 btu/lbm
H8 = h_{f4}	241.99 btu/lbm

Por consiguiente se aplican las ecuaciones (4.8) y (4.9):

$$W_{tur} = (1374.4 - 1187.4) + (1402.9 - 1288.4) + \left(1 - \frac{241.99 - 72.12}{1288.42 - 72.12}\right) (1288. - 1014.10) =$$

$$=$$

$$W_{tur} = 537.69 \text{ btu/lbm}$$

$$e = \frac{(1374.4 - 1187.4) + (1402.9 - 1288.4) + \left(1 - \frac{241.99 - 72.12}{1288.42 - 72.12}\right) (1288. - 1014.10)}{(1374.4 - 241.99) + (1402.9 - 1187.49)} =$$

$$e = 0.398 = 39.80\%$$

4.6.1 Usando software de simulación

Los datos de los ejemplos anteriores son nuevamente aplicados al programa para simular un ciclo Rankine utilizando una etapa de regeneración y una de recalentamiento. Los datos aplicados a la simulación se describen en la figura 4.19.

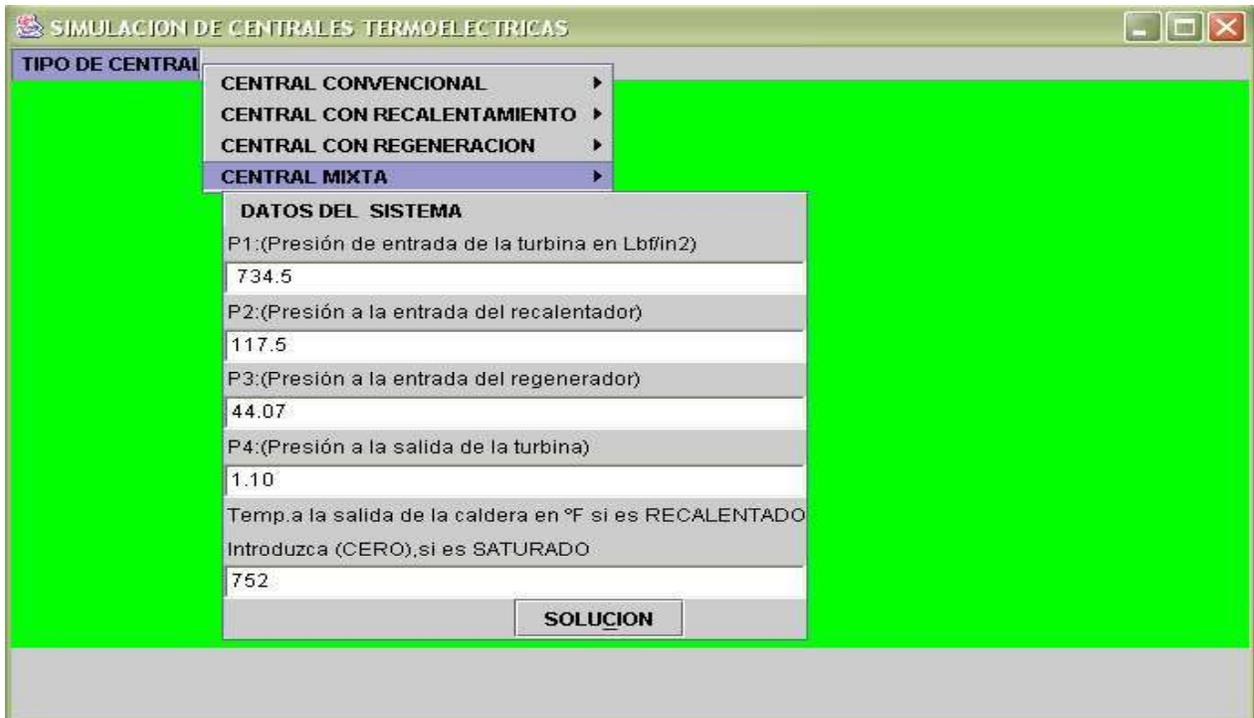


Figura 4.19 Datos de entrada al simulador

El simulador entrega directamente los resultados del problema los cuales se dan en la figura 4.20.

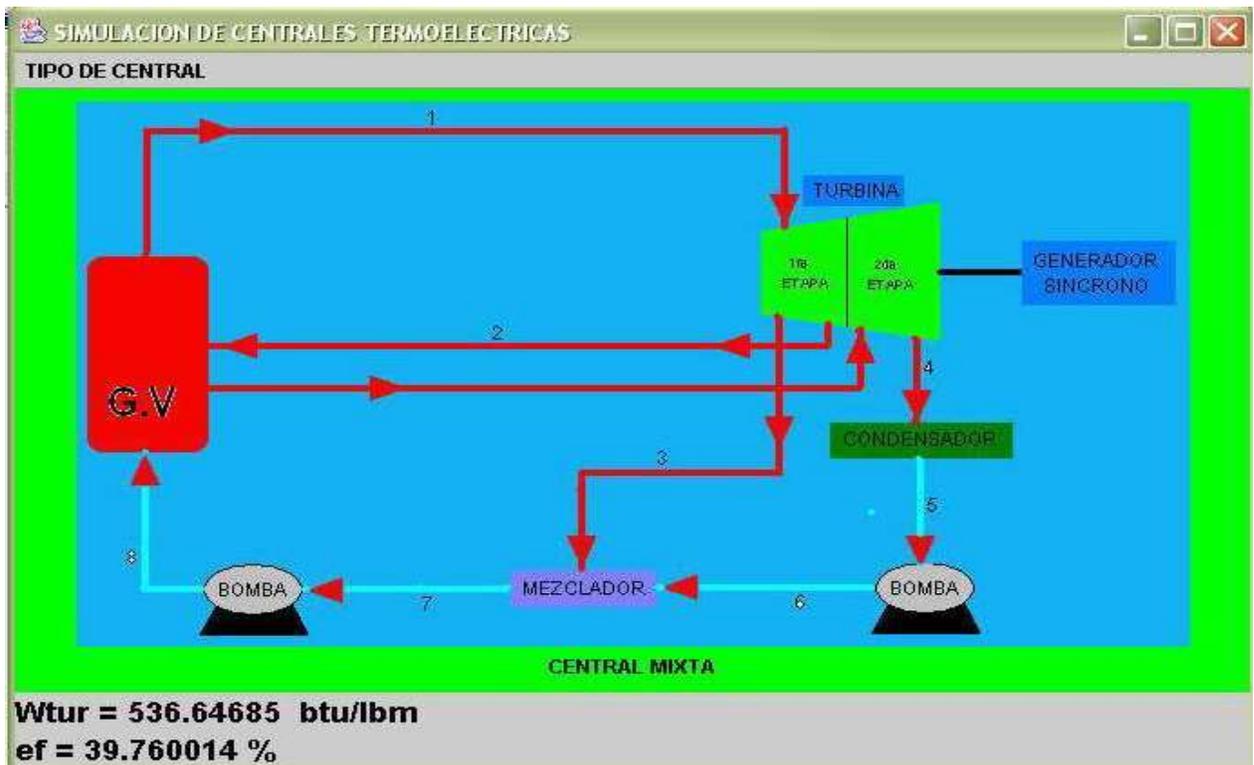


Figura 4.20 Resultados del simulador

4.7 Comparación de resultados

En la tabla 4.27 se realiza una comparación de los resultados de cada una de las modificaciones del ciclo Rankine.

Tabla 4.27 Comparación de ciclos

CICLO	TIPO DE OPERACION	EFICIENCIA MEDIANTE	
		Diagrama de Mollier	Software de Simulación
CONVENCIONAL	Vapor saturado y seco	33.25%	32.73%
CONVENCIONAL	Vapor sobrecalentado	37.30%	37.03 %
UNA ETAPA DE RECALENTAMIENTO	Vapor Sobrecalentado	37.93%	37.86 %
UNA ETAPA DE REGENERACION	Vapor sobrecalentado	39.60 %	39.36 %
MIXTO	Vapor sobrecalentado	39.80%	

En la tabla anterior se muestra como el uso del software de simulación tiene unos resultados muy próximos a los obtenidos manualmente usando el diagrama de Mollier. Por otro lado, existe una diferencia muy notable en el tiempo en que se muestran los resultados ambos sistemas de solución.

CAPITULO 5

Conclusiones y recomendaciones

5.1 Conclusiones

Hoy en día el tiempo es un factor limitante en cualquier tipo de actividad que realiza el ser humano y a consecuencia de ello se producen grandes pérdidas económicas que perjudican seriamente el crecimiento de una organización. Debido a esto surge la necesidad de crear sistemas de cómputo a partir de información conocida del problema a resolver.

El sistema que se presenta en esta tesis se enfoca a la solución de centrales termoeléctricas y sus diferentes modificaciones, y se basa en un diagrama que muestra el comportamiento de los diferentes estados físicos del agua a diferentes presiones y temperaturas donde la cantidad de información es suficiente para tratar el problema. Para capturar la información de este diagrama a la computadora es un proceso complejo ya que es necesario separar la información correspondiente de cada región del diagrama de Mollier y almacenarla en matrices numéricas de acuerdo a su presión y temperatura, presión y humedad o en la zona de saturación dependiendo de la región de que se trate. Una vez organizada la información que se ha obtenido del diagrama se necesita crear un programa que sea capaz de comparar los datos que fueron aplicados por el usuario con los datos obtenidos del diagrama, posteriormente si los datos del usuarios son iguales a los obtenidos del diagrama el algoritmo deberá realizar las operaciones que correspondan al tipo de central que le esta indicando el usuario. Finalmente, si los datos introducidos por el usuario no son iguales a los obtenidos del diagrama el programa deberá hacer una comparación de los datos que rodean al original y realizar una interpolación para que determine un valor y después realice las operaciones que correspondan al tipo de central que le indique el usuario.

Para la elaboración del programa es necesario tomar en cuenta las diferentes modificaciones y sus diferentes zonas de operación de las centrales termoeléctricas, su análisis es la base de la programación estructurada, es decir, la descomposición de una tarea en tareas más pequeñas. El programa no es otra cosa que una colección de funciones que son llamadas sucesivamente por la función principal y única del

programa. Cada función tiene sus propias variables que necesita para ejecutar las operaciones que le han sido programadas y almacenar los resultados de estas, es decir, una función es un modulo independiente que se utiliza cada vez que los datos de entrada lo determinen.

5.2 Recomendaciones

Para hacer un software de simulación como el que se trata en esta tesis es necesario hacer un estudio total al problema que se desea resolver por medio de la computadora. Es decir, necesitamos conocer el comportamiento del sistema con distintos parámetros de operación, conocer el funcionamiento y el comportamiento de cada elemento que componen al sistema.

Una vez conocida la problemática a resolver hay que buscar una entre varias alternativas de lenguajes de programación ya que de las características de este dependerá la forma como sea implementado el sistema para llegar al resultado esperado y nos proporcione la manera más fácil para su uso.

Para mejorar este programa se sugiere implementar un código que realice varios tipos de conversión de unidades. Además de otro código que muestre gráficamente los componentes de una central termoeléctrica y con ellos simular cualquier configuración de las centrales.

Finalmente para que el usuario pueda utilizar este software es necesario que conozca lo más elemental de una central termoeléctrica ya que sin ello no podría interpretar los resultados que proporciona el programa

Bibliografía

- [1] Programación en java
Francisco Javier Cevallos
Segunda edición
- [2] Principios de Termodinámica
Francisco Javier Cevallos
Cuarta edición
- [3] Interfaces gráficas y aplicaciones para Internet
Francisco Javier Cevallos
Segunda edición
- [4] La Biblia de java 2
Holzner steven - Anaya
Segunda edición
- [5] [http:// ocwus.us.es/arquitectura-e-ingenieria](http://ocwus.us.es/arquitectura-e-ingenieria)
- [6] [http:// www.cfe.gob.mx/es/LaEmpresa/generaciónelectricidad/Termo](http://www.cfe.gob.mx/es/LaEmpresa/generaciónelectricidad/Termo)
- [7] [http:// www.planetaazul.com.mx](http://www.planetaazul.com.mx)
- [8] [http:// html.rincondelvago.com/energia](http://html.rincondelvago.com/energia)
- [9] [http:// www.quiminet.com.mx/nt5](http://www.quiminet.com.mx/nt5)
- [10] [http:// durangomexico.espacioblog.com/post/2008/10/28/una-de-las-termoelectricas-mas-grandes-mexico](http://durangomexico.espacioblog.com/post/2008/10/28/una-de-las-termoelectricas-mas-grandes-mexico)
- [11] [http:// www.cideiber.com/infopaises/Mexico/Mexico-05-04.html](http://www.cideiber.com/infopaises/Mexico/Mexico-05-04.html)

APÉNDICE A

Programa del simulador

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

class prueba1 extends JFrame{
public static void main(String args[]) {
prueba1 f = new prueba1();
f.setSize(690, 490);
f.setVisible(true); }

protected JMenuBar menuBar;
Label etiq; // una etiqueta que dice "Archivo: "
TextField campo;
TextField campo1; // El campo de texto
TextField campo2;
TextField campo3;
TextField campo4;
TextField campo5;
TextField campo6;
TextField campo7;
TextField campo8;
TextField campo9;
TextField campo10;
TextField campo11;
TextField campo12;
TextField campo13;
TextField campo14;
TextField campo15;
JMenu menu;
JTextField Pantalla1;
JPanel jPnPanel=new JPanel();
JTextField text = new JTextField(20);
double p1=0,p2=0,t=0;
double p11=0,p21=0,p3,t1=0;
double p12=0,p22=0,p32,t12=0;
double p13=0,p23=0,p33,p4=0,t13=0;
JButton boton;
JButton boton1;
JButton boton2;
JButton boton3;
JLabel saludo;
JLabel saludo1;
JLabel jlabel;
```

```

double wu[]=new double [2];
Container contentPane = getContentPane();
public prueba1(){
super("SIMULACION DE CENTRALES TERMOELECTRICAS");
menuBar = new JMenuBar();
menuBar.setLayout(new BorderLayout(menuBar, BorderLayout.Y_AXIS));
getContentPane().add(BorderLayout.NORTH, menuBar);
contentPane.setBackground(Color.green);
jlabel = new JLabel(new ImageIcon("image.jpg"));
primero("TIPO DE CENTRAL          ");
addWindowListener(new WindowAdapter(){
public void windowClosing(WindowEvent e) {
System.exit(0); } });
}
public void primero(String title) {
getContentPane().add(jPnPanel, BorderLayout.SOUTH);
jPnPanel.setLayout(new GridLayout(0,1));
saludo=new javax.swing.JLabel();
saludo.setText(" ");
saludo.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
saludo.setFont(new java.awt.Font("Dialog",1,18));
jPnPanel.add(saludo);
saludo1=new javax.swing.JLabel();
saludo1.setText(" ");
saludo1.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
saludo1.setFont(new java.awt.Font("Dialog",1,18));
jPnPanel.add(saludo1);
saludo1.setBounds(92, 36, 504, 300 );
JMenu m = (JMenu)menuBar.add(new HorizontalMenu(title));
JMenu m1 = (JMenu)m.add(new HorizontalMenu("CENTRAL CONVENCIONAL"));
m1.add("DATOS DEL SISTEMA          ");
etiq= new Label("P1:(Presión de entrada de la turbina en Lbf/in2) ");
campo= new TextField(15);
campo.setText(" ");
m1.add(etiq);
m1.add(campo);
etiq= new Label("P2:(Presión a la salida de la turbina) ");
campo1= new TextField(10);
campo1.setText(" ");
m1.add(etiq);
m1.add(campo1);
etiq= new Label("Temp.a la salida de la caldera en °F si es RECALENTADO");
m1.add(etiq);
etiq= new Label("Introduzca (CERO),si es SATURADO");
campo2= new TextField(10);
campo2.setText(" ");
m1.add(etiq);
m1.add(campo2);

```

