

5
3
14
LOPEZ

PROYECTO DE
PROGRAMA DE CÁLCULO DE
AEROENFRIADORES

1998

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES

I. C. A. I.



PROYECTO DE
PROGRAMA DE CÁLCULO DE
AEROENFRIADORES

AMELIA RUESTRA LÓPEZ

Madrid, 1998

EL PRESENTE PROYECTO CONTIENE LOS DOCUMENTOS SIGUIENTES.

Documento n° 1	Memoria (total 84 páginas) Memoria (67 páginas) Anejo 1 (17 páginas)
Documento n° 2	Listados (122 páginas)
Documento n°3	Planos (1 página)
Documento n° 4	Cálculos (1 página)
Documento n° 5	Estudio Económico (2 páginas)
Documento n° 6	Pliego de Condiciones (7 páginas) Condiciones Generales (2 páginas) Condiciones Económicas(2 páginas) Condiciones Técnicas(2 páginas)
Documento n° 7	Presupuesto(10 páginas) Mediciones (2 páginas) Precios Unitarios (1 página) Presupuesto Parcial (3 páginas) Presupuesto general(1 página)

**ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INDUSTRIALES**

I.C.A.I.

**PROYECTO DE PROGRAMA DE CÁLCULO
DE AEROENFRIADORES**

V. B.

AMELIA Riestra López

Madrid, Septiembre de 1998

MEMORIA

MEMORIA

ÍNDICE DE LA MEMORIA.

1. INTRODUCCIÓN	3
1.1 OBJETIVO DEL PROYECTO	3
1.2 ENTORNO DE TRABAJO	4
1.3 HERRAMIENTAS DE PROGRAMACIÓN	5
1.3.1 <i>Microsoft Windows</i>	6
1.3.2 <i>Microsoft Visual Basic</i>	7
1.3.3 <i>Microsoft Access 97</i>	10
1.3.4 <i>Microsoft Word 97</i>	12
2.FUNCIONAMIENTO DE AEROENFRIADORES	13
3.NOCIONES TERMODINÁMICAS	14
3.1 DIAGRAMA PSICROMÉTRICO	15
3.2 INTERCAMBIO DE CALOR	25
3.3 DINÁMICA DE FLUIDOS	27
3.4 CARACTERÍSTICAS DE FLUIDOS	29

4. FASES DEL PROYECTO	31
4.1 REPASO TERMODINÁMICO	31
4.2 APRENDIZAJE DE VISUAL BASIC 5.0	32
4.3 CONEXIONES CON BASES DE DATOS	33
4.4 CONEXIONES CON MICORSOFT WORD	37
5. ESTRUCTURA DEL PROYECTO	40
5.1 PROCESO DE CÁLCULO	40
5.2 BASES DE DATOS UTILIZADAS	48
5.2.1 <i>Base de datos fluidos</i>	48
5.2.2 <i>Base de datos Aeros</i>	50
5.2.3 <i>Base de datos Bclientes</i>	63
6. VALORACIÓN ECONÓMICA	65
7. BIBLIOGRAFÍA	66

1. INTRODUCCIÓN

1.1. Objetivos del proyecto.

El objetivo del proyecto es realizar una aplicación de cálculo de un módulo de aeroenfriadores.

El proceso de cálculo anterior se realizaba manualmente, de ahí que se necesite una automatización del método, para así poder reducir el tiempo de calculo y el coste en términos de temporales que acarrea la empresa.

El proyecto deberá realizar el cálculo del aeroenfriador cumpliendo unas especificaciones marcadas por el cliente. Cada uno de los resultados del cálculo, así como el cliente y el proyecto que haya sido encargado, deben ser guardados en una base de datos para facilitar su posterior visualización.

1.2 Entorno de trabajo.

Al ser un programa que necesite una interfaz gráfica y suficientemente sencilla para el usuario y al estar vinculado a una utilización dinámica y rápida, el entorno Windows es idóneo para este tipo de aplicación.

Al estar hoy en día tan generalizado dicha herramienta hará que el proyecto sea más versátil y con un campo de aplicación más amplio, además de contar con un entorno visual idóneo para la ejecución de proyecto.

Con el sistema Windows tenemos la facilidad de disponer de varias aplicaciones simultáneas, que pueden ser necesarias a su vez para el desarrollo del cálculo.

1.3 Herramientas de programación.

Como he comentado en el apartado anterior es necesario Windows para nuestra aplicación.

Como herramientas de programación bajo Windows he utilizado las siguientes:

- **Microsoft Windows 95**, de soporte de los programas posteriores
- **Microsoft Visual Basic v. 5**, como herramienta principal.
- **Microsoft Access 97**, para el diseño y creación de las bases de datos.
- **Microsoft Word 97**, utilizado para la oferta al cliente

A continuación describo con mas detenimiento cada uno de estos apartados.

1.3.1 Microsoft Windows.

El proyecto se ha realizado en Windows 95, pero es también compatible con Windows NT, no siendo así con versiones anteriores.

Trabajaremos con 32 bits.

El entorno Windows para este tipo de programas es el ideal, ya que se consigue una relación con el usuario más sencilla, de forma que el programa está instalado como cualquier otro de Microsoft.

1.3.2 Microsoft Visual Basic.

Cuando se lanzó Visual Basic 1.0, Bill Gates lo describió como asombroso. Steve Gibson, en *infoworld* dijo que Visual Basic era un "increíble nuevo milagro" y que "iba a cambiar radicalmente la forma en que la gente piensa y utiliza Windows.

Bill Gates define Visual Basic como "una herramienta sencilla, pero potente para desarrollar aplicaciones Windows en Basic", ya que este lenguaje era su predecesor, como su propio nombre indica. Esto no parece suficiente para justificar toda la expectación levantada respecto al tema, hasta que se cae en la cuenta que Windows es utilizado por millones de personas y que desarrollar una aplicación para Microsoft Windows precisaba anteriormente un experto programador en C provisto de alrededor de 10 Kilos de documentación para el determinado compilador en C y los aditamentos esenciales.

Visual Basic 2.0 era más rápido e incluso más fácil de utilizar, Visual Basic 3.0 añadió simplemente un modo de controlar las bases

de datos más potentes existentes, Visual Basic 4.0 incrementa un soporte para desarrollo en 32 bits e inicia el proceso de convertir Visual Basic en un lenguaje de programación completamente orientado a objetos. Visual Basic 5.0 añade los controles ActiveX, la posibilidad de crear páginas Web, es decir, orientándose a Internet y la mejora de OLE.

Los interfaces gráficos de usuario, o IGU(Graphical User Interfaces) han revolucionado la industria de las microcomputadoras. Quizá más importante a largo plazo que el aspecto para las aplicaciones para Windows es la sensación que proporcionan. Las aplicaciones para Windows tienen generalmente una interfaz de usuario consistente. Esto significa que los usuarios tienen más tiempo para dominar la aplicación sin tener que preocuparse de qué teclas deben pulsarse y para qué dentro de los menús y de los cuadros de diálogo.

El sistema de programación en Visual Basic es perfecto para el entorno Windows y sus aplicaciones. Es una programación orientada a eventos con herramientas de diseño virtual, lo que permite aplicaciones muy importantes para Windows.

Y por último, pero no con menor importancia, ha sido fundamental la elección de Visual Basic como lenguaje de programación debido a que al pertenecer a la familia Microsoft, cualquier tipo de comunicación con un programa de la misma resulta más sencillo. En el caso de este proyecto, hubo que comunicarse con Microsoft Access y con Microsoft Word, de tal forma que en nuestro caso resultó más fácil y determinante.

1.3.3. *Microsoft Access 97.*

Antes de comentar el programa Microsoft Access debemos introducir algunos términos importantes para la comprensión del objetivo del mismo, como por ejemplo qué es una base de datos. Una *base de datos* es un conjunto de información relacionada con un tema o propósito particular. Un *sistema de administración de bases de datos* (DBMS) es un sistema que almacena y recupera información en una base de datos. Es el archivador o agenda.

Microsoft Access es un *sistema de administración de bases de datos relacionales* (RDBMS) que almacena y recupera la información de acuerdo con relaciones definidas por el usuario.

Con él se pueden crear consultas, informes, formularios, tablas de datos y relacionarlos entre ellos, consiguiendo una efectividad resaltable al no tener que repetir ningún dato dentro de sus registros.

En el caso de este proyecto, se utilizará para guardar diversos datos sobre los tipos de baterías, características de las mismas y una base de datos de los clientes de la empresa, con los proyectos y cálculos desarrollados para sus pedidos.

Aprovecha al máximo la potencia gráfica de Windows ofreciendo métodos visuales de acceso a sus datos o proporcionando maneras simples y directas de presentar y trabajar con su información.

Las potentes posibilidades de consulta y conexión son una ayuda para encontrar la información deseada.

1.3.3. Microsoft Word 97.

Desde su introducción en su primera versión para MS-DOS a comienzos de los ochenta Microsoft Word marca la pauta en tratamiento de textos y es actualmente el más popular para el entorno Windows.

Como resultado de las exhaustivas investigaciones realizadas, así como de las opiniones y peticiones recogidas de los usuarios, responde a los criterios de utilización que hoy día demandan.

En el proyecto fue necesario para realizar los documentos de las ofertas a los clientes.

2. FUNCIONAMIENTO GENERAL DE AEROENFRIADORES.

El aeroenfriador es el objeto de nuestro cálculo. Funciona basándose en intercambios de calor.

Un aeroenfriador o refrigerador es un equipo formado por una batería de tubos y aletas, y un número determinado de ventiladores, todo ello integrado en una carcasa de chapa. Su función es la de enfriar un fluido (agua , aceite, glicol, vapor, etc....) con aire.

El fluido entra en los tubos que van atravesando las aletas. El aire es impulsado por unos ventiladores. El sistema funciona de forma que el fluido se enfría al pasar por las aletas que le están dando el calor al aire. En sí mismo es un sistema de intercambio de calor, que estudiaremos con mayor profundidad en apartados posteriores.

3.NOCIONES TERMODINÁMICAS.

Al estar este programa orientado a utilizarse por personas que conozcan la materia de intercambiadores de calor y Aeroenfriadores, y lo más importante, que conozcan con detenimiento el producto que ofrece la empresa (tanto los tipos de baterías, sus diferencias entre ellas,...) el programa no está diseñado para un uso generalizado, es decir, la persona que utilice este programa debe saber en cada momento qué significado tienen las variables que están siendo introducidas.

3.1. Diagrama psicrométrico.

Primero definiremos el aire húmedo, el cual es una mezcla de aire seco y vapor de agua en la cual se trata el aire seco como si fuera un componente puro.

En los procesos de climatización e intercambio de calor, el diagrama psicrométrico es fundamental para la realización de los diversos cálculos y las variables que de él se obtienen son indispensables para nuestro programa de cálculo

Dicho diagrama es un conjunto de líneas y curvas que representan características del aire húmedo:

- *Temperatura de bulbo seco*: se refiere a la temperatura medida por un termómetro común colocado en la mezcla
- *Temperatura de bulbo húmedo*: es prácticamente igual a la de saturación adiabática. Se mide en un termómetro de bulbo húmedo, que es un termómetro ordinario de líquido en vidrio cuyo bulbo está rodeado por muselina humedecida en agua

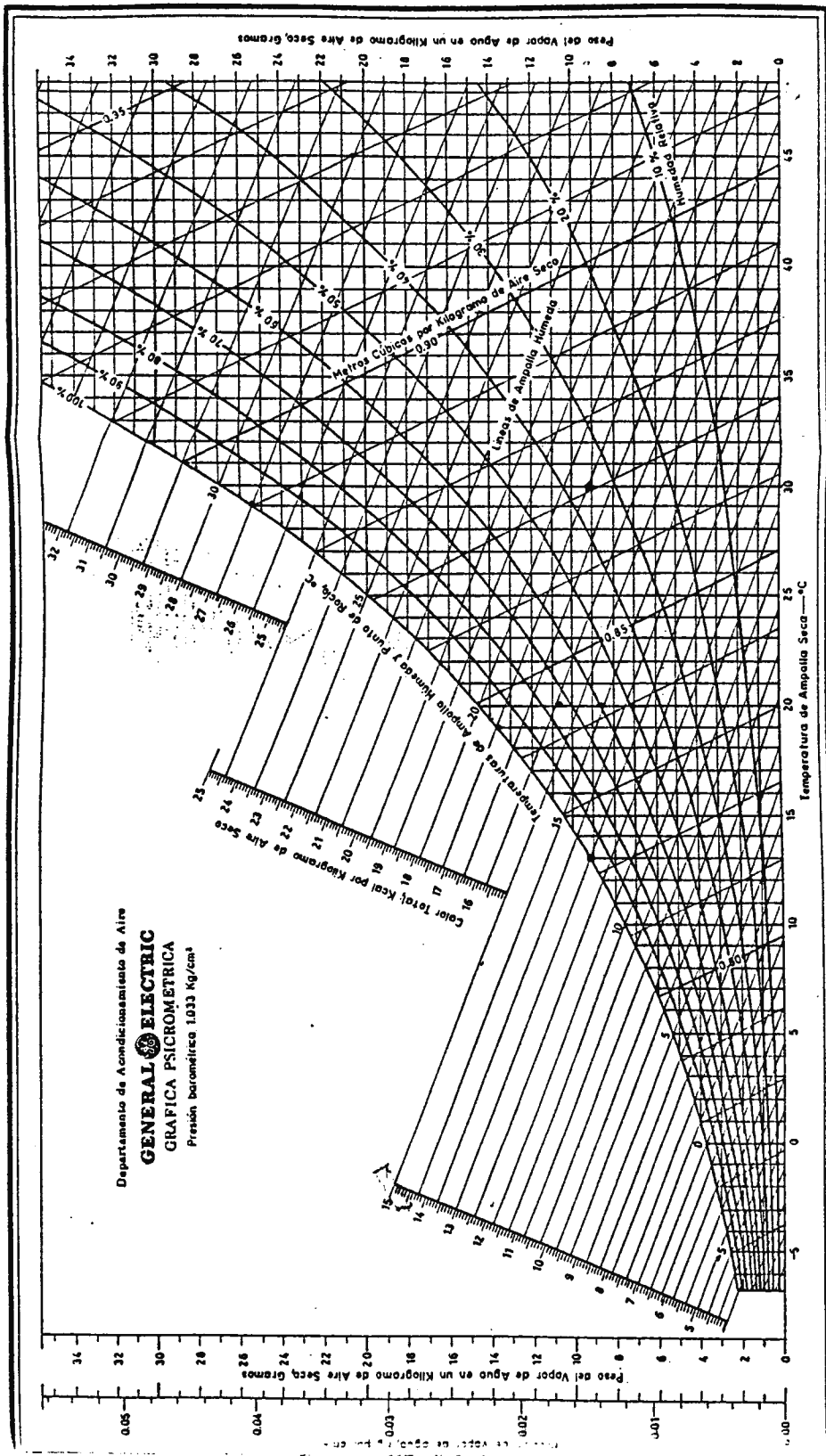
- *Humedad relativa*: es el cociente de la fracción molar del vapor de agua en una muestra de aire húmedo y la fracción molar en una muestra de aire húmedo saturado a la misma temperatura y presión de la mezcla.

- *Humedad absoluta* : es el cociente entre la masa de vapor y de agua.

- *Entalpía*:

- *Volumen específico*: es el inverso de la densidad.

Un ejemplo de diagrama psicrométrico se muestra a continuación:



El problema de automatizar el cálculo de las variables psicométricas se encuentra en que el diagrama esta basado en tablas (la presión de saturación del agua) y las curvas no son fácilmente aproximables por funciones sencillas.

Nuestro primer problema fue encontrar unas aproximaciones suficientemente buenas de las curvas del diagrama psicrométrico. Sin embargo, las aproximaciones nos aportan una ventaja bastante importante: el poder realizar los cálculos a varias presiones atmosféricas, cosa que antes, con el diagrama psicrométrico no podíamos realizar, ya que este es para una determinada presión, además de que el cálculo es más sencillo y rápido se dispone de un programa que lo efectúe comparado con el método tradicional de mirar en el diagrama.

Obtenemos además mayor fiabilidad en los cálculos ya que al ser el método tradicional un método gráfico y normalmente hay que hacer algún tipo de aproximación ya que no están resaltados todos los posibles puntos, se acumula un error que intrínsecamente a los métodos gráficos puede alcanzar un valor de hasta el 10%.

La aproximación del diagrama psicrométrico que conseguimos fue la siguiente:

Presión del agua pura saturada:

Para temperaturas comprendidas entre 0° y 100°

$$\ln(PWS) = \frac{C}{T} + C_2 + C_3T + C_4T^2 + C_5T^3 + C_6T^4 + C_7\ln(t)$$

Siendo C_i constantes con valores de:

$$C_1 = -5674.5359$$

$$C_2 = 6.3925247$$

$$C_3 = -0.9677843E-2$$

$$C_4 = 0.62215701E-6$$

$$C_5 = 0.20747825E-8$$

$$C_6 = -0.9484024E-12$$

Para temperaturas entre 0° y 200°:

$$\ln(PWS) = \frac{C}{T^8} + C_9 + C_{10}T + C_{11}T^2 + C_{12}T^3 + C_{13}\ln(T)$$

Con los valores de los coeficientes:

$$C_7=4.1635019$$

$$C_8=-5800.2206$$

$$C_9=1.3914993$$

$$C_{10}=-0.04860239$$

$$C_{11}=0.41764768E-4$$

$$C_{12}=-0.14452093E-7$$

$$C_{13}=6.5459673$$

Siendo T la temperatura a la que queremos calcular la presión de saturación, en °K y obteniendo PWS en Pascales.

Relacion de humedad en saturación.

$$WS = \frac{0.62198PWS}{P.PWS}$$

WS : gr. agua/Kg a.s.

P : Presión atmosférica

Humedad absoluta

$$W = \frac{(2501 - 2.381(tH - 273.15))WSH - (t - tH)}{(2501 + 1.805(t - 273.15) - 4.186(tH - 273.15))}$$

Siendo:

T : Temperatura seca

TH : Temperatura húmeda

WSH : Relación de humedad en saturación a la temperatura húmeda.

Grado de saturación

$$MU = \frac{W}{WS}$$

W: Humedad absoluta (Kg agua / Kg a.s.)

WS : Humedad absoluta de saturación a la temperatura
seca(Kg agua/Kg a.s.)

Temperatura de rocío

$$PW = \frac{P \cdot W}{0.62198 + W}$$

PW : Presión de saturación a la temperatura de rocío.

P : Presión atmosférica.

W: Humedad absoluta a la temperatura seca.

Humedad relativa(tanto por l)

$$FI = \frac{MU}{(1 - (1 - MU) \frac{PWS}{PA}}$$

MU : Grado de saturación.

PWS : presión de saturación a la temperatura seca.

PA : Presión atmosférica.

Volumen específico:

$$V = 0.287055 \cdot \frac{T}{P \cdot 100} \cdot (1 + 1.6078 \cdot W)$$

Siendo:

T : Temperatura seca(°K)

P : Presión (bar)

W: Humedad absoluta(Kg agua/Kg a.s.)

3.2 Intercambio de calor

Como he comentado anteriormente, el proceso de cálculo se basa en un intercambio de calor entre el aire húmedo y el fluido.

El proceso consiste básicamente en el enfriamiento de un fluido que pasa por unos tubos y transversalmente se disponen aletas por las que pasa el aire. Las aletas tienen el objeto de incrementar el área efectiva de transferencia de calor. La distribución de las aletas en los intercambiadores de calor de placas puede ser muy compleja. Los ingenieros de diseño han hecho alarde de gran ingenio para proporcionar aletas que incrementan el área de transferencia de calor sin caídas de presión excesivas y cuya fabricación resulta económica.

En nuestro caso tenemos un intercambiador de calor de una sola corriente y de flujo cruzado.

Los problemas usuales de diseño de intercambiadores de calor consisten en idear una unidad con un rendimiento dado en cuanto a

transferencia de calor ,sujeto a una serie de restricciones: (1) bajo coste, (2) bajo costo de operación, (3) limitaciones de tamaño, peso, ..., (4) facilidad de mantenimiento.

Sin embargo, un elemento de vital importancia dentro de los intercambiadores de calor es la pérdida de presión o pérdida de carga. Dentro de este aspecto hay que señalar que el flujo del fluido por los tubos depende de una serie de coeficientes, íntimamente relacionados con el número de Reynolds (nos indica la turbulencia del fluido), número de Nusselt y a partir de diversas consideraciones aplicamos las siguientes fórmulas para el cálculo de características importantes del proceso de diseño, que estudiaremos en el siguiente apartado.

3.3 Dinámica de Fluidos

Nos centramos en la forma de calcular la pérdida de carga o presión de nuestro fluido, velocidades del fluido,...

La pérdida de carga es fundamental, ya que de ella depende que la instalación deba ser suplementada por una bomba que le imprima la velocidad necesaria al fluido.

El programa calcula las siguientes características del sistema, basándose en fórmulas aproximadas y dependientes del tipo de batería escogido.

Velocidad del agua por los tubos:

$$V_w (m/s) = \frac{Q_w (m^3/h)}{n \cdot \frac{\pi \cdot D_i^2}{4} \cdot 3600}$$

Siendo las variables utilizadas:

Q_w : Caudal del aire por unidad.

n : Número de circuitos

D_i : Diámetro de los tubos

Pérdida de carga (calor):

$$\Delta H_{wR} = C1 \cdot V_w \cdot \frac{T \cdot R}{n} \cdot (A^n(m) + C2)$$

Donde:

V_w : Velocidad del fluido por los tubos.

T : Número de tubos.

R : Rango o número de filas de la batería.

A : Longitud de la batería.

$C1, C2$: Coeficientes relacionados con el tipo de batería.

Siendo T la altura del módulo, R el número de filas de la batería, y A la longitud del sistema (conjunto de módulos que forman el aero).

$$V_w = \frac{Q_w}{671 \cdot n}$$

Siendo Q_w el caudal de líquido por módulo y por unidad que en conjunto forman el aeroenfriador. La variable n indica el número de circuito que recorre el agua al meterse por los tubos.

3.4 Características de los Fluidos

Dentro de nuestro no debemos olvidarnos de características importantes del fluido que estamos tratando, ya que es una parte fundamental del proceso de cálculo que estamos realizando.

A continuación se indicarán las características de los fluidos que atañen al programa, junto con alguna definición o aclaración teórica de cada una.

Densidad: La densidad de un fluido es su masa por unidad de volumen. Varía ampliamente entre ellos. Para condiciones atmosféricas, la densidad atmosférica es de 1.22 kg/m^3 , la del agua es aproximadamente de 1000 kg/m^3 y la del mercurio es de 13500 kg/m^3 . Para un fluido concreto, su densidad varía con la presión y la temperatura. Esta variación con la temperatura se podrá observar en el programa en el apartado de **Fluido**.

Calor específico: La relación entre la energía interna o entalpía de un fluido y su temperatura del mismo se expresa en función del calor específico. Los calores específicos importantes son el *calor específico a temperatura constante* (c_v) y el *calor específico a presión constante* (c_p), los cuales se definen como:

$$c_v = \frac{\partial \tilde{u}}{\partial T_v} \quad \text{y} \quad c_p = \frac{\partial \tilde{h}}{\partial T_p}$$

Viscosidad: Un fluido es una sustancia que sufre una deformación continua cuando se somete a un esfuerzo cortante, y éste es función de la velocidad de deformación. Para la mayoría de los fluidos, este esfuerzo es proporcional a la velocidad de deformación. La constante de proporcionalidad se denomina viscosidad y para calcularla se usa la *ley de Newton de la viscosidad* (para los fluidos denominados newtonianos). La viscosidad de los fluidos varía considerablemente entre ellos. Depende notablemente de la temperatura, pero no tanto de la presión. Esta característica de los fluidos influye en la velocidad que tiene el fluido en un determinado flujo.

Conductividad Térmica: Es la cantidad de calor que pasa por unidad de tiempo a través de una pared de superficie y espesor unidad cuando la diferencia de temperaturas entre las caras es también la unidad. (kcal/h.m.°C).

4.FASES DEL PROYECTO

En este apartado iré explicando las fases de aprendizaje y desarrollo del proyecto.

4.1. Repaso termodinámico.

Como he indicado anteriormente, el proyecto está pensado para ser utilizado de forma bastante específica, con lo cual, la materia en que se basa la programación es también específica.

De esta forma, antes de comenzar con la programación fue necesario un acercamiento a las variables termodinámicas que atañen al desarrollo de la aplicación.

Como he explicado anteriormente, los temas importantes dentro de este apartado son el diagrama psicrométrico, intercambios de calor, características y dinámica de fluidos.

4.2. Aprendizaje de Visual Basic

Este lenguaje de programación que ya he explicado anteriormente es bastante intuitivo. Sin embargo, como todos los lenguajes de programación puede llegar a desarrollar aplicaciones de gran importancia y envergadura.

Para el programa era fundamental el desarrollo de la parte de Visual Basic que está orientada a objetos, ya que las conexiones con Access y con Word se realizan creando objetos del tipo de estos dos programas.

4.3. Conexiones de Visual Basic con bases de datos

Ha sido sin duda alguna la parte de programación más complicada de todo el proyecto, precisamente por la dificultad de aprendizaje.

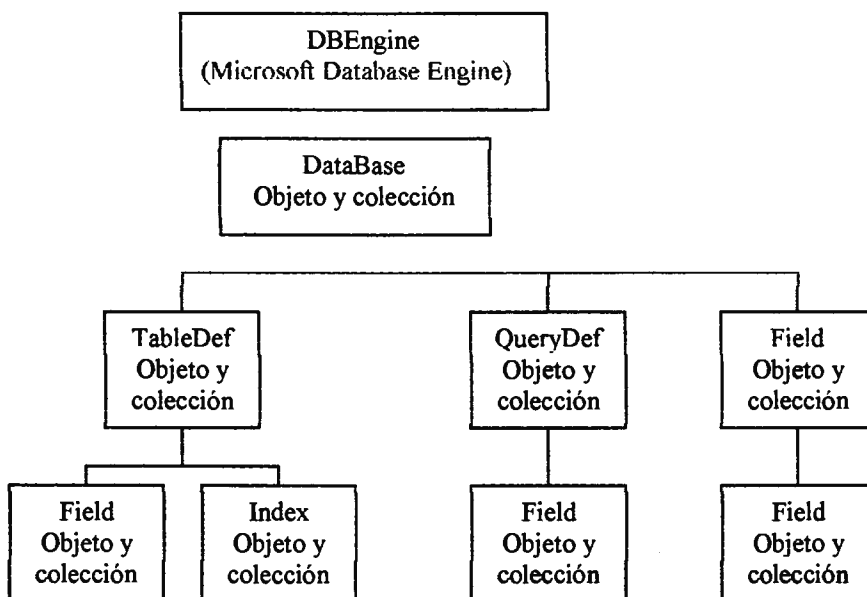
Sin embargo, Microsoft tiene diseñados todos sus programas y sistemas para que la comunicación DAO, OLE entre ellos sea bastante eficaz.

Así, en Visual Basic disponemos de varias formas para realizar la conexión:

- *ControlData*: Es un control especial con el que podemos crear un objeto que represente la base de datos. Su logotipo es:

Esta forma de conexión es similar a la segunda, si bien no es demasiado buena en tiempo de ejecución.

Objeto DataBase: a continuación se muestra un árbol de la jerarquía de objetos de Access:



- *DataBase* : Representa una base de datos abierta en el área de trabajo. La colección *DataBase* se refiere a todos los objetos *Database* abiertos en el área de trabajo.

- *TableDef* : Representa la definición de una tabla del objeto *DataBase*. Un *TableDef* es un contenedor *Field* (Campo) e *Index* (Índice) de la tabla. La colección *TableDefs* se refiere a todas las definiciones de tabla de un objeto *DataBase*.

- *QueryDef*: Representa la definición de una consulta en un objeto DataBase. Un QueryDef es un contenedor de los objetos Field de la consulta. La colección QueryDefs se refiere a todas las definiciones de consulta del objeto DataBase.

- *Recordset*: representa o bien los registros de una tabla de base de un objeto Database o bien los registros resultantes de una consulta. La mayoría de los trabajos de DAO entrañarán la manipulación de objetos Recordset de una manera u otra. La colección Recordset se refiere a todos los objetos Recordset abiertos en un objeto Database.

- *Field*: Un objeto Field representa una columna de datos. Para un objeto TableDef o QueryDef, un objeto Field contiene las especificaciones de un campo utilizado en una tabla o consulta. Para un objeto Recordset, un objeto Field representa el valor de un campo en particular para el registro activo. La colección Fields se refiere a todos los campos de un objeto TableDef, QueryDef o Recordset.

- *Index*: Un objeto Index especifica un orden para los registros en un objeto TableDef. La colección Indexes se refiere a todos los objetos Index almacenados en un objeto TableDef.

Aunque es fácil acceder a otras bases de datos como FoxPro, Dbase,..., actualmente, Access se encuentra en auge por su gran potencial en lo referente a acceso a datos y presentación de los mismos.

4.4. Conexiones de Vb con Microsoft Word

Continuamos con las comunicaciones del programa con otras aplicaciones de Microsoft.

La forma de conexión con Word es similar a la de Access, dentro de la programación de objetos.

