



18  
24

# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

ESCUELA NACIONAL DE ESTUDIOS  
PROFESIONALES

**ACATLÁN**

***LA HERRAMIENTA HOT PARA EL DESARROLLO  
DE SISTEMAS INFORMÁTICOS***

TESIS PARA OBTENER EL TÍTULO DE  
**LICENCIADO EN MATEMÁTICAS  
APLICADAS Y COMPUTACIÓN.**

**RANFERI MOLINA BENITEZ**

ASESOR DE TESIS:  
LIC. SARA CAMACHO CANCINO



ACATLÁN

1995

FALLA DE ORIGEN



## **UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso**

### **DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

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

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

**Mamá:**

*Gracias por tu amor y gran cariño. El apoyo incondicional que siempre me brindas es un respaldo muy importante en los momentos difíciles. Te quiero mucho.*

**Papá:**

*Gracias por tu comprensión y la sólida formación de valores morales que en mí inculcaste. Te quiero.*

**Baldo:**

*La admiración y el respeto que te tengo me ha servido de ejemplo y fue un soporte que me permitió terminar la carrera. Eres un buen HERMANO, te quiero.*

**A mis hermanos:**

*Mando, estimo en gran medida tu AMISTAD y admiro tu valor ante la vida; Toño, cuentas con mi apoyo y cariño. Ericka, te quiero.*

*Agradezco a mi novia VERÓNICA el haber estado conmigo en todo momento, ayudando y alentándome a seguir adelante. Gracias por tu confianza, Te amo.*

**Ing. Rafael Olivera Puente:**

*Le agradezco todo el tiempo que me ha brindado. Es una magnífica persona, dispuesta en todo momento a brindar su ayuda; no exagero al decir que sin su intervención no hubiese sido posible este trabajo. Muchas Gracias.*

*Quiero mencionar a la Lic. Sara Camacho Cancino por su valiosa y oportuna ayuda.*

## ÍNDICE

PÁG.

### INTRODUCCIÓN.

#### **CAPÍTULO 1. CONCEPTOS GENERALES.**

<i>1.1 Realidad, Modelo y Lenguaje.....</i>	<i>2</i>
<i>1.2 El Modelo de Mayer.....</i>	<i>3</i>
<i>1.3 Sistema.....</i>	<i>6</i>
<i>1.4 Información y Datos.....</i>	<i>9</i>
<i>1.5 Sistemas Informáticos.....</i>	<i>11</i>
<i>1.6 Generador de Aplicaciones.....</i>	<i>12</i>

#### **CAPÍTULO 2. DESARROLLO DE SISTEMAS INFORMÁTICOS.**

<i>2.1 Fases en el Desarrollo de Sistemas Informáticos.....</i>	<i>15</i>
<i>2.2 Los Distintos Tipos de Herramientas Para el Desarrollo de Sistemas Informáticos.....</i>	<i>24</i>
<i>    2.2.1 Paradigma Orientado a Procesos.....</i>	<i>25</i>
<i>    2.2.2 Paradigma Orientado a Datos.....</i>	<i>27</i>
<i>    2.2.3 Paradigma Orientado a Objetos.....</i>	<i>28</i>

---

**CAPÍTULO 3. LA HERRAMIENTA HOT.**

<i>3.1</i> ¿Qué es HOT?.....	<i>31</i>
<i>3.2</i> Objetos HOT.....	<i>35</i>
<i>3.3</i> Módulo de Desarrollo.....	<i>40</i>
<i>3.4</i> Módulo de Ejecución.....	<i>60</i>
<i>3.5</i> Ligas.....	<i>64</i>

**CAPÍTULO 4. APLICACIONES.**

<i>4.1</i> Sistema de Ventas.....	<i>69</i>
<i>4.2</i> Sistema de Control de Equipo de Cómputo.....	<i>87</i>

**CONCLUSIONES.**

**BIBLIOGRAFÍA.**

---

## **INTRODUCCIÓN**

*El principal motivo detrás de la evolución del software es la necesidad que tienen las empresas de agilizar sus operaciones, debido a la creciente presión competitiva a la que están sometidas, lo cual se traduce en la necesidad de que su personal sea más productivo, que se reduzcan los costos y gastos de operación, al mismo tiempo que se generen productos y servicios más rápido y de mayor calidad.*

*Una organización con éxito es aquella que está capacitada para administrar y utilizar la información con el fin de tomar decisiones oportunas y eficaces.*

*Para darle un tratamiento adecuado al gran caudal de información existente se emplea la computadora, que se ha convertido en instrumento imprescindible, logrando como resultado optimizar los recursos con los que se cuentan.*

*La información que sirve para la toma de decisiones es uno de los recursos más valiosos con que cuentan las organizaciones y para proporcionarla de manera rápida, eficiente y confiable se elaboran los sistemas informáticos. Su elaboración requiere una metodología para su correcto desarrollo; si no se cuenta con un método para el desarrollo de sistemas de información, los desarrolladores enfrentan diversos problemas como son: la estructuración de las actividades no está bien definida, el seguimiento de las acciones a realizar no son claramente establecidas, y en general no existe un orden para llevar a buen fin el sistema de información.*

---

El método propuesto por el presente trabajo para desarrollar sistemas son las fases del Ciclo de Vida de Sistemas ya que considero que el enfoque que proporciona es adecuado y completo. En este método se pueden agrupar de manera arbitraria las diferentes fases, dependiendo de la magnitud y alcance que pretenda el sistema, manteniendo la esencia de cada una de ellas; es aplicable a cualquier sistema, sobre todo para aquellos que apoyarán rutinas de operación bien entendidas.

El personal encargado del proyecto pueden concentrarse en la terminación de cada una de las fases, estableciendo y vigilando fechas límites. Con el método del Ciclo de Vida de Sistemas la asignación a cada integrante del grupo de Desarrollo se realiza mas fácil, ya que las actividades se encuentran en etapas; además de que muestra la serie de acontecimientos que se presentan en cualquier Desarrollo y la secuencia en la que ocurren normalmente.

Existen muchas herramientas para el desarrollo de sistemas informáticos que poseen características especiales (costo, capacidad, facilidad de uso, tipos de ayuda, cantidad de recursos que consume, etc.) que determinan su utilización por cualquier empresa. En este trabajo se realiza un estudio de la herramienta HOT, describiendo sus características, explicando el enfoque proporcionado por esta herramienta, la forma de desarrollar una aplicación; elaborando dos de ellas de manera completa para mejor comprensión. Se propone como una buena alternativa para el desarrollo de sistemas de información porque sus características la pueden convertir en una herramienta muy productiva que cumple con los requisitos para desarrollar estos sistemas de manera satisfactoria.

HOT es una herramienta para el Desarrollo de Sistemas Informáticos, elaborada en México por un grupo de personas encabezadas por el Lic. Ariel Tejera, en 1989; con esta herramienta se han desarrollado actualmente más de cinco mil instalaciones,

---

entre las aplicaciones se encuentran los sistemas administrativos de: Facturación, Control de Autos, Contabilidad, Nóminas, Sistemas de Control Escolar, Inventarios, Embotelladoras, etc. Está hecha en C++ y todo el desarrollo se efectúa mediante programación visual (se observa lo que se hace, tal y como aparecerá en el sistema; sin necesidad de ligar se ve lo que se está realizando). No se requiere saber programar en forma tradicional para poder utilizar y explotar la herramienta.

HOT es un software reciente y está considerado en la nueva generación de herramientas llamadas "generadores de aplicaciones". En el Desarrollo de estas aplicaciones adquiere especial importancia la adecuada elaboración del modelo Entidad/Relación (E/R).

En esta Tesis se proporciona al lector la idea general de una nueva tendencia para producir sistemas de información.

La oportunidad de participar en desarrollos completos con HOT así como asistir a cursos sobre la herramienta me permitieron observar que es eficiente y que posee características que pueden convertirse en factor determinante a la hora de decidir el adquirir una herramienta para el Desarrollo de Sistemas, tales como el tiempo de desarrollo en aplicaciones, el costo, que es un paquete nacional y de que utiliza pocos recursos de la computadora fue lo que me motivó a elegir como tema de Tesis "La Herramienta HOT para el desarrollo de Sistemas Informáticos".

Esta Tesis va dirigida a las personas interesadas en conocer una alternativa profesional para el Desarrollo de Sistemas Informáticos. El alcance que pueda tener este trabajo depende en gran medida del enfoque que se le desee dar, desde conocer las características principales de HOT hasta elaborar una aplicación completa con esta herramienta.

Es de hacer notar que en este trabajo se utilizan indistintamente los términos "Sistemas de Información" y "Sistemas Informáticos"; de igual manera, cuando empleo el término de

---

"Aplicación con HOT" me refiero a la elaboración de un Sistema Informático desarrollado con la herramienta HOT; así, en vez de hablar de un Sistema Informático de Ventas, podré hablar de la Aplicación Sistema de Ventas.

Es conveniente para la comprensión del presente trabajo, tener nociones del Diseño de Base de Datos, así como entender qué es un Diagrama Entidad Relación (E/R). Esto es importante porque, teóricamente, antes de desarrollar cualquier Aplicación se debe efectuar el Diseño adecuado, sustentado por el Diagrama E/R correcto que represente gráficamente todas las tablas o archivos que intervendrán en el sistema.

El presente trabajo está organizado de la siguiente forma:

El Capítulo I llamado Conceptos Generales, indica los términos más comunes en la tesis y se exponen algunos conceptos que son necesarios para comprender el trabajo. Hace mención también al Modelo de Mayer resaltando su importancia por la explicación que ofrece del cómo surgen las soluciones a los problemas.

El Capítulo II denominado el Desarrollo de Sistemas Informáticos, indica las etapas principales que debe poseer un sistema al ser desarrollado; asumo el enfoque del Ciclo de Vida de Sistemas. Es importante lograr su comprensión pero de ninguna manera se debe concebir como un conjunto de reglas inflexibles ya que en la realidad algunas de estas fases se mezclan y se entrelazan. En el Desarrollo de un sistema se tienen que efectuar revisiones constantes a las etapas que ya han sido cubiertas, esto servirá para guiarnos y mantenernos sobre los objetivos para los que fue concebido el sistema.

El Capítulo III es llamado La Herramienta HOT. Incluye los aspectos sobresalientes de este software así como la exposición general que hará posible comprender una aplicación completa con HOT.

El Capítulo IV son dos aplicaciones que ilustran de manera detallada cómo se desarrolla un Sistema Informático con HOT;

---

*incluyen los Diagramas E/R indicados. Utilizo para la explicación de los sistemas las pantallas que presenta HOT (tanto de desarrollo como de ejecución).*

*HOT proporciona la alternativa de la combinación de colores en pantalla, usados en las diferentes partes que integran la herramienta, como son: menús, mensajes que envía, datos de la aplicación, etc. Esto se vuelve valioso por la claridad y la fácil identificación de cada elemento; teniendo el desarrollador pleno control sobre ello, pudiendo combinarlos según su agrado y conveniencia. Esta Tesis está impresa en blanco y negro, por lo cual dichos colores no se pueden observar.*

*Espero que la Tesis provoque la inquietud de efectuar aplicaciones con esta herramienta y probar en la práctica su verdadero valor.*

---

# CAPÍTULO 1

## Conceptos Generales

Se inicia la investigación explicando el significado de realidad, modelo y lenguaje; posteriormente se define lo que significa sistema; estos conceptos son importantes para la comprensión del presente trabajo. Puesto que uno de los objetivos principales es elaboración y creación de sistemas informáticos utilizando la herramienta HOT, es fundamental tener una idea clara y precisa de lo que se entenderá como sistema, información, datos, sistemas informáticos, etc. todo ello con la finalidad de comprender como se está modelando un problema y como se le dará solución.

Al hablar de resolver un problema se debe partir del hecho de que éste se encuentra ubicado en un determinado "universo" de elementos y de acontecimientos, al que se le llama realidad. Posteriormente se analiza cuáles son todas las alternativas para su posible solución; pero para ello es necesario construir un modelo que explique en forma cualitativa y cuantitativamente el problema que se está estudiando y la realidad en que se halla. El modelo así creado implica una cierta manera de manipularlo, a través de un cierto conjunto de símbolos, al cual se le denomina lenguaje.

## **1.1 Realidad, modelo y lenguaje**

### **Realidad**

**Definiremos realidad como todo lo que existe, Incluyéndonos a nosotros mismos. La realidad es posible modificarla y está formada por entidades. Se entenderá por entidad a cualquier objeto cuya existencia está comprobada: una persona, un lugar, una cosa. Todas las entidades tienen cierto tipo de propiedades, es decir, características muy particulares. Estas diferencias nos permiten distinguirlas con nuestros sentidos.**

Las entidades generalmente se agrupan de acuerdo con sus características, en conjuntos. La palabra "lápiz" por ejemplo, es un conjunto de entidades que tiene la misma propiedad: en esencia ser un objeto para escribir.

Color, consistencia, dimensiones, sabores, etc. son ejemplos de las muchas propiedades que hacen diferente a una entidad de otra. Precisamente gracias a ésta diversidad de características de los objetos es que la realidad existe.

### **Modelo**

**Un modelo es una simplificación abstracta de la realidad.** El hombre requiere de esta herramienta para enfrentarse al mundo real y modificarlo. Dichos modelos a su vez ayudan a establecer la comunicación y por tanto nos sirven como base para el conocimiento. Las abstracciones se emplean para trabajar después con el mundo real, porque manejar la realidad tal como se nos presenta resulta muy complejo, pero si se hacen modelos (simplificaciones), podemos manipularla con menor o ninguna dificultad.

## Lenguaje

Desde que el hombre existe ha tenido la necesidad de comunicarse con sus semejantes para poder enfrentarse a la realidad. En un principio, la comunicación entre los seres humanos se diferenciaban muy poco con la de los animales; para expresar mutuamente sus deseos o necesidades, el hombre simplemente gesticulaba o emitía sonidos guturales, en nada parecido a las palabras actuales.

Sin embargo, conforme las necesidades fueron aumentando, la comunicación fue haciéndose cada vez más compleja. Manejar "ideas" o modelos de la realidad, sin mas herramienta que unos cuantos sonidos, no era ya tarea sencilla. Así comienzan a aparecer los primeros símbolos. Se determina que **el lenguaje es una compleja unión de símbolos y reglas empleada para el manejo de modelos.**

Esta definición abarca cualquier tipo de lenguaje y encaja de manera correcta cuando se habla de lenguajes de computación. (Ver Figura 1).

### 1.2 El modelo de Mayer

Mayer<sup>1</sup> es un investigador que ha sido poco reconocido y difundido; no obstante sus aportaciones a la ciencia son importantes y trascendentes. En su libro " Problem Solving" plantea un modelo aparentemente sencillo y evidente : **el modelo Entrada-Proceso-Salida-Almacén**, que ayudó a dar respuestas a problemas específicos; a éste modelo se le llama modelo de Mayer.

El modelo de Mayer ha sido ampliamente utilizado tanto en economía, como en psicología y en muchas otras ramas de la ciencia. **Constituye, de hecho, uno de los cimientos sobre el que se basaron para el invento de la computadora.**

---

<sup>1</sup> Científico alemán que vivió hacia fines del siglo XIX y principios del actual.

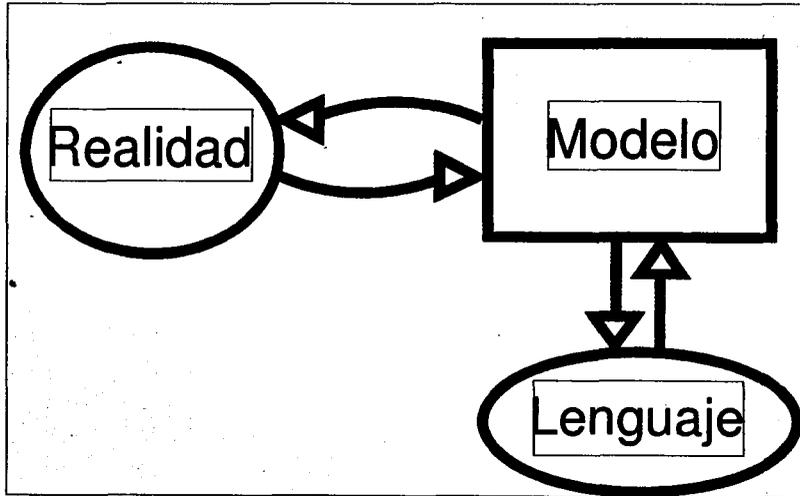


Figura 1. Concepto de Realidad, Modelo y Lenguaje.

Dos son los elementos fundamentales que Mayer aporta a la ciencia: el primero es la solución de problemas, en la que supone un doble mecanismo de generación y evaluación de ideas; el segundo se refiere al proceso creativo.

Este modelo surgió para poder explicar los resultados de los experimentos que en materia de conducta realizaron Ivan Pavlov (1849-1936) y el mismo Mayer, en los que trabajaron a base de estímulos y respuestas con animales y con seres humanos. Así, podemos considerar al estímulo como entrada, que tanto en el individuo como en una máquina opera o desencadena un proceso, el cual da como resultado una respuesta, es decir, una salida, la que no depende directamente de las entradas, sino más bien de las "experiencias" almacenadas.

El proceso o procedimiento consiste en una secuencia predefinida de operaciones, acciones y condiciones desencadenadas a partir de un estímulo, de una entrada. Norbert Wiener (1894-1964), el padre de la cibernética, supuso que este proceso se encuentra grabado o almacenado dentro de nuestras neuronas en forma similar a los bits en la memoria de una computadora.

En su libro, Mayer plantea el problema de los dos hilos para ejemplificar su teoría de que el hombre tiene dos procesos: uno que genera ideas y otro que las evalúa, y afirma que cada uno de ellos inhibe al otro. Se supone que se encuentran dos hilos colgados del techo y amarrados entre sí, alguien los desata y quedan separados; el problema consiste en volverlos a amarrar, pues al tratar de unirlos resulta que no alcanzan los brazos para sujetar ambos hilos. ¿Cuál es la solución a este problema?. Existen muchas, una podría ser que se desatara del techo un hilo, se amarrara al otro y finalmente se volviera a sujetar del techo. Otra solución sería poner una cinta adhesiva al extremo de un hilo para pegarla al techo y así, tomando el otro hilo, unirlos. También se puede utilizar un tercer objeto, digamos un zapato, para balancear uno de los hilos y así acercarlo al otro.

Se pueden encontrar muchas más soluciones si ponemos a funcionar el proceso de generación de ideas, y al final decidir cuál es la mejor opción. La generación o tormenta de ideas es uno de los métodos empleados en la administración para el logro de las metas planteadas.

El modelo de Mayer , modernizado, ha dado origen a la "teoría de sistemas". (Ver Figura 2).

### **1.3 Sistema**

Todas las entidades se encuentran interrelacionadas de manera muy estrecha y ubicadas en un medio ambiente. Para realizar cualquier acción es fundamental comprender el marco que la rodea, ya que cualquier movimiento de alguna de las partes puede provocar efectos en el resto. Así, se ha creado el concepto de sistemas usado para referirse a entidades complejas que están formadas a su vez por otras entidades: el cuerpo humano, una empresa o una institución , por ejemplo. La importancia de los sistemas es que es posible planear u operar de manera más acertada.

La noción de sistema es el resultado del desarrollo de conceptos como realidad, modelo, lenguaje y modelo de Mayer.

Existen diferentes clasificaciones de sistemas, la más común es la que se divide en naturales y artificiales; la diferencia principal entre unos y otros radica en la rapidez con que evolucionan. Los sistemas naturales tienen una evolución muy lenta, no van de manera directa a un objetivo; en cambio los sistemas artificiales, creados por el hombre, se encaminan directamente a un propósito, ya que es la consecución de un fin específico para lo que fueron hechos.