```

boton=new javax.swing.JButton();
boton.setText("SOLUCION");
boton.setToolTipText("pulsacion");
boton.setMnemonic('C');
m1.add(boton);
boton.setBounds(40, 40, 24, 10 );
boton.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
intermedio1(evt); } });

JMenu N1 = (JMenu)m.add(new HorizontalMenu("CENTRAL CON
RECALENTAMIENTO"));
N1.add("DATOS DEL SISTEMA");
etiq= new Label("P1:(Presión de entrada de la turbina en Lbf/in2) ");
campo3= new TextField(15);
campo3.setText(" ");
N1.add(etiq);
N1.add(campo3);
etiq= new Label("P2:(Presión a la entrada del recalentamiento) ");
campo4= new TextField(10);
campo4.setText(" ");
N1.add(etiq);
N1.add(campo4);
etiq= new Label("P3:(Presión a la salida de la turbina) ");
campo5= new TextField(10);
campo5.setText(" ");
N1.add(etiq);
N1.add(campo5);
etiq= new Label("Temp.a la salida de la caldera en °F si es RECALENTADO");
N1.add(etiq);
etiq= new Label("Introduzca (CERO),si es SATURADO");
campo6= new TextField(10);
campo6.setText(" ");
N1.add(etiq);
N1.add(campo6);
boton1=new javax.swing.JButton();
boton1.setText("SOLUCION");
boton1.setToolTipText("pulsacion");
boton1.setMnemonic('C');
N1.add(boton1);
boton1.setBounds(40, 40, 24, 10 );
boton1.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
intermedio2(evt); } });

JMenu M1 = (JMenu)m.add(new HorizontalMenu("CENTRAL CON
REGENERACION"));

```

```

M1.add("DATOS DEL SISTEMA");
etiq= new Label("P1:(Presión de entrada de la turbina en Lbf/in2) ");
campo7= new TextField(15);
campo7.setText(" ");
M1.add(etiq);
M1.add(campo7);
etiq= new Label("P2:(Presión a la entrada del regenerador) ");
campo8= new TextField(10);
campo8.setText(" ");
M1.add(etiq);
M1.add(campo8);
etiq= new Label("P3:(Presión a la salida de la turbina) ");
campo9= new TextField(10);
campo9.setText(" ");
M1.add(etiq);
M1.add(campo9);
etiq= new Label("Temp.a la salida de la caldera en °F si es RECALENTADO");
M1.add(etiq);
etiq= new Label("Introduzca (CERO),si es SATURADO");
campo10= new TextField(10);
campo10.setText(" ");
M1.add(etiq);
M1.add(campo10);
boton2=new javax.swing.JButton();
boton2.setText("SOLUCION");
boton2.setToolTipText("pulsacion");
boton2.setMnemonic('C');
M1.add(boton2);
boton2.setBounds(40, 40, 24, 10 );

boton2.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
intermedio3(evt); } });

JMenu W1 = (JMenu)m.add(new HorizontalMenu("CENTRAL MIXTA"));
W1.add("DATOS DEL SISTEMA");
etiq= new Label("P1:(Presión de entrada de la turbina en Lbf/in2) ");
campo11= new TextField(15);
campo11.setText(" ");
W1.add(etiq);
W1.add(campo11);
etiq= new Label("P2:(Presión a la entrada del recalentador) ");
campo12= new TextField(10);
campo12.setText(" ");
W1.add(etiq);
W1.add(campo12);

```

```

etiq= new Label("P3:(Presión a la entrada del regenerador) ");
campo13= new TextField(10);
campo13.setText(" ");
W1.add(etiq);
W1.add(campo13);
etiq= new Label("P4:(Presión a la salida de la turbina) ");
campo14= new TextField(10);
campo14.setText(" ");
W1.add(etiq);
W1.add(campo14);
etiq= new Label("Temp.a la salida de la caldera en °F si es RECALENTADO");
W1.add(etiq);
etiq= new Label("Introduzca (CERO),si es SATURADO");
campo15= new TextField(10);
campo15.setText(" ");
W1.add(etiq);
W1.add(campo15);
boton3=new javax.swing.JButton();
boton3.setText("SOLUCION");
boton3.setToolTipText("pulsacion");
boton3.setMnemonic('C');
W1.add(boton3);
boton3.setBounds(40, 40, 24, 10 );
boton3.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
intermedio(evt); } });
}
class HorizontalMenu extends JMenu {
HorizontalMenu(String label) {
super(label);
JPopupMenu pm = getPopupMenu();
pm.setLayout(new BorderLayout(pm, BorderLayout.Y_AXIS)); }
public void setPopupMenuVisible(boolean b) {
boolean isVisible = isPopupMenuVisible();
if (b != isVisible) {
if ((b==true) && isShowing()){
// Set location of popupMenu (pulldown or pullright)
// Perhaps this should be dictated by L&F
int x =20;
int y = 0;
Container parent = getParent();
Dimension s = getSize();
if (parent instanceof JPopupMenu) {
x = 10;
y = s.height; }

```

```

else
{
x = s.width;
y = 10; }
getPopupMenu().show(this, x, y);
}
else {
getPopupMenu().setVisible(false);}
}}}
public void intermedio1(java.awt.event.ActionEvent evt) {
p1 = Double.parseDouble(campo.getText());
p2 = Double.parseDouble(campo1.getText());
t = Double.parseDouble(campo2.getText());
double k=0;
String tipo="CONVENCIONAL";
wu=MOLLIER.operaciones_mixtas(p1,p2,k,k,t,tipo);
saludo.setText(("Wtur =")+Float.toString((float)wu[0])+" btu/lbm");
saludo1.setText(("ef =")+Float.toString((float)wu[1])+" %");
saludo.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
jlabel.setVisible(false);
jlabel = new JLabel("CENTRAL CONVENCIONAL", new ImageIcon("convencional.jpg"),
JLabel.CENTER);
jlabel.setVerticalTextPosition(JLabel.BOTTOM);
jlabel.setHorizontalTextPosition(JLabel.CENTER);
jlabel.setBorder(BorderFactory.createEtchedBorder());
contentPane.add(jlabel); }
public void intermedio2(java.awt.event.ActionEvent evt) {
p11 = Double.parseDouble(campo3.getText());
p21 = Double.parseDouble(campo4.getText());
p3 = Double.parseDouble(campo5.getText());
t1 = Double.parseDouble(campo6.getText());
double k=0;
String tipo="RECALENTAMIENTO";
wu=MOLLIER.operaciones_mixtas(p11,p21,p3,k,t1,tipo);
saludo.setText(("Wtur =")+Float.toString((float)wu[0])+" btu/lbm");
saludo1.setText(("ef =")+Float.toString((float)wu[1])+" %");
jlabel.setVisible(false);
jlabel = new JLabel("CENTRAL CON RECALENTAMIENTO", new
ImageIcon("recalentamiento.jpg"), JLabel.CENTER);
jlabel.setVerticalTextPosition(JLabel.BOTTOM);
jlabel.setHorizontalTextPosition(JLabel.CENTER);
jlabel.setBorder(BorderFactory.createEtchedBorder());
contentPane.add(jlabel); }

```

```

public void intermedio3(java.awt.event.ActionEvent evt) {
    p12 = Double.parseDouble(campo7.getText());
    p22 = Double.parseDouble(campo8.getText());
    p32 = Double.parseDouble(campo9.getText());
    t12 = Double.parseDouble(campo10.getText());
    double k=0;
    String tipo="REGENERATIVO";
    wu=MOLLIER.operaciones_mixtas(p12,p22,p32,k,t12,tipo);
    saludo.setText(("Wtur =")+Float.toString((float)wu[0])+" btu/lbm");
    saludo1.setText(("ef =")+Float.toString((float)wu[1])+" %");
    jlabel.setVisible(false);
    jlabel = new JLabel("CENTRAL CON REGENERACION", new
    ImagemIcon("regeneracion.jpg"), JLabel.CENTER);
    jlabel.setVerticalTextPosition(JLabel.BOTTOM);
    jlabel.setHorizontalTextPosition(JLabel.CENTER);
    contentPane.add(jlabel); }
public void intermedio(java.awt.event.ActionEvent evt) {
    p13 = Double.parseDouble(campo11.getText());
    p23 = Double.parseDouble(campo12.getText());
    p33 = Double.parseDouble(campo13.getText());
    p4 = Double.parseDouble(campo14.getText());
    t13 = Double.parseDouble(campo15.getText());
    String tipo="MIXTO";
    wu=MOLLIER.operaciones_mixtas(p13,p23,p33,p4,t13,tipo);
    saludo.setText(("Wtur =")+Float.toString((float)wu[0])+" btu/lbm");
    saludo1.setText(("ef =")+Float.toString((float)wu[1])+" %");
    jlabel.setVisible(false);
    jlabel = new JLabel("CENTRAL MIXTA", new ImagemIcon("MIXTA.jpg"),
    Label.CENTER);
    jlabel.setVerticalTextPosition(JLabel.BOTTOM);
    jlabel.setHorizontalTextPosition(JLabel.CENTER);
    contentPane.add(jlabel); }
}

class MOLLIER {
public static double[] operaciones_mixtas(double a,double b, double c, double d,double
t,String TIPO) {
    String TS;
    double wy[]=new double[2];
    double P1=a;
    double P2=b;
    double P3=c;
    double P4=d;
    double T=t;
    if(t==0){
    TS="SATURADO";}
    else{
    TS="SOBRECALENTADO";}
}
}

```

```

double INTEV[]=new double [2];
double INTEV2[]=new double [2];
double INTEH[]=new double [2];
double INTEH2[]=new double [2];
double INTEH3[]=new double [2];
double HF[]=new double [1];
double HF2[]=new double [1];
double HU[]=new double [12];
double HU2[]=new double [12];
double HU3[]=new double [12];
if(TS=="SATURADO") {
double SAT1[]=saturado(P1);
HU=humedad(P2,SAT1[1]);
HF=saturado1(P2);
INTEH=INTERPOLACIONDOBLE(P2,SAT1[1],HU[0],HU[1],HU[3],HU[2],HU[5],HU[4],
HU[7],HU[10],HU[9],HU[8],HU[11],HU[6]);
wy=TIPOS(TIPO,SAT1[0],SAT1[1],INTEH[0],HF[1],HF[0],INTEH[1],HF[1]);}
if(TS=="SOBRECALENTADO"){
int P[]=presion(P1);
int T1[]=temperatura(T);
double H1=0;
double H2=0;
double S1=0;
double S2=0;
double h1=0,h2=0,s1=0,s2=0;
/AMBAS CORRECTAS
if(P[1]==-1 && T1[1]==-1){
H1 = entalpia(P[0],T1[0]);
S1 = ENTROPIA(P[0],T1[0]);
INTEV[0]=H1;
INTEV[1]=S1; }
//PRESION CORRECTA
if(P[1]==-1 && T1[1]!=-1){
H1 = entalpia(P[0],T1[0]);
H2 = entalpia(P[0],T1[1]);
S1 = ENTROPIA(P[0],T1[0]);
S2 = ENTROPIA(P[0],T1[1]);
INTEV=INTERPOLACION(T,T1[2],T1[3],H1,H2,S1,S2);}
//TEMPERATURA CORRECTA
if(T1[1]==-1 && P[1]!=-1){
H1 = entalpia(P[0],T1[0]);
H2 = entalpia(P[1],T1[0]);
S1 = ENTROPIA(P[0],T1[0]);
S2 = ENTROPIA(P[1],T1[0]);
INTEV=INTERPOLACION(P1,P[2],P[3],H1,H2,S1,S2);}

```

```

//AMBAS INCORRECTAS
// double h1=0,h2=0,s1=0,s2=0;
if(P[1]!=-1 && T1[1]!=-1){
H1 = entalpia(P[0],T1[0]);
h1 = entalpia(P[0],T1[1]);
S1 = ENTROPIA(P[0],T1[0]);
s1 = ENTROPIA(P[0],T1[1]);
H2 = entalpia(P[1],T1[0]);
h2 = entalpia(P[1],T1[1]);
S2 = ENTROPIA(P[1],T1[0]);
s2 = ENTROPIA(P[1],T1[1]);
INTEV=INTERPOLACIONDOBLE(P1,T,P[2],P[3],T1[2],T1[3],H1,h1,S1,s1,H2,h2,S2,s2);
if(TIPO=="CONVENCIONAL"){
HU=humedad(P2,INTEV[1]);
INTEH=INTERPOLACIONDOBLE(P2,INTEV[1],HU[0],HU[1],HU[3],HU[2],HU[5],HU[4],
HU[7],HU[10],HU[9],HU[8],HU[11],HU[6]);
HF=saturado1(P2);
INTEH2=INTERPOLACIONDOBLE(P3,INTEV2[1],HU[0],HU[1],HU[3],HU[2],HU[5],
HU[4],HU[7],HU[10],HU[9],HU[8],HU[11],HU[6]);
wy= TIPOS(TIPO,INTEV[0],INTEV[1],INTEH[0],INTEV2[0],HF[0],INTEH2[0],HF[1]); }
}
if(TIPO=="RECALENTAMIENTO" || TIPO=="MIXTO"){
P=presion(P2);
//AMBAS CORRECTAS
if(P[1]==-1 && T1[1]==-1){
H1 = entalpia(P[0],T1[0]);
S1 = ENTROPIA(P[0],T1[0]);
INTEV2[0]=H1;
INTEV2[1]=S1; }
//PRESION CORRECTA
if(P[1]==-1 && T1[1]!=-1){
H1 = entalpia(P[0],T1[0]);
H2 = entalpia(P[0],T1[1]);
S1 = ENTROPIA(P[0],T1[0]);
S2 = ENTROPIA(P[0],T1[1]);
INTEV2=INTERPOLACION(T,T1[2],T1[3],H1,H2,S1,S2);}
//TEMPERATURA CORRECTA
if(T1[1]==-1 && P[1]!=-1){
H1 = entalpia(P[0],T1[0]);
H2 = entalpia(P[1],T1[0]);
S1 = ENTROPIA(P[0],T1[0]);
S2 = ENTROPIA(P[1],T1[0]);
INTEV2=INTERPOLACION(P1,P[2],P[3],H1,H2,S1,S2);}

```