Los objetos de las aplicaciones de Office están organizados en una jerarquía. El objeto más general –el objeto Application – se refiere al programa. Los siguientes objetos serán importantes para el desarrollo del proyecto:

- *Objeto Document*: En Word aparece directamente debajo del objeto Application en la jerarquía de objetos. Tenemos también la colección Documents que se refiere a todos los documentos que estén activos. Dentro de este objeto disponemos de varias propiedades importantes, aunque no serán objeto de nuestro estudio, que conviene resaltar, como *Add* (crear un documento nuevo), *GrammarChecked* (revisa la ortografía del documento), *Name* (devuelve el nombre del documento). Los métodos de este objeto Document son *Activate*

(activa el documento solicitado), *Checkspelling* (revisa la ortografía del documento especificado), *Close* (cierra el documento indicado), *PrintOut* (imprime el documento especificado), *SaveAs* (guarda el documento indicado con el nombre introducido).

- *Objeto Range*: Word no dispone de objetos separados para sus importantes unidades de texto: el carácter y la palabra. Consigue estos elementos como instancias de una clase genérica: el objeto *Range*, el cual se define como una sección contigua de texto en un documento, por lo que puede ser desde un carácter hasta un documento entero. Existen dos métodos para devolver un objeto *range*: el método *Range* y la propiedad de *Range* del objeto *Document*. Dentro del objeto *Range* tenemos varias propiedades a reseñar como *Characters* (devuelve los caracteres del objeto), *End* (devuelve la posición del último carácter), *Font* (devuelve o establece un objeto *Font* que especifica el formato a usar en el rango), *Text* (devuelve o establece el texto del rango)

- *Objeto Characters* : es una colección que representa todos los caracteres de cualquier objeto seleccionado, con lo cual el usuario se debe referir a caracteres individuales mediante un índice.

- *Objeto Words*: es una colección que representa todas las palabras de cualquier objeto especificado. El usuario hace referencia a palabras individuales mediante un índice.

- *Objeto Paragraph* : De caracteres y palabras, pasamos al fragmento de texto: párrafo. Un objeto Paragraph es un miembro de la colección Paragraphs, que representa todos los párrafos de un documento. El usuario lo selecciona mediante un índice.

5. ESTRUCTURA DEL PROYECTO

5.1 Proceso de cálculo

Pasamos a comentar más en profundidad algunos aspectos importantes del programa de cálculo.

El proceso de cálculo consiste en hallar la batería necesaria para las especificaciones del cliente. Éste nos indicará los valores de las entradas siguientes:

- *Potencia* : Potencia a disipar por la batería.
- *Te* : Temperatura del fluido a la entrada de los tubos del aerofriador.
- *Tu* : Temperatura del fluido tras pasar por los tubos del aerofriador.
- *Q_w* : Caudal del fluido por los tubos.
- *P* : Presión atmosférica a la que se encuentra el lugar donde se ubicará el aerofriador.

- TS : Temperatura de bulbo seco del ambiente en el que se instalará el sistema.

- HR, TH, X : Tercera variable para el cálculo del psicrométrico, que puede ser la Humedad Relativa, la Temperatura Húmeda o la Humedad absoluta del lugar donde se instalará la batería.

Existen también otras variables de entrada que a juicio del usuario del programa, serán escogidas para desarrollar el tipo de aeroenfriador adecuado, como son:

- M : Margen de sobredimensionamiento, es decir, una especie de margen de seguridad para el aeroenfriador.

- $FC2$: Coeficiente necesario, ya que las fórmulas están diseñadas para el agua, de tal forma que al usarse habitualmente glicoles, por ejemplo, hay que adaptar sus características a las fórmulas del agua.

- *Fluido*: Fluido que será objeto del enfriamiento.

El desarrollo del cálculo será explicado a continuación:

Empezaremos por la parte del fluido:

Temperatura media del fluido:

$$T_{mw} = \frac{T_e + T_u}{2} \quad (1)$$

Balance de potencia del fluido:

$$P = Q_w \cdot \rho \cdot c_p \cdot \Delta T_w \quad (2)$$

Donde:

$$\rho = \rho(T_{mw}) \text{ y } C_p = C_p(T_{mw})$$

Continuando con el aire húmedo, debemos escoger el diámetro del ventilador que impulsará el aire, el tipo de batería y el rango de la misma (R). Con estos datos obtendremos:

$$(\Delta T / \Delta T_m)_{máx} \quad Q_i \text{ (caudal de aire por cada ventilador)}$$

Con las fórmulas aproximadas del diagrama psicrométrico conseguimos el resto de las variables del aire húmedo a la entrada del aeroenfriador, a las que denominaré con una “'”.

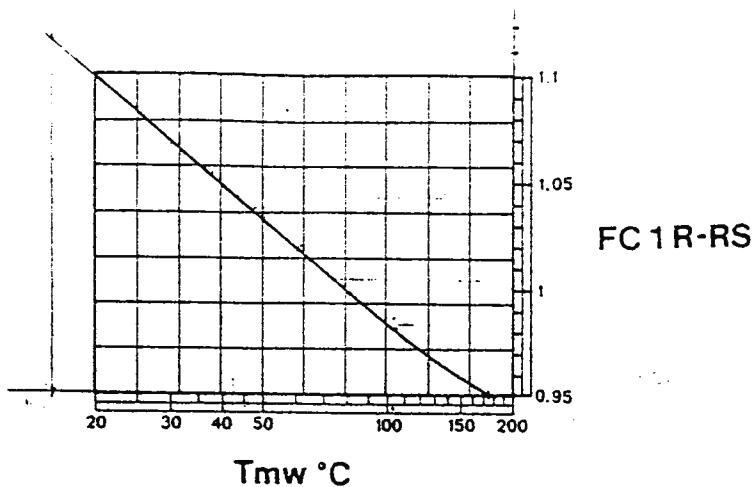
Debemos conocer las características de salida del aire húmedo. El proceso de intercambio de calor se realiza a *Humedad absoluta constante* y a *Presión constante*.

Para conocer la temperatura de bulbo seco de salida aplicaremos la fórmula del incremento de temperatura máximo:

$$\frac{\Delta T}{\Delta T_m} = \frac{TS'' - TS'}{(Te - TS'') - (Tu - TS')} \quad (3)$$

$$Ln \left[\frac{(Te - TS'')}{(Tu - TS')} \right]$$

Donde obtenemos $\Delta T/\Delta T_m$ con el valor de $(\Delta T/\Delta T_m)_{max}$ y un coeficiente que denominamos FC1 calculado mediante el gráfico:



$$\frac{\Delta T}{\Delta T_m} = \frac{\left(\frac{\Delta T}{\Delta T_m} \right)_{m\acute{a}x}}{FC1 \cdot FC2} \quad (4)$$

Como el intercambio de calor se realiza a humedad absoluta constante y a la misma presión atmosférica, tenemos ya las tres variables necesarias para proseguir con el cálculo del psicrométrico de las condiciones de salida.

Denominando como J a la entalpía y v al volumen específico del aire, continuamos con el cálculo de la *Potencia*:

$$P_i = Q_i \cdot \frac{\Delta J}{v} \quad (5)$$

Siendo ΔJ el incremento de entalpía entre la entrada y la salida.

Proseguimos con la ecuación del número de ventiladores necesarios:

$$n^\circ \text{ ventiladores} = \frac{\text{Potencia} \cdot (1 + 0.01) \cdot M}{P_i} \quad (6)$$

Al ser el usuario un experto en aeroenfriadores, deberá escoger el número adecuado de unidades y ventiladores basándose en el cálculo anterior, siempre que el número de ventiladores escogido se encuentre en la base de datos. Sin embargo, el programa indicará al usuario si el número no está en la base de datos.

El margen real de dimensionamiento será:

$$M_{real} = \frac{n^{\circ} unidades \cdot n^{\circ} ventiladores_escogido}{n^{\circ} ventiladores_calculado} \quad (7)$$

La siguiente etapa comienza por:

$$n^{\circ} circuitos \cdot n^{\circ} tubos / circuito = T \cdot R \quad (8)$$

La superficie de la batería la calcularemos como:

$$S = C \cdot \frac{T \cdot R \cdot A''}{paso} \quad (9)$$

Siendo :

A'' : Longitud de la batería.

$Paso$: Paso entre las aletas.

C : Coeficiente característico del tipo de batería.

El volumen interno:

$$V_{int} = 1.05 \cdot C_1 \cdot T \cdot R \cdot (A'' + C_2) \quad (10)$$

Las fórmulas de la velocidad del agua por los tubos y de las pérdidas de carga han sido ya expuestas en el apartado de *dinámica de fluidos*.

Velocidad del fluido por los colectores:

$$V_{wcol} = \frac{Q_w}{\frac{\pi \cdot D_{col}^2}{4} \cdot 3600} \quad (11)$$

Una mención especial merece la forma de calcular algunas temperaturas. Para calcular su valor disponemos de la presión de saturación, que a una determinada presión corresponde unívocamente a una temperatura. Al tener una expresión complicada de cálculo, ya que no es un simple polinomio (dispone de logaritmos). Escogí para realizarlo el método de la bisección, en un intervalo donde sabemos que se va a encontrar la solución.

En el programa se puede comprobar que la aproximación del cálculo a los valores reales es buena. Sin embargo, en los puntos extremos las funciones dejan de tener la forma que aproximamos, por esto mismo es importante resaltar que las aproximaciones son correctas para temperaturas mayores de -100°C y menores de 70°C .

Es también resaltable la parte del programa referente a los fluidos. Consta de varias opciones:

- *Nuevo* : Para crear un nuevo fluido. El programa aproxima cada característica por una función que pasa por los puntos determinados por el usuario y almacena los coeficientes del polinomio en la base de datos.

- *Borrar* : Borra el fluido seleccionado por el usuario.

- *Ver* : Permite ver las características de un fluido determinado en un rango de temperaturas.

5.2 Bases de datos utilizadas

5.2.1 Fluidos

Es el que guardamos el nombre del fluido y sus características .
Por cada fluido se guardan tres términos en la base de datos. Pueden ser los coeficientes de una recta de valor constante, una recta con pendiente y una parábola, dependiendo de los puntos que haya introducido el usuario.

De esta forma, aplicando las ecuaciones correspondientes, podremos conocer el valor de las propiedades del fluido a una temperatura cualquiera.

La base de datos de fluidos está abierta a nuevos posibles fluidos. Consta de una tabla denominada *Tablafluidos* que se muestra a continuación:

fluido	a0d	a1d	a2d	a0cp	a1cp	a2cp
ethglycol20	1034,1	-,215	-2E-03	,9254	2,1E-04	1,4E-06
ethglycol30	1057	-,5		0,8751	6,05E-04	-5E-07

a0c	a1c	a2c	a0v	a1v	a2v	ldfluido
,4371875	,0006325	-,0000023	2,389237	-,04089	,0002	31
,415	2,65E-04	-1,5E-06	2,1325	-0,03E-02	1,2E-04	33

Donde los coeficientes de segundo orden son los de índice 2, los de primero son los de índice 1 y el término independiente, el de índice 0.

Las letras “d”, “cp”, “c” y “v” de los coeficientes corresponden a densidad, calor específico, conductividad térmica y viscosidad.

5.2.2 Aeros

Contiene información necesaria sobre algunas características de los módulos comercializables de los aeros de la empresa. Esta base de datos es sólo de consulta, si bien la base de datos está abierta, es decir, se pueden introducir nuevos tipos de baterías con nuevas características y a su vez retirar posibles tipos que se hayan quedado obsoletos. Tiene información imprescindible sobre tipos de baterías, número de líneas, etc., para realizar correctamente el cálculo.

Dispone de varias tablas que mostraremos a continuación:

Tabla diámetros:

Iddiámetro	diámetro	Potencia	revoluciones
1	700	1.1 / 0.55	950 / 720
2	450	0.64 / 0.51	1370 / 1060
3	800	1.4 / 0.81	870 / 620

Donde los campos existentes tienen el significado:

1. Iddiámetro : Identificador del diámetro que se ha escogido. Es la clave principal de la tabla.

2. Diámetro : Diámetro del motoventilador.

3. Potencia : Potencia del módulo.

4. Revoluciones: Revoluciones a las que gira el
motoventilador.

Tabla tipobat :

klbat	nombreat	coef1	coef2	coef3	coef4	paso
1	2.1 SIMPLE	,18625	1,775	,43	,32625	2,1
2	2.1 DOBLE	,18625	1,775	,43	,32625	2,1
3	3.2 SIMPLE	,18625	1,775	,43	,32625	3,2
4	3.2 DOBLE	,18625	1,775	,43	,32625	3,2
5	NH5 SIMPLE	,06642	1,75	,62	,11644	5
6	NH5 DOBLE	,06642	1,75	,62	,11644	5
7	BCB SIMPLE	,081	1,75	,56	,142	2
8	BCB DOBLE	,081	1,75	,56	,142	2
9	BCO SIMPLE	,081	1,75	,56	,142	3
10	BCO DOBLE	,081	1,75	,56	,142	3

numerobat	tubosaletas	Dtubos	coefS	c1vint	c2vint
3012	Cu/Al	,0119	1,56	,11122	,07
3012	Cu/Al	,0119	1,56	,11122	,07
3012	Cu/Al	,0119	1,56	,11122	,07
3012	Cu/Al	,0119	1,56	,11122	,07
6020	Fe/Fe	,017	7,2	,227	,2
6020	Fe/Fe	,017	7,2	,227	,2
6016	Cu/Al	,0154	3,6	,1865	,14
6016	Cu/Al	,0154	3,6	,1865	,14
6016	Cu/Al	,0154	3,6	,1865	,14
6016	Cu/Al	,0154	3,6	,1865	,14

Donde los campos son:

- **Idbat** : identificador del tipo de batería escogido.
- **nombreat** : denominación de las baterías.
- **coef1-coef4** : coeficientes para el cálculo de la pérdida de carga.
- **paso** : distancia entre las aletas.
- **Numerobat** : número que denomina las baterías.
- **tubosaletas** : material en que están construidos ls tubos y las aletas.
- **Dtubos** : diámetro de los tubos por los que fluye el fluido.
- **CoefS** : coeficiente necesario para calcular la superficie.
- **c1vint, c2vint**: coeficientes necesarios para calcular el volumen interno.

Tabla tipobatdiam:

<i>I</i> bat	<i>I</i> batdiam	<i>I</i> ddiametro	<i>T</i>
1	1	3	56
2	2	3	72
3	3	3	56
4	4	3	72
5	5	1	20
6	6	1	20
5	7	2	10
6	8	2	10
7	9	1	20
8	10	1	20
7	11	2	10
8	12	2	10
9	13	1	20
10	14	1	20
9	15	2	10
10	16	2	10

Siendo:

- *I*bat : Identificador de la batería.
- *I*ddiametro : Identificador del diámetro.
- *I*batdiam: Identificador de las posibles combinaciones de los diámetros y baterías.
- *T* : Número de tubos.

Tabla datos:

Idbatdiam	numero líneas	caudal	DT/DTm	Iduno
1	3	17800	1,18	81
1	4	16800	1,64	82
1	6	15300	2,6	83
2	3	15400	1,06	84
2	4	14200	1,5	85
3	2	19700	,5	86
3	3	18800	,79	87
3	4	18000	1,09	88
3	6	16800	1,7	89
4	3	16900	,68	90
4	4	15900	,95	91
4	6	14400	1,5	92
5	3	14250	,42	29
5	4	13750	,6	30
5	5	13375	,79	31
5	6	13000	,9	32
5	7	12750	1,13	33
5	8	12350	1,3	34
6	3	14250	,42	35
6	4	13750	,6	36
6	5	13375	,79	37
6	6	13000	,9	38
6	7	12750	1,13	39
6	8	12350	1,3	40
7	3	5900	,43	69
7	4	5700	,6	70
7	5	5600	,79	71
7	6	5400	,95	72
7	7	5300	1,13	73
7	8	5100	1,3	74
8	3	5900	,43	75
8	4	5700	,6	76
8	5	5600	,79	77
8	6	5400	,95	78
8	7	5300	1,13	79
8	8	5100	1,3	80
9	3	13600	,76	1
9	4	13125	1,05	2
9	5	12650	1,3	3
9	6	12300	1,6	4
9	7	12000	1,9	5
9	8	11500	2,2	6

9	9	11250	2,5	7
10	3	13600	,76	8
10	4	13125	1,05	9
10	5	12650	1,3	10
10	6	12300	1,6	11
10	7	12000	1,9	12
10	8	11500	2,2	13
10	9	11250	2,5	14
11	3	5700	,77	41
11	4	5500	1,05	42
11	5	5300	1,3	43
11	6	5100	1,6	44
11	7	5000	1,95	45
11	8	4900	2,2	46
11	9	4800	2,5	47
12	3	5700	,77	48
12	4	5500	1,05	49
12	5	5300	1,3	50
12	6	5100	1,6	51
12	7	5000	1,95	52
12	8	4900	2,2	53
12	9	4800	2,5	54
13	3	14300	,5	15
13	4	14000	,65	16
13	5	13800	,85	17
13	6	13400	1	18
13	7	13000	1,25	19
13	8	12800	1,45	20
13	9	12500	1,65	21
14	3	14300	,5	22
14	4	14000	,65	23
14	5	13800	,85	24
14	6	13400	1	25
14	7	13000	1,25	26
14	8	12800	1,45	27
14	9	12500	1,65	28
15	3	5900	,5	55
15	4	5800	,68	56
15	5	5600	,86	57
15	6	5500	1,05	58
15	7	5400	1,28	59
15	8	5300	1,45	60
15	9	5200	1,67	61
16	3	5900	,5	62
16	4	5800	,68	63
16	5	5600	,86	64

16	6	5500	1,05	65
16	7	5400	1,28	66
16	8	5300	1,45	67
16	9	5200	1,67	68

- **Idbatdiam** : Explicado en la tabla anterior.
- **numerolíneas** : Rango de la batería o número de filas.
- **caudal** : Caudal de aire que pasa por los ventiladores.
- **DT/DTm** : incremento de temperatura máximo.
- **Iduno** : Clave principal.

Tabla módulos:

Id módulos	numeromódulos	Idbatdian	Asegunda
1	1	9	1020
2	2	9	2040
3	3	9	3060
4	4	9	4080
5	5	9	5100
6	6	10	3060
7	8	10	4080
8	10	10	5100
9	12	10	6120
10	4	10	2040
11	1	13	1020
12	2	13	2040
13	3	13	3060
14	4	13	4080
15	5	13	5100
16	6	14	3060
17	8	14	4080
18	10	14	5100
19	12	14	6120
20	4	14	2040
21	1	5	1020
22	2	5	2040
23	3	5	3060
24	4	5	4080
25	5	5	5100
26	6	6	3060
27	8	6	4080
28	10	6	5100
29	12	6	6120
30	4	6	2040
31	1	15	850
32	2	15	1700
33	3	15	2550
34	4	15	3400
35	5	15	4250
36	6	16	2550
37	8	16	3400
38	10	16	4250
39	12	16	5100
40	4	16	1700
41	1	11	850
42	2	11	1700

43	3	11	2550
44	4	11	3400
45	5	11	4250
46	6	12	2550
47	8	12	3400
48	10	12	4250
49	12	12	5100
50	4	12	1700
51	1	7	850
52	2	7	1700
53	3	7	2550
54	4	7	3400
55	5	7	4250
56	6	8	2250
57	8	8	3400
58	10	8	4250
59	12	8	5100
60	4	8	1700
61	2	1	2450
62	3	1	3700
63	4	1	4950
64	5	1	6200
65	4	2	2740
66	6	2	4160
67	8	2	5580
68	10	2	7000
69	2	3	2450
70	3	3	3700
71	4	3	4950
72	5	3	6200
73	4	4	2740
74	6	4	4160
75	8	4	5580
76	10	4	7000

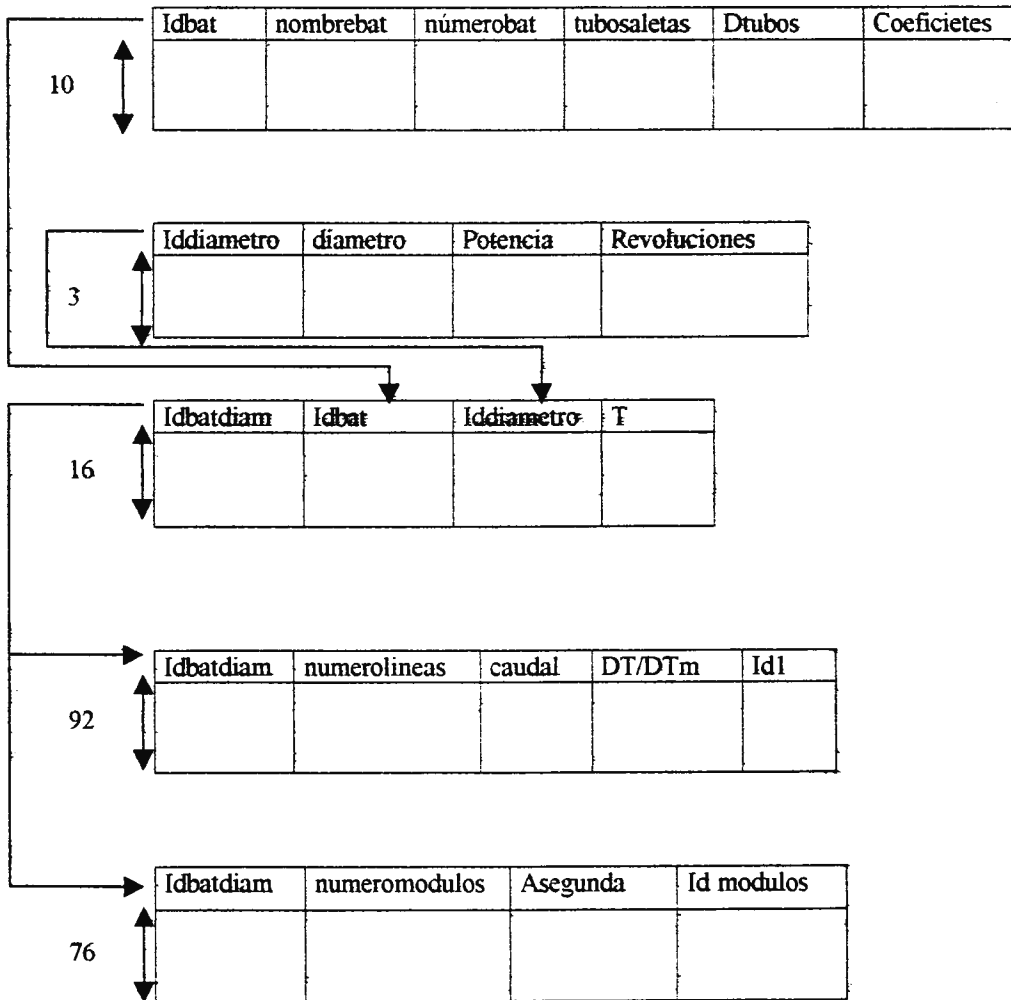
Los campos de la tabla son:

- **Id modulos** : identificador del elemento de la tabla.
- **Numeromodulos** : Número de módulos posibles para el tipo de batería.
- **Idbatdiam** : Identificador ya comentado anteriormente.
- **Asegunda**: Longitud de la batería.

Tabla colectores:

Idcolector	diamcolector	DN	denominacion
1	,14	DN125	5"
2	,115	DN100	4"
3	,09	DN80	3"
4	,077	DN65	2"1/2
5	,06	DN50	2"
6	,048	DN40	1"1/2
7	,033	DN25	1"
8	,027	DN20	3/4"
9	,0213	DN15	1/2"

Las relaciones entre las tablas de la base de datos *Aeros* son:



5.2.3 *Bclientes*

Es la base de datos que guarda los resultados de un cálculo, junto con las variables de entrada y el cliente y el proyecto correspondiente a dicho cálculo.

Esta base de datos por lo tanto será de escritura y de lectura y debe tener capacidad suficiente para registrar todo el conjunto de cálculos y ofertas correspondientes a un *item* (nombre técnico del campo en la base de datos).

Tiene las tablas:

Tablacliente: Tabla en la que se guarda el nombre del cliente, la dirección, teléfono, etc. y se consigue un índice para ese cliente.

Tablaproyecto: En esta tabla se guarda el nombre del proyecto, el índice del cliente al que pertenece y un índice para el proyecto.

Tablaitem: Almacenamos el nombre del ítem, el índice del cliente y del proyecto al cual pertenece el cálculo y los resultados del proceso de cálculo.

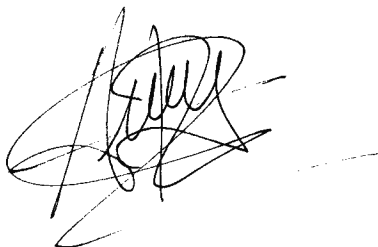
Tusuario: Almacena los datos de los distintos usuarios del programa.

6. VALORACIÓN ECONÓMICA

El precio total de desarrollo de este proyecto asciende a **ocho millones, doscientas cuarenta y nueve mil, setenta y siete pesetas.**

Madrid, a 23 de Agosto de 1998

Firmado: Amelia Riestra López



7.BIBLIOGRAFÍA

1.MORAN-SHAPIRO.

Fundamentos de Termodinámica Técnica.

Volumen II.

2.GARY CORNELL.

Manual de Visual Basic 4 para Windows 95.

Ed. Mcgraw/Hill.

3.MICROSOFT.

Manual de usuario de Access.

4.PAUL McFREDIES

VBA 97

Ed. Prentice Hall

5.GERHART, GROSS, HOCHSTEIN

Fundamentos de Mecánica de Fluidos.

Ed. Addison- Westley Iberoamericana.

6.A.F. MILLS.

Transferencia de Calor.

Ed. Irwin.

ANEJO 1

Dentro de este apartado debemos comenzar por la instalación del programa. Debido a la comunicación con bases de datos y Word, el programa debe buscar unos determinados ficheros en unos directorios escogidos en el tiempo de diseño, con lo cual es fundamental para un buen funcionamiento del cálculo que los siguientes ficheros se encuentren en el lugar correcto:

Bases de datos :

Aeros.mdb, Bclientes.mdb, Fluidos.mdb

Documentos de Word y plantillas:

Offerta.doc, Oferta1.doc, Offerta.dot deben estar en:

C:\Aeros

El programa de instalación es muy sencillo de utilizar, sin embargo he de señalar algo importante referido de nuevo a los ficheros anteriores.

El programa de instalación nos pregunta donde queremos que sea instalado el programa aeros. Tenemos por defecto el directorio “C:\Archivos de Programa\Aeros”, pero para una mayor facilidad se debe cambiar a “***C:\Aeros***”, ya que en caso de que no se haga, los archivos

mencionados anteriormente deberán ser copiados en el directorio
“C:\Aeros”.

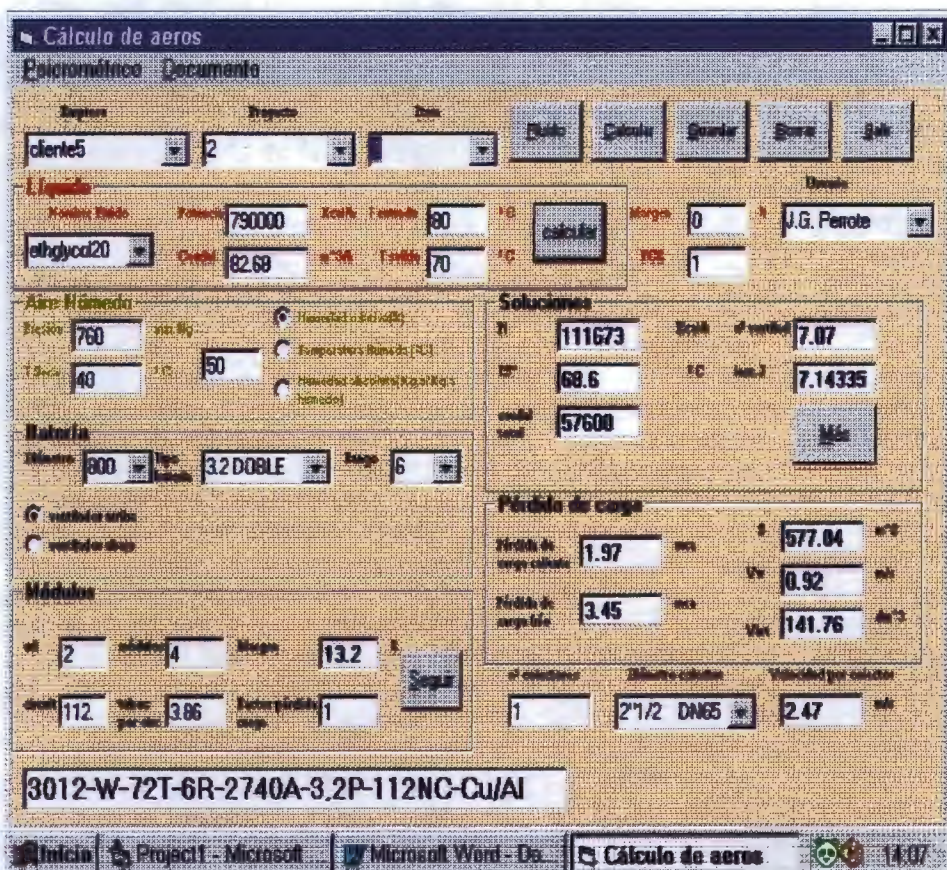
UTILIZACIÓN DEL PROGRAMA

Si estamos utilizando directamente el ejecutable, se ejecutará con hacer doble click en el icono *aeros.exe*. Si tenemos la versión instalada, se podrá ejecutar también como un programa de microsoft, es decir, en el apartado programas.

Al cargarse la pantalla del programa, los combos se van rellenando con la información de las bases de datos. Al terminar el suceso denominado *Form_Load* tendremos la pantalla que aparece en la página siguiente.

El próximo paso será decidir para qué quiere usar el programa.

