17

2E

# UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE QUIMICA

## UN EJEMPLO DE APLICACION DE BASES DE DATOS RELACIONALES EN INGENIERIA QUIMICA: "BOMBAS, COMPRESORES Y VALVULAS"



# T E S I S

QUE PARA OBTENER EL TITULO DE:

## INGENIERO QUIMICO

P R E S E N T A:

FRANCISCO ARROYO GUITRON

MEXICO, D, F, 1995

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE QUIMICA

# UN EJEMPLO DE APLICACION DE BASES DE DATOS RELACIONALES EN INGENIERIA QUIMICA : "BOMBAS, COMPRESORES Y VALVULAS"

T E S I S

QUE PARA OBTENER EL TITULO DE :

INGENIERO QUIMICO

PRESENTA :

FRANCISCO ARROYO GUITRON

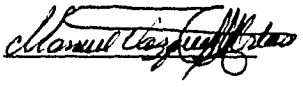
MEXICO, D.F.

**JURADO ASIGNADO SEGUN ELTEMA :**

| | |
|---|---|
| **Presidente** | **Prof : Caritino Moreno Padilla** |
| **Vocal** | **Prof : Manuel Vázquez Islas** |
| **Secretario** | **Prof : Ramiro Domínguez Danache** |
| **1er. Suplente** | **Prof : Gisella González Mariscal** |
| **2do. Suplente** | **Prof : Victor Manuel Vargas Chávez** |

El presente trabajo se desarrolló en la Facultad de Química de la UNAM.

**ASESOR DEL TEMA :**

**Manuel Vázquez Islas :** 

**SUSTENTANTE:**

**Francisco Arroyo Güitrón :** 

**A mi Mamá :**
La esperanza no es la convicción de que las cosas saldrán bien, sino la certidumbre de que algo tiene sentido, sin importar su resultado final.

**A Pepe, Benjamín y Lupita :**
No creo en el pesimismo. Si algo no sucede como deseamos, hay que seguir adelante sin doblegarse. Si uno piensa que va a llover, lloverá.

**A mi abuelita :**
Si desea el éxito, no lo busque; limítese a hacer lo que ama y en lo que cree. El éxito vendrá por añadidura.

**A Edith :**
Cuando creas que todo está perdido, recuerda que todavía nos queda el futuro.

**A mis amigos :**
El entusiasmo es un volcán en cuyo cráter no crece la hierba del titubeo.

# INDICE

# 1. INTRODUCCION.

Toda actividad intelectual se caracteriza por un conjunto de conceptos fundamentales y de técnicas específicas. Las técnicas son la manifestación de los conceptos en su aplicación a situaciones particulares. Los principios fundamentales permanecen iguales a través del tiempo, proporcionando las bases fundamentales para el desarrollo y la evaluación de técnicas.

Desde la década de 1970 se ha dirigido cada vez más atención a la tecnología del desarrollo del software. Se ha diversificado la utilización de las computadoras y se ha hecho más compleja y crítica para la sociedad moderna; como resultado de esto, el campo de la ingeniería de productos de programación ha evolucionado para convertirse en una disciplina tecnológica de considerable importancia.

Conforme las computadoras se multiplican y son más útiles para las sociedades, la demanda de productos de alta calidad se incrementa con celeridad.

Un sistema de programación de bases de datos debe desempeñar en su empleo diario, funciones específicas usando menos tiempo ó menos recursos humanos ó industriales que los que se requerían antes de tenerlo. Además, deberá ser eficiente, escrito con claridad y fácil de entender.

El factor más importante de la calidad de un producto es su utilidad, es decir, que el producto de programación satisfaga las necesidades del usuario.

La confiabilidad del producto está definida como la capacidad de un programa para desempeñar una función requerida bajo ciertas condiciones durante un tiempo específico. El grado de confiabilidad deseado en un producto particular puede ser expresado en términos del costo de la falla del producto, así, la cantidad de esfuerzo gastado en obtener confiabilidad debe ser función del costo de las imperfecciones del producto.

El período de vida común de un producto de programación es de uno a tres años en su desarrollo y de cinco a quince en su uso.

El desarrollo de un sistema de bases de datos consta de las siguientes fases:

- De análisis
- De diseño
- De instrumentación
- Prueba del sistema y
- Fase de mantenimiento

La primera fase consta de dos subfases: planeación y definición de requisitos.

Las actividades principales durante la planeación incluyen la comprensión del problema, estudio de factibilidad, desarrollo de la estrategia de solución recomendada, determinación de los criterios de aceptación y planeación del proceso de desarrollo.

En la segunda, el diseño estructural comprende la identificación de los componentes de la programación, su desacoplamiento y descomposición en módulos de procesamiento y estructuras de datos conceptuales, y la especificación de las interconexiones entre componentes.

La fase de instrumentación en el desarrollo del producto incluye la traducción de las especificaciones del diseño en código fuente, así como su depuración, documentación y pruebas. Las pruebas del sistema comprenden dos tipos de actividades: -pruebas de integración y de aceptación-. El desarrollo de una estrategia para integrar los componentes de un sistema de programación en una unidad funcional, requiere una planeación cuidadosa de modo que se disponga de los módulos cuando éstos se necesiten.

Las pruebas de aceptación se relacionan con la planeación y ejecución de varios tipos de pruebas para demostrar que el sistema de programación instrumentado satisface las necesidades establecidas en el documento de requisitos. Una vez aceptado por el usuario, el sistema de programación se

entrega para operación y se inicia la fase de mantenimiento del modelo de ciclo de vida por fases. Las actividades de mantenimiento incluyen mejoras de las capacidades, adaptación a nuevos ambientes de procesamiento y corrección de fallas del sistema.

Durante el tiempo de vida de desarrollo de un producto, se deben realizar varias actividades que comprenden planeación, desarrollo, servicios, publicaciones, control de calidad, apoyo y mantenimiento. La planeación identifica usuarios externos y necesidades, lleva a cabo estudios de factibilidad y supervisa el desarrollo de principio a fin. El desarrollo especifica, diseña, instrumenta, depura, prueba e integra el producto. Los servicios proporcionan herramientas automatizadas y recursos computacionales para todas las actividades y efectúa la administración de la configuración, distribución del producto y distintos apoyos administrativos. La publicación elabora los manuales del usuario, instrucciones de instalación, principios de operación y otros documentos de apoyo. El control de calidad se encarga de la evaluación del código fuente y de las publicaciones antes de su entrega. El apoyo promueve el producto, entrena a los usuarios, instala el producto y favorece la relación permanente entre los usuarios y las demás tareas. El mantenimiento corrige errores y hace mejoras ligeras durante la vida del producto.

El objetivo de esta tesis es proporcionar de una manera útil, didáctica y práctica todos los beneficios y apoyo que proporcionan en la actualidad los sistemas de bases de datos. Es decir, la tesis está dirigida específicamente a los alumnos y profesores involucrados en el estudio y aprendizaje de equipos de bombeo y de control de flujo, mediante un enfoque básicamente didáctico y pedagógico.

El trabajo ha sido dividido en tres partes: en la primera parte, presentada en el capítulo dos, se dan los aspectos generales y conceptos utilizados en la realización de un sistema de bases de datos, como lo son el hardware, software, datos y usuarios, entre otros, así como la importancia y evolución del concepto de bases de datos. En la segunda parte, se presentan los capítulos tercero y cuarto. En el tercer capítulo se expone la parte medular de un sistema de bases de datos, así como su estructura y metodología, a su vez se explica de manera detallada la forma como fue realizado el sistema, el lenguaje de programación

empleado, incluyéndose también el listado en el Anexo II de esta tesis. Y en el capítulo cuarto se presenta el sistema de bases de datos ya como una aplicación, proporcionándose toda la información acerca del uso de cada uno de los módulos que la conforman, explicándose detalladamente el uso de menús, botones, teclas rápidas y ayudas. Finalmente, en la tercera etapa comprendida por el capítulo cinco se exponen las conclusiones y beneficios que proporciona la aplicación al desarrollo tecnológico y científico, por el tipo de orientación y enfoque que se le da.

## 2. ANTECEDENTES

El procesamiento de información ha constituido una de las tareas básicas de cualquier civilización. Debido al crecimiento económico y demográfico, existen necesidades crecientes de administrar grandes cantidades de datos interrelacionados.

Los datos interrelacionados considerados en conjunto forman lo que se llama un sistema. La parte medular de cualquier sistema de información la constituyen sus datos almacenados.

### 2.1 Terminología Básica

Es importante mencionar algunos conceptos básicos importantes antes de definir detalladamente una base de datos:

- Una empresa, compañía, institución educativa, etc. son diferentes tipos de organización.
- Una entidad es una persona, lugar, cosa, evento ó concepto acerca del cual se registra información. Toda entidad tiene algunos atributos básicos que la caracterizan, al atributo frecuentemente se le llama elemento de datos ó campo de datos.
- El valor de los datos es la información ó los datos mismos contenidos en cada campo de datos. Los valores que toman los campos se conocen como datos. Al conjunto de valores tomados por los campos de datos de una entidad se le llama ocurrencia de la entidad.
- Un registro de datos es una colección de valores tomados por campos de datos relacionados.
- Los archivos de datos están formados por un conjunto de registros de datos.

Un sistema de bases de datos incluye cuatro componentes principales: datos, hardware, software y usuarios.

### 2.1.1 Hardware y Software

Existen dos componentes para un sistema de información computarizado: el *hardware* y el *software*.

- *Hardware* . Se le llama Sistema Físico (Hardware) al conjunto de mecanismos y dispositivos físicos que componen el sistema de procesos de datos: procesadores, placas de memorias, cables, conectores, etc. También se le define como un conjunto de circuitos y dispositivos periféricos que conforman en sí una computadora.

- *Software*. Se le llama Sistema Lógico (Software) o Sistema programado al conjunto de normas que permiten que el hardware realice la función para la cual lo queremos emplear, es decir, es el conjunto de instrucciones que le dicen a la computadora lo que tiene que hacer.

Mediante el hardware se realizan los cálculos y manipulaciones de datos necesarias para resolver el problema, pero es el software quien indica al hardware en que secuencia y bajo qué lógica hay que hacer los cálculos y las manipulaciones de datos. El hardware materializa la solución al problema pero bajo la dirección o inteligencia del software.

El hardware constituye la máquina real y gracias al software se convierte la máquina real en una máquina virtual que se puede usar y programar.

Entre los elementos más comunes que integran el hardware, por un lado y el software, por otro, están los siguientes:

**Hardware.**

- Computadora
    - Unidad Central de Proceso (UCP)
        - Unidad de Control
        - Unidad Aritmética y Lógica
    - Memoria Central
    - Unidades de Entrada y Salida

- Periféricos
    - Impresora
    - Discos magnéticos
    - Alambres
    - Conexiones
    - Ratón
    - Digitalizador

- **Software.**
    - Programas

Las funciones de la Unidad Central de Proceso consisten en leer y escribir contenidos de las celdas de memoria, llevar y traer datos entre celdas de memoria y registros especiales, y descodificar y ejecutar las instrucciones de un programa.

La unidad de control realiza las siguientes funciones importantes:

1. Generar señales de tiempo y control
2. Transferir dados de, o hacia la unidad de memoria o entrada/salida
3. Realizar operaciones aritméticas y lógicas
4. Reconocer señales externas

La Unidad Aritmética y Lógica se encarga de efectuar las operaciones relacionadas con los cálculos numéricos y simbólicos. Las operaciones que estas subunidades pueden efectuar son: suma y resta de dos números de punto fijo, multiplicación y división de punto fijo, manipulación de los bits de los registros y del acumulador (operaciones lógicas AND, OR, NOT), y comparación del contenido de dos registros (para averiguar si los números que contienen son iguales, o cuál es mayor).

La función de la Unidad de Memoria Central es retener información para que pueda ser tomada o accesada en cualquier momento. La memoria contiene:

1. El sistema operativo.
2. El programa de aplicación o conjunto de instrucciones que se requieren para llevar a cabo una determinada acción.
3. Los datos de entrada al programa.
4. Los datos de salida como resultado de la ejecución del programa.

Dentro de la amplia gama de memorias que existen en el mercado, hay tres grupos principales:

1. Memoria RAM (Random Access Memory)
2. Memoria ROM (Read Only Memory)
3. Memoria de núcleos magnéticos (CORE)

Las Unidades de Entrada y Salida típicas son:

1. Elementos de entrada: teclado, lectora óptica y lectora de tarjetas
2. Elementos de salida: pantalla CRT, impresora
3. Elèmentos almacenadores: cintas magnéticas, cartuchos, disco magnético (floopy disk).

Todos los cuales forman parte del Hardware.

Existe un tercer concepto, híbrido entre ambos (hardware y software) que ha sido bautizado como firmware y que cobra gran importancia en los nuevos ordenadores.

Una clase de firmware son las memorias de solo lectura (Read-Only-Memory o ROM). Las ROM son chips de memoria que contienen un cierto software grabado en forma irreversible para que no pueda ser destruido. La computadora lee el programa en la ROM y lo ejecuta pero no puede modificarlo, es firme, no borrable.

Uno de los beneficios que se persiguen es la seguridad (puesto que la memoria ROM no puede ser destruida inadvertidamente, el sistema operativo no puede ser alterado por errores del usuario). Otro de los beneficios es que se libera la memoria normal (RAM que significa Random Access Memory) para las

aplicaciones del usuario. Un tercer beneficio es la protección frente a copias no autorizadas.

Desde el punto de vista de su funcionalidad, el software se puede clasificar en dos grandes grupos o bloques:

- Programas de sistema (o software básico), que controlan y optimizan la operación de la máquina
- Programas de aplicación (o software de aplicación, que controlan y optimizan la operación de la máquina

Los programas del sistema hacen que el usuario pueda usar en forma cómoda y amigable complejos sistemas hardware. Actúan como un intermediario entre el usuario y el hardware.

Los programadores de aplicaciones, no necesitan conocer a fondo el modo de funcionamiento interno del hardware, basta con que conozcan las necesidades de información de sus aplicaciones y cómo usar el sistema operativo para conseguir satisfacer estas necesidades. Sus programas serán independientes del hardware específico que se utilice y podrán ser transportados sin grandes problemas de adaptación a otras máquinas y otros entornos operativos.

Los programas de sistema se dividen en dos grupos:

- Programas de control
- Programas de servicio

Los programas de control son los que van orientados a facilitar, automatizar y mejorar el rendimiento de los procesos en el ordenador (encadenamiento automático de programas, simultaneidad de operación de periféricos, tratamiento de errores, etc.) y, a su vez, podemos separarlos en tres grupos según sea el tipo de recurso que optimizan:

- Gestión del sistema
- Gestión de trabajos
- Gestión de datos

Los programas de servicio son los que van orientados a proporcionar facilidades de expresión y comunicación al usuario (traductores, cargadores, clasificadores, etc.)

La memoria principal de una computadora es relativamente pequeña, por lo que la mayoría de los datos se conservan en dispositivos de almacenamiento que se vinculan con la computadora por medio de canales. Estos dispositivos se denominan dispositivos de almacenamiento *periféricos ó secundarios.*

### 2.1.2 Datos

La descripción de los datos y la de las relaciones que entre ellos existen adopta una de dos formas: lógica o física. La descripción física de los datos se ocupa de como se los registra o almacena en el hardware. Y la descripción lógica, en cambio se refiere a la forma como los datos se presentan al programador de aplicaciones o a sus usuarios.

Una vez, que ya se mencionaron las formas como se almacenan los datos es importante mencionar una característica deseable, que es la Independencia entre ellos.

Se dice que una aplicación es dependiente de los datos si es imposible cambiar:

- Su estructura de almacenamiento, es decir, la manera como están físicamente registrados o almacenados los datos
- O la estrategia de acceso sin afectar la aplicación.

En sistemas de bases de datos, es deseable no permitir que las aplicaciones sean dependientes de los datos, al menos por las siguientes razones:

- Aplicaciones diferentes requerirán vistas diferentes de los mismos datos.
- La base de datos debe tener libertad de modificar la estructura de almacenamiento o la estrategia de acceso (o ambas) en respuesta al cambio de necesidades sin tener que alterar las aplicaciones existentes.

Otro punto importante a mencionar es la arquitectura de las bases de datos, la cual se divide en tres niveles generales:

- Interno
- Conceptual
- Externo

El nivel interno es el más cercano al almacenamiento físico, es decir, el que concierne a la manera como los datos se almacenan en realidad, el nivel externo es el más cercano a los usuarios, es decir, el que atañe a la manera como cada usuario ve los datos y el nivel conceptual puede considerarse como el que define una vista de la comunidad de usuarios.