```

//AMBAS INCORRECTAS
if(P[1]!=-1 && T1[1]!=-1){
H1 = entalpia(P[0],T1[0]);
h1 = entalpia(P[0],T1[1]);
S1 = ENTROPIA(P[0],T1[0]);
s1 = ENTROPIA(P[0],T1[1]);
H2 = entalpia(P[1],T1[0]);
h2 = entalpia(P[1],T1[1]);
S2 = ENTROPIA(P[1],T1[0]);
s2 = ENTROPIA(P[1],T1[1]);
INTEV2=INTERPOLACIONDOBLE(P2,T,P[2],P[3],T1[2],T1[3],H1,h1,S1,s1,H2,h2,S2,
s2);}
if(TIPO=="RECALENTAMIENTO"){
HU=humedad(P2,INTEV[1]);
INTEH=INTERPOLACIONDOBLE(P2,INTEV[1],HU[0],HU[1],HU[3],HU[2],HU[5],HU[4],
HU[7],HU[10],HU[9],HU[8],HU[11],HU[6]);
HU=humedad(P3,INTEV2[1]);
HF=saturado1(P3);
INTEH2=INTERPOLACIONDOBLE(P3,INTEV2[1],HU[0],HU[1],HU[3],HU[2],HU[5],
HU[4],HU[7],HU[10],HU[9],HU[8],HU[11],HU[6]);
wy= TIPOS(TIPO,INTEV[0],INTEV[1],INTEH[0],INTEV2[0],INTEH2[0],HF[0],HF[1]);}
}}
if(TIPO=="REGENERATIVO"){
HU=humedad(P2,INTEV[1]);
INTEH=INTERPOLACIONDOBLE(P2,INTEV[1],HU[0],HU[1],HU[3],HU[2],HU[5],HU[4],
HU[7],HU[10],HU[9],HU[8],HU[11],HU[6]);
HU2=humedad(P3,INTEV[1]);
INTEH2=INTERPOLACIONDOBLE(P3,INTEV[1],HU2[0],HU2[1],HU2[3],HU2[2],HU2[5],
HU2[4],HU2[7],HU2[10],HU2[9],HU2[8],HU2[11],HU2[6]);
HF=saturado1(P3);
HF2=saturado1(P2);
wy=TIPOS(TIPO,INTEV[0],INTEV[1],INTEH[0],INTEH2[0],INTEV2[0],HF[0],HF2[0]); }
if(TIPO=="MIXTO"){
HU=humedad(P2,INTEV[1]);
INTEH=INTERPOLACIONDOBLE(P2,INTEV[1],HU[0],HU[1],HU[3],HU[2],HU[5],HU[4],
HU[7],HU[10],HU[9],HU[8],HU[11],HU[6]);
HU2=humedad(P3,INTEV2[1]);
INTEH2=INTERPOLACIONDOBLE(P3,INTEV2[1],HU2[0],HU2[1],HU2[3],HU2[2],
HU2[5],HU2[4],HU2[7],HU2[10],HU2[9],HU2[8],HU2[11],HU2[6]);
HU3=humedad(P4,INTEV2[1]);
INTEH3=INTERPOLACIONDOBLE(P4,INTEV2[1],HU3[0],HU3[1],HU3[3],HU3[2],
HU3[5],HU3[4],HU3[7],HU3[10],HU3[9],HU3[8],HU3[11],HU3[6]);
HF=saturado1(P3);
HF2=saturado1(P4);
wy= TIPOS1(TIPO,INTEV[0],INTEV[1],INTEH[0],INTEV2[0],INTEH2[0],INTEH3[0],HF[0],
HF2[0]); }
return(wy);
}

```

```

public static int[] presion(double P) {
double x=0,y=0,Z=0,W=0;
int a[]=new int [4];
double p[]={0,1,1.5,2,3,4,5,6,8,10,20,30,40,50,60,80,100,150,200,250,300,400,500,
600,800,1000,1200,1600,2000,3000,4000,5500};
for(int i=0;i<=p.length-1;i++){
if(P==p[i]) {
x=(int)i;
y=(int)-1; }
if((P>p[i])&&(P<p[i+1])){
x=(int)i;
y=(int)i+1;
Z=(int)p[i];
W=(int)p[i+1];}
a[0]=(int)x;
a[1]=(int)y;
a[2]=(int)Z;
a[3]=(int)W; }
return(a);}

public static int[] temperatura(double T) {
int x=0,y=0,z=0,w=0;
int a[]=new int [4];
double t[]={100,150,200,250,300,350,400,450,500,550,600,650,700,750,800,850,900,
950,1000,1050,1100};
for(int j=0;j<=t.length-1;j++){
if(T==t[j]){
x=(int)j;
y=(int)-1; }

```

```

if((T>t[j])&&(T<t[j+1])){
x=(int)j;
y=(int)j+1;
z=(int)t[j];
w=(int)t[j+1]; }
a[0]=(int)x;
a[1]=(int)y;
a[2]=(int)z;
a[3]=(int)w; }
return(a);}

public static double[] saturado(double P) {
double R1=0,R2=0,R3=0,R4=0,R5=0,Rh=0,Rs=0,Rt=0;
double a[]=new double [3];
double p[]={0,1,1.5,2,3,4,5,6,8,10,14.69,20,30,40,50,60,80,100,150,200,250,300,400,
500,600,800,1000,1200,1600,2000,3000};
for(int i=0;i<=p.length-1;i++) {
double H[]={0,1105,1111.1,1116,1122,1127,1131,1134,1139,1143.3,1150.4,1156.3,
1165,1169,1174,1177,1183,1187,1194,1198,1203,1203,1204,1204,1203,1197,1192,
1183,1161,1135,1020};
double S[]={0,1.98,1.95,1.92,1.88,1.86,1.84,1.83,1.805,1.787,1.756,1.73,1.70,1.68,
1.66,1.64,1.62,1.60,1.57,1.545,1.53,1.51,1.484,1.46,1.445,1.415,1.39,1.366,1.324,
1.284,1.161};
double T[]={0,101.7,114,126,141.5,1.53,1.62,170,183,1.93,212,228,250,267.2,281,
292.7,312,327.8,358.4,382,400,417,444.6,467,486,518,544.6,567,604,635,695};
if(P==p[i]){
a[0]=H[i];
a[1]=S[i];
a[2]=T[i]; }

```

```

if(P>p[i] && P<p[i+1]){
R1=p[i]-p[i+1];
R2=P-p[i];
R3=H[i]-H[i+1];
R4=S[i]-S[i+1];
R5=T[i]-T[i+1];
Rh=((R2*R3)/R1)+H[i];
Rs=((R2*R4)/R1)+S[i];
Rt=((R2*R5)/R1)+T[i];
a[0]=Rh;
a[1]=Rs;
a[2]=Rt; } }
return(a); }

public static double[] saturado1(double P) {
double R1=0,R2=0,R5=0,R6=0,Rhf=0,Rhv=0;
double a[]=new double [2];
double p[]={0,1,1.5,2,3,4,5,6,8,10,14.69,15,20,30,40,50,60,80,100,150,200,250,300,
400,500,600,800,1000,1200,1600,2000,3000};
for(int i=0;i<=p.length-1;i++){
double vf[]={0,0.01614,0.01614,0.01623,0.01630,0.01636,0.01640,0.01645,0.01653,
0.01659,0.01672,0.01672};
double hf[]={0,69.7,81.8,94,109.37,120.86,130.13,137.96,150.8,161.1,180,181,196,
218.8,236,250,262,298.4,330.5,355.4,376,393.8,424,449.4,471.6,509.7,542.4,571.7,62
3.7,671.7,802.5};
double vfg[]={0,333.6,253.66,173.73,118.71,90.63,73.52,61.98,47.34,38.42,26.80,26.29};
double hfg[]={0,1036.3,1029.25,1022.2,1013.2,1006.4,1001,996.2,988.5,982.1,
970.3,969.7};

```

```
if(P==p[i]){
a[0]=hf[i]; }
if(P>p[i] && P<p[i+1]) {
R1=p[i]-p[i+1];
R2=P-p[i];
// R5=vf[i]-vf[i+1];
R6=hf[i]-hf[i+1];
Rhf=((R2*R6)/R1)+hf[i];
// Rhv=((R2*R5)/R1)+vf[i];
a[0]=Rhf;
a[1]=Rhv; } }
return(a);}
```

```

public static double entalpia(int P,int T){
double R=0.0;
//PRESION 0 1 1.5 2 3 4 5 6 8 10 20 30 40 50 60 80 100
150 200 250 300 400 500 600 800 1000 1200 1600 2000 3000 4000 5500
double H[][]={{0.00,1108,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,
0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,
0.00}, // para 100°F
{0.00,1129,1129,1129,1127,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,
0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00},
// para 150°F
{0.00,1150,1150,1150,1150,1150,1149,1148,1147,0.00,0.00,0.00,0.00,0.00,0.00,
0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00}, // para
200°F
{0.00,1173,1173,1173,1172,1172,1171,1171,1170,1170,1167,1165,0.00,0.00,0.00,0.00,
0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00}, // para
250°F
{0.00,1197,1197,1197,1200,1196,1195,1195,1194,1193,1190,1190,1187,1185,1180,
0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00},
// para 300°F
{0.00,1220,1220,1219,1219,1218,1218,1218,1218,1216,1215,1215,1212,1210,1208,
1205,1200,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00},
// para 350°F
{0.00,1241,1241,1241,1241,1241,1240,1240,1240,1240,1240,1238,1237,1235,1235,
1230,1228,1220,1210,1203,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00
}, // para 400°F
{0.00,1265,1265,1265,1265,1265,1265,1264,1263,1262,1260,1260,1260,1258,
1255,1252,1247,1240,1233,1229,1205,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00}, // para 450°F
{0.00,0.00,1289,1289,1289,1289,1288,1288,1288,1288,1288,1285,1285,1283,1282,
1280,1280,1274,1270,1265,1258,1245,1234,1216,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00}, // para 500°F

```

{0.00,0.00,1310,1310,1310,1310,1310,1310,1310,1310,1310,1310,1309,1308,1308,1308,
1305,1304,1300,1295,1290,1287,1277,1269,1255,1230,1204,0.00,0.00,0.00,0.00,0.00,
0.00}, // para 550°F

{0.00,0.00,0.00,1335,1335,1335,1335,1335,1335,1334,1333,1332,1331,1331,1331,
1330,1330,1325,1320,1318,1315,1306,1298,1290,1273,1248,1220,1167,0.00,0.00,0.00
,0.00}, // para 600°F

{0.00,0.00,0.00,0.00,1360,1360,1360,1360,1360,1360,1359,1358,1358,1357,1355,1354
,1353,1350,1350,1346,1340,1335,1329,1320,1305,1290,1270,1225,1180,0.00,0.00,
0.00}, // para 650°F

{0.00,0.00,0.00,0.00,1383,1383,1383,1383,1383,1383,1382,1381,1380,1380,1385,1380
,1380,1376,1372,1370,1378,1361,1356,1350,1340,1325,1309,1277,1249,1100,0.00,
0.0 0}, // para 700°F

{0.00,0.00,0.00,0.00,0.00,1408,1408,1408,1408,1408,1408,1407,1407,1406,1405,1404,
1404,1400,1400,1397,1393,1390,1384,1380,1370,1360,1350,1318,1300,1210,0.00,
0.00}, // para 750°F

{0.00,0.00,0.00,0.00,0.00,1432,1432,1431,1431,1432,1431,1431,1430,1430,1430,1430,
1429,1426,1425,1420,1420,1415,1410,1408,1399,1389,1380,1360,1340,1260,1180,
1020}, // para 800°F

{0.00,0.00,0.00,0.00,0.00,0.00,1458,1458,1458,1458,1458,1458,1457,1455,1455,1454,
1453,1451,1450,1448,1445,1442,1440,1437,1427,1419,1410,1390,1375,1320,1260,
1160}, // para 850°F

{0.00,0.00,0.00,0.00,0.00,0.00,0.00,1482,1483,1482,1481,1481,1480,1480,1480,1480,
1480,1478,1475,1475,1472,1470,1465,1460,1454,1450,1440,1425,1410,1360,1320,
1235}, // para 900°F

{0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,1509,1509,1508,1508,1508,1506,1506,1505,
1505,1503,1501,1500,1500,1496,1493,1490,1483,1478,1470,1460,1445,1405,1370,
1300}, // para 950°F

{0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,1533,1533,1532,1532,1531,1531,1531,1530,
1530,1530,1529,1528,1525,1522,1520,1520,1510,1505,1500,1488,1475,1440,1410,
1355}, // para 1000°F

```
{0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,1560,1560,1559,1559,1558,1558,1558,
1558,1555,1553,1553,1551,1550,1548,1545,1540,1533,1529,1518,1505,1480,1450,
1400}, // para 1050°F
{0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,1585,1584,1584,1585,1583,1583,
1582,1581,1580,1580,1579,1576,1573,1570,1567,1560,1558,1550,1535,1510,1485,
1440}, // para 1100°F
{0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,
0.00,0.00,0.00,0.00,0.00,1600,1598,1593,1590,1584,1576,1567,1540,1520,1480},
// para 1150°F
{0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,
0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,
//para 1200°F };
R=(H[T][P]);
return(R);}

public static double ENTROPIA(int P,int T) {
double r=0;
// PRESION 0 1 1.5 2 3 4 5 6 8 10 20 30 40 50 60 80
100 150 200 250 300 400 500 600 800 1000 1200 1600 2000 3000 4000
5500
double S[][]={{0.00,1.980,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,
0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,
0.000,0.000,0.000,0.000,0.000,0.000},//para 100°F
{0.00,2.010,1.970,1.940,1.890,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,
0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,
0.000,0.000,0.000,0.000},//para 150°F
{0.00,2.050,2.010,1.975,1.930,1.900,1.875,1.850,1.820,1.800,0.000,0.000,0.000,
0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,
0.000,0.000,0.000,0.000},//para 200°F
{0.00,2.090,2.040,2.010,1.964,1.935,1.905,1.888,1.850,1.830,1.750,1.700,0.000,
0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,
0.000,0.000,0.000,0.000},//para 250°F
```

{0.00,2.110,2.070,2.040,2.000,1.960,1.935,1.915,1.882,1.860,1.780,1.735,1.695,1.670,1.650,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000},//para 300°F

{0.00,2.150,2.100,2.070,2.020,1.970,1.965,1.945,1.912,1.890,1.810,1.765,1.730,1.705,1.685,1.650,1.620,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000},//para 350°F

{0.00,2.170,2.130,2.095,2.050,2.019,1.995,1.975,1.940,1.920,1.840,1.793,1.760,1.735,1.715,1.680,1.655,1.600,1.560,1.530,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000},//para 400°F

{0.00,2.200,2.150,2.120,2.080,2.047,2.020,2.000,1.970,1.945,1.870,1.820,1.790,1.760,1.740,1.710,1.680,1.635,1.595,1.560,1.540,1.480,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000},//para 450°F

{0.00,0.000,2.180,2.150,2.100,2.070,2.046,2.025,1.995,1.970,1.890,1.850,1.810,1.790,1.770,1.740,1.710,1.660,1.625,1.600,1.570,1.530,1.495,1.460,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000},//para 500°F

{0.00,0.000,2.200,2.170,2.125,2.090,2.070,2.050,2.018,1.990,1.920,1.870,1.840,1.805,1.795,1.760,1.735,1.690,1.650,1.620,1.600,1.560,1.530,1.500,1.445,1.400,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000},//para 550°F

{0.00,0.000,0.000,2.190,2.150,2.118,2.090,2.070,2.040,2.015,1.940,1.890,1.860,1.840,1.820,1.785,1.760,1.710,1.680,1.650,1.630,1.590,1.560,1.535,1.490,1.440,1.400,1.330,0.000,0.000,0.000,0.000,0.000,0.000,0.000},//para 600°F

{0.00,0.000,0.000,0.000,2.170,2.140,2.115,2.095,2.065,2.040,1.960,1.920,1.880,1.860,1.840,1.810,1.780,1.740,1.700,1.680,1.650,1.615,1.585,1.560,1.520,1.480,1.450,1.380,1.325,0.000,0.000,0.000,0.000},//para 650°F

{0.00,0.000,0.000,0.000,2.190,2.160,2.135,2.115,2.085,2.060,1.980,1.940,1.900,1.880,1.860,1.830,1.800,1.757,1.723,1.700,1.680,1.640,1.610,1.590,1.550,1.510,1.480,1.430,1.380,1.230,0.000,0.000},//para 700°F

{0.00,0.000,0.000,0.000,0.000,2.180,2.159,2.135,2.110,2.080,2.000,1.960,1.930,1.900,1.880,1.850,1.825,1.780,1.785,1.720,1.700,1.660,1.635,1.610,1.575,1.540,1.520,1.460,1.430,1.320,0.000,0.000},//para 750°F

{0.00,0.000,0.000,0.000,0.000,2.200,2.175,2.157,2.125,2.100,2.020,1.980,1.945,1.920,1.900,1.870,1.845,1.800,1.770,1.740,1.720,1.680,1.660,1.635,1.600,1.570,1.540,1.500,