Si la finalidad es realizar un nuevo cálculo, podrá escoger entre un nuevo cliente, un nuevo proyecto dentro de un cliente ya existente o un item o cálculo nuevo dentro de un cliente y proyecto ya existentes. Esto se escogerá mediante los combos superiores izquierdos con las etiquetas de cliente, proyecto, item. En realidad, lo que denominamos item es el cálculo en sí de un aeroenfriador que cumple las especificaciones de un proyecto escogido dentro de la demanda de un cliente determinado.



Como se ha podido apreciar, la pantalla tiene unas partes claramente diferenciadas. La primera de ellas es la parte de **Líquido**.

Líquido

En esta parte se desarrolla el balance de energía del fluido.

Deben ser introducidos tanto el nombre del fluido, pinchando en algunos de los nombres del combo, como tres de las cuatro variables del fluido para que con la cuarta se cierre el balance. El cálculo puede ser orientativo, es decir, una vez calculada una variable puede probar a cambiar cualquiera de las otras tres, pero sabiendo que deben ser introducidas tres de ellas, de forma que si no hay texto en dos de ellas el programa se lo indicará, al igual que lo hará si tiene cuatro introducidas y aprieta el botón *Calcular*.

Cuando ya tiene determinadas todas las variables del fluido, debe escoger el **Margen de sobredimensión** y **FC2**, que como ya expliqué anteriormente se tratan de un coeficiente de seguridad y un factor para adaptarse a las fórmulas del proceso que está referidas al agua.

Al hacer click en el botón de *Calcular* de la parte de la pantalla superior derecha, comenzará el proceso de cálculo de las variables de salida del aire, del número de módulos necesarios, etc., que vendrán indicados en la parte central derecha en la parte de *Soluciones*, donde tenemos:

Potencia disipada por módulo, Temperatura de salida del aire, número de ventiladores necesarios y el incremento de entalpía.

Como el usuario sabe lo que está utilizando, basándose en el número de ventiladores necesarios, pasará a calcular la siguiente parte:

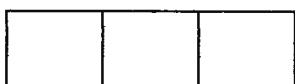
Módulos

Debe introducirse el *número de unidades*, el número de módulos o ventiladores, y o bien el número de circuitos o el número de tubos por circuito, de forma que al escribir uno, automáticamente se tenga el otro.

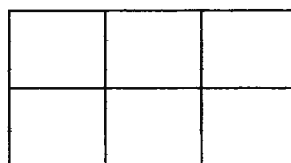
Al escribir el número de módulos, se buscará en la base de datos para saber si la batería escogida puede formar ese número de módulos. Hay que tener en cuenta que las batería pueden ser simple o dobles, de

forma que una doble, el número de módulos debe ser siempre par, ya que la disposición es de la forma:

Simple:



Doble:

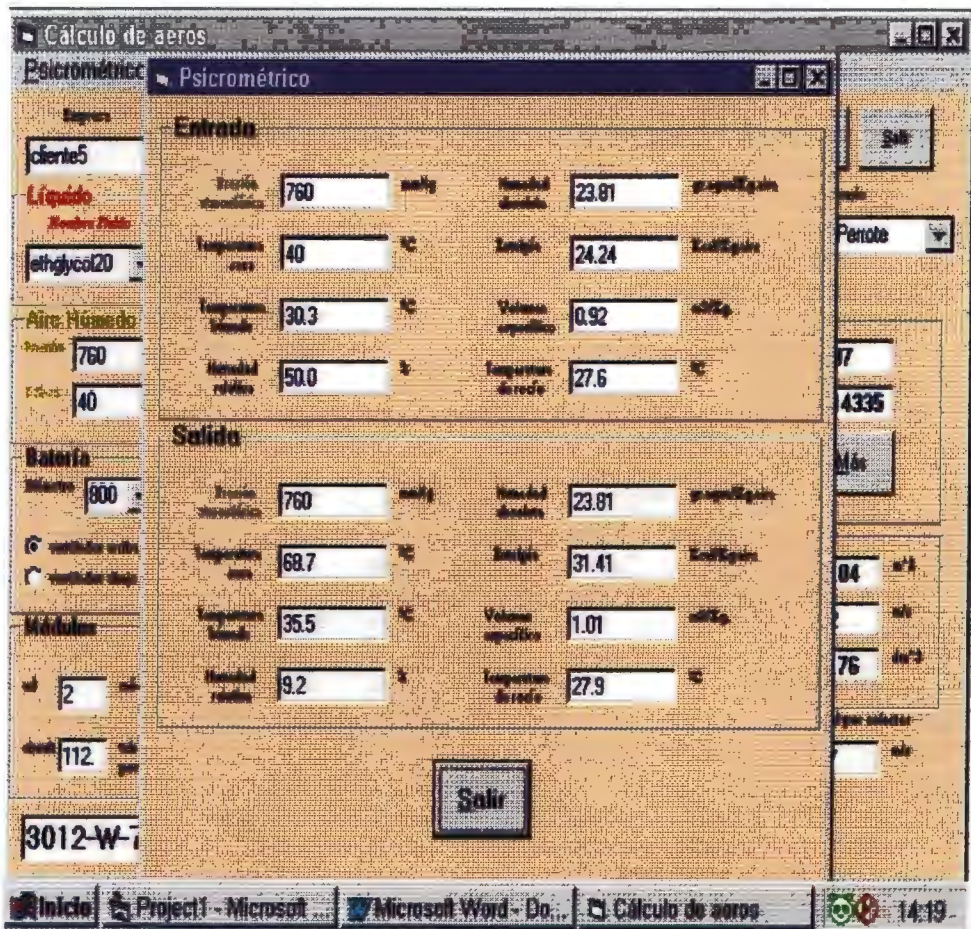


Si el número introducido no está en la base de datos, el programa mostrará un mensaje. Si existe, aparecerá junto a las anteriores soluciones una indicando el caudal total.

Al apretar el botón calcular de esta parte, obtendremos las siguientes soluciones de la pérdida de carga, margen real,...

Podremos también saber más características del cálculo apretando el botón *Más*.

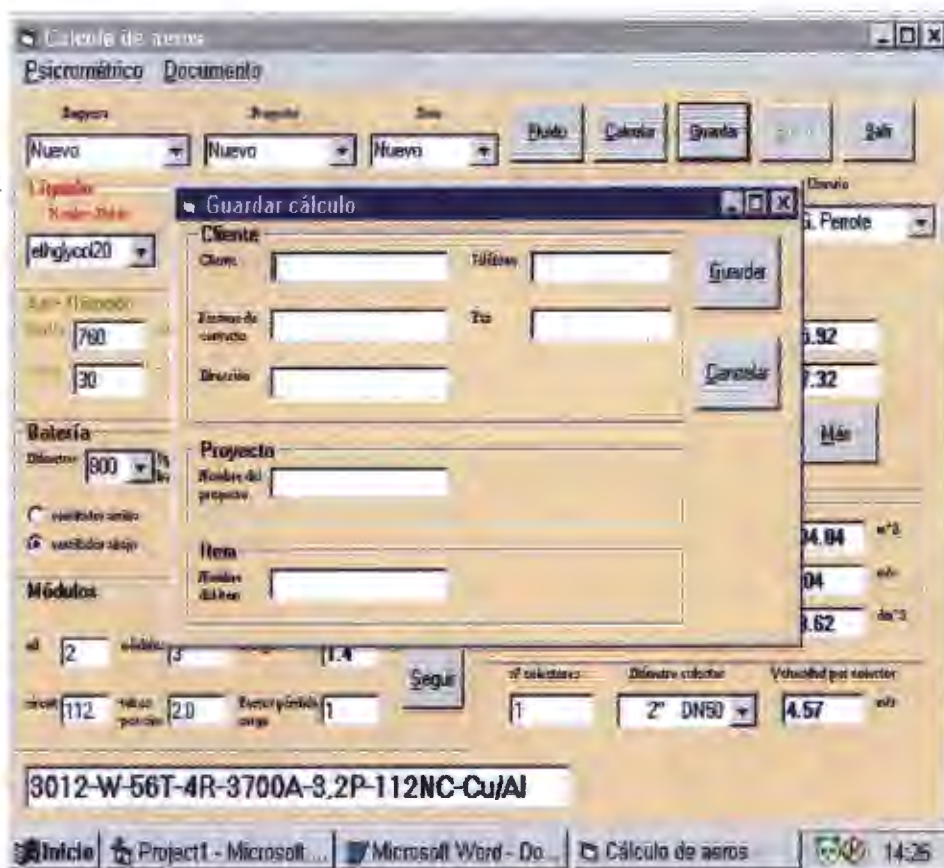
Se pueden observar todas las variables del psicrométrico haciendo click en la esquina superior izquierda de la ventana en *Psicrométrico*, obteniendo la pantalla que mostramos a continuación.



Una vez realizado el cálculo, tiene la opción de guardarlo incorporándolo a la base de datos. De esta forma, apretando el botón *Guardar*, aparecerá una nueva ventana donde se preguntará el nombre del cliente, proyecto e ítem dependiendo de si son o no, nuevos. La pantalla se muestra en la siguiente página.

No se podrán introducir nombres de clientes que ya existan (el programa no lo permitirá), al igual que no podrá haber dos proyectos con el mismo nombre dentro de un cliente.

Si se introduce un nombre para un ítem que ya existe dentro del proyecto y del cliente, el programa avisará y dará la opción de reemplazar el cálculo antiguo por el nuevo.



Se podrán borrar clientes, proyectos e items de la base de datos.

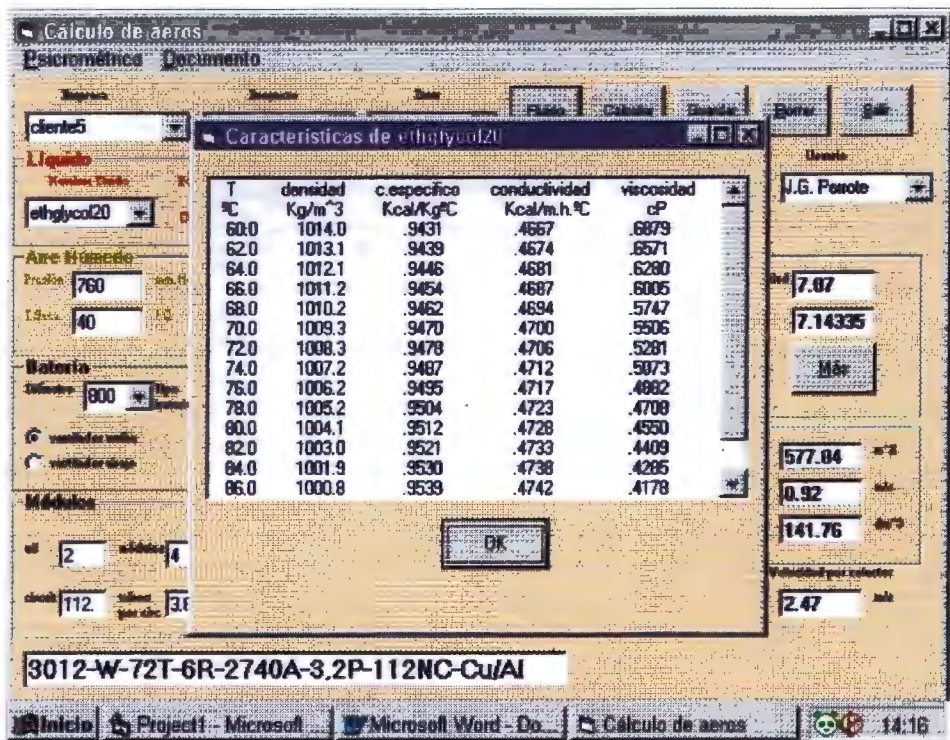
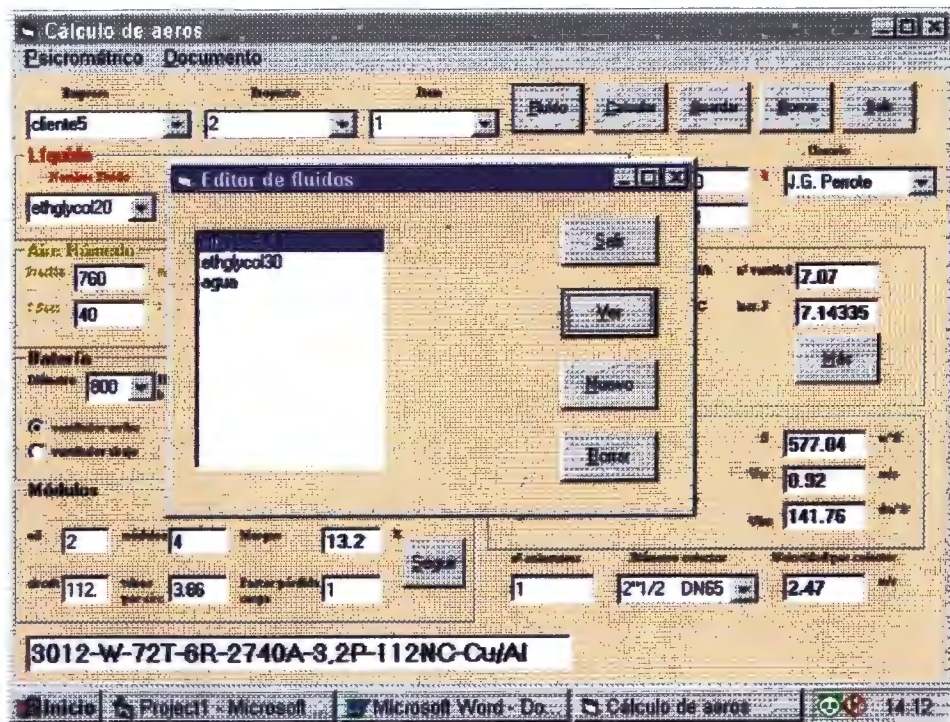
La forma de hacer esto es ir borrando cada item dentro de un proyecto. Cuando un proyecto no tenga ningún item por debajo, se podrá apretar el botón de *Borrar* y se irán así borrando cada proyecto dentro de un cliente. Una vez borrados todos los proyectos por debajo de un cliente, se podrá borrar el cliente.

El programa está hecho de forma que al hacer click en *Borrar*, siempre se pregunte al usuario si realmente desea borrarlo, ya que es una precaución en caso de error del usuario.

Una parte todavía no estudiada del programa es la referente al botón superior derecho *Fluido*.

Al apretarlo, aparecerá una nueva ventana con varias opciones.

Si apretamos *Ver* tras haber escogido un fluido de la lista nos aparecerá otra nueva ventana, donde deberemos escoger la temperatura inicial del fluido, la final y el incremento entre ellas, de forma que al apretar el botón *Ok* tengamos una lista de las características del fluido para cada temperatura, como se puede observar en la siguiente página.



Si en la ventana inicial de *Fluido* apretamos *Borrar* tras haber marcado uno de la lista, éste será eliminado de la base de datos.

Si apretamos *Nuevo* tendremos otra nueva aplicación, donde tendremos que introducir el nombre del fluido, que no puede estar en la base de datos, aunque de esto se encargará el programa, unas temperaturas y sus características a esas temperaturas.

Sabemos que si se introduce una sola temperatura, los valores de densidad, calor específico, viscosidad y conductividad serán constantes con la temperatura. Si metemos dos temperaturas y sus correspondientes características, estas serán lineales con la temperatura y si introducimos tres temperaturas tendrán dependencia parabólica.

Sólo nos queda por comentar el menú *Documento*.

Antes de pulsarlo, deben haber sido escogidos un cliente y un proyecto. Al apretarlo se creará un documento de Word con los diversos items que posea dicho proyecto.

LISTADOS

LISTADOS

Al ser Visual Basic un lenguaje llevado por eventos y de entorno Windows y visual, es difícil mostrar todo su código de golpe, no como puede suceder en otros lenguajes, como el C.

De esta forma, iré mostrando los formularios que considero importantes y sus rutinas fundamentales.

El resto de los formularios no los mostraré porque están simplemente formados por varias cajas de texto que muestran soluciones y un botón de salir, con lo cual, el formulario no conlleva ningún código y no creo que merezca la pena mencionarlos.

1. PRINCIPALFORM.

Es, como indica su nombre, el formulario principal del proyecto, donde encontramos las distintas formas de usar el proyecto.

Lo primero que hará el programa será cargar este formulario en pantalla, con lo cual, estudiaremos el suceso *Form_Load*.

1.1 Form_load.

En esta subrutina se llena los combos con todos los datos de las baterías, los clientes, los colectores y los distintos usuarios.

Private Sub Form_Load()

Dim diametro As Currency

*'Inicialización de variables booleanas necesarias para
'Llevar el control.*

clickproy = False
clientenuevo = False
proyectonuevo = False
itemnuevo = False

'Llenamos el combo de los fluidos

combofluido.Clear
Combo2.Clear
Combo3.Clear
Combo1.Clear

*'Por defecto tendremos diámetro de 800, batería 3.2 SIMPLE
'y un rango de 4 filas.*
Combo1.Text = "4"

```
Combo2.Text = "800"  
Combo3.Text = "3.2 SIMPLE"  
diamventil = "800"
```

*'creamos el objeto fluido que contiene una base de datos
'y un objeto recordset que contiene la tabla Tablafluidos
'dentro de la base de datos fluidos.mdb*

```
Set bdfluidos = OpenDatabase("c:\aeros\fluidos.mdb")  
Set  
rsfluidos=bdfluidos.OpenRecordset("tablafluidos",dbOpenDynaset)  
rsfluidos.MoveFirst
```

*'Añade los nombres de los fluidos de la tabla al combo
'de los fluidos*

```
Do While Not rsfluidos.EOF  
    combofluido.AddItem rsfluidos.Fields("fluido")  
    rsfluidos.MoveNext
```

Loop

*'Llenamos los combos con los distintos tipos de baterías
'con los diámetros posibles de los ventiladores
'y con los números de filas posibles*

```
Set bdaeros = OpenDatabase("c:\aeros\aeros.mdb")
```

'primero buscamos los diámetros

```
Set rsaerostipos = bdaeros.OpenRecordset("diametros")  
rsaerostipos.MoveFirst  
Do While Not rsaerostipos.EOF  
    Combo2.AddItem (rsaerostipos.Fields("diametro"))  
    rsaerostipos.MoveNext
```

Loop

*'Subrutina que llena las baterías posibles para ese diámetro.
Call llenobateria*

*'Subrutina que llena los rangos posibles para la batería y
'el diámetro escogido
Call llenolineas*

'Al tener un rango por defecto, se busca en la tabla

*'y guardamos el caudal y el incrementode temperatura
'máximo en las variables globales, correspondiente a
'un rango de 4, diámetro 800 y batería 3.2 SIMPLE*

Dim filas As Integer
Dim rsaerosfil As Recordset
Dim busco As String

filas = Val(Combo1.Text)
busco = "SELECT [DT/DTm],[caudal] FROM datos WHERE"
 & " (numerolíneas = " & filas & ") AND (Idbatdiam = " &
ibatdiam & ")"
Set rsaerosfil = bdaeros.OpenRecordset(busco, dbOpenDynaset)
Qi = rsaerosfil.Fields("caudal")
DELTA = rsaerosfil.Fields("DT/DTm")
Set rsaerosfil = Nothing

'llenamos los combos de la base de datos de clientes

Set bdclientes = OpenDatabase("C:\aeros\Bcliente.mdb")
Set rsclientes = bdclientes.OpenRecordset("Tablacliente")
combocliente.AddItem "Nuevo"
rsclientes.MoveFirst
Do While Not rsclientes.EOF
 combocliente.AddItem (rsclientes.Fields("nempresa"))
 rsclientes.MoveNext
Loop
comboproy.AddItem "Nuevo"
comboitem.AddItem "Nuevo"

'llenamos los combos de los diámetros de los colectores

Dim vari As String
Set rscolect = bdaeros.OpenRecordset("colectores")
Do While Not rscolect.EOF
 vari = rscolect.Fields("denominacion") & " " &
 rscolect.Fields("DN")
 combocol.AddItem vari
 rscolect.MoveNext
Loop

'Llenamos el combo del usuario.

*'el usuario por defecto será Juan G. Perrote
'y el teléfono y el fax, los que siguen.*

Dim rsusuarios As Recordset
Set rsusuarios = bdclientes.OpenRecordset("tusuario")
rsusuarios.MoveFirst

```
Combo4.AddItem "Nuevo"  
Do While Not rsusuarios.EOF  
    Combo4.AddItem rsusuarios.Fields("nombreusuario")  
    rsusuarios.MoveNext  
Loop  
rsusuarios.Close  
nombreusuario = "J.G. Perrote"  
telusuario = " 91 379 06 05"  
faxusuario = " 91 358 12 57"
```

```
'Movemos el formulario a la mitad de las pantalla.  
Me.Move (Screen.Width - Me.Width) / 2, (Screen.Height -  
(Me.Height + 350)) / 2
```

End Sub

1.2 Command_Click.

Corresponde a los cálculos del fluido. Realiza el balance de potencia y calcula la cuarta variable que falta.

Es el primer cálculo que hay que realizar para poder diseñar la batería.

Private Sub Command3_Click()

'hace el balance de potencia del fluido

```
Dim quetengo, quetengo2, nombrefluido, quitaespa, uno, suerte As  
String  
Dim cual(0 To 3) As String  
Dim rsprima As Recordset  
Dim den, caudal_liq As Currency  
Dim incrementwprima, incremenTw, Tuprima, Teprima As Double  
Dim a0d, a1d, a2d, a0cp, a1cp, a2cp As Double  
Dim criterio As String  
Dim cadenas As Long
```

```
Dim cadena, sincoma As String
For i = 0 To 3
    cual(i) = ""
Next i
```

```
'empezamos a ver que tenemos
For i = 0 To 3
    cual(i) = Text2(i).Text
    quetengo2 = Text2(i).Text
    If cual(i) <> "" Then
        control_liq = control_liq + 1
    End If
    If quetengo2 = "" Then
        control_liq_blanco = control_liq_blanco + 1
    End If
Next i
```

```
'control de los inputs del usuario
If control_liq = 4 Then
    MsgBox " Debe rellenar sólo 3 de las 4 variables del líquido", 16
    Command1.Enabled = True
    Exit Sub
End If
If control_liq_blanco > 1 Then
    MsgBox "Debe rellenar 3 de las 4 variables del líquido", 16
    Command1.Enabled = True
    Exit Sub
End If
```

```
If combofluido.ListIndex <> -1 Or (datosya = True And
    combofluido.Text <> "") Then
    Pot = 0
    Qw = 0
    Te = 0
    Tu = 0
    Tmw = 0
    solomuestro = False
    control_liq = 0
```

```
'buscamos las características del fluido escogido
nombrefluido = combofluido.Text
Set bdffluidos = OpenDatabase("c:\aeros\fluidos.MDB")
```

```
criterio = "SELECT a0d,a1d,a2d,a0cp,a1cp,a2cp FROM tablafluidos  
WHERE (fluido = ' ' & nombrefluido & ' ')"
```

```
Set rsfprima = bdffluidos.OpenRecordset(criterio)  
a0d = rsfprima.Fields("a0d")  
a1d = rsfprima.Fields("a1d")  
a2d = rsfprima.Fields("a2d")  
a0cp = rsfprima.Fields("a0cp")  
a1cp = rsfprima.Fields("a1cp")  
a2cp = rsfprima.Fields("a2cp")  
If cual(1) <> "" And cual(2) <> "" And cual(3) <> "" And cual(0) = ""  
Then  
    'tenemos que calcular la potencia  
    opcion = 1  
  
    Qw = Val(Text2(1).Text)  
    Te = Val(Text2(2).Text)  
    Tu = Val(Text2(3).Text)  
  
    Tmw = 0.5 * Te + 0.5 * Tu  
    tipos = 0  
    densidad = caracteristica(Tmw, tipos, a0d, a1d, a2d, a0cp, a1cp,  
a2cp)  
    tipos = 1  
    calorespecifico = caracteristica(Tmw, tipos, a0d, a1d, a2d, a0cp,  
a1cp, a2cp)  
    Pot = Qw * densidad * calorespecifico * (Te - Tu)  
End If
```

```
If cual(0) <> "" And cual(2) <> "" And cual(3) <> "" Then  
    'tenemos que calcular el caudal  
    opcion = 4  
    Pot = Val(Text2(0).Text)  
    Te = Val(Text2(2).Text)  
    Tu = Val(Text2(3).Text)  
    Tmw = 0.5 * Te + 0.5 * Tu  
    tipos = 0  
    densidad = caracteristica(Tmw, tipos, a0d, a1d, a2d, a0cp, a1cp,  
a2cp)  
    tipos = 1  
    calorespecifico = caracteristica(Tmw, tipos, a0d, a1d, a2d, a0cp,  
a1cp, a2cp)  
    Qw = Pot / (densidad * calorespecifico * (Te - Tu))  
End If
```

```
If cual(1) <> "" And cual(2) <> "" And cual(0) <> "" And cual(3) = ""
Then
    'calculamos la temperatura de salida del liquido
    Pot = Val(Text2(0).Text)
    Te = Val(Text2(2).Text)

    Qw = Val(Text2(1).Text)

    incremTwprima = Pot / (Qw * 1000)
    Tuprima = Te - incremTwprima
    Tmw = 0.5 * Tuprima + 0.5 * Te
    tipos = 0
    densidad = caracteristica(Tmw, tipos, a0d, a1d, a2d, a0cp, a1cp,
a2cp)
    tipos = 1
    calorespecifico = caracteristica(Tmw, tipos, a0d, a1d, a2d, a0cp,
a1cp, a2cp)
    den = Qw * densidad * calorespecifico
    incremenTw = Pot / den
    Tu = Te - incremenTw
    TS = Val(Text3(1).Text)
    If Text3(1).Text <> "" And TS > Tu Then
        MsgBox " se ha obtenido una temperatura de salida del liquido
        menor a la del aire húmedo.Deberá cambiar alguna de las
        dos", 16
        Exit Sub
    End If
    opcion = 2
End If

If cual(1) <> "" And cual(3) <> "" And cual(0) <> "" And cual(2) = ""
Then
    'calculamos la temperatura de entrada del liquido
    opcion = 3
    Pot = Val(Text2(0).Text)
    Tu = Val(Text2(3).Text)
    Qw = Val(Text2(1).Text)

    incremTwprima = Pot / (Qw * 1000)
    Teprima = Tu + incremTwprima
    Tmw = 0.5 * Teprima + 0.5 * Tu
    tipos = 0
    densidad = caracteristica(Tmw, tipos, a0d, a1d, a2d, a0cp, a1cp,
a2cp)
```

```
tipos = 1
calorespecifico = característica(Tmw, tipos, a0d, a1d, a2d, a0cp,
a1cp, a2cp)
incrementTw = Pot / (Qw * densidad * calorespecifico)
Te = Tu + incrementTw
End If
'Mostramos el resultado por pantalla
Select Case opcion
  Case 1:
    solucion = True
    cadenas = Pot
    Text2(0).Text = Format(cadenas)
    solucion = False
  Case 3:
    solucion = True
    cadena = Format(Te, "0.0")
    sincoma = quitarcoma(cadena)
    Text2(2).Text = sincoma
    solucion = False

  Case 2:
    solucion = True
    cadena = Format(Tu, "0.0")
    sincoma = quitarcoma(cadena)
    Text2(3).Text = sincoma
    solucion = False

  Case 4:
    solucion = True
    cadena = Format(Qw, "0.00")
    sincoma = quitarcoma(cadena)
    Text2(1).Text = sincoma
    solucion = False

End Select
primercalculo = True
datosya = True
cambiado = True
Else:
  'el usuario no ha escogido fluido
  MsgBox " Debe escoger un fluido", 16
  Exit Sub
End If
End Sub
```

1.3. característica.

En esta función se calcula la densidad o el calor específico del fluido a la temperatura media, en función de una variable de paso llamada tipo, la cual puede ser 1 o 0.

Los parámetros que se le mandan son la temperatura media, los coeficientes de la base de datos del calor específico y la densidad y el tipo.

Function característica(Tmw, tipos, a0d, a1d, a2d, a0cp, a1cp, a2cp) As Double

```
'calcula la densidad y el calor específico a la  
'temperatura media del agua  
'manda los coeficientes que ha encontrado en la  
'tabla correspondientes al fluido escogido  
'tipos es para distinguir si es la densidad  
' o cp  
If tipos = 0 Then  
    característica = a2d * Tmw ^ 2 + a1d * Tmw + a0d  
End If  
  
If tipos = 1 Then  
    característica = a2cp * Tmw ^ 2 + a1cp * Tmw + a0cp  
End If
```

End Function

1.4. Quitarcoma.

Al tener configurado Windows con idioma español, los decimales los escribe con coma en lugar de con puntos, lo cual nos puede dar problemas con la función *Val* de Visual Basic, ya que la coger un número con decimales si tiene comas, no cogerá los decimales.

Function quitarcoma(cadena) As String

*'Debido a problemas con la función Val y las comas
'esta función coge una cadena, mira cada uno de sus caracteres
'cuando encuentra una coma la sustituye por un punto*

```
Dim lugar, longitud As Integer
longitud = Len(cadena)
For lugar = 1 To longitud
    If Mid(cadena, lugar, 1) = "," Then
        Mid(cadena, lugar, 1) = "."
    End If
Next lugar
'actualiza el valor de la función, que es el resultado  
'de la misma
quitarcoma = cadena
End Function
```

1.5.Llenobateria.

Una vez escogido el diámetro, con el identificador del tipo de diámetro, mediante consultas podremos obtener los tipos de batería correspondientes y llenaremos el combo de las baterías con los nombres obtenidos.