### 2.1.3 Seudocódigo

La palabra Seudocódigo se refiere a la descripción verbal o escrita de la solución ya encontrada o sugerida para resolver un problema. Y esta se realiza a través del empleo de frases y palabras clave como:

- Comienza
- Si ....entonces....
- Otro
- Mientras ....
- Haz...
- Lee
- Escribe
- Repetir
- Termina

A continuación se muestra un ejemplo:

```
INICIAR tablas y contadores; ABRIR archivos
LEER el primer registro del texto
MIENTRAS haya más registros de texto REPETIR
        MIENTRAS haya más palabras en el registro de texto REPETIR
                EXTRAER la siguiente palabra
                BUSCAR en tabla-de-palabra la palabra extraída
                SI la palabra extraída se encontró ENTONCES
```

INCREMENTAR el contador de ocurrencias de la palabra extraída
SI NO
INSERTAR la palabra extraída en la tabla-de-palabra
FIN SI
INCREMENTAR el contador de palabras procesadas
FIN MIENTRAS al final del registro del texto
FIN MIENTRAS cuando todos los registros de texto han sido procesados
IMPRIMIR la tabla-de-palabras y el contador de palabras procesadas
CERRAR archivos
TERMINAR el programa

Las descripciones verbales, gracias a los modernos editores de textos pueden modificarse fácilmente, pero no ofrecen tanta claridad y lamentablemente con bastante frecuencia no son lo suficientemente precisas. Las ventajas en cuanto a precisión, claridad y capacidad de modificación de ambos medios de presentación, se reúnen en el seudocódigo.

Los seudocódigos no se rigen por ninguna norma, de modo que pueden diseñarse libremente. No obstante, en la aplicación práctica conviene respetar los siguientes elementos:

- Las palabras clave, que describen el proceso. Estas siempre deben consignarse en letras mayúsculas para que resalten.
- Lenguaje natural, con el que se formulan las acciones.
- Indentación del texto para poner de manifiesto la estructura.

El seudocódigo puede ser utilizado tanto en el diseño arquitectónico como en el detallado, al igual que los diagramas de flujo, este puede ser utilizado a cualquier nivel de abstracción, con el uso del seudocódigo, el diseñador describe las características del sistema usando frases cortas y concisas en español, las cuales se encuentran estructuradas por medio del uso de palabras clave como si-entonces-si no, mientras-repetir y fin.

A causa de su sencillez el seudocódigo se aprende con facilidad, de modo que resulta muy adecuado para la comunicación con el correspondiente departamento de especialistas o profesionales. Resulta siempre de gran ayuda cuando la solución obtenida debe ajustarse a las necesidades del usuario. La introducción de cambios o modificaciones no supone ningún problema. Si el seudocódigo se formaliza en exceso, puede generarse directamente una estructura de programa en el lenguaje de programación correspondiente, a través de un generador adecuado. Por lo que el programador no tiene más que convertir las instrucciones en lenguaje común a lenguaje de programación. De este modo se evitan muchas posibilidades de error que surgen a lo largo de la programación de la estructura del proceso.

### 2.1.4. Diagramas de Flujo

Los diagramas de flujo representan gráficamente las vías, por las que circulan las informaciones dentro de un sistema.

Son diagramas que se utilizan para describir un diseño de sistemas de alto nivel y muestran como se transforman los datos al pasar de un componente del sistema a otro.

Los diagramas de flujo estructurados representan la forma más tradicional para especificar y documentar los detalles algorítmicos de un producto de programación, estos diagramas utilizan cajas rectangulares para especificar las acciones, cajas en forma de rombos para las proposiciones de decisión, lineas para las interconexiones entre las diversas cajas, así como una variedad de formas especiales para denotar las entradas, salidas, almacenamientos, etc.

Ya que la estructura de los diagramas de flujo es equivalente al seudocódigo, se tiene el mismo poder de expresión, ambos, pueden ser usados para expresar cualquier algoritmo posible, sin embargo, los diagramas de flujo estructurados pueden ser preferidos en aquellas situaciones en donde la claridad del control deba ser recalcado.

Los diagramas de flujo estructurados subrayan el flujo de los mecanismos de control debido a la naturaleza gráfica de su representación visual. Por lo anterior, son apropiados cuando estos mecanismos de decisión y las secuencias de control deban ser atendidos.

Pero por otro lado, los diagramas tradicionales permiten que se viole fácilmente el principio de una entrada y una salida, además de no ser legibles o modificables por una máquina, como lo puede ser el seudocódigo.

### 2.1.5 Usuarios

Existen tres clases generales de usuarios de computadoras:

- El programador de aplicaciones, encargado de escribir programas de aplicación que utilicen bases de datos
- El usuario final que accesa la base de datos desde una terminal. Un usuario final es una persona que toma decisiones y que utiliza información obtenida al accesar la base de datos. También proporciona datos que serán almacenados en dicha base.
- El administrador de bases de datos, que es una persona cuya responsabilidad central es controlar los datos de operación, su trabajo requiere destreza técnica y capacidad de entender e interpretar los requerimientos administrativos a nivel gerencial.

## 2.2 Bases de Datos

El desarrollo de las bases de datos colectivas será sin duda una de las actividades más importantes en el campo de la informática en lo que resta del presente decenio. En todas las esferas de la vida y en todas las áreas de la industria y el comercio, las bases de datos ampliarán en gran medida las posibilidades de acción abiertas al hombre.

La base de datos puede definirse como una colección de datos interrelacionados, almacenados en conjunto sin redundancias perjudiciales ó innecesarias. Su finalidad es la de servir a una aplicación ó más, de la mejor manera posible, los datos se almacenan de modo que resulten independientes

de los programas que los usan, se emplean métodos bien determinados para incluir datos nuevos y para modificar ó extraer los datos almacenados. Se dice que un sistema comprende una colección de bases de datos cuando éstas son totalmente independientes desde el punto de vista estructural.

La expresión base de datos no se refiere a todos los tipos de registro, sino a una colección limitada y específica de estos. Dentro de un sistema, coexisten, por lo general, varias bases de datos, no obstante, se supone que los contenidos de estas bases son independientes. Las colecciones de bases de datos de esta clase se denominan *sistemas de bases de datos.*

La idea básica en la implantación de una base de datos es aquella en la que los mismos datos deben ser aprovechados para tantas aplicaciones como sea posible.

En una base de datos bien diseñada se evita la redundancia perjudicial o innecesaria. La redundancia no controlada acarrea varios inconvenientes. En primer lugar tenemos el costo adicional del almacenamiento de copias múltiples de los mismos datos. En segundo término, y esto es mucho más serio, el hecho de que para actualizar por lo menos una parte de las copias redundantes es preciso recurrir a múltiples operaciones de actualización.

La seguridad de los datos y la posibilidad de reconstruirlos en caso de falla son aspectos de la mayor importancia en el diseño de toda base de datos.

La capacidad para utilizar la base de datos sin conocer los detalles de representación se conoce como *independencia de datos.* Esta última es decisiva, ya que una empresa puede gastar grandes cantidades de tiempo y recursos en realizar cambios muy simples, como incrementar la longitud de un campo o agregar un campo nuevo. Un diseño ideal de la base de datos deberá permitir, hasta cierto grado, cambios de esta naturaleza que no afecten los programas de aplicación.

En suma, las razones que justifican la independencia de datos son las siguientes:

- Permitir al administrador de la base de datos hacer cambios de contenido, localización, representación y organización en la base de datos, sin necesidad de volver a escribir los programas de aplicación que utilizan la base de datos.
- Permitir al proveedor de equipo de procesamiento de datos y de software introducir nuevas tecnologías sin que se tenga que reprogramar la aplicación del cliente.
- Facilitar el compartimiento de datos al permitir que parezca que los mismos datos están organizados de manera diferente para los diversos programas de aplicación.
- Simplificar el desarrollo de programas de aplicación, y en particular, facilitar el desarrollo de programas para procesamiento interactivo con base de datos.
- Proporcionar la centralización de control que necesita el administrador de la base de datos para garantizar seguridad e integridad de la base de datos.

Una base de datos debe tener las siguientes características:

- Debe satisfacer las necesidades actuales de información.
- No solo debe satisfacer las necesidades actuales, sino también debe hacerlo en un tiempo razonable, es decir debe satisfacer los requerimientos de funcionamiento.
- Debe satisfacer los requerimientos previstos e imprevistos de los usuarios finales.
- Debe ser fácilmente expandible con la reorganización y expansión de la empresa.
- Debe ser fácil de modificar en los medios de software y hardware cambiantes.
- Una vez que la base tenga datos correctos almacenados, éstos deben permanecer así.
- Antes de insertar datos en la base se debe verificar su validez.
- Sólo personas autorizadas deben tener acceso a los datos almacenados en la base.

¿Cuál es la principal diferencia entre una base de datos y el archivo de datos?

Una base de datos puede tener más de un uso y los múltiples usos pueden satisfacer múltiples enfoques de los datos almacenados. Un archivo de datos puede tener más de un uso, pero solo puede satisfacer un enfoque de datos almacenados.

La base de datos permite no solo la lectura de los datos almacenados, sino la continua modificación de los que son necesarios para el control de las operaciones.

En las grandes operaciones de procesamiento que no cuentan con el apoyo de una base de datos, hay tantos datos redundantes que resulta prácticamente imposible mantenerlos todos en el mismo nivel de actualización.

La base de datos debe prestarse a una fácil reestructuración siempre que haya que agregarle nuevos tipos de datos ó utilizarla para nuevas aplicaciones. Además de que la seguridad de los datos y la posibilidad de reconstruirlos en caso de falla son aspectos de la mayor importancia en el diseño de toda base de datos.

Un sistema de bases de datos se utiliza porque proporciona a la entidad (organización comercial, científica, técnica u otra) un control centralizado de sus datos de operación que constituyen uno de sus activos más valiosos.

### 2.2.1 Evolución del concepto de base de datos

La expresión base de datos comenzó a popularizarse al principio de los años 60's. Antes de esa época, en el mundo de la informática se hablaba de archivos y de conjuntos de datos.

La expresión ha ido evolucionando a través del tiempo de la siguiente manera:

1) Años 60´s.

- Archivos organizados de modo secuencial simple.
- Estructura física de los datos esencialmente igual a la estructura de los archivos lógicos.
- Procesamiento en lotes sin acceso en tiempo real.
- Del mismo archivo existen varias copias porque se guardan las generaciones anteriores de datos.
- El software se ocupa solo de las operaciones de entrada-salida.
- El programador de aplicaciones diseña la distribución física de los datos y la incorpora a los programas de aplicación.
- Si se cambia la estructura de los datos ó los dispositivos de almacenamiento, los programas de aplicación deben volver a escribirse, compilar y probarse.
- Los datos se diseñan y optimizan, por lo general, para una única aplicación.
- Alto nivel de redundancia entre los archivos de datos.

2) Finales de los años 60´s

- Es posible el acceso secuencial ó el acceso directo a los registros.
- El procesamiento se hace por lotes.
- Se distingue la organización lógica de la organización física, pero las relaciones entre ellas son bastante sencillas.
- Pueden cambiarse las unidades de almacenamiento sin necesidad de modificar los programas de aplicación.
- Las estructuras de datos son por lo general de los tipos secuencia, secuencial indexados ó de acceso directo simplemente.
- Hay tendencia al diseño y optimización de los datos principalmente para una aplicación.

3) Años 70´s

- De los mismos datos físicos se derivan múltiples bases de datos lógicos.
- Se puede tener acceso a los mismos datos de diferentes maneras, según los requisitos de la aplicación.

- La organización de almacenamiento físico es independiente de los programas de aplicación. Puede cambiarse a menudo para mejorar el desempeño de la base de datos sin modificación de los programas de aplicación.
- Se utilizan formas de organización de datos muy complejas sin que ello se refleje en los programas de aplicación.

4) Requisitos actuales para los sistemas de bases de datos.

- El software procurará la independencia lógica y física de los datos, permitiendo que exista una vista lógica global.
- Los datos pueden evolucionar sin que se incurra en costos de mantenimiento excesivos.
- Se proveen medios para que un administrador de datos actúe como controlador y custodio de los datos y asegure que la organización de estos es siempre la mejor para los usuarios en general.
- Se proveen procedimientos eficaces para controlar la seguridad e integridad de los datos.
- En algunos sistemas se utilizan archivos invertidos para permitir una rápida exploración de la base.
- Las bases de datos se diseñan de modo que provean respuestas a tipos de averiguación no previstos por el diseñador.
- Se facilita la migración de datos.
- El software provee un lenguaje para la descripción de datos para el administrador de datos, un lenguaje de órdenes para el programador de aplicaciones.

### 2.2.2 Objetivos de una organización de bases de datos

Existen dos tipos de objetivos para la organización de bases de datos:

- Primarios.
- Secundarios.

Los objetivos primarios son:

- Los datos podrán utilizarse de múltiples maneras, es decir, diferentes usuarios, que perciben diferentemente los mismos datos, pueden emplearlos de distintas maneras.
- Se protegerá la inversión intelectual. No será necesario rehacer los programas y las estructuras lógicas existentes cuando se modifique la base de datos.
- Disminuir el costo de almacenamiento y el uso de los datos, así como minimizar del costo de los cambios.
- Menor proliferación de datos. Las necesidades de la nuevas aplicaciones se satisfacerán con los datos existentes mejor que creando nuevos archivos, evitándose así la excesiva proliferación de datos.
- Los pedidos de datos se atenderán con la rapidez adecuada según el uso que de ellos habrá de hacerse.
- Claridad para que los usuarios sepan qué datos se encuentran a su disposición y los comprenderán sin dificultad.
- Facilidad de uso, los usuarios tendrán fácil acceso a los datos, las complejidades internas son ajenas al usuario, gracias al sistema de administración de la base.
- Flexibilidad. Los datos podrán ser utilizados ó explorados de manera flexible, con diferentes caminos de acceso.
- El sistema evitará las versiones múltiples de los mismos ítems de datos con diferentes estados de actualización.
- Se evitará el acceso no autorizado a los datos. Los mismos datos podrán estar sujetos a diferentes restricciones de acceso para diferentes usuarios.
- Los datos estarán protegidos contra fallos, es decir, que si al realizar la ejecución del sistema a través de un comando aparece un mensaje de error, los datos no sufrirán ningún cambio o pérdida.
- Disponibilidad, los datos se hallarán inmediatamente disponibles para los usuarios casi todas las veces que los necesiten.

Los objetivos secundarios son:

- Independencia física de los datos. El hardware de almacenamiento y las técnicas físicas de almacenamiento podrán ser modificados sin obligar a la modificación de los programas de aplicación.
- Independencia lógica de los datos. Podrán agregarse nuevos items de datos, ó expandirse la estructura lógica general, sin que sea necesario reescribir los programas de aplicación existentes.
- Redundancia controlada
- Adecuada rapidez de acceso
- Adecuada rapidez de exploración
- Normalización de los datos dentro de un organismo

## 3. DESARROLLO DEL SISTEMA DE BASES DE DATOS

La programación es un arte que depende de la habilidad individual, la atención al detalle y el conocimiento de cómo utilizar los instrumentos disponibles de la mejor manera.

Historicamente, los incrementos más significativos en la productividad de un proceso de manufactura o construcción se han conseguido cuando la destreza humana está apoyada por instrumentos automatizados.

Una de las tareas principales es definir bien el problema, a continuación dar una lista de actividades para resolver el problema y después es seleccionar el lenguaje de programación que se va a utilizar en la aplicación del sistema. La elección de un lenguaje de programación apropiado reduce al mínimo las dificultades de codificar un diseño, reduce la cantidad de pruebas de programas necesarias y hace al programa más legible y, por tanto, más fácil de mantener.

El desarrollo de un *sistema de bases de datos* sugiere el uso de varias bases de datos, que contienen datos interrelacionados y cuya finalidad es la de servir a la aplicación, de la mejor manera posible. Se dice que un sistema comprende una colección de bases de datos cuando éstas son totalmente independientes desde el punto de vista estructural.

La fase de análisis del desarrollo de productos de programación implica la planeación del proyecto y la definición de los requisitos.

Una especificación de requisitos para la producción de software debe tener un conjunto de propiedades deseables. En particular, un documento de requisitos debe ser:

- Correcto
- Completo
- Consistente
- No ambiguo
- Funcional
- Verificable
- Rastreable
- Fácilmente modificable

Un conjunto de requisitos incompleto ó incorrecto puede producir un producto de programación que satisfaga sus requisitos, pero no los del cliente. Una especificación inconsistente establece requisitos contradictorios en diferentes partes del documento, pero un requisito ambiguo se presta a distintas interpretaciones de diferentes personas.

Los requisitos de programación deben ser de naturaleza funcional, es decir, deben describir lo que se necesita sin implicar cómo cumplirá el sistema con estos requisitos.

## 3.1 Conceptos Fundamentales del Diseño de Software

En el diseño de software, existen tres tipos distintos de actividades:

- el diseño externo
- diseño arquitectónico
- diseño detallado

El diseño externo de software requiere de concebir, planear y especificar sus características de un producto de programación. Estas características incluyen la definición de despliegues en pantalla y los formatos de los reportes, la definición de las entradas y salidas de datos, así como las características funcionales, los requerimientos de desempeño y la estructura general del producto.