```

1.460,1.360,1.280,1.140},//para 800°F
{0.00,0.000,0.000,0.000,0.000,0.000,2.195,2.175,2.145,2.120,2.040,2.000,1.965,1.940,
1.920,1.890,1.865,1.820,1.785,1.760,1.740,1.705,1.680,1.657,1.620,1.590,1.560,1.520,
1.490,1.410,1.340,1.250},//para 850°F
{0.00,0.000,0.000,0.000,0.000,0.000,2.195,2.162,2.140,2.060,2.020,1.980,1.960,
1.940,1.910,1.880,1.840,1.800,1.780,1.760,1.725,1.700,1.680,1.640,1.610,1.585,1.550,
1.515,1.440,1.390,1.300},//para 900°F
{0.00,0.000,0.000,0.000,0.000,0.000,2.180,2.160,2.035,2.035,2.000,1.980,
1.960,1.925,1.900,1.855,1.820,1.800,1.780,1.745,1.720,1.695,1.660,1.635,1.610,1.570,
1.540,1.470,1.420,1.350},//para 950°F
{0.00,0.000,0.000,0.000,0.000,0.000,2.200,2.173,2.100,2.050,2.020,1.995,
1.980,1.940,1.920,1.873,1.840,1.820,1.800,1.760,1.740,1.720,1.680,1.650,1.630,1.595,
1.560,1.500,1.450,1.390},//para 1000°F
{0.00,0.000,0.000,0.000,0.000,0.000,2.190,2.115,2.070,2.040,2.010,
1.995,1.960,1.935,1.890,1.860,1.835,1.815,1.780,1.755,1.735,1.700,1.670,1.650,1.610,
1.580,1.520,1.480,1.420},//para 1050°F
{0.00,0.000,0.000,0.000,0.000,0.000,2.130,2.090,2.055,2.030,
2.010,1.980,1.955,1.910,1.880,1.850,1.830,1.800,1.770,1.750,1.720,1.690,1.670,1.635,
1.600,1.543,1.500,1.440},//para 1100°F
{0.00,0.000,0.000,0.000,0.000,0.000,1.790,1.770,1.735,1.710,1.685,1.650,
1.620,1.560,1.520,1.465},// para 1150°F
{0.00,0.000,0.000,0.000,0.000,0.000,1.640,1.580,1.540,1.490},// para 1200°F };
r=(S[T][P]);
return(r); }

```

```

public static double[] humedad(double P2,double S){
double S1=0,S2=0.0,h1=0,h2=0,hu1=0,hu2=0,E1=0,E2=0,HUM1=0,HUM2=0;

```

```

double r[]=new double [12];
if(P2>=0 && P2<1){
double EN[]={0.00,0.00,0.00};
double H[]={0.00,0.00,0.00};
double HU[]={0.00,0.00,0.00};
double EN1[]={2.2,2.17,2.15,2.11,2.09,2.05,2.01,1.975,1.94,1.90,1.87,1.83,1.80,1.76,
1.72,1.68,1.645,1.61,1.57,1.535,1.50,1.40};
double H1[]={1265,1241,1220,1197,1173,1150,1129,1105,1085,1065,1045,1023,1005,
982,960,940,920,897,878,857,838,815};
double HU1[]={450,400,350,300,250,200,150,1,2,4,6,8,10,12,14,16,18,20,22,24,26,28};
for(int i=0;i<=EN1.length-1;i++){
if(S==0.00){
S1=0;
h1=0;
hu1=0; }
if(S<EN1[i]&&S>EN1[i+1]){
S1=EN1[i];
S2=EN1[i+1];
h1=0;
h2=0;
hu1=0;
hu2=0;}
for(int j=0;j<=EN1.length-1;j++){

```

```

if(S<EN1[j]&&S>EN1[j+1]) {
E1=H1[j];

```

```

E2=H1[j+1];
HUM1=HU1[j];
HUM2=HU1[j+1];}}
r[0]=0; r[1]=1; r[2]=S1; r[3]=S2; r[4]=h1; r[5]=h2; r[6]=HUM1; r[7]=hu2; r[8]=E1; r[9]=E2;
r[10]=hu1;r[11]=HUM2; }}

if(P2>=1 && P2<1.5){
double EN[]={2.2,2.17,2.15,2.11,2.09,2.05,2.01,1.975,1.94,1.90,1.87,1.83,1.80,1.76,
1.72,1.68,1.645,1.61,1.57,1.535,1.50,1.40};
double H[]={1265,1241,1220,1197,1173,1150,1129,1105,1085,1065,1045,1023,1005,
982,960,940,920,897,878,857,838,815};
double HU[]={450,400,350,300,250,200,150,1,2,4,6,8,10,12,14,16,18,20,22,24,26,28};
double EN1[]={2.2,2.18,2.15,2.13,2.10,2.07,2.04,2.01,1.97,1.94,1.91,1.88,1.84,1.80,
1.765,1.73,1.70,1.66,1.62,1.585,1.55,1.515,1.48,1.44,1.41};
double H1[]={1310,1289,1265,1241,1220,1197,1173,1150,1129,1110,1090,1070,1050,
1030,1009,990,969,948,928,906,886,866,845,823,803};
doubleHU1[]={550,500,450,400,350,300,250,200,150,1,2,4,6,8,10,12,14,16,18,20,22,
24,26,28,30};
for(int i=0;i<=EN.length-1;i++){
if(S==EN[i]){
S1=EN[i];
h1=H[i];
hu1=HU[i]; }
if(S<EN[i]&&S>EN[i+1]){
S1=EN[i];
S2=EN[i+1];
h1=H[i];
h2=H[i+1];
hu1=HU[i];
hu2=HU[i+1];}
for(int j=0;j<=EN1.length-1;j++){
if(S<EN1[j]&&S>EN1[j+1]) {

```

```

E1=H1[j];
E2=H1[j+1];
HUM1=HU1[j];
HUM2=HU1[j+1];}}
r[0]=1; r[1]=1.5; r[2]=S1; r[3]=S2; r[4]=h1; r[5]=h2; r[6]=hu1; r[7]=hu2; r[8]=E1; r[9]=E2;
r[10]=HUM1; r[11]=HUM2; }}

```

```

if(P2>=1.5 && P2<2){
double EN[]={2.2,2.18,2.15,2.13,2.10,2.07,2.04,2.01,1.97,1.94,1.91,1.88,1.84,1.80,
1.765,1.73,1.70,1.66,1.62,1.585,1.55,1.515,1.48,1.44,1.41};
double H[]={1310,1289,1265,1241,1220,1197,1173,1150,1129,1110,1090,1070,1050,
1030,1009,990,969,948,928,906,886,866,845,823,803};
double HU[]={550,500,450,400,350,300,250,200,150,1,2,4,6,8,10,12,14,16,18,20,22,
24,26,28,30};
double EN1[]={2.19,2.17,2.15,2.12,2.095,2.07,2.04,2.01,1.975,1.94,1.92,1.885,1.85,
1.815,1.78,1.745,1.71,1.675,1.64,1.605,1.57,1.535,1.50,1.47,1.43,1.40};
double H1[]={1335,1310,1289,1265,1241,1219,1197,1173,1150,1129,1117,1096,1075,
1055,1035,1013,993,973,951,932,910,890,870,850,830,810};
double HU1[]={600,550,500,450,400,350,300,250,200,150,1,2,4,6,8,10,12,14,16,18,20,
22,24,26,28,30};
for(int i=0;i<=EN.length-1;i++){
if(S==EN[i]){
S1=EN[i];
h1=H[i];
hu1=HU[i];}
if(S<EN[i]&&S>EN[i+1]){
S1=EN[i];
S2=EN[i+1];
h1=H[i];
h2=H[i+1];
hu1=HU[i];
hu2=HU[i+1];}

```

```

for(int j=0;j<=EN1.length-1;j++){
if(S<EN1[j]&&S>EN1[j+1]) {
E1=H1[j];
E2=H1[j+1];
HUM1=HU1[j];
HUM2=HU1[j+1];}}
r[0]=1.5; r[1]=2; r[2]=S1; r[3]=S2; r[4]=h1; r[5]=h2; r[6]=hu1; r[7]=hu2; r[8]=E1; r[9]=E2;
r[10]=HUM1; r[11]=HUM2; }}

```

```

if(P2>=2 && P2<3){
double EN[]={2.19,2.17,2.15,2.12,2.095,2.07,2.04,2.01,1.975,1.94,1.92,1.885,1.85,
1.815,1.78,1.745,1.71,1.675,1.64,1.605,1.57,1.535,1.50,1.47,1.43,1.40};
double H[]={1335,1310,1289,1265,1241,1219,1197,1173,1150,1129,1117,1096,1075,
1055,1035,1013,993,973,951,932,910,890,870,850,830,810};
double HU[]={600,550,500,450,400,350,300,250,200,150,1,2,4,6,8,10,12,14,16,18,20,
22,24,26,28,30};
double EN1[]={2.19,2.17,2.15,2.125,2.10,2.08,2.05,2.02,2.0,1.964,1.93,1.885,1.855,
1.82,1.785,1.753,1.72,1.68,1.65,1.62,1.58,1.55,1.515,1.48,1.445,1.415,1.38,1.35};
double H1[]={1383,1360,1335,1310,1289,1265,1241,1219,1200,1172,1150,1122,1102,
1081,1060,1040,1020,1000,980,960,940,920,900,880,860,840,819,800};
double HU1[]={700,650,600,550,500,450,400,350,300,250,200,150,1,2,4,6,8,10,12,14,
16,18,20,22,24,26,28,30,32};
for(int i=0;i<=EN.length-1;i++){
if(S==EN[i]){
S1=EN[i];
h1=H[i];
hu1=HU[i];}
if(S<EN[i]&&S>EN[i+1]){
S1=EN[i];
S2=EN[i+1];
h1=H[i];
h2=H[i+1];

```

```

hu1=HU[i];
hu2=HU[i+1];}
for(int j=0;j<=EN1.length-1;j++){
if(S<EN1[j]&&S>EN1[j+1]) {
E1=H1[j];
E2=H1[j+1];
HUM1=HU1[j];
HUM2=HU1[j+1];}}
r[0]=2; r[1]=3; r[2]=S1; r[3]=S2; r[4]=h1; r[5]=h2; r[6]=hu1; r[7]=hu2; r[8]=E1; r[9]=E2;
r[10]=HUM1; r[11]=HUM2;}}

if(P2>=3 && P2<4){
double EN[]={2.19,2.17,2.15,2.125,2.10,2.08,2.05,2.02,2.0,1.964,1.93,1.885,1.855,
1.82,1.785,1.753,1.72,1.68,1.65,1.62,1.58,1.55,1.515,1.48,1.445,1.415,1.38,1.35};
double H[]={1383,1360,1335,1310,1289,1265,1241,1219,1200,1172,1150,1122,1102,
1081,1060,1040,1020,1000,980,960,940,920,900,880,860,840,819,800};
double HU[]={700,650,600,550,500,450,400,350,300,250,200,150,1,2,4,6,8,10,12,14,
16,18,20,22,24,26,28,30,32};
double EN1[]={2.20,2.18,2.16,2.14,2.118,2.09,2.07,2.047,2.019,1.97,1.96,1.935,1.90,
1.86,1.83,1.80,1.765,1.73,1.70,1.67,1.635,1.60,1.565,1.535,1.50,1.465,1.435,1.40,
1.365,1.335};
double H1[]={1432,1408,1383,1360,1335,1311,1289,1265,1241,1218,1196,1172,1150,
1128,1108,1088,1068,1048,1025,1008,988,967,948,926,905,886,867,848,825,805};
double HU1[]={800,750,700,650,600,550,500,450,400,350,300,250,200,150,1,2,4,6,8,
10,12,14,16,18,20,22,24,26,28,30,32};
for(int i=0;i<=EN.length-1;i++){
if(S==EN[i]){
S1=EN[i];
h1=H[i];
hu1=HU[i]; }
if(S<EN[i]&&S>EN[i+1]){
S1=EN[i];

```

```

S2=EN[i+1];
h1=H[i];
h2=H[i+1];
hu1=HU[i];
hu2=HU[i+1];}
for(int j=0;j<=EN1.length-1;j++){
if(S<EN1[j]&&S>EN1[j+1]) {
E1=H1[j];
E2=H1[j+1];
HUM1=HU1[j];
HUM2=HU1[j+1];}}
r[0]=3; r[1]=4; r[2]=S1; r[3]=S2; r[4]=h1; r[5]=h2; r[6]=hu1; r[7]=hu2; r[8]=E1; r[9]=E2;
r[10]=HUM1; r[11]=HUM2; } }

if(P2>=4 && P2<5) {
double EN[]={2.20,2.18,2.16,2.14,2.118,2.09,2.07,2.047,2.019,1.97,1.96,1.935,1.90,
1.86,1.83,1.80,1.765,1.73,1.70,1.67,1.635,1.60,1.565,1.535,1.50,1.465,1.435,1.40,1.36
5,1.335};
double H[]={1432,1408,1383,1360,1335,1311,1289,1265,1241,1218,1196,1172,1150,
1128,1108,1088,1068,1048,1025,1008,988,967,948,926,905,886,867,848,825,805};
double HU[]={800,750,700,650,600,550,500,450,400,350,300,250,200,150,1,2,4,6,8,10,
12,14,16,18,20,22,24,26,28,30,32};
double EN1[]={2.195,2.175,2.159,2.135,2.115,2.09,2.07,2.046,2.02,1.995,1.965,1.935,
1.905,1.875,1.845,1.81,1.78,1.75,1.72,1.68,1.65,1.62,1.59,1.555,1.52,1.49,1.46,1.425,
1.39,1.36,1.33};
double H1[]={1458,1432,1408,1383,1360,1335,1310,1288,1265,1240,1218,1195,1171,
1150,1130,1110,1090,1070,1050,1030,1010,990,970,950,930,910,890,870,850,830,
810};
double HU1[]={850,800,750,700,650,600,550,500,450,400,350,300,250,200,150,1,2,4,
6,8,10,12,14,16,18,20,22,24,26,28,30,32};
for(int i=0;i<=EN.length-1;i++){

```