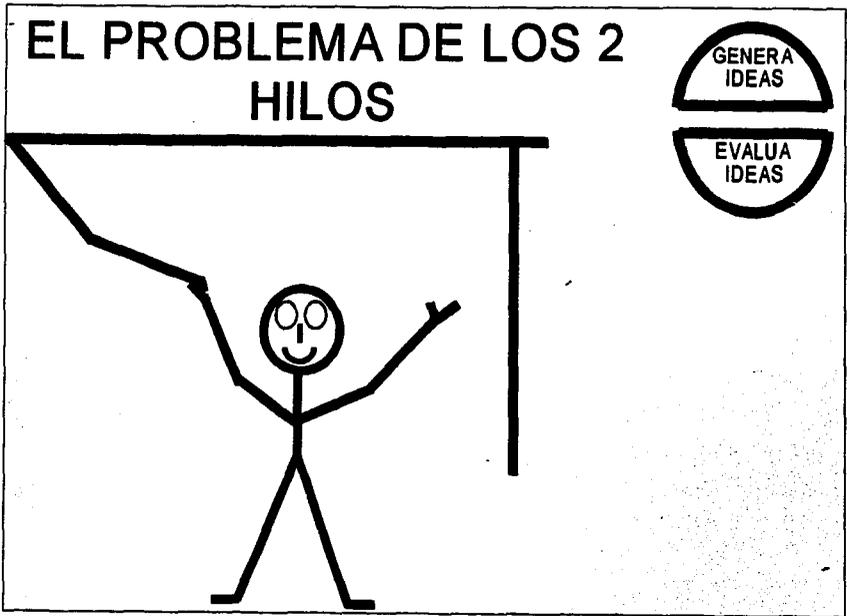


Figura 2. Problema de los dos hilos de Mayer

**Se entenderá como sistema al conjunto de entidades relacionadas coherentemente entre sí por alguna forma de organización.** Dichas entidades interactúan para alcanzar un objetivo, ese trabajo conjunto está motivado por el ambiente, por lo tanto el sistema tiene entradas, procesos y salidas. Murdick define lo que es un sistema de manera mas concisa: " Un sistema es esencialmente un grupo de elementos que funcionan juntos para alcanzar objetivos comunes".<sup>2</sup>

El sistema existe porque el ambiente lo requiere y sólo sobrevive gracias a la función que cumple. El sistema para cumplir con su función principal, comúnmente lo hace a través de subfunciones, subsubfunciones, etc., hasta llegar a operaciones elementales. Por medio de la forma en que el sistema está organizado podemos descomponerlo en todas las funciones y subfunciones que emplea de acuerdo con la complejidad del mismo; los niveles superiores deberán ejecutar las funciones más importantes o generales, las inferiores o dependientes ejecutarán funciones más específicas.

Todo sistema tiene como razón de existir la consecución de un objetivo y para observar la manera en que el sistema se comporta es necesario plantearse metas, es decir, etapas siempre medibles y cuantificables. Para que las metas se cumplan es necesario emprender acciones que a su vez requieran tomar decisiones previas a su ejecución. La finalidad de la toma de decisiones es optar por la mejor alternativa entre las diversas que se presentan; en la base de cualquier decisión se encuentra la información, el percibir o estar consciente de algo , distinguir un estado. La información puede obtenerse directamente o a través del proceso de datos, que consiste en clasificar, agrupar, calcular, sumar, comparar, recuperar algo grabado o escrito; para todo este proceso de datos nos servimos de las computadoras que deben de tener los programas

---

<sup>2</sup> Robert G. Murdick. "Sistemas de Información Administrativa". Prentice - Hall. México 1988.

adecuados para que los datos se transformen en información.

## 1.4 Información y datos

Conforme transcurre el tiempo, la realidad que el hombre enfrenta es cada día más compleja, los aportes científicos y tecnológicos cada vez son más abundantes; los sucesos se van acumulando y dejan una gran cantidad de experiencias que resulta difícil asimilar. Por esto, para entender por qué ocurrió algún hecho y cómo es que se desarrollan los acontecimientos, es necesario que se conozcan los elementos que en él ocurrieron: se necesita información.

La información aporta siempre elementos y experiencias que nos permiten, si se utiliza correctamente, enfrentar de manera adecuada situaciones determinadas en la vida diaria - el trabajo, el estudio, el hogar -. Actualmente, en todas las áreas en las que nos desenvolvemos, es difícil avanzar en la solución de problemas si no se cuenta con la información suficiente, se ha convertido en un elemento indispensable para toda persona que se encuentre en un proceso de superación o que se halle con frecuencia en situaciones en que se tiene que tomar decisiones, como los científicos, ejecutivos, funcionarios, etc.

Frecuentemente se utilizan las palabras información y datos indistintamente, como sinónimo. Se introducirán los términos de mundo real y mundo verbal para distinguir los conceptos de información y datos.

**El mundo real es el espacio físico donde existimos, la realidad. Todos pertenecemos a él y está poblado de entidades (personas, animales, hechos). El mundo verbal es el modelo o modelos que el hombre ha construido para comprender lo que nos rodea; en él se definen atributos, cada uno de ellos con sus respectivos valores y propiedades con base en las experiencias que el hombre**

**ha adquirido. Los datos pertenecen al mundo real; la información al verbal.**

Todo lo que se sabe de una entidad puede ser representado siempre en atributos y valores de él. Por ejemplo hablemos de Andrés Monte: "Andrés Monte trabaja en la Unidad Control de Gestión e Informática de la C.F.E., es Analista Programador y gana N\$2,200.00 mensuales; es casado y tiene 3 hijos; vive en Altares 567 departamento No. 5; su teléfono es 567-89-12".

Poseemos modelos o atributos que representan las propiedades que pudiera tener y se representan en el mundo verbal.

Nombre = Andrés Monte, nombre es el atributo, Andrés Monte es el valor; dirección=Altares 567 Depto. No. 5, dirección es el atributo, Altares Depto. 5 es el valor.

Lo mismo puede hacerse si se desea hacer referencia a un artículo, a un alumno, a una cuenta bancaria o a cualquier otro elemento de un conjunto de entidades; el primer concepto es una clase de propiedad o característica en el mundo real; el segundo es una de las posibilidades que puede tomar el atributo. Todo lo que conocemos y podemos decir de Andrés Monte o de cualquier otra entidad está expresada siempre en términos de los modelos que se usan.

En la práctica esto suele complicarse ya que existen diversos conjuntos de entidades relacionadas entre sí; en una escuela, por ejemplo, los alumnos están relacionados con los grupos y con los profesores, sin embargo se debe distinguir cada conjunto individualmente.

**Estar informado es darse cuenta de algo, estar consciente de algún fenómeno, percibir las características y propiedades del mundo real de acuerdo con los modelos previamente elaborados.** En el proceso que implica la toma de decisiones, hablar de información es referirse a los elementos que permitan aproximarse a la situación actual y predecir la futura, de manera que el grado de incertidumbre se reduzca; para que la información pueda ser útil en la toma de decisiones deberá tener como características esenciales ser veraz, completa y oportuna; de no ser así no sólo

no es útil, sino que podría ocasionar decisiones erróneas y costosas.

**El registro de los sucesos, de los resultados de un fenómeno, las magnitudes o cifras de observación directa, sean numéricos o no, son catalogados como datos;** grabados o escritos pueden ser leídos o interpretados por distintas personas e incluso en diferentes épocas. Se refieren a las entidades del mundo real, a sus características y diferencias y pueden ser transportados, copiados y reproducidos un sinnúmero de veces.

Es posible tener datos y eso no significa que poseamos información, ya que por sí mismos es probable que no sean útiles; deben ser leídos y procesados para estar en condiciones de servir; esto significa clasificarlos, intercalarlos, compararlos, resumirlos, etc. De hecho las computadoras sólo procesan datos, que teóricamente representan la realidad, los resultados de esta tarea debe utilizarse como información para la toma de decisiones.

La información afecta al comportamiento; en cambio, los datos no lo hacen por su formato o por su ubicación en relación con el posible usuario en determinado momento. Los datos son símbolos que describen un objeto, condición o situación particular.

## **1.5 Sistemas informáticos**

El ambiente en el que se desenvuelven las personas y las empresas, la enorme competencia y las diferentes tecnologías que se manejan, hacen indispensable el conocimiento de la situación real, de la información veraz y oportuna que conduzca a decisiones que permitan sobrevivir y avanzar. Para contar con ello en todo momento se han creado los **sistemas de información que son una serie de pasos para procesar sistemáticamente datos, asegurar su validez, almacenamiento y su oportunidad. El concepto de sistema de información incluye el uso de computadoras para**

**agilizar este trabajo. "... un sistema informático está constituido por las personas, máquinas y procedimientos que se usan para llevar a cabo una tarea informática o de procesos de datos".<sup>3</sup>**

Actualmente ya no se hacen sólo análisis de datos, estudio de formas o flujo de papeles, porque en realidad lo que importa no son éstos o las soluciones parciales o temporales; lo fundamental es el análisis de información; es decir, qué requiere una persona para desempeñar su puesto eficientemente y sin problemas, y sobre todo, qué necesita saber de las demás personas relacionadas con él.

Los sistemas informáticos son prácticamente indispensables en todas las empresas -pequeñas, medianas o grandes- y se desarrollan día a día, obteniéndose mejores resultados y extendiendo su aplicación cada vez a más áreas.

## **1.6 Generador de Aplicaciones.**

Un generador de aplicaciones es un conjunto de programas o elementos de software que permiten especificar toda una aplicación a un elevado nivel. Incluyen la capacidad para desarrollar aplicaciones, aceptar datos de entrada, validar datos, realizar cálculos y seguir una lógica de proceso, interactuar con los archivos y producir algunas salidas. Permiten que los programadores recaben con mayor rapidez la información necesaria para los usuarios. Pero, como los comandos son relativamente sencillos de utilizar, muchos usuarios que no son programadores pueden desarrollar sus propias aplicaciones, cambiar o modificar los sistemas con poca dificultad con el fin de extraer la información precisa que requieren para tomar una decisión o controlar una actividad.

---

<sup>3</sup> E. Alcalde y M. García. "informática Básica", Serie "Informática de Gestión". Mc Graw-Hill. México 1992.

El acceso más oportuno a la información apropiada puede conducir a un mayor desempeño en la organización, que es precisamente el propósito de los sistemas informáticos.

## **CAPÍTULO 2**

### **Desarrollo de Sistemas Informáticos**

El software de una computadora es un elemento muy importante en el desarrollo de sistemas informáticos y es una de las claves para que resulten eficaces y eficientes. Se puede adquirir de tres fuentes: elaboración propia, paquetes comprados y elaboración por contrato.

Al realizar un sistema informático los principales objetivos que se deben considerar son:

- a) Que el sistema sea el indicado, y,
- b) que el sistema esté bien hecho.

Las salidas producidas por el sistema deben ser precisas, confiables y completas, estando la información disponible para cuando se le requiera.

Las fases para el desarrollo de sistemas informáticos varían en casi todos los autores, algunos llaman de un nombre a una fase y la generalizan; otros la desglosan e indican subdivisiones, pero al final coinciden en los puntos principales. A éstas se les denomina ciclo de vida de sistemas.

## **2.1 Fases en el desarrollo de sistemas informáticos**

Un sistema informático nace de cierta necesidad y para su correcto desarrollo se deben de seguir ciertas fases o etapas.

Las etapas que se consideran y analizan en este trabajo son las siguientes:<sup>4</sup>

- Necesidad del sistema.
- Evaluación de la factibilidad.
- Análisis de necesidades.
- Diseño de sistemas lógicos.
- Diseño de sistemas físicos.
- Pruebas.
- Puesta en marcha y evaluación.
- Mantenimiento.

---

<sup>1</sup> Murdick menciona cuatro etapas: Planeación y programación, diseño conceptual, diseño detallado e implantación. Op. Cit. pag. 423

**Necesidad del sistema.-** En esta primera etapa los usuarios se dan cuenta que es necesario un cambio en los procedimientos o la manera de hacer determinada actividad. Esta necesidad de cambio se puede manifestar de diferentes formas:

- 1) Demasiado esfuerzo en la terminación de cierta actividad.
- 2) Excesivo trabajo para un grupo de personas.
- 3) Errores e inconsistencias en el sistema empleado actualmente, etc.

Los usuarios se percatan de que existen problemas y mediante solicitud o presentando el proyecto a la Gerencia o Departamento de Sistemas indican que es necesario la creación de un nuevo sistema que resuelva la problemática a la que se enfrentan; de esta manera al departamento de proyectos, de informática o sistemas se entera que debe de considerar la solicitud o propuesta para la creación de un nuevo sistema.

Puede ser que el interesado en promover y desarrollar un nuevo sistema sea el Departamento de Informática o Sistemas. En este caso deben considerarse aspectos importantes como son el entusiasmo e interés de los usuarios por el nuevo sistema, el grado de rechazo, etc. De no considerarse estos aspectos se corre el riesgo de que los potenciales usuarios no deseen cooperar ni brindar información que pudiera ser necesaria para para la creación del sistema; e incluso puede ser que una vez implantado no lo usen porque no lo entienden o simplemente no lo deseen.

**Evaluación de la factibilidad.-** Antes de involucrarse de lleno en la creación del sistema informático, se necesita un juicio de confianza para que el proyecto pueda ser llevado a cabo con una razonable restricción de recursos, tanto técnicos, económicos y operativos; es decir, se realiza un estudio de Factibilidad. Los tipos y

características de Factibilidad son:

**La factibilidad técnica.-** Se refiere a la disponibilidad del equipo, del software, del conocimiento que se tenga para desarrollar un sistema que corresponda a las necesidades del usuario; verificar si se tiene el equipo necesario para desarrollar e implantar el sistema, cuanto costará desarrollar el software, etc. Se deben considerar las siguientes preguntas:

- ¿Se encuentra disponible el equipo necesario?
- ¿Tiene el personal la experiencia para desarrollar el software?
- ¿El personal tiene la experiencia requerida para poder ampliarlo?
- ¿Se tiene equipo suficiente para desarrollar el proyecto en el departamento?

**La factibilidad económica.-** Se refiere a que el costo de la creación del sistema sea aceptable y que los resultados justifiquen ese costo. Se deben considerar algunas preguntas como las siguientes:

- ¿Cuál es el costo de la elaboración del sistema?
- ¿Cuál es el costo de mantenimiento del sistema?
- ¿Cuál es el costo operativo general?
- ¿Cuáles son las ventajas en comparación con los otros métodos utilizados?

**La factibilidad operativa.-** Indica el efecto que tendrá el sistema sobre los usuarios que lo utilizarán. Si la nueva aplicación no tendrá aceptación entre los usuarios

potenciales, no tiene caso desarrollarla. Se debe contestar de manera satisfactoria las siguientes preguntas:

- ¿Provocará el nuevo sistema cambios bruscos en las operaciones actuales?
- ¿Qué tanto ayudará la aplicación a los usuarios y a quienes proporcionarán los datos?
- ¿Provocará la aplicación desplazamiento de trabajo?
- ¿La resistencia por parte de los usuarios será muy grande?

Este estudio de factibilidad lo desarrollará el Departamento de Sistemas con el personal mas experimentado. Se debe de estudiar con igual detalle los tres tipos de factibilidad, ya que ocurre con frecuencia que debido a la situación de la empresa un tipo de factibilidad se estudia mas a fondo causando un desequilibrio que puede causar problemas en las etapas de implantación y de mantenimiento.

Si se determina que la propuesta para el desarrollo del sistema no es factible, se descartan o se regresan de donde procedieron para que las afinen. Si se juzga factible se inicia la nueva fase denominada Analisis de las Necesidades del Usuario, el cual señala el inicio del desarrollo de sistemas informáticos.

**Análisis de las necesidades.-** Durante esta etapa los analistas de sistemas asignados al proyecto determinan y describen las necesidades de información del usuario de tal forma que el diseño y la elaboración puedan ser llevadas a cabo posteriormente. En esta fase se adoptan los procedimientos para determinar detalladamente las necesidades; la importancia de esto radica en que se debe describir correctamente y a detalle todas las características que debe contener el sistema ya terminado.

Los analistas determinarán cómo está operando el sistema actualmente (si es que existe), de qué manera han sido satisfechas las necesidades de la información de los diversos usuarios, y dónde pueden existir problemas. Con el análisis las necesidades llegan a ser evidentes, así se pueden establecer las especificaciones para el nuevo sistema.

**Diseño lógico de sistemas.-** La información reunida durante la etapa de análisis de necesidades se utiliza para diseñar el nuevo sistema; durante esta fase interesa la formulación de las especificaciones funcionales -postulados que indican lo que el sistema debe de hacer, cómo lo que hacer, y la secuencia que deben seguir los datos de entrada, el proceso, las salidas, etc.

Es necesario la intervención del usuario para tener la certeza de que los procedimientos que se utilizan estén definidos correctamente y que el nuevo sistema esté diseñado para satisfacerlos. El sistema debe ser aceptado por los usuarios; puede ocurrir que no lo acepten si no intervinieron en su desarrollo.

En el diseño lógico se detallan los métodos de procesamiento de la información; se diseñan y se describen los documentos de entrada, las salidas, procedimientos de control, especificaciones de archivos y de base de datos, etc. Se especifican las dimensiones de los campos, registros y archivos, así como la frecuencia del proceso.

Los diseñadores pueden especificar inicialmente varios diseños posibles para los posibles usuarios y para que el personal de programación y de sistemas las analicen y las critiquen; se pueden identificar y corregir las fallas en el diseño. Los diseños inadecuados se vuelven a hacer y se plantean especificaciones más convenientes basándose en la retroalimentación de cualquiera que haya intervenido en el proceso del diseño.

**Construcción física de un sistema.-** El personal de programación empieza con la aplicación de los sistemas de información: codificación de programas, desarrollo de los formatos de los registros, estructura de datos y subesquemas en el diseño de los archivos y de las bases de datos. En esta fase es donde se estructura y desarrolla el software del sistema informático.

El software se puede adquirir de tres fuentes: elaborarlo en la organización, mandarlo hacer con empresas que se dedican a la elaboración de software y/o comprar paquetes disponibles en el mercado. La elección que se realice dependerá del tamaño del sistema, complejidad, importancia, etc.

Antes de que la organización adopte el sistema de información y forme parte de ella, se deben tomar precauciones para asegurarse de la precisión y exactitud del software, archivos y procedimientos. Cada aplicación debe ser examinada cuidadosamente para eliminar todas las dificultades; esto se realiza probando todo el software, el sistema de información en su conjunto y los procedimientos que apoyan las diferentes actividades del procesamiento.

**Pruebas.-** Se deben realizar primero las pruebas de todos los programas y posteriormente efectuar la prueba de todo el sistema; por lo general, un programa bien escrito se encuentra dividido en un conjunto de módulos integrados aunque lógicamente distintos. Cada módulo se prueba por separado para detectar errores; luego pueden conectarse entre sí y probar el programa completo como un todo; al observar primeramente los módulos separados se pueden aislar los errores de codificación y de lógica, evitando así los problemas que puedan surgir durante las conexiones entre módulos. No se debe ni pensar en iniciar la utilización del sistema sabiendo que contiene errores de lógica.

Cuando se terminan de probar todos los programas, se procede a probar toda la

aplicación. Este es el punto crítico en el desarrollo de sistemas; se efectúa para determinar qué tan bien se comportará el sistema y si cumple con las especificaciones originales. Con la prueba de sistema se tiene la certeza de que los diferentes módulos y los programas de esa aplicación son compatibles y que se pueden llevar a cabo con éxito las operaciones deseadas.

Es en las pruebas donde se debe descubrir cualquier problema de incompatibilidad que pueda surgir entre los programas o entre los módulos, así como también determinar si los espacios designados a los archivos y las dimensiones de las tablas son los necesarios para la aplicación.

La prueba del sistema es la última oportunidad para verificar el sistema antes de que se ponga en marcha.

**Puesta en marcha y Evaluación.**- Una vez que ya se han completado todas las pruebas y se observa que el sistema funciona adecuadamente (al mejor juicio de los usuarios) se puede poner en operación.

Los usuarios no son las únicas personas que van a tratar directamente con el sistema, sino también aquellos que proporcionarán datos o formas, y otros, que de alguna manera utilizarán informes y los documentos que genera. Todas estas personas deben estar concientes de la importancia de sus papeles y la manera en que afectan o son afectados por las actividades o funciones del sistema.

A medida que los usuarios conozcan mejor la operación y uso del software, el sistema se utilizará en toda su potencia en forma regular en la organización.

Durante y después de la puesta en operación se hacen las evaluaciones del sistema y de su operación. Los usuarios y el personal que se encargó de desarrollar el sistema desearán tener la seguridad de que funcione adecuadamente y que proporcione los resultados que se supone debe dar.

**Mantenimiento.**- El desarrollo de un sistema no se da por terminado cuando ya ha sido instalado y puesto en operación. En casi todos los sistemas deben hacerse continuamente mejoras, ya sea para corregir errores, satisfacer nuevas necesidades o para aprovechar la nueva tecnología. En algunas ocasiones, parte del grupo de personas que participó en el desarrollo de sistemas es asignado por un período de tiempo (meses o años, incluso) al área en donde se estructuró el sistema, y concederles la responsabilidad para el mantenimiento del mismo.

(Ver Figura 3).

**Papel de HOT en el Ciclo de Vida de Sistemas.**- Aún cuando HOT no elimina ninguna fase del ciclo de vida de sistemas sí introduce algunas diferencias en cuanto a la concepción de algunas de estas etapas, por ejemplo, con Hot las pruebas se pueden realizar en el momento en que se está desarrollando la aplicación. Esto, en un momento determinado, puede considerarse una ventaja, porque si el sistema no está quedando como fue concebido se realizan los cambios pertinentes antes de que el desarrollo siga avanzando.