El diseño interno incluye la concepción, la planeación y la especificación de la estructura interna y de los detalles de proceso del producto de programación. Las metas del diseño interno son: las de especificar la estructura interna, los detalles de procesamiento, de guardar las decisiones tomadas en el diseño e indicar el porqué ciertas alternativas y acuerdos fueron aceptados, elaborar los planes de pruebas y proporcionar una guía para su instrumentación, para las pruebas y las actividades de mantenimiento.

El diseño arquitectónico se preocupa del refinamiento de la vista conceptual del sistema, identificando funciones internas del proceso, así como la definición de cadenas de datos locales y su almacenamiento. Además del establecimiento de las relaciones e interconexiones entre las funciones, los datos y el almacenamiento de los mismos.

Un sistema de software bien diseñado es fácil de aplicar y mantener, además de ser comprensible y confiable. Los sistemas mal diseñados, aunque puedan funcionar, pueden ser caros de mantener, difíciles de probar y poco confiables. La etapa de diseño, es, por lo tanto, la parte más importante del proceso de desarrollo del software.

Los conceptos fundamentales en el diseño de la programación incluyen la abstracción, estructura, guardado de información, modularidad, concurrencia, verificación y los aspectos estéticos en el diseño.

### 3.1.1 Abstracción

La abstracción es la herramienta intelectual que permite trabajar con los conceptos independientemente de las instancias particulares de estos; durante la definición de los requerimientos y el diseño, la abstracción permite la separación de los aspectos conceptuales de un sistema de los que será más tarde instrumentado.

Durante el diseño de la programación, la abstracción permite organizar y dirigir los procesos de pensamiento al posponer las consideraciones estructurales y algorítmicas hasta que las características funcionales, las cadenas de datos y los almacenamientos hayan quedado definidos.

Existen tres mecanismos para la abstracción en el diseño de los productos de programación, los cuales son:

- La abstracción funcional
- Abstracción en los datos
- Abstracción en el control

Estos mecanismos permiten controlar la complejidad del proceso de diseño por medio de proceder sistemáticamente de lo abstracto a lo concreto.

La abstracción funcional incluye al uso de subprogramas parametrizados. La habilidad de parametrizar un subprograma y enlazar diferentes valores de los parámetros para diferentes llamados del subprograma es un mecanismo poderoso de abstracción, este tipo de abstracción puede ser generalizada para una colección de subprogramas, denominados "grupos".

La abstracción en los datos incluye la especificación de los tipos de datos o de los objetos por medio de especificar operaciones permitidas sobre los objetos, aquí, los detalles de la representación y el manejo son eliminados.

Los tipos de datos abstractos lo son en el sentido de que los detalles de la representación del dato y la forma de instrumentar a las funciones que los manipulan, quedan escondidos dentro del grupo que instrumenta dicho tipo de dato.

La abstracción del control es la tercera forma común del uso de la abstracción en el diseño de productos de programación. Esta es utilizada para establecer un efecto deseado sin necesidad de definir el mecanismo exacto del control.

### 3.1.2 Estructura

La estructura es una característica fundamental de los productos de programación. El uso de una estructuración permite que un sistema grande sea definido en términos de unidades más pequeñas y manejables con una clara definición de las relaciones entre las diferentes partes del sistema.

### 3.1.3 Modularidad

Un módulo de programación puede considerarse como una entidad definida que tiene las siguientes características:

- Contienen instrucciones, lógica de proceso y estructuras de datos.
- Pueden ser compilados aparte y almacenados en una biblioteca.
- Pueden quedar incluidos dentro de un programa
- Los módulos pueden usar a otros módulos.

La modularización permite al diseñador descomponer un sistema en sus unidades funcionales con el fin de imponer un ordenamiento jerárquico en el uso de las funciones, igualmente permite la instrumentación de abstracciones de datos y el desarrollo independiente de subsistemas útiles.

Los sistemas modulares consisten en unidades claramente definidas y manejables con las intercaras claramente definidas entre los diversos módulos. Las propiedades deseadas de un sistema modular cumplen con los siguientes criterios:

- Cada abstracción de un proceso es un subsistema claramente definido y con el potencial de ser útil para otras aplicaciones.
- Cada función en cada abstracción tiene un propósito específico, claramente definido.
- Cada función maneja no más de una estructura de datos principal del sistema.
- Las funciones comparten datos globales en forma selectiva, es ciertamente fácil de identificar todas las rutinas que comparten una estructura de datos principal.

La modularidad mejora la claridad del diseño, que a su vez facilita la instrumentación, la depuración, las pruebas, la documentación y el mantenimiento de un producto de programación.

La modularización permite al diseñador descomponer un sistema en sus unidades funcionales con el fin de imponer un ordenamiento jerárquico en el uso de las funciones.

### 3.1.4 Concurrencia

Los sistemas de programación pueden ser categorizados como secuenciales ó concurrentes. En un sistema secuencial solo una porción del sistema se encuentra activa en un momento dado; los sistemas concurrentes tienen procesos independientes que pueden ser activados en forma simultánea, si existen procesadores múltiples. En un procesador sencillo, los procesos concurrentes pueden quedar entrelazados en el momento de ejecución, esto permite la instrumentación de sistemas multiprogramados en tiempo real ó tiempo compartido.

Existen problemas específicos para sistemas concurrentes como la condición de bloqueo o deadlock, la exclusión mutua y la sincronización de los procesos y la condición de bloqueo es una condición indeseable que ocurre cuando todos los procesadores de un sistema se quedan esperando a que otros procesadores complementen la acción de sus procesos respectivos para poder continuar.

La concurrencia es un principio fundamental en diseño de la programación, ya que el paralelismo en los productos de programación introduce mayor complejidad y grados de libertad adicionales en el proceso del diseño.

### 3.1.5 Verificación

La verificación es un concepto fundamental en el diseño de la programación, ya que el diseño es el punto entre los requerimientos del cliente y la instrumentación que satisface esos requerimientos. Un diseño es verificable si puede demostrarse que el diseño generará el producto que satisface los requerimientos del cliente. Esto se desarrolla comúnmente en dos pasos:

1. Verificación que la definición de los requisitos de programación satisface las necesidades del usuario (verificación de los requerimientos)

2. Verificación que el diseño satisface la definición de los requisitos (verificación del diseño).

Los planes de pruebas son un producto del proceso del diseño, el plan de pruebas debe estar acoplado a las especificaciones demostrables del sistema.

La confirmación de un sistema de software en un proceso continuo en cada etapa del ciclo de vida del software.

La prueba de los programas es la parte del proceso de confirmación que suele realizarse durante la aplicación y también, en una forma distinta, cuando ésta ha terminado. La prueba consiste en ejercitar el programa utilizando datos similares a los datos reales que habrán de ser ejecutados por el programa,

observar los resultados y deducir la existencia de errores o insuficiencias del programa a partir de las anomalías de ese resultado.

Depuración es el proceso de localizar dónde se produjeron esos errores y corregir el código incorrecto.

Es muy importante comprender que la prueba nunca demuestra que un programa es correcto. Siempre es posible que existan errores aun después de la prueba más completa. La prueba de programas solo puede demostrar la presencia de errores en un programa, no su ausencia.

### 3.1.6 Estética

Las consideraciones estéticas son fundamentales para el diseño; la simplicidad, elegancia y claridad de un propósito distinguen a los productos de alta calidad de los mediocres.

La elegancia matemática o la belleza estructural, se refiere a aquellas propiedades que van más allá de la satisfacción de los requerimientos.

Es difícil listar objetivamente los criterios para evaluar los factores estéticos en un producto de programación, sin embargo un producto estéticamente agradable es fácilmente reconocido.

### 3.1.7 Acoplamiento y cohesión

Una meta fundamental en el diseño de los productos de la programación es la de estructurar al producto de tal forma que el número y la complejidad de las interacciones entre los diversos módulos sea minimizada.

La fuerza del acoplamiento entre dos módulos está influida por la complejidad de la interfaz, por el tipo de conexión y por el tipo de comunicación existentes.

La modificación de un bloque común de datos o de control puede requerir de modificaciones en todas las rutinas que se encuentren acopladas a ese bloque, por otro lado, si los módulos se comunican solamente por los parámetros y si las intercaras entre módulos permanecen constantes, los detalles internos de los módulos pueden ser modificados sin tener que modificar las rutinas que usan los módulos modificados.

La comunicación entre módulos incluye el pasaje de datos, de elementos de control, así como de las modificaciones del código de un módulo hacia otro. El grado de acoplamiento es menor para la comunicación de datos, mayor para la de conceptos de control y mucho mayor en el caso de módulos que modifican el código de otros módulos.

El acoplamiento entre módulos puede ser considerado de la siguiente forma:

- Acoplamiento del contenido
- Acoplamiento de zonas compartidas
- Acoplamiento del control
- Acoplamiento por zonas de datos
- Acoplamiento de datos

El acoplamiento del contenido ocurre cuando un módulo modifica los valores locales o las instrucciones de algún otro módulo.

En el acoplamiento de zonas compartidas, los módulos son enlazados en forma conjunta por medio de zonas globales para las estructuras de datos.

El acoplamiento del control se realiza entre módulos de tal forma que un módulo controla la secuencia de proceso de otro.

El acoplamiento por zonas de datos es similar al de zonas compartidas, excepto que los elementos globales son compartidos en forma selectiva entre las diversas rutinas que requieren de los datos.

Y por último, el acoplamiento de datos incluye el uso de listas de parámetros para pasar a los elementos entre rutinas.

La cohesión interna de un módulo se mide en términos de la fuerza de unión de los elementos dentro del módulo. Esta cohesión ocurre en el siguiente orden:

- Cohesión coincidental
- Cohesión lógica
- Cohesión temporal
- Cohesión en la comunicación
- Cohesión secuencial
- Cohesión funcional
- Cohesión informacional

La cohesión coincidental ocurre cuando los elementos dentro de un módulo no tienen relación aparente entre cada uno de ellos.

La cohesión lógica implica algunas relaciones entre los elementos de un módulo , como por ejemplo, en uno que desempeñe todas las funciones de entrada y salida, o en otro dedicado a la edición general de los datos.

Los módulos con cohesión temporal presentan muchas de las desventajas de los lógicamente unidos.

Los elementos de un módulo que contiene cohesión en la comunicación se refieren al mismo conjunto de datos de entrada o salida, por ejemplo la instrucción imprima y perfore el archivo de salida, presenta una cohesión en la comunicación.

La cohesión secuencial de los elementos ocurre cuando la salida de un elemento es la entrada para el siguiente.

La cohesión funcional representa un tipo fuerte, y por ende deseable, de amarre de los elementos de un módulo debido a que todos los elementos se encuentran relacionados al desempeño de una sola función.

La cohesión informacional de elementos en un módulo ocurre cuando el módulo contiene una estructura de datos compleja, así como varias rutinas que manejan dicha estructura. Este tipo de cohesión es similar a la cohesión en la comunicación en tanto que ambas se refieren a una sola entidad de datos.

### 3.1.8 Instrumentación

La fase de instrumentación del desarrollo de la programación tiene que ver con la traducción de las especificaciones de diseño a código fuente. El objetivo principal de la instrumentación es el escribir código fuente y la documentación interna de modo que la concordancia del código con sus especificaciones sea fácil de verificar y que se faciliten la depuración, pruebas y modificaciones.

Sencillez, claridad y elegancia son los sellos de los buenos programas; oscuridad, ingeniosidad y complejidad son indicaciones de un diseño inadecuado y un pensamiento mal orientado.

La claridad del código fuente se mejora mediante técnicas de codificación estructurada, buen estilo de codificación, documentos adecuados de apoyo, buenos comentarios internos y por las características que proporcionan los lenguajes de programación modernos.

Un subprograma recursivo es uno que se llama a sí mismo, ya sea directamente o bien indirectamente; la recursividad es una técnica de programación poderosa y elegante. Cuando se le usa apropiadamente, los subprogramas recursivos son fáciles de entender. De esta manera mediante un uso apropiado de la recursividad se cumplen los objetivos de claridad y eficiencia.

El estilo es la ruta consistente de elecciones hechas entre caminos alternos de lograr un efecto deseado.

En programación de computadoras, el estilo de codificación se manifiesta en las rutas que usa el programador para expresar una acción ó un resultado deseado.

Un buen estilo de codificación puede superar muchas de las deficiencias de un lenguaje de programación primitivo, mientras que un estilo pobre puede frustrar los propósitos de un excelente lenguaje. El objetivo de un buen estilo de codificación es proporcionar un código fácil de comprender y sencillo.

La programación por computadora incluye el código fuente de un sistema y todos los documentos de apoyo generados durante el análisis, diseño, instrumentación, pruebas y mantenimiento del sistema. La documentación interna incluye prólogos estándar para unidades de compilación y subprogramas, los aspectos autodocumentados del código fuente, y los comentarios internos incrustados en el código fuente.

### 3.2 Factores que influyen en la calidad y la productividad

El desarrollo y mantenimiento de programas son tareas muy complejas.

El grado de formalidad y la cantidad de tiempo asignada varía de acuerdo con el tamaño y complejidad del producto que se desarrollará. Existe una gran diferencia entre escribir un pequeño programa para uso propio y el desarrollo y mantenimiento de un producto de programación.

El tamaño de un proyecto es un factor importante que determina el nivel de control administrativo y el tipo de herramientas y técnicas necesarias en un proyecto de programación.

Uno de los factores que generalmente se interpretan como señal de calidad, es la facilidad de mantenimiento que ofrece un sistema. El mantenimiento del software consiste en la posibilidad de eliminar errores, el afinamiento, así como en la posibilidad de la ampliación, para incorporar nuevas funciones.

Otro criterio de calidad, es la forma en que el sistema realiza la interacción con el usuario. Uno de los factores decisivos es el funcionamiento correcto del software, es decir que todas las funciones se ejecuten correctamente, y que el sistema reconozca las situaciones ilícitas.

Es también muy importante la fiabilidad. Aunque no existe ningún sistema totalmente inmunizado contra el bloqueo total del sistema, dichas situaciones deberían limitarse a representar únicamente excepciones.

También es interesante la eficiencia del sistema. Con esto generalmente se hace referencia a la velocidad de ejecución de las funciones. Sin embargo, la

eficiencia también comprende el aprovechamiento óptimo de los recursos disponibles, tales como la memoria, los periféricos, etc. Estos dos criterios de eficiencia, sin embargo casi siempre resultan contradictorios. Los programas rápidos requieren una mayor cantidad de memoria, mientras que los programas que gestionan óptimamente la memoria, son frecuentemente más lentos.

Otro de los factores decisivos para el usuario, es el enlace hombre-máquina, es decir, aquellas porciones del sistema, que ve realmente el usuario. Aquí son diversos los factores, tales como la confortabilidad y la claridad, que desempeñan un papel importante. También la seguridad de un sistema protección contra la pérdida de datos y contra accesos no permitidos por parte de terceros.

La calidad del producto y la productividad del programador puede elevarse al mejorar los procesos necesarios para el desarrollo y mantenimiento de los productos. Algunos factores que influyen sobre la calidad y productividad se muestran a continuación:

- Capacidad Individual
- Comunicación del grupo
- Complejidad del producto
- Notación adecuada
- Enfoque sistemático
- Cambio de control
- Nivel tecnológico
- Confiabilidad requerida
- Tiempo disponible
- Entendimiento del problema
- Estabilidad de los requerimientos
- Habilidades necesarias
- Facilidades y recursos
- Entrenamiento adecuado
- Habilidades administrativas
- Metas adecuadas
- Expectativas crecientes

*Capacidad Individual.* La producción y mantenimiento de productos de programación son tareas laboriosas, por lo que la productividad y la calidad son funciones directas de la capacidad y esfuerzo individuales. Existen dos aspectos en la capacidad: la competencia global del individuo y su familiaridad con el área particular de aplicación. La falta de familiaridad con el área de aplicación puede implicar baja productividad y poca calidad.

*Comunicación en el grupo.* Se considera que la programación en una actividad individual y privada. Con el incremento del tamaño de un producto, disminuye la productividad debido al aumento en complejidad de las interacciones entre los diversos componentes del sistema.

*Complejidad de un producto.* Existen tres niveles o tipos de complejidad en un producto generalmente aceptados:

- Programas de aplicación
- Programas de apoyo
- Programas del sistema operativo

Los programas de aplicación tienen el más alto nivel de productividad mientras que los programas de sistema operativo tienen el menor, medido en término del número de líneas de código por día de programador.

El esfuerzo requerido para desarrollar y mantener un producto de programación es una función lineal del tamaño del producto y su complejidad.

*Notaciones apropiadas.* Una buena notación puede aclarar las relaciones e interacciones de importancia, mientras que las notaciones deficientes complican e interfieren con la buena práctica de la disciplina. Las notaciones apropiadas son vehículos de comunicación entre el personal asignado al proyecto y plantean la posibilidad de usar herramienta automatizada de programación para manejar las notaciones verificando su uso correcto. Esto puede beneficiar un proyecto en particular, pero se obtendrán más beneficios cuando se adopte, en todas las organizaciones e industrias, un conjunto pequeño de notaciones bien definidas para los proyectos de programación.