Sub llenobateria()

*'subrutina que llena el combo correspondiente a los
'distintos tipos de batería determinados por
'el diámetro escogido*

Dim eldiam As Integer
Dim buscando As String

eldiam = Val(Combo2.Text)
rsaerostipos.MoveFirst

Do While Not rsaerostipos.EOF

*'cuando encuentra el diámetro, actualiza las variables
'idtipodiam, que determina el identificador del diámetro,
'revoluciones y potencia, variable dependiente sólo
'del diámetro necesaria para el documento de la oferta*

If eldiam = rsaerostipos.Fields("diametro") Then
 idtipodiam = rsaerostipos.Fields("Iddiametro")
 revoluciones = rsaerostipos.Fields("revoluciones")
 potencia = rsaerostipos.Fields("potencia")
 diamventil = rsaerostipos.Fields("diametro")
Exit Do

Else:
 rsaerostipos.MoveNext
End If

Loop

*'usando SQL abre un recordset con una consulta
'dependiente del identificador del diámetro con las baterías.*

buscando = " SELECT [Idbat],[Idbatdiam] FROM tipobatdiam " _
 & " WHERE [Iddiametro] = " & idtipodiam & " "

Dim rsdos, rstres As Recordset
Dim comparo As Integer

Set rsdos = bdaeros.OpenRecordset(buscando, dbOpenDynaset)
Set rstres = bdaeros.OpenRecordset("tipobat", dbOpenDynaset)
rsdos.MoveFirst
rstres.MoveLast
Do While Not rsdos.EOF
 comparo = rsdos.Fields("Idbat")
 buscando = " Idbat = " & comparo & " "

```
rstres.FindFirst buscando
Combo3.AddItem rstres.Fields("nombrebat")
rstres.MoveLast
rsdos.MoveNext
Loop
rsdos.Close
rstres.Close
```

End Sub

1.6.Llenolineas.

En esta subrutina, sabiendo el diámetro escogido y el tipo de batería mediante consultas podemos tener los posibles rangos correspondientes a lo escogido.

Sub llenolineas()

*'subrutina que llena el combo correspondiente a los
'distintos rangos determinados por el diámetro
'escogido y la batería.
'coge cuando encuentra la batería los coeficientes
'coef1,coef2,coef3,coef4 para calcular la pérdida
'de carga posteriormente.
'el paso para cálculos posteriores
'c1vint,c2vint coeficientes para el cálculo
'del volumen interno*

```
Dim nbat As String
Dim rstres As Recordset
```

```
nbat = Combo3.Text
Set rstres = bdaeros.OpenRecordset("tipobat", dbOpenDynaset)
rstres.MoveFirst
Do While Not rstres.EOF
  If nbat <> rstres.Fields("nombrebat") Then
    rstres.MoveNext
  Else:
    coef1 = rstres.Fields("coef1")
    coef2 = rstres.Fields("coef2")
    coef3 = rstres.Fields("coef3")
```

```
coef4 = rstres.Fields("coef4")
coef5 = rstres.Fields("coefs")
paso = rstres.Fields("paso")
idtipobat = rstres.Fields("Idbat")
c1vint = rstres.Fields("c1vint")
c2vint = rstres.Fields("c2vint")
numerobat = rstres.Fields("numerobat")
tubosaletas = rstres.Fields("tubosaletas")
Dtubos = rstres.Fields("Dtubos")
Exit Do
End If
Loop
rstres.Close

'se hace una consulta de la tabla tipobatdiam
'en la que tenemos las combinaciones de
'diámetros y baterías

Dim buscando As String
Dim rscuatro As Recordset
buscando="SELECT [Idbatdiam],[T] FROM tipobatdiam WHERE " _
&"(Iddiametro=" & idtipodiam & ")AND(Idbat=" & idtipobat & ")"

Set rscuatro = bdaeros.OpenRecordset(buscando, dbOpenDynaset)
idbatdiam = rscuatro.Fields("Idbatdiam")
T = rscuatro.Fields("T")
rscuatro.Close

buscando = " SELECT [numerolíneas],[caudal],[DT/DTm] FROM" _
& " datos WHERE Idbatdiam = " & idbatdiam & ""

Set rscinco = bdaeros.OpenRecordset(buscando, dbOpenDynaset)
rscinco.MoveFirst
Do While Not rscinco.EOF
    Combo1.AddItem rscinco.Fields("numerolíneas")
    rscinco.MoveNext
Loop

End Sub
```


1.7.Command4_Click.

Desde esta rutina se llama a hacer_cálculos, donde se calcula el número de módulos necesarios y otras soluciones. Con este botón se hace la segunda parte del cálculo, donde se hallan las variables del psicrométrico y el número de módulos necesarios para disipar la potencia determinada.

Private Sub Command4_Click()

'segundo cálculo

'control de posibles errores

If primercalculo = False Then

 MsgBox "Debe calcular antes la parte del líquido", 16

 Exit Sub

End If

If Text2(0).Text = "" Then

 MsgBox "Debe introducir la potencia", 16

 Exit Sub

End If

If Text2(1).Text = "" Then

 MsgBox "Debe introducir el caudal del líquido", 16

 Exit Sub

End If

If Text2(2).Text = "" Then

 MsgBox "Debe introducir la temperatura de entrada", 16

 Exit Sub

End If

If Text2(3).Text = "" Then

 MsgBox "Debe introducir la temperatura de salida", 16

 Exit Sub

End If

If (Val(Text3(1).Text)) = Te Then

 MsgBox "Los valores de temperaturas introducidos no son

```
    correctos", 16
    Text3(1).Text = ""
    Exit Sub
End If
If (Val(Text3(1).Text)) = Tu Then
    MsgBox "Los valores de temperaturas introducidos no son
    correctos", 16
    Text3(1).Text = ""
    Exit Sub
End If

If (Val(Text3(1).Text) < Val(Text3(2).Text)) And
    Option1(1).Value = True Then
    MsgBox "Las temperaturas introducidas no son correctas", 16
    Text3(1).Text = ""
    Exit Sub
End If
If Text3(0).Text = "" Then
    MsgBox "Debe introducir un valor para la presión
    atmosférica", 16
    Exit Sub
End If
If Text3(1).Text = "" Then
    MsgBox "Debe introducir un valor para la temperatura seca",
    16
    Exit Sub
End If

If Text3(2).Text = "" Then
    MsgBox "Debe introducir todas las variables del aire", 16
    Exit Sub
End If

If Combo2.Text = "" Then
    MsgBox "Debe escoger un tipo de diámetro", 16
    Exit Sub
End If
If Combo3.Text = "" Then
    MsgBox "Debe escoger un tipo de batería", 16
    Exit Sub
End If
If Combo1.Text = "" Then
    MsgBox "Debe escoger un rango", 16
    Exit Sub
End If
```

```
If Text2(5).Text = "" Then
    MsgBox "Debe introducir un valor para FC2", 16
    Exit Sub
End If
If Text2(4).Text = "" Then
    MsgBox "Debe introducir un valor para el margen", 16
    Exit Sub
End If
que = r
solomuestro = False
'subrutina que realiza los cálculos
Call hacer_calculos
cambiado = True

End Sub
```

1.8. hacer_calculos.

Sub hacer_calculos()

```
'en esta subrutina se realizan los cálculos del psicrométrico
'y del número de módulos necesarios
malomalo = False
Dim cadena, sincoma As String

'cogemos las variables de entrada
M = Val(Text2(4).Text)
FC2 = Val(Text2(5).Text)

P = Val(Text3(0).Text)
TS = Val(Text3(1).Text)

'esta opción nos indica cuál es la tercera variable de
'entrada del psicrométrico
'actualiza una variable global tipo dependiendo del caso
If Option1(0).Value = True Then
    var3 = Val(Text3(2).Text)
    tipo = 1
End If
```

```
If Option1(1).Value = True Then
    var3 = Val(Text3(2).Text)
    tipo = 2
End If
```

```
If Option1(2).Value = True Then
    var3 = Val(Text3(2).Text)
    tipo = 3
End If
```

```
V = 0
H = 0
X = 0
```

```
'calcula el coeficiente de corrección
'correspondiente a la temperatura media del fluido
FC1 = coef_correccion(Tmw)
If FC1 < 1 Then FC1 = 1
```

```
'el incremento final de temperatura del aire
incrementoOk = DELTA / (FC1 * FC2)
```

```
'llamamos a la subrutina que nos resuelve la ecuación
Tsprima = resolucion_TS_incremento(incrementoOk, Te, Tu, TS)
```

```
'llama a la subrutina que maneja el cálculo del
'psicrométrico.
```

```
Call cosaspsicrometrico(P, TS, var3, tipo)
```

```
'si el punto no es posible, malomalo=true
```

```
If malomalo = True Then
```

```
    'borra la pantalla, lo que va detrás del cálculo
```

```
    Text2(4).Text = "0"
```

```
    Text2(5).Text = "1"
```

```
    For i = 0 To 3
```

```
        Text5(i).Text = ""
```

```
    Next i
```

```
    Text1(0).Text = ""
```

```
    Text1(1).Text = "1"
```

```
    Text1(2).Text = ""
```

```
    Text1(3).Text = "1"
```

```
    Text1(4).Text = ""
```

```
    For i = 0 To 5
```

```
Text6(i).Text = ""
Next i
Text7.Text = ""
Text4(0).Text = Format(1)
Text4(1).Text = ""
combocol.Text = ""
Text3(2).Text = ""

malomalo = False
Exit Sub
End If

'si el cálculo es correcto
Hent = H / 4.178

Hentrada = H / 4.178
Ventrada = V
Vent = Ventrada
V = 1 / Vent
cadena = Format(V, "0.000")
densiaire = quitarcoma(cadena)
Trocient = Trocio
THent = TH
TSent = TS
Pent = P
HRent = HR
'si el dato de entrada era distinto de la
'humedad absoluta debido a reajustes por
'unidades, se debe multiplicar por 1000
If tipo <> 3 Then
    Xentrada = X * 1000
    Xent = Xentrada
End If
If tipo = 3 Then
    Xentrada = X
    Xent = Xentrada
End If

'el proceso es a x constante
tipo = 3

var3 = Xentrada

'las fórmulas sólo nos valen para temperaturas
'inferiors a 70°
```

*'si es mayor se resuleve mediante una regla
'de tres teniendo en cuanta que es lineal con
'la temperatura.*

```
If Tsprima > 69 Then
  Tsmayor = TS
  TSS = 69
  Call cosaspsicrometrico(P, TSS, var3, tipo)
  If malomalo = True Then
    Text2(4).Text = "0"
    Text2(5).Text = "1"
    For i = 0 To 3
      Text5(i).Text = ""
    Next i

    Text1(0).Text = ""
    Text1(1).Text = "1"
    Text1(2).Text = ""
    Text1(3).Text = "1"
    Text1(4).Text = ""

    For i = 0 To 5
      Text6(i).Text = ""
    Next i
    Text7.Text = ""
    Text4(0).Text = Format(1)
    Text4(1).Text = ""
    combocol.Text = ""
    Text3(2).Text = ""
    malomalo = False

    Exit Sub
  End If

  'el cálculo es bueno y tenemos los datos de
  'salida del aire
  Hsalidaini = H / 4.178
  Vsalida = V
  Xsalida = X
  Hsali = Hsalidaini
  Vsali = Vsalida
  Xsali = Xsalida
  Trociosali = Trocio
  THsali = TH
  Psali = P
  TSsali = Tsprima
```

```
HRsali = HR
TSS = 39
'calculamos a 39° para hacer la linealidad
Call cosaspsicrometrico(P, TSS, var3, tipo)
If malomalo = True Then
    Text2(4).Text = "0"
    Text2(5).Text = "1"
    For i = 0 To 3
        Text5(i).Text = ""
    Next i

    Text1(0).Text = ""
    Text1(1).Text = "1"
    Text1(2).Text = ""
    Text1(3).Text = "1"
    Text1(4).Text = ""

    For i = 0 To 5
        Text6(i).Text = ""
    Next i
    Text7.Text = ""
    Text4(0).Text = Format(1)
    Text4(1).Text = ""
    combocol.Text = ""
    Text3(2).Text = ""
    malomalo = False

    Exit Sub
End If

'con estas operaciones hacemos la regla de tres
Hsalidafin = H / 4.178
incrementoiniH = Hsalidaini - Hsalidafin
incrementoH = (Tsprima - TS) * incrementoiniH / 30
Hsali = Hent + incrementoH
End If

'si la temperaturar de salida está por debajo
'de los 70°
If Tsprima < 69 Then
    Call cosaspsicrometrico(P, Tsprima, var3, tipo)
    If malomalo = True Then Exit Sub
    Hsalida = H / 4.178
    Vsalida = V
    Xsalida = X
```

```
Hsali = Hsalida
Vsali = Vsalida
Xsali = Xsalida
Trociosali = Trocio
THsali = TH
Psali = P
TSsali = Tsprima
HRsali = HR
incrementoH = Hsalida - Hentrada
End If

'si tenemos los ventiladores arriba o abajo
'está en la opción4 y nos define el volumen
'específico del aire que vamos a utilizar
If Option4(0).Value = True Then
    V = Vsalida
End If
If Option4(1).Value = True Then
    V = Ventrada
End If

'La potencia disipada por módulo es el
'caudal de aire por ventilador por el
'incremento de entalpía entre la entrada y la
'salida y dividido por el volumen específico.
Pi = Qi * incrementoH / V

'el número de módulos necesarios será
N_modulos = Pot * (1 + M * 0.01) / Pi

'comienza el proceso de salida por pantalla
'del resultado de los cálculos-
Text5(0).Text = Pi
cadena = Format(Tsprima, "#0.#")
sincoma = quitarcoma(cadena)
Text5(1).Text = sincoma
cadena = Format(N_modulos, "#0.##")
sincoma = quitarcoma(cadena)
Text5(3).Text = sincoma
cadena = Format(incrementoH, "#0.##")
sincoma = quitarcoma(cadena)
Text5(2).Text = sincoma
segundocalculo = True
End Sub
```

1.9. *cosaspsicrometrico.*

Esta subrutina lleva el control de cuales son las variables del psicrométrico introducidas y a cuál de las subrutinas del mismo debe dirigirse.

Sub cosaspsicrometrico(P, TS, var3, tipo As Variant)

*'esta subrutina se encarga de llamar a la subrutina
'que calcula el psicrométrico adecuada.*

Dim i, grado, tipotemp, opcion1, opcion2, opcion3 As Integer
Dim Compruebo, texto(0 To 7), coef_iniciales_posi(0 To 5),
coef_iniciales_neg(0 To 6), coef_variables(0 To 6) As Double

```
If tipo = 2 Then
    TH = var3

    Call Calcula_PTSTH(P, TS, TH)
    If HR > 100 Or HR < 0 Then
        malomalo = True
        Exit Sub
    End If

End If
If tipo = 3 Then
    X = var3 * 0.001
    Call Calcula_PTSX(P, TS, X)
    If HR > 100 Or HR < 0 Then
        malomalo = True
        Exit Sub

    End If
End If
If tipo = 1 Then
    HR = var3
    Call Calcula_PTSHR(P, TS, HR)
End If

End Sub
```

1.10. Calcula_PTSTH.

Calcula el resto de las variables del diagrama psicrométrico, basándose en la presión atmosférica, la temperatura seca y la temperatura húmeda.

Las fórmulas utilizadas son una aproximación del diagrama psicrométrico, pertenecientes al programa Básica.

Sub Calcula_PTSTH(P, TS, TH)

Dim PS, TTS, PS1, Suma, LogPS, LogPSH, MU, WS, HRR, PSH,
PSH1, WSH, suma2, log_pstrocio, PSTrocio, TTH,
coef_iniciales_posi(0 To 5), coef_iniciales_neg(0 To 6)
As Double
Dim posit, pposit, i, j, M, l, grado, tipotemp As Integer
Dim coef_variables(0 To 6) As Variant

```
posit = 0
coef_iniciales_posi(2) = 1.3914993
coef_iniciales_posi(3) = -0.04860239
coef_iniciales_posi(4) = 0.000041764768
coef_iniciales_posi(5) = -0.000000014452093
coef_iniciales_posi(0) = -5800.2206 'el que está dividido por T
coef_iniciales_posi(1) = 6.5459673 'el que multiplica al log
coef_iniciales_neg(2) = 6.3925247
coef_iniciales_neg(3) = -0.009677843
coef_iniciales_neg(4) = 0.00000062215701
coef_iniciales_neg(5) = 2.0747825E-09
coef_iniciales_neg(6) = -9.484024E-13
coef_iniciales_neg(0) = -5674.5359 'el que va dividido por T
coef_iniciales_neg(1) = 4.16535019 'el del logaritmo
```

```
TTS = TS + 273.15
```

```
If (TS > 0) Or (TS = 0) Then
    posit = 1 'Var global si TS es positivo
    PS1 = 0
```

```
For i = 2 To 5
  j = i - 2
  Suma = coef_iniciales_posi(i) * TTS ^ j
  PS1 = PS1 + Suma
Next i
LogPS = (coef_iniciales_posi(0) / TTS + coef_iniciales_posi(1) *
Log(TTS) + PS1)
PS = Exp(LogPS) * 0.00750061683
End If

If (TS < 0) Then
  PS1 = 0
  For i = 2 To 6
    j = i - 2
    Suma = coef_iniciales_neg(i) * TTS ^ j
    PS1 = PS1 + Suma
  Next i
  LogPS = (coef_iniciales_neg(0) / TTS + coef_iniciales_neg(1) *
    Log(TTS) + PS1)
  PS = Exp(LogPS) * 0.00750061683
End If
TTH = TH + 273.15

If (TH > 0) Or (TH = 0) Then
  posit = 1 'Var global si TS es positivo
  PSH1 = 0
  For M = 2 To 5
    l = M - 2
    suma2 = coef_iniciales_posi(M) * TTH ^ l
    PSH1 = PSH1 + suma2
  Next M
  LogPSH = (coef_iniciales_posi(0) / TTH +
    coef_iniciales_posi(1) * Log(TTH) + PSH1)
  PSH = Exp(LogPSH) * 0.00750061683
End If
If (TH < 0) Then
  PSH1 = 0
  For M = 2 To 6
    l = M - 2
    suma2 = coef_iniciales_neg(M) * TTH ^ l
    PSH1 = PSH1 + suma2
  Next M
  LogPSH = (coef_iniciales_neg(0) / TTH + coef_iniciales_neg(1)
    * Log(TTH) + PSH1)
  PSH = Exp(LogPSH) * 0.00750061683
```

End If

$$\text{WSH} = 0.62198 * \text{PSH} / (\text{P} - \text{PSH})$$

$$\text{X} = ((2501 - 2.381 * \text{TH}) * \text{WSH} - (\text{TS} - \text{TH})) / (2501 + 1.805 * \text{TS} - 4.186 * \text{TH})$$

$$\text{WS} = 0.62198 * \text{PS} / (\text{P} - \text{PS})$$

$$\text{MU} = \text{X} / \text{WS}$$

$$\text{HR} = \text{MU} / (1 - (1 - \text{MU}) * (\text{PS} * 0.001315789474 / \text{P} * 0.0013157895)) * 100$$

If HR > 100 Or HR < 0 Then

MsgBox " Punto incorrecto", 15

malomalo = True

Exit Sub

End If

$$\text{V} = 0.287055 * \text{TTS} / (\text{P} * 0.133322368421) * (1 + 1.6078 * \text{X})$$

$$\text{H} = \text{TS} + \text{X} * (2501 + 1.805 * \text{TS})$$

$$\text{PSTrocio} = (\text{P} * 133.322368421 * \text{X} / (0.62198 + \text{X}))$$

log_pstrocio = Log(PSTrocio)

On Error GoTo final

If TH < 0 Then

pposit = 0

For i = 0 To 6

If i <> 2 Then

coef_variables(i) = coef_iniciales_neg(i)

End If

Next i

coef_variables(2) = coef_iniciales_neg(2) - Log(PSTrocio)

End If

If TH > 0 Or TH = 0 Then

pposit = 1

For i = 0 To 5

If i <> 2 Then

coef_variables(i) = coef_iniciales_posi(i)

End If

Next i

coef_variables(2) = coef_iniciales_posi(2) - Log(PSTrocio)

coef_variables(6) = 0

End If

Trocio = Resolver_ecuacion(coef_variables(), log_pstrocio)

```
GoTo seacabo
final:
    error = True
seacabo:

End Sub
```

1.11. *Calcula_PTSHR.*

Subrutina similar a la anterior.

Sub *Calcula_PTSHR(P, TS, HR)*

'en esta subrutina se realiza el cálculo del psicrométrico

```
Dim PS, TTS, PS1, Suma, LogPS, MU, WS, HRR, log_pstroccio,
PSTroccio, coef_iniciales_posi(0 To 5), coef_iniciales_neg(0 To 6)
```

```
As Double
```

```
Dim posit, pposit, i, j, grado, tipotemp As Integer
```

```
Dim coef_variables(0 To 6) As Variant
```

```
posit = 0
```

*'estos coeficientes son los del cálculo de la presión
'de saturación*

```
coef_iniciales_posi(2) = 1.3914993
```

```
coef_iniciales_posi(3) = -0.04860239
```

```
coef_iniciales_posi(4) = 0.000041764768
```

```
coef_iniciales_posi(5) = -0.000000014452093
```

```
coef_iniciales_posi(0) = -5800.2206
```

'el que está dividido por T

```
coef_iniciales_posi(1) = 6.5459673
```

'el que multiplica al log

```
coef_iniciales_neg(2) = 6.3925247
```

```
coef_iniciales_neg(3) = -0.009677843
```

```
coef_iniciales_neg(4) = 0.00000062215701
```

```
coef_iniciales_neg(5) = 2.0747825E-09
```

```
coef_iniciales_neg(6) = -9.484024E-13
```

```
coef_iniciales_neg(0) = -5674.5359
```

'el que va dividido por T

```
coef_iniciales_neg(1) = 4.16535019
```

'el del logaritmo

'pasamosla temperatura a grados Kelvin

TTS = TS + 273.15

'los coeficientes son distintos en el caso

'de la temperatura positiva o negativa.

If (TS > 0) Or (TS = 0) Then

posit = 1 *'Var global si TS es positivo*

PS1 = 0

For i = 2 To 5

j = i - 2

Suma = coef_iniciales_posi(i) * TTS ^ j

PS1 = PS1 + Suma

Next i

'con la presión de saturación de la temperatura

'seca

LogPS = (coef_iniciales_posi(0) / TTS + coef_iniciales_posi(1)
* Log(TTS) + PS1)

PS = Exp(LogPS) * 0.00750061683

End If

If (TS < 0) Then

PS1 = 0

For i = 2 To 6

j = i - 2

Suma = coef_iniciales_neg(i) * TTS ^ j

PS1 = PS1 + Suma

Next i

LogPS = (coef_iniciales_neg(0) / TTS + coef_iniciales_neg(1) *
Log(TTS) + PS1)

PS = Exp(LogPS) * 0.00750061683

End If

'cn las fórmulas se realiza la aproximación del psicrométrico

HRR = HR / 100

WS = 0.62198 * PS / (P - PS)

**MU = HRR * (P * 0.001315789474 - PS * 0.001315789474) /
(P * 0.001315789474 - PS * 0.001315789474 * HRR)**

X = MU * WS

V = 0.287055 * TTS / (P * 0.133322368421) * (1 + 1.6078 * X)

H = TS + X * (2501 + 1.805 * TS)

PSTrocio = (P * 133.322368421 * X / (0.62198 + X))

If PSTrocio <= 0 Then

MsgBox "El punto introducido no es correcto", vbAbortRetryIgnore
puntomalo = True

```
Exit Sub
End If

log_pstroccio = Log(PSTroccio)
If TS < 4 Then
  pposit = 0
  For i = 0 To 6
    If i <> 2 Then
      coef_variables(i) = coef_iniciales_neg(i)
    End If
  Next i
  coef_variables(2) = coef_iniciales_neg(2) - Log(PSTroccio)
End If
If TS > 4 Or TS = 4 Then
  pposit = 1
  For i = 0 To 5
    If i <> 2 Then
      coef_variables(i) = coef_iniciales_posi(i)
    End If
  Next i
  coef_variables(2) = coef_iniciales_posi(2) - Log(PSTroccio)
  coef_variables(6) = 0
End If

Troccio = Resolver_ecuacion(coef_variables(), log_pstroccio)
If TS < 4 Then
  pposit = 0
  For i = 0 To 6
    coef_variables(i) = coef_iniciales_neg(i)
  Next i
End If
If TS > 4 Or TS = 4 Then
  pposit = 1
  For i = 0 To 5
    coef_variables(i) = coef_iniciales_posi(i)
  Next i
coef_variables(6) = 0
End If
TH = resolucion(coef_variables(), P, TS, X)

End Sub
```

1.12. *Calcula_PTSX*

Similar a las dos anteriores.

Sub Calcula_PTSX(P, TS, X)

Dim PS, TTS, PS1, Suma, LogPS, MU, WS, HRR, log_pstrocio,
PSTroccio, coef_iniciales_posi(0 To 5), coef_iniciales_neg(0 To 6)
As Double
Dim posit, pposit, i, j, grado, tipotemp As Integer
Dim coef_variables(0 To 6) As Variant

```
posit = 0
coef_iniciales_posi(2) = 1.3914993
coef_iniciales_posi(3) = -0.04860239
coef_iniciales_posi(4) = 0.000041764768
coef_iniciales_posi(5) = -0.000000014452093
coef_iniciales_posi(0) = -5800.2206 'el que está dividido por T
coef_iniciales_posi(1) = 6.5459673 'el que multiplica al log
coef_iniciales_neg(2) = 6.3925247
coef_iniciales_neg(3) = -0.009677843
coef_iniciales_neg(4) = 0.00000062215701
coef_iniciales_neg(5) = 2.0747825E-09
coef_iniciales_neg(6) = -9.484024E-13
coef_iniciales_neg(0) = -5674.5359 'el que va dividido por T
coef_iniciales_neg(1) = 4.16535019 'el del logaritmo
```

```
TTS = TS + 273.15
If (TS > 0) Or (TS = 0) Then
    posit = 1 'Var global si TS es positivo
    PS1 = 0
    For i = 2 To 5
        j = i - 2
        Suma = coef_iniciales_posi(i) * TTS ^ j
        PS1 = PS1 + Suma
    Next i
    LogPS = (coef_iniciales_posi(0) / TTS + coef_iniciales_posi(1) *
        Log(TTS) + PS1)
    PS = Exp(LogPS) * 0.00750061683
End If
If (TS < 0) Then
    PS1 = 0
    For i = 2 To 6
```

```
    j = i - 2
    Suma = coef_iniciales_neg(i) * TTS ^ j
    PS1 = PS1 + Suma
Next i
LogPS = (coef_iniciales_neg(0) / TTS + coef_iniciales_neg(1) *
        Log(TTS) + PS1)
PS = Exp(LogPS) * 0.00750061683
End If
```

```
WS = 0.62198 * PS / (P - PS)
MU = X / WS
HR = MU / (1 - (1 - MU) * (PS * 0.001333224 /
    P * 0.0013157895)) * 100
```

```
If HR > 100 Or HR < 0 Then
    MsgBox " Punto incorrecto", 16
    puntomalo = True
    Exit Sub
End If
```

```
V = 0.287055 * TTS / (P * 0.133322368421) * (1 + 1.6078 * X)
H = TS + X * (2501 + 1.805 * TS)
PSTrocio = (P * X / (0.62198 + X)) * 133.322368
log pstrocio = Log(PSTrocio)
```

```
If TS < 4 Then
    pposit = 0
    For i = 0 To 6
        If i <> 2 Then
            coef_variables(i) = coef_iniciales_neg(i)
        End If
    Next i
    coef_variables(2) = coef_iniciales_neg(2) - Log(PSTrocio)
End If
```

```
If TS > 4 Or TS = 4 Then
    pposit = 1
    For i = 0 To 5
        If i <> 2 Then
            coef_variables(i) = coef_iniciales_posi(i)
        End If
    Next i
```

```
    Next i
    coef_variables(6) = 0
End If
```

Trocio = Resolver_ecuacion(coef_variables(), log_pstrocio)

If TS < 4 Then

 pposit = 0

 For i = 0 To 6

 coef_variables(i) = coef_iniciales_neg(i)

 Next i

End If

If TS > 4 Or TS = 4 Then

 pposit = 1

 For i = 0 To 5

 coef_variables(i) = coef_iniciales_posi(i)

 Next i

 coef_variables(6) = 0

End If

TH = resolucion(coef_variables(), P, TS, X)

X = X * 1000

End Sub

1.13.Resolver_ecuación.

Para calcular la temperatura nos basamos en la fórmula de la presión de saturación del aire húmedo, que sabemos que sigue la fórmula para temperaturas positivas:

$$\ln(PWS) = \frac{C}{T} + C_2 + C_3T + C_4T^2 + C_5T^3 + C_6T^4 + C_7\ln(t)$$

Sabemos la presión de saturación y nos basaremos en el método de la bisección, hasta que ambos miembros de la ecuación sean iguales.