La herramienta Hot permite ir corriendo la aplicación a medida que se va desarrollando; los avances que se van logrando con el sistema los puede ir evaluando el usuario, de modo que la aceptación se va haciendo gradualmente.

Una característica importante en Hot es el poder adaptar aplicaciones ya elaboradas para implantarlas a otras sin trabajar demasiado, esto significa que conociendo la estructura general de una aplicación, podemos borrar o cambiar objetos (Tareas, Procesos, Menús, etc.) que no sirvan para el sistema que se trata de desarrollar; insertando y creando aquellos que son necesarios y útiles para el correcto funcionamiento de la nueva aplicación.

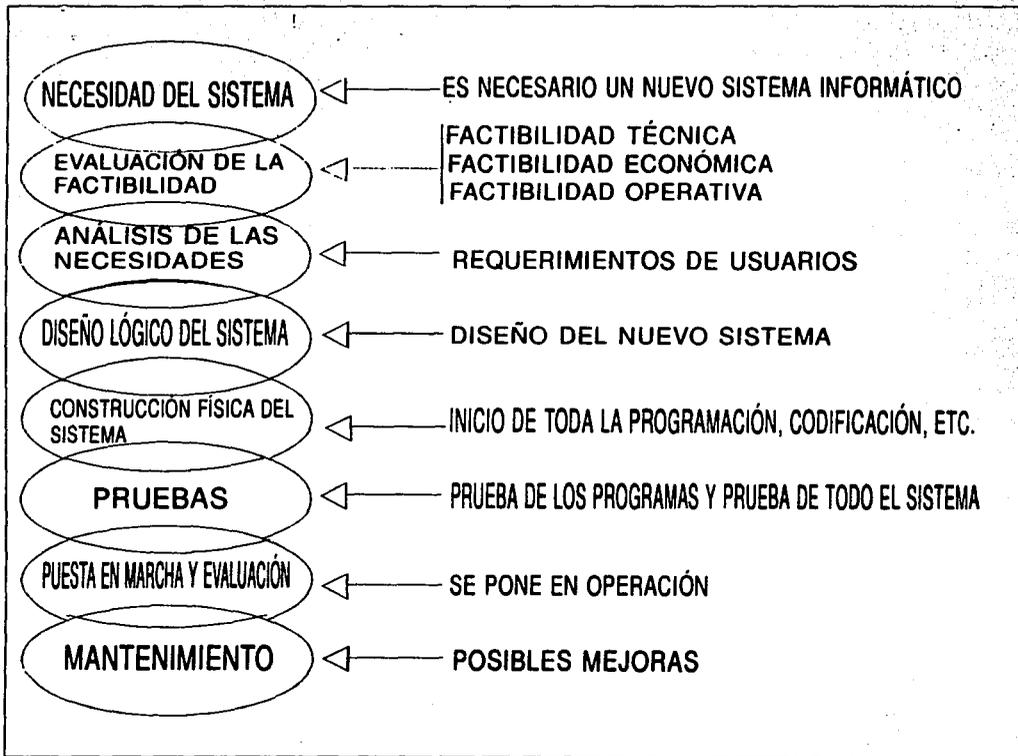


Figura 3. Principales Fases en el Desarrollo de Sistemas Informáticos

Los objetos en Hot se pueden generar, copiar, mover, cambiar, adaptar y borrar de manera simple. Los objetos son en sí los elementos de un sistema informático: los menús, las tareas, los procesos, las tablas, las celdas, etc. donde al utilizarlas adecuadamente podremos fácil y rápidamente desarrollar nuevas aplicaciones.

## **2.2 Los distintos tipos de herramientas para el desarrollo de Sistemas Informáticos.**

Para el desarrollo de sistemas informáticos se han utilizado diversas herramientas como son:

- Lenguajes ensambladores y macroensambladores
- Lenguajes de tercera generación o lenguajes orientados como son:
  - Fortran
  - Cobol
  - Basic
  - Pascal
  - lenguaje "C", ..
  - ..etc
- Lenguajes de cuarta generación o generadores de programas.
- Herramientas CASE (Computer Aided Software Engenering)
- Sistemas manejadores de Bases de Datos
- OOD (Object oriented Design)
- OOP (Object Oriented Programming)

Cada día que pasa se construyen y/o mejoran nuevas herramientas, el objetivo es facilitar el desarrollo de sistemas informáticos, menor esfuerzo, costo y tiempo con

mayor flexibilidad, adaptabilidad y capacidad.

Cuando se va desarrollar un sistema informático se debe decidir qué herramienta se va utilizar. Para auxiliar en esta elección se ha desarrollado un modelo que agrupa las diferentes herramientas que están disponibles actualmente. El modelo se clasifica en tres grupos:

- Paradigma orientado a procesos.
- Paradigma orientado a datos.
- Paradigma orientado a objetos.

Un paradigma equivale a un modelo a seguir, a un enfoque determinado para resolver tipos de problemas.

La clasificación es importante porque ubica de manera clara cuál va ser el enfoque que se utilizará para resolver el problema y así desarrollar el sistema de información.

A continuación se mencionan de manera concisa las características principales de los paradigmas considerados, esto con la finalidad de ubicar concretamente en dónde está clasificado.

### **2.2.1 Paradigma orientado a procesos.**

El paradigma orientado a procesos, se fundamenta en definir programas de computadora basados en qué instrucciones se van a ejecutar, en qué orden y bajo qué condiciones.

Un proceso, en computadora, consiste en una serie de acciones y condiciones. Las acciones son instrucciones imperativas que se ejecutan en un cierto momento después de que son copiadas de la memoria a la unidad de control. Estas instrucciones son:

- de tipo aritmético (suma, resta, multiplicación, división, números enteros o con

decimales o con notación científica, mantisa y exponente, decimales o binarias, etc..),

- de tipo entrada/salida (lee una tarjeta, acepta dato de un teclado, lee o graba desde o hacia un disco o cinta magnética, despliega en un video, imprime, etc.)
- de tipo movimiento, mueve un dato de una posición a otra, edita (puntos, comas, ..), mueve desde o hacia la memoria o los registros de la unidad de control, etc...
- de tipo llamado a subrutinas, ...

Las condiciones determinan la lógica u orden de ejecución de las instrucciones. Ha habido muchas instrucciones pero con las siguientes tres estructuras de control se puede hacer todo:

- Sequence, se ejecuta una instrucción después de otra, esto se utiliza cuando el proceso se ejecuta una sola vez.
- If then else, se ejecuta uno u otro proceso depende del resultado de la evaluación de una expresión o predicado, esto se utiliza cuando a veces se hace "A" o a veces se hace "B".
- Do Until, se ejecuta un proceso, 0, 1, 2, .. o n veces de acuerdo al resultado de la evaluación de una expresión. Si el resultado es cierto se termina la ejecución, si es falso se repite otra vez. Esto se utiliza cuando un proceso se va a ejecutar 0, 1 o n veces.

Los procesos están formados por una serie de instrucciones de tipo de acciones o condiciones. El proceso determina el orden o secuencia en que estas se deben ejecutar, a veces se le llama proceso, algoritmo, secuencia, rutina, sub-rutina, método, regla, ley, etc...

Este paradigma se utiliza desde cualquier lenguaje de programación tradicional

como serían los ensambladores, macro-ensambladores, Fortran, Cobol, Basic, pascal, lenguaje "C", etc. Aunque es el más antiguo también es el más flexible, el más eficiente en cuanto al uso de los equipos, pero paradójicamente es el menos indicado para obtener resultados rápidos por la complejidad y especialización que requieren.

### **2.2.2 Paradigma orientado a datos.**

Debido a la conveniencia de estructurar los datos, y a la enorme importancia de facilitar la programación para el almacenamiento y recuperación de datos almacenados en dispositivos magnéticos - sobre por todo la reducción del costo y el tiempo -, surgieron una serie de técnicas que son agrupadas en el paradigma orientado a datos.

La característica es el poder almacenar y recuperar datos del mejor modo apegado a la realidad. Se deberán definir archivos o tablas por cada conjunto de entidades diferentes y tablas que las relacionen. Una base de datos es un conjunto de archivos interrelacionados. Como ejemplo de algunas técnicas dentro de este paradigma se pueden considerar las bases de datos jerárquicas, de red y relacionales, el método de Jackson,.. etc.

De este paradigma han surgido una serie de paquetes comerciales que han tenido mucho éxito como son: Oracle, Informix, Ingres, Unify, Sybase, dBase, etc.

Adicionalmente han surgido paquetes para facilitar el análisis y la normalización de datos e incluso algunos sirven para la generación de código fuente; es el caso de los paquetes llamados CASE (Computer Aided Software Engineering).

Con este paradigma se han resuelto muchas aplicaciones a nivel mundial (aplicaciones sobre Control de Calidad, de Costos, de Producción, Contabilidades, Cobranzas, etc.) pero aún resulta difícil y lento el desarrollo de nuevas aplicaciones.

### **2.2.3 Paradigma orientado a objetos.**

Este es el nuevo y mas reciente paradigma que se ha desarrollado. Los resultados de su aplicación en unos casos han sido impresionantes, aunque en otros el cambio de "filosofía" no ha podido asimilarse.

Los objetos, nombre por demás común, se pueden considerar como estructuras de datos, y también se toman, en algunas ocasiones, como rutinas o procesos. En este paradigma se encuentra el Diseño Orientado a Objetos (OOD), en inglés, Object Oriented Desing, y la Programación Orintada a Objetos (OOP), Object Oriented Programming.

El análisis orientado a objetos, el diseño orientado a objetos y la programación orientada a objetos, comprenden un conjunto de actividades de ingeniería del software para la construcción de sistema orientado a objetos. Estos métodos permiten obtener el modelo de un problema representando clases, objetos, atributos y operaciones como componentes principales de modelización.

Los objetos modelizan casi cualquier aspecto identificable del ámbito del problema: entidades externas, cosas, sucesos, papeles, unidades organizativas, lugares y estructuras pueden ser representadas como objetos.

Un objeto es una abstracción muy cómoda y poderosa, muy fácil de compactar, es un modelo muy útil para hacer representaciones de la realidad.

Actualmente, por ejemplo, en un CAE (Computer Aided Engineering) aplicado a la Ingeniería química, un objeto podrá ser una caldera, un reactor químico, una torre de destilación fraccionada, un cambiador de calor, etc. y en base a esto el diseño de una nueva planta química se podrá hacer casi como un juego, la computadora calculará los flujos, las temperaturas, entalpías, entropías, etc. El diseño será en adelante algo mas fácil de hacer, cambiar, adaptar, etc... Esto mismo se ha aplicado a otras ingenierías como son civil, mecánica, eléctrica, electrónica, etc.

El Diseño Orientado a Objetos, crea una representación del campo del problema

del mundo real y la hace corresponder con el ámbito de la solución, que es el software. Se puede anunciar la metodología del OOD de la siguiente manera:

"...el diseñador puede crear sus propios tipos abstractos de datos y abstracciones funcionales y hacer corresponder el campo del mundo real con esas abstracciones [objetos] creadas por el propio programador. Esta correspondencia será la mayoría de las veces mucho más natural, ya que el rango de tipos abstractos de datos que puede inventar el diseñador es virtualmente ilimitado..."<sup>5</sup>

Esto se puede clasificar dentro de este paradigma porque pertenece al Diseño Orientado a Objetos (OOD). Se basa en creación, modificación y eventual borrado de objetos prácticos para el desarrollo de sistemas, como son los Menús, Tareas, Procesos, etc.

---

<sup>2</sup> PRESMAN S, Roger. Ingeniería de Software, un Enfoque Práctico. Tercera Edición, pág. 415.

## **CAPÍTULO 3**

### **LA HERRAMIENTA HOT.**

Actualmente existe la necesidad de poner en marcha proyectos automatizados en diferentes áreas administrativas en el menor tiempo posible y con las mejores razones de costo-beneficio. HOT cumple con estas características y se presenta como una alternativa para el desarrollo de sistemas informáticos.

HOT es una serie de programas que facilitan el desarrollo de sistemas informáticos. El diseño fue inspirado en el uso de la celda de datos como un objeto en forma similar a los paquetes que manejan hojas de cálculo electrónicas. La celda es un espacio reservado y delimitado para la captura de caracteres, incluyendo fórmulas, no se puede revolver con otra celda:

- 1.- Las celdas de datos pertenecen a una forma (o documento).
- 2.- Cada celda de datos tiene una fórmula única que las evalúa.

En el ambiente HOT el usuario ensambla los documentos de su aplicación insertando celdas y definiendo sus fórmulas de evaluación. De esta manera el usuario

elabora sus formatos (pantallas), reportes o combinaciones de formatos y reportes para su aplicación.

Los objetos de HOT necesarios para una aplicación son:

- \* Ventanas
- \* Nodos (documentos jerárquicos, menús).
- \* Celdas
- \* El texto que complementa la presentación.

Los objetivos principales de HOT se enuncian de la siguiente forma:

- \* Producir aplicaciones de calidad.
- \* Reducir el esfuerzo de desarrollo.
- \* Flexibilidad y facilidad de mantenimiento.
- \* Portabilidad de programas.

### **3.1 ¿ QUE ES HOT ?**

Es una herramienta de programación que facilita el desarrollo y ejecución de sistemas informáticos con las siguientes características generales:

- Utiliza la tecnología más reciente de orientación a objetos, sin embargo, no se requiere conocer C++ ni el uso de bibliotecas especializadas.
- El sistema desarrollado resulta fácil de utilizar, muy eficiente en el consumo de recursos, (tipos de equipo, cantidad de memoria, tiempos de respuesta, capacidad de los archivos, con ayudas al usuario, etc), muy confiable y fácilmente transportable.
- HOT se especializa en sistemas informáticos de tipo administrativo en base a transacciones.

- Es, de hecho, un lenguaje de programación, aunque los usuarios generalmente solo utilizan el medio ambiente automático de generación de aplicaciones a base de inserción, borrado y modificación de objetos.

HOT es un generador de aplicaciones administrativas, las cuales llegar a ser tan grandes y complejas como se quiera. Se pueden lograr resultados rápidos, confiables, perdurables y eficientes en cuanto a la optimización de recursos materiales y humanos.

Los sistemas producidos por HOT pueden ser instalados en diversos ambientes abarcando los siguientes:

- MS/DOS modo Monousuario      El más común de los sistemas operativos que corre en los equipos IBM PCs y/o compatibles. En este se basa la presente investigación y las aplicaciones.
  
- MS/DOS modo multiusuario      Se puede conectar una o dos terminales o PCs simulando terminales en cada uno de los puertos seriales. Solo es multiusuario en HOT.
  
- MS/DOS modo red      Se instala HOT en el servidor, los equipos o estaciones de trabajo conectados utilizan el HOT del servidor.
  
- UNIX o XENIX en PC      Se utiliza UNIX desde una PC en forma multiusuario con múltiples consolas (MVGA), con terminales asíncronas (multipuerto tonto o inteligente) o con red UNIX (Ethernet,..)



**Software**

MS-DOS 2.x o mayor  
CONFIG. SYS buffers=10  
files=20  
shell=c:\command.com/p/e:512

HOT tiene funciones poderosas que permiten tomar partes de diferentes aplicaciones y unir las en una nueva, o indicarle a una tarea que se ejecute sobre otro(s) archivo(s), logrando mayor velocidad en la creación de ciertos sistemas.

Cuenta con un reorganizador automático que actualiza en todas las tareas de la aplicación todas las modificaciones hechas a la definición de la base de datos, incluso después de haber generado información.

HOT está diseñado, como ya se mencionó, para trabajar en ambientes multiusuario en diferentes plataformas y con Portabilidad total. Las mismas aplicaciones y archivos de datos generados en MS-DOS pueden ser copiados y utilizados en XENIX, UNIX de SCO y las demás versiones de UNIX como la AT&T, ULTRIX, etc.

La eficiencia de HOT en cuanto al consumo de recursos es óptima, ya que ocupa mucho menos espacio tanto en RAM como en Disco en comparación con ORACLE, INFORMIX y otras herramientas parecidas. Esto permite desarrollar sistemas con equipos accesibles como PC's con 640 K bytes en RAM y un disco duro, y correr las aplicaciones en máquinas más grandes, o bien correr en PC's multiusuarios (UNIX de SCO) aplicaciones tan complejas que típicamente son vistas en computadoras grandes.

HOT optimiza al máximo el recurso humano al amplificar la productividad de los programadores especializados y al poner al alcance de los usuarios menos especializados todo el poder de la información manejada con una herramienta sofisticada y no tan difícil de usar. Puede integrarse fácilmente en empresas donde se usen otros sistemas y paquetes ya que cuenta con funciones de exportación e importación de

sistemas de datos a formato ASCII y exportación a hojas de cálculos tipo Lotus.

Integra en un solo concepto todas las herramientas necesarias para lograr resultados rápidos y eficientes cuya especialidad es la generación de paquetería tanto comercial como institucional.

### **3.2 OBJETOS HOT.**

En HOT se utiliza el concepto de programación visual, se insertan, borra, mueven y cambian los objetos. No se requiere de programación en forma tradicional.

Los principales objetos son:

- La aplicación (todo queda integrado)
- Las tablas (archivos indexados)
- Las celdas (campos de datos)
- Los nodos jerárquicos (Menús y/o Tareas)
- Las ventanas (una parte de la pantalla)
- Los documentos (hojas o transacciones)
- Los registros (celdas de una misma tabla)

Los objetos tienen atributos, el desarrollador va colocando o ajustando los objetos y sus atributos.

Se muestra una breve explicación de los objetos, posteriormente se efectúa de manera mas extensa:

#### **APLICACIÓN**

Un solo archivo con toda la aplicación, las definiciones de tablas, la estructura jerárquica de menús y tareas,

	las definiciones de cada tarea, ventanas, niveles, etc.
<b>TABLAS</b>	Equivale a los archivos secuenciales con índices.
<b>MENÚS</b>	Nodos para navegación de los usuarios.
<b>TAREAS</b>	Procesos de transacciones simples o complejas.

### **Objeto aplicación**

La descripción completa de una aplicación o sistema informático en HOT está almacenada en un solo archivo con el nombre de la aplicación y con una extensión ".apl". En este archivo se encuentra en forma "monolítica", la estructura jerárquica de la aplicación, sus tablas, sus menús, sus tareas, etc. es decir toda la definición de la aplicación.

Este archivo se crea, consulta y modifica por el módulo de desarrollo y se utiliza como base en el módulo de ejecución.

La aplicación se puede exportar e importar desde y hacia el módulo de desarrollo, en forma total o parcial en forma de lenguaje. (F9 exim). En el lenguaje se definen la aplicación y sus objetos y aunque se puede modificar, prácticamente no se utiliza porque se genera automáticamente.

## **Objeto tabla**

Contiene la definición de una tabla o archivo del sistema, se incluyen todas y cada una de las celdas con sus principales atributos (Name, Frmt, Key, etc. que se verán a detalle).

Cada tabla se accederá en el módulo de ejecución como un archivo secuencial con índices, es decir podremos hacer accesos en forma secuencial o al azar después de evaluar las celdas llave.

Las celdas contendrán el valor que tengan grabado en el archivo y los nombres de las celdas servirán como nombres de las variables en las expresiones que las utilicen.

Las tablas deberán definirse previamente a las tareas ya que cada tarea se basa en la utilización de una tabla primaria y opcionalmente se podrán hacer ligas con otras tablas.

El criterio para definir las tablas es el mismo que para las bases de datos, una tabla para cada conjunto de entidades distinto con algunas tablas que las relacionen.

## **Objeto menú**

El menú es parte de la estructura de una aplicación. Nos sirve para facilitar la "navegación" a los usuarios, es decir, para seleccionar lo que se quiere hacer en forma más ordenada, agrupándose las funciones por temas.

Una aplicación puede tener o no tener nodos menú, puede tener menús, sub-menús, sub-sub menús, etc con los niveles de profundidad que se requieran.

Al momento de la ejecución el menú se presenta en forma de ventana y de manera muy sencilla como una opción al usuario que deberá seleccionar.

Al momento del desarrollo lo único que tenemos que hacer es insertar los menús en la manera más apropiada, en forma de hijo (KID) o HERMANO (Brother) y darle un nombre que oriente al usuario. Posteriormente se insertarán los sub-menús o tareas que presentarán al usuario. Con un poco de práctica definir los menús ocupará sólo unos minutos.

## Objeto tarea

Una tarea es un proceso de datos, es una opción que tiene el usuario al momento de la ejecución. Se presenta generalmente como un documento, pantalla o transacción. Corresponde a un hecho real que hay que registrar, consultar, actualizar, etc.