```

if(S==EN[i]){
S1=EN[i];
h1=H[i];
hu1=HU[i]; }
if(S<EN[i]&&S>EN[i+1]){
S1=EN[i];
S2=EN[i+1];
h1=H[i];
h2=H[i+1];
hu1=HU[i];
hu2=HU[i+1];}
for(int j=0;j<=EN1.length-1;j++){
if(S<EN1[j]&&S>EN1[j+1]) {
E1=H1[j];
E2=H1[j+1];
HUM1=HU1[j];
HUM2=HU1[j+1];}}
r[0]=4; r[1]=5; r[2]=S1; r[3]=S2; r[4]=h1; r[5]=h2; r[6]=hu1; r[7]=hu2; r[8]=E1; r[9]=E2;
r[10]=HUM1; r[11]=HUM2 }}

if(P2>=5 && P2<6) {
double EN[]={2.195,2.175,2.159,2.135,2.115,2.09,2.07,2.046,2.02,1.995,1.965,1.935,
1.905,1.875,1.845,1.81,1.78,1.75,1.72,1.68,1.65,1.62,1.59,1.555,1.52,1.49,1.46,1.425,
1.39,1.36,1.33};
double H[]={1458,1432,1408,1383,1360,1335,1310,1288,1265,1240,1218,1195,1171,
1150,1130,1110,1090,1070,1050,1030,1010,990,970,950,930,910,890,870,850,830,
810};
double HU[]={850,800,750,700,650,600,550,500,450,400,350,300,250,200,150,1,2,4,6,
8,10,12,14,16,18,20,22,24,26,28,30,32};
double EN1[]={2.195,2.175,2.157,2.135,2.115,2.095,2.07,2.05,2.025,2.00,1.975,1.945,
1.915,1.888,1.85,1.83,1.80,1.77,1.738,1.70,1.67,1.64,1.61,1.58,1.545,1.51,1.48,1.45,
1.42,1.39,1.355,1.32,1.29};

```

```

double H1[]={1482,1458,1431,1408,1383,1360,1335,1310,1288,1265,1240,1218,
1195,1171,1149,1135,1115,1094,1075,1055,1035,1015,995,975,955,935,915,895,876,
855,835,815,800};
double HU1[]={900,850,800,750,700,650,600,550,500,450,400,350,300,250,200,150,
1,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34};
for(int i=0;i<=EN.length-1;i++){
if(S==EN[i]){
S1=EN[i];
h1=H[i];
hu1=HU[i]; }
if(S<EN[i]&&S>EN[i+1]){
S1=EN[i];
S2=EN[i+1];
h1=H[i];
h2=H[i+1];
hu1=HU[i];
hu2=HU[i+1];}
for(int j=0;j<=EN1.length-1;j++){
if(S<EN1[j]&&S>EN1[j+1]) {
E1=H1[j];
E2=H1[j+1];
HUM1=HU1[j];
HUM2=HU1[j+1];}}
r[0]=5; r[1]=6; r[2]=S1; r[3]=S2; r[4]=h1; r[5]=h2; r[6]=hu1; r[7]=hu2; r[8]=E1;
r[9]=E2;r[10]=HUM1; r[11]=HUM2; } }

if(P2>=6 && P2<8) {
double EN[]={2.195,2.175,2.157,2.135,2.115,2.095,2.07,2.05,2.025,2.00,1.975,1.945,
1.915,1.888,1.85,1.83,1.80,1.77,1.738,1.70,1.67,1.64,1.61,1.58,1.545,1.51,1.48,1.45,1.
42,1.39,1.355,1.32,1.29};

```

```

double H[]={1482,1458,1431,1408,1383,1360,1335,1310,1288,1265,1240,1218,1195,
1171,1149,1135,1115,1094,1075,1055,1035,1015,995,975,955,935,915,895,876,855,
835,815,800};
double HU[]={900,850,800,750,700,650,600,550,500,450,400,350,300,250,200,150,
1,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34};
double EN1[]={2.2,2.18,2.162,2.145,2.125,2.11,2.085,2.065,2.04,2.018,1.995,1.97,1.94,
1.912,1.882,1.85,1.82,1.805,1.78,1.75,1.72,1.68,1.652,1.62,1.59,1.56,1.53,1.50,1.47,
1.44,1.405,1.375,1.34,1.31,1.28};
double H1[]={1533,1509,1483,1458,1431,1408,1383,1360,1335,1310,1288,1264,1240,
1218,1194,1170,1148,1140,1120,1100,1080,1060,1040,1020,1000,980,960,940,920,
900,882,863,842,823,803};
double HU1[]={1000,950,900,850,800,750,700,650,600,550,500,450,400,350,300,250,
200,1,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34};
for(int i=0;i<=EN.length-1;i++){
if(S==EN[i]){
S1=EN[i];
h1=H[i];
hu1=HU[i]; }
if(S<EN[i]&&S>EN[i+1]){
S1=EN[i];
S2=EN[i+1];
h1=H[i];
h2=H[i+1];
hu1=HU[i];
hu2=HU[i+1];}
for(int j=0;j<=EN1.length-1;j++){
if(S<EN1[j]&&S>EN1[j+1]) {
E1=H1[j];
E2=H1[j+1];
HUM1=HU1[j];
HUM2=HU1[j+1];}}

```

```
r[0]=6; r[1]=8; r[2]=S1; r[3]=S2; r[4]=h1; r[5]=h2; r[6]=hu1; r[7]=hu2; r[8]=E1; r[9]=E2;
r[10]=HUM1; r[11]=HUM2; }}
```

```
if(P2>=8 && P2<10) {
```

```
double EN[]={2.2,2.18,2.162,2.145,2.125,2.11,2.085,2.065,2.04,2.018,1.995,1.97,1.94,
1.912,1.882,1.85,1.82,1.805,1.78,1.75,1.72,1.68,1.652,1.62,1.59,1.56,1.53,1.50,1.47,
1.44,1.405,1.375,1.34,1.31,1.28};
```

```
double H[]={1533,1509,1483,1458,1431,1408,1383,1360,1335,1310,1288,1264,1240,
1218,1194,1170,1148,1140,1120,1100,1080,1060,1040,1020,1000,980,960,940,920,
900,882,863,842,823,803};
```

```
double HU[]={1000,950,900,850,800,750,700,650,600,550,500,450,400,350,300,250,
200,1,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34};
```

```
double EN1[]={2.19,2.173,2.16,2.14,2.12,2.10,2.08,2.06,2.04,2.015,1.99,1.97,1.945,
1.92,1.89,1.86,1.83,1.80,1.79,1.76,1.73,1.70,1.67,1.64,1.61,1.58,1.55,1.52,1.49,1.46,
1.43,1.40,1.365,1.335,1.305,1.275};
```

```
double H1[]={1560,1533,1509,1482,1458,1432,1408,1383,1360,1334,1310,1288,1263,
1240,1216,1193,1170,1147,1144,1124,1104,1084,1064,1044,1025,1005,986,967,950,
930,910,890,870,850,830,810};
```

```
double HU1[]={1050,1000,950,900,850,800,750,700,650,600,550,500,450,400,350,
300,250,200,1,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34};
```

```
for(int i=0;i<=EN.length-1;i++){
```

```
if(S==EN[i]){
```

```
S1=EN[i];
```

```
h1=H[i];
```

```
hu1=HU[i];}
```

```
if(S<EN[i]&&S>EN[i+1]){
```

```
S1=EN[i];
```

```
S2=EN[i+1];
```

```
h1=H[i];
```

```
h2=H[i+1];
```

```
hu1=HU[i];
```

```
hu2=HU[i+1];}
```

```

for(int j=0;j<=EN1.length-1;j++){
if(S<EN1[j]&&S>EN1[j+1]) {
E1=H1[j];
E2=H1[j+1];
HUM1=HU1[j];
HUM2=HU1[j+1];}}
r[0]=8; r[1]=10; r[2]=S1; r[3]=S2; r[4]=h1;r[5]=h2; r[6]=hu1; r[7]=hu2; r[8]=E1; r[9]=E2;
r[10]=HUM1; r[11]=HUM2; }}

if(P2>=10 && P2<20){
double EN[]={2.19,2.173,2.16,2.14,2.12,2.10,2.08,2.06,2.04,2.015,1.99,1.97,1.945,
1.92,1.89,1.86,1.83,1.80,1.79,1.76,1.73,1.70,1.67,1.64,1.61,1.58,1.55,1.52,1.49,1.46,
1.43,1.40,1.365,1.335,1.305,1.275};
double H[]={1560,1533,1509,1482,1458,1432,1408,1383,1360,1334,1310,1288,1263,
1240,1216,1193,1170,1147,1144,1124,1104,1084,1064,1044,1025,1005,986,967,950,
930,910,890,870,850,830,810};
double HU[]={1050,1000,950,900,850,800,750,700,650,600,550,500,450,400,350,300,
250,200,1,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34};
double EN1[]={2.13,2.115,2.10,2.035,2.06,2.04,2.02,2.0,1.98,1.96,1.94,1.92,1.89,1.87,
1.84,1.81,1.78,1.75,1.73,1.70,1.68,1.65,1.62,1.59,1.56,1.538,1.51,1.48,1.45,1.42,1.40,
1.37,1.34,1.31,1.28,1.26,1.23};
double H1[]={1585,1560,1532,1508,1481,1458,1431,1408,1382,1359,1333,1310,1288,
1262,1240,1215,1190,1167,1155,1138,1118,1100,1080,1060,1040,1020,1002,983,965,
945,927,908,890,868,850,830,810};
doubleHU1[]={1100,1050,1000,950,900,850,800,750,700,650,600,550,500,450,400,
350,300,250,200,1,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36};
for(int i=0;i<=EN.length-1;i++){
if(S==EN[i]){
S1=EN[i];
h1=H[i];
hu1=HU[i];}
if(S<EN[i]&&S>EN[i+1]){

```

```

S1=EN[i];
S2=EN[i+1];
h1=H[i];
h2=H[i+1];
hu1=HU[i];
hu2=HU[i+1];}
for(int j=0;j<=EN1.length-1;j++){
if(S<EN1[j]&&S>EN1[j+1]) {
E1=H1[j];
E2=H1[j+1];
HUM1=HU1[j];
HUM2=HU1[j+1];}}
r[0]=10; r[1]=20; r[2]=S1; r[3]=S2; r[4]=h1; r[5]=h2; r[6]=hu1; r[7]=hu2; r[8]=E1;
r[9]=E2; r[10]=HUM1; r[11]=HUM2;}}

if(P2>=20 && P2<30) {
double EN[]={2.13,2.115,2.10,2.035,2.06,2.04,2.02,2.0,1.98,1.96,1.94,1.92,1.89,1.87,
1.84,1.81,1.78,1.75,1.73,1.70,1.68,1.65,1.62,1.59,1.56,1.538,1.51,1.48,1.45,1.42,1.40,
1.37,1.34,1.31,1.28,1.26,1.23};
double H[]={1585,1560,1532,1508,1481,1458,1431,1408,1382,1359,1333,1310,1288,
1262,1240,1215,1190,1167,1155,1138,1118,1100,1080,1060,1040,1020,1002,983,965,
945,927,908,890,868,850,830,810};
doubleHU[]={1100,1050,1000,950,900,850,800,750,700,650,600,550,500,450,400,
350,300,250,200,1,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36};
double EN1[]={2.09,2.07,2.05,2.035,2.02,2.0,1.98,1.96,1.94,1.92,1.89,1.87,1.85,1.82,
1.793,1.763,1.735,1.70,1.69,1.65,1.62,1.59,1.57,1.54,1.51,1.485,1.46,1.43,1.41,1.38,
1.355,1.325,1.30,1.27,1.245,1.22,1.193};
double H1[]={1584,1559,1532,1508,1481,1458,1431,1407,1381,1358,1332,1309,1285,
1260,1238,1215,1190,1165,1146,1126,1108,1090,1070,1050,1030,1012,994,975,957,
934,920,900,880,860,842,825,805};
double HU1[]={1100,1050,1000,950,900,850,800,750,700,650,600,550,500,450,400,
350,300,1,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36};

```

```

for(int i=0;i<=EN.length-1;i++){
if(S==EN[i]) {
S1=EN[i];
h1=H[i];
hu1=HU[i]; }
if(S<EN[i]&&S>EN[i+1]){
S1=EN[i];
S2=EN[i+1];
h1=H[i];
h2=H[i+1];
hu1=HU[i];
hu2=HU[i+1];}
for(int j=0;j<=EN1.length-1;j++){
if(S<EN1[j]&&S>EN1[j+1]) {
E1=H1[j];
E2=H1[j+1];
HUM1=HU1[j];
HUM2=HU1[j+1];}}
r[0]=20; r[1]=30; r[2]=S1; r[3]=S2; r[4]=h1; r[5]=h2; r[6]=hu1; r[7]=hu2; r[8]=E1; r[9]=E2;
r[10]=HUM1; r[11]=HUM2; }}

if(P2>=30 && P2<40) {
double EN[]={2.09,2.07,2.05,2.035,2.02,2.0,1.98,1.96,1.94,1.92,1.89,1.87,1.85,1.82,
1.793,1.763,1.735,1.70,1.69,1.65,1.62,1.59,1.57,1.54,1.51,1.485,1.46,1.43,1.41,1.38,
1.355,1.325,1.30,1.27,1.245,1.22,1.193};
double H[]={1584,1559,1532,1508,1481,1458,1431,1407,1381,1358,1332,1309,1285,
1260,1238,1215,1190,1165,1146,1126,1108,1090,1070,1050,1030,1012,994,975,957,
934,920,900,880,860,842,825,805};
doubleHU[]={1100,1050,1000,950,900,850,800,750,700,650,600,550,500,450,400,
350,300,1,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36};

```