**Function Resolver_ecuacion(coef_variables(), log_pstrocio)
As Double**

```
Dim coef_variables_aux(0 To 6) As Variant
Dim encontrado, i As Integer
Dim una, X_sup, X_inf, X_supant, X_infant, X_media,
    result_anterior, result_media, error, punto, signo1, signo2,
    coef_iniciales_posi(0 To 5), coef_iniciales_neg(0 To 6)
    As Double
Dim dos_soluciones As Boolean

X_sup = 70
X_inf = -80
encontrado = 0
result_anterior = 0
X_supant = X_sup
X_infant = X_inf
dos_soluciones = False
coef_iniciales_posi(2) = 1.3914993
coef_iniciales_posi(3) = -0.04860239
coef_iniciales_posi(4) = 0.000041764768
coef_iniciales_posi(5) = -0.000000014452093
coef_iniciales_posi(0) = -5800.2206 'el que está dividido por T
coef_iniciales_posi(1) = 6.5459673 'el que multiplica al log
coef_iniciales_neg(2) = 6.3925247
coef_iniciales_neg(3) = -0.009677843
coef_iniciales_neg(4) = 0.00000062215701
coef_iniciales_neg(5) = 2.0747825E-09
coef_iniciales_neg(6) = -9.484024E-13
coef_iniciales_neg(0) = -5674.5359 'el que va dividido por T
coef_iniciales_neg(1) = 4.16535019 'el del logaritmo

Do While i < 10000
    X_media = 0.5 * (X_sup + X_inf)
    punto = X_media
    If (punto > 0) Then
        For i = 0 To 5
```

```
    If i <> 2 Then
        coef_variables_aux(i) = coef_iniciales_posi(i)
    End If

Next i
coef_variables_aux(2) = coef_iniciales_posi(2) - log_pstrocio
coef_variables_aux(6) = 0
End If

If punto < 0 Then
    For i = 0 To 6
        If i <> 2 Then
            coef_variables_aux(i) = coef_iniciales_neg(i)
        End If
    Next i
    coef_variables_aux(2) = coef_iniciales_neg(2) - log_pstrocio
End If

result_media = result(coef_variables_aux(), punto)

error = result_media - result_anterior
If error < 0 Then error = error * (-1)
If error < 0.001 Then
    encontrado = 1
    Resolver_ecuacion = X_media
Exit Do
End If

punto = X_sup
If (punto > 0) Then
    For i = 0 To 5
        If i <> 2 Then
            coef_variables_aux(i) = coef_iniciales_posi(i)
        End If

    Next i
    coef_variables_aux(2) = coef_iniciales_posi(2) - log_pstrocio
    coef_variables_aux(6) = 0
End If

If punto < 0 Then
    For i = 0 To 6
```

```
        If i <> 2 Then
            coef_variables_aux(i) = coef_iniciales_neg(i)
        End If
    Next i
    coef_variables_aux(2) = coef_iniciales_neg(2) - log_pstrocio
End If
result_sup = result(coef_variables_aux(), punto)

punto = X_inf
If (punto > 0) Then
    For i = 0 To 5
        If i <> 2 Then
            coef_variables_aux(i) = coef_iniciales_posi(i)
        End If

        Next i
        coef_variables_aux(2) = coef_iniciales_posi(2) - log_pstrocio
        coef_variables_aux(6) = 0
    End If

    If punto < 0 Then
        For i = 0 To 6
            If i <> 2 Then
                coef_variables_aux(i) = coef_iniciales_neg(i)
            End If
        Next i
        coef_variables_aux(2) = coef_iniciales_neg(2) - log_pstrocio
    End If
    result_inf = result(coef_variables(), punto)

    signo1 = result_media * result_sup
    signo2 = result_media * result_inf

    'aquí realizamos el cambio de semiintervalos
    'dependiendo de si el producto de las soluciones es positivo
    'o negativo
    If signo1 < 0 And signo2 > 0 Then
        X_inf = X_media

    End If

    If signo2 < 0 And signo1 > 0 Then
        X_sup = X_media
```

```
End If
If (signo2 < 0 And signo1 < 0) Or (signo2 > 0 And signo1 > 0) Then
  resultadoant = result_sup
  una = dossoluciones(resultadoant, X_sup, X_inf,
    coef_variables(), log_pstrocio)
  Resolver ecuacion = una
  Exit Function
End If

result_anterior = result_media
i = i + 1
Loop

End Function
```

1.14.result.

Basándose en la fórmula:

$$\ln(PWS) = \frac{C}{T} + C_2 + C_3T + C_4T^2 + C_5T^3 + C_6T^4 + C_7\ln(t)$$

Esta función calcula el segundo miembro de esta ecuación, de forma que hay que pasarle los coeficientes y la temperatura a la que lo calcularemos.

Function result(coef_variables_aux(), punto) As Double

*'calcula el resultado de la presión de saturación
'a la temperatura seca, que será lo que tendremos que igualar*

Dim i, j As Integer

Dim parcial, a, P, q, result1, ppunto, logresult As Double

result1 = 0

```
ppunto = punto + 273.15
For i = 2 To 6
  j = i - 2
  Select Case j
    Case 0: a = 1
    Case 1: a = ppunto
    Case 2: a = ppunto ^ 2
    Case 3: a = ppunto ^ 3
    Case 4: a = ppunto ^ 4

  End Select
  parcial = coef_variables_aux(i) * a

  result1 = result1 + parcial
Next i
P = coef_variables_aux(0) / ppunto
q = coef_variables_aux(1) * Log(ppunto)
result = coef_variables_aux(0) / ppunto + coef_variables_aux(1) *
Log(ppunto) + result1
```

End Function

1.15. dossoluciones.

Si en el último intervalo donde se estudió la función, ambos valores de la función tienen el mismo signo, tenemos dos soluciones en el intervalo, con lo cual nos iremos al final del último intervalo y lo recorreremos de unidad en unidad comprobando los signos de la función. Cuando encontremos el intervalo de una unidad, el cual lo recorreremos ahora de décima en décima.

**Function dossoluciones(resultadoant,X_sup,X_inf,
coef_variables(),log_pstrocio) As Currency**

*'llamamos a esta función si al coger el intervalo correspondiente
'en el método de la bisección nos encontramos con dos
'soluciones.*

*'cogemos los puntos anteriores donde la función tenía distinto
'signo y nos movemos de unidad en unidad.*

'posteriormente lo haremos con un incremento de una décima

Dim coef_variables_aux(0 To 6) As Variant
Dim incremento, X_anterior, X_media, resultadoact As Currency
Dim primera_vez As Boolean
Dim coef_iniciales_posi(0 To 5), coef_iniciales_neg(0 To 6)
As Currency

primera_vez = True

coef_iniciales_posi(2) = 1.3914993
coef_iniciales_posi(3) = -0.04860239
coef_iniciales_posi(4) = 0.000041764768
coef_iniciales_posi(5) = -0.000000014452093
coef_iniciales_posi(0) = -5800.2206 'el que está dividido por T
coef_iniciales_posi(1) = 6.5459673 'el que multiplica al log
coef_iniciales_neg(2) = 6.3925247
coef_iniciales_neg(3) = -0.009677843
coef_iniciales_neg(4) = 0.00000062215701
coef_iniciales_neg(5) = 2.0747825E-09
coef_iniciales_neg(6) = -9.484024E-13
coef_iniciales_neg(0) = -5674.5359 'el que va dividido por T
coef_iniciales_neg(1) = 4.16535019 'el del logaritmo

X_media = X_sup - 1 'comienzo por la siguiente
incremento = 1

Do While X_media > X_inf
punto = X_media
If (punto > 0) Then
For i = 0 To 5
If i <> 2 Then
coef_variables_aux(i) = coef_iniciales_posi(i)
End If
Next i
coef_variables_aux(2) = coef_iniciales_posi(2) - log_pstrocio


```
        coef_variables_aux(6) = 0
    End If

    If punto < 0 Then
        For i = 0 To 6
            If i <> 2 Then
                coef_variables_aux(i) = coef_iniciales_neg(i)
            End If
        Next i
        coef_variables_aux(2) = coef_iniciales_neg(2) - log_pstrocio
    End If
    resultadoact = result(coef_variables_aux(), punto)
    signo1 = resultadoact * resultadoant
    If signo1 < 0 Then

        If primera_vez = False Then
            dossoluciones = X_media
            Exit Function
        End If

        If primera_vez = True Then
            incremento = 0.1
            X_media = X_anterior
            primera_vez = False
        End If
    End If

    X_anterior = X_media
    resultadoant = resultadoact
    X_media = X_media - incremento
Loop

End Function
```

1.16.resolución.

Esta función nos ayuda a calcular la temperatura húmeda.

También está basada en el método de la bisección y es muy similar a la función *Resolver_ecuación*.

Function resolucion(coef_variables(), P, TS, X) As Double

*'en esta función es donde se aplica el método de la
'bisección*

Dim coef_variables_aux(0 To 6) As Variant
Dim encontrado, i As Integer
Dim X_sup, X_inf, X_media, result_anterior, result_media,
error, punto, signo1, signo2, coef_iniciales_posi(0 To 5),
coef_iniciales_neg(0 To 6) As Double

*'definimos una temperatura superior e inferior iniciales
'donde se encontrará la nuestra*

X_sup = 70
X_inf = -100
encontrado = 0
result_anterior = 0
coef_iniciales_posi(2) = 1.3914993
coef_iniciales_posi(3) = -0.04860239
coef_iniciales_posi(4) = 0.000041764768
coef_iniciales_posi(5) = -0.000000014452093
coef_iniciales_posi(0) = -5800.2206 'el que está dividido por T
coef_iniciales_posi(1) = 6.5459673 'el que multiplica al log
coef_iniciales_neg(2) = 6.3925247
coef_iniciales_neg(3) = -0.009677843
coef_iniciales_neg(4) = 0.00000062215701
coef_iniciales_neg(5) = 2.0747825E-09
coef_iniciales_neg(6) = -9.484024E-13
coef_iniciales_neg(0) = -5674.5359 'el que va dividido por T
coef_iniciales_neg(1) = 4.16535019 'el del logaritmo

'comienza el bucle del método

Do While i < 100000

'hallamos el puntoo medi del intervalo

X_media = 0.5 * (X_sup + X_inf)

punto = X_media

If (punto > 0) Then

For i = 0 To 5

coef_variables_aux(i) = coef_iniciales_posi(i)

Next i

coef_variables_aux(6) = 0

End If

If punto < 0 Then

For i = 0 To 6

coef_variables_aux(i) = coef_iniciales_neg(i)

Next i

End If

'llamamos a la función result para que nos dé

'el resultado correspondiente a la temperatura

result_media = valor(coef_variables_aux(), punto, P, TS, X)

'el error es el resultado anterior menos el actual

'cundo sea mínimo, sabremos la solución y sale de la función

error = result_media - result_anterior

If error < 0 Then error = error * (-1)

If error < 0.001 Then

encontrado = 1

resolucion = X_media

Exit Do

End If

'si no es todavía calcula es lresultado del punto superior

punto = X_sup

If (punto > 0) Then

For i = 0 To 5

coef_variables_aux(i) = coef_iniciales_posi(i)

Next i

coef_variables_aux(6) = 0

End If

If punto < 0 Then

For i = 0 To 6

coef_variables_aux(i) = coef_iniciales_neg(i)

Next i

End If

result_sup = valor(coef_variables_aux(), punto, P, TS, X)

```
'Y del inferior
punto = X_inf
If (punto > 0) Then
  For i = 0 To 5
    coef_variables_aux(i) = coef_iniciales_posi(i)

  Next i
  coef_variables_aux(6) = 0
End If

If punto < 0 Then
  For i = 0 To 6
    coef_variables_aux(i) = coef_iniciales_neg(i)
  Next i
End If
result_inf = valor(coef_variables_aux(), punto, P, TS, X)

'dependiendo del signo nos quedaremos con el
'semiintervalo superior o el inferior
signo1 = result_media * result_sup
If signo1 < 0 Then
  X_inf = X_media
End If
signo2 = result_media * result_inf
If signo2 < 0 Then
  X_sup = X_media
End If
  result_anterior = result_media
  i = i + 1
Loop

End Function
```

1.17. valor.

```
Function valor(coef_variables_aux(), punto, P, TS, X)  
As Double
```

```
'calcula el resultado de la presión de saturación
'a la temperatura seca, que será lo que tendremos que igualar
```

```
Dim i, j As Integer
Dim parcial, a, result1, ppunto, logresult, PP As Double
result1 = 0
ppunto = punto + 273.15
For i = 2 To 6
    j = i - 2
    Select Case j
        Case 0: a = 1
        Case 1: a = ppunto
        Case 2: a = ppunto ^ 2
        Case 3: a = ppunto ^ 3
        Case 4: a = ppunto ^ 4

    End Select
    parcial = coef_variables_aux(i) * a
    result1 = result1 + parcial
Next i
logresult = coef_variables_aux(0) / ppunto + coef_variables_aux(1)
            * Log(ppunto) + result1
valorinter = Exp(logresult) * 0.00750061683
PP = valorinter / (P - valorinter)
valor = ((2500 - 2.381 * punto) * 0.62198 * PP - (TS - punto)) /
        (2501 + 1.805 * TS - 4.186 * punto) - X
```

End Function

1.18. clrsc

```
Sub clrsc()
'borra la pantalla
Dim i As Integer
Call borrar_var_psicro

combofluido.Text = ""
For i = 0 To 3
    Text2(i).Text = ""
Next i
Text2(5).Text = "1"
For i = 0 To 2
    Text3(i).Text = ""
```

```
Next i
Option1(0).Value = True
If Combo2.Text <> "800" Then
    Combo3.Clear
    Combo1.Clear
    Combo3.Text = "3.2 SIMPLE"
    Combo2.Text = "800"
    Combo1.Text = "4"
    Call llenobateria
    Call llenolineas
    Option4(1).Value = True
Else:
    If Combo3.Text <> "3.2 SIMPLE" Then
        Combo3.Text = "3.2 SIMPLE"
        Combo1.Text = "4"
        Call llenolineas
        rscinco.MoveLast
        buscar = " numerolíneas = '4' "
        r = 4
        rscinco.FindFirst (buscar)
        Qi = rscinco.Fields("caudal")
        DELTA = rscinco.Fields("DT/DTm")
        rscinco.Close

    Else:
        If Combo1.Text <> "4" Then
            Combo1.Text = "4"
            rscinco.MoveLast
            buscar = " numerolíneas = 4 "
            r = 4
            rscinco.FindFirst (buscar)
            Qi = rscinco.Fields("caudal")
            DELTA = rscinco.Fields("DT/DTm")
            rscinco.Close
        End If
    End If

End If

For i = 0 To 3
    Text5(i).Text = ""
Next i
Text5(4).Text = ""
Text5(4).Visible = False
```

```
Label8(22).Visible = False
Text1(0).Text = ""
Text1(1).Text = "1"
Text1(2).Text = ""
Text1(3).Text = "1"
Text1(4).Text = ""

For i = 0 To 5
    Text6(i).Text = ""
Next i
Text7.Text = ""
Text4(0).Text = Format(1)
Text4(1).Text = ""
combocol.Text = ""
Dim frmP As New fpsicro
For j = 0 To 7
    frmP.variable(j).Text = ""
    frmP.Text1(j).Text = ""
Next j

cambiado = False

End Sub
```

1.19. Command6_Click

Pasamos a la tercera parte del cálculo , donde hallamos la pérdida de carga, y demás características importantes de la batería.

```
Private Sub Command6_Click()
'tercera parte del cálculo
'control de errores
On Error GoTo quepasa
If segundocalculo = False Then
    MsgBox "Debe calcular los módulos necesarios primero", 16
    Exit Sub
End If
```

```
If buscomod = False Then
    MsgBox "No se ha introducido el número de módulos o
    ventiladores" _
        & " o no se ha encontrado en la base de datos", 16
    Exit Sub
End If
If Qw = 0 Then
    MsgBox " No hay datos sobre el caudal del líquido", 16
    Exit Sub
End If
If N_modulos = 0 Then
    MsgBox " No hay datos sobre el número de módulos o ventiladores
    necesarios", 16
    Exit Sub
End If

If Text1(1).Text = "" Then
    MsgBox "Debe introducir el número de unidades", 16
    Exit Sub
End If
If Text1(0).Text = "" Then
    MsgBox "Debe introducir el número de ventiladores o módulos",
    16
    Exit Sub
End If
If Text1(2).Text = "" Then
    MsgBox "Debe introducir el número de circuitos", 16
    Exit Sub
End If
If Text1(3).Text = "" Then
    MsgBox "Debe introducir el factor para la pérdida de carga", 16
    Exit Sub
End If

'calculamos la pérdida de carga, el volumen interno,
'la velocidad del agua por los tubos, la superficie
Dim rscoef As Recordset
Dim critcoef As String
Dim nvent, nuni As Integer
Dim perdidacalor, perdidafrio, Vw As Double
Dim perdidaprev As Double
Dim numventprima As Integer
Dim S, factorcito As Double
Dim cadena, sincoma, bateriafinal As String
```

```
numero = Val(Text1(2).Text)
factorcito = Val(Text1(3).Text)
nuni = Val(Text1(1).Text)
If datosya = True Then
    Qw = Val(Text2(1).Text)
    n = Val(Text1(2).Text)
End If
Qwfin = Qw / nuni
Vw = Qwfin / (n * 2827.43338823 * Dtubos ^ 2)

perdidaprev = (coef1 * ((Vw) ^ coef2) * T * r * ((Asegunda / 1000)
+ coef3) / n)
perdidacalor = perdidaprev * factorcito
perdidafrio = (perdidacalor * coef4 / coef1)

If datosya = True Then
    N_modulos = Val(Text5(3).Text)
End If

nvent = Val(Text1(0).Text)
numventprima = nuni * nvent
margen = ((numventprima / N_modulos) - 1) * 100

S = coef5 * T * r * Asegunda / (paso * 1000)
Vint = 1.05 * c1vint * T * r * (Asegunda / 1000 + c2vint)

solucion = True

'lo sacamos por pantalla
cadena = Format(perdidacalor, "0.00")
sincoma = quitarcoma(cadena)
Text6(0).Text = sincoma

cadena = Format(perdidafrio, "#0.00")
sincoma = quitarcoma(cadena)
Text6(1).Text = sincoma

cadena = Format(margen, "##0.0")
sincoma = quitarcoma(cadena)
Text6(4).Text = sincoma
```

```
cadena = Format(S, "##0.00")
sincoma = quitarcoma(cadena)
Text6(5).Text = sincoma
```

```
cadena = Format(Vint, "##0.00")
sincoma = quitarcoma(cadena)
Text6(3).Text = sincoma
```

```
cadena = Format(Vw, "###0.00")
sincoma = quitarcoma(cadena)
Text6(2).Text = sincoma
```

```
bateriafinal = "" & numerobat & "-W-" & T & "T-" & r & "R-" _
    & "" & Asegunda & "A-" & paso & "P-" & n & "NC-" &
    tubosaletas & ""
Text7.Text = bateriafinal
solucion = False
```

```
tercercalculo = True
cambiado = True
'habilito el botón de guardar el cálculo una vez que ya
'ha terminado el mismo y si en los combos de cliente,
'proyecto o item hay algún "Nuevo"
If (nitem = "Nuevo") Or (nempresa = "Nuevo") Or (nproy = "Nuevo")
Then
```

```
    Command7.Enabled = True
End If
GoTo finn
quepasa:
    MsgBox "Debe hacer los cálculos anteriores", 16
    Exit Sub
finn:
End Sub
```

1.20.Command8_Click

En esta subrutina mostramos más características de la batería, como son FC1, la densidad, el calor específico, la temperatura media del fluido, el caudal por módulo, el incremento de temperatura máximo y el incremento de temperatura.

Private Sub Command8_Click()

'mostramos más variables del cálculo

Dim frmM As New frmM

cadena = Format(densidad, "#0.0")

sincoma = quitarcoma(cadena)

frmM.Text1(0).Text = sincoma

cadena = Format(calorespecifico, "#0.0000")

sincoma = quitarcoma(cadena)

frmM.Text1(1).Text = sincoma

cadena = Format(0.5 * (Te + Tu), "#0.0")

sincoma = quitarcoma(cadena)

frmM.Text1(2).Text = sincoma

cadena = Format(Qi)

sincoma = quitarcoma(cadena)

frmM.Text1(3).Text = sincoma

cadena = Format(FC1, "#0.00")

sincoma = quitarcoma(cadena)

frmM.Text1(4).Text = sincoma

```
cadena = Format(DELTA, "#0.00")
sincoma = quitarcoma(cadena)
frmM.Text1(5).Text = sincoma

If Text2(5).Text <> "" Then
    incrementoOk = DELTA / (FC1 * Val(Text2(5).Text))
Else:
    MsgBox "No tenemos coeficiente FC2", 16
    Exit Sub
End If

cadena = Format(incrementoOk, "#0.00")
sincoma = quitarcoma(cadena)
frmM.Text1(6).Text = sincoma
frmM.Show vbModal

End Sub
```

1.21. Text1_Lostfocus.

El suceso *Getfocus* se da al escribir o para el tabulador en una caja de texto o botón. Al apretar de nuevo el tabulador o pinchar con el ratón en otro lugar, la caja de tecto anterior pierde el foco, es decir, se da el suceso *Lostfocus*.

Text1 es un array de controles:

El de índice 0 es el número de ventiladores. Al darse el suceso Lostfocus, se debe comprobar si el número de módulos está en la base de datos para el tipo de batería, diámetro y rango escogido.

El de índice 1 es el número de unidades.

El de índice 2 es el número de circuitos.

El de índice 4 es el número de tubos por circuito.

El de índice 3 es el factor para la pérdida de carga.

Private Sub Text1_LostFocus(Index As Integer)

Dim cadena, sincoma As String

If Text1(2).Text <> "" Or Text1(4).Text <> "" Then
 r = Val(Combo1.Text)

Select Case Index

*'si es el número de circuitos podremos calcular
 'el número de tubos por circuito.*

Case 2:

If Text1(2).Text <> "" Then
 n = Val(Text1(2).Text)
 valorn = T * r / n
 solucion = True
 'sacamos por pantalla
 cadena = Format(valorn, "0.0")
 sincoma = quitarcoma(cadena)
 Text1(4).Text = sincoma
 solucion = False

End If

*'si es el número de tubos por circuito, calculamos
 'el número de circuitos.*

Case 4:

If Text1(4).Text <> "" Then
 valorn = Val(Text1(4).Text)
 n = T * r / valorn
 solucion = True

```
        'sacamos el valor por pantalla
        cadena = Format(n, "#0.")
        sincoma = quitarcoma(cadena)
        Text1(2).Text = sincoma
        solucion = False
    End If

    'si es el número de módulos, lo buscamos en la base
    'datos.
    Case 0:
        If Text1(0).Text = "" Then
            Text5(4).Text = ""
            Text5(4).Visible = False
            Label8(22).Visible = False
        End If

        End Select
    End If
    If (Index = 0) And (Text1(0).Text <> "") Then
        Call buscaenbd
    End If
    If Index = 1 And Text1(1).Text <> "" Then
        unidad = Val(Text1(1).Text)
    End If

End Sub
```

1.21. Buscaenbd.

Sirve a la rutina anterior para buscar el número de módulos en la base de datos.

```
Sub buscaenbd()
    Dim otrabdaeros As Database
    Dim rsmodulos As Recordset
    Dim nmodulos As Integer
    Dim crita As String
    Set otrabdaeros = OpenDatabase("C:\aeros\aedros.mdb")
```

```
'la consulta es
crita = " SELECT numeromodulos,Asegunda FROM modulos
WHERE ( idbatdiam = " & idbatdiam & " )"
nmodulos = Val(Text1(0).Text)

Set rsmodulos = otrabdaeros.OpenRecordset(crita, dbOpenDynaset)
rsmodulos.MoveFirst
Do While Not rsmodulos.EOF
    If nmodulos <> rsmodulos.Fields("numeromodulos") Then
        rsmodulos.MoveNext
    Else:
        Asegunda = rsmodulos.Fields("Asegunda")
        cadena = Format(Qi * nmodulos)
        sincoma = quitarcoma(cadena)
        Text5(4).Text = sincoma
        Text5(4).Visible = True
        Label8(22).Visible = True
        buscomod = True
        segundocalculo = True
        Exit Sub
    End If
Loop
' si llega aquí no lo ha encontrado en la base
' de datos, luego mensaje de error

MsgBox " El número de ventiladores introducido no es
        correcto", 16
Text1(0).Text = ""

End Sub
```

1.23. Command7_Click

Pasamos a guardar el cálculo. Tendremos que saber si los datos que son obligatorios están. Desde aquí llamará a la subrutina que se encarga de saber si el item, el proyecto o el cliente son nuevos.

Private Sub Command7_Click()

'el usuario quiere guardar el cálculo

```
If combofluido.Text = "" Then
    MsgBox "Debe escoger un fluido", 16
    Exit Sub
End If
```

```
If Combo2.Text = "" Then
    MsgBox "Debe escoger un tipo de diámetro de la batería", 16
    Exit Sub
End If
```

```
If Combo3.Text = "" Then
    MsgBox "Debe escoger un tipo de batería", 16
    Exit Sub
End If
```

```
If Combo1.Text = "" Then
    MsgBox "Debe escoger un rango", 16
    Exit Sub
End If
```

Call queguardo

End Sub

1.24.queguardo.

Esta subrutina comprueba cuál es nuevo.

Sub queguardo()

*'subrutina con la que sabemos si hay que guardar todo,
'el proyecto y el item o sólo el item.*

'dependiendo de esto, se muestra lo necesario:

'la caja de texto del item, la de proyecto y la del cliente

'definimos un objeto que es el formulario

'fguardardt.

Dim frmguardar As New fguardardt

```
If combocliente.Text <> "Nuevo" Then
  If comboproj.Text <> "Nuevo" Then
    'es un item nuevo dentro del proyecto
    'sólo enseñe la frame del item
    itemnuevo = True
    frmguardar.Frame1.Visible = False
    frmguardar.Frame2.Visible = False
    frmguardar.Height = 2400
    frmguardar.Frame3.Move 50, 400
    frmguardar.soloitem = True

    'actualizamos el valor de los identificadores
    'de proyecto e item para dirigirnos a la tabla
    frmguardar.idc = idempresa
    frmguardar.idp = idproy
  Else:
    'es un proyecto nuevo dentro del cliente
    'y un item sólo enseñe la frame del item
    'y del proyecto
    proyectonuevo = True
    frmguardar.Frame1.Visible = False

    frmguardar.Height = 3000
    frmguardar.Frame2.Move 50, 200
    frmguardar.Frame3.Move 50, 1200
    frmguardar.itemproy = True
    'actualizamos el identificador del cliente
    frmguardar.idc = idempresa
    clickproy = True
  End If
Else:
  clientenuevo = True
  frmguardar.tres = True
  frmguardar.itemproy = True
End If

'mostramos el formulario
frmguardar.Show vbModal
'hay que saber si se ha cancelado guardar los cálculos
If Tpasos.Text <> "0" Then
  'nos basamos en el texto de la caja de texto
  'Tpasos: si es un cero es que no se ha cancelado
  'se puede meter en la base de datos
  If clientenuevo = True Then
    'llamamos a la subrutina que se encarga de
```

```
'guardar el cálculo
Call guardartodo
'actualizamos las variables booleanas para
'siguientes operaciones
guardado = True
cambiado = False
clientenuevo = False
Exit Sub
End If
If proyectonuevo = True Then
'la surutina guardaproitem guarda los datos
'correspondientes al proyecto y al item
Call guardaproitem

'actualizamos las variables booleanas para
'siguientes operaciones
guardado = True
cambiado = False
proyectonuevo = False
Exit Sub
End If
If itemnuevo = True And sustituido = False Then
'solo gurda los datos del item
Call guardaitem

'actualizamos las variables booleanas para
'siguientes operaciones
guardado = True
cambiado = False
itemnuevo = False

End If

'liberamos la memoria correspondiente al objeto frmguardar
Set frmguardar = Nothing
End If
sustituido = False

End Sub
```

1.25.guardartodo.

Sub guardartodo()

*'en esta subrutina sabemos que tenemos que guardar
'el cliente, el proyecto y el ítem*

Dim rsguardando As Recordset
Dim guardoemp, guardoproj As Integer

*'abre un recordset con la tabla tablacliente
'añade uno nuevo y guarda el identificador*

Set rsguardando = bdclientes.OpenRecordset("tablacliente")

rsguardando.MoveLast
rsguardando.AddNew
If nombrepersona = "" Then
 rsguardando.Fields("npersona") = "0"
Else:
rsguardando.Fields("npersona") = nombrepersona
End If

combocliente.Text = nombrecliente
combocliente.AddItem nombrecliente
rsguardando.Fields("nempresa") = nombrecliente

If telefono <> "" Then
 rsguardando.Fields("telefono") = telefono
Else:
 rsguardando.Fields("telefono") = "0"
End If

If direccion <> "" Then
 rsguardando.Fields("direccion") = direccion
Else:
 rsguardando.Fields("direccion") = "0"
End If

If fax <> "" Then
 rsguardando.Fields("fax") = fax
Else:
 rsguardando.Fields("fax") = "0"

End If

rsguardando.Update
rsguardando.MoveNext

guardoemp = rsguardando.Fields("Idcliente")
idempresa = guardoemp
rsguardando.Close

*'abre un recordset con la tabla tablaproyecto
'añade uno nuevo y guarda también la correspondencia
'con el cliente*

Dim rsgproy As Recordset
Dim elproy As String

Set rsgproy = bdclientes.OpenRecordset("tablaproyecto")
rsgproy.MoveLast
rsgproy.AddNew
rsgproy.Fields("Idcliente") = guardoemp
rsgproy.Fields("nproyecto") = nombreproy
comboproj.Text = nombreproy
comboproj.AddItem nombreproy
rsgproy.Update
rsgproy.MoveNext
elproy = rsgproy.Fields("Idproyecto")
idproy = elproy
clickproy = True
rsgproy.Close

'hace lo mismo con los datos del item

Dim elitem As String
Set rsgitem = bdclientes.OpenRecordset("tablaitem")
rsgitem.MoveLast
rsgitem.AddNew
rsgitem.Fields("Idproyecto") = elproy
rsgitem.Fields("Idcliente") = guardoemp
rsgitem.Fields("nitem") = nombreitem
comboitem.Text = nombreitem
iditem = rsgitem.Fields("Iditem")

*'con esta subrutina guardamos todos los datos de la
'pantalla de cálculo y algunos más que serán necesarios
'para el documento de la oferta*

Call guardopantalla
comboitem.AddItem nombreitem

End Sub

1.26. *guardaproitem.*

Guarda los datos del proyecto y del item. El cliente ya existía.