Las tareas podrán ser muy sencillas o muy complejas dependiendo de lo que se vaya a realizar. Una tarea sencilla podrá ser una consulta y/o actualización de una tabla, o un reporte sencillo. Una tarea compleja podrá tener varios niveles jerárquicos, ligas con varias tablas, abrirá una o varias ventanas, etc., operará una transacción hasta sus últimas consecuencias.

Durante la elaboración de una factura, por ejemplo, se podrá consultar dinámicamente los datos y el saldo de cliente con sus límites de crédito, la existencia de los artículos, las comisiones a los vendedores, la generación de la póliza contable, etc. Una vez operada la transacción todavía se podrá modificar aún lo que ya está actualizado.

Internamente existe sólo un tipo de proceso. Este proceso servirá para dar consultas y/o actualizaciones: altas, bajas y cambios; (**Ver Figura 4**). Cuando tenga captura en las celdas, servirá para barrer (procesar todos los registros) toda la tabla o podrá generar un reporte en pantalla, a un archivo, o a la impresora dependiendo de cómo se definió. Podrá asociarse con uno o más procesos similares en forma de niveles

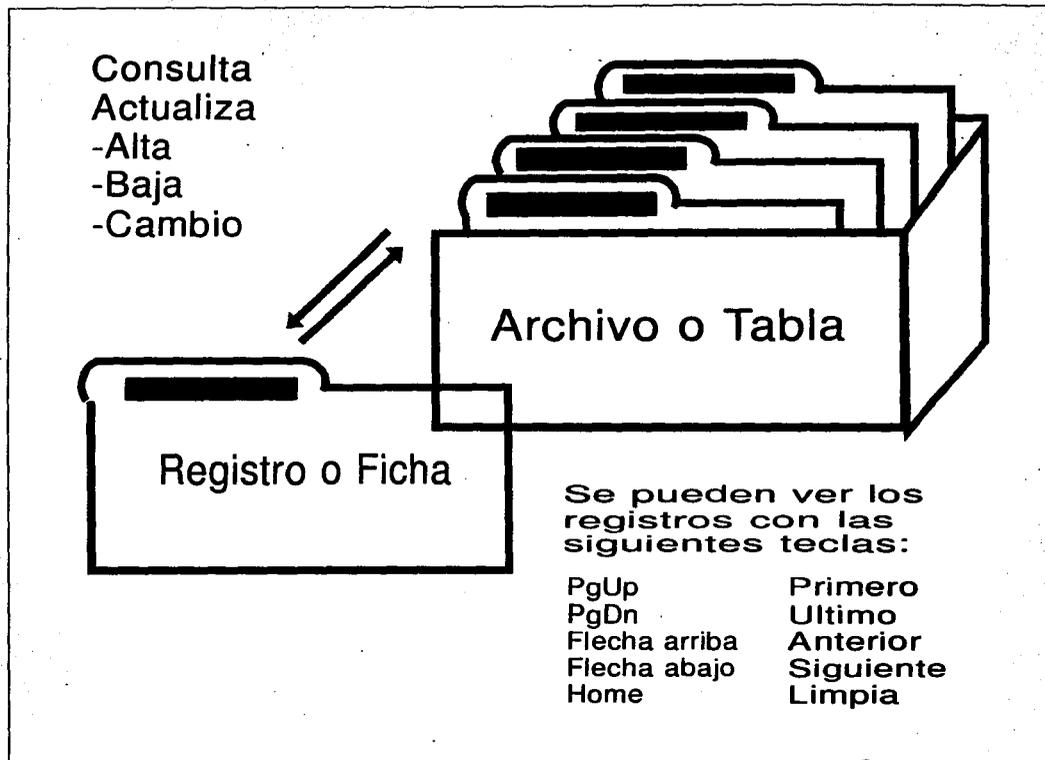


Figura 4. Proceso en HOT

(jerarquías) o ventanas. Los procesos podrán utilizar celdas locales o celdas de otras tablas ligadas una o varias veces.

Se podrá repetir una o varias veces el proceso de evaluación de celdas y en general insertando un proceso, o grupos de procesos, actualizando los valores de los atributos de los procesos y/o de las celdas se podrá lograr la operación de una transacción hasta sus últimas consecuencias.

Utilizando el diagrama de la estructura parcial de cada tarea se podrá ver rápidamente como quedó definida la tarea, que tablas, etc.

### **3.3 MÓDULO DE DESARROLLO.**

Al analizar el desarrollo de una aplicación en HOT nos encontramos con la rapidez, aspecto muy importante para la elaboración de sistemas informáticos. Esta rapidez se logra por varios factores, como son:

1. Es una herramienta no algorítmica, a diferencia de los lenguajes tradicionales y otras herramientas de cuarta generación, no hay que escribir algoritmos ni códigos ya que HOT consiste de objetos con los que se logra una auténtica programación visual.
2. Posee un depurador integrado en línea que permite pasar de desarrollo a ejecución de forma inmediata, de tal manera que no hay que compilar ni ligar, permitiendo correr la aplicación paso a paso conforme es desarrollada y hacer las modificaciones que pide el usuario casi al instante.
3. Diccionario interactivo de objetos. Poder cambiar la definición de la base de datos actualizándose automáticamente en toda la aplicación.

4. **Desarrollo a partir de prototipos.** Permite quitar y poner partes de una o varias aplicaciones para crear una nueva. Cambiar la (s) tabla(s) sobre las que actúa una tarea. Concepto de familias de objetos. (Hijos, Hermanos,...)

Antes de empezar a desarrollar cualquier sistema informático, independientemente de la herramienta que se utilice, es necesario y conveniente que previamente se elabore el diagrama Entidad/Relación (E/R) adecuado.

Un diagrama E/R muestra en forma gráfica las tablas que componen la base de datos y la relación que existe entre cada una de ellas, es decir, una tabla es un archivo que contiene datos sobre un conjunto de entidades y por medio de líneas podemos relacionar una tabla con otra, con el objeto de visualizar las relaciones que hay entre un archivo y otro.

Una base de datos es un conjunto de archivos relacionados y el diagrama E/R permite ver, definir y analizar rápidamente la estructura de una base de datos de una empresa o de una aplicación. Aplicaciones tales como una nómina, control escolar, control de producción, control de materiales, etc.

Después del diagrama es preciso tener una idea clara de los menús y de las transacciones que se van a procesar; al saber que menús se van ocupar se podrá organizar la información. Con éstos dos elementos se procede a crear una aplicación en HOT.

Primeramente se crea un subdirectorio con el nombre de la aplicación para poder tener un espacio propio para el desarrollo de la base de datos y que no se revuelva con las demás aplicaciones. Con este subdirectorio y el HOT ya instalado, se procede a dar un HOT + y el nombre de la aplicación. Este debe ser un nombre de archivo en MS-DOS con ocho caracteres alfanuméricos empezando por un carácter alfabético. Al momento de dar esa instrucción, HOT entra al módulo de desarrollo y crea una aplicación vacía: un objeto aplicación vacía. Se crea el objeto en base a los parámetros ya establecidos

y se le tiene que dar una afinada o ajuste para que quede el objeto adecuado a lo que se desea.

El Objeto Aplicación Vacío en este momento tiene un menú principal con un nombre cualquiera denominado standard menú, y una tabla cualquiera que se llama standard table. A esta aplicación tenemos que modificarla y adaptarla a las necesidades.

Teniendo ya la aplicación creada y el diagrama E/R se procede a crear tabla por tabla dentro de HOT. Para crear cada tabla se oprime la tecla F3 y posicionarse en una de las tablas; al principio solamente existe una, y se oprime la tecla INSERT para poder copiar la tabla en base al modelo seleccionado. En ese momento ya podemos cambiar los nombres y generar los atributos de cada una de las celdas. Si se requieren más celdas se pueden insertar de las tablas que se tengan en ese momento, y así sucesivamente ir creando cada una de las celdas y formando la tabla.

Cada celda tiene siete atributos, en este momento de la creación de la tabla, únicamente importan tres: el NAME, FORMAT y KEY.

El atributo NAME representa el nombre de la celda, que al mismo tiempo sirve como etiqueta o como letrero al momento de desplegarse y como variable al momento de usarse en una fórmula.

El atributo FORMAT define el tipo de celda, hay tres tipos de celda: numéricas, alfanuméricas y celda de fecha. Format define la longitud de la celda que puede ser alfanumérica y numérica; de hecho todas las celdas trabajan en forma similar, en el sentido de que se pueden almacenar en una tabla, recuperar, cambiar, etc., la diferencia es que en las celdas numéricas podemos hacer sumas, restas, multiplicaciones, divisiones, divisiones enteras, residuos; en cambio las celdas alfanuméricas son generalmente texto, con las que se pueden hacer solo algunas operaciones sobre ellas; y las celdas de fecha con las que únicamente se pueden hacer operaciones como: restar fechas para obtener un número de días que hay entre dos fechas, sumar a una fecha una constante para obtener una fecha futura, restar una

constante de una fecha para sacar una fecha anterior, etc.

El atributo KEY define si es o no es llave la celda. Cada tabla tiene una sola llave sin embargo, esta llave puede estar formada por una o mas celdas; la primera celda en orden de importancia tiene la llave número 1, la segunda tendría el numero 2 y la tercera el número 3, etc. Las celdas que tienen en este atributo el valor cero son las celdas de datos y la distinción entre una celda de datos y una celda de llave es que primero se evalúan las celdas llaves, y una vez que están los registros se evalúan los datos.

Todas las aplicaciones tienen una estructura arbórea, una estructura jerárquica a base de menús o tareas (**Ver figura 5**). Los menús, que no son estrictamente necesarios en la aplicación, permiten facilitar la navegación a los usuarios finales.

El nodo principal o raíz es un menú del cual podemos tener otros menús o submenús y tener tareas o procesos de transacciones. De esta manera se crean los menús que se deseen y las tareas debajo de los menús que correspondan. Se puede hacer que una tarea tenga uno o varios procesos y que esos procesos podrían ser a la vez menús, ventanas o niveles, etc. Ya teniendo la idea de como hacer la estructura de la aplicación, cada vez que se desee insertar un menú simplemente nos posicionamos en el lugar adecuado, oprimimos INSERT y con esto insertamos un hijo o un hermano, de tal manera que podamos dominar la estructura que se quiera.

Cuando se selecciona el menú simplemente tenemos que darle el nombre y eso es todo. Si queremos hacer una tarea, lo primero que nos pregunta la máquina es sobre qué tabla primaria se va a desarrollar esa tarea, la tabla primaria es el archivo principal de la tarea (**Ver Figura 6**).

Cuando se genera una tarea se produce automáticamente el primer proceso, es decir una tarea puede tener una estructura jerárquica de uno o varios procesos. Al principio cuando se crea una tarea se crea el primer proceso y se define cual es la tabla primaria del primer proceso.

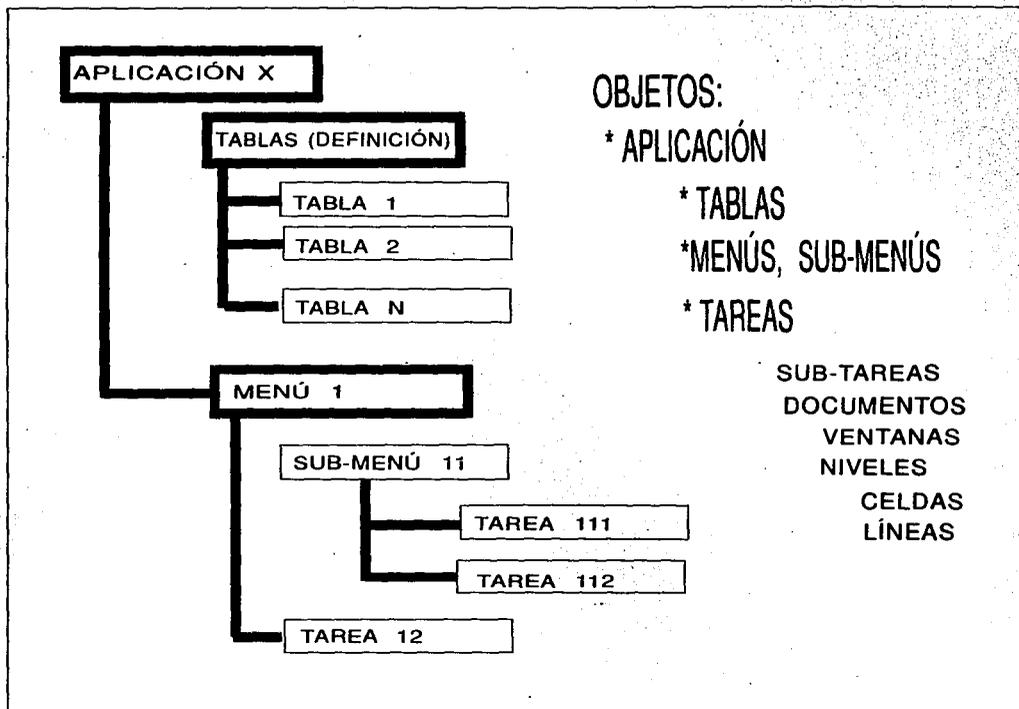


Figura 5. Estructura Jerárquica de una Aplicación

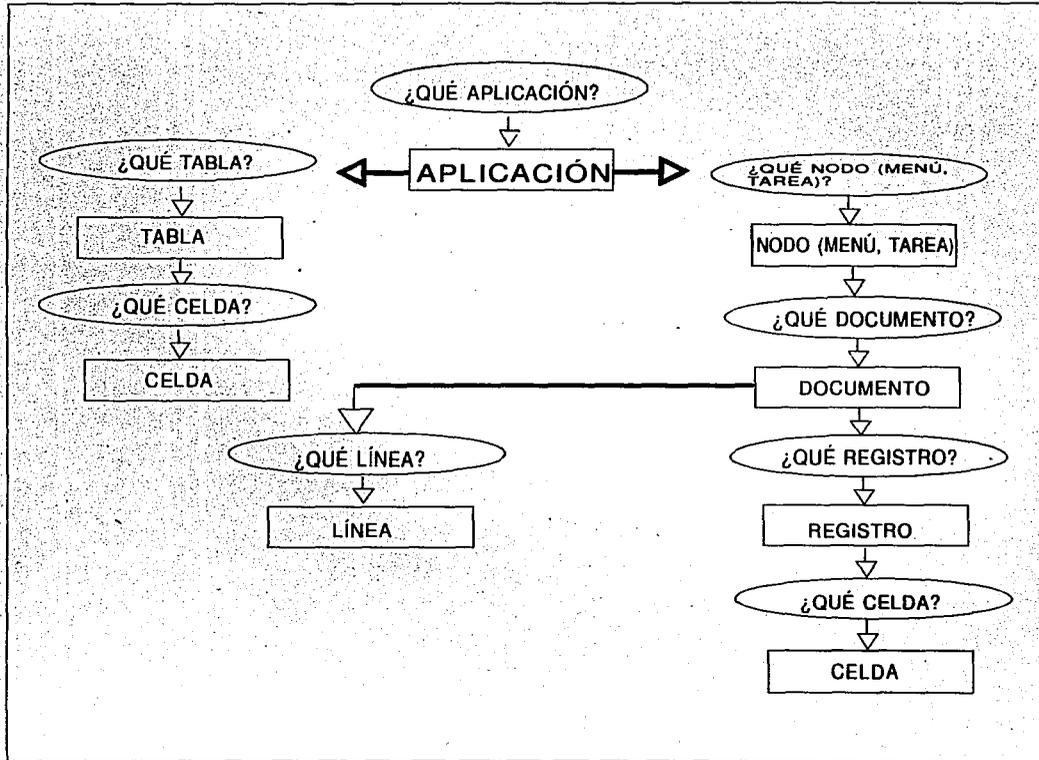


Figura 6. Módulo de Desarrollo

Existen tres tipos de procesos que se generan automáticamente, dependiendo de las fórmulas que contengan las celdas llaves. Si las celdas llaves no tienen nada - valores nulos -, la máquina lo toma como que va a procesar todo el archivo. Es como una secretaria a quien se le indica que se desea procesar el archivo de alumnos, no se le dice más; ella supone que va a trabajar desde el primero de los alumnos hasta el último. Estos procesos la máquina automáticamente les pone un símbolo +, quiere decir que efectúa un barrido a todo el archivo y típicamente se utilizan para reportes, copias de archivos o para hacer un proceso de corrección de un dato a otro etc.

El segundo tipo de proceso es cuando las fórmulas de las celdas llaves de la tabla primaria tienen una captura, eso quiere decir que el usuario va a definir cuáles son los registros que va a leer o actualizar, puede ser un registro o más, no se sabe cuantos ni cuales registros; pueden ser los primeros, los últimos, los de enmedio, del orden que al usuario se le ocurra; los define en el momento que él decida y da un ESC en el momento que quiera terminar el proceso, este es un proceso interactivo y la máquina lo reconoce con el signo de interrogación (?). Es como si se le indicara a la secretaria que se va a actualizar el archivo de alumnos y el usuario le va a decir cuales; ella va a estar pendiente del usuario a qué horas el usuario quiera consultar y/o actualizar un archivo, es decir, dar altas, bajas, cambios, reportes, etc...

El tercer tipo de proceso, que la máquina representa con un símbolo de menos (-), es el proceso de tipo específico. Se va a leer un registro, puede ser que esté, puede ser que no esté; terminándose el registro, la máquina acaba, ya que se le indicó que solamente se quiere un solo registro. Es como si se le dice a la secretaria que se va a trabajar sobre el alumno 4025; ella busca ese alumno, si está, presenta lo que tiene, y si no está lo indica, al terminar ese registro acaba el proceso.

Insertando procesos hermanos o insertando procesos hijos, podemos hacer que una sola tarea pudiera tener uno o más procesos de diferentes tipos y en diferentes

momentos. Pueden ser tipo ventanas, procesos independientes que abren una ventana, la colocan encima del video y cuando termina desaparece. Pueden ser niveles, procesos más dependientes que utilizan el mismo documento y el mismo espacio de trabajo. El usuario determina qué menús y qué tareas va a necesitar y dentro de cada tarea si es un solo proceso o son varios procesos.

Las teclas que utiliza HOT son de fácil manejo: la tecla de INSERT para insertar nuevos objetos, DELET para borrar objetos creado, con la tecla PgDown - PgUp cambiamos los atributos de cada celda como son el nombre, el formato, si es llave, si se oculta o no se oculta los datos, la tecla de MASK que sirve para seleccionar registros, la tecla de CALC que sirve para evaluar la celda dándole al posibilidad de captura o no captura, pudiendo tener un valor inicial o no, y dándole la posibilidad de validar o no.

Durante el módulo de desarrollo de HOT podemos crear verificar, revisar o cambiar una aplicación. Se utilizan algunas teclas básicas que son las teclas de función.

A continuación se da una breve explicación de las teclas básicas que se utilizan para el desarrollo de aplicaciones con HOT, esto es importante porque contribuye directamente sobre el manejo de la herramienta.

La tecla F1 que es la ayuda en línea, es una tecla sensible al momento en donde estamos trabajando, es decir, si estamos trabajando con celdas la ayuda nos la da sobre atributos de celda, si estamos desarrollando en estructuras nos va a dar ayuda en las estructuras, etc. Esta función es de tipo hipertexto, seleccionar dentro de una ayuda la siguiente pantalla de acuerdo a una palabra clave o inclusive podemos poner la inicial de la letra, por ejemplo de función una F y nos sale todas las palabras que empiezan con F o con T en dado caso que quisiéramos poner una T etc.

La tecla F2 permite ejecutar inmediatamente una aplicación, se puede estar desarrollando y en cualquier momento, dar F2 para probar, al terminar volvemos al módulo de desarrollo. Esto proporciona la facilidad de probar en forma interactiva los desarrollos que estamos haciendo en ese momento. (Ver Figura 7).