```

double EN1[]={2.055,2.04,2.02,2.0,1.98,1.965,1.945,1.93,1.90,1.88,1.86,1.84,1.81,
1.79,1.76,1.73,1.695,1.68,1.65,1.625,1.60,1.57,1.55,1.52,1.50,1.47,1.445,1.42,1.39,
1.37,1.34,1.32,1.29,1.265,1.24,1.215,1.19};
double H1[]={1584,1559,1531,1508,1480,1457,1430,1407,1380,1358,1331,1308,1285,
1260,1237,1212,1187,1170,1150,1131,1115,1095,1078,1060,1040,1020,1000,983,965,
945,930,910,890,870,850,835,815};
double HU1[]={1100,1050,1000,950,900,850,800,750,700,650,600,550,500,450,400,
350,300,1,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38};
for(int i=0;i<=EN.length-1;i++){
if(S==EN[i]) {
S1=EN[i];
h1=H[i];
hu1=HU[i]; }
if(S<EN[i]&&S>EN[i+1]){
S1=EN[i];
S2=EN[i+1];
h1=H[i];
h2=H[i+1];
hu1=HU[i];
hu2=HU[i+1];}
for(int j=0;j<=EN1.length-1;j++){
if(S<EN1[j]&&S>EN1[j+1]) {
E1=H1[j];
E2=H1[j+1];
HUM1=HU1[j];
HUM2=HU1[j+1];}}
r[0]=30; r[1]=40; r[2]=S1;
r[3]=S2; r[4]=h1; r[5]=h2; r[6]=hu1; r[7]=hu2; r[8]=E1; r[9]=E2; r[10]=HUM1;
r[11]=HUM2; }}

```

```

if(P2>=40 && P2<50) {
double EN[]={2.055,2.04,2.02,2.0,1.98,1.965,1.945,1.93,1.90,1.88,1.86,1.84,1.81,1.79,
1.76,1.73,1.695,1.68,1.65,1.625,1.60,1.57,1.55,1.52,1.50,1.47,1.445,1.42,1.39,1.37,1.3
4,1.32,1.29,1.265,1.24,1.215,1.19};
double H[]={1584,1559,1531,1508,1480,1457,1430,1407,1380,1358,1331,1308,1285,
1260,1237,1212,1187,1170,1150,1131,1115,1095,1078,1060,1040,1020,1000,983,965,
945,930,910,890,870,850,835,815};
double HU[]={1100,1050,1000,950,900,850,800,750,700,650,600,550,500,450,400,350,
300,1,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38};
double EN1[]={2.03,2.01,1.995,1.98,1.96,1.94,1.92,1.90,1.88,1.86,1.84,1.805,1.79,1.76,
1.735,1.705,1.67,1.66,1.635,1.61,1.58,1.56,1.535,1.51,1.48,1.46,1.43,1.41,1.38,1.36,1.
33,1.31,1.285,1.26,1.235,1.21,1.185,1.16};
double H1[]={1585,1558,1531,1506,1480,1455,1430,1406,1380,1357,1331,1308,1283,
1260,1235,1210,1185,1173,1155,1138,1120,1100,1080,1064,1045,1025,1008,990,970,
950,935,915,897,880,860,842,822,805};
double HU1[]={1100,1050,1000,950,900,850,800,750,700,650,600,550,500,450,400,
350,300,1,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40};
for(int i=0;i<=EN.length-1;i++){
if(S==EN[i]) {
S1=EN[i];
h1=H[i];
hu1=HU[i];}
if(S<EN[i]&&S>EN[i+1]){
S1=EN[i];
S2=EN[i+1];
h1=H[i];
h2=H[i+1];
hu1=HU[i];
hu2=HU[i+1];}
for(int j=0;j<=EN1.length-1;j++){

```

```

if(S<EN1[j]&&S>EN1[j+1]) {
E1=H1[j];
E2=H1[j+1];
HUM1=HU1[j];
HUM2=HU1[j+1];}
r[0]=40; r[1]=50; r[2]=S1; r[3]=S2; r[4]=h1; r[5]=h2; r[6]=hu1; r[7]=hu2; r[8]=E1; r[9]=E2;
r[10]=HUM1; r[11]=HUM2; }}

if(P2>=50 && P2<60){
double EN[]={2.03,2.01,1.995,1.98,1.96,1.94,1.92,1.90,1.88,1.86,1.84,1.805,1.79,1.76,
1.735,1.705,1.67,1.66,1.635,1.61,1.58,1.56,1.535,1.51,1.48,1.46,1.43,1.41,1.38,1.36,
1.33,1.31,1.285,1.26,1.235,1.21,1.185,1.16};
double H[]={1585,1558,1531,1506,1480,1455,1430,1406,1380,1357,1331,1308,1283,
1260,1235,1210,1185,1173,1155,1138,1120,1100,1080,1064,1045,1025,1008,990,970,
950,935,915,897,880,860,842,822,805};
double HU[]={1100,1050,1000,950,900,850,800,750,700,650,600,550,500,450,400,350,
300,1,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40};
double EN1[]={2.01,1.995,1.98,1.96,1.94,1.92,1.90,1.88,1.86,1.84,1.82,1.795,1.77,1.74,
1.715,1.685,1.65,1.62,1.60,1.57,1.547,1.52,1.50,1.475,1.45,1.423,1.40,1.375,1.35,1.32
5,1.30,1.28,1.255,1.23,1.205,1.18,1.16};
double H1[]={1583,1558,1531,1506,1480,1455,1430,1405,1385,1355,1331,1308,1282,
1258,1235,1208,1180,1160,1140,1123,1105,1087,1069,1050,1030,1012,995,978,958,
940,920,903,885,868,850,830,811};
double HU1[]={1100,1050,1000,950,900,850,800,750,700,650,600,550,500,450,400,
350,1,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40};
for(int i=0;i<=EN.length-1;i++){
if(S==EN[i]){
S1=EN[i];
h1=H[i];
hu1=HU[i];}

```

```

if(S<EN[i]&&S>EN[i+1]){
S1=EN[i];
S2=EN[i+1];
h1=H[i];
h2=H[i+1];
hu1=HU[i];
hu2=HU[i+1];}
for(int j=0;j<=EN1.length-1;j++){
if(S<EN1[j]&&S>EN1[j+1]) {
E1=H1[j];
E2=H1[j+1];
HUM1=HU1[j];
HUM2=HU1[j+1];}}
r[0]=50; r[1]=60; r[2]=S1; r[3]=S2; r[4]=h1; r[5]=h2; r[6]=hu1; r[7]=hu2; r[8]=E1; r[9]=E2;
r[10]=HUM1; r[11]=HUM2; }}

if(P2>=60 && P2<80) {
double EN[]={2.01,1.995,1.98,1.96,1.94,1.92,1.90,1.88,1.86,1.84,1.82,1.795,1.77,1.74,
1.715,1.685,1.65,1.62,1.60,1.57,1.547,1.52,1.50,1.475,1.45,1.423,1.40,1.375,1.35,
1.325,1.30,1.28,1.255,1.23,1.205,1.18,1.16};
double H[]={1583,1558,1531,1506,1480,1455,1430,1405,1385,1355,1331,1308,1282,
1258,1235,1208,1180,1160,1140,1123,1105,1087,1069,1050,1030,1012,995,978,958,
940,920,903,885,868,850,830,811};
double HU[]={1100,1050,1000,950,900,850,800,750,700,650,600,550,500,450,400,
350,1,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40};
double EN1[]={1.98,1.96,1.94,1.925,1.91,1.89,1.87,1.85,1.83,1.81,1.785,1.76,1.74,
1.71,1.68,1.65,1.62,1.60,1.575,1.55,1.53,1.505,1.48,1.46,1.44,1.41,1.39,1.36,1.34,
1.32,1.295,1.27,1.25,1.225,1.20,1.18,1.155};
double H1[]={1583,1558,1530,1505,1480,1454,1430,1404,1380,1354,1330,1305,1280,
1255,1230,1205,1182,1165,1147,1130,1110,1094,1075,1058,1040,1020,1003,985,970,
950,930,915,895,878,860,840,825};

```

```

double HU1[]={1100,1050,1000,950,900,850,800,750,700,650,600,550,500,450,400,
350,1,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40};
for(int i=0;i<=EN.length-1;i++){
if(S==EN[i]) {
S1=EN[i];
h1=H[i];
hu1=HU[i]; }
if(S<EN[i]&&S>EN[i+1]) {
S1=EN[i];
S2=EN[i+1];
h1=H[i];
h2=H[i+1];
hu1=HU[i];
hu2=HU[i+1];}
for(int j=0;j<=EN1.length-1;j++){
if(S<EN1[j]&&S>EN1[j+1]) {
E1=H1[j];
E2=H1[j+1];
HUM1=HU1[j];
HUM2=HU1[j+1];}}
r[0]=60; r[1]=80; r[2]=S1; r[3]=S2; r[4]=h1; r[5]=h2; r[6]=hu1; r[7]=hu2; r[8]=E1; r[9]=E2;
r[10]=HUM1; r[11]=HUM2; }}

if(P2>=80 && P2<100) {
double EN[]={1.98,1.96,1.94,1.925,1.91,1.89,1.87,1.85,1.83,1.81,1.785,1.76,1.74,1.71,
1.68,1.65,1.62,1.60,1.575,1.55,1.53,1.505,1.48,1.46,1.44,1.41,1.39,1.36,1.34,1.32,1.29
5,1.27,1.25,1.225,1.20,1.18,1.155};
double H[]={1583,1558,1530,1505,1480,1454,1430,1404,1380,1354,1330,1305,1280,
1255,1230,1205,1182,1165,1147,1130,1110,1094,1075,1058,1040,1020,1003,985,
970,950,930,915,895,878,860,840,825};

```

```

double HU[]={1100,1050,1000,950,900,850,800,750,700,650,600,550,500,450,400,350,
1,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40};
double EN1[]={1.955,1.935,1.92,1.90,1.88,1.865,1.845,1.825,1.80,1.78,1.76,1.735,
1.71,1.68,1.655,1.62,1.60,1.58,1.56,1.535,1.51,1.49,1.468,1.445,1.42,1.40,1.38,1.357,
1.33,1.31,1.284,1.265,1.24,1.22,1.20,1.173,1.15};
double H1[]={1582,1558,1530,1505,1480,1453,1429,1404,1380,1353,1330,1304,1280,
1252,1228,1200,1189,1170,1150,1135,1117,1100,1080,1063,1045,1028,1010,990,975,
955,940,920,900,885,868,850,832};
double HU1[]={1100,1050,1000,950,900,850,800,750,700,650,600,550,500,450,400,
350,1,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40};
for(int i=0;i<=EN.length-1;i++){
if(S==EN[i]){
S1=EN[i];
h1=H[i];
hu1=HU[i];}
if(S<EN[i]&&S>EN[i+1]) {
S1=EN[i];
S2=EN[i+1];
h1=H[i];
h2=H[i+1];
hu1=HU[i];
hu2=HU[i+1];}
for(int j=0;j<=EN1.length-1;j++){
if(S<EN1[j]&&S>EN1[j+1]) {
E1=H1[j];
E2=H1[j+1];
HUM1=HU1[j];
HUM2=HU1[j+1];}}
r[0]=80; r[1]=100; r[2]=S1; r[3]=S2; r[4]=h1; r[5]=h2; r[6]=hu1; r[7]=hu2; r[8]=E1; r[9]=E2;
r[10]=HUM1; r[11]=HUM2; } }

```

.....

```

if(P2>=100 && P2<150){
double EN[]={1.955,1.935,1.92,1.90,1.88,1.865,1.845,1.825,1.80,1.78,1.76,1.735,1.71,
1.68,1.655,1.62,1.60,1.58,1.56,1.535,1.51,1.49,1.468,1.445,1.42,1.40,1.38,1.357,1.33,
1.31,1.284,1.265,1.24,1.22,1.20,1.173,1.15};
double H[]={1582,1558,1530,1505,1480,1453,1429,1404,1380,1353,1330,1304,1280,
1252,1228,1200,1189,1170,1150,1135,1117,1100,1080,1063,1045,1028,1010,990,975,
955,940,920,900,885,868,850,832};
double HU[]={1100,1050,1000,950,900,850,800,750,700,650,600,550,500,450,400,350,
1,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40};
double
EN1[]={1.91,1.89,1.873,1.855,1.84,1.82,1.80,1.78,1.757,1.74,1.71,1.69,1.66,1.635,1.60,
1.57,1.55,1.53,1.507,1.485,1.46,1.44,1.42,1.40,1.38,1.36,1.34,1.316,1.295,1.273,1.25,
1.23,1.21,1.19,1.17,1.145,1.105};
double H1[]={1581,1555,1530,1503,1478,1451,1426,1400,1376,1350,1325,1300,1274,
1247,1220,1195,1178,1160,1142,1125,1110,1090,1073,1055,1040,1020,1005,988,970,
950,935,920,900,885,868,850,810};
double HU1[]={1100,1050,1000,950,900,850,800,750,700,650,600,550,500,450,400,1,
2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,45};
for(int i=0;i<=EN.length-1;i++) {
if(S==EN[i]) {
S1=EN[i];
h1=H[i];
hu1=HU[i]; }
if(S<EN[i]&&S>EN[i+1]){
S1=EN[i];
S2=EN[i+1];
h1=H[i];
h2=H[i+1];
hu1=HU[i];
hu2=HU[i+1];}
for(int j=0;j<=EN1.length-1;j++){

```

```

if(S<EN1[j]&&S>EN1[j+1]) {
E1=H1[j];
E2=H1[j+1];
HUM1=HU1[j];
HUM2=HU1[j+1];}
r[0]=100; r[1]=150; r[2]=S1; r[3]=S2; r[4]=h1; r[5]=h2; r[6]=hu1; r[7]=hu2; r[8]=E1;
r[9]=E2; r[10]=HUM1; r[11]=HUM2; }}
if(P2>=150 && P2<200) {
double EN[]={1.91,1.89,1.873,1.855,1.84,1.82,1.80,1.78,1.757,1.74,1.71,1.69,1.66,
1.635,1.60,1.57,1.55,1.53,1.507,1.485,1.46,1.44,1.42,1.40,1.38,1.36,1.34,1.316,1.295,
1.273,1.25,1.23,1.21,1.19,1.17,1.145,1.105};
double H[]={1581,1555,1530,1503,1478,1451,1426,1400,1376,1350,1325,1300,1274,
1247,1220,1195,1178,1160,1142,1125,1110,1090,1073,1055,1040,1020,1005,988,970,
950,935,920,900,885,868,850,810};
double HU[]={1100,1050,1000,950,900,850,800,750,700,650,600,550,500,450,400,1,2,
4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,45};
double EN1[]={1.88,1.86,1.84,1.82,1.80,1.785,1.77,1.745,1.723,1.70,1.68,1.65,1.625,
1.595,1.56,1.55,1.525,1.505,1.485,1.465,1.445,1.425,1.40,1.385,1.365,1.345,1.325,
1.305,1.285,1.265,1.245,1.225,1.205,1.185,1.165,1.145,1.105};
double H1[]={1580,1553,1529,1501,1475,1450,1425,1400,1372,1350,1320,1295,1270,
1240,1210,1200,1180,1165,1150,1135,1115,1100,1080,1064,1047,1030,1012,997,980,
962,945,930,912,896,880,860,820};
double HU1[]={1100,1050,1000,950,900,850,800,750,700,650,600,550,500,450,400,1,
2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,45};
for(int i=0;i<=EN.length-1;i++) {
if(S==EN[i]) {
S1=EN[i];
h1=H[i];
hu1=HU[i]; }

```