Sub *guardaproitem()*

*'subrutina encargada de guardar los datos del item y
'del proyecto. Abre un recordset con los identificadores
'del cliente, añade un registro nuevo y acaba
'guardando los datos, tras coger el identificador del
'item activo*

Dim rbusca As Recordset
Dim busca As String
elcliente = combocliente.Text

Dim elcliente2 As String
Set rbusca = bdclientes.OpenRecordset("tablacliente",
dbOpenDynaset)

Do While Not rbusca.EOF
elcliente2 = rbusca.Fields("nempresa")

*'encuentra el identificador del cliente
'lo guarda en idempresa*
If elcliente2 = elcliente Then
idempresa = rbusca.Fields("Idcliente")
Exit Do
Else:
rbusca.MoveNext
End If

Loop

Dim rsbproy As Recordset
*'abre un recordset de la tabla tablaproyecto
'va al final, añade un registro nuevo
'y guarda el identificador del cliente y el nombre
'del proyecto. Asigna el identificador del proyecto
'en idproy*
Set rsbproy = bdclientes.OpenRecordset("tablaproyecto",
dbOpenDynaset)
rsbproy.MoveLast

```
rsbproy.AddNew
rsbproy.Fields("Idcliente") = idempresa
rsbproy.Fields("nproyecto") = nombreproy
'añade el nuevo proyecto al combo de los proyectos
' y actualiza el texto del mismo
comboproj.Text = nombreproy
comboproj.AddItem nombreproy
rsbproy.Update
rsbproy.MoveNext
idproy = rsbproy.Fields("Idproyecto")
rsbproy.Close

' abre otro para el item.
' guarda los identificadores del proyecto y cliente
' y el nombre del item
Set rsgitem = bdclientes.OpenRecordset("tablaitem", dbOpenDynaset)
rsgitem.MoveLast
rsgitem.AddNew
rsgitem.Fields("Idcliente") = idempresa
rsgitem.Fields("Idproyecto") = idproy
rsgitem.Fields("nitem") = nombreitem

'añade el nuevo item al combo de los items
' y actualiza el texto del mismo

comboitem.Text = nombreitem
comboitem.AddItem nombreitem
iditem = rsgitem.Fields("Iditem")

' con esta subrutina guardamos todos los datos de la
' pantalla de cálculo y algunos más que serán necesarios
' para el documento de la oferta
Call guardopantalla

End Sub
```

1.27. guardaitem.

Sub guardaitem()

*'subrutina encargada de guardar los datos del item
'abre un recordset con los identificadores del
'proyecto y del item, añade un registro nuevo y acaba
'guardando los datos, tras coger el identificador del
'item activo*

```
Set rsgitem = bdclientes.OpenRecordset("tablaitem", dbOpenDynaset)
rsgitem.MoveLast
rsgitem.AddNew
rsgitem.Fields("Idproyecto") = idproy
rsgitem.Fields("Idcliente") = idempresa
rsgitem.Fields("nitem") = nombreitem
comboitem.Text = nombreitem
comboitem.AddItem nombreitem
iditem = rsgitem.Fields("Iditem")
```

*'con esta subrutina guardamos todos los datos de la
'pantalla de cálculo y algunos más que serán necesarios
'para el documento de la oferta*

Call guardopantalla

End Sub

1.28.guardopantalla.

Sub guardopantalla()

*'en esta subrutina guardamos todos los datos
'importantes del cálculo
'es posible que el usuario no desee guardar todos
'un cálculo completo, de forma que si en algún
'sitio no hay dato, mete xxx en el campo correspon
'diente de la tabla*

```
If combofluido.Text = "" Then
```

```
    rsgitem.Fields("nfluido") = "xxx"  
Else:  
    rsgitem.Fields("nfluido") = combofluido.Text  
End If
```

```
If Text2(0).Text = "" Then  
    rsgitem.Fields("lpotencia") = "xxx"  
Else:  
    rsgitem.Fields("lpotencia") = Text2(0).Text  
End If
```

```
If Text2(1).Text = "" Then  
    rsgitem.Fields("lcaudal") = "xxx"  
Else:  
    rsgitem.Fields("lcaudal") = Text2(1).Text  
End If
```

```
If Text2(2).Text = "" Then  
    rsgitem.Fields("lTin") = "xxx"  
Else:  
    rsgitem.Fields("lTin") = Text2(2).Text  
End If
```

```
If Text2(3).Text = "" Then  
    rsgitem.Fields("lTOut") = "xxx"  
Else:  
    rsgitem.Fields("lTOut") = Text2(3).Text  
End If
```

```
If Text2(5).Text = "" Then  
    rsgitem.Fields("FC2") = "xxx"  
Else:  
    rsgitem.Fields("FC2") = Text2(5).Text  
End If
```

```
If Text2(4).Text = "" Then  
    rsgitem.Fields("lmargen") = "xxx"  
Else:  
    rsgitem.Fields("lmargen") = Text2(4).Text  
End If
```

```
If Text3(0).Text = "" Then  
    rsgitem.Fields("presion") = "xxx"  
Else:  
    rsgitem.Fields("presion") = Text3(0).Text
```


End If

```
If Text3(1).Text = "" Then
    rsgitem.Fields("aTin") = "xxx"
Else:
    rsgitem.Fields("aTin") = Text3(1).Text
End If
```

```
If Text3(2).Text = "" Then
    rsgitem.Fields("avar") = "xxx"
Else:
    rsgitem.Fields("avar") = Text3(2).Text
End If
```

```
If Option1(0).Value = True Then
    rsgitem.Fields("atipovar") = 1
End If
If Option1(1).Value = True Then
    rsgitem.Fields("atipovar") = 2
End If
If Option1(2).Value = True Then
    rsgitem.Fields("atipovar") = 3
End If
```

```
If Combo2.Text = "" Then
    rsgitem.Fields("tipodiam") = "xxx"

Else:
    rsgitem.Fields("tipodiam") = Combo2.Text
End If
```

```
If Combo3.Text = "" Then
    rsgitem.Fields ("tipobat")

Else:
    rsgitem.Fields("tipobat") = Combo3.Text
End If
```

```
If Combo1.Text = "" Then
    rsgitem.Fields("nlineas") = "xxx"
Else:
    rsgitem.Fields("nlineas") = Combo1.Text
End If
```

```
If Text1(2).Text = "" Then
    rsgitem.Fields("varcir") = "xxx"
```

```
Else:  
    cadena = Format(Text1(2).Text, "#.0")  
    rsgitem.Fields("varcir") = quitarcoma(cadena)  
End If
```

```
If Text1(4).Text = "" Then  
    rsgitem.Fields("tuboscirc") = "xxx"  
Else:  
    cadena = Format(Text1(4).Text, "#.0")  
    rsgitem.Fields("tuboscirc") = quitarcoma(cadena)  
End If
```

```
If Text1(3).Text = "" Then  
    rsgitem.Fields("factor") = "xxx"  
Else:  
    rsgitem.Fields("factor") = Text1(3).Text  
End If
```

```
If Text5(0).Text = "" Then  
    rsgitem.Fields("Pi") = "xxx"  
Else:  
    rsgitem.Fields("Pi") = Text5(0).Text  
End If
```

```
If Text5(1).Text = "" Then  
    rsgitem.Fields("Tssalida") = "xxx"  
Else:  
    rsgitem.Fields("Tssalida") = Text5(1).Text  
End If
```

```
If Text6(4).Text = "" Then  
    rsgitem.Fields("margenreal") = "xxx"  
Else:  
    rsgitem.Fields("margenreal") = Text6(4).Text  
End If
```

```
If Text1(0).Text = "" Then  
    rsgitem.Fields("nmodulos") = "xxx"  
Else:  
    rsgitem.Fields("nmodulos") = Text1(0).Text  
End If
```

```
If Text1(1).Text = "" Then  
    rsgitem.Fields("nunidad") = "xxx"  
Else:  
    rsgitem.Fields("nunidad") = Text1(1).Text
```

End If

If Qi = 0 Then
 rsgitem.Fields("caudalmod") = "xxx"
Else:
 rsgitem.Fields("caudalmod") = Format(Qi)
End If

If Text6(0).Text = "" Then
 rsgitem.Fields("pcc") = "xxx"
Else:
 rsgitem.Fields("pcc") = Text6(0).Text
End If

If Text6(1).Text = "" Then
 rsgitem.Fields("pcf") = "xxx"
Else:
 rsgitem.Fields("pcf") = Text6(1).Text
End If

If DELTA = 0 Then
 rsgitem.Fields("DT/DTm") = "xxx"
Else:
 cadena = Format(DELTA, "#.0")
 rsgitem.Fields("DT/DTm") = quitarcoma(cadena)
End If

If FC1 = 0 Then
 rsgitem.Fields("FC1") = "xxx"
Else
 cadena = Format(FC1, "#.00")
 rsgitem.Fields("FC1") = quitarcoma(cadena)
End If

If densidad = 0 Then
 rsgitem.Fields("densidad") = "xxx"
Else:
 cadena = Format(densidad, "000")
 rsgitem.Fields("densidad") = quitarcoma(cadena)
End If

If calorespecifico = 0 Then
 rsgitem.Fields("Cp") = "xxx"

```
Else:  
  cadena = Format(calorespecifico, "0.000")  
  rsgitem.Fields("Cp") = quitarcoma(cadena)  
End If
```

```
If Text6(5).Text = "" Then  
  rsgitem.Fields("S") = "xxx"  
Else:  
  rsgitem.Fields("S") = Text6(5).Text  
End If
```

```
If Text5(3).Text = "" Then  
  rsgitem.Fields("ventcalc") = "xxx"  
Else:  
  rsgitem.Fields("ventcalc") = Text5(3).Text  
End If
```

```
If combocol.Text = "" Then  
  rsgitem.Fields("diamcol") = "xxx"  
Else:  
  rsgitem.Fields("diamcol") = combocol.Text  
End If
```

```
If Text4(0).Text = "" Then  
  rsgitem.Fields("ncol") = "xxx"  
Else:  
  rsgitem.Fields("ncol") = Text4(0).Text  
End If
```

```
If Text4(1).Text = "" Then  
  rsgitem.Fields("Vcol") = "xxx"  
Else:  
  rsgitem.Fields("Vcol") = Text4(1).Text  
End If
```

```
If Text6(3).Text = "" Then  
  rsgitem.Fields("Vint") = "xxx"  
Else:  
  rsgitem.Fields("Vint") = Text6(3).Text  
End If
```

```
If Text6(2).Text = "" Then  
  rsgitem.Fields("Vw") = "xxx"  
Else:  
  rsgitem.Fields("Vw") = Text6(2).Text
```

End If

```
If incrementoH = 0 Then
    rsgitem.Fields("inremH") = "xxx"
Else:
    cadena = Format(incrementoH, "0.000")
    rsgitem.Fields("inremH") = quitarcoma(cadena)
End If
```

```
If Text7.Text = "" Then
    rsgitem.Fields("batfin") = "xxx"
Else:
    rsgitem.Fields("batfin") = Text7.Text
End If
```

```
If denom = "" Then
    rsgitem.Fields("denom") = "xxx"
Else:
    rsgitem.Fields("denom") = denom
End If
```

```
If DN = "" Then
    rsgitem.Fields("DN") = "xxx"
Else:
    rsgitem.Fields("DN") = DN
End If
```

```
If revoluciones = "" Then
    rsgitem.Fields("revoluciones") = "xxx"
Else:
    rsgitem.Fields("revoluciones") = revoluciones
End If
```

```
If potencia = "" Then
    rsgitem.Fields("potencia") = "xxx"
Else:
    rsgitem.Fields("potencia") = potencia
```

End If

```
If densiaire = "" Then
    rsgitem.Fields("densiaire") = "xxx"
Else:
    rsgitem.Fields("densiaire") = densiaire
End If
```

```
If HRent = "" Then
  rsgitem.Fields("HR") = "xxx"
Else:
  cadena = Format(HRent, "##")
  rsgitem.Fields("HR") = cadena
End If
If tubosaletas = "" Then
  rsgitem.Fields("tubaletas") = "xxx"
Else:
  rsgitem.Fields("tubaletas") = tubosaletas
End If

rsgitem.Fields("paso") = paso

rsgitem.Update
rsgitem.Close

End Sub
```

1.29. resolución_TS_incremento.

Con esta rutina calculamos la temperatura de salida del aire húmedo, aplicando el método de la bisección , aplicando la ecuación:

$$\frac{\Delta T}{\Delta T_m} = \frac{TS'' - TS'}{(Te - TS'') - (Tu - TS')} \cdot \frac{1}{\ln \left[\frac{(Te - TS'')}{(Tu - TS')} \right]}$$

Donde calcularemos TS'' al conseguir el mismo valor es los dos miembros de la ecuación.

**Function resolucion_TS_incremento(incrementoOk, Te, Tu, TS)
As Double**

*'Esta función resuelve la temperatura de salida
'del aire*

Dim encontrado, i As Integer
Dim X_sup, X_inf, X_media, result_anterior, result_media, error,
punto, signo1, signo2 As Double

X_sup = Te - 0.5

X_inf = TS + 0.5

encontrado = 0

result_anterior = 0

Do While i < 10000

 X_media = 0.5 * (X_sup + X_inf)

 punto = X_media

**result_media_TS = valor_ecuacion(punto, incrementoOk, Te,
 Tu, TS)**

 If valormalo = False Then

 error = result_media_TS - result_anterior

 If error < 0 Then error = error * (-1)

 If error < 0.0001 Then

 encontrado = 1

resolucion_TS_incremento = X_media

 Exit Do

 End If

 punto = X_sup

**result_sup = valor_ecuacion(punto, incrementoOk, Te,
 Tu, TS)**

 If valormalo = False Then

 punto = X_inf

**result_inf = valor_ecuacion(punto, incrementoOk, Te,
 Tu, TS)**

 If valormalo = False Then

 signo1 = result_media_TS * result_sup

 If signo1 < 0 Then

 X_inf = X_media

 End If

 signo2 = result_media_TS * result_inf

```
        If signo2 < 0 Then
            X_sup = X_media
        End If
        If signo1 > 0 Then
            X_sup = X_media
        End If
        result_anterior = result_media_TS
    End If
End If
End If
```

Loop

End Function

1.30. valor_ecuacion.

Function valor_ecuacion(punto, incrementoOk, Te, Tu, TS)
As Double

'calcula el valor que tiene la función determinada
'en tmW para calcular la temperatura de salida del aire

Dim loga, a, b, incremT, DTm As Double

a = Te - punto

b = Tu - TS

If (a <> b) And b <> 0 And a <> 0 Then

DTm = (a - b) / Log(a / b)

Else:

punto = punto + 1

a = Te - punto

b = Tu - TS

DTm = (a - b) / Log(a / b)

End If

If a = 0 Then

punto = punto + 1

a = Te - punto

b = Tu - TS

DTm = (a - b) / Log(a / b)

End If

```
incrementT = punto - TS
valor_ecuacion = (incrementT / DTm) - incrementoOk
GoTo final
error:
  MsgBox "el punto introducido no es correcto", 16
  End
final:
```

End Function

1.31. Combo2_Click.

Es el combo correspondiente al tipo de diámetro.

Private Sub Combo2_Click()

```
Combo3.Clear
'ha escogido diámetro
If Combo2.ListIndex <> -1 Then
  If segundocalculo = True Then
    cambiado = True
    Qw = Val(Text2(1).Text)
    Te = Val(Text2(2).Text)
    Tu = Val(Text2(3).Text)
    Pot = Val(Text2(0).Text)
    Text1(0).Text = ""
    Text1(1).Text = "1"
    Text1(2).Text = ""
    Text1(3).Text = "1"
    Text1(4).Text = ""
    For i = 0 To 5
      Text6(i).Text = ""
    Next i
    For i = 0 To 3
      Text5(i).Text = ""
    Next i
    Text5(4).Text = ""
```

```
Text5(4).Visible = False
Label8(22).Visible = False
Text7.Text = ""
Text4(0).Text = Format(1)
Text4(1).Text = ""
combocol.Text = ""
```

```
End If
```

```
diamventil = Combo2.Text
'llama a llena el combo de las baterías
```

```
Call llenobateria
```

```
End If
```

```
End Sub
```

1.32.Combo3_Click.

Es el combo de las baterías.

```
Private Sub Combo3_Click()
```

```
'ha escogido batería
If Combo3.ListIndex <> -1 Then
  If segundocalculo = True Then
    cambiado = True
    Qw = Val(Text2(1).Text)
    Te = Val(Text2(2).Text)
    Tu = Val(Text2(3).Text)
    Pot = Val(Text2(0).Text)
    Text1(0).Text = ""
    Text1(1).Text = "1"
    Text1(2).Text = ""
    Text1(3).Text = "1"
    Text1(4).Text = ""
    For i = 0 To 5
      Text6(i).Text = ""
    Next i
    For i = 0 To 3
      Text5(i).Text = ""
    Next i
```

```
Text5(4).Text = ""
Text5(4).Visible = False
Label8(22).Visible = False
Text7.Text = ""
Text4(0).Text = Format(1)
Text4(1).Text = ""
combocol.Text = ""
```

End If

Combo1.Clear

'llama a llenar las líneas

Call llenolineas

End If

End Sub

1.33. Combo1_Click.

Ha escogido el rango.

Private Sub Combo1_Click()

'el usuario ha escogido el rango

'esta subrutina actualiza alguns variables necesarias

Dim filas As Integer

Dim rsaerosfil, rsaer As Recordset

Dim busco As String

filas = Val(Combo1.Text)

If Combo1.ListIndex <> -1 Then

 If segundocalculo = True Then

 'borra lo posterior si se ha cambiado

 cambiado = True

 Qw = Val(Text2(1).Text)

 Te = Val(Text2(2).Text)

 Tu = Val(Text2(3).Text)

 Pot = Val(Text2(0).Text)

 Text1(0).Text = ""

 Text1(1).Text = "1"

```
Text1(2).Text = ""
Text1(3).Text = "1"
Text1(4).Text = ""
For i = 0 To 5
    Text6(i).Text = ""
Next i
For i = 0 To 3
    Text5(i).Text = ""
Next i
Text5(4).Text = ""
Text5(4).Visible = False
Label8(22).Visible = False
Text7.Text = ""
Text4(0).Text = Format(1)
Text4(1).Text = ""
combocol.Text = ""
```

End If

'crea una consulta con la batería y el diámetro

'escogidos y el rango

```
busco = "SELECT [DT/DTm],[caudal] FROM datos WHERE" _
    & " (numerolíneas = " & filas & " ) AND (Idbatdiam = " &
    idbatdiam & ")"
```

```
Set rsaerosfil = bdaeros.OpenRecordset(busco, dbOpenDynaset)
```

'sólo hay uno que lo cumple

```
Qi = rsaerosfil.Fields("caudal")
```

```
DELTA = rsaerosfil.Fields("DT/DTm")
```

```
Command4.Enabled = True
```

End If

End Sub

1.34. Combocliente_Dropdown.

Este suceso se da al apretar la flecha del combo, antes de que la lista de los items del combo se muestre. En nuestro caso es útil para controlar si el usuario no ha guardado el cálculo y por error, por ejemplo, hace click en alguno de los combos de cliente, proyecto o

item. Es proceso para los tres es el mismo, con lo que mostraré solamente uno de ellos.

Private Sub combocliente_DropDown()

```
'se da al apretar el combocliente  
If guardado = False And cambiado = True And cancelar = False Then  
    'si el cálculo ha cmabiado y no ha sido cambiado  
    'pregunta si quiere hacerlo  
    respuesta = MsgBox("¿quiere guardar el cálculo?", vbOKCancel +  
        vbCritical + vbDefaultButton1, "aeros")  
    If respuesta = 2 Then  
        guardado = False  
        cancelar = True  
        Exit Sub  
    End If  
    If respuesta = 1 Then  
        'hay que guardarlo  
        clientenuevo = True  
        Call queguardo  
    End If  
  
End If  
  
solucion = False  
  
End Sub
```

1.35. Combocliente_Click.

Se da al escoger un elemento del combo.

Private Sub combocliente_Click()

```
'ha escogido un cliente  
solucion = True  
Dim criterio, otrocrit As String
```

```
'borro pantalla
Call clrsc
If combocliente.Text <> "Nuevo" Then
  ' si no es nuevo y ha apretado bien comienza
  ' el proceso de ver qué proyectos corresponden
  ' al cliente
  If combocliente.ListIndex <> -1 Then
    clickproy = False
    cambiado = False
    guardado = False
    Command5.Enabled = True
    indexcliente = combocliente.ListIndex
    idproy = -1
    iditem = -1

    comboproy.Clear
    comboproy.AddItem "Nuevo"
    comboitem.Clear
    comboproy.Text = "Nuevo"
    comboitem.Text = "Nuevo"
    comboitem.AddItem "Nuevo"
    nempresa = combocliente.Text
    'buscamos el identificador de tipo correspondiente a
    'la empresa escogida
    If nempresa <> "" Then
      clientenuevo = False
      rsclientes.MoveFirst

      Do While Not rsclientes.EOF
        If rsclientes.Fields("nempresa") = nempresa Then
          idempresa = rsclientes.Fields("idcliente")
          If rsclientes.Fields("npersona") = "0" Then
            nperson = ""
          Else:
            nperson = rsclientes.Fields("npersona")
          End If
        End If
      Loop

      Exit Do
    Else:
      rsclientes.MoveNext
    End If
  Loop
```

'hacemos una consulta

```
criterio = " SELECT [nproyecto],[Idproyecto] FROM  
Tablaproyecto WHERE [Idcliente] = " & idempresa & " "
```

*'Llenamos el siguiente combo que corresponde a los
'datos de los proyectos correspondientes al cliente
'escogido*

If idempresa <> 0 Then

```
Set rsproy = bdclientes.OpenRecordset(criterio,  
dbOpenDynaset)
```

Else:

```
Exit Sub
```

End If

Do While Not rsproy.EOF

```
comboproj.AddItem (rsproy.Fields("nproyecto"))
```

```
rsproy.MoveNext
```

Loop

End If

End If

Else:

'el cliente es nuevo

'simplemente actualizamos unas variables

```
clientenuevo = True
```

```
guardado = False
```

```
comboproj.Clear
```

```
comboitem.Clear
```

```
comboproj.Text = "Nuevo"
```

```
comboitem.Text = "Nuevo"
```

```
proyetonuevo = True
```

```
itemnuevo = True
```

```
cambiado = False
```

```
clickproy = False
```

End If

End Sub

1.36. Comboproy_Click.

Private Sub comboproy_Click()

solucion = True

'ha escogido proyecto

Dim criterio, otrocrit As String

Dim rsroy2 As Recordset

Call clrsc

If comboproy.Text <> "Nuevo" Then

 If comboproy.ListIndex <> -1 Then

 guardado = False

 clickproy = True

 iditem = -1

 indexproy = comboproy.ListIndex

 proyetonuevo = False

 comboitem.Clear

 comboitem.Text = "Nuevo"

 comboitem.AddItem "Nuevo"

 nproy = comboproy.Text

'buscamos el identificador de tipo correspondiente a

'la empresa escogida

 Set rsroy2 = bdclientes.OpenRecordset("tablaproyecto",
 dbOpenDynaset)

 rsroy2.MoveFirst

 Do While Not rsroy2.EOF

 If (rsroy2.Fields("nproyecto") = nproy) And

 (rsroy2.Fields("Idcliente") = idempresa) Then

 idproy = rsroy2.Fields("Idproyecto")

 Exit Do

 Else:

 rsroy2.MoveNext

 End If

 Loop

'consulta

criterio = " SELECT [nitem],[Iditem] FROM Tablaitem WHERE
 [Idproyecto] = " & idproy & " "

'Llenamos el siguiente combo que corresponde a los

*'datos de los item correspondientes al proyecto
'escogido*

```
    If idproy <> 0 Then  
        Set rsitem = bdclientes.OpenRecordset(criterio,  
            dbOpenDynaset)
```

```
    Else:  
        Exit Sub  
    End If
```

```
    Do While Not rsitem.EOF  
        comboitem.AddItem (rsitem.Fields("nitem"))  
        rsitem.MoveNext
```

```
    Loop
```

```
End If
```

```
Else:
```

```
    proyectonuevo = True  
    comboitem.Clear  
    comboitem.Enabled = True  
    comboitem.Text = "Nuevo"  
    comboitem.AddItem "Nuevo"  
    guardado = False  
    clickproy = False
```

```
End If
```

```
End Sub
```