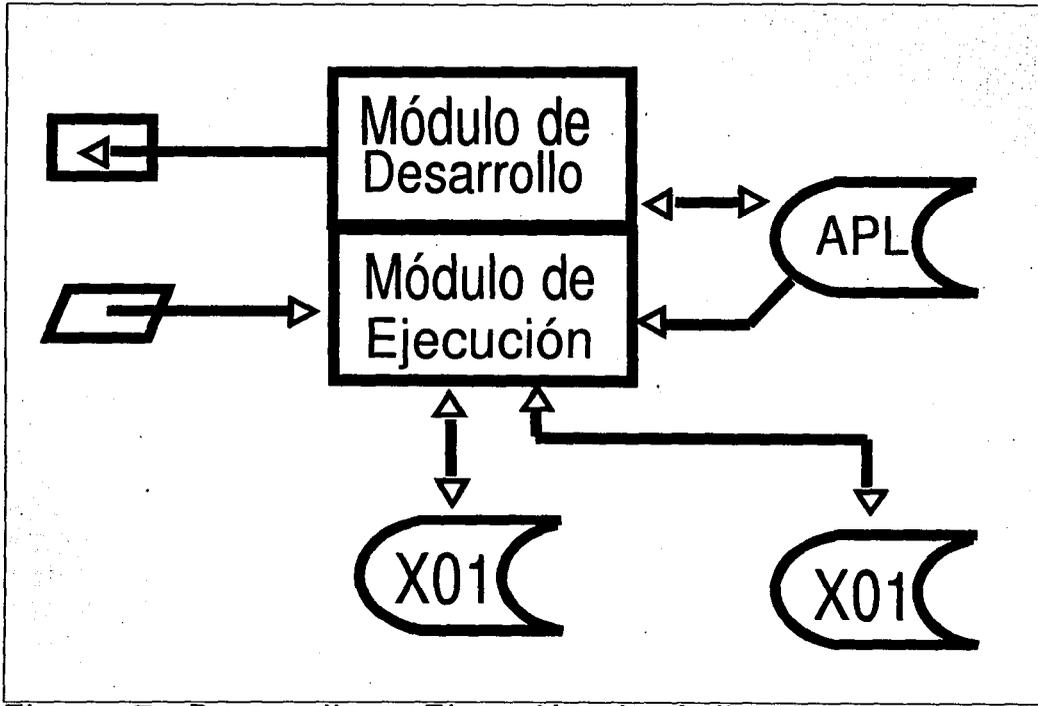


Figura 7. Desarrollo y Ejecución de Aplicaciones en HOT

La tecla F3 sirve para el manejo de tablas, proporciona la pantalla en donde se mencionan todas las tablas que tiene la aplicación. Se puede seleccionar la tabla que se desee mediante el posicionamiento del cursor con las flechas y un ENTER, al hacerlo se entra a la definición de la tabla así como al editor de celdas, donde podemos ver, celda por celda, lo que contiene cada tabla.

La tecla F5 permite definir los valores de los atributos de un nodo en particular, por ejemplo, si el nodo es una tarea, se tiene la opción de decidir si es reporte o no, si hay un file create, es decir que cada vez que se corre la aplicación se crea un archivo, y el file delete, etc. Se puede introducir el número de la ayuda, para que al momento de ejecución si el usuario está en ese momento y oprime F1 le presenta en la pantalla ayuda con el número que se haya colocado. Se tienen otros atributos como son los permisos que son letras de la "a" a la "g" de tal manera que se puede seleccionar hasta siete tipos de usuarios diferentes, por ejemplo, administradores, cajeros, lecturistas, etc. contándose con la facilidad de definir para quién, qué procesos, menús y tareas se utilizan. Lo interesante es que una sola aplicación con los permisos adecuados, presentan diferentes menús y diferentes tareas de acuerdo a cómo se registró el usuario al momento de entrar a HOT, eso se logra mediante las variables de ambiente en particular una variable que se llama HOT PERMS en donde se define qué tipo de usuario es.

La importancia de esto radica en que de acuerdo a como se registra el usuario al momento de entrar, le presenta la pantalla adecuada al puesto en que se ubica, por ejemplo, si se tratara de un sistema de cobranza de un servicio, los lecturistas podrían capturar sus lecturas mediante una pantalla específica para ellos, los cajeros podrían hacer los cobros mediante una pantalla adecuada, el administrador podría hacer lecturas pero no cobros, en fin, cada usuario podría definir su función de tal manera que nos permita la seguridad de que nadie va a registrar cobros si no es cajero, nadie va a

registrar lecturas si no es lectorista. Además de la seguridad al sistema proporciona una presentación casi automática de los diferentes menús de un solo sistema, esto es útil en sistemas medianos o grandes en donde existen diferentes tipos de usuarios; inclusive pueden estar trabajando simultáneamente mediante técnicas de procesos multiusuarios, de tal manera que cada quien realice su función, por ejemplo, si una persona viene a pagar y no tiene lectura tiene que recurrir al lectorista y ya que tiene la lectura acude al cajero para pagar; esto determina las funciones exactas que cada usuario debe tener de acuerdo al diseño que se quiera y flexibilidad deseada.

La tecla F6 proporciona los atributos de las celdas. Se puede consultar y/o actualizar todos los atributos de las celdas seleccionadas.

La tecla F7 es el editor de textos, permite a un proceso o a una tabla poder editarla. Se puede poner constantes, acomodar las celdas y colocarlas con una mejor presentación. El objeto es la línea y permite definir los atributos de la línea, por ejemplo, en caso de reporte si las líneas son encabezados, de detalle o de pie de página etc.

La tecla F8 es una tecla para salvar la aplicación, para guardarla. Es conveniente sobre todo cuando se va a hacer algún cambio en una aplicación que está funcionando. Antes de hacer la modificación hay que salvarla, de tal manera de que si lo que nosotros hagamos no funciona por alguna razón, podamos regresar a la versión anterior y no destruir lo que ya habíamos hecho y que ya teníamos como una versión estable. La tecla F8 en el momento que se da se crea una copia de la aplicación con el mismo nombre y en el mismo directorio pero con la terminación ".SAV".

La tecla F9 se utiliza generalmente para exportar algunas de las aplicaciones o partes de aplicaciones; es útil para los usuarios que deseen efectuar una aplicación similar; o al revés, podemos importar de otra persona o de otra aplicación algunas tablas o algunos procesos o aplicaciones completas mediante esta tecla. Esto es útil sobre todo en el momento en que una aplicación grande se van a desarrollar por varias personas, esto permite que cada quien desarrolle independientemente y en un momento dado,

podamos unir o separar algunas partes de la aplicación. Esta tecla nos permite reorganizar la aplicación, volverla a crear con menos espacio, más eficientemente y también verificar que no hay ningún problema en la aplicación, recompilar, etc. es una herramienta interna práctica y útil.

Combinando la tecla SHIFT con otras funciones se puede obtener por ejemplo SHIFT F2 se repite lo que anteriormente fue teclado para no volverlo a teclear, esto es conveniente cuando se quiere copiar una fórmula grande de una celda a otra celda, de una tarea a otra. Con SHIFT F7 podemos cambiar los colores de la pantalla, con la finalidad de poder visualizar de manera más precisa, tanto la aplicación como los mensajes enviados, etc.

HOT evalúa celda por celda, en un orden definido por la propia herramienta, de acuerdo a las fórmulas y la posición de éstas en la pantalla (de arriba hacia abajo, de izquierda a derecha). Si una celda depende de otra, se evalúa primero la otra. La celda llave se evalúa primero, después se lee el registro (está o no está) y después se evalúan los datos. **(Ver Figura 8).**

Las celdas tienen dos partes: la etiqueta o nombre y su valor correspondiente. Cada celda, como ya se mencionó, posee un conjunto de atributos que controlan su desempeño y su presentación. En una tabla deberá haber al menos una celda de llave y una de datos.

Las celdas pueden ser globales (para toda la aplicación) o locales (sólo para el documento), éstas pueden modificarse libremente. Algunas modificaciones a celdas globales, como el nombre, afectan a toda la aplicación.

Dentro del proceso la celda podrá ser de:

La tabla primaria                      La celda pertenece a la tabla primaria o principal.

Local                                      La celda no se grabará en ninguna tabla, se utiliza como auxiliar.

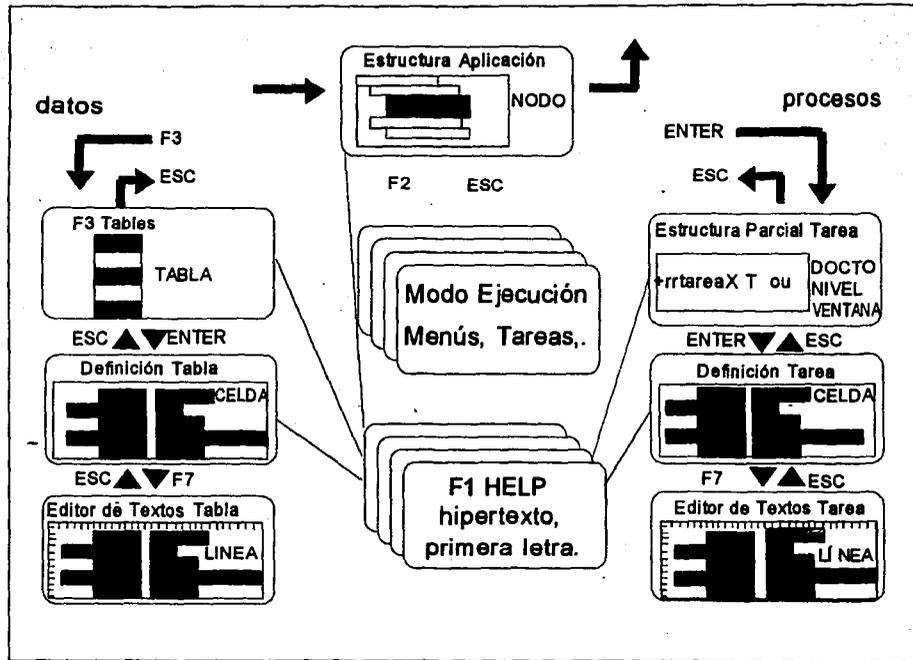


Figura 8. Diversas pantallas del Módulo de Desarrollo de HOT. Desde cada una de ellas se pueden insertar, borrar, mover o cambiar los distintos objetos.

Liga

Pertenece a otro registro de otra tabla (o de la misma)

Cada celda, como ya se mencionó, posee un conjunto de atributos que controlan su desempeño y presentación. A continuación se muestra de manera breve los atributos de las celdas, su importancia radica en que éstos llevan el control de la aplicación, indicando todo el proceso del sistema informático. Los atributos de una celda son los siguientes: NAME, FRMT, KEY, HIDE, CALC, MASK y RECORD.

### **El atributo NAME**

Este atributo define el nombre de la celda, tal como aparece en ventanas y fórmulas; posee las siguientes características relevantes:

- \* Los nombres de las celdas deberán ser únicos en cada tabla (archivo)
- \* Deberá ser alfanumérico empezando con un caracter alfabético
- \* En medio podrá tener espacios o el caracter de subrayado
- \* La longitud máxima es de 39 caracteres
- \* Podrán utilizarse letras mayúsculas o minúsculas y siempre se desplegarán tal como se definieron
- \* El HOT permite el uso de mayúsculas y minúsculas indistintamente

---

## EL atributo FRMT

Este atributo define las siguientes características de su celda:

- \* El tipo de dato (Texto, Numérico, o Fecha),
- \* El formato de despliegue (comas, justificación, relleno, etc),
- \* La longitud de despliegue,
- \* La longitud de almacenamiento (solo celdas llave),

Contiene una cadena de caracteres de longitud variable, hasta lo permitido en el formato de despliegue.

Puede estar justificada a la derecha o a la izquierda.-

La longitud máxima es de 240 caracteres.

Se eliminarán los espacios de los extremos de la cadena, los internos no se alterarán.

### Frmt - Formato texto en celdas

Contiene varias opciones de formato, las más importantes son: E, V, X

Clave	Significado	Ejemplo
E9	DÍGITOS	'1234'
V9	MAYÚSCULAS, PUNTUACIÓN ESPACIOS	'AB1.2'
X9	CUALQUIER CARACTER	'Aa1.2'

**Frmt - Formato numérico en celdas**

Se utiliza para cantidades, contadores, números seriales, etc. Su longitud máxima es de 78 dígitos (que incluye el signo, 68 enteros, punto decimal, y 10 dígitos de fracciones decimales). Las opciones incluyen el signo de pesos, comas fracciones y relleno de ceros a la izquierda. Los formatos más importantes son:

<b>Clave</b>	<b>Significado</b>	<b>Ejemplo</b>
N9.1	SIN COMAS, SIGNO, FRACCIONES	'-999.9'
\$9.1	COMAS, SIGNO, FRACCIONES, SIGNO PESOS	'\$9,999.9'
09	CEROS A LA IZQUIERDA, SIMPLE	'000099'

**Frmt - Formato de fecha en celdas -**

Las fechas pueden tener diferentes formatos:

<b>Código</b>	<b>Formato de Despliegue</b>
'Y'	'AA/MM/DD' (año/mes/día)
'M'	'MM/DD/AA' (mes/día/año)
'D'	'DD/MM/AA' (día/mes/año)

## **El atributo KEY**

Este atributo indica si la celda es una celda de llave, esto es, si la celda es parte de la identificación de los registros de la tabla o archivo. Tiene las siguientes características:

- \* Cada registro en una tabla o archivo deberá ser identificado por su celda o sus celdas de llave (Al menos una).
- \* Cada identificación del registro deberá ser única dentro del archivo. (No se pueden duplicar llaves).
- \* Los registros son clasificados automáticamente dentro de la tabla o archivo, en orden ascendente.

Este atributo es numérico, la celda con KEY = 1 es la de la más alta prioridad en la secuencia de clasificación, la 2 será la siguiente, etc.

Se podrán utilizar celdas de llave con tipo de dato texto, números o fechas.

Puede haber hasta 15 celdas llave, pero la longitud total combinada deberá ser menos de 250 bytes. Para convertir una celda no llave a una celda llave solo teclee el número que indica la prioridad de clasificación. Para regresarlo a celda no llave teclee un 0.

Los números negativos como valores de las celdas llaves, así mismo como los formatos de textos justificados a la derecha, pueden causar desordenes inesperados en los registros.

## El atributo HIDE

Este atributo de celda controla la visibilidad de la celda cuando la aplicación se está ejecutando. En modo desarrollo, todas las celdas son visibles. Durante la ejecución el medio ambiente despliega las celdas de acuerdo con el valor de este atributo. Este atributo puede también afecta la visibilidad del "nombre" o etiqueta de la celda. Cuando la celda contiene N, ambos son visibles; L, el dato es visible y la etiqueta no; B, ambos son invisibles.

## El atributo CALC

En una celda podemos tener como atributo una fórmula. Esta fórmula consiste en una expresión que permite capturar o no, dar valores iniciales y validar el contenido de cada celda:

[ <captura> ] < valor inicial >; < validación >

La expresión anterior contiene operandos (constantes o variables), operadores y funciones en formato de lenguaje "C" que se interpreta dinámicamente durante la ejecución. Los nombres de las celdas en sí son las variables que se utilizan en las expresiones; el contenido podrá variar por asignaciones o según el registro que se esté accedando. A continuación se muestran algunos ejemplos aclaratorios:

### Fórmula en atributo CALC

### Qué hace

[<e.captura>]

Solo captura

<e.asignación>

Solo asignación

---

<e.validación>	Solo validación
[<e.captura>]<e. asignación>	Captura y asignación
[<e. captura>];<e.validación>	Captura y validación
<e. asignación>;<e. validación>	Asignación y validación
[<e.captura>]<e.asignación> <e.validación>	Captura, asignación y validación

<e.captura>= Expresión de captura (captura o no captura) de datos.

<e. asignación>= Expresión de asignación (valor inicial).

<e. validación>= Expresión de validación (pasa o no pasa).

## El valor actual de la celda actual.

!! Constante nula.

<b>Ejemplo de fórmula en Celda</b>	<b>Significado</b>
[]	Captura dato
["Deme el valor inicial"]	Captura con prompt " "
Existencia + cantidad	Incrementa existencia con cantidad
# + cantidad	Incrementa existencia con cantidad

<code>cantidad + #</code>	Incrementa existencia con cantidad
<code>[] # + cantidad</code>	Sugiere existencia más cantidad pero permite la captura
<code>["Deme el valor inicial"] 0</code> <code>(Existencia==!) ? 0; #</code>	Sugiere 0, permite captura con "prompt" Si existencia es nula; por cero, si no, por el valor de la existencia
<code>+ Existencia</code>	Si existencia es nula, pon cero, si no, pon el valor de la existencia
<code>[ ] + Existencia</code>	Si existencia es nula, pon cero, si no, pon el valor de la existencia, permite captura
<code>[ # ==!"Existencia inicial":!]</code>	Si existencia es nula requiere captura, si ya hay algo no se permite la captura.

Los operadores que utiliza HOT son básicamente los que se utilizan en lenguaje "C", como son: operaciones booleanas, multiplicación, división, multiplicación, etc.

## El atributo MASK

Este atributo esta dado por una expresión, una combinación de operandos y operadores, con la cual se determina una condición para seleccionar un subconjunto de registros de la tabla principal que serán procesados por la jerarquía actual.

Hay tres formatos de expresiones MASK que son:

- \* Tipo Rangos. Se compara el valor de la celda actual que deberá estar entre dos rangos, su formato es e1; e2. Por ejemplo, para la celda

- 'PRECIO': '500:600'. Aquí, la celda PRECIO deberá estar entre 500 y 600.
- \* Tipo Igual. Se compara el valor de la celda actual y deberá ser igual al valor de la expresión, su formato es e3. Por ejemplo, para la celda 'SEXO': "M". Aquí, la celda SEXO deberá ser igual a M.
  - \* Tipo General. Es una expresión condicional general independiente del valor de la celda actual. Debe empezar con paréntesis, su formato es e4. Por ejemplo, En cualquier celda:  
'(PRECIO!=!!&&500<=price&&&PRECIO<=600)' o también,  
'(SEXO=="M)'.  
En estos casos no depende de la celda actual.

## El atributo RECORD

Este atributo muestra en pantalla de qué registro son las celdas.

## 3.4 MÓDULO DE EJECUCIÓN

Con HOT se puede desarrollar y probar una aplicación en forma interactiva. Toda la aplicación incluyendo la definición de las tablas y sus celdas, la estructura jerárquica de menús y tareas, las ventanas, los documentos, los registros, las fórmulas de las celdas, etc. se almacenan en un solo archivo del sistema operativo. Este archivo, que representa el objeto aplicación, se interpreta por el módulo de ejecución cuando se corre la aplicación y es susceptible de leerse y modificarse nuevamente por el módulo de desarrollo. (Ver Figura 9).

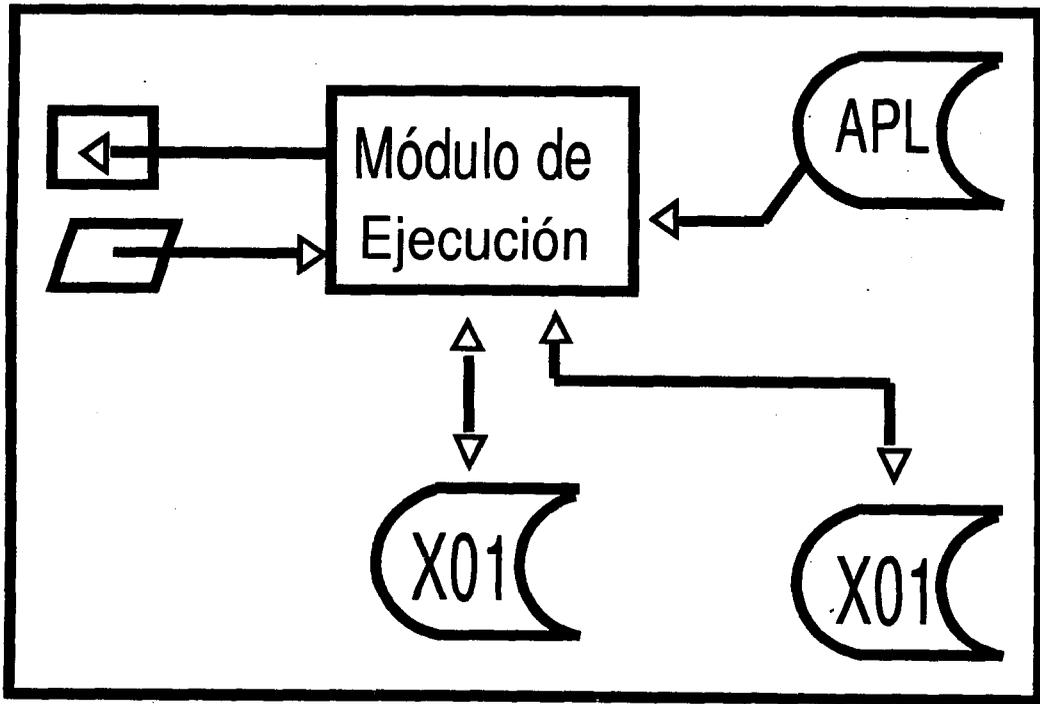


Figura 9. Ejecución de Aplicaciones en HOT Modo Uniusuario.

Las tablas o archivos indexados se consultan y/o actualizan durante la ejecución. HOT permite desarrollar rápidamente aplicaciones complejas. Dominando sus características desarrollador podrá resolver cualquier aplicación por compleja que parezca.