```

if(S<EN[i]&&S>EN[i+1]){
S1=EN[i];
S2=EN[i+1];
h1=H[i];
h2=H[i+1];
hu1=HU[i];
hu2=HU[i+1];}
for(int j=0;j<=EN1.length-1;j++){
if(S<EN1[j]&&S>EN1[j+1]) {
E1=H1[j];
E2=H1[j+1];
HUM1=HU1[j];
HUM2=HU1[j+1];}}
r[0]=150; r[1]=200; r[2]=S1; r[3]=S2; r[4]=h1; r[5]=h2; r[6]=hu1; r[7]=hu2; r[8]=E1;
r[9]=E2; r[10]=HUM1; r[11]=HUM2; }}

if(P2>=200 && P2<250) {
double EN[]={1.88,1.86,1.84,1.82,1.80,1.785,1.77,1.745,1.723,1.70,1.68,1.65,1.625,
1.595,1.56,1.55,1.525,1.505,1.485,1.465,1.445,1.425,1.40,1.385,1.365,1.345,1.325,1.3
05,1.285,1.265,1.245,1.225,1.205,1.185,1.165,1.145,1.105};
double H[]={1580,1553,1529,1501,1475,1450,1425,1400,1372,1350,1320,1295,1270,
1240,1210,1200,1180,1165,1150,1135,1115,1100,1080,1064,1047,1030,1012,997,980,
962,945,930,912,896,880,860,820};
double HU[]={1100,1050,1000,950,900,850,800,750,700,650,600,550,500,450,400,1,2,
4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,45};
double EN1[]={1.85,1.835,1.82,1.80,1.78,1.76,1.74,1.72,1.70,1.68,1.65,1.62,1.60,1.56,
1.53,1.505,1.49,1.47,1.45,1.43,1.41,1.39,1.37,1.35,1.335,1.315,1.30,1.28,1.26,1.24,1.2
2,1.20,1.18,1.16,1.14,1.105};
double H1[]={1580,1553,1528,1500,1475,1448,1420,1397,1370,1346,1318,1290,1265,
1233,1202,1185,1170,1150,1136,1120,1100,1086,1070,1052,1038,1020,1004,988,970,
953,938,920,900,888,870,832};

```

```

double HU1[]={1100,1050,1000,950,900,850,800,750,700,650,600,550,500,450,
1,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,45};
for(int i=0;i<=EN.length-1;i++) {
if(S==EN[i]) {
S1=EN[i];
h1=H[i];
hu1=HU[i]; }
if(S<EN[i]&&S>EN[i+1]){
S1=EN[i];
S2=EN[i+1];
h1=H[i];
h2=H[i+1];
hu1=HU[i];
hu2=HU[i+1];}
for(int j=0;j<=EN1.length-1;j++){
if(S<EN1[j]&&S>EN1[j+1]) {
E1=H1[j];
E2=H1[j+1];
HUM1=HU1[j];
HUM2=HU1[j+1];}}
r[0]=200; r[1]=250; r[2]=S1; r[3]=S2; r[4]=h1; r[5]=h2; r[6]=hu1; r[7]=hu2; r[8]=E1;
r[9]=E2; r[10]=HUM1; r[11]=HUM2; }}

if(P2>=250 && P2<300) {
double EN[]={1.85,1.835,1.82,1.80,1.78,1.76,1.74,1.72,1.70,1.68,1.65,1.62,1.60,
1.56,1.53,1.505,1.49,1.47,1.45,1.43,1.41,1.39,1.37,1.35,1.335,1.315,1.30,1.28,1.26,
1.24,1.22,1.20,1.18,1.16,1.14,1.105};
double H[]={1580,1553,1528,1500,1475,1448,1420,1397,1370,1346,1318,1290,1265,
1233,1202,1185,1170,1150,1136,1120,1100,1086,1070,1052,1038,1020,1004,988,970,
953,938,920,900,888,870,832};
double HU[]={1100,1050,1000,950,900,850,800,750,700,650,600,550,500,450,1,2,4,
6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,45};

```

```

double EN1[]={1.83,1.815,1.80,1.78,1.76,1.74,1.72,1.70,1.68,1.65,1.63,1.60,1.57,
1.54,1.51,1.49,1.475,1.455,1.437,1.42,1.40,1.38,1.36,1.34,1.325,1.31,1.29,1.27,1.252,
1.235,1.215,1.20,1.18,1.16,1.14,1.105,1.05};
double H1[]={1579,1551,1525,1500,1472,1445,1420,1393,1378,1340,1315,1287,1258,
1229,1205,1188,1170,1155,1140,1122,1105,1090,1074,1058,1040,1025,1010,992,978,
960,945,928,913,895,880,845,800};
double HU1[]={1100,1050,1000,950,900,850,800,750,700,650,600,550,500,450,1,2,4,6,
8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,45,50};
for(int i=0;i<=EN.length-1;i++) {
if(S==EN[i]) {
S1=EN[i];
h1=H[i];
hu1=HU[i];}
if(S<EN[i]&&S>EN[i+1]){
S1=EN[i];
S2=EN[i+1];
h1=H[i];
h2=H[i+1];
hu1=HU[i];
hu2=HU[i+1];}
for(int j=0;j<=EN1.length-1;j++){
if(S<EN1[j]&&S>EN1[j+1]) {
E1=H1[j];
E2=H1[j+1];
HUM1=HU1[j];
HUM2=HU1[j+1];}}
r[0]=250; r[1]=300; r[2]=S1; r[3]=S2; r[4]=h1; r[5]=h2; r[6]=hu1; r[7]=hu2; r[8]=E1;
r[9]=E2; r[10]=HUM1; r[11]=HUM2; }}

```

```

if(P2>=300 && P2<400) {
double EN[]={1.83,1.815,1.80,1.78,1.76,1.74,1.72,1.70,1.68,1.65,1.63,1.60,1.57,1.54,
1.51,1.49,1.475,1.455,1.437,1.42,1.40,1.38,1.36,1.34,1.325,1.31,1.29,1.27,1.252,1.235,
1.215,1.20,1.18,1.16,1.14,1.105,1.05};
double H[]={1579,1551,1525,1500,1472,1445,1420,1393,1378,1340,1315,1287,1258,
1229,1205,1188,1170,1155,1140,1122,1105,1090,1074,1058,1040,1025,1010,992,978,
960,945,928,913,895,880,845,800};
double HU[]={1100,1050,1000,950,900,850,800,750,700,650,600,550,500,450,1,2,4,
6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,45,50};
double EN1[]={1.80,1.78,1.76,1.745,1.725,1.705,1.68,1.66,1.64,1.615,1.59,1.56,1.53,
1.48,1.47,1.45,1.435,1.415,1.40,1.38,1.36,1.345,1.33,1.31,1.295,1.28,1.26,1.243,1.227,
1.21,1.19,1.173,1.155,1.140,1.095,1.055};
double H1[]={1576,1550,1522,1496,1470,1442,1415,1390,1361,1335,1306,1277,1245,
1205,1190,1174,1158,1142,1127,1110,1095,1080,1065,1050,1032,1018,1000,987,970,
955,940,923,910,890,853,820};
double HU1[]={1100,1050,1000,950,900,850,800,750,700,650,600,550,500,1,2,4,6,
8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,45,50};
for(int i=0;i<=EN.length-1;i++) {
if(S==EN[i]) {
S1=EN[i];
h1=H[i];
hu1=HU[i]; }
if(S<EN[i]&&S>EN[i+1]){
S1=EN[i];
S2=EN[i+1];
h1=H[i];
h2=H[i+1];
hu1=HU[i];
hu2=HU[i+1];}
for(int j=0;j<=EN1.length-1;j++){

```

```

if(S<EN1[j]&&S>EN1[j+1]) {
E1=H1[j];
E2=H1[j+1];
HUM1=HU1[j];
HUM2=HU1[j+1];}
r[0]=300; r[1]=400; r[2]=S1; r[3]=S2; r[4]=h1; r[5]=h2; r[6]=hu1; r[7]=hu2; r[8]=E1;
r[9]=E2; r[10]=HUM1; r[11]=HUM2; } }

if(P2>=400 && P2<500) {
double EN[]={1.80,1.78,1.76,1.745,1.725,1.705,1.68,1.66,1.64,1.615,1.59,1.56,1.53,
1.48,1.47,1.45,1.435,1.415,1.40,1.38,1.36,1.345,1.33,1.31,1.295,1.28,1.26,1.243,1.227,
1.21,1.19,1.173,1.155,1.140,1.095,1.055};
double H[]={1576,1550,1522,1496,1470,1442,1415,1390,1361,1335,1306,1277,1245,
1205,1190,1174,1158,1142,1127,1110,1095,1080,1065,1050,1032,1018,1000,987,970,
955,940,923,910,890,853,820};
double HU[]={1100,1050,1000,950,900,850,800,750,700,650,600,550,500,1,2,4,6,8,
10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,45,50};
double EN1[]={1.79,1.77,1.755,1.74,1.72,1.70,1.68,1.66,1.635,1.61,1.585,1.56,1.53,
1.495,1.465,1.45,1.43,1.415,1.40,1.38,1.365,1.35,1.33,1.315,1.30,1.283,1.268,1.25,1.2
35,1.22,1.20,1.185,1.17,1.15,1.14,1.10,1.06};
double H1[]={1600,1573,1548,1520,1493,1465,1440,1410,1384,1356,1329,1298,1269,
1234,1205,1190,1175,1160,1145,1130,1115,1100,1085,1070,1053,1039,1023,1010,99
3,980,962,949,932,920,900,865,830};
double HU1[]={1150,1100,1050,1000,950,900,850,800,750,700,650,600,550,500,1,2,4,
6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,45,50};
for(int i=0;i<=EN.length-1;i++) {
if(S==EN[i]) {
S1=EN[i];
h1=H[i];
hu1=HU[i]; }

```

```

if(S<EN[i]&&S>EN[i+1]){
S1=EN[i];
S2=EN[i+1];
h1=H[i];
h2=H[i+1];
hu1=HU[i];
hu2=HU[i+1];}
for(int j=0;j<=EN1.length-1;j++){
if(S<EN1[j]&&S>EN1[j+1]) {
E1=H1[j];
E2=H1[j+1];
HUM1=HU1[j];
HUM2=HU1[j+1];}}
r[0]=400; r[1]=500; r[2]=S1; r[3]=S2; r[4]=h1; r[5]=h2; r[6]=hu1; r[7]=hu2; r[8]=E1;
r[9]=E2; r[10]=HUM1; r[11]=HUM2; } }

```

```

if(P2>=500 && P2<600){
double EN[]={1.79,1.77,1.755,1.74,1.72,1.70,1.68,1.66,1.635,1.61,1.585,1.56,1.53,
1.495,1.465,1.45,1.43,1.415,1.40,1.38,1.365,1.35,1.33,1.315,1.30,1.283,1.268,1.25,1.2
35,1.22,1.20,1.185,1.17,1.15,1.14,1.10,1.06};
double H[]={1600,1573,1548,1520,1493,1465,1440,1410,1384,1356,1329,1298,1269,
1234,1205,1190,1175,1160,1145,1130,1115,1100,1085,1070,1053,1039,1023,1010,99
3,980,962,949,932,920,900,865,830};
double HU[]={1150,1100,1050,1000,950,900,850,800,750,700,650,600,550,500,1,2,4,
6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,45,50};
double EN1[]={1.77,1.75,1.735,1.72,1.695,1.68,1.657,1.635,1.61,1.59,1.56,1.535,1.50,
1.46,1.448,1.43,1.415,1.40,1.38,1.365,1.35,1.335,1.32,1.305,1.29,1.275,1.26,1.245,1.2
3,1.213,1.20,1.18,1.165,1.15,1.135,1.10,1.06};
double H1[]={1598,1570,1545,1520,1490,1460,1437,1408,1380,1350,1320,1290,1255,
1216,1203,1190,1175,1160,1145,1130,1116,1100,1088,1070,1058,1041,1030,1013,
1000,984,970,955,940,925,910,880,835};

```

```

double HU1[]={1150,1100,1050,1000,950,900,850,800,750,700,650,600,550,500,1,2,
4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,45,50};
for(int i=0;i<=EN.length-1;i++) {
if(S==EN[i]) {
S1=EN[i];
h1=H[i];
hu1=HU[i];}
if(S<EN[i]&&S>EN[i+1]){
S1=EN[i];
S2=EN[i+1];
h1=H[i];
h2=H[i+1];
hu1=HU[i];
hu2=HU[i+1];}
for(int j=0;j<=EN1.length-1;j++){
if(S<EN1[j]&&S>EN1[j+1]) {
E1=H1[j];
E2=H1[j+1];
HUM1=HU1[j];
HUM2=HU1[j+1];}}
r[0]=500; r[1]=600; r[2]=S1; r[3]=S2; r[4]=h1; r[5]=h2; r[6]=hu1; r[7]=hu2; r[8]=E1;
r[9]=E2; r[10]=HUM1; r[11]=HUM2; } }

if(P2>=600 && P2<800) {
double EN[]={1.77,1.75,1.735,1.72,1.695,1.68,1.657,1.635,1.61,1.59,1.56,1.535,1.50,
1.46,1.448,1.43,1.415,1.40,1.38,1.365,1.35,1.335,1.32,1.305,1.29,1.275,1.26,1.245,1.2
3,1.213,1.20,1.18,1.165,1.15,1.135,1.10,1.06};
double H[]={1598,1570,1545,1520,1490,1460,1437,1408,1380,1350,1320,1290,1255,
1216,1203,1190,1175,1160,1145,1130,1116,1100,1088,1070,1058,1041,1030,1013,10
00,984,970,955,940,925,910,880,835};
double HU[]={1150,1100,1050,1000,950,900,850,800,750,700,650,600,550,500,1,2,4,6,
8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,45,50};

```

```

double EN1[]={1.735,1.72,1.70,1.68,1.66,1.64,1.62,1.60,1.575,1.55,1.52,1.49,1.445,
1.415,1.40,1.385,1.375,1.36,1.345,1.33,1.315,1.305,1.29,1.275,1.26,1.247,1.233,1.22,
1.205,1.19,1.175,1.16,1.145,1.135,1.10,1.06};
double H1[]={1593,1567,1540,1510,1383,1354,1327,1399,1370,1340,1305,1273,1230,
1200,1185,1170,1158,1144,1130,1117,1102,1090,1075,1060,1048,1033,1020,1005,99
1,978,965,950,937,924,890,854};
double HU1[]={1150,1100,1050,1000,950,900,850,800,750,700,650,600,550,1,2,4,6,8,
10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,45,50};
for(int i=0;i<=EN.length-1;i++) {
if(S==EN[i]) {
S1=EN[i];
h1=H[i];
hu1=HU[i]; }
if(S<EN[i]&&S>EN[i+1]){
S1=EN[i];
S2=EN[i+1];
h1=H[i];
h2=H[i+1];
hu1=HU[i];
hu2=HU[i+1];}
for(int j=0;j<=EN1.length-1;j++){
if(S<EN1[j]&&S>EN1[j+1]) {
E1=H1[j];
E2=H1[j+1];
HUM1=HU1[j];
HUM2=HU1[j+1];}}
r[0]=600; r[1]=800; r[2]=S1; r[3]=S2; r[4]=h1; r[5]=h2; r[6]=hu1; r[7]=hu2; r[8]=E1;
r[9]=E2; r[10]=HUM1; r[11]=HUM2; }}

```