*Enfoques sistemáticos.* En cada campo del conocimiento existen ciertos procedimientos y técnicas aceptadas, la existencia de estas prácticas normales es una de las características que distinguen esa disciplina profesional.

*Control de cambios.* La flexibilidad en un programa es un gran beneficio y a su vez una gran fuente de dificultad. Las notaciones y procedimientos que permitan seguir la secuencia y determinar el impacto de un posible cambio son necesarios para hacer visible el costo verdadero de una modificación aparentemente pequeña al código fuente. Por otro lado, el uso de notaciones y técnicas apropiadas hace que se puedan realizar cambios controlados, sin menoscabo de la calidad del producto.

*Nivel tecnológico.* El nivel tecnológico utilizado en un proyecto de programación incluye aspectos como selección del lenguaje, ambiente computacional, prácticas de programación y herramientas de programación disponibles. Los lenguajes de programación modernos proveen características mejoradas para la

definición y manejo de datos, estructuras de construcción mejoradas para la definición del flujo de control, mejores facilidades de modularización, manejo eficiente de condiciones y facilidades para la programación concurrente.

El ambiente computacional se refiere al conjunto de características del equipo y los programas disponibles para el desarrollo, uso y mantenimiento del producto. La estabilidad y disponibilidad del ambiente computacional influyen notablemente en la productividad y la calidad del producto.

Las técnicas modernas de programación comprenden el uso de un análisis sistemático y técnicas de diseño, nomenclatura apropiada, codificación estructurada, técnicas de depuración y estudio de documentos y código fuente y pruebas sistemáticas.

Las herramientas de programación van desde las herramientas más elementales como ensambladores y depuradores sencillos hasta ambientes totales de programación que incorporan herramientas para la administración y el control del desarrollo del proceso.

*Nivel de confiabilidad.* Todo producto de programación debe poseer un nivel elemental de confiabilidad, sin embargo, la alta confiabilidad solo se consigue con gran cuidado en el análisis, diseño, instrumentación, pruebas y mantenimiento del producto de programación. Se requiere tanto recursos humanos como equipo para obtener un aumento en la confiabilidad, lo anterior conduce a una reducción en la productividad, medida solo en términos de líneas de código producido durante un mes.

*Capacitación del problema.* En un proyecto de programación un asunto común de difícil solución es la incomprensión de la verdadera naturaleza del problema. En ocasiones la naturaleza misma del problema no se revela hasta que el proyecto, o la mayor parte de este, se ha terminado y se halla operando, otras veces, la solución automatizada cambia la naturaleza del problema, y el cambio no es visto con claridad sino hasta que el sistema ha quedado instalado.

*Tiempo disponible.* Algunos investigadores creen que al extender un proyecto más allá de su duración nominal se incrementa el esfuerzo requerido para este, es evidente, sin embargo, que la dificultad de un proyecto y, por lo mismo, la productividad del programador y la calidad de los programas, son funciones sensibles al tiempo disponible para su desarrollo y modificación.

*Especialización requerida.* El ejercicio de la ingeniería de programación requiere de una gran gama de habilidades y especialidades. La definición de las necesidades y las actividades de diseño son de tipo conceptual y requieren habilidad en resolución de problemas.

*Facilidades y recursos.* Los factores relacionados con el trabajo, como buen acceso a la máquina y un lugar silencioso para laborar, resultan ser más importantes que los factores relacionados con la clase (tratos especiales).

*Entrenamiento adecuado.* Las habilidades que suelen faltar en programadores novatos, son la incapacidad de:

- Expresarse claramente en su lengua materna
- Desarrollar y validar requisitos de software y diseñar las especificaciones
- Trabajar dentro de áreas específicas de aplicación
- Desempeñar labores de mantenimiento de programas
- Efectuar análisis económicos
- Laborar con técnicas de administración de proyectos
- Trabajar en grupo

*Habilidades gerenciales.* Los proyectos de programación son, por lo común, supervisados por gerentes que tienen poco conocimiento, acerca de la ingeniería de programación.

*Metas apropiadas.* La meta principal de la ingeniería de software es el desarrollo de productos de programación que cumplan con los requisitos del uso deseado; idealmente, todo producto de programación debe proporcionar niveles óptimos de generalidad, eficiencia y confiabilidad. El esfuerzo desorganizado dedicado a mejorar marginalmente algunas características deseadas con una excesiva eficiencia, van en contra de la productividad del programador. La poca confiabilidad y eficiencia perjudican la calidad del producto. Se puede obtener un punto medio entre la productividad y los factores de calidad, mediante el mantenimiento dentro de las metas y requisitos establecidos para el producto durante la etapa de planeación.

*Expectativas crecientes.* El problema de mayor persistencia en la ingeniería de software es del crecimiento constante de las expectativas del producto. Existen dos aspectos interrelacionados al respecto:

- Está la preocupación de que tanta funcionalidad, confiabilidad y desempeño puede obtenerse con un esfuerzo determinado
- Se halla el aspecto relacionado con las limitantes de la tecnología de programación

Existe un progreso constante en el desarrollo de herramientas y técnicas para mejorar la calidad y productividad de un programador. Sin embargo, la diversidad, el tamaño y la complejidad de las aplicaciones crecen con más rapidez que nuestra capacidad de manejar tan creciente demanda.

Existen todavía muchos otros factores que influyen en la productividad de los programadores, incluyendo la familiaridad, el acceso y la estabilidad del sistema de cómputo utilizado para desarrollar o modificar los programas, la memoria y las limitantes de tiempo del producto, la experiencia con el lenguaje de programación, además, el tamaño de la base de datos.

### 3.3 Planeación del Proyecto de Programación

El primer componente del manejo de la ingeniería de software es la planeación efectiva del desarrollo del software. El primer paso en la planeación del proyecto es definir y documentar las suposiciones, objetivos y coacciones del proyecto.

El objetivo de la etapa de planeación del proyecto es identificar todos los requerimientos extremos para completar el proyecto. Otra decisión crítica es determinar los recursos requeridos para el proyecto.

La administración efectiva de un proyecto de software depende de la planeación detallada de su avance, anticipando problemas que puedan surgir y preparando con anticipación soluciones tentativas a ellos.

La falta de planeación es la causa principal de retrasos en programación, incremento de costos, poca calidad y altos costos de mantenimiento en los desarrollos de productos de programación.

Con frecuencia, se dice que es imposible una planeación inicial, porque la información precisa sobre las metas del proyecto, necesidades del cliente y restricciones del producto no se conocen al comenzar el proyecto de desarrollo, sin embargo, uno de los principales propósitos de esta fase es aclarar los objetivos, necesidades y restricciones. La dificultad de la planeación no debe desalentar tan importante actividad.

Un producto de programación se entiende mejor según se desarrollan el análisis, el diseño y la instrumentación; sin embargo, el proyecto de desarrollo no debe estar superditado a la disponibilidad de suficiente información para iniciar la planeación preliminar. Se debe reconocer que los planes preliminares se modificarán según vayan evolucionando los productos; la planeación para el cambio es uno de los aspectos clave con los que se logra el éxito.

Los pasos requeridos para planear un proyecto de programación se listan en la **Tabla 3.1** y se analizan en las secciones posteriores:

Tabla 3.1 Planeacion de un proyecto de Programación

Para definir el problema es necesario:

- Desarrollar un enunciado definitivo del problema por resolver. Incluir una descripción de la situación actual, restricciones del problema y de las metas que se lograrán. El enunciado del problema debe realizarse empleando terminología del cliente.
- Justificar una estrategia de solución computarizada para el problema.
- Identificar las funciones por realizar, las restricciones, el subsistema de equipo electrónico, el subsistema del producto de programación y el del personal.
- Determinar los objetivos y requisitos en el nivel del sistema para el proceso de desarrollo y los productos finales.
- Establecer criterios de alto nivel para la aceptación del sistema.

Para el desarrollo de una estrategia de solución es deseable:

- Esbozar varias estrategias de solución, sin considerar las restricciones.
- Realizar un estudio de factibilidad para cada estrategia.
- Recomendar una estrategia de solución, indicando por qué se rechazan las otras.
- Desarrollar una lista de prioridades para las características del producto.

En la planeación del proceso de desarrollo sería adecuado:

- Definir un modelo de ciclo de vida y una estructura organizacional para el proyecto.
- Planear las actividades de administración de la configuración, control de calidad y validación.
- Determinar las herramientas por fase, técnicas y notación por utilizar.
- Establecer estimados preliminares de costo para el desarrollo del sistema.
- Establecer un programa preliminar para el desarrollo.
- Establecer estimados preliminares de recursos de cómputo necesarios para operar y mantener el sistema.
- Preparar un glosario de términos.
- Identificar fuentes de información y referirse a ellas a lo largo del plan del proyecto.

### 3.3.1 Definición de Proyecto.

Los sistemas computacionales, como otros productos de la tecnología, se desarrollan en respuesta a requerimientos detectados. Las fuentes que producen las ideas de productos de programación incluyen las necesidades del usuario generadas externamente, las necesidades internas de la organización, y los planes o misiones organizacionales.

El primer paso en la planeación de un proyecto de programación es preparar, en la terminología del usuario, un enunciado breve del problema que se solucionará y de las restricciones que existen en su resolución. El enunciado definitivo del problema debe incluir una descripción de la situación actual y de las metas que debe lograr el nuevo sistema.

El segundo paso en la planeación de un proyecto de programación es determinar lo apropiado de una solución computacional. Además de ser eficaza en términos de costo, un sistema computacional debe aceptarse social y políticamente.

#### 3.3.1.1 Metas y Requisitos

Las metas son logros por alcanzar; sirven para establecer el marco de referencia para el proyecto de desarrollo del producto de programación. Y pueden ser cualitativas o cuantitativas:

- Meta cualitativa para el proceso: el proceso de desarrollo debe de mejorar las habilidades profesionales del personal de control de calidad
- Meta cuantitativa para el proceso: el sistema se debe de entregar en un plazo máximo de 12 meses
- Meta cualitativa para el producto: el sistema debe hacer más interesante el trabajo de los usuarios

Algunas metas se aplican a todos los proyectos y productos. Por ejemplo, todo producto de programación debe de ser útil, confiable, comprensible y eficiente en costos. Todo proceso de desarrollo debe de producir productos finales a tiempo y dentro de los estimados de costo, y debe permitir que el personal del proyecto aprenda nuevas habilidades. Otras metas, como transportabilidad, entrega anticipada de subsistemas, y facilidad de uso para los no programadores, dependerán de la situación particular.

Los requisitos especifican las capacidades que debe tener un sistema para la solución de un problema. Los requisitos pueden establecer también estándares de desarrollo y de control de calidad tanto para el desarrollo como para el producto; deben de ser cuantificados siempre que sea posible.

Las metas y los requisitos de alto nivel se pueden expresar en términos de atributos de calidad que el sistema deberá poseer. Estos atributos de calidad de alto nivel pueden, a su vez, expresarse en términos de atributos que se pueden obtener en los productos finales. Tal como se cita en la **Tabla 3.2.**

Tabla 3.2 Glosario de Atributos de Calidad

- Portabilidad. Es la facilidad con que un producto de programación puede ser transferido de un sistema de computo a otro, o de un ambiente a otro.
- Confiabilidad. Capacidad de un programa de realizar un función requerida bajo cierta condiciones durante un periodo determinado.
- Eficiencia. Grado con el que un producto de programación efectúa sus funciones, mediante un mínimo de recursos computacionales.
- Exactitud. Especificación cualitativa de ausencia de error.
- Error. Discrepancia entre un condición a valor claculado y la condición real, especificada, a valor correcto teórico.
- Solidez. Grado con el que un producto de programación puede continuar operando correctamente a pesar de la introducción de datos invalidos.
- Corrección. Grado en el que un producto de programación esta libre de defectos de diseño y de codificación, esto es, libre de fallas.

Los planes describen los mecanismos que se ocuparán para lograr la metas y requisitos. Por ejemplo, el objetivo de entregar los productos finales a tiempo se puede expresar en términos de alcanzar logros importantes del proyecto a tiempo. Un logro es un heco significativo en el ciclo de vida del desarrollo del producto de programación.

Para planear el modo de alcanzar cada logro a tiempo, se debe responder preduntas como las siguientes:

1. ¿ Cuántos logros son apropiados ?
2. ¿ Cuándo ocurren ?
3. ¿ Que recursos se necesita para alcanzar cada logro ?
4. ¿ Quién será el responsable de alcanzarlos ?
5. ¿ Que se debe de cumplir para alcanzar cada logro ?
6. ¿ Que significa alcanzar cada logro ?

La consideración de estas preguntas llevará a aspectos como modelos de ciclo de vida, estimación de cósto y cantidad de personal para el proyecto. Las **Tablas 3.3** y **3.4** presentan algunos factores que deben considerarse cuando se establecen metas y requisitos para los productos de programación y los procesos de su desarrollo.

Tabla 3.3 Algunos factores que deben considerarse en la planeación del proyecto

| |
|---|
| • Técnicas de estimación que se utilizará.; precisión requerida. |
| • Modelo de ciclo de vida, funciones de control y revisiones. |
| • Estructura organizacional. |
| • Nivel de formalidad en especificaciones, planes de prueba , etc. |
| • Nivel de verificación y validadción. |
| • Nivel de administración de la configuración requerida. |
| • Nivel de control de calidad requerida. |
| • Responsabilidades de seguimiento y mantenimiento. |
| • Herramienta que se desarrollarán y emplearán. |
| • Contratación y capacitación de personal. |

Tabla 3.4 Algunos factores que deben considerarse al establecer metas del proytecto

| |
|---|
| • Nuevas capacidades por proporcionarse. |
| • Previas capacidades para preservar/mejorar. |
| • Nivel de complejidad del usuario. |
| • Requisitos de eficiencia |
| • Requisitos de confiabilidad |
| • Modificaciones factibles. |
| • Prioridades de instrumentación y subconjuntos iniciales. |
| • Requistos de portabilidad. |
| • Asuntos de seguridad. |

### 3.3.2 Planeación del Proceso de Desarrollo

La planeación del proceso de desarrollo de un producto de programación comprende varias consideraciones importantes. La primera es definir un modelo para el ciclo de vida del producto. Este ciclo incluye todas las actividades requeridas para definirlo, desarrollarlo, probarlo, entregarlo, operarlo y mantenerlo.

Es esencial definir un modelo de ciclo de vida para cada proyecto de programación, puesto que permite clasificar y controlar las diferentes actividades necesarias para el desarrollo y mantenimiento del producto.

#### 3.3.2.1 Modelo de las fases de Ciclo de Vida.

El modelo de fases divide el ciclo de vida del producto de programación en una serie de actividades sucesivas; cada fase requiere información de entrada, procesos y resultados, todos ellos bien definidos.
El modelo de fases comprende: el análisis, diseño, instrumentación, pruebas y mantenimiento. La Tabla 3.5 muestra las diferentes etapas del ciclo de vida.

**Tabla 3.5 Modelo de fases para el ciclo de vida de desarrollo de productos de programación.**

| Fase | Etapas | Procedimiento |
|---|---|---|
| Análisis | Iniciación | Revisión preliminar |
| | Estudio de viabilidad | Procedimientos existentes |
| | | Sistemas alternativos |
| | | Estimaciones de costos |
| | Planeación | Detalles de los procedimientos actuales |
| | | Recolección de datos en volúmenes, sistemas de entrada y salida, registros |
| Diseño | Diseño | Sistema ideal no limitado |
| | | Revisiones para hacer aceptable el ideal |
| Instrumentación | Especificaciones | Lógica de procesado |
| | | Diseño de los registros |
| | | Sistemas de entrada y salida de datos |
| | | Requisitos de programación |
| | | Procedimientos manuales |
| | Programación | Programación |
| Pruebas del Sistema | Pruebas | Unidades de prueba |
| | | Pruebas combinadas de módulos |
| | | Pruebas de aceptación |
| | Entrenamiento y enseñanza | Capacitación del uso del sistema |
| | Instalación | Instalación del sistema |
| Mantenimiento | Operaciones | Mantenimiento |
| | | Ampliación |

El ***análisis*** consta de dos fases: planeación y definición de requisitos. Las actividades principales durante la planeación se resumen en la Tabla 3.1.

La planeación consta de la definición del sistema y el plan del proyecto. La definición, por lo regular se expresa en español y puede contener cuadros, figuras, gráficas, etc. El formato de la definición del sistema se presenta en la **Tabla 3.6**.