La estructura de menús y tareas podrá tener muchos niveles de profundidad o de detalle. Las tareas pueden tener una (lo normal) o varias ventanas, esto le permite gran dinamismo y muy buena presentación.

Cada ventana puede tener uno (lo normal) o varios documentos. Un documento puede llamar a otro en forma jerárquica en uno o varios niveles, durante el proceso se podrá tener acceso interactivo o podrá barrer todos los registros o solo los seleccionados. Cada documento está asociado a una tabla en forma primaria y podrá consultar y/o actualizar celdas locales o de otras tablas relacionadas (ligas).

Uno de los objetivos básicos de HOT es hacer sentir al usuario final que es muy fácil de usar.

La estructura de los menús y las tareas es simple y el manejo de los datos es sencillo y consistente. La edición de campos para corregir, la selección de opciones en los menús, la posibilidad de utilizar un archivo como entrada en lugar del teclado, la selección de donde y como se imprimen, despliegan o almacenan los reportes, etc. son siempre las mismas para todas las aplicaciones y paquetes.

Todo esto una vez que el usuario lo entiende le permite una operación muy simplificada y consistente, con facilidad de obtener ayudas dinámicamente.

HOT no sólo puede consultar y/o actualizar a una o varias tablas durante el proceso de una transacción, sino que aún cuando ya se hayan operado las modificaciones es posible que si el usuario se arrepiente pueda deshacerlas. **(Ver Figura 10).**

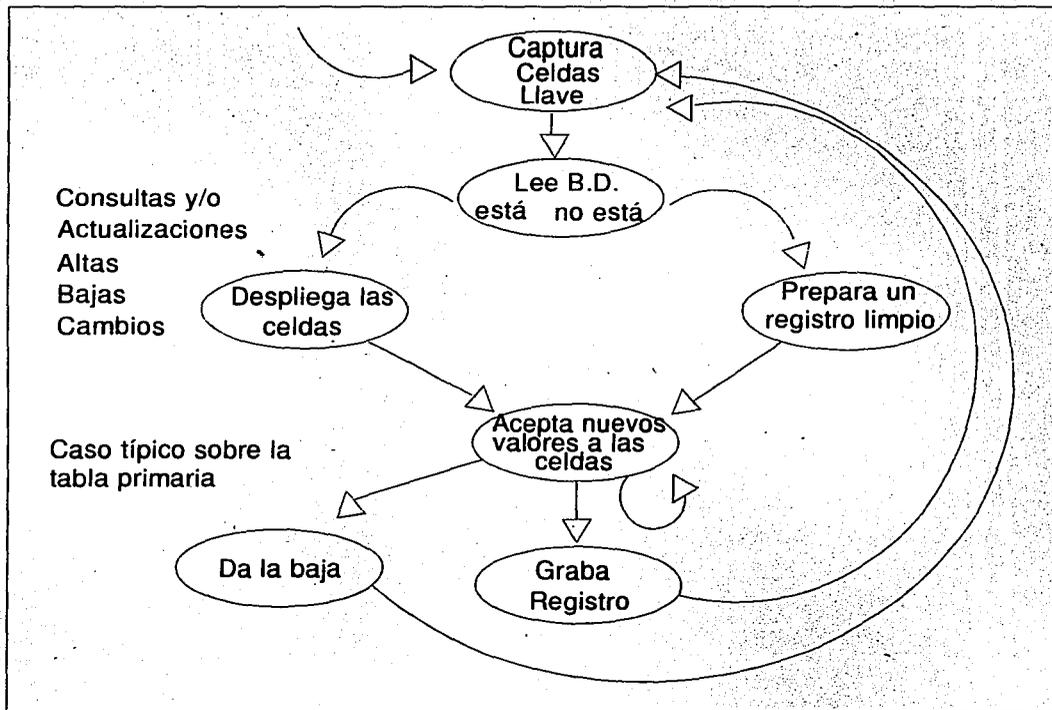


Figura 10. Ciclo Básico de HOT

Por ejemplo, en el caso de captura de pedidos por teléfono, el usuario pide los datos del cliente, la aplicación abre una ventana con los datos del cliente, sus datos generales, sus límites de crédito y posibles alarmas. El usuario abre un pedido anota las condiciones, la clave del vendedor, etc. Por cada renglón del pedido se accesa la tabla de artículos y se consulta y actualiza las existencias.

La idea es actualizar en línea la transacción hasta sus últimas consecuencias, por ejemplo actualizar los saldos del cliente, las comisiones del vendedor, las existencias de los artículos y generar todos los movimientos contables.

Si el cliente pidió 100 clavos y se arrepiente después de actualizar diciendo que no eran 100 sino 50 y no eran clavos sino tornillos, la aplicación podrá deshacer las modificaciones y hacerlas con las nuevas condiciones. A la existencia de clavos se le suman 100 para devolverlos y a la existencia de tornillos se le restan 50, si es que tiene para actualizarla. Con los cálculos de precio por cantidad se actualizan de la misma manera los importes.

Se podrá localizar cada pedido y una vez seleccionado se podrá localizar para consulta y/o modificación cada renglón del pedido utilizando los niveles de documentos.

Se requiere práctica para concebir la solución a cada problema, pero lo más importante es definir el problema, HOT ayuda a resolver los procesos de la forma más rápida y simple.

### **3.5 LIGAS**

Una liga es una relación que existe entre dos o mas tablas y se deben definir en las tareas o procesos. Para poder efectuar una liga es necesario que por lo menos exista un registro en cada tabla referida.

---

<b>Tipo de celda</b>	<b>Descripción</b>
De la Tabla Primaria	Corresponde al registro que esté en el momento de la evaluación, según su ciclo natural.
Locales	Estas celdas no se leen ni se graban de una tabla, permanecen en memoria durante la ejecución del documento, son muy útiles para guardar resultados parciales, para captura de datos antes de leer o grabar registros, para acumuladores, switches, etc.
De los registros de liga	Dependen del valor del registro que se leyó inmediatamente después que se evaluaron las llaves.
De otros documentos superiores en jerarquía	Se pueden consultar y/o actualizar celdas de documentos padres. Para referenciarlos se requiere saber el nombre del documento, si es de la tabla primaria, local o de alguna liga. El nombre está compuesto por el nombre de documento (si no se pone, supone el documento actual), un punto, el nombre del registro (supone la tabla primaria), el operador -> y el nombre de la celda.

---

---

## Nombres de las celdas de una LIGA

Los nombres de las celdas traídas en el registro de liga se conservan de la definición de la tabla original.

Para poder referenciarlas se requiere incluir el nombre del registro además del nombre de la celda, utilizando el operador ->. Esta regla se requiere para poder utilizar en un mismo documento más de una celda con el mismo nombre pero de diferente registro. El nombre del registro es el nombre de la tabla más un número consecutivo de veces que se ha hecho liga con la misma tabla. Esto permite hacer varias ligas a la misma tabla en el mismo documento. (Ver Figura 11).

Ejemplos:

<b>Nombre</b>	<b>Descripción</b>
DEPTO1->Código	Celda Código del registro DEPTO1 de la tabla DEPTO.
DEPTO2->Código	Celda Código del registro DEPTO2 de la -- tabla DEPTO.
ARTIC->Precio	Celda Precio del registro ARTIC1 de la tabla ARTIC.
Loc->Confirma	Celda Confirma tipo local.

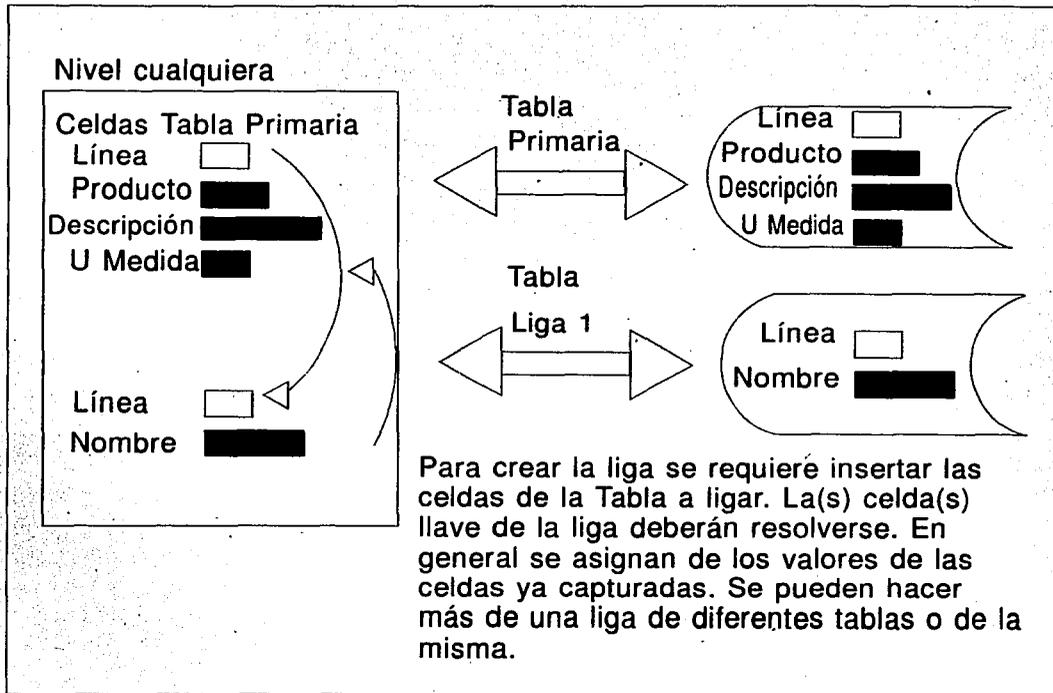


Figura 11. Ligas -Referencias a Otras Tablas-

## **CAPÍTULO 4**

### **APLICACIONES**

En este capítulo se efectúan dos aplicaciones con HOT, que retoman los elementos explicados a lo largo de este trabajo. Las aplicaciones y todas las pantallas son tomadas de la plataforma MS-DOS. El estilo de desarrollo no es una regla inflexible y es particular de la persona que utilice esta herramienta.

Los sistemas informáticos desarrollados son representativos ya que indican la manera de efectuar una aplicación con la herramienta HOT. El capítulo se presenta de la siguiente forma: en las dos aplicaciones se elaboran el Diagrama E/R y la Estructura Jerárquica de esa aplicación, con lo que se pretende dar claridad y una secuencia lógica para efectuar el desarrollo. En el primer sistema se presentan todas las pantallas y menús que intervienen en él, procurando no ahondar demasiado porque ya se explicaron los elementos necesarios para entender todo el proceso.

## **4.1 SISTEMA DE VENTAS**

### **APLICACIÓN:**

Sistema de Ventas

### **OBJETIVO GENERAL:**

El objetivo de esta aplicación es proveer un sistema que permita efectuar operaciones propias de un almacén o tienda.

### **OBJETIVO PARTICULAR:**

Explicar en forma gráfica y detallada el proceso que debe seguirse para la creación de un sistema con la Herramienta HOT. Se presentan las pantallas que intervienen en el desarrollo, haciendo mención de las particularidades de cada una de ellas.

### **CARACTERÍSTICAS:**

Permite la captura de altas, bajas, modificaciones y consultas de los artículos deseados para la Base de Datos. Se pueden obtener los siguientes reportes, ya sea en pantalla o enviarlo a algún archivo para posteriormente manejarlos con cualquier editor de textos.

- Reporte de Zonas: Indica todas las zonas registradas para operaciones en el negocio
- Reporte de Artículos: Presenta los artículos existentes.
- Reporte Artículos por Línea: Muestra los artículos que existen por cada línea considerada.
- Reporte de Líneas de Productos: Contempladas para clasificación de los

artículos por Línea.

- Reporte de Clientes: Indica los clientes registrados en el negocio.

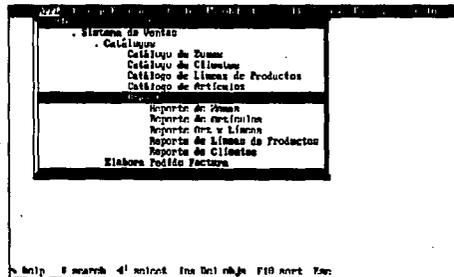
Permite efectuar "adquisiciones" o compras actualizando las existencias, etc.

Visualizar la factura obtenida para la compra o pedido de algún posible cliente.

El Diagrama E/R se muestra en la **Figura 12**. Se contemplarán seis entidades (Zonas, Clientes, Líneas, Artículo, FacEnc, FacDet) con sus correspondientes tablas; se incluyen en este diagrama tres de los atributos de HOT: Format, Name y Key.

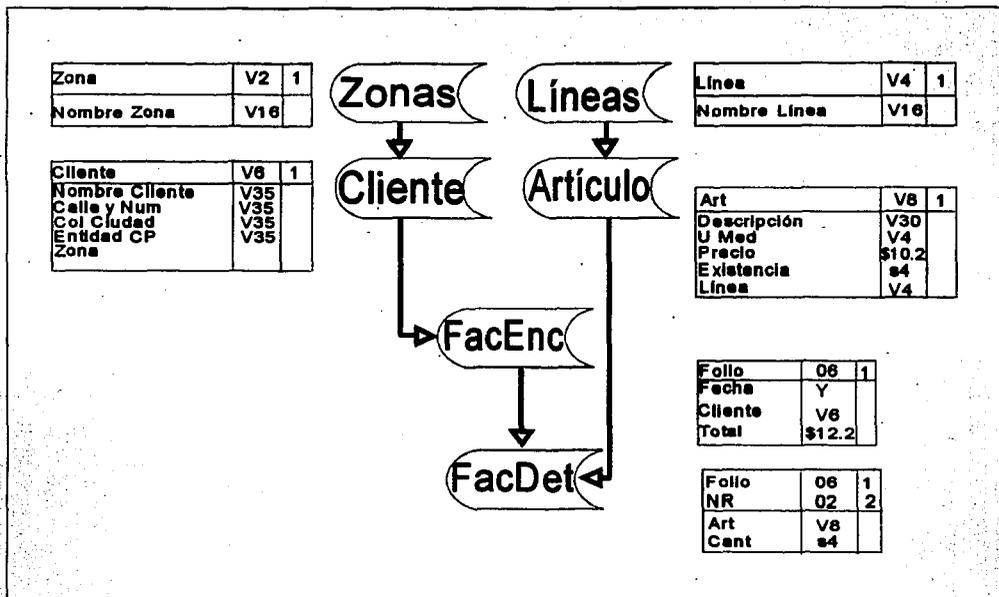
La estructura de la aplicación se muestra en la **Figura 13**, es importante porque se visualiza de inmediato como se desea que quede la aplicación una vez terminada.

Las pantallas que intervienen en la aplicación, son las siguientes:



Esta es la pantalla de la Aplicación e indica información general sobre ella:

\* El nombre de la Aplicación es Sistema de Ventas, que actúa como menú principal ( lo indica el punto que aparece a la izquierda del nombre).



**Figura 12. Diagrama E/R  
Con definición de tablas  
(Name, Frmt, Key)**

# Sistema de ventas

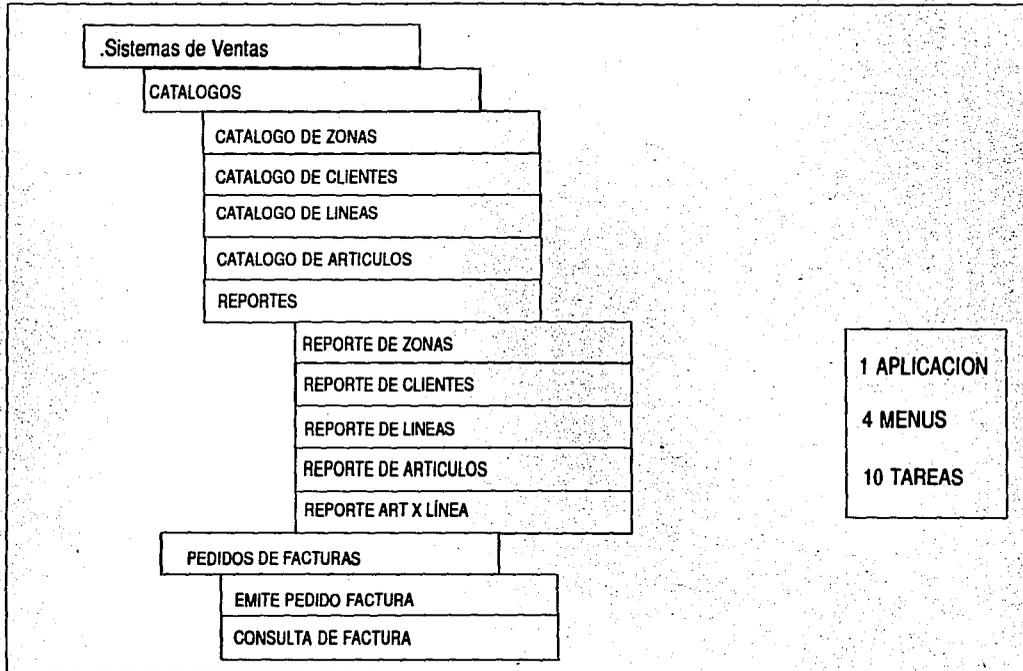


Figura 13. Estructura de la Aplicación

\* Posee un menú llamado Catálogos ( es un submenú de Sistema de Ventas) donde se ubican las siguientes tareas:

- Catálogo de Zonas.
- Catálogo de Clientes.
- Catálogo de Artículos.
- Catálogo de Líneas de Productos.

\* El siguiente menú, que es un submenú de catálogos, es el menú llamado Reportes al que lo integran varias tareas que son las que producirán las salidas de la Aplicación.

\* Se encuentra una tarea llamada Elabora Pedido Factura, que es el diseño y la elaboración de una factura para el Sistema de Ventas.

Las siguientes pantallas muestran las tablas que posee la Aplicación; estas tablas son las mismas que muestra la Figura 12, de ahí la importancia de elaborar de forma correcta el Diagrama E/R.

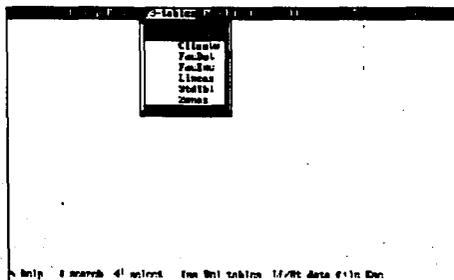


Tabla de Artículos

Ejemplo del uso de la herramienta BIT	
Sistema de Datos	
Catálogo de Artículos	
Artículos	
Descripción Artículos	
Id Artículos	
Precio Artículos	
Distancia Artículos	
Línea Artículos	

Nombre de campo: FID, nombre: Pg 2, atributo: nombre, nombre de campo: nombre, nombre de campo: nombre

Tabla de Cliente

Ejemplo del uso de la herramienta BIT	
Sistema de Datos	
Catálogo de Clientes	
Clientes	
Nombre del Cliente	
Calle y Número del Cliente	
Código Postal del Cliente	
Distancia del Cliente	
Sexo del Cliente	

Nombre de campo: FID, nombre: Pg 2, atributo: nombre, nombre de campo: nombre, nombre de campo: nombre

Tabla FacDet

FacDet:FacDet->Detalle FacDet	
Fecha Facto	Cant
NO Art	
Co FacDet	FacD

^ help \*8 nave F10 aux PgD atre [aux]l rcll \*4 drag sM% LK P4 FndM

Tabla FacEnc

FacEnc:FacEnc->Detalle FacEnc	
Ejemplo del uso de la herramienta [H] Sistema de Unidades	
FACTURA	Fecha Facto Fecha FacEnc
Cliente FacEnc	
	Total FacEnc

^ help \*8 nave F10 aux PgD atre [aux]l rcll \*4 drag sM% LK P4 FndM

Tabla de Líneas

Línea:Línea->>Línea Línea	
Ejemplo del uso de la herramienta [H] Sistema de Unidades	
Código de Línea de Producción	
Línea Lín	
Reserv Línea Línea	

^ help \*8 nave F10 aux PgD atre [aux]l rcll \*4 drag sM% LK P4 FndM

## Tabla de Zonas

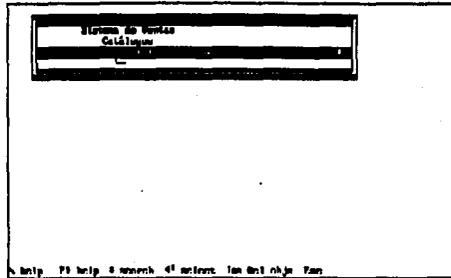
Sistema de Ventas - Nueva	
Zonas	
Ejemplo del uso de la herramienta de	
Sistema de Ventas	
Catálogo de Zonas	
-----	
Zona de	
Nombre: Zona Nueva	

^ help \*l rana F10 any Pjé attr lms/dl null 4l drag add% tab P4 r000

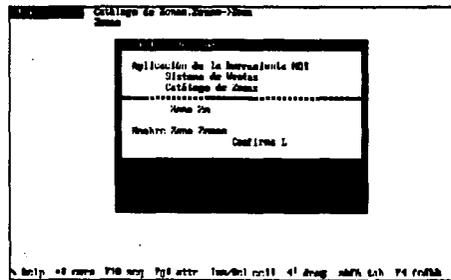
Esta es la estructura Parcial Extendida de una tarea, ( TASK ). Proporciona la siguiente información:

- \* Al sistema al que pertenece la tarea (Sistema de Ventas) y en qué menú se ubica.
- \* El nombre de la tarea (Catálogo de Zonas)
- \* La tabla primaria de la tarea (Zonas) y si es que el proceso se va o no a actualizar dicho archivo (ON).
- \* Indica si se trata de una tarea interactiva con el usuario (?)