```

if(P2>=800 && P2<1000) {
double EN[]={1.735,1.72,1.70,1.68,1.66,1.64,1.62,1.60,1.575,1.55,1.52,1.49,1.445,
1.415,1.40,1.385,1.375,1.36,1.345,1.33,1.315,1.305,1.29,1.275,1.26,1.247,1.233,1.22,
1.205,1.19,1.175,1.16,1.145,1.135,1.10,1.06};
double H[]={1593,1567,1540,1510,1383,1354,1327,1399,1370,1340,1305,1273,1230,
1200,1185,1170,1158,1144,1130,1117,1102,1090,1075,1060,1048,1033,1020,1005,99
1,978,965,950,937,924,890,854};
double HU[]={1150,1100,1050,1000,950,900,850,800,750,700,650,600,550,1,2,4,6,8,
10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,45,50};
double EN1[]={1.71,1.69,1.67,1.65,1.635,1.61,1.59,1.57,1.54,1.51,1.48,1.44,1.40,1.39,
1.375,1.36,1.345,1.335,1.32,1.31,1.30,1.285,1.27,1.26,1.25,1.235,1.22,1.21,1.20,1.185,
1.17,1.155,1.145,1.13,1.10,1.07};
double H1[]={1590,1560,1533,1505,1478,1450,1419,1389,1360,1325,1290,1248,1204,
1193,1180,1167,1153,1140,1128,1115,1100,1088,1075,1063,1050,1036,1023,1010,
1000,985,970,958,945,982,900,870};
double HU1[]={1150,1100,1050,1000,950,900,850,800,750,700,650,600,550,1,2,4,6,8,
10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,45,50};
for(int i=0;i<=EN.length-1;i++) {
if(S==EN[i]) {
S1=EN[i];
h1=H[i];
hu1=HU[i]; }
if(S<EN[i]&&S>EN[i+1]){
S1=EN[i];
S2=EN[i+1];
h1=H[i];
h2=H[i+1];
hu1=HU[i];
hu2=HU[i+1];}
for(int j=0;j<=EN1.length-1;j++){

```

```

if(S<EN1[j]&&S>EN1[j+1]) {
E1=H1[j];
E2=H1[j+1];
HUM1=HU1[j];
HUM2=HU1[j+1];}
r[0]=800; r[1]=1000; r[2]=S1; r[3]=S2; r[4]=h1; r[5]=h2; r[6]=hu1; r[7]=hu2; r[8]=E1;
r[9]=E2; r[10]=HUM1; r[11]=HUM2; } }

```

```

if(P2>=1000 && P2<1200) {
double EN[]={1.71,1.69,1.67,1.65,1.635,1.61,1.59,1.57,1.54,1.51,1.48,1.44,1.40,1.39,
1.375,1.36,1.345,1.335,1.32,1.31,1.30,1.285,1.27,1.26,1.25,1.235,1.22,1.21,1.20,1.185,
1.17,1.155,1.145,1.13,1.10,1.07};
double H[]={1590,1560,1533,1505,1478,1450,1419,1389,1360,1325,1290,1248,1204,
1193,1180,1167,1153,1140,1128,1115,1100,1088,1075,1063,1050,1036,1023,1010,
1000,985,970,958,945,982,900,870};
double HU[]={1150,1100,1050,1000,950,900,850,800,750,700,650,600,550,1,2,4,6,8,
10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,45,50};
double EN1[]={1.685,1.67,1.65,1.63,1.610,1.585,1.56,1.54,1.52,1.48,1.45,1.40,1.362,
1.35,1.34,1.33,1.32,1.305,1.29,1.28,1.27,1.26,1.25,1.235,1.225,1.21,1.20,1.19,1.18,
1.165,1.15,1.14,1.13,1.10,1.07};
double H1[]={1584,1558,1529,1500,1470,1440,1410,1380,1350,1309,1270,1220,1182,
1170,1160,1150,1135,1122,1110,1100,1085,1072,1060,1050,1037,1020,1012,1000,
990,977,963,950,940,910,880};
double HU1[]={1150,1100,1050,1000,950,900,850,800,750,700,650,600,1,2,4,6,8,10,
12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,45,50};
for(int i=0;i<=EN.length-1;i++) {
if(S==EN[i]) {
S1=EN[i];
h1=H[i];
hu1=HU[i]; }
}

```

```

if(S<EN[i]&&S>EN[i+1]){
S1=EN[i];
S2=EN[i+1];
h1=H[i];
h2=H[i+1];
hu1=HU[i];
hu2=HU[i+1];}
for(int j=0;j<=EN1.length-1;j++) {
if(S<EN1[j]&&S>EN1[j+1]) {
E1=H1[j];
E2=H1[j+1];
HUM1=HU1[j];
HUM2=HU1[j+1];}}
r[0]=1000;
r[1]=1200; r[2]=S1; r[3]=S2; r[4]=h1; r[5]=h2; r[6]=hu1; r[7]=hu2; r[8]=E1; r[9]=E2;
r[10]=HUM1; r[11]=HUM2; } }
if(P2>=1200 && P2<1600) {
double EN[]={1.685,1.67,1.65,1.63,1.610,1.585,1.56,1.54,1.52,1.48,1.45,1.40,1.362,
1.35,1.34,1.33,1.32,1.305,1.29,1.28,1.27,1.26,1.25,1.235,1.225,1.21,1.20,1.19,1.18,
1.165,1.15,1.14,1.13,1.10,1.07};
double H[]={1584,1558,1529,1500,1470,1440,1410,1380,1350,1309,1270,1220,1182,
1170,1160,1150,1135,1122,1110,1100,1085,1072,1060,1050,1037,1020,1012,1000,
990,977,963,950,940,910,880};
double HU[]={1150,1100,1050,1000,950,900,850,800,750,700,650,600,1,2,4,6,8,10,
12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,45,50};
double EN1[]={1.65,1.635,1.61,1.595,1.57,1.55,1.52,1.50,1.46,1.43,1.38,1.33,1.315,
1.31,1.30,1.29,1.28,1.27,1.26,1.25,1.24,1.23,1.22,1.21,1.20,1.19,1.18,1.17,1.16,1.15,
1.14,1.13,1.12,1.10,1.07};
double H1[]={1576,1550,1518,1488,1460,1425,1390,1360,1318,1277,1225,1167,1155,
1150,1140,1130,1120,1105,1096,1085,1076,1065,1055,1045,1034,1022,1010,1000,
990,980,970,958,950,920,900};

```

```

double HU1[]={1150,1100,1050,1000,950,900,850,800,750,700,650,600,1,2,4,6,8,10,
12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,45,50};
for(int i=0;i<=EN.length-1;i++) {
if(S==EN[i]) {
S1=EN[i];
h1=H[i];
hu1=HU[i]; }
if(S<EN[i]&&S>EN[i+1]){
S1=EN[i];
S2=EN[i+1];
h1=H[i];
h2=H[i+1];
hu1=HU[i];
hu2=HU[i+1];}
for(int j=0;j<=EN1.length-1;j++){
if(S<EN1[j]&&S>EN1[j+1]) {
E1=H1[j];
E2=H1[j+1];
HUM1=HU1[j];
HUM2=HU1[j+1];}}
r[0]=1200; r[1]=1600; r[2]=S1; r[3]=S2; r[4]=h1; r[5]=h2; r[6]=hu1; r[7]=hu2; r[8]=E1;
r[9]=E2; r[10]=HUM1; r[11]=HUM2; } }

if(P2>=1600 && P2<2000) {
double EN[]={1.65,1.635,1.61,1.595,1.57,1.55,1.52,1.50,1.46,1.43,1.38,1.33,1.315,1.31,
1.30,1.29,1.28,1.27,1.26,1.25,1.24,1.23,1.22,1.21,1.20,1.19,1.18,1.17,1.16,1.15,1.14,
1.13,1.12,1.10,1.07};
double H[]={1576,1550,1518,1488,1460,1425,1390,1360,1318,1277,1225,1167,1155,
1150,1140,1130,1120,1105,1096,1085,1076,1065,1055,1045,1034,1022,1010,1000,
990,980,970,958,950,920,900};
double HU[]={1150,1100,1050,1000,950,900,850,800,750,700,650,600,1,2,4,6,8,10,
12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,45,50};

```

```

double EN1[]={1.64,1.62,1.60,1.58,1.56,1.54,1.515,1.49,1.46,1.43,1.38,1.325,1.28,
1.28,1.27,1.26,1.255,1.245,1.235,1.225,1.22,1.21,1.20,1.195,1.19,1.18,1.17,1.16,1.15,
1.14,1.13,1.12,1.11,1.09,1.07};
double H1[]={1595,1567,1535,1505,1475,1445,1410,1375,1340,1300,1244,1180,1130,
1130,1120,1110,1104,1090,1082,1070,1063,1055,1045,1035,1030,1020,1010,1000,
990,980,970,960,950,925,905};
double HU1[]={1200,1150,1100,1050,1000,950,900,850,800,750,700,650,1,2,4,6,8,10,
12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,45,50};
for(int i=0;i<=EN.length-1;i++) {
if(S==EN[i]) {
S1=EN[i];
h1=H[i];
hu1=HU[i]; }
if(S<EN[i]&&S>EN[i+1]){
S1=EN[i];
S2=EN[i+1];
h1=H[i];
h2=H[i+1];
hu1=HU[i];
hu2=HU[i+1];}
for(int j=0;j<=EN1.length-1;j++) {
if(S<EN1[j]&&S>EN1[j+1]) {
E1=H1[j];
E2=H1[j+1];
HUM1=HU1[j];
HUM2=HU1[j+1];}}
r[0]=1600; r[1]=2000; r[2]=S1; r[3]=S2; r[4]=h1; r[5]=h2; r[6]=hu1; r[7]=hu2; r[8]=E1;
r[9]=E2; r[10]=HUM1; r[11]=HUM2; }}

```

```

if(P2==2000) {
double EN[]={1.64,1.62,1.60,1.58,1.56,1.54,1.515,1.49,1.46,1.43,1.38,1.325,1.28,1.28,
1.27,1.26,1.255,1.245,1.235,1.225,1.22,1.21,1.20,1.195,1.19,1.18,1.17,1.16,1.15,1.14,
1.13,1.12,1.11,1.09,1.07};
double H[]={1595,1567,1535,1505,1475,1445,1410,1375,1340,1300,1244,1180,1130,
1130,1120,1110,1104,1090,1082,1070,1063,1055,1045,1035,1030,1020,1010,1000,
990,980,970,960,950,925,905};
double HU[]={1200,1150,1100,1050,1000,950,900,850,800,750,700,650,1,2,4,6,8,10,
12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,45,50};
for(int i=0;i<=EN.length-1;i++) {
if(S==EN[i]) {
S1=EN[i];
h1=H[i];
hu1=HU[i]; }
if(S<EN[i]&&S>EN[i+1]){
S1=EN[i];
S2=EN[i+1];
h1=H[i];
h2=H[i+1];
hu1=HU[i];
hu2=HU[i+1];}
r[0]=2000; r[1]=0;
r[2]=S1; r[3]=S2; r[4]=h1; r[5]=h2; r[6]=0; r[7]=hu2; r[8]=0; r[9]=0; r[10]=hu1; r[11]=0; }}
return(r); }

```

```

public static double[] INTERPOLACION(double ori,int men,int may,double H1,double
H2,double S1,double S2) {
double R1=0,R2=0,R3=0,R4=0,RH=0,RS=0;
double R[]=new double [2];
R1=ori-men;
R2=men-may;
R3=H1-H2;
R4=S1-S2;
R[0]= (((R1*R3)/R2)+H1);
R[1]= (((R1*R4)/R2)+S1);
return(R); }

```

```

public static double[] INTERPOLACIONDOBLE(double oriP,double oriT,double menP,
double mayP,double menT,double mayT,double H1,double h1,double S1,double s1,
double H2,double h2,double S2,double s2) {
double R1=0,R2=0,R3=0,R4=0,R5=0,R6=0,HE1=0,HE2=0,ES1=0,ES2=0,RH=0,RS=0;
double R[]=new double [2];
R1=oriT-menT;
R2=menT-mayT;
R3=H1-h1;
R4=H2-h2;
R5=S1-s1;
R6=S2-s2;
HE1= (((R1*R3)/R2)+H1);
HE2= (((R1*R4)/R2)+H2);
ES1= (((R1*R5)/R2)+S1);
ES2= (((R1*R6)/R2)+S2);

```

```
//SEGUNDA INTERPOLACION
```

```
double R1F=oriP-menP;
```

```
double R2F=menP-mayP;
```

```
double R3F=HE1-HE2;
```

```
double R4F=ES1-ES2;
```

```
R[0]=(((R1F*R3F)/R2F)+HE1);
```

```
R[1]=(((R1F*R4F)/R2F)+ES1);
```

```
return(R); }
```

```
public static double[] TIPOS(String TIPO,double H,double S,double H2,double  
H3,double h4,double hf,double hf2){
```

```
double wu[]=new double[2];
```

```
if(TIPO=="CONVENCIONAL"){
```

```
double R=0,Q=0,H4=0,ef=0;
```

```
H4=h4;
```

```
Q=H-H4;
```

```
wu[0]=H-H2;
```

```
wu[1]=((wu[0]/Q)*100); }
```

```
if(TIPO=="RECALENTAMIENTO"){
```

```
double H5=0,H6=0,r=0,r1=0,q=0,q1=0,w=0,ef=0;
```

```
H5=hf;
```

```
H6=H5;
```

```
r= H-H2;
```

```
r1=H3-h4;
```

```
q=H-H6;
```

```
q1=H3-H2;
```

```
wu[0]=r+r1;
```

```
wu[1]=(wu[0]/(q+q1)*100); }
```

```

if(TIPO=="REGENERATIVO"){
double r1=0,r2=0,r3=0,r4=0,e=0,w=0;
r1=H-H2;
r2=((hf2-hf)/(H2-hf));
r3=H2-H3;
r4=H-hf2;
wu[0]=(r1+((1-r2)*r3));
wu[1]=(wu[0]/r4)*100; }
return(wu);
}

public static double[] TIPOS1(String TIPO,double H,double S,double H2,double
H3,double h4,double H5,double hf,double hf2){
double wu[]=new double[2];
if(TIPO=="MIXTO"){
double r1=0,r2=0,r3=0,r4=0,r5=0,e=0,w=0;
r1=H-H2;
r2=H3-h4;
r3=((hf-hf2)/(h4-hf2));
r4=h4-H5;
r5=((H-hf)+(H3-H2));
wu[0]=((r1+r2)+((1-r3)*r4));
wu[1]=(wu[0]/r5)*100; }
return(wu);
}

```