**Tabla 3.6 Formato de la definición del sistema**

- Definición del problema
- Justificación del sistema
- Metas del sistema y del proyecto
- Restricciones del sistema y del proyecto
- Funciones que se proporcionarán (equipo/programación/personal)
- Características del usuario
- Ambientes de desarrollo/operación/mantenimiento
- Estrategia de solución
- Prioridades para las características del sistema
- Criterios de aceptación del sistema
- Fuentes de información
- Glosario de términos

El plan del proyecto tiene el modelo del ciclo de vida que se utilizará, la estructura organizacional del proyecto, la programación preliminar del desarrollo, estimados preliminares de costos y productos, así como herramientas y técnicas que se emplearán, y estándares que se seguirán. Los elementos que se deben incluír en el plan de proyecto se listan en la **Tabla 3.7.**

**Tabla 3.7 Formato del plan del proyecto.**

- Modelo del ciclo de vida
- Terminología/Logros/productos finales
- Estructura organizacional
- Estructura de administración
- Requisitos preliminares de personal y recursos
- Programación preliminar del desarrollo
- Estimado preliminar de costos
- Mecanismos de supervisión y control del proyecto
- Herramientas y técnicas que se emplearán
- Lenguajes de programación
- Requisitos de pruebas
- Documentos de apoyo necesario
- Formas de demostración y entrega
- Programación de entrenamiento
- Plan de instalación
- Entrega final

En el modelo de fases, el *diseño* de la programación viene después de la planeación. Este se refiere a la identificación de los componentes de la programación, especificando las relaciones entre ellos, la estructura de la programación y manteniendo un registro de las decisiones, proporcionando un documento base para la instrumentación. El diseño se divide en estructural y detallado.

El diseño estructural comprende la identificación de los componentes de la programación, su desacoplamiento y descomposición en módulos de procesamiento y estructuras de datos conceptuales y la especificación de las interconexiones entre componentes.

El diseño detallado se refiere a detalles de cómo empacar módulos de procesamiento y cómo instrumentar los algoritmos, las estructuras de datos y sus interconexiones.

La fase de ***instrumentación*** en el desarrollo de producto incluye la traducción de las especificaciones del diseño en código fuente, así como su depuración, documentación y pruebas. Los lenguajes de programación modernos proporcionan muchas características para mejorar la calidad del código fuente, como elementos estructurados, tipos de datos predefinidos o definidos por el usuario, verificación de tipos, reglas flexibles de cobertura, mecanismos para manejo de excepciones, elementos concurrentes y módulos con compilación separada.

Los errores descubiertos durante la fase de instrumentación pueden ser errores en las intercaras de datos entre rutinas, errores lógicos en los algoritmos, errores en las estructuras de datos y de falta de consideración de casos de procesamiento.

Una de las metas principales del modelo de fases para el desarrollo de productos de programación es la eliminación de errores de requisitos y diseño, antes de iniciada la instrumentación.

Las ***pruebas*** del sistema comprenden dos tipos de actividades: pruebas de integración y pruebas de aceptación.

Las pruebas de aceptación se relacionan con la planeación y ejecución de varios tipos de pruebas para demostrar que el sistema de programación instrumentado satisface las necesidades establecidas en el documento de requisitos.

Una vez aceptado por el cliente, el sistema de programación se entrega para operación y se inicia la fase de mantenimiento del modelo de ciclo de vida por fases. Las actividades de mantenimiento incluyen mejoras de las capacidades,

adaptación a nuevos ambientes de procesamiento y corrección de fallas del sistema.

### 3.3.2.2 Logros, documentos y revisiones.

El análisis siguiente presenta documentos característicos, logros de proyecto, revisiones y puntos de control utilizados en el modelo de fases del desarrollo de productos de programación.

Se prepara una Definición del proyecto y un Plan del proyecto, usando los formatos de las **Tablas 3.6** y **3.7.** Se realiza una revisión de la factibilidad del producto para determinar la continuación del proyecto, redirigir o continuar como se mostró en la definición del sistema y el pan del proyecto.

Se prepara una versión preliminar del Manual del Usuario; proporciona un mecanismo de comunicación entre cliente y equipo de desarrollo; se efectúa utilizando información de la definición del sistema y de los resultados de los estudios de prototipos y pruebas de despliegues e informes. Con frecuencia, las capacidades que se describen en el Manual del usuario preliminar forman parte de la especificación de requisitos. El bosquejo de un Manual del usuario ordinario se ilustra en la **Tabla 3.8.**

**Tabla 3.8 Bosquejo del manual del usuario**

| |
|---|
| • Introducción |
| • Panorama y exposición del producto |
| • Terminología y características básicas |
| • Resumen de informes y despliegues |
| • Bosquejo del manual |
| • Pasos iniciales |
| • Arranque |
| • Modo de ayuda |
| • Corrida ejemplo |
| • Modos de operación |
| • Comandos/diálogos/informes |
| • Características especializadas |
| • Sintaxis de los comandos y opciones del sistema |

Se prepara una especificación de requisitos para la producción de software, definiendo con claridad y precisión cada requerimiento, así como las interfases externas hacia el equipo, otros programas y el personal. Cada requisito se debe definir de manera que pueda verificarse con un método como inspección, demostración, análisis o pruebas. El formato de la Especificación de requisitos para la producción de software se muestra en la **Tabla 3.9.**

**Tabla 3.9 Formato de la Especificación de requisitos para la producción de software.**

| |
|---|
| Panorama del producto y resumen. |
| Ambientes de desarrollo/operación/mantenimiento |
| Interfases externas y flujos de datos |
| Despliegues al usuario/formato de informes |
| Resúmen de comandos del usuario |
| Diagramas de flujos de datos de alto nivel |
| Fuentes y destinos lógicos de datos |
| Almacenamientos lógicos de datos |
| Diccionario lógico de datos |
| Especificaciones funcionales |
| Requisitos de operación |
| Condiciones de excepción/manejo de excepciones |
| Subconjuntos iniciales y prioridades de instrumentación |
| Modificaciones y mejores previstas |
| Criterios de aceptación |
| Pruebas funcionales y de operación |
| Estándares de documentación |
| Guías de diseño |
| Fuentes de información |
| Glosario de términos |

Se prepara una versión preliminar del plan de verificación del software que establece los métodos que se usarán y los resultados que se obtendrán en la verificación de cada requisito definido en la especificación para la producción de software. El bosquejo de plan de verificación del software se muestra en la **Tabla 3.10.**

**Tabla 3.10 Bosquejo del plan de verificación del software**

| |
|---|
| • Requisitos que se verificarán |
| • Plan de verificación del diseño |
| • Plan de pruebas de código fuente |
| • Criterios de terminación de pruebas |
| • Plan de verificación de documentos |
| • Herramientas técnicas que se utilizarán |

Se realiza una revisión de los requisitos de software para asegurar la consistencia entre la definición del sistema, el plan del proyecto, la especificación de los requisitos para la producción de software, el plan de verificación del software y el manual del usuario preliminar.

Las metas principales de esta revisión son el acuerdo de todos en la terminología; interpretación uniforme de las especificaciones, y la exposición de áreas de problemas. Una vez que las características principales se han establecido, se lleva a cabo una revisión final de requisitos y se genera un documento aprobado formalmente por el cliente y el equipo de desarrolo.

El equipo de diseño que puede ser o no el mismo que el del análisis, genera la Especificación del diseño de software en dos etapas, primero el documento de diseño estructural, y en seguida, después de una revisión preliminar del diseño, el documento de diseño detallado. La información que contienen la especificación estructural y detallada se muestra en **Tabla 3.11** y **3.12**.

**Tabla 3.11 Contenido de una especificación de diseño arquitectónico**

- Diagrama de flujo de datos del producto
- Descripción conceptual de estructuras y bases de datos
- Nombres, unidades, y otros atributos de los elementos de datos
- Nombre y descripción funcional de cada módulo
- Especificación de intercaras para cada módulo
- Estructura de interconexión entre módulos
- Interconexiones entre módulos y estructuras de datos
- Restricciones de tiempo
- Condiciones de excepción

**Tabla 3.12 Contenido de la especificación del diseño detallado**

- Descripción física de estructuras y bases de datos
- Especificación de diccionario para todos los elementos de datos
- Algoritmos detallados para cada módulo que se genera
- Adaptaciones necesarias para el código existente que será reutilizado
- Técnicas específicas de programación necesarias para resolver problemas especiales
- Procedimientos de inicio
- Pruebas para autorizaciones y manejo de excepciones
- Empacado de módulos en la instrumentación física

Se lleva a cabo una revisión del diseño preliminar para evaluar la consistencia respecto a la especificación para la producción de software. Se puede necesitar más de una revisión para resolver los problemas, y se requiere una aprobación final del administrador del proyecto.

Después del diseño detallado, se realiza una revisión crítica, cuyo propósito es determinar si se acepta la especificación del diseño de sotware; la revisión requiere autorización del administrador del proyecto.

Durante la fase de diseño, el plan de prueba de aceptación se expande para incluir los métodos que se usarán para verificar que el diseó y el código fuente sean consistentes y completos respecto a los requisitos y diseño.

Se lleva a cabo una verificación de la programación para evaluzar su consistencia e integridad y para revisar la versión preliminar del plan de prueba de aceptación. Dicho plan incluye los casos de prueba reales, resultados esperados y capacidades que demostrará cada caso. El plan de aceptación se inicia durante la fase de diseño y se concluye durante la instrumentación. El bosquejo de este plan se presenta en la **Tabla 3.13**.

**Tabla 3.13 Formato del plan de prueba de aceptación**

| |
|---|
| • Requisitos que se verificarán<br>• Casos de prueba para cada requisito<br>• Resultado esperado de cada caso de prueba<br>• Capacidades demostradas por cada prueba |

Durante la instrumentación, se efectúan revisiones del código fuente, de manera que se asegura que todo el código ha sido revisado, por lo menos, por una persona más que el programador que lo escribió y antes de que se integre al producto final.

Durante la evolución del producto se realizan inspecciones y recorridos para verificar la integridad, consistencia y adecuación del producto. Los elementos que se pueden intervenir son especificación de requisitos, documentos de diseño, código fuente y casos de prueba.

El manual del usuario, los planes de instalación y entrenamiento, y el plan de mantenimiento del software se concluyen durante la fase de instrumentación. Dependiendo de la naturaleza del producto, la instalación y el entrenamiento del cliente, pueden ser simples o bastante complicados, por lo que en la programación del desarrollo se deben asignar tiempo y recursos suficientes para estas actividades. El mantenimiento es una responsabilidad contractual opcional de la organización que desarrollo el producto, en algunos casos ésta produce un plan de mantenimiento y en otras ocasiones es el cliente quien lo hace.

Antes de la última entrega del producto, se realiza una revisión final que confirma el cumplimiento de todos los requisitos y verifica que el código fuente y todos los documentos externos estén completos, sean consistentes y se hallen

listos para entregarse. Se prepara un Resumen de verificación del software que describe los resultados de las revisiones, auditorías, inspecciones, y pruebas realizadas durant el ciclo de desarrollo.

Finalmente, se redacta un Legado del proyecto que resume el proyecto y registra los aciertos y errores durante su desarrollo. Un error frecuente de las organizaciones que desarrollan productos de programación, es no asignar ni tiempo ni recursos para esta fase, lo cual dificulta el aprovechamiento de experiencias pasadas tanto de éxito como de fracaso. El bosquejo de un legado común se presenta en la **Tabla 3.14**.

**Tabla 3.14 Formato del legado del proyecto**

| |
|---|
| • Descripción del proyecto |
| • Expectativas iniciales |
| • Situación actual del proyecto |
| • Areas que falta atender |
| • Registro de actividades/tiempo |
| • Lecciones técnicas aprendidas |
| • Lecciones administrativas aprendidas |
| • Recomendaciones para proyectos futuros |

No todas las actividades presentadas se necesitan para todos los productos de programación; sin embargo, se indican los logros, documentos y revisiones que pueden ocurrir durante un desarrollo.

### 3.3.3 Otras actividades de la planeación.

Otras actividades de la planeación comprenden la administración de la configuración y el control de calidad, la validación y verificación y las herramientas y técnicas de cada fase.

#### 3.3.3.1 Planeación para la Administración y el Control de Calidad

La actividad de control de calidad del software está muy relacionada con las actividades de verificación y comprobación efectuadas en cada etapa del ciclo de vida del software.

El control de calidad y otras actividades de verificación y comprobación son en realidad muy distintas: el control de calidad es una función administrativa y la comprobación y verificación son parte del proceso de desarrollo del software.

Bersoff (1984) da la siguiente definición de control de calidad:

El control de calidad consiste en aquellos procedimientos, técnicas e instrumentos aplicados por profesionales para garantizar que un producto cumple o supera los estándares predefinidos durante el ciclo de desarrollo de un producto; sin estándares específicos preestablecidos, el control de calidad implica que un producto cumpla o supere un nivel mínimo industrial o comercialmente aceptable de calidad.

Los temas que se deben contemplar en un Plan de Control de Calidad son:

1. Propósito y alcances del plan.
2. Documentos referidos en el plan.
3. Estructura organizacional,tares y responsabilidades relacionadas con la calidad del producto.
4. Documentos que se deben preparar y revisiones que deben efectuarse para la adecuación de la documentación.
5. Estandares, prácticas y convenciones que se utilizaran.
6. Revisiones y auditorias que deben llevarse a cabo.

La actividad de control de calidad implica participar en revisiones de diseño, recorridos de programas, etc. e informar sobre la calidad total del producto durante su desarrollo. También implica revisar que el producto terminado y su documentación asociada se apeguen a los estándares existentes.

Uno de los factores que generalmente se interpretan como señal de calidad, es la facilidad de mantenimiento que ofrece un sistema. El mantenimiento del software consiste en la posibilidad de eliminar errores, el afinamiento, es decir las mejoras en cuanto a las prestaciones, así como en la posibilidad de la ampliación, para incorporar nuevas funciones.

También es interesante la eficiencia del sistema, con esto generalmente se hace referencia a la velocidad de ejecución de las funciones. Sin embargo, la eficiencia también comprende el aprovechamiento óptimo de los recursos disponibles, tales como la memoria, los periféricos, etc. Estos dos criterios de eficiencia, sin embargo casi siempre resultan contradictorios. Los programas rápidos requieren una mayor cantidad de memoria, mientras que los programas que gestionan óptimamente la memoria, son frecuentemente más lentos. Otros factores importantes son la confortabilidad y la claridad, además de la seguridad del sistema - protección contra la pérdida de datos y contra accesos no permitidos por parte de terceros.

### 3.3.3.2 Planeación para Verificación y Validación Externas

La verificación asegura que los productos estén completos y sean consistentes con otros y con las necesidades del usuario. Es decir verificar que las especificaciones del diseño estén completas y que sean consistentes con la

definición del sistema y las especificaciones del software, que el código fuente sea consistente con el diseño y los requisitos.

La validación se relaciona con probar la calidad del software en su medio de operación, y suele consistir en la planeación y realización de casos de prueba.

Probar es demostrar que no hay errores presentes en un programa, el propósito de probar es mostrar que el programa realiza correctamente las funciones esperadas y probar es el proceso que lleva a confiar en que un programa hará lo que se supone que debe hacer.

Estas definiciones son incorrectas porque describen casi lo opuesto de lo que debemos considerar como prueba de un programa.

De aquí que no se deba probar un programa para mostrar que funciona, más bien es conveniente comenzar con la suposición de que el programa contiene errores y luego probar el programa para encontrar tantos como sea posible. Una definición más apropiada es: Prueba es el proceso de ejecutar un programa con el fin de encontrar errores.

- Prueba de aceptación

El proceso de prueba, al igual que el de programación, debe avanzar en etapas, siendo cada una de ellas la continuación lógica de la etapa anterior.

En el proceso de prueba se pueden identificar cinco etapas:

1. Prueba de funciones : La prueba de funciones o de unidades es el nivel básico en donde se prueban las funciones que componen un módulo para garantizar que operan de manera correcta. En un sistema de diseño apropiado, cada función debe tener una sola especificación definida con claridad. Las funciones no deben depender de otras funciones de su mismo nivel, para posibilitar la prueba de cada función como una entidad aislada, sin la presencia de otras funciones.

2. Prueba del módulo : Se compone de varias funciones que pueden cooperar entre sí. Después de haber probado cada función individual, es necesario probar la cooperación de estas funciones cuando componen un módulo. Debe ser posible probar un módulo como una entidad aislada, sin la presencia de otros módulos del sistema.

3. Prueba de subsistemas : Esta prueba es el siguiente paso del proceso en el cual los módulos se agrupan para formar subsistemas. Puesto que los módulos cooperan y se comunican, la prueba de subsistemas se debe centrar en la prueba de las intercaras de aquellos, dando por supuesto que los módulos son correctos.

4. Prueba del sistema : La prueba de sistema se lleva a cabo cuando se integran los subsistemas para conformar el sistema completo. En esta etapa, el proceso de prueba tiene que ver con el hallazgo de errores en el diseño y la codificación.

5. Prueba de aceptación : Todas las pruebas se realizan mediante el empleo de datos generados por la organización encargada de construir el sistema. La prueba de aceptación del sistema se efectúa con datos reales; la información con la que el sistema deberá operar.

En lugar de confirmar un programa mediante pruebas, la verificación implica demostrar con métodos matemáticos, la correspondencia entre un programa y sus especificaciones.