1.37. Comboitem_Click.

En esta rutina buscamos el item escogido y mostramos el cálculo almacenado en la base de datos por pantalla. Si en la base de datos hay "xxx" significa que no teníamos nada almacenado y no saca por pantalla.

```
Private Sub comboitem_Click()
    'se ha escogido un item
    Dim buscar, cadena, sincoma As String
    Dim rsitem2 As Recordset
    Call clrsc
    If comboitem.Text <> "Nuevo" Then
        'si no es nuevo lo busca en la base de datos
        'y saca los datos por pantalla
        solucion = True
        solomuestro = True
        clickitem = True
        itemnuevo = False
        indexitem = comboitem.ListIndex
        If indexitem <> -1 Then
            guardado = False
            Set rsitem2 = bdclientes.OpenRecordset("Tablaitem",
                dbOpenDynaset)

            nitem = comboitem.Text

            Do While Not rsitem2.EOF
                If rsitem2.Fields("nitem") = nitem And
                    rsitem2.Fields("Idproyecto") = idproy And rsitem2.Fields("Idcliente")
                    = idempresa Then
                    Exit Do
                Else:
                    rsitem2.MoveNext
                End If
            Loop
            iditem = rsitem2.Fields("Iditem")

            If rsitem2.Fields("Ipotencia") = "xxx" Then
                Text2(0).Text = ""
            Else:
                cadena = rsitem2.Fields("Ipotencia")
                sincoma = quitarcoma(cadena)
                Text2(0).Text = sincoma
                Pot = Val(sincoma)
            End If

            If rsitem2.Fields("Icaudal") = "xxx" Then
                Text2(1).Text = ""
            Else:
                cadena = rsitem2.Fields("Icaudal")
                sincoma = quitarcoma(cadena)
            End If
        End If
    End If
End Sub
```

```
Text2(1).Text = sincoma  
Qw = Val(sincoma)  
End If
```

```
If rsitem2.Fields("ITin") = "xxx" Then  
    Text2(2).Text = ""  
Else:  
    cadena = rsitem2.Fields("ITin")  
    sincoma = quitarcoma(cadena)  
    Text2(2).Text = sincoma  
    Te = Val(sincoma)  
End If
```

```
If rsitem2.Fields("ITout") = "xxx" Then  
    Text2(3).Text = ""  
Else:  
    cadena = rsitem2.Fields("ITout")  
    sincoma = quitarcoma(cadena)  
    Text2(3).Text = sincoma  
    Tu = Val(sincoma)  
End If
```

```
If rsitem2.Fields("FC2") = "xxx" Then  
    Text2(4).Text = ""  
Else  
    cadena = rsitem2.Fields("FC2")  
    sincoma = quitarcoma(cadena)  
    Text2(5).Text = sincoma  
    FC2 = Val(sincoma)  
End If
```

```
cadena = rsitem2.Fields("Imargen")  
sincoma = quitarcoma(cadena)  
Text2(4).Text = sincoma  
M = Val(sincoma)
```

```
'Dim nombre As String  
'Dim cua, i As Integer  
'combofluido.ListIndex = xxx  
combofluido.Text = rsitem2.Fields("nfluido")  
'nombre = combofluido.Text
```

```
'For i = 1 To combofluido.ListCount - 1  
    'If combofluido.List(i) = nombre Then
```

```
' cua = combofluido.ItemData
' combofluido.ListIndex = cua
End If
Next i

If rsitem2.Fields("presion") = "xxx" Then
    Text3(0).Text = ""
Else:
    cadena = rsitem2.Fields("presion")
    sincoma = quitarcoma(cadena)
    Text3(0).Text = sincoma
    P = Val(sincoma)
End If

If rsitem2.Fields("aTin") = "xxx" Then
    Text3(1).Text = ""
Else:
    cadena = rsitem2.Fields("aTin")
    sincoma = quitarcoma(cadena)
    Text3(1).Text = sincoma
    TS = Val(sincoma)
End If

If rsitem2.Fields("avar") = "xxx" Then
    Text3(2).Text = ""
Else:
    cadena = rsitem2.Fields("avar")
    sincoma = quitarcoma(cadena)
    Text3(2).Text = sincoma
    var3 = Val(sincoma)
End If

If rsitem2.Fields("HR") = "xxx" Then
    HRent = ""
Else:
    HRent = rsitem2.Fields("HR")

End If
tipovar = rsitem2.Fields("atipovar")
Select Case tipovar
    Case 1:
        Option1(0).Value = True
    Case 2:
```

```
Option1(1).Value = True
Case 3:
Option1(2).Value = True
End Select
```

```
Combo2.Text = rsitem2.Fields("tipodiam")
```

```
Combo3.Clear
```

```
Call llenobateria
```

```
Combo3.Text = rsitem2.Fields("tipobat")
```

```
Combo1.Clear
```

```
Call llenolineas
```

```
Combo1.Text = rsitem2.Fields("nlineas")
```

```
Dim filas As Integer
```

```
Dim rsaerosfil As Recordset
```

```
Dim busco As String
```

```
filas = Val(Combo1.Text)
```

```
busco = "SELECT [DT/DTm],[caudal] FROM datos WHERE" _  
    & " (numerolineas = " & filas & " ) AND (Idbatdiam = " &  
    idbatdiam & ")"
```

```
Set rsaerosfil = bdaeros.OpenRecordset(busco, dbOpenDynaset)
```

```
Qi = rsaerosfil.Fields("caudal")
```

```
DELTA = rsaerosfil.Fields("DT/DTm")
```

```
If rsitem2.Fields("tuboscirc") = "xxx" Then
```

```
Text1(4).Text = ""
```

```
Else:
```

```
cadena = rsitem2.Fields("tuboscirc")
```

```
sincoma = quitarcoma(cadena)
```

```
Text1(4).Text = sincoma
```

```
valorn = Val(sincoma)
```

```
End If
```

```
If rsitem2.Fields("ventabajo") = True Then
```

```
Option4(1).Value = True
```

```
Else:
```

```
Option4(0).Value = True
```

```
End If
```

```
If rsitem2.Fields("varcir") = "xxx" Then
```

```
Text1(2).Text = ""
Else:

Text1(2).Text = rsitem2.Fields("varcir")
n = Val(Text1(2).Text)

End If

If rsitem2.Fields("factor") = "xxx" Then
Text1(3).Text = ""
Else:
cadena = rsitem2.Fields("factor")
sincoma = quitarcoma(cadena)
Text1(3).Text = rsitem2.Fields("factor")
End If

If rsitem2.Fields("nmodulos") <> "xxx" Then
Text1(0).Text = rsitem2.Fields("nmodulos")
Call buscaenbd
Else:
Text1(0).Text = ""
End If

If rsitem2.Fields("nunidadas") = "xxx" Then
Text1(1).Text = ""
Else:
Text1(1).Text = rsitem2.Fields("nunidadas")
nuni = Val(Text1(1).Text)
End If

If rsitem2.Fields("Pi") = "xxx" Then
Text5(0).Text = ""
Else:
cadena = rsitem2.Fields("Pi")
sincoma = quitarcoma(cadena)
Text5(0).Text = sincoma
Pi = Val(sincoma)
End If

If rsitem2.Fields("Tssalida") = "xxx" Then
Text5(1).Text = ""
Else:
cadena = rsitem2.Fields("Tssalida")
sincoma = quitarcoma(cadena)
Text5(1).Text = sincoma
```

```
Tsprima = Val(sincoma)
End If
```

```
If rsitem2.Fields("densidad") = "xxx" Then
    densidad = 0
Else:
    cadena = rsitem2.Fields("densidad")
    sincoma = quitarcoma(cadena)
    densidad = Val(sincoma)
End If
```

```
If rsitem2.Fields("Cp") = "xxx" Then
    calorespecifico = 0
Else:
    cadena = rsitem2.Fields("Cp")
    sincoma = quitarcoma(cadena)
    calorespecifico = Val(sincoma)
End If
```

```
If rsitem2.Fields("FC1") = "xxx" Then
    FC1 = 0
Else:
    cadena = rsitem2.Fields("FC1")
    sincoma = quitarcoma(cadena)
    FC1 = Val(sincoma)
End If
```

```
If rsitem2.Fields("ventcalc") = "xxx" Then
    Text5(3).Text = ""
Else:
    cadena = rsitem2.Fields("ventcalc")
    sincoma = quitarcoma(cadena)
    Text5(3).Text = sincoma
    N_modulos = Val(sincoma)
    segundocalculo = True
End If
```

```
If rsitem2.Fields("pcc") = "xxx" Then
    Text6(0).Text = ""
Else:
    cadena = rsitem2.Fields("pcc")
    sincoma = quitarcoma(cadena)
    Text6(0).Text = sincoma
    pcc = Val(sincoma)
```

```
tercercalculo = True  
End If
```

```
If rsitem2.Fields("pcf") = "xxx" Then  
    Text6(1).Text = ""  
Else:  
    cadena = rsitem2.Fields("pcf")  
    sincoma = quitarcoma(cadena)  
    Text6(1).Text = sincoma  
    pcf = Val(sincoma)  
End If
```

```
If rsitem2.Fields("margenreal") = "xxx" Then  
    Text6(4).Text = ""  
Else:  
    cadena = rsitem2.Fields("margenreal")  
    sincoma = quitarcoma(cadena)  
    Text6(4).Text = sincoma  
    margen = Val(sincoma)  
End If
```

```
If rsitem2.Fields("S") = "xxx" Then  
    Text6(5).Text = ""  
Else:  
    cadena = rsitem2.Fields("S")  
    sincoma = quitarcoma(cadena)  
    Text6(5).Text = sincoma  
    S = Val(sincoma)  
End If
```

```
If rsitem2.Fields("ncol") = "xxx" Then  
    Text4(0).Text = ""  
Else:  
    cadena = rsitem2.Fields("ncol")  
    sincoma = quitarcoma(cadena)  
    Text4(0).Text = sincoma  
    ncol = Val(sincoma)  
End If
```

```
If rsitem2.Fields("Vw") = "xxx" Then  
    Text6(2).Text = ""  
Else:  
    cadena = rsitem2.Fields("Vw")  
    sincoma = quitarcoma(cadena)  
    Text6(2).Text = sincoma
```



```
Vw = Val(sincoma)
End If
```

```
If rsitem2.Fields("Vint") = "xxx" Then
    Text6(3).Text = ""
Else:
    cadena = rsitem2.Fields("Vint")
    sincoma = quitarcoma(cadena)
    Text6(3).Text = sincoma
    Vint = Val(sincoma)
End If
```

```
If rsitem2.Fields("Vcol") = "xxx" Then
    Text4(1).Text = ""
Else:
    cadena = rsitem2.Fields("Vcol")
    sincoma = quitarcoma(cadena)
    Text4(1).Text = sincoma
    vcol = Val(sincoma)
End If
```

```
If rsitem2.Fields("incrementH") = "xxx" Then
    Text5(2).Text = ""
Else:
    cadena = rsitem2.Fields("incrementH")
    sincoma = quitarcoma(cadena)
    Text5(2).Text = sincoma
    incrementoH = Val(sincoma)
End If
```

```
If rsitem2.Fields("diamcol") = "xxx" Then
    combocol.Text = ""
Else:
    cadena = rsitem2.Fields("diamcol")
    sincoma = quitarcoma(cadena)
    combocol.Text = sincoma
    denom = combocol.Text
    rscolect.MoveFirst
    cadena = combocol.Text
    denom = solodenominacion(cadena)
```

```
Do While Not rscolect.EOF
```

```
    If rscolect.Fields("denominacion") = denom Then
        DN = rscolect.Fields("DN")
```

```
        diamcolect = rscolect.Fields("diamcolector")
    Exit Do
Else:
    rscolect.MoveNext
End If
Loop

haydiam = True
End If

If rsitem2.Fields("batfin") = "xxx" Then
    Text7.Text = ""
Else:
    Text7.Text = rsitem2.Fields("batfin")
End If

cancelar = False
cambiado = False
guardado = False
datosya = True
If Text2(0).Text <> "" And Text2(1).Text <> "" And
    Text2(2).Text <> "" And Text2(3).Text <> "" Then
    primercalculo = True
End If
End If
solucion = False
Else:
    'si es nuevo sólo actualiza variables
    solomuestro = False
    itemnuevo = True
    guardado = False
    GoTo maslejos
End If
lejos:

maslejos:
End Sub
```

1.38. Text4_Lostfocus.

Esta variable corresponde al número de colectores. Si está escogido ya el colector, saca automáticamente por pantalla la velocidad del fluido por el colector.

```
Private Sub Text4_LostFocus(Index As Integer)  
Dim ncol As Integer  
Dim cadena, sincoma As String  
If datosya = True Then  
  
    Dim unidades, ncole As Double  
  
    ncole = Val(Text4(0).Text)  
    unidades = ncole * unidad  
    Qwfin = Val(Text2(1).Text) / unidades  
End If  
  
If (Index = 0) Then  
    If (Text4(0).Text <> "") And haydiam = True Then  
        vcol = Qwfin / (2827.43338823 * diamcolect ^ 2)  
        cadena = Format(vcol, "#0.##")  
        sincoma = quitarcoma(cadena)  
        Text4(1).Text = sincoma  
    End If  
End If  
  
End Sub
```

1.39. combocol_Click.

Es similar al anterior; Si Text4 tiene algo escrito, al escoger un colector sacará por pantalla la velocidad.

Private Sub combocol_Click()

*'se ha escogido un término del combo de los colectores
'si tenemos texto en el número de colectores podremos
'seguir con los cálculos*

Dim cadena, texto As String
indexcol = combocol.ListIndex

If indexcol <> -1 Then
 cambiado = True
 denom = combocol.Text
 rscolect.MoveFirst
 cadena = combocol.Text
 denom = solodenominacion(cadena)
 If datosya = True Then

 Dim ncol, unidades As Double
 unidad = Val(Text1(1).Text)
 ncol = Val(Text4(0).Text)
 unidades = ncol * unidad
 'hallamos el caudal por colector y por unidad.
 Qwfin = Val(Text2(1).Text) / unidades
 End If

 Do While Not rscolect.EOF
 *'actualizamos variables correspondientes
 'al colector*
 If rscolect.Fields("denominacion") = denom Then
 DN = rscolect.Fields("DN")
 diamcolect = rscolect.Fields("diamcolector")
 Exit Do
 Else:
 rscolect.MoveNext
 End If
 Loop

 haydiam = True
 If (Text4(0).Text <> "") And haydiam = True Then
 vcol = Qwfin / (2827.43338823 * (diamcolect ^ 2) *
 (Val(Text4(0).Text)))
 cadena = Format(vcol, "#0.##")

```
        sincoma = quitarcoma(cadena)
        Text4(1).Text = sincoma
        haydiam = False
    End If
Else:
    haydiam = False
End If

End Sub
```

1.40. combo4_Click.

Para cambiar de usuario o añadir alguno nuevo.

Private Sub Combo4_Click()

*'al apretar un usuario o nuevo se cambia el usuario de
'la aplicación*

```
If Combo4.Text = "Nuevo" Then
    'si es nuevo hay que meterlo en la base de datos
    'crea un objeto y le pide los datos
    Dim fusu As New fusuario
    fusu.Show vbModal

Else:
    'si ya existe hay que cambiar los datos
    Dim rsusuario As Recordset
    Dim nusuario, buscando As String

    nusuario = Combo4.Text
    Set rsusuario = bdclientes.OpenRecordset("tusuario")
    rsusuario.MoveFirst
    Do While Not rsusuario.EOF
        If nusuario = rsusuario.Fields("nombreusuario") Then
            nombreusuario = rsusuario.Fields("nombreusuario")
            telusuario = rsusuario.Fields("tlfusuario")
            faxusuario = rsusuario.Fields("faxusuario")
            Exit Do
        Else:
            rsusuario.MoveNext
        End If
    End Do
End If
```

```
Loop  
End If  
Combo4.Text = nombreusuario
```

End Sub

1.41. fpsicro_Click.

Muestra las variables del psicrométrico. Distingue si se ha realizado todo el proceso de cálculo o si bien, simplemente se han sacado datos de la base de datos. En el segundo caso debe hacer él el cálculo,

Private Sub fpsicro_Click()

```
If solomuestro = True Then  
'Este primer caso sólo lo hace si los datos han sido  
'simplemente mostrados, es decir, se hace el cálculo del  
'psicrométrico completo y se actualizan las variables.
```

```
M = Val(Text2(4).Text)  
FC2 = Val(Text2(5).Text)
```

```
P = Val(Text3(0).Text)  
TS = Val(Text3(1).Text)
```

```
If Option1(0).Value = True Then  
var3 = Val(Text3(2).Text)  
tipo = 1  
End If
```

```
If Option1(1).Value = True Then  
var3 = Val(Text3(2).Text)  
tipo = 2  
End If
```

```
If Option1(2).Value = True Then
    var3 = Val(Text3(2).Text)
    tipo = 3
End If
```

```
V = 0
H = 0
X = 0
```

```
FC1 = coef_correccion(Tmw)
If FC1 < 1 Then FC1 = 1
```

```
incrementoOk = DELTA / (FC1 * FC2)
```

```
Tsprima = resolucion_TS_incremento(incrementoOk, Te, Tu, TS)
```

```
Call cosaspsicrometrico(P, TS, var3, tipo)
```

```
    If malomalo = True Then
```

```
        'Call clrsc
        Text2(4).Text = "0"
        Text2(5).Text = "1"
        For i = 0 To 3
            Text5(i).Text = ""
        Next i
```

```
        Text1(0).Text = ""
        Text1(1).Text = "1"
        Text1(2).Text = ""
        Text1(3).Text = "1"
        Text1(4).Text = ""
```

```
        For i = 0 To 5
            Text6(i).Text = ""
        Next i
        Text7.Text = ""
        Text4(0).Text = Format(1)
        Text4(1).Text = ""
        combocol.Text = ""
        Text3(2).Text = ""
```

```
    malomalo = False
    Exit Sub
End If
```

```
Hent = H / 4.178

Hentrada = H / 4.178
Ventrada = V
Vent = Ventrada
V = 1 / Vent
cadena = Format(V, "0.000")
densaire = quitarcoma(cadena)
Trocient = Trocio
THent = TH
TSent = TS
Pent = P
HRent = HR
If tipo <> 3 Then
  Xentrada = X * 1000
  Xent = Xentrada
End If
If tipo = 3 Then
  Xentrada = X
  Xent = Xentrada
End If

tipo = 3

var3 = Xentrada

If Tsprima > 69 Then
  Tsmayor = TS
  TSS = 69
  Call cosaspsicrometrico(P, TSS, var3, tipo)
  If malomalo = True Then
    Text2(4).Text = "0"
    Text2(5).Text = "1"
    For i = 0 To 3
      Text5(i).Text = ""
    Next i

    Text1(0).Text = ""
    Text1(1).Text = "1"
    Text1(2).Text = ""
    Text1(3).Text = "1"
    Text1(4).Text = ""

    For i = 0 To 5
      Text6(i).Text = ""
```



```
Next i
Text7.Text = ""
Text4(0).Text = Format(1)
Text4(1).Text = ""
combocol.Text = ""
Text3(2).Text = ""
malomalo = False

Exit Sub
End If
Hsalidaini = H / 4.178
Vsalida = V
Xsalida = X
Hsali = Hsalidaini
Vsali = Vsalida
Xsali = Xsalida
Trociosali = Trocio
THsali = TH
Psali = P
TSSali = Tsprima
HRsali = HR
TSS = 39
Call cosaspsicrometrico(P, TSS, var3, tipo)
If malomalo = True Then
Text2(4).Text = "0"
Text2(5).Text = "1"
For i = 0 To 3
Text5(i).Text = ""
Next i

Text1(0).Text = ""
Text1(1).Text = "1"
Text1(2).Text = ""
Text1(3).Text = "1"
Text1(4).Text = ""

For i = 0 To 5
Text6(i).Text = ""
Next i
Text7.Text = ""
Text4(0).Text = Format(1)
Text4(1).Text = ""
combocol.Text = ""
Text3(2).Text = ""
malomalo = False
```

```
Exit Sub
End If

Hsalidafin = H / 4.178

incrementoiniH = Hsalidaini - Hsalidafin
incrementoH = (Tsprima - TS) * incrementoiniH / 30
Hsali = Hent + incrementoH
End If
If Tsprima < 69 Then
Call cosaspsicrometrico(P, Tsprima, var3, tipo)
If malomalo = True Then Exit Sub
Hsalida = H / 4.178
Vsalida = V
Xsalida = X

Hsali = Hsalida
Vsali = Vsalida
Xsali = Xsalida
Trociosali = Trocio
THsali = TH
Psali = P
TSSali = Tsprima
HRsali = HR
incrementoH = Hsalida - Hentrada
End If
If Option4(0).Value = True Then
V = Vsalida
End If
If Option4(1).Value = True Then
V = Ventrada
End If
End If

Dim frmX As New fpsicro
cadena = Format(Pent)
frmX.Variable(0).Text = quitarcoma(cadena)
cadena = Format(TSent)
frmX.Variable(1).Text = quitarcoma(cadena)
cadena = Format(THent, "#0.0")
frmX.Variable(2).Text = quitarcoma(cadena)
cadena = Format(HRent, "#0.0")
frmX.Variable(3).Text = quitarcoma(cadena)
cadena = Format(Xent, "#0.00")
```

```
frmX.Variable(4).Text = quitarcoma(cadena)
cadena = Format(Hent, "#0.00")
frmX.Variable(5).Text = quitarcoma(cadena)
cadena = Format(Vent, "#0.00")
frmX.Variable(6).Text = quitarcoma(cadena)
cadena = Format(Trocioent, "#.0")
frmX.Variable(7).Text = quitarcoma(cadena)
cadena = (Psali)
frmX.Text1(0).Text = quitarcoma(cadena)
cadena = Format(Tsprima, "##0.0")
frmX.Text1(1).Text = quitarcoma(cadena)
cadena = Format(THsali, "#0.0")
frmX.Text1(2).Text = quitarcoma(cadena)
cadena = Format(HRsali, "#0.0")
frmX.Text1(3).Text = quitarcoma(cadena)
cadena = Format(Xsali, "#0.00")
frmX.Text1(4).Text = quitarcoma(cadena)
cadena = Format(Hsali, "#0.00")
frmX.Text1(5).Text = quitarcoma(cadena)
cadena = Format(Vsali, "#0.00")
frmX.Text1(6).Text = quitarcoma(cadena)
cadena = Format(Trociosali, "#0.0")
frmX.Text1(7).Text = quitarcoma(cadena)
```

```
frmX.Show vbModal
```

End Sub

1.42. fdoc_Click.

Para crear el documento de todos los items pertenecientes al
proyecto elegido.

Private Sub fdoc_Click()

```
'el usuario quiere crear un documento  
'debe haber escogido cliente y proyecto  
If clickproy = True Then  
    Dim rsdoc As Recordset  
    Dim buscarr As String  
    Dim i As Integer  
  
    buscarr = " SELECT * FROM Tablaitem WHERE " _  
        & " Idproyecto = " & idproy & " "  
  
    'debemos mirar si el proyecto contiene items  
    Set rsdoc = bdclientes.OpenRecordset(buscarr)  
    'si no hay va a notenemos  
    On Error GoTo notenemos  
    rsdoc.MoveFirst  
  
    'crea los objetos de word, copia las plantillas  
    'crea los rangos y guarda las variables en  
    'el texto de los rangos  
    Dim objword, documento, docword, primerapag, segundapag  
        As Object  
    Dim nombredoc As String  
    Dim texto As String  
    Dim rango, rango2 As Range  
    Dim unidades As Integer  
  
    Set objword = CreateObject("Word.Application.8")  
  
    nombredoc = "C:\aeros\oferta1.doc"  
    Set primerapag = objword.Documents  
    primerapag.Open nombredoc  
    Set rango = objword.Documents(nombredoc).Range(Start:=0,  
End:=932)  
    rango.Copy  
    primerapag.Close  
  
    Set docword = objword.Documents  
    docword.Add ("C:\aeros\oferta.dot")  
    objword.Visible = True  
    Set rango2 = objword.ActiveDocument.Range(Start:=0, End:=0)  
    rango2.Collapse (wdCollapseStart)  
    rango2.Paste
```

```
nombredoc = "C:\aeros\offerta2.doc"
Set segundapag = objword.Documents
segundapag.Open nombredoc
Set rango = objword.Documents(nombredoc).Range
                (Start:=0, End:=2005)
rango.Copy
objword.Documents(1).Close

i = 0
Dim ncad As Integer
Dim cadena As String

cadena = nproy
ncad = Len(cadena)
Set rango = objword.ActiveDocument.Range(Start:=389, End:=389
                + ncad)
rango.Text = cadena

cadena = nempresa
ncad = Len(cadena)
Set rango = objword.ActiveDocument.Range(Start:=101, End:=101
                + ncad)
rango.Font.Bold = True
rango.Text = cadena

cadena = nperson
ncad = Len(cadena)
Set rango = objword.ActiveDocument.Range(Start:=15, End:=15 +
                ncad)
rango.Font.Bold = True
rango.Text = cadena

cadena = nombreusuario
ncad = Len(cadena)
Set rango = objword.ActiveDocument.Range(Start:=(66),
End:=((66)) + ncad)
rango.Text = cadena
Set rango = objword.ActiveDocument.Range(Start:=894, End:=894
                + ncad)
rango.Text = cadena

cadena = telusuario
ncad = Len(cadena)
Set rango = objword.ActiveDocument.Range(Start:=(217),
                End:=((217)) + ncad)
```

rango.Text = cadena

cadena = faxusuario
ncad = Len(cadena)
Set rango = objword.ActiveDocument.Range(Start:=(257),
End:=((257) + ncad))
rango.Text = cadena

'lo hace tantas veces como items haya

Do While Not rsdoc.EOF

Set rango2 = objword.ActiveDocument.Range(Start:=(932 + i *
2005), End:=(932 + (i) * 2005))

rango2.Collapse (wdCollapseStart)

rango2.Paste

Dim ittem As String

ittem = rsdoc.Fields("nitem")

cadena = Format(ittem)

ncad = Len(cadena)

Set rango = objword.ActiveDocument.Range(Start:=(932 + i *
2005 + 162), End:=((932 + i * 2005 + 162)) + ncad)

rango.Text = cadena

cadena = nproy

ncad = Len(cadena)

Set rango = objword.ActiveDocument.Range(Start:=(932 + i *
2005 + 14), End:=((932 + i * 2005 + 14)) + ncad)

rango.Text = cadena

cadena = nempresa

ncad = Len(cadena)

Set rango = objword.ActiveDocument.Range(Start:=(932 + i *
2005 + 71), End:=((932 + i * 2005 + 71)) + ncad)

rango.Text = cadena

If rsdoc.Fields("nunidades") = "xxx" Then

cadena = ""

Else:

cadena = rsdoc.Fields("nunidades")

End If

ncad = Len(cadena)

Set rango = objword.ActiveDocument.Range(Start:=(932 + i *

```
                2005 + 445), End:=((932 + i * 2005 + 445)) + ncad)
rango.Text = cadena
unidades = Val(cadena)
If unidades = 1 Then
    Set rango = objword.ActiveDocument.Range(Start:=(932 + i *
                2005 + 818), End:=((932 + i * 2005 + 818)) + 25)
    rango.Text = "
End If
cadena = ""
caden = ""
If rsdoc.Fields("Ipotencia") = "xxx" Then
    cadena = ""
Else:
    cadena = rsdoc.Fields("Ipotencia")

End If
intermedia = Val(cadena)
lpot = intermedia / unidades
cadena = Format(lpot / 1000)
Dim lugar, longitud As Integer
Dim caracter As String
longitud = Len(cadena)
For lugar = 1 To longitud
    caracter = Mid(cadena, lugar, 1)
    If caracter = "," Or caracter = "." Then
        GoTo final
    Else:
        caden = caden & caracter
    End If
Next lugar
final:
cadena = caden
ncad = Len(cadena)
Set rango = objword.ActiveDocument.Range(Start:=(932 + i *
                2005 + 530), End:=((932 + i * 2005 + 530)) + ncad)
rango.Text = cadena

If rsdoc.Fields("lcaudal") = "xxx" Then
    cadena = ""
Else:
    cadena = rsdoc.Fields("lcaudal")

End If
intermedia = Val(cadena)
lcaudal = intermedia / unidades
```

```
ncad = Len(cadena)
Set rango = objword.ActiveDocument.Range(Start:=(932 + i *
      2005 + 1126), End:=((932 + i * 2005 + 1126)) + ncad)
rango.Text = cadena

If rsdoc.Fields("ITin") = "xxx" Then
  cadena = ""
Else:
  cadena = rsdoc.Fields("ITin")
  cadena = cadena & " / "
End If
ncad = Len(cadena)
Set rango = objword.ActiveDocument.Range(Start:=(932 + i *
      2005 + 1211), End:=((932 + i * 2005 + 1211)) + ncad)
rango.Text = cadena
donde = 932 + i * 2005 + 1211 + ncad
If rsdoc.Fields("ITout") = "xxx" Then
  cadena = ""
Else:
  cadena = rsdoc.Fields("ITout")
End If
ncad = Len(cadena)
Set rango = objword.ActiveDocument.Range(Start:=donde,
      End:=donde + ncad)
rango.Text = cadena

cadena = rsdoc.Fields("Imargen")
ncad = Len(cadena)
Set rango = objword.ActiveDocument.Range(Start:=(932 + i *
      2005 + 611), End:=((932 + i * 2005 + 611)) + ncad)
rango.Text = cadena

cadena = rsdoc.Fields("nfluido")
ncad = Len(cadena)
Set rango = objword.ActiveDocument.Range(Start:=(932 + i *
      2005 + 902), End:=((932 + i * 2005 + 902)) + ncad)
rango.Font.Bold = True
rango.Text = cadena

If rsdoc.Fields("presion") = "xxx" Then
  cadena = ""
Else:
  cadena = rsdoc.Fields("presion")
End If
```



```
ncad = Len(cadena)
Set rango = objword.ActiveDocument.Range(Start:=(932 + i *
      2005 + 1169), End:=((932 + i * 2005 + 1169)) + ncad)
rango.Text = cadena
```

```
cadena = rsdoc.Fields("aTin")
cadena = cadena & " / "
ncad = Len(cadena)
Set rango = objword.ActiveDocument.Range(Start:=(932 + i *
      2005 + 1343), End:=((932 + i * 2005 + 1343)) + ncad)
rango.Text = cadena
donde = 932 + i * 2005 + 1343 + ncad
```

```
cadena = rsdoc.Fields("Tssalida")
ncad = Len(cadena)
Set rango = objword.ActiveDocument.Range(Start:=donde,
      End:=donde + ncad)
rango.Text = cadena
```

```
cadena = rsdoc.Fields("tipodiam")
ncad = Len(cadena)
Set rango = objword.ActiveDocument.Range(Start:=(932 + i *
      2005 + 1605), End:=((932 + i * 2005 + 1605)) + ncad)
rango.Text = cadena
```

```
If rsdoc.Fields("nmodulos") <> "xxx" Then
  cadena = rsdoc.Fields("nmodulos")
```

```
Else:
  cadena = ""
End If
ncad = Len(cadena)
Set rango = objword.ActiveDocument.Range(Start:=(932 + i *
      2005 + 1539), End:=((932 + i * 2005 + 1539)) + ncad)
rango.Text = cadena
ventiladores = Val(cadena)
```

```
If rsdoc.Fields("densidad") = "xxx" Then
  cadena = ""
Else:
  cadena = rsdoc.Fields("densidad")
End If
ncad = Len(cadena)
```

```
Set rango = objword.ActiveDocument.Range(Start:=(932 + i *  
2005 + 961), End:=((932 + i * 2005 + 961)) + ncad)  
rango.Text = cadena  
  
If rsdoc.Fields("Cp") = "xxx" Then  
    cadena = ""  
Else:  
    cadena = rsdoc.Fields("Cp")  
End If  
ncad = Len(cadena)  
Set rango = objword.ActiveDocument.Range(Start:=(932 + i *  
2005 + 1052), End:=((932 + i * 2005 + 1052)) + ncad)  
rango.Text = cadena  
  
If rsdoc.Fields("pcc") = "xxx" Then  
    cadena = ""  
Else:  
    cadena = rsdoc.Fields("pcc")  
End If  
ncad = Len(cadena)  
Set rango = objword.ActiveDocument.Range(Start:=(932 + i *  
2005 + 1292), End:=((932 + i * 2005 + 1292)) + ncad)  
rango.Text = cadena  
  
If rsdoc.Fields("S") = "xxx" Then  
    cadena = ""  
Else:  
    cadena = rsdoc.Fields("S")  
End If  
ncad = Len(cadena)  
Set rango = objword.ActiveDocument.Range(Start:=(932 + i *  
2005 + 1774), End:=((932 + i * 2005 + 1774)) + ncad)  
rango.Text = cadena  
  
If rsdoc.Fields("Vint") = "xxx" Then  
    cadena = ""  
Else:  
    cadena = rsdoc.Fields("Vint")  
End If  
ncad = Len(cadena)  
Set rango = objword.ActiveDocument.Range(Start:=(932 + i *  
2005 + 1836), End:=((932 + i * 2005 + 1836)) + ncad)  
rango.Text = cadena  
cadena = ""  
cad = ""
```

```
If rsdoc.Fields("potencia") = "xxx" Then
    cadena = ""
    Else:
        cadena = rsdoc.Fields("potencia")
    End If
    ncad = Len(cadena)
    Set rango = objword.ActiveDocument.Range(Start:=(932 + i *
        2005 + 1668), End:=((932 + i * 2005 + 1668)) + ncad)
    rango.Text = cadena
    Set rango = objword.ActiveDocument.Range(Start:=(932 + i *
        2005 + 1668), End:=((932 + i * 2005 + 1668)) + 3)
    rango.Font.Bold = True

If rsdoc.Fields("revoluciones") = "xxx" Then
    cadena = ""
    Else:
        cadena = rsdoc.Fields("revoluciones")
    End If
    ncad = Len(cadena)
    Set rango = objword.ActiveDocument.Range(Start:=(932 + i *
        2005 + 1732), End:=((932 + i * 2005 + 1732)) + ncad)
    rango.Text = cadena
    Set rango = objword.ActiveDocument.Range(Start:=(932 + i *
        2005 + 1732), End:=((932 + i * 2005 + 1732) + 3))
    rango.Font.Bold = True

If rsdoc.Fields("caudalmod") = "xxx" Then
    cadena = ""
    Else:
        cadena = rsdoc.Fields("caudalmod")
    End If
    intermedio = Val(cadena) * ventiladores
    cadena = Format(intermedio)
    ncad = Len(cadena)
    Set rango = objword.ActiveDocument.Range(Start:=(932 + i *
        2005 + 1248), End:=((932 + i * 2005 + 1248)) + ncad)
    rango.Text = cadena

If rsdoc.Fields("densiaire") = "xxx" Then
    cadena = ""
    Else:
        cadena = rsdoc.Fields("densiaire")
    End If
    ncad = Len(cadena)
    Set rango = objword.ActiveDocument.Range(Start:=(932 + i *
```

```
                2005 + 1001), End:=((932 + i * 2005 + 1001)) + ncad)
rango.Text = cadena

If rsdoc.Fields("HR") = "xxx" Then
    cadena = ""
Else:
    cadena = rsdoc.Fields("HR")
End If
ncad = Len(cadena)
Set rango = objword.ActiveDocument.Range(Start:=(932 + i *
                2005 + 1090), End:=((932 + i * 2005 + 1090)) + ncad)
rango.Text = cadena

cadena = ""
If rsdoc.Fields("tubaletas") = "xxx" Then
    caden = ""
Else:
    caden = rsdoc.Fields("tubaletas")
End If
longitud = Len(caden)
For lugar = 1 To longitud
    caracter = Mid(caden, lugar, 1)
    If caracter = "/" Then
        GoTo finali
    Else:
        cadena = cadena & caracter
    End If
Next lugar
finali:
cadena = cadena & " "
ncad = Len(cadena)
Set rango = objword.ActiveDocument.Range(Start:=(932 + i *
                2005 + 1516), End:=((932 + i * 2005 + 1516)) + ncad)
rango.Text = cadena

cadena = ""
longitud = Len(caden)
For lugar = 1 To longitud
    caracter = Mid(caden, lugar, 1)
    If caracter = "/" Then
        For y = (lugar + 1) To longitud
            caracter = Mid(caden, y, 1)
            cadena = cadena & caracter
        Next y
    End If
Next lugar
Exit For
```

```
    End If
Next lugar
cadena = cadena & " "
ncad = Len(cadena)
Set rango = objword.ActiveDocument.Range(Start:=(932 + i *
    2005 + 1575), End:=((932 + i * 2005 + 1575)) + ncad)
rango.Text = cadena

If rsdoc.Fields("denom") = "xxx" Then
    cadena = ""
Else:
    cadena = rsdoc.Fields("denom")
End If
If rsdoc.Fields("ncol") <> "1" Then
    Dim ncol As String
    ncol = rsdoc.Fields("ncol")
    cadena = ncol & " x " & cadena
End If
ncad = Len(cadena)
Set rango = objword.ActiveDocument.Range(Start:=(932 + i *
    2005 + 1638), End:=((932 + i * 2005 + 1638)) + ncad)
rango.Text = cadena

If rsdoc.Fields("DN") = "xxx" Then
    cadena = ""
Else:
    cadena = rsdoc.Fields("DN")
End If
ncad = Len(cadena)
Set rango = objword.ActiveDocument.Range(Start:=(932 + i *
    2005 + 1706), End:=((932 + i * 2005 + 1706)) + ncad)
rango.Text = cadena

cadena = rsdoc.Fields("paso")
ncad = Len(cadena)
Set rango = objword.ActiveDocument.Range(Start:=(932 + i *
    2005 + 1584), End:=(932 + (i * 2005 + 1584) + ncad))
rango.Text = quitarcoma(cadena)
i = i + 1
rsdoc.MoveNext
Loop
cadena = Format(i + 1)
ncad = Len(cadena)
Set rango = objword.ActiveDocument.Range(Start:=205, End:=205
```

```
                + ncad)
    rango.Text = cadena
    Exit Sub
notenemos:
    MsgBox "No hay items correspondientes al proyecto seleccionado",
        16
    Exit Sub
Else:
    MsgBox "No hay un proyecto seleccionado ", 16
End If

End Sub
```