Esta estructura es similar a todas las encontradas en esta aplicación.



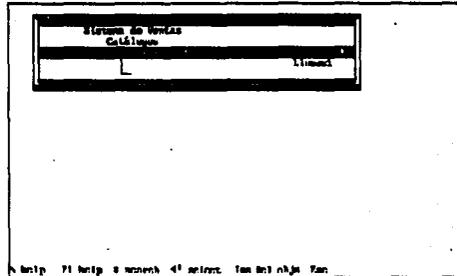
La siguiente figura muestra la tarea de Catálogo de Zonas, lista para ser ejecutada e introducir datos. Esta pantalla se "captura" estando en modo NAME y muestra lo que se ha capturado en este modo. Si se hubiera capturado en modo CALC, FORMAT o cualquiera de los siete modos hubiese presentado del lado izquierdo superior lo que se ha capturado en cada uno de esos modos.



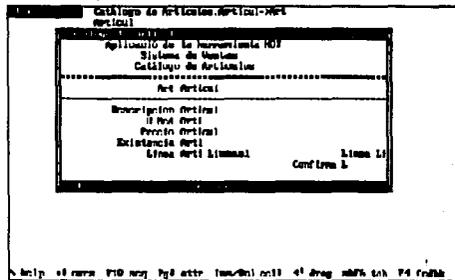
Estructura parcial extendida: Indica que se trata de una TAREA interactiva (?), que actualiza la tabla o archivos CLIENTE. La línea que aparece debajo de CLIENTE ON representa que existe una liga con la Tabla de Zonas y se trata de la primera liga a esta tabla, lo indica el número 1.





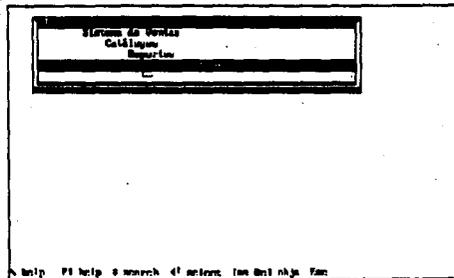


Tarea llamada Catálogo de Artículos. Ejecutando la Aplicación (F2), está lista para capturar información referente a los artículos que se manejan.

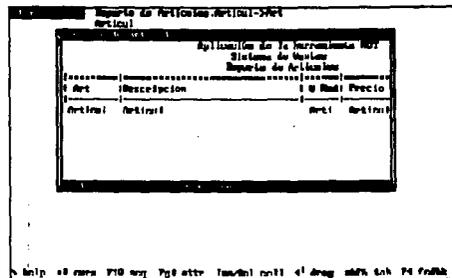


Estructura Parcial Extendida de la Tarea Reporte de Zonas. Actúa sobre la Tabla de Zonas. Es un Reporte, esto se sabe por la letra "r" que aparece a la izquierda de la Tarea. A la letra "r" acompaña un signo "+" que indica que se efectuará un barrido, es decir, accederá registro por registro hasta terminar con todo el archivo.

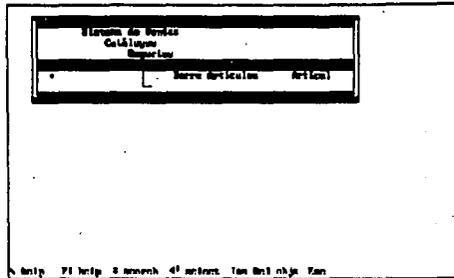




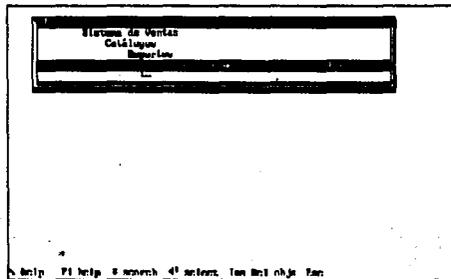
Tarea llamada Reporte de Artículos. El reporte, que se desplegará en pantalla puede abarcar más que la pantalla o la ventana en la que se muestra, visualizándolo completamente con las teclas de flecha derecha o izquierda (-> <-).



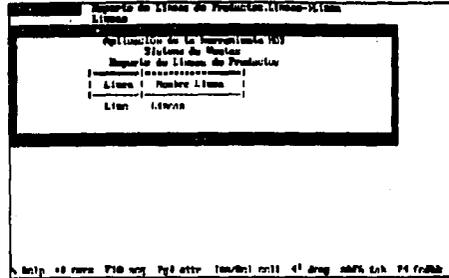
Estructura Parcial Extendida de la tarea Reporte Art. y Línea, es un reporte, y posee un nivel llamado Barre Artículos que actúa sobre Artículo. Con esto se le indica a HOT que lo que se desea es que se procesen en este nivel únicamente los registros correspondientes a las líneas que se imprimieron en el nivel superior. (El nivel Reporte Art y Líneas).



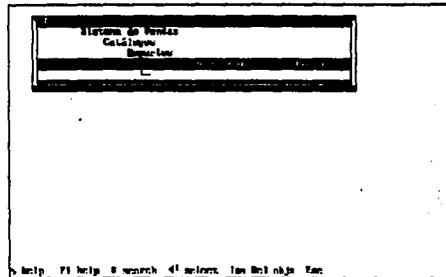
Estructura Parcial Extendida llamada Reporte de Líneas de Productos, que actúa sobre la tabla líneas. Es un Reporte.



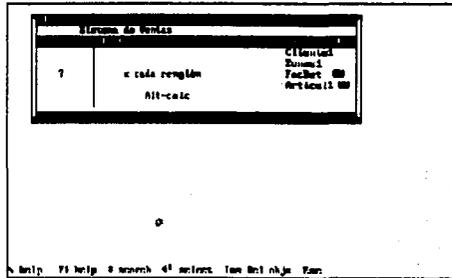
Tarea llamada Reporte de Líneas de Productos. Se imprimirán todas las líneas que se hayan capturado con su correspondiente nombre.



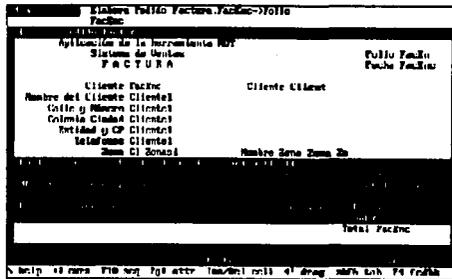
Estructura Parcial Extendida de la tarea Reporte de Clientes que actúa sobre la Tabla Cliente.



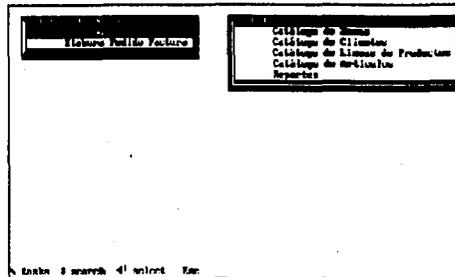
Estructura Parcial Extendida de la tarea Elabora Pedido Factura, que actúa y actualiza la Tabla FacEnc. Posee dos ligas, una con la Tabla Clientes y la otra con Zonas. Tiene un nivel llamado "x cada renglón" que actualiza la Tabla FacDet; esta posee una liga en la tabla Artícul. ALT-CALC es un nivel que indica una operación de cálculo que se efectuará en esta tarea.



Tarea llamada Elabora Pedido Factura. Las celdas se componen del nombre y de la especificación de la Tabla primaria a la que pertenece cada celda.



Al ejecutar la Aplicación se presenta el menú con todas las alternativas posibles.



De esta forma, se observa la manera de efectuar una aplicación con la Herramienta HOT. Como se ve, la programación visual aporta una valiosa ayuda a los desarrolladores, quienes se percatan de los acontecimientos tal y como están sucediendo, a diferencia de los lenguajes de programación tradicional, con los cuales se debe hacer el programa, ligarlo, correrlo y verificar que esté correcto y sea exactamente lo que se requiere.

Esta aplicación fue desarrollada y explicada a detalle, mostrando las pantallas más ilustrativas que intervienen en el desarrollo, con la finalidad de entender perfectamente todo el proceso paso a paso del cómo se efectúa una aplicación con HOT.

## **4.2 CONTROL DE EQUIPO DE COMPUTO**

### **APLICACIÓN:**

Control de Equipo de Cómputo.

### **OBJETIVO GENERAL:**

Llevar un registro preciso del equipo de cómputo que existe en alguna área de trabajo en particular, tales como PC s, impresoras, plotters, etc.

### **OBJETIVOS PARTICULARES:**

a) Saber con precisión el equipo de cómputo con el que cuenta un área y tener perfectamente ubicado a qué usuario está asignado dicho equipo. Tal ubicación se logra por el Proceso que existe de que cada Equipo tiene asignado un Responsable.

b) Percatarse de las características que posee el equipo registrado tanto de memoria, capacidad de disco, velocidad de impresión, y en general, de todas las particularidades del equipo disponible. Esto con la finalidad de tener un control adecuado y en caso de ser necesario nuevas adquisiciones, saber cuando, de qué tipo y a qué proveedor se le comprarán.

### **CARACTERÍSTICAS:**

Permite la captura de altas, bajas, modificaciones y consultas del equipo de cómputo considerado, como son PC s, impresoras, etc. El sistema cuenta con Reportes de Fallas, los cuales indican qué tipo de avería existe y el Responsable de ese equipo; puede ser importante tanto para el usuario como para el administrador que deseen conocer los motivos que están originando las fallas.

Se deben tomar decisiones de acuerdo a lo que arrojen los diferentes reportes ya sean medidas correctivas o preventivas. Si el causante es el equipo en sí, se podría hablar con el proveedor y explicarle que determinado lote de PC's o Impresoras están saliendo defectuosas; si se determina que es el Responsable del equipo el causante de las continuas fallas que se reportan, se puede capacitar al personal o llamar su atención, etc.

El Sistema de Control de Equipo de Cómputo permite verificar el software que tiene instalado las PC's registradas, con el objeto de verificar en su momento las licencias que existen en el Área de trabajo.

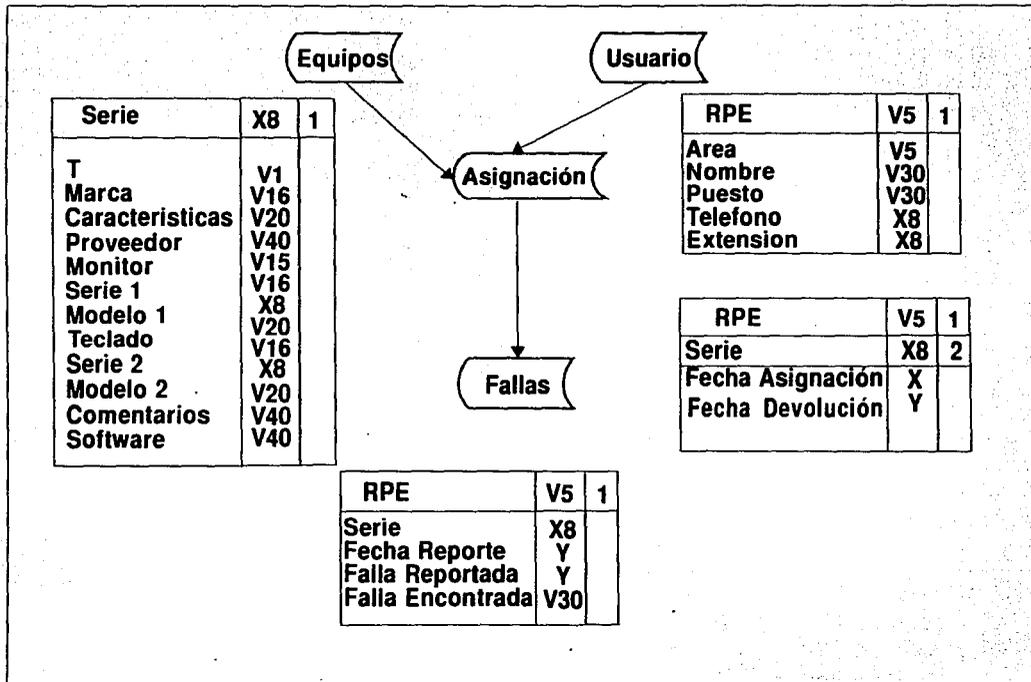
La aplicación posee alternativas de búsqueda para fallas específicas. Con esta opción se puede conocer, por ejemplo, En el mes de Agosto a cuantas PC's le ha fallado el disco duro (HD), a qué equipo le ha dado problemas el drive A, etc.

En resumen estas son las características generales del Sistema Control de Equipo de Cómputo. En la **Figura 14** se muestra el Diagrama (E/R) con sus respectivas tablas, incluyendo la definición del nombre, formato y llaves. La Estructura General de la Aplicación se encuentra en la **Figura 15**.

Se muestran las pantallas que corresponden al módulo de desarrollo explicando lo que significa y representa cada una de ellas.

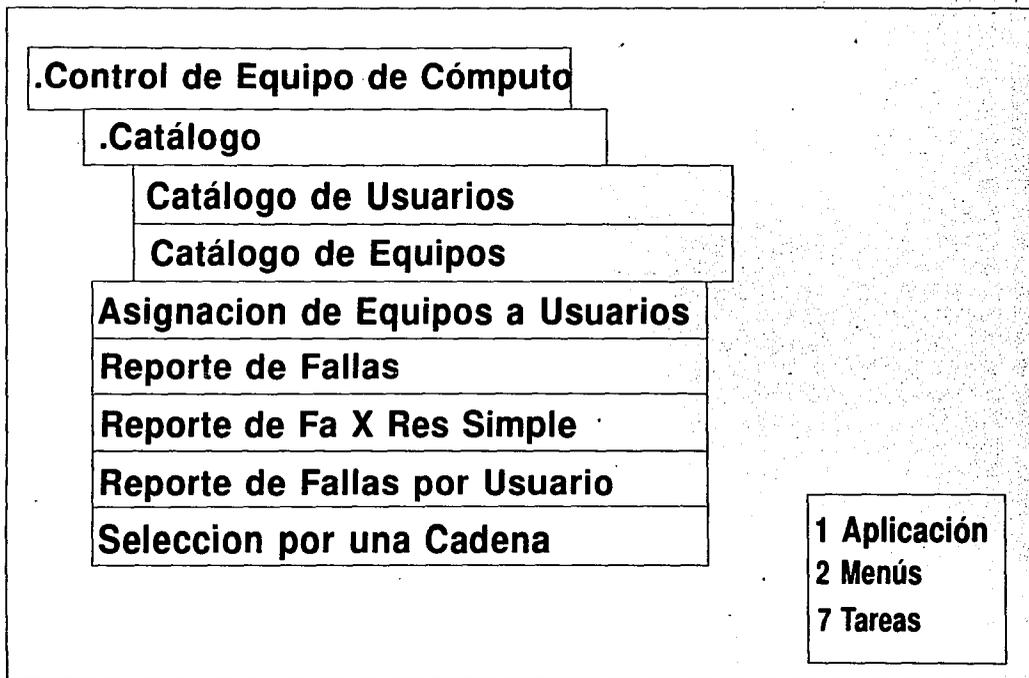
La primera pantalla de la aplicación proporciona la información siguiente:

- \* El nombre de la aplicación: Control de Equipo de Cómputo.
- \* La estructura de la aplicación con los Menús y Tareas que intervienen en ella.

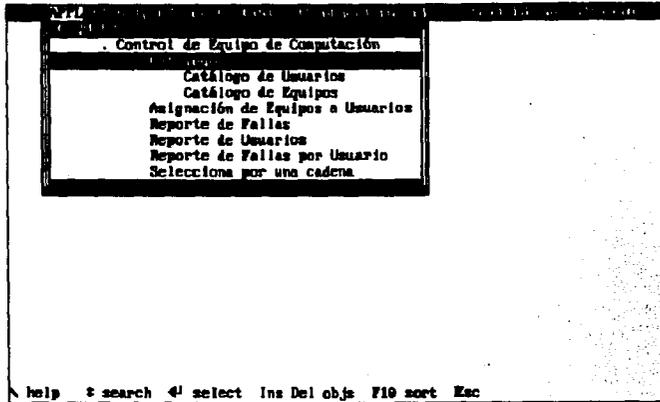


**Figura 14. Diagrama E/R.  
Con Definición de Tabla (Name, Frmt, key)**

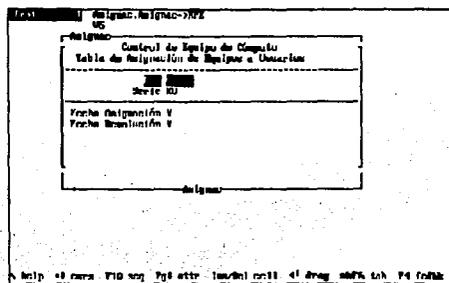
## Sistema de Control de Equipo de Cómputo

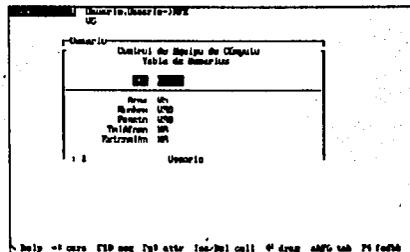
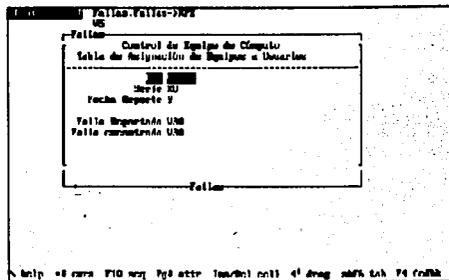
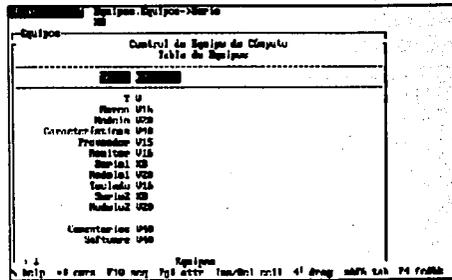


**Figura 15. Estructura de la Aplicación**



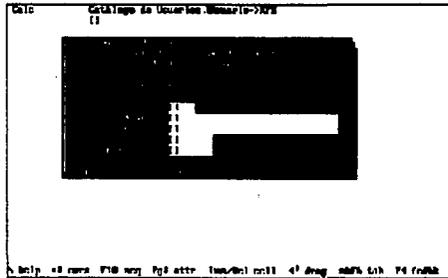
El sistema cuenta con cuatro tablas o archivos: Asignac, Equipos, Fallas y Usuario; las cuales se muestran en las siguientes pantallas. Los atributos de HOT necesarios para llevar el control del sistema (FORMAT, CALC, KEY, MASK, etc) ya se encuentran definidos.





La siguiente figuras muestran la Tareas que contiene la aplicación, del lado superior izquierdo se encuentra el modo en el que se ubica la pantalla. Notar que en todas se encuentran en modo CALC.