### 3.3.3.2.1 Pruebas de Unidad y Depuración

Las pruebas de unidad comprenden el conjunto de pruebas efectuadas por un programador individual, antes de la integración de la unidad en un sistema más grande.

Una unidad de programa suele ser lo suficientemente pequeña como para que el programador que la desarrolló pueda probarla con minuciosidad, esta prueba debe ser mucho más minuciosa que el examen al que se someterá cuando la unidad se integre en el producto de software en desarrollo. Hay cuatro casos de prueba que se ejecutarán a una unidad de programa:

- Pruebas funcionales
- Pruebas de desempeño
- Pruebas de tensión
- Pruebas de estructura

Los casos de prueba funcional implican ejercitar el código con valores nominales de entrada para los cuales se conocen los resultados, además de valores límites (valores mínimos, máximos y valores sobre y justo fuera de los límites funcionales).

Las pruebas de desempeño determinan la cantidad de tiempo de ejecución empleado en varias partes de la unidad, la eficiencia global del programa, el tiempo de respuesta y la utilización de dispositivos por la unidad de programa.

Las pruebas de tensión son aquellas diseñadas para romper, en forma intencional la unidad. También se les llama a las pruebas funcional, de desempeño y de tensión pruebas de caja negra.

Usando este último nombre la persona que efectúa la prueba considera el programa como una caja negra, es decir, se desentiende completamente del comportamiento y estructura internos del programa. En cambio su interés se dirige solamente a encontrar circunstancias en las cuales el programa no se comporta de acuerdo con sus especificaciones. Los datos de prueba son preparados solamente en base a las especificaciones.

De acuerdo con este enfoque, si se desea encontrar todos los errores en un programa, el criterio deberá ser probar la entrada de datos exhaustivamente. Para asegurar la detección de todos los errores de este tipo habría que usar, no solamente los datos de entrada válidos sino también todos los datos de entrada posibles.

Las actividades principales en las pruebas estructurales son: decidir cuales rutas ejercitar; obtener los datos de prueba para ejercitar esas rutas, determinar el criterio de cobertura de la prueba que se usará, ejecutar los casos de prueba y medir la cobertura de la prueba lograda cuando se ejercitaron estos casos. A las pruebas de estructura también se les denomina pruebas de caja blanca o de cristal.

Las pruebas de caja blanca permiten examinar la estructura interna del programa. Usando esta estrategia, el operador de la prueba obtiene datos de prueba a partir del examen de la lógica del programa.

Aún cuando fuera posible probar con éxito todas las rutas a través de un programa, la corrección no estaría garantizada por las pruebas en las rutas porque el programa puede tener rutas ignoradas y errores computacionales que los casos de prueba particulares elegidos han descubierto.

La depuración es el proceso de aislar y corregir las causas de los errores conocidos. El éxito en la depuración requiere habilidades altamente desarrolladas en la solución de problemas. Los métodos de depuración que suelen ocuparse son la inducción, la deducción y el encadenamiento hacia atrás. La depuración por inducción necesita los siguientes pasos:

1. Reunir la información disponible. Enumerar los hechos conocidos acerca de la falla y los hechos conocidos concernientes a los casos de prueba exitosos
2. Buscar patrones. Examinar la información reunida en busca de condiciones que diferencien el caso con falla de los exitosos.
3. Formular una ó más hipótesis. Obtener una o más hipótesis de las relaciones observadas.
4. Demostrar o desechar cada hipótesis. Volver a examinar la información disponible para determinar si la hipótesis explica o no todos los aspectos del problema observado.

5. Realizar las correcciones adecuadas. Hacer las correcciones indicadas por la evaluación de las distintas hipótesis.
6. Verificar la corrección. Volver a correr los casos con falla para asegurarse que el arreglo corrige el síntoma observado. Si el arreglo no tuvo éxito, se regresa al paso 1.

La depuración por deducción procede como sigue:

- Listar las posibles causas de la falla observada
- Usar la información disponible para eliminar varias hipótesis
- Elaborar las restantes hipótesis
- Probar o rechazar cada hipótesis
- Determinar las correcciones apropiadas
- Verificar las correcciones

La depuración por encadenamiento hacia atrás implica que el código fuente se recorra hacia atrás desde el punto donde se observó el error, en un intento por identificar el punto exacto en que ocurrió el error.

Las técnicas de depuración tradicionales utilizan proposiciones de salida de diagnóstico, vaciados instantáneos, rastreos selectivos sobre valores de datos y flujo de control, y puntos de control dependientes de las instrucciones.

Un vaciado instantáneo es una representación al nivel de la máquina del estado parcial o total de un programa en un punto particular en la secuencia de ejecución. Un vaciado instantáneo estructurado es una representación al nivel fuente del estado parcial o total de un programa.

Una facilidad del rastreo es listar los cambios en componentes selectos del estado. En su forma más simple, un rastreo imprimirá todos los cambios en los valores de los datos para todas las variables y todos los cambios en el flujo del control. Un rastreo selectivo rastreará variables específicas y el flujo del control en regiones específicas del texto fuente.

Una facilidad de punto de control tradicional interrumpe la ejecución del programa y transfiere el control a la terminal del programador cuando la ejecución alcanza una instrucción de control especificada en el código fuente.

### 3.3.3.2.2 Pruebas del Sistema

La prueba por módulos (o prueba por unidades) es un proceso que consiste en probar subprogramas individuales, subrutinas o procedimientos pertenecientes a un programa. Existen tres causas para efectuar la prueba por módulos:

1. La prueba por módulos es una forma de poder manejar las posibilidades combinatorias de las pruebas, puesto que la atención es inicialmente enfocada a unidades menores del programa.
2. La prueba por módulos facilita la tarea de localización y corrección de errores (debugging), puesto que cuando se descubre un error, este estará presente en un módulo en particular.
3. Finalmente la prueba por módulos introduce un cierto paralelismo en el proceso de prueba puesto que da la oportunidad de probar simultáneamente múltiples módulos.

Las pruebas del sistema implican dos clases de actividades: pruebas de integración y pruebas de aceptación.

Las estrategias para integrar los componentes del software en un producto que funcione incluyen las estrategias ascendente, descendente y de emparedado.

La estrategia de integración dicta el orden en que los módulos deben estar disponibles, por lo que ejerce gran influencia en el orden en el que se escriben, depuran y han las pruebas de unidad de los módulos.

Las pruebas de aceptación implican la planeación y ejecución de pruebas funcionales, de desempeño y de tensión para verificar que el sistema realizado satisfaga sus requisitos.

La integración ascendente es la estrategia tradicional empleada para integrar los componentes de un sistema de software en un todo funcionando.

La integración ascendente consiste en pruebas de unidad, seguidas por pruebas de subsistemas y luego por pruebas del sistema completo. Las primeras tienen el objetivo de descubrir errores en los módulos individuales del sistema. Estos módulos se prueban aislados unos de otros en un ambiente artificial llamado arreo de prueba.

El propósito principal de las pruebas de subsistemas es verificar la operación de las interfases entre módulos en el subsistema. Se deben probar tanto las interfases de control como las de datos.

La integración descendente empieza con la rutina principal y una o dos rutinas inmediatamente subordinadas en la estructura del sistema. La prueba de integración descendente integrada ofrece las siguientes ventajas:

1. La integración del sistema se distribuye en toda la fase de implantación. Los módulos se integran a medida que se desarrollan.
2. Las intercaras del nivel más alto se prueban primero y con más frecuencia.
3. Las rutinas del nivel más alto proporcionan un arreo de prueba natural para las rutinas de los niveles inferiores.
4. Los errores se localizan en los nuevos módulos e intercaras que se estén añadiendo.

Comparación de pruebas ascendentes y descendentes.

Pruebas descendentes.
Ventajas.
1. Ventajosa si aparecen fallas grandes en la parte superior del programa
2. Una vez incorporadas las funciones de entrada/salida resulta fácil la representación de casos de pruebas.
3. La estructura previa del programa permite demostraciones y ayuda a mantener la moral.

Desventajas.
1. Se requieren módulos auxiliares.
2. Los módulos auxiliares son a menudo más complicados de lo que al principio parece.
3. Antes de incorporar las funciones de entrada/salida, la representación de los casos de prueba es difícil.
4. Las condiciones de prueba pueden ser imposibles, o muy difíciles de crear.
5. La observación de la salida de la prueba es más difícil.
6. Permite pensar que prueba y diseño pueden superponerse.
7. Induce a diferir la terminación de la prueba de ciertos módulos.

La integración emparedado es predominantemente descendente, pero las técnicas ascendentes se usan en algunos módulos y subsistemas. Esta mezcla mitiga muchos de los problemas encontrados en las pruebas descendentes puras y, además retiene las ventajas de la integración descendente al nivel de subsistema y del sistema.

Una de las preguntas más difíciles de responder cuando se realiza la prueba de un programa es cuándo dar ésta por terminada, puesto que no hay manera de saber si el último error detectado es el único que quedaba. En realidad no es razonable esperar, salvo que se trate de un programa reducido, que se descubran todos los errores.

Los criterios de terminación usados en la práctica carecen de sentido y son contraproducentes. Los más comunes son:

- Terminar la prueba cuando el tiempo establecido para misma ha expirado.

- Terminar la prueba cuando todos los casos de prueba se ejecutan sin detectar errores.

Existen tres categorías de criterios de mayor utilidad:

- La primera categoría, aunque no la mejor, es basar la terminación en el uso de métodos específicos de diseño de casos de prueba.
- La segunda categoría de criterios y quizá la más valiosa, es establecer los requerimientos de terminación en términos positivos. Puesto que el objeto que se busca con las pruebas es encontrar errores, por qué no hacer que el criterio sea la detección de un número predefinido de errores?

Se puede notar que este tipo de criterio refuerza la definición de prueba. Tiene, sin embargo, dos problemas, ambos salvables. Un problema es cómo determinar el número de errores a ser detectados. La obtención de este número requiere:

- Una estimación del número total de errores del programa
- Una estimación del porcentaje de estos errores que pueden encontrarse fácilmente mediante las pruebas
- Estimación de qué fracción de errores se originan en procesos particulares de diseño y durante cuáles fases de las pruebas se puede esperar que sean detectados estos errores.

### 3.3.3.3 Mantenimiento de Software

El término mantenimiento se Software se usa para describir las actividades de la ingeniería de software que ocurren después de entregar un producto al usuario.

La fase de mantenimiento del ciclo de vida del software es el periodo en el que un producto desempeña un trabajo útil.

Las actividades de mantenimiento implican mejorar los productos de software, adaptarlo a nuevos ambientes, y corregir problemas. La mejora de los productos de software puede dar como resultado proporcionar nuevas capacidades funcionales, mejorar los despliegues al usuario y los modos de interacción, revalorar los documentos externos y la documentación interna, o revalorar las características del desempeño de un sistema.

El objetivo principal del desarrollo del software debe ser la producción de sistemas de software que facilite su propio mantenimiento.

Las actividades de análisis durante el mantenimiento de software implican la comprensión del alcance y efecto de una modificación deseada, además de las

restricciones para hacer la modificación. El diseño durante el mantenimiento supone rediseñar el producto para incorporar los cambios deseados. Entonces éstos deben implantarse, la documentación interna del código se debe actualizar, y se deben proyectar nuevos casos de prueba para evaluar la adecuación de la modificación. También, los documentos de apoyo (especificaciones de requisitos y diseño, plan de prueba, principios de operación, manual del usuario, directorios de referencias cruzadas, etc.) se deben actualizar para ilustrar los cambios.

Muchas actividades realizadas durante el desarrollo del software mejoran el mantenimiento de un producto de software, algunas de ellas se listan a continuación:

- Actividades de análisis
  - Desarrollo de estándares y guías
  - Fijar logros en los documentos de apoyo
  - Especificar procedimientos de control de calidad
  - Identificar probables mejoras del producto
  - Determinar recursos requeridos para el mantenimiento
  - Estimar costos de mantenimiento

- Actividades de diseño arquitectónico
  - Subrayar la claridad y modularidad como criterios de diseño
  - Diseñar para facilitar probables mejoras
  - Usar notaciones estandarizadas para documentar flujos de datos, funciones, estructura e interconexiones
  - Observar los principios de cubrimiento de información, abstracción de datos y descomposición jerárquica hacia abajo

- Actividades de diseño detallado
  - Uso de notaciones estandarizadas, para especificar algoritmos, estructuras de datos y procedimientos para especificar intercaras
  - Especificar efectos colaterales y manejo de excepciones para cada rutina
  - Proporcionar directorios con referencia cruzada

- Actividades de implementación
  - Usar estructuras de una sola entrada y una sola salida
  - Usar sangrado estándar en las estructuras
  - Usar un estilo de codificación simple y claro
  - Usar constantes simbólicas para asignar parámetros a las rutinas
  - Proporcionar margen de recursos
  - Proporcionar prólogos estándar de documentación en cada rutina
  - Apegarse a las guías de comentarios estándar internos

- Otras actividades
  Desarrollar una guía de mantenimiento
  Desarrollar un juego de pruebas
  Proporcionar la documentación del juego de pruebas

La fase de análisis del desarrollo de software se relaciona con la determinación de los requisitos y las restricciones del usuario, y el establecimiento de la factibilidad del producto. Desde el punto de vista del mantenimiento, las actividades más importantes sucedidas durante el análisis son señalar estándares y principios generales para el proyecto y para los productos de trabajo de modo que se garantice la uniformidad de los productos; establecer marcas de logros para asegurar que los productos de trabajo se producen en el tiempo programado; especificar los procedimientos de control de calidad para garantizar el desarrollo de documentos de alta calidad.

Se pueden desarrollar varios tipos de estándares y principios generales para mejorar el mantenimiento del software. Los formatos estándar para los documentos de requisitos y las especificaciones de diseño, las convenciones de codificación estructurada, y los formatos estandarizados para los documentos de apoyo como el plan de prueba, los principios de operación, el manual de instalación y el del usuario contribuyen a la comprensión, y por lo tanto al mantenimiento del software.

El diseño estructural se relaciona con el desarrollo de los componentes funcionales, de las estructuras conceptuales de los datos, y de las interconexiones en un sistema de software. La actividad más importante para mejorar el mantenimiento durante el diseño estructural es recalcar la claridad, la modularidad y la facilidad de modificación como los principales criterios de diseño.

La implantación del sistema, al igual que el diseño del mismo, debe tener como objetivo principal el producir un software fácil de comprender y modificar. Además de que se deben usar construcciones de una sola entrada y una sola salida del código.

Hay dos documentos de apoyo particularmente importantes que se deben preparar durante el ciclo de desarrollo del software para facilitar las actividades de mantenimiento. Estos documentos son la guía de mantenimiento y la descripción del conjunto de pruebas. La guía de mantenimiento brinda una descripción técnica de las capacidades operacionales del sistema completo, diagramas de jerarquía, gráficas de llamados y directorios de referencia cruzadas del sistema.

La documentación y el mantenimiento del software son partes tediosas pero esenciales de la producción y uso de sistemas.

La documentación puede describir la manera de utilizar el programa, la razón por la que se escribió y las técnicas utilizadas en su construcción y debe esclarecer cualquier aspecto oscuro del programa. El mantenimiento -el proceso de hacer modificaciones después de haber entregado el programa- requiere la comprensión del programa y esto se logra estudiando su código y la documentación asociada.

Todos los grandes sistemas, con independencia de su aplicación, tienen una cantidad enorme de documentación asociada. Esta puede clasificarse como documentación del usuario o del sistema. La documentación del usuario se compone de aquellos documentos relacionados con las funciones del sistema, sin referirse a la forma de aplicarlas. La documentación del sistema, por otra parte, describe todos los aspectos del diseño, implantación y pruebas del sistema.

La información proporcionada junto con el sistema debe satisfacer varios requisitos. Tiene que describir:

1. Cómo usar el sistema. Sin esto, aun el sistema más simple resulta inútil.
2. Cómo instalar y operar el sistema
3. Los requisitos y diseño de todo el sistema
4. La aplicación del sistema y los procedimientos de prueba, para poderle dar mantenimiento

La documentación del usuario debe proporcionar una visión inicial precisa del sistema. No es literatura comercial, así que no debe destacar en demasía las características del sistema novedosas o muy poderosas, ni debe ser poco realista acerca de las capacidades del sistema. No debe ser indispensable que el usuario lea la mayor parte de la documentación para encontrar cómo utilizar de forma sencilla el sistema.

Hay al menos cinco documentos que se pueden considerar bajo en encabezamiento de documentación del usuario:

- Una descripción funcional sobre lo que puede hacer el sistema
- Un documento que explique cómo instalar el sistema y adecuarlo para configuraciones particulares del hardware
- Un manual introductorio que explique, en términos sencillos, cómo iniciarse en el sistema
- Un manual de referencia que describa con detalle las ventajas del sistema disponibles para el usuario y cómo se pueden usar

- Una guía del operador (si se requiere un operador del sistema), que explique cómo debe reaccionar éste ante situaciones surgidas mientras el sistema se encuentra en uso

La descripción funcional no debe entrar en detalles ni necesita cubrir todas las características del sistema, sino que debe proporcionar una visión general y, cuando se lea junto con un manual introductorio, debe permitir al usuario decidir si el sistema es apropiado a sus necesidades.