1.43. Command1_Click.

Entramos en la parte del estudio del fluido:

Private Sub Command1_Click()

'muestra las opciones del comando fluido

```
Dim frmX As New feditorfl
```

```
Set rsfluidos = bdffluidos.OpenRecordset("tablafluidos",
    dbOpenDynaset)
```

```
rsfluidos.MoveFirst
```

```
Do While Not rsfluidos.EOF
```

```
    frmX.listafluidos.AddItem rsfluidos.Fields("fluido")
```

```
    rsfluidos.MoveNext
```

```
Loop
```

```
frmX.Tclien.Text = combocliente.Text
```

```
frmX.Tproy.Text = comboprooy.Text
```

```
frmX.Titem.Text = comboitem.Text
```

```
frmX.Show vbModal
```

```
End Sub
```

2.FEDITORFL.

2.1. *Command2_Click.*

Al apretar este botón, un nuevo formulario aparecerá. Con él podremos ver las características del fluido que hatamos escogido dentro de un rango de temperaturas y con un increneto que escogeremos.

Private Sub Command2_Click()

```
If listafluidos.ListIndex = -1 Then
    MsgBox "Debe escoger un fluido", 16
    Exit Sub
End If
Dim frmZ As New fdisplay
fluidodisplay = listafluidos.Text
frmZ.Caption = fluidodisplay
frmZ.Tfluido.Text = fluidodisplay
Unload Me
frmZ.Show vbModal
Me.Move (Screen.Width - Me.Width) / 2, (Screen.Height -
    Me.Height) / 2
End Sub
```

2.2. *Command3_Click.*

Para crear un nuevo fluido. Aparecerá otr ventana, donde introduciremos las características del fluido y sus correspondientes temperaturas.

Private Sub Command3_Click()

```
Dim frmX As New fnuevofluido  
frmX.Tclien.Text = Tclien.Text  
frmX.Tproy.Text = Tproy.Text  
frmX.Titem.Text = Titem.Text
```

```
Unload Me  
frmX.Show vbModal
```

```
End Sub
```

2.3. Command4_Click.

Vamos a borrar un fluido. Primero le localizaremos dentro de la tabla y después lo eliminaremos.

```
Private Sub Command4_Click()
```

```
If listafluidos.ListIndex = -1 Then  
    MsgBox "Debe escoger un fluido", 16  
    Exit Sub  
End If
```

```
Dim bdfluidos As Database  
Dim rsfluidos As Recordset  
Dim frmPrin As New principalform
```

```
Unload frmPrin
```

```
nombrefluido = listafluidos.Text  
Set bdfluidos = OpenDatabase("C:\aeros\fluidos.mdb")  
Set rsfluidos = bdfluidos.OpenRecordset("tablafluidos",  
dbOpenDynaset)  
rsfluidos.MoveFirst
```

```
Do While Not rsfluidos.EOF  
    If nombrefluido = rsfluidos.Fields("fluido") Then  
        rsfluidos.Delete  
        'rsfluidos.Close
```



```
Exit Do
Else: rsfluidos.MoveNext
End If
Loop

rsfluidos.MoveFirst
principalform.combofluido.Clear
Do While Not rsfluidos.EOF
    principalform.combofluido.AddItem rsfluidos.Fields("fluido")
    rsfluidos.MoveNext

Loop

rsfluidos.Close
bdffluidos.Close
Unload Me

End Sub
```

3.FDISPLAY.

3.1.Ttemp_Lostfocus.

Comprobaremos que la temperatura final es mayor a la inicial.

```
Private Sub Ttemp_LostFocus(Index As Integer)
Select Case Index
Case 0:
    If Ttemp(1).Text <> "" Then
        If Val(Ttemp(0).Text) > Val(Ttemp(1).Text) Then
            MsgBox "La temperatura final debe ser mayor a la primera",
                16
            Ttemp(1).Text = ""
        End If
    End If
Case 1:
    If Ttemp(0).Text <> "" Then
        If Val(Ttemp(0).Text) > Val(Ttemp(1).Text) Then
            MsgBox "La temperatura final debe ser mayor a la primera",
                16
            Ttemp(1).Text = ""
        End If
    End If
End Select
End Sub
```

3.2.Command1_Click.

En esta rutina, buscamos inicialmente el fluido en la base de datos. Guardamos los coeficientes en unas variables y hacemos un bucle desde la temperatura inicial a la final, donde vamos calculando el valor de la densidad, del calor específico, de la viscosidad y conductividad para cada temperatura y la añadimos a una cadena, la cual la introduciremos en la pantalla en un control que es una lista.

Private Sub Command1_Click()

```
If Ttemp(0).Text = "" Or Ttemp(1).Text = "" Or Tinc.Text = "" Then
    MsgBox "Debe introducir todas las variables", 16
    Exit Sub
End If
```

```
Dim frmP As New ftextover
Dim nom As String
```

```
Tini = Ttemp(0).Text
Tfini = Ttemp(1).Text
increment = Tinc.Text
N_fluido = Tfluido.Text
```

```
Unload Me
Dim bdfluidos As Database
Dim rsfluidos As Recordset
Dim buscar, nombrefluido As String
```

```
Set bdfluidos = OpenDatabase("c:\aeros\fluidos.mdb")
Set rsfluidos = bdfluidos.OpenRecordset("tablafluidos",
dbOpenDynaset)
```

```
nombrefluido = N_fluido
buscar = " fluido = " & nombrefluido & ""
```

```
rsfluidos.MoveLast
rsfluidos.FindFirst buscar
frmP.Caption = "Características de " & nombrefluido
```

```
Dim i As Integer
Dim Tinicial, Tfinal, incremento As Currency
Dim tprima As String
```

```
frmP.List1.Clear
frmP.List1.AddItem " T densidad c.especifico
conductividad viscosidad"
frmP.List1.AddItem " °C Kg/m^3 Kcal/Kg°C
Kcal/m.h.°C cP"
```

```
salto: i = 0
```

```
Tinicial = Val(Tini)
Tfinal = Val(Tfini)
incremento = Val(increm)
a0d = rsfluidos.Fields("a0d")
a1d = rsfluidos.Fields("a1d")
a2d = rsfluidos.Fields("a2d")
a0cp = rsfluidos.Fields("a0cp")
a1cp = rsfluidos.Fields("a1cp")
a2cp = rsfluidos.Fields("a2cp")
a0c = rsfluidos.Fields("a0c")
a1c = rsfluidos.Fields("a1c")
a2c = rsfluidos.Fields("a2c")
a0v = rsfluidos.Fields("a0v")
a1v = rsfluidos.Fields("a1v")
a2v = rsfluidos.Fields("a2v")

Do While Tinicial <= Tfinal
  a = a0d
  b = a1d
  c = a2d
  tipo = 1
  parte1 = resultado(a, b, c, Tinicial, tipo)
  a = a0cp
  b = a1cp
  c = a2cp
  tipo = 2
  parte2 = resultado(a, b, c, Tinicial, tipo)
  If a0c = 0 And a1c = 0 And a2c = 0 Then
    parte3 = ""
  Else:
    a = a0c
    b = a1c
    c = a2c
    tipo = 3
    parte3 = resultado(a, b, c, Tinicial, tipo)
  End If
  If a0v = 0 And a1v = 0 And a2v = 0 Then
    parte4 = ""
  Else:
    a = a0v
    b = a1v
    c = a2v
    tipo = 4
    parte4 = resultado(a, b, c, Tinicial, tipo)
```

```
End If
If Tinicial < 10 Then
    cadena = Format(Tinicial, "0.0")
    cadena = " " & cadena
End If
If Tinicial >= 10 And Tinicial < 100 Then
    cadena = Format(Tinicial, "00.0")
    cadena = " " & cadena
End If

If Tinicial = 100 Or Tinicial > 100 Then
    cadena = Format(Tinicial, "##.0")
End If

tprima = quitarcoma(cadena)
dato = " " & tprima & " " & parte1 & " " & parte2 & " "
        & parte3 & " " & parte4 & " "
frmP.List1.AddItem dato
frmP.List1.ItemData(i) = i
Tinicial = Tinicial + incremento
i = i + 1
Loop

frmP.Show vbModal
Me.Move (Screen.Width - Me.Width) / 2, (Screen.Height -
Me.Height) / 2
End Sub

Private Sub Command2_Click()
Unload Me

End Sub
```

4.FUSUARIO.

4.1. *Command1_Click*

Es la rutina que se encarga de almacenar los datos del nuevo usuario introducido.

Private Sub Command1_Click()

Dim nusuario, buscando As String

If Text1.Text = "" Then

 MsgBox "Debe introducir el nombre del nuevo usuario", 16

 Exit Sub

End If

If Text2.Text = "" Then

 MsgBox "Debe introducir el teléfono del nuevo usuario", 16

 Exit Sub

End If

If Text3.Text = "" Then

 MsgBox "Debe introducir el fax del nuevo usuario", 16

 Exit Sub

End If

Dim rsusuario As Recordset

Dim bdclientes As Database

Set bdclientes = OpenDatabase("C:\aeros\Bcliente")

If principalform.Combo4.Text = "Nuevo" Then

 Set rsusuario = bdclientes.OpenRecordset("tusuario")

 rsusuario.MoveLast

 rsusuario.AddNew

 rsusuario.Fields("nombreusuario") = Text1.Text

 nusuario = Text1.Text

 principalform.nombreusuario = Text1.Text

 rsusuario.Fields("tlfusuario") = Text2.Text

 principalform.telusuario = Text2.Text

 rsusuario.Fields("faxusuario") = Text3.Text

 principalform.faxusuario = Text3.Text

 rsusuario.Update

 rsusuario.Close

```
principalform.Combo4.Text = nusuario  
principalform.Combo4.AddItem Text1.Text  
Unload Me  
Set bdclientes = Nothing  
Set rsusuario = Nothing  
  
End If  
  
End Sub
```

5.FNUEVOFLUIDO.

5.1.Command1_Click

Se encarga de hacer las aproximaciones (constante, lineal y parabólica) de cada una de las características con la temperatura y almacena los coeficientes y el nombre del nuevo fluido en la base de datos.

```
Private Sub Command1_Click()
If Tnomfluido.Text = "" Then
    MsgBox "Debe escribir un nombre para el fluido", 16
    Exit Sub
End If
If Ttemp(0).Text = "" Then
    MsgBox "Debe escribir al menos un valor de temperatura", 16
    Exit Sub
End If
If Tdensidad(0).Text = "" Then
    MsgBox "Debe escribir al menos un valor de densidad", 16
    Exit Sub
End If
If Tcp(0).Text = "" Then
    MsgBox "Debe escribir al menos un valor de calor específico", 16
    Exit Sub
End If

Dim nombrefluido As String
Dim bdfluidos As Database
Dim rsfluidos2 As Recordset
Dim encontrado As Boolean
Dim a, b, c As Double
Dim texto1, texto2, texto3 As String
Dim cuantos As Integer
Dim a0d, a1d, a2d, a0cp, a1cp, a2cp, a0c, a1c, a2c, a0v, a1v, a2v As
Double
```

```
nombrefluido = Tnomfluido.Text
Set bdffluidos = OpenDatabase("C:\aeros\fluidos.mdb")
Set rsfluidos2 = bdffluidos.OpenRecordset("tablafluidos")
rsfluidos2.MoveFirst
```

```
Do While Not rsfluidos2.EOF
  If nombrefluido = rsfluidos2.Fields("fluido") Then
    MsgBox "El fluido ya existe. Debe cambiar el nombre", 16
    Tnomfluido.Text = ""
    Exit Sub
  Else: rsfluidos2.MoveNext
End If
Loop
```

```
texto1 = Ttemp(0).Text
texto2 = Ttemp(1).Text
texto3 = Ttemp(2).Text
```

```
cuantos = cuantos_datos(texto1, texto2, texto3)
Select Case cuantos
```

```
Case 1:
  a0d = Val(Tdensidad(0).Text)
  a1d = 0
  a2d = 0
  a0cp = Val(Tcp(0).Text)
  a1cp = 0
  a2cp = 0
  a0c = Val(Tcond(0).Text)
  a1c = 0
  a2c = 0
  a0v = Val(Tvisc(0).Text)
  a1v = 0
  a2v = 0
```

```
Case 2:
  a = Val(Tdensidad(0).Text)
  b = Val(Tdensidad(1).Text)
  Call primer_orden(a, b)
  a0d = sol1
  a1d = sol2
  a2d = 0
```

```
  a = Val(Tcp(0).Text)
  b = Val(Tcp(1).Text)
```

```
Call primer_orden(a, b)
a0cp = sol1
a1cp = sol2
a2cp = 0
```

```
a = Val(Tcond(0).Text)
b = Val(Tcond(1).Text)
Call primer_orden(a, b)
a0c = sol1
a1c = sol2
a2c = 0
```

```
a = Val(Tvisc(0).Text)
b = Val(Tvisc(1).Text)
Call primer_orden(a, b)
a0v = sol1
a1v = sol2
a2v = 0
```

Case 3:

```
a = Val(Tdensidad(0).Text)
b = Val(Tdensidad(1).Text)
c = Val(Tdensidad(2).Text)
Call segundo_orden(a, b, c)
a0d = sol1
a1d = sol2
a2d = sol3
```

```
a = Val(Tcp(0).Text)
b = Val(Tcp(1).Text)
c = Val(Tcp(2).Text)
Call segundo_orden(a, b, c)
a0cp = sol1
a1cp = sol2
a2cp = sol3
```

```
a = Val(Tcond(0).Text)
b = Val(Tcond(1).Text)
c = Val(Tcond(2).Text)
Call segundo_orden(a, b, c)
a0c = sol1
a1c = sol2
a2c = sol3
```

```
a = Val(Tvisc(0).Text)
b = Val(Tvisc(1).Text)
c = Val(Tvisc(2).Text)
Call segundo_orden(a, b, c)
a0v = sol1
a1v = sol2
a2v = sol3
Case 0:
Exit Sub
```

End Select

```
If Tnomfluido.Text = "" Then
MsgBox "Debe introducir el nombre del fluido", 16
GoTo fin
End If
```

```
rsfluidos2.MoveLast
rsfluidos2.AddNew
rsfluidos2.Fields("a0d") = a0d
rsfluidos2.Fields("a1d") = a1d
rsfluidos2.Fields("a2d") = a2d
rsfluidos2.Fields("a0cp") = a0cp
rsfluidos2.Fields("a1cp") = a1cp
rsfluidos2.Fields("a2cp") = a2cp
rsfluidos2.Fields("a0c") = a0c
rsfluidos2.Fields("a1c") = a1c
rsfluidos2.Fields("a2c") = a2c
rsfluidos2.Fields("a0v") = a0v
rsfluidos2.Fields("a1v") = a1v
rsfluidos2.Fields("a2v") = a2v
rsfluidos2.Fields("fluido") = Tnomfluido.Text
rsfluidos2.Update
rsfluidos2.Close
```

```
principalform.combocliente.Text = Tlien.Text
principalform.comboproy.Text = Tproy.Text
principalform.comboitem.Text = Titem.Text
principalform.Tpaso.Text = nombrefluido
principalform.combofluido.AddItem nombrefluido
```

```
Unload Me
fin:
```

End Sub

ESTUDIO ECONÓMICO

Debido a que el proceso de cálculo era anteriormente manual, conllevaba una pérdida de tiempo bastante importante, y eran horas de pérdidas de ingenieros.

Con el programa, un cálculo que antes podía llevar varios minutos, es ahora cuestión de segundos.

Con lo cual, concluimos con que el desarrollo del proyecto es rentable.

CÁLCULOS

**Este proyecto, debido a su carácter informático,
no tiene cálculos**

PLANOS

PLANOS

**Este proyecto, debido a su carácter
informático, no contiene planos.**

PLIEGO DE CONDICIONES

ÍNDICE DE PLIEGO DE CONDICIONES.

1. CONDICIONES GENERALES	----- 2
2. CONDICIONES ECONÓMICAS	----- 4
3. CONDICIONES TÉCNICAS Y PARTICULARES	----- 6

1. CONDICIONES GENERALES

Las condiciones y cláusulas que se establecen en el documento, tratan sobre la contratación, por parte de una persona física o jurídica, del Programa de Cálculo de aeroenfriadores desarrollado en este proyecto.

El cumplimiento de estas condiciones obliga a ambas partes y son las que a continuación se exponen:

1. Las dos partes se comprometen desde la fecha de la firma del contrato a cumplir todo lo que se estipula en él.
2. En el caso de reclamación o discrepancia en lo concerniente al cumplimiento de lo pactado por cualquiera de las partes, una vez agotada cualquier vía de entendimiento, se tramitará el asunto por la vía de lo legal.
3. El vendedor queda obligado a facilitar a la otra parte cualquier información que contribuya a mejorar la instalación y funcionamiento del sistema, siempre que le requiera para ello.

4. El comprador a su vez, queda obligado a explicar al fabricante todas las características del proceso a calcular, con el objeto de facilitar la instalación, quedando el proveedor libre de responsabilidad sobre cualquier defecto que surja por el incumplimiento de dicha obligación.

5. El plazo de entrega será de tres semanas a partir de la firma del contrato.

6. Si la entrega se retrasase más de las tres semanas acordadas, el comprador podrá rescindir el contrato, siéndose retribuidas las cantidades abonadas.

7. Queda fijado el plazo de un año de garantía a partir de la fecha de entrega del sistema. La garantía queda anulada al expirar el plazo o si se demuestra que el sistema ha sido objeto de manipulación indebida.

2. CONDICIONES ECONÓMICAS

1. El valor de la venta quedará fijado de común acuerdo por ambas partes

2. Los plazos para los pagos serán los siguientes:
 - 20% a la firma del contrato

 - 60% en el momento de la entrega

 - 20% a los 45 días

3. Cualquier demora en el pago de lo estipulado sufrirá un recargo del 5% sobre la cantidad retenida.

4. Los gastos de envío y embalaje correrán a cuenta del vendedor.

5. El vendedor acepta la responsabilidad sobre cualquier defecto o averías causadas durante el transporte.

6. Hasta la expiración del plazo de garantía, la totalidad de los gastos ocasionados por reparaciones, correrán a cargo del vendedor.

3. CONDICIONES TÉCNICAS Y PARTICULARES.

1. Los equipos informáticos han de estar homologados a la reglamentación técnica vigente en España en el día de la fecha del presente proyecto.
2. El teclado del equipo será teclado español.
3. El lugar de instalación del equipo debe ajustarse a las condiciones de temperatura y humedad indicados.
4. Los soportes de los equipos deben ser firmes y sus emplazamientos fijos.
5. La alimentación eléctrica debe ser de 220V 50Hz monofásica. Deberá ajustarse a todo lo estipulado en el reglamento de baja tensión. La estabilidad del sistema es un factor estremadamente importante; en el caso de verse comprometida su fiabilidad, deberá añadirse un sistema de alimentación ininterrumpida de las características adecuadas.
6. Sistema operativo Windows 95 o Windows NT.

7. La unidad central deberá tener 16 Mb mínimo de memoria RAM dinámica.

8. La versión de Office debe ser la 8.0 o versión 97.

PRESUPUESTO

PRESUPUESTO

PRESUPUESTO , INDICE

1.MEDICIONES	2
1.1 HORAS DE ESTUDIO DEL LENGUAJE VISUAL BASIC	2
1.2 HORAS DE ESTUDIO DE ACCESS 97	2
1.3 HORAS DE ESTUDIO DE CONEXIONES ENTRE VISUAL BASIC A BASES DE DATOS	3
1.4 HORAS DE ESTUDIO DE CONEXIONES DE VISUAL BASIC Y WORD	3
1.5 HORAS DE REALIZACION DE MODULOS Y PRUEBAS	3
1.6 HORAS EMPLEADAS PARA LA ELABORACION DE LOS DOCUMENTOS DEL PROYECTO	3
2.PRECIOS UNITARIOS	4
3.PRESUPUESTO PARCIAL	4
3.1 COSTES DE INGENIERIA	4
3.2 COSTES DE MATERIAL	5
3.3 COSTES DE LOS INTERESES DEL CURSO	5
3.4 COSTES DEL DESARROLLO DEL PROYECTO	6
3.5COSTE COMERCIAL	6
4.PRESUPUESTO GENERAL	7

1.MEDICIONES

En este epígrafe intento analizar el coste en términos temporales que lleva incorporado el estudio que he realizado para la elaboración del proyecto, tanto en aprender nuevos lenguajes de programación como en la profundización de sistemas que ya conocía como el Windows, para poder centrarme en necesidades específicas de mi proyecto.

Para poderlo analizar mas detalladamente he desglosado el estudio de las mediciones en diferentes partidas que analizaré a continuación.

1.1 horas de estudio del lenguaje de programación Visual Basic.

El lenguaje de programación visual Basic utilizado es la versión 5.0, para la obtención de un buen conocimiento de este sistema estimo que se necesitan aproximadamente unas 200 horas.

Horas de estudio: 200 horas de estudio.

Estas horas corresponden a un analista senior.

1.2 Horas de estudio de Access 97

Esta base de datos se centra en la dinámica de Windows, es la más importante en estos momentos y la más utilizada por las empresas para gestionar sus datos. Al no tener ningún conocimiento anterior un estudio que permita poseer la suficiente capacidad para un perfecto manejo de esta base de datos lo he estimado en 30 horas.

Horas de estudio: 30 horas

Estas horas corresponden a un analista senior.

1.3. Horas de estudio de conexión de Visual Basic a bases de datos

Esta vinculación entre Visual Basic a bases de datos se trata de una operativa muy reciente y muy utilizada en la actualidad, como se trata de una utilización muy especializada para programadores, existe poca bibliografía para el estudio de esta conexión estimando las horas necesarias para poder conocer el funcionamiento de estas conexiones en 50 horas.

Horas de estudio: 50 horas

Estas horas corresponden a un analista senior.

1.4 Horas de conexión entre Visual Basic y Word.

Hasta el momento de realizar el proyecto, no conocía nada de comunicaciones entre Visual Basic y word, con lo cual he tenido que dedicarle un tiempo aproximado de 30 horas.

Horas de estudio: 30 horas

Correspondiente a analista senior

1.5 Horas necesarias para la realización de módulos y pruebas de las distintas partes.

Estimando que el desarrollo de 1000 líneas, en lenguaje de alto nivel, requieren un total de 180 horas de trabajo del ingeniero de programación, (0.18 horas /línea). El programa definitivo sin tener en cuenta las pruebas realizadas, se puede considerar en torno a unas 3000 líneas.

Horas de programación: 540 horas

Correspondiente a analista senior.

1.6 Horas empleadas para la elaboración de los documentos del proyecto.

Se redactan aproximadamente dos paginas de documentación por hora, con esta cifra y el numero de hojas (aproximadamente 100), sin contar el listado de los códigos.

Horas de documentación: 30 horas

Corresponden a oficial administrativo.

2.PRECIOS UNITARIOS

En la elaboración de este presupuesto se han considerado los siguientes precios unitarios:

Horas de analista senior: 7000 pts / hora

Horas de oficial administrativo: 2000 pts / hora

3.PRESUPUESTO PARCIAL .

3.1 Costes de Ingeniería.

Horas de estudio:

• Lenguaje de programación:	200	horas
• Access 97:	30	horas
• Visual Basic y Access:	50	horas
• Visual Basic y Word:	30	horas

TOTAL : 310 horas

X 7000 pts / hora 2.170.000 pts

Horas de desarrollo del programa:

• **Módulos del sistema:** 540 horas

X 7000 pts / hora 3.780.000 pts.

Horas de documentación: 30 horas

X 2000 pts / hora 60.000 pts.

Coste total de Ingeniería: 6.010.000 pts

3.2 Costes de material.

El coste de ordenadores, programas, compilador, etc..., que han sido empleados para el desarrollo de este proyecto fue:

DISPOSITIVO	PRECIO
Pentium 120	150.000
Monitor gráfico S-VGA	45000
Ratón	3500
Microsoft Visual Basic 5	100.000
Microsoft Office 97	30.000
Microsoft Windows 95	20.000
TOTAL	348.500 pts.

Estimaremos el tiempo de utilización de materia de un 75% del tiempo de realización del proyecto, lo que hace un total de 725 horas.

Suponiendo que los materiales se amortizan al cabo de 7.000 horas de utilización, la amortización del material asciende a

Hora de uso del material:	725 horas
Coste del material:	100 pts/hora
Amortización del material:	72.500 pts.

3.3 Coste de los intereses en curso.

Se intenta valorar el coste de lo invertido en el curso del proyecto.
Si suponemos el coste del dinero de un 18% anual, teniendo en cuenta
que la duración es de nueve meses será:

Intereses del dinero(18% * 9/12)=13.5%

Costes de Ingeniería:	6.010.000 pts
Coste total de los intereses	814.355 pts

3.4 Coste del desarrollo del proyecto.

El coste del proyecto es :

Coste de Ingeniería:	6.010.000 pts
Coste de material:	348.500 pts
Coste total de intereses	814.355 pts

Coste del desarrollo del proyecto: 7.172.855 pts

3.5 Coste comercial.

Se consideran los siguientes costes para poder comercializar el producto. Entre paréntesis se indican los porcentajes sobre el precio total del proyecto.

Gastos de administración(3%):	215.486 pts
Gastos de ventas(6%):	430.371 pts
Intereses variables(6%):	430.371 pts

Coste total comercial: 1.076.228 pts

4. PRESUPUESTO GENERAL.

El resultado de los estudios anteriores es el siguiente:

- **Coste de Ingeniería:** 6.010.000 pts
 - **Coste de material:** 348.500 pts
 - **Coste total de intereses** 546.075 pts
-

Coste del desarrollo del proyecto: 7.176.228 pts

- **Gastos de administración(3%):** 215.480 pts
 - **Gastos de ventas(6%):** 430.371 pts
 - **Intereses variables(6%):** 430.371 pts
-

Coste total comercial: 1.076.222pts

COSTES TOTALES: 8.249.077pts