En el módulo de desarrollo, desde luego, intervinieron todos los elementos explicados a lo largo de la Tesis ( Inserción de menús, Tareas, Proceso, ligas, Indicación de Reportes, etc), presentándose la aplicación terminada y lista para ser ejecutada y cargar los datos de la Base mediante la Tecla F2.





por cada FALLE.Fallas-Reporte  
 (1) Ofrec ("C", "99-INSL") y Ofrec ("C", "99-INSL") : objeto ()  
 Reporte de Fallas----- por cada fallo  
 Control de Equipo de Computa  
 Reporte de Fallas

Apellido	Nombre	Confirm
Area	Puesto	
Apellido	Nombre	Confirm
Area	Puesto	
Fecha	Fecha de Insignific	Fecha de Insignific
	Notas	Confirm

\* Inip \* 8 cara \* 18 wq \* Pg 8 attr \* Ins/Del ctrl \* 41 drag \* 88% Sol \* P4 \* Falla

Reporte de Usuarios,Usuarios-2025  
 Reporte de Usuarios

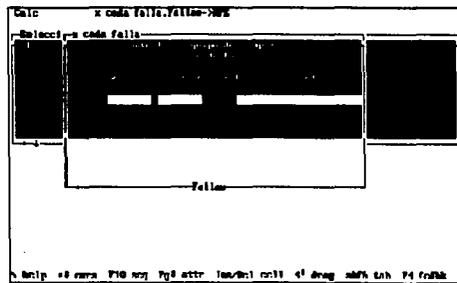
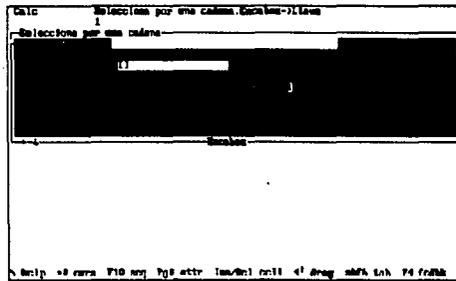
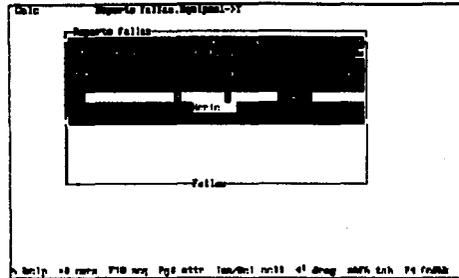
Apellido	Nombre	Confirm
Apellido	Nombre	Confirm

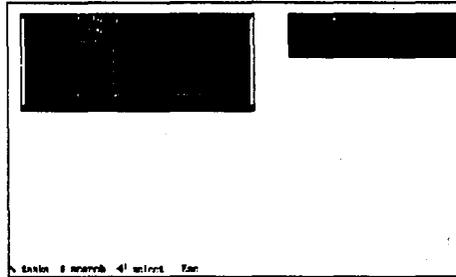
\* Inip \* 8 cara \* 18 wq \* Pg 8 attr \* Ins/Del ctrl \* 41 drag \* 88% Sol \* P4 \* Falla

Reporte de Fallas por Usuario,Usuarios-2025  
 Reporte de Fallas por Usuario

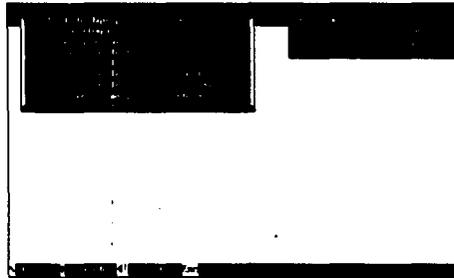
Apellido	Nombre	Confirm
Apellido	Nombre	Confirm

\* Inip \* 8 cara \* 18 wq \* Pg 8 attr \* Ins/Del ctrl \* 41 drag \* 88% Sol \* P4 \* Falla





Las pantallas que intervienen en el sistema una vez que se ejecuta son las siguientes. En cada una de las ellas la forma de entrar a alguna opción deseada, es únicamente posicionarse con el cursor y darle ENTER.



Catálogo de Usuarios  
 Control de Equipos de Computo  
 Tabla de Usuarios  
 -----  
 B/E: 0607  
 -----  
 Nombre: PFERREIRA CRISTIAN ROBERTO  
 Puesto: DIRECTOR GENERAL DE SISTEMAS  
 Teléfono:  
 Extensión: 5181  
 Usuario:

Catálogo de Usuarios  
 Control de Equipos de Computo  
 Tabla de Usuarios  
 -----  
 B/E: 0608  
 -----  
 Nombre: TORO GARCIA ROBY  
 Puesto: TECNICO PROGRAMADOR  
 Teléfono:  
 Extensión: 5181  
 Usuario:

Catálogo de Equipos  
 Control de Equipos de Computo  
 Tabla de Equipos  
 -----  
 Serie: 123  
 -----  
 Marca: NEC  
 Modelo: 486 PC  
 Características: 4 MBH HD 128  
 Procesador: 486333  
 Memoria: 128 MB ELCACION  
 Disco 1: 200 MB  
 Disco 2: 100 MB  
 Disco 3: 100 MB  
 Disco 4: 100 MB  
 Caracteres: 486 123  
 Software: WINDOWS 3.11  
 Equipo:

FALLA DE ORIGEN

Asignación de Equipos a Usuarios

Control de Equipo de Computo

Tabla de Asignación de Equipos a Usuarios

---

RFC 36477 FELIPE RAMON CABRILLO ESCOBAR  
 DIABLANTE ESPECIALIZADO ACD

T P    Modelo 486 386

Area 123

Fecha Asignación: [REDACTED]      Confirma

Asigna:

Reporte de Fallas

Reporte de Fallas

por cada falla

---

Usuario: RFE 36477 Ramon FELIPE RAMON CABRILLO ESCOBAR  
 Area 17000 Puerto DIABLANTE ESPECIALIZADO ACD      Confirma

---

Equipo: [REDACTED]

Modelo: [REDACTED]

Area: [REDACTED]      Modelo: 486/386      Confirma: 1

Fecha Reporte: [REDACTED]      Falla: [REDACTED]

---

35-07-11	FALLA DE HD	NO CARGA INFORMACION
35-07-12	FALLA LOS DISCOS FLEXIBLES	USINCA
35-07-23	NO SIRVE EL DRIVE A	NO SIRVE A NINGUNO
35-07-25	NO RECONOCE EL DISCO C	LE ENTREGAN EL COMPROMISO

35-07-25 [REDACTED]

Falla:

Los Reportes que nos envía la Aplicación de Control de Equipo de Cómputo se muestran en las siguientes páginas. Se pueden elaborar todos los reportes que se consideren necesarios utilizando los atributos de cualquier tabla que se haya definido.

Sistema de Control de Fallas de Equipos  
 Desde 95657 FERNANDO CARRILLO ESCALONA  
 Hasta A5852 RANFERI MOLINA

RPE 95657 FERNANDO CARRILLO ESCALONA

Area H7000

Serie	Fecha Reporte	Falla Reportada	Falla encontrada
123	95/07/24	FALLA EL DRIVE A	DRIVE A DAÑADO
123	95/07/25	NO INICIALIZA SISTEMA OPERATIV	LE BORRARON EL COMMAND.COM
123	95/07/26	NO RECONOCE DISCO DURO	DESCONFIGURADO EL SETUP
12345	95/07/11	FALLA DE HD	NO CABE INFORMACION
12345	95/07/12	FALLA LOS DISCOS FLEXIBLES	VIRUS
12345	95/07/23	NO SIRVE EL DRIVE A	DRIVE A DAÑADO
12345	95/07/25	NO RECONOCE EL DISCO C	LE BORRARON EL COMMAND.COM

RPE 9AJXD OMAR GARCIA KURY

Area HO710

RPE A5852 RANFERI MOLINA

Area H700

Serie	Fecha Reporte	Falla Reportada	Falla encontrada
1	95/01/01	FALLA FLOPPY	CAMBIO DE CONFIGURACION
1	95/02/01	SUCIA	LIMPIEZA
123456	95/07/11	NO SIRVE DRIVA A	LIMPIEZA
123456	95/07/12	HD NO ACEPTA	CAMBIO DE HD
2	95/01/01	NO IMPRIME PAREJO	CAMBIO DE TONNER
2	95/02/01	ATORO HOJAS	LIMPIEZA
2	95/03/01	NO RECIBE SEÑAL	CAMBIO DE CABLE

---

**Control de Equipo de Cómputo  
Reporte de Usuarios**

RPE	Area	Nombre	Puesto	Extensión	
95657	H7000	FERNANDO CARRILLO ESCALONA	DIBUJANTE ESPECIALIZADO ACAD	5181	
9AJXD	HO710	OMAR GARCIA KURY	TECNICO PROGRAMADOR	5181	
A5852	H700	RANFERI MOLINA	ING.	5181	
C6210	0H700	CHRISTIAN DOMINGUEZ NEGRON	TECNICO	5168	

Control de Equipo de Cómputo  
Tabla de Asignación de Equipos a Usuarios

Marca	Serie	Fecha Reporte	Falla Reportada	Falla encontrada
PIACER	123	95/07/24	FALLA EL DRIVE A	DRIVE A DAÑADO.
PIACER	123	95/07/25	NO INICIALIZA SISTEMA	OPERATIVO LE BORRARON EL COMMAND.COM
PIACER	123	95/07/26	NO RECONOCE DISCO DURO	DESCONFIGURADO EL SETUP
COMPAC	12345	95/07/11	FALLA DE HD	NO CABE INFORMACION
COMPAC	12345	95/07/12	FALLA LOS DISCOS FLEXIBLES	VIRUS
COMPAC	12345	95/07/23	NO SIRVE EL DRIVE A	DRIVE A DAÑADO
COMPAC	12345	95/07/25	NO RECONOCE EL DISCO C	LE BORRARON EL COMMAND.COM

Control de Equipo de Cómputo  
Tabla de Asignación de Equipos a Usuarios

Marca	Serie	Fecha Reporte	Falla Reportada	Falla encontrada
PIACER	123	95/07/24	FALLA EL DRIVE A	DRIVE A DAÑADO
PIACER	123	95/07/25	NO INICIALIZA SISTEMA OPERATIVO	LE BORRARON EL COMMAND.COM
PIACER	123	95/07/26	NO RECONOCE DISCO DURO	DESCONFIGURADO EL SETUP
COMPAC	12345	95/07/11	FALLA DE HD	NO CABE INFORMACION
COMPAC	12345	95/07/12	FALLA LOS DISCOS FLEXIBLES	VIRUS
COMPAC	12345	95/07/23	NO SIRVE EL DRIVE A	DRIVE A DAÑADO
COMPAC	12345	95/07/25	NO RECONOCE EL DISCO C	LE BORRARON EL COMMAND.COM

Control de Equipo de Cómputo  
Tabla de Asignación de Equipos a Usuarios

TI	Marca	Serie	Fecha Reporte	Falla Reportada	Falla encontrada
P	ACER	1123	95/07/24	FALLA EL DRIVE A	DRIVE A DAÑADO
P	ACER	1123	95/07/25	NO INICIALIZA SISTEMA OPERATIVO	LE BORRARON EL COMMAND.COM
P	ACER	1123	95/07/26	NO RECONOCE DISCO DURO	DESCONFIGURADO EL SETUP
COMPAC		12345	95/07/11	FALLA DE HD	NO CABE INFORMACION
COMPAC		12345	95/07/12	FALLA LOS DISCOS FLEXIBLES	VIRUS
COMPAC		12345	95/07/23	NO SIRVE EL DRIVE A	DRIVE A DAÑADO
COMPAC		12345	95/07/25	NO RECONOCE EL DISCO C	LE BORRARON EL COMMAND.COM

Control de Equipo de Cómputo  
Tabla de Asignación de Equipos a Usuarios

TI	Marca	Serie	Fecha Reporte	Falla Reportada	Falla encontrada
P	ACER	1123	95/07/24	FALLA EL DRIVE A	DRIVE A DAÑADO
P	ACER	1123	95/07/25	NO INICIALIZA SISTEMA OPERATIVO	LE BORRARON EL COMMAND.COM
P	ACER	1123	95/07/26	NO RECONOCE DISCO DURO	DESCONFIGURADO EL SETUP
COMPAC		12345	95/07/11	FALLA DE HD	NO CABE INFORMACION
COMPAC		12345	95/07/12	FALLA LOS DISCOS FLEXIBLES	VIRUS
COMPAC		12345	95/07/23	NO SIRVE EL DRIVE A	DRIVE A DAÑADO
COMPAC		12345	95/07/25	NO RECONOCE EL DISCO C	LE BORRARON EL COMMAND.COM

Control de Equipo de Cómputo  
Tabla de Asignación de Equipos a Usuarios

Marca	Serie	Fecha Reporte	Falla Reportada	Falla encontrada
PIACER	123	95/07/24	FALLA EL DRIVE A	DRIVE A DAÑADO
PIACER	123	95/07/25	NO INICIALIZA SISTEMA OPERATIVO	LE BORRARON EL COMMAND.COM
PIACER	123	95/07/26	NO RECONOCE DISCO DURO	DESCONFIGURADO EL SETUP
COMPAC	12345	95/07/11	FALLA DE HD	NO CABE INFORMACION
COMPAC	12345	95/07/12	FALLA LOS DISCOS FLEXIBLES	VIRUS
COMPAC	12345	95/07/23	NO SIRVE EL DRIVE A	DRIVE A DAÑADO
COMPAC	12345	95/07/25	NO RECONOCE EL DISCO C	LE BORRARON EL COMMAND.COM

---

**Control de Equipo de Cómputo  
Reporte de Fallas**

RPE	Serie	Fecha Reporte	Falla Reportada	Falla encontrada
03720	3	95/04/01	DISCO DURO NO LO RECONOCE	VIRUS REMOVIDO
95657	12345	95/07/12	FALLA LOS DISCOS FLEXIBLES	VIRUS
C6210	123	95/01/01	ESPACIO EN DISCO INSUFICIENTE	DETECCION DE VIRUS

**Control de Equipo de Cómputo  
Reporte de Fallas**

RPE	Serie	Fecha Reporte	Falla Reportada	Falla encontrada
95657	123	95/07/24	FALLA EL DRIVE A	DRIVE A DAÑADO
95657	12345	95/07/23	NO SIRVE EL DRIVE A	DRIVE A DAÑADO

## CONCLUSIONES

HOT es una buena herramienta para el desarrollo de sistemas informáticos y de utilizarse de manera eficiente, puede elevar considerablemente la productividad en cualquier organización; todo esto se logrará cuando se tenga práctica.

HOT no es una herramienta fácil de utilizar y de ninguna manera simple de aprender, esto se debe a que la "filosofía" o el enfoque proporcionado es diferente al que estamos acostumbrados cuando programamos en forma tradicional. Las teclas básicas que se utilizan son fáciles de confundir. La complejidad también radica en los mensajes de error que envía en el módulo de desarrollo, ya que son breves, no son muy explícitos y están en inglés.

Cuando se inicia el aprendizaje de HOT, en muchas ocasiones es desesperante y llega un momento en el que no se sabe qué hacer, se crea mucha confusión al no saber en qué pantalla estamos y qué fue lo que hicimos incorrecto. Esto también nos sucede a las personas que sabemos programar en cualquier otro lenguaje y no estamos exentos a las dificultades arriba expuestas. Paradójicamente, hay personas que sin saber mucho de computación, comprenden rápido la forma de comportarse de HOT, esto se debe a que no poseen distintos enfoques de programación que pudiesen producir confusión.

Una vez entendida la "filosofía" de HOT la herramienta va demostrando sus virtudes, tanto de rapidez en el desarrollo como de ejecución.

HOT es una herramienta muy productiva para desarrolladores que se vuelven expertos con ella. Pero es exclusiva para personas que saben diseñar e implementar sistemas; cuando alguien desea elaborar una aplicación con esta herramienta, conviene antes de iniciar la "programación", tener perfectamente definido y especificado la conceptualización completa para formar la Base de Datos. Se deben conocer qué tablas

---

o archivos se van utilizar, qué tipo de datos contendrán esos archivos, que relación existe entre cada uno de esas tablas, etc. Esto se logra realizando el diagrama E/R correspondiente.

Como se recalcó a largo de la Tesis, para poder visualizar la aplicación es conveniente hacer el Diagrama E/R (Entidad/Relación), definir la estructura general de la misma y saber qué menús y tareas se van insertar.

El desarrollador, entonces, deberá ser una persona experta - o personas con mucha intuición o habilidad en este tipo de actividades - , con experiencia en el desarrollo de sistemas, con conocimientos y gran sentido del costo-beneficio para reconocer una herramienta que le permita desarrollar adecuadamente lo que desea, ajustándose a los recursos que posee.

Considero que una de las claves para manejar adecuadamente la herramienta HOT es tener la idea muy clara de qué es lo que realmente está sucediendo al insertar menús, tareas, ligas, celdas locales, al cambiarle atributos a las celdas, etc. Uno de los aspectos más difíciles de entender y de utilizar es el proceso CALC; donde verificamos si en la Base de datos creada se va a capturar información, se va proporcionar un valor inicial, o se va validar. De hecho, esto puede convertirse en inconveniente ya que los cálculos complejos deben efectuarse a través del uso de fórmulas y funciones en las celdas, dificultando el mantenimiento de las aplicaciones.

Con HOT rápidamente se pueden hacer procesos medianos e inclusive complejos, definir una Base de Datos simple, así como proporcionar un programa que actualice esa tabla mediante altas, bajas, cambios y consultas. De forma sencilla se pueden hacer procesos en donde la tabla primaria sea una y mediante ligas se pidiesen acceder diferentes tablas. Se pueden elaborar sistemas desde muy sencillos hasta muy complejos en poco tiempo y utilizando programación visual. Esto último es importante porque con este tipo de programación nos percatamos de cómo va quedando toda la aplicación y si le añadimos que en cualquier momento la podamos ejecutar, nos proporcionan ventajas que pueden traducirse en ahorro de tiempo considerable, ya que vamos desarrollando

---

y probando; si algo no gusta simplemente regresamos al módulo de desarrollo y modificamos hasta que nos agrada o cumpla con los requerimientos establecidos, de tal forma que la aceptación por el usuario se va haciendo gradualmente.

Con HOT se pueden desarrollar aplicaciones tan grandes y complejas como se requieran, que incluyan diferentes niveles, ventanas, submenús, etc. que soporten cantidad considerable de información o registros. Actualmente, por mencionar una aplicación grande, se está desarrollando para un Municipio del Estado de Morelos un sistema que tendrá por función llevar el registro de pagos de los usuarios de agua potable.

Un aspecto importante, es el costo de la herramienta, comparándolo con el de cualquier software que exista en el mercado para el desarrollo de sistemas informáticos con características similares a HOT, resulta mucho más barato éste último. Esto puede ser un factor determinante al momento de elegir una herramienta que se accesible y eficiente para las necesidades de cualquier organización.

HOT proporciona ayuda en línea, esto significa que en cualquier lugar y momento del desarrollo que deseemos pedir ayuda, únicamente la solicitamos y nos la presenta de acuerdo al tema que estemos tratando en ese instante. Es conveniente y muy útil siempre ya que generalmente surgen pequeños tropiezos en cualquier lugar del desarrollo con HOT, ahorrándonos el estar constantemente consultando la ayuda o solicitándola a los expertos. Desafortunadamente toda la documentación y las ayudas están en inglés y es una lástima que una herramienta hecha en México tenga toda su documentación en este idioma, esto dificulta su uso, especialmente en usuarios no técnicos.

HOT es un producto cien por ciento mexicano y considero que es importante conocerlo y evaluarlo de manera objetiva. Si la herramienta convence, tratar de impulsar y apoyar estos proyectos que son de calidad y son el resultado del talento y trabajo de un grupo de mexicanos que se esfuerzan por realizar productos que puedan competir en el mercado.

---

## BIBLIOGRAFÍA

BORGE LANGEFORD.

"Teoría de los Sistemas de Información".

Edit. El Ateneo. Argentina 1976.

Págs. 305.

GEORGE M. SCOTT.

"Principios de Sistemas de Información".

Mc Graw-Hill. México 1988.

Págs. 646.

JOHN G. BURCH.

"Sistemas de Información: Teoría y Práctica".

Limusa. México 1986.

Págs. 320

HENRY F. KORTH, ABRAHAM SILBERSCHATZ.

"Fundamentos de Bases de Datos".

1ra. Edición, México 1987.

Págs. 270

KORTH, H.F., SILBERSCHATZ, A.

"Database Systems Concepts".

McGraw-Hill, Auckland, 1986.

Págs. 769.

---

**NOTAS TOMADAS DEL CURSO "HOT".**

Impartido por el ING. RAFAEL OLIVERA PUENTE.

Asesor de la Gerencia de Informática y Telecomunicaciones de Comisión Federal de Electricidad.

**PAREDANS, J.**

"Database".

Academic Press, London, 1987.

Págs. 456.

**PC WORLD.**

"Data base for you".

Volunen 12, No. 6 USA. Junio 1994

Pág. 11.

**PRESSMAN S. ROGER.**

"Ingeniería de Software. Un Enfoque Práctico".

Tercera Edición, México 1993.

Págs. 425

**SENN JAMES A.**

"Sistemas de Información para la Administración".

Grupo Editorial Iberoamérica. México 1991.

Págs. 728.

**TSCHRITZIN, D.C., LOCHOVSKY, F.H.**

"Data Base Management Systems".

Academic Press, New York, 1977.

Págs. 642.

---