El manual introductorio debe presentar un prólogo informal que describa el uso normal del sistema. Debe explicar cómo iniciarse en el sistema y cómo se pueden utilizar sus cualidades comunes. Debe ilustrarse en forma amplia con ejemplos. El manual introductorio también debe decir al usuario del sistema cómo salir de un problema cuando las cosas funcionan mal.

Los principiantes, cualquiera que sea su preparación y experiencia, cometerán inevitablemente errores. Es vital proporcionar información fácilmente disponible sobre cómo recuperarse de esos errores y reiniciar un trabajo útil.

El manual de referencia es el documento definitivo sobre el uso del sistema. La característica más importante de un manual de referencia el que debe ser completo.

Además de describir con detalle las cualidades del sistema y su uso, el manual de referencia también debe describir los informes de error generados, las situaciones en las que surgen esos errores y, si es apropiado, remitir al usuario a una descripción de la característica en la que está el error.

En primer lugar, debe contener una descripción del medio legible para la máquina en la que se proporcione el sistema: su formato, los códigos de caracteres utilizados, cómo está escrita la información y los archivos que componen el sistema. Después, debe describir la configuración mínima de hardware que se requiere para ejecutar el sistema, los archivos permanentes que deben establecerse, cómo iniciar el sistema y los archivos dependientes de la configuración que se deben modificar para adecuar el sistema a una aplicación particular.

Los sistemas de ayuda en línea, que contienen información breve acerca del sistema, son otra característica que evita al usuario perder tiempo consultando manuales.

El término documentación del sistema se refiere a todos los documentos que pertenecen a la aplicación del sistema, desde la especificación de los requisitos hasta el plan de pruebas de aceptación final.

Los documentos que componen la documentación del sistema deben incluir:

- La definición de requisitos y, quizás, una fundamentación asociada.
- Una especificación general de los sistemas que muestre cómo se descomponen los requisitos en un conjunto de programas interactuantes. Este documento no se requiere cuando el sistema se aplica por medio de un solo programa.
- Por cada programa del sistema, una descripción de cómo se descompone ese programa en componentes y una declaración sobre la especificación de cada componente.
- Por cada unidad, una descripción de su operación. Esta no necesita extenderse a la descripción de acciones del programa, pues tales acciones se deben documentar utilizando comentarios dentro del programa
- Un plan de pruebas amplio que describa cómo se prueba cada unidad de programa
- Un plan de prueba que muestre cómo se efectuó la prueba de integración, esto es, la prueba de todas las unidades y programas juntos
- Un plan de pruebas de aceptación, diseñado junto con el usuario del sistema. Este plan debe describir las pruebas que es necesario pasar antes de aceptar el sistema

La calidad de la documentación es tan importante como la calidad del programa.
Un procedimiento estándar para producir, revisar y acomodar la documentación es una ayuda importante para el control de la calidad de la documentación. Nunca será excesivo destacar la importancia de los mecanismos para el control de la calidad en la producción de documentos. Una documentación pobre quizá confunda más que ayude al usuario, con la consecuencia de que es probable que se haga un uso mínimo de ella.

Dentro de una organización, es útil establecer un formato estándar y solicitar que todos los documentos se ajusten a él. Dicho estándar puede incluir la descripción de un formato para la cubierta frontal para ser adoptado en todos los documentos, las convenciones para la numeración y anotación de página, los métodos para hacer referencia a otros documentos y la numeración de los títulos y subtítulos.

Escribir documentos bien no es fácil ni constituye un proceso de una sola etapa. Se debe redactar, leer, criticar y volver a escribir, repitiendo el proceso hasta que se produzca un documento satisfactorio.

Estos principios generales, dispuestos en un estilo que se puede usar en los manuales de instrucción del software, son:

- Utilizar formas gramaticales activas en vez de pasivas
- No emplear frases largas que presenten varios hechos distintos
- No hacer referencia a la información sólo con el número de referencia
- Detallar los hechos siempre que sea posible
- Si determinada descripción es compleja, debe repetirse
- Ser concreto
- Ser preciso y definir los términos utilizados
- Utilizar párrafos cortos
- Utilizar títulos y subtítulos
- Utilizar construcciones gramaticales y ortografía correctas

Un mecanismo para examinar y mejorar los documentos es establecer inspecciones de documentos. Estas se utilizarán de la misma manera que las inspecciones del código para la detección de errores en los programas. Durante la inspección de un documento se critica el texto, se señalan las omisiones y se hacen sugerencias para mejorarlo.

La necesidad de contar con indices efectivos de los documentos es de especial importancia cuando la documentación se mantiene en línea. No tiene caso disponer de la información si el lector no puede encontrarla con la facilidad y rapidez necesarias.

### 3.3.3.4 Planeación de las Herramientas y Técnicas Específicas de cada Fase del Proyecto

Para el desarrollo de la especificación de requisitos, el diseño estructural y detallado, y el código fuente, se pueden emplear herramientas automatizadas, notaciones especializadas y técnicas modernas. Además, en las pruebas de módulos, sistema y de aceptación también pueden ocuparse herramientas automatizadas.

Para controlar el desarrollo se utilizan herramientas de administración como el método PERT, gráficas de Gantt, estructuras de división del trabajo, gráficas de personal, etc. El empleo de estas herramientas, técnicas y notaciones necesita tiempo para su uso y para el entrenamiento requerido. Esto debe ser anticipado durante la fase de planeación.

# 4. APLICACION DEL SISTEMA DE BASES DE DATOS.

## 4.1 PLANEACION DE PROCESO DE DESARROLLO

### 4.1.1 Definición del Proyecto.

**Definición del Problema.**

El laboratorio de Ingeniería Química actualmente no cuenta con los recursos y espacios necesarios para mostrar todas las válvulas y bombas existentes en la industria, para poder enseñar de una manera adecuada a los alumnos la asignatura de flujo de fluidos.

**Justificación.**

Como una respuesta al problema anteriormente descrito, se propone el sistema "*Válvulas y Bombas*", el cual utilizando la tecnología actual de desarrollo de software, presenta características innovadoras como la presentación de fotos, que se almacenan en una base de datos, con lo cual el alumno podrá ver y consultar diferentes tipos de bombas y válvulas con solo oprimir un botón.

#### 4.1.1.1 Metas y Requisitos.

**Meta cualitativa del proceso.**
La persona encargada de desarrollar el sistema deberá cumplir con los siguientes objetivos al finalizar.

- Reafirmar conocimientos de la asignatura de Programación y Computación, profundizando en el campo de bases de datos.
- Conocer las necesidades existentes en el laboratorio de Ingeniería Química relacionadas con Válvulas y Bombas.
- Conocer los aspectos involucrados en el desarrollo de un producto de software.
- Reafirmar información acerca de los diferentes tipos de equipos de válvulas y bombas.

**Meta cuantitativa del proceso.**
El sistema *"Válvulas y Bombas"* deberá entregarse en un plazo no mayor a 60 días hábiles.

**Meta cualitativa para el producto.**

- El resultado será un sistema sumamente amigable, es decir que sea sencillo de operar y ejecutar, facilitando con esto el estudio de los equipos (bombas, válvulas y compresores) anteriormente descritos.
- Cuenta con un banco de datos extenso de información de válvulas y bombas mostrando características y usos , entre otros.

- Programa realizado en ambiente Windows, que utiliza y aprovecha todas las ventajas que este ambiente tiene.
- Cuenta con un sistema de ayuda en linea, para facilitar su uso.

### 4.1.1.2 Análisis.

#### 4.1.1.2.1 Iniciación

**a) Restricciones del sistema y del proyecto**

El sistema solo funciona en la plataforma Windows.

El manejo de Bases de Datos gráficas todavía es lento y requiere de una gran cantidad de memoria de almacenamiento.

**b) Funciones que se proporcionarán (equipo/programación/personal)**

En cuanto al Hardware requerido para el buen funcionamiento del Software es indispensable que se cumplan los siguientes requisitos:

| Componentes | Descripción / Observaciones |
|---|---|
| Microprocesador | 80386 ó superior |
| RAM | 4 MB |
| Disco Duro | Se requiere un disco duro y un mínimo de 20 MB |
| Monitor | EGA ó superior |
| Microsoft Windows | Versión 3.1 ó posterior |
| Ratón | Aunque no es imprescindible, resulta recomendable |

Antes de arrancar el sistema es necesario verificar algunos valores de los archivos CONFIG.SYS y AUTOEXEC.BAT.

El archivo config.sys debe contener los siguientes valores:
FILES = 40
BUFFERS = 40

Si se modifican los archivos config.sys o autoexec.bat, salga de la aplicación y reinicie la computadora para que los cambios realizados tengan efecto.

A su vez, el sistema será desarrollado por una sola persona.

**c) Características del usuario.**
El usuario debe tener conocimientos de la plataforma Windows, así como debe conocer el sistema operativo DOS.

**d) Ambientes de desarrollo.**
Se desarrolló en plataforma Windows, debido a la gran facilidad y excelente presentación que esta plataforma presenta, además de que el desarrollar en Windows, el programador hereda todas la utilierías que tiene dicha plataforma, como: las impresoras, los tipos de letras, el administrador de impresión, etc.

**e) Criterios de aceptación del sistema**
Los atributos de calidad definidos en la Tabla 3.2 se aplican al sistema de la siguiente manera:

| Atributos | Si | No |
|---|---|---|
| Portabilidad | | X |
| Confiabilidad | X | |
| Eficiencia | X | |
| Exactitud | N/A | |
| Error | | N/A |
| Solidez | X | |
| Corrección | X | |

Portabilidad. El sistema está diseñado en Paradox para Windows, por lo cual no se puede exportar a diferentes lenguajes de programación. Mientras que la base de datos si puede exportarse a otros sistemas de bases de datos, siempre que estos puedan manejar imágenes, gráficos.

Confiabilidad. El programa tiene la capacidad de trabajar sin fallar, el tiempo que sea requerido.

Eficiencia. El Paradox para Windows dispone de un manejador de bases de datos muy rápido, pero hay que tomar en cuenta, que el procesamiento de imágenes en una base de datos es todavía lento. Por lo que se recomienda tener un equipo de computación adecuado (véase requerimientos del sistema).

Exactitud. No aplica.

Error. No aplica.

Solidez. El sistema cuenta con plantillas especiales para evitar la introducción de registros no validos que afecten el funcionamiento del sistema. La operación del sistema es muy sencilla, además, cuenta con el respaldo de ayuda en linea donde se muestran los pasos necesarios para ejecutar correctamente el sistema.

Corrección. El programa fue sometido a una serie de pruebas de validación para garantizar el correcto funcionamiento de cada modulo del sistema.

**f) Fuentes de información**

Recopilación de Información. Se visito a diferentes empresas para conseguir la información más reciente en el mercado de Bombas y Válvulas. La información teórica de los equipos se puede consultar en el **Anexo I.**

**4.1.1.2.2 Planeación**

**a) Modelo de las fases del ciclo de vida de desarrollo del sistema de bases de datos.**

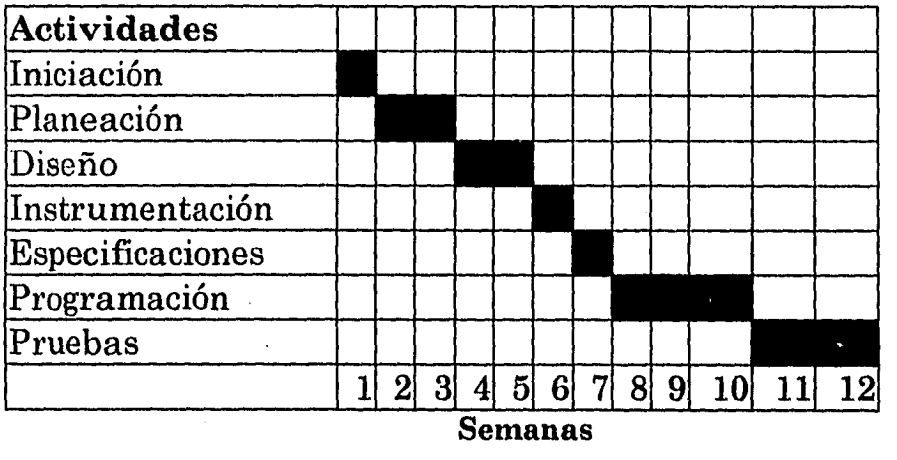



**FIG. 4.1 Modelo de las fases del ciclo de vida.**

**b) Estructura Organizacional.**

Todo el sistema será desarrollado por el alumno que presenta esta tesis.

**c) Requisitos preliminares del personal.**

El personal deberá conocer ampliamente el desarrollo de programas en Ambiente Windows, así como, las necesidades del laboratorio de Ingeniería. Deberá tener conocimientos de bases de datos relaciónales.

## d) Lenguajes de Programación.

El paquete de Bases de Datos Relacional Paradox para Windows fue elegido como herramienta de desarrollo del Sistema de Bases de Datos de Bombas y Válvulas, porque ofrece las siguientes ventajas:

- Paradox para Windows proporciona al entorno Microsoft Windows un paquete de Bases de Datos relacional de gran rapidez, fácil utilización y máxima funcionalidad, adaptado a las necesidades de gestión de datos de los usuarios.
- Es un programa de máxima eficiencia con el que pueden realizarse tareas de gran complejidad, de manera rápida y fácil.
- Permite controlar al máximo la creciente cantidad de información que se maneja a diario.
- Se pueden gestionar datos a cualquier nivel
- Permite utilizar gráficos que facilitan el manejo de información del sistema, además de que mejoran la apreciación y estética del mismo.

Las necesidades de los usuarios de bases de datos se incrementan día con día.

El volumen de datos con los que se trabaja a diario es cada vez mayor, de ahí la necesidad de incluirlos en tablas de tamaño reducido y fácil utilización y establecer enlaces entre éstas a fin de consultar sus datos y crear informes y fechas multitabla. Paradox permite realizar estas tarea de forma rápida y sencilla.

Los sistemas de bases de datos relacionales permiten establecer relaciones (denominadas enlaces) entre varias tablas, de forma que es posible recoger o combinar datos de diferentes tablas para obtener resultados específicos.

Los programas de bases de datos de gran capacidad siempre se han caracterizado por su difícil aprendizaje y lento funcionamiento, este Sistema de Bases de Datos de Bombas y Válvulas, por el contrario de todos los demás facilita su manejo y utilidad.

En el Sistema de Bases de Datos de Bombas y Válvulas se utilizan diversos objetos, como tablas, informes o consultas, entendiendo por objetos los componentes que sirven para almacenar y presentar datos.

Las tablas utilizadas están dispuestas en filas y columnas. Cada fila contiene toda la información disponible acerca de un equipo en particular, y constituye un registro. Cada columna contiene una categoría de los datos que componen un registro, y recibe el nombre de campo.

Existen diferentes categoría de campos, únicamente se mencionan a continuación las utilizadas para el diseño del Sistema:

**Tabla 4.3 Tipos de campos utilizados en la programación del sistema.**

| Tipo de Campo | Tamaño de Campo | Descripción |
|---|---|---|
| Alfanumérico | 1 a 255 | Contiene letras, números, símbolos especiales, o cualquier otro carácter imprimible. |
| Numérico | N / D | Contiene números comprendidos en el intervalo de -10 a 10 , con un máximo de 15 cifras. |
| Memo con formato | de 0 a 240 archivos | Los campos memo pueden contener letras, números, símbolos especiales o cualquier otro carácter imprimible, así como saltos de línea, tabuladores, justificación y demás caracteres de impresión. |
| Gráfico | de 0 a 240 archivos | Contienen imágenes. Estas pueden crearse mediante alguna aplicación de dibujo o de diseño. |

A través de los informes se pueden imprimir datos y figuras de una base de datos o como resultado del enlace de diversas tablas. Las consultas se refieren únicamente a la visualización en pantalla de dichos informes de un equipo en particular.

### 4.1.1.3 Diseño.

### 4.1.1.3.1 Diseño Detallado.

El diseño detallado se refiere a detalles de cómo empacar módulos de procesamiento y cómo instrumentar los algoritmos, las estructuras de datos y sus interconexiones.

### a) Entradas y salidas del sistema.

Una vez que ya se mencionaron los conceptos de Hardware y Software requeridos para el manejo adecuado del Sistema de Bases de Datos, es importante definir los datos de entrada y salida.

La única información de entrada que se requiere para utilizar el sistema es la siguiente:

- Elegir el tema de interés
- Tipo de equipo que se desea visualizar
- Información sus características, usos, etc.
- Impresión de dicha información

Los datos que reporta el sistema como salida son:

- La figura del equipo elegido
- El nombre completo
- Descripción

- Características
- Usos
- Materiales de construcción

### b) Diagramas de Flujo.

Para comprender más a fondo el funcionamiento del Sistema de Bases de Datos se hace indispensable la descripción del seudocódigo y el diagrama de flujo.

El Seudocódigo del Sistema es el siguiente:

```
Inicio del Sistema de Bases de Datos
Desea información del tema de Bombas ó Válvulas
Si es Bombas
        Qué tipo específico de Bombas desea
        Seleccione el tipo
        Visualización del equipo y sus características
        Desea imprimir el equipo y sus características
        Si es si
                Imprime
        Si es no
                Desea visualizar otro tipo de Bombas
        Si es si
                Aparece el equipo en pantalla
        Si es no
                Desea salir de la aplicación y regresar al menú principal
                Si es si
                        Regresa al menú
Si es Válvulas
Qué tipo específico de Válvulas desea
Seleccione el tipo
Visualización del equipo y sus características
Desea imprimir el equipo y sus características
Si es si
        Imprime
Si es no
        Desea visualizar otro tipo de Válvula
        Si es si
                Aparece el equipo en pantalla
        Si es no
                Desea salir de la aplicación y regresar al menú principal
                Si es si
                        Regresa al menú
Desea salir del sistema de bases de datos
Si es si
Sale del Sistema
Si es no
Puede elegir visualizar Válvulas ó Bombas
```

#### 4.1.1.3.1 Diseño Estructural.

El diseño estructurado de este sistema fue desarrollado como una técnica, el enfoque básico de este el la conversión sistemática del diagrama de flujo de datos. Y se muestra a continuación:

El sistema tiene dos módulos principales :

1. Modulo de Válvulas.
2. Modulo de Bombas.

Siendo cada uno independiente entre sí. Detallándose en el Manual de Usuario, las características, tipos y usos de cada modulo.

### 4.1.1.4 Instrumentación.

Como se menciono anteriormente, la instrumentación del sistema fue desarrollada en Paradox para Windows.

En el **Anexo II** se muestra la codificación del sistema.

### 4.1.1.3 Pruebas.

Al sistema se le aplicaron las siguientes pruebas:

#### 4.1.1.3.1 Análisis Estático.

- Se vio que no existieran números incorrectos de argumentos en una llamada de rutina.
- Que no existieran segmentos de código inalcanzables.
- Modificación de argumentos ficticios dentro del subprograma.
- Referencia a funciones en expresiones ejecutadas condicionalmente.
- Que no existieran identificadores referidos pero no fijados o viceversa.
- Que no existieran identificadores declarados por omisión.
- Listado de rutinas llamada por cada rutina.
- Listados de rutinas que llaman a cada rutina.

#### 4.1.1.3.2 Análisis Dinámico.

- Trazado de mensajes para cada proposición ejecutable.
- Impresión del nombre de rutina de la entrada y salida de la misma.
- Puntos de ruptura, examen de estado, ejecución de un paso.

#### 4.1.1.3.3 Optimización.

- Tiempo total, porcentaje de tiempo y tiempo promedio dedicados a cada rutina.
- Tiempo incluido y excluido en rutinas llamadas.
- Comparación del tiempo promedio de una sola corrida, con el tiempo promedio de corridas medias.

#### 4.1.1.3.4 Depuración.

La depuración procedió como sigue :

- Listar las posibles causas de la falla observada.
- Usar la información disponible para eliminar varias hipótesis.
- Elaborar las restantes hipótesis.
- Probar o rechazar cada hipótesis.
- Determinar las correcciones apropiadas.
- Verificar las correcciones.

## 4.2 DESCRIPCION DEL SISTEMA

### 4.2.1 PARTE I

#### 4.2.1.1 Introducción

La guía de usuario constituye una importante ayuda para el uso de "Bombas, compresores y válvulas". Esta guía contiene información acerca del entorno del sistema, y de la manera de utilizar sus diferentes módulos.

Si es la primera vez que va a utilizar este sistema , es aconsejable que lea con cuidado este manual, en el que se describen los conceptos básicos para utilizarlo.

Antes de proceder a utilizar el sistema y con el objeto de conseguir mejores resultados, es conveniente que el usuario este familiarizado con Windows. No obstante, si precisa más información acerca de Windows, consulte la documentación que acompaña al sistema.

##### 4.2.1.1.1 Conceptos Básicos

Este capítulo trata de los términos y conceptos fundamentales que habrá de utilizar al trabajar en el sistema .

En primer lugar presenta las conceptos de módulos, base de datos, base de datos temporales, datos, campos, menúes, mensajes, teclas rápidas, botones, puntero, ayudas, reportes.

Datos.
Información que contienen las bases de datos.

Campos.
Columna de información en una base de datos. El conjuntos de todos los campos relacionados componen un registro. El sistema usa los siguientes campos :

| Tipo de campo | Símbolo | Valores de tamaño |
|---|---|---|
| Alfanumérico | A | 1 - 255 (necesario) |
| Memo (formato) | F | 1 - 240 (optativo) |
| Gráfico | G | 1 - 240 (optativo) |

Módulos.
Son las secciones en que esta dividido el sistema, siendo cada uno independiente entre sí, Cada módulo está conformado de sus propios elementos, tales como, base de datos, menúes, teclas rápidas, botones, etc.

Base de datos.
La expresión de base de datos no se refiere a todos los tipos de registro, sino a una colección limitada y específica de estos. La base de datos puede definirse también como una relación de datos interrelacionados, almacenados en conjuntos sin redundancias perjudiciales o innecesarias, su finalidad es la de servir a una aplicación o más, de la mejor manera posible.

Menús.
Presentación de las opciones y elementos disponibles.

Mensajes.
Expresión que aparece en una ventana de mensajes o en la barra de estado del escritorio. Instrucciones que aparecen en pantallas muestran al usuario los pasos que debe seguir durante una operación.

Teclas rápidas.
Son las teclas que se tienen que presionar para realizar una serie de operaciones dentro del sistema y que son equivalentes a las acciones de desplazamiento y selección que puede ejecutar el ratón.

Botones.
Objeto gráfico a través del cual el usuario tiene acceso directo a ciertas funciones o rutinas que están señaladas en el mismo.

Puntero.
Un símbolo visual que indica la posición del ratón en pantalla.

Ayudas.
Sistema de ayuda en línea, que presenta información sobre la operación que se está realizando, o de algún tópico en particular.

Reportes.
Información de las bases de datos que pueden imprimirse en papel, mediante una presentación preliminar.

## 4.2.2 PARTE II

### 4.2.2.1 ¿Qué es ?

El sistema "Bombas, compresores y válvulas", el cual utilizando la tecnología actual de desarrollo de software, presenta características innovadoras como la presentación de fotos, que se almacenan en una base de datos, con lo cual el alumno podrá ver y consultar diferentes tipos de bombas, compresores y válvulas con solo oprimir un botón.

### 4.2.2.2 Módulos

El sistema esta dividido en tres módulos :

1 . Principal. Es el módulo de acceso con el cual están interrelacionados con los otros dos.

2. Bombas y Compresores. En este módulo están contenidos las base de datos, los reportes, y ayudas para obtener información de los diferentes tipos de equipos.

3. Válvulas. En este módulo están contenidos las base de datos, los reportes y ayudas para obtener información de los diferentes tipos de equipos.

#### 4.2.2.2.1 Módulo Principal

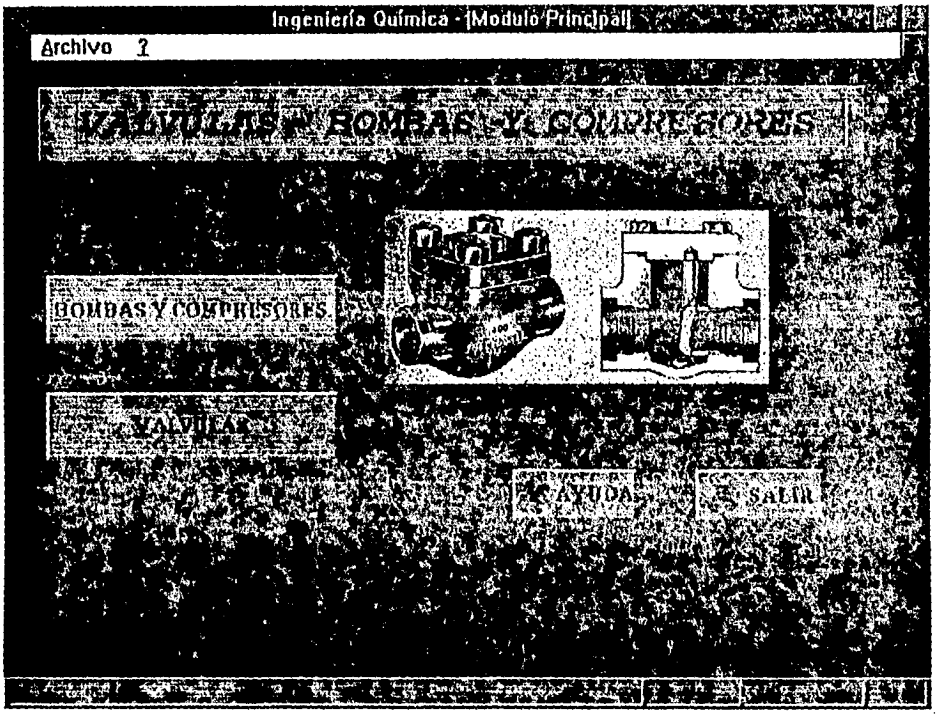
##### 4.2.2.2.1.1 Objetivo

Es un módulo de presentación, que permite elegir o seleccionar el módulo de Bombas y Compresores, o el módulo de válvulas. Además , existe un menú especial, donde se puede consultar la tesis dinámicamente.

##### 4.2.2.2.1.2 Iniciando

Para iniciar el sistema ubíquese con el puntero sobre el icono correspondiente a la aplicación y pulse dos veces con el botón derecho del ratón.

A continuación verá la siguiente pantalla :




La cual se divide en las siguientes partes:

Menús .
Esta formado por :

*Archivo* a través del cual se tiene acceso al submenu de :
- *Bombas y Compresores.* Accesa al módulo de Bombas y Compresores.
- *Válvulas.* Accesa al modulo de Válvulas.
- *Salir.* Sale de la aplicación y regresa a Windows.

*?* a través del cual se tiene acceso al submenu de :
- *Tesis.* Permite ver el contenido de la tesis.
- *Sistema.* Proporciona información para el manejo adecuado del sistema.
- *Información Técnica.* Información detallada de los equipos que conforman el sistema.

Botones.
Este módulo tiene los siguientes botones :
- *Bombas y Compresores.* Accesa al módulo de Bombas y Compresores.
- *Válvulas.* Accesa al módulo de Válvulas.
- *Salir.* Sale de la aplicación y regresa a Windows.

- *Ayuda*. Proporciona información para el manejo adecuado del sistema.

Teclas rápidas.
Este módulo tiene las siguientes teclas rápidas:

- *Alt + A*. Abre el menú de Archivo.
- *Alt + ?*. Abre el menú de ayuda.
- *Alt + B*. Llama al módulo de Bombas y Compresores.
- *Alt + V*. Llama al módulo de Válvulas.
- *Alt + S*. Sale de la aplicación y regresa a Windows.
- *Alt + Y*. Proporciona información para el manejo adecuado del sistema.
- *F9*. Sale de la aplicación y regresa a Windows.
- *F1*. Proporciona información para el manejo adecuado del sistema.

Mensajes.
Al ejecutar cualquier método de salida del sistema aparecerá un mensaje, para confirmar dicha acción.

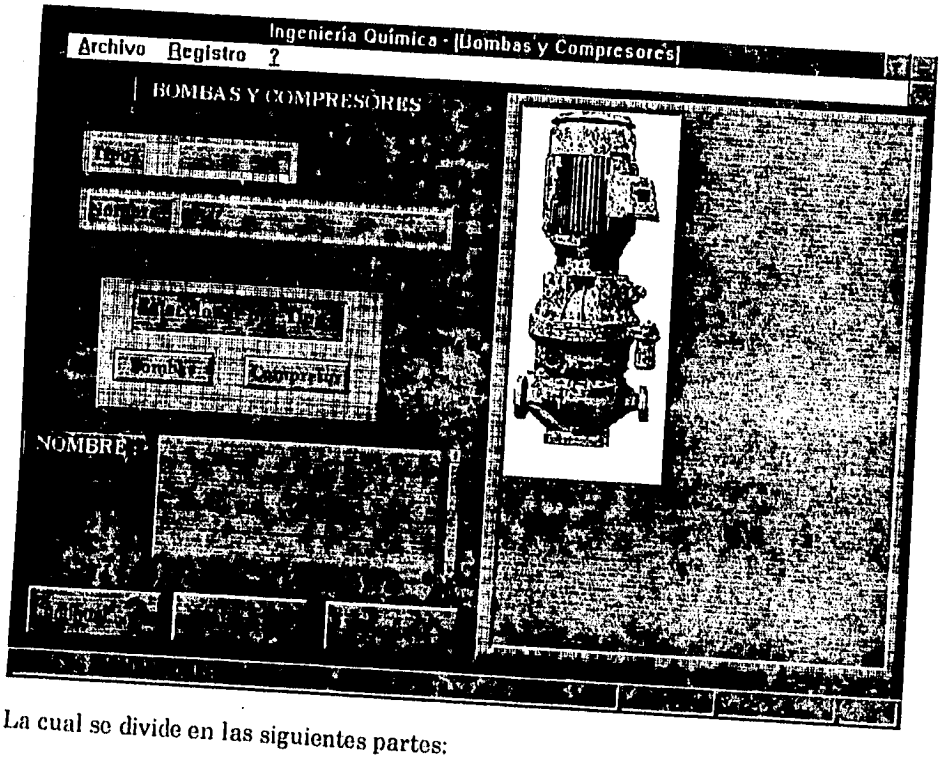
#### 4.2.2.2.2 Módulo Bombas y Compresores

##### 4.2.2.2.2.1 Objetivo

Ver los tipos, características, usos, especificaciones, así como su fotografía. Además se puede imprimir la información del equipo seleccionado.

##### 4.2.2.2.2.2 Iniciando

Una vez ya dentro del módulo verá la siguiente pantalla :




La cual se divide en las siguientes partes:

Menús .
Está formado por :

*Archivo* a través del cual se tiene acceso al submenu de :
- *Imprimir.* Imprime la información del equipo que se visualiza en pantalla.
- *Impresora.* Especifica y selecciona el tipo de impresora deseado.
- *Salir.* Sale del submodulo y regresa al modulo principal.

*Registro* a través del cual se tiene acceso al submenu de :
- *Primero.* Se desplaza al primer registro de la base de datos.
- *Ultimo.* Se desplaza al último registro de la base de datos.
- *Anterior.* Se desplaza un registro anterior en la base de datos.
- *Siguiente.* Se desplaza un registro hacia adelante en la base de datos.

*?* a través del cual se tiene acceso al submenu de :
- *Indice.* Proporciona información para el manejo adecuado del sistema.
- *Uso de la ayuda.* Explica como utilizar la ayuda en linea.
- *Información Técnica.* Información detallada de los equipos que conforman el sistema.

Botones.
Este módulo tiene los siguientes botones :

- *Bombas.* Busca todas las bombas disponibles en la base de datos, para que elija cualquiera de ellas.
- Compresores. Busca todos los compresores disponibles en la base de datos, para que elija cualquiera de ellos.
- Continuar. Del equipo seleccionado muestra información adicional.
- *Salir.* Sale de la aplicación y regresa a Windows.
- *Ayuda.* Proporciona información para el manejo adecuado del sistema.

Teclas rápidas.
Este módulo tiene las siguientes teclas rápidas:

- *Alt + A.* Abre el menú de Archivo.
- *Alt + ?.* Abre el menú de ayuda.
- *Alt + R.* Abre el menú de registro.
- *Alt + C.* Del equipo seleccionado muestra información adicional.
- *Alt + S.* Sale de la aplicación y regresa a Windows.
- *Alt + Y.* Proporciona información para el manejo adecuado del sistema.
- *Ctrl + F11.* Se desplaza al primer registro de la base de datos.
- *Ctrl + F12.* Se desplaza al último registro de la base de datos.
- *F11.* Se desplaza un registro anterior en la base de datos.
- *F12.* Se desplaza un registro hacia adelante en la base de datos.
- *F9.* Sale de la aplicación y regresa a Windows.
- *F1.* Proporciona información para el manejo adecuado del sistema.

Mensajes.
Al ejecutar cualquier método de salida del sistema aparecerá un mensaje, para confirmar dicha acción.
Al imprimir, aparecerá un mensaje de "¿Quiere imprimir ahora?", en este punto deberá tener la impresora lista.

**Procedimiento.**

1. Entre al módulo de Bombas y Compresores.
2. Con las teclas rápidas *F11, F12, Ctrl+F11* o *Ctrl+F12* puede ver todos los equipos en la base de datos.
3. Si desea ver bombas o compresores exclusivamente, seleccione el botón de Bombas o Compresores.
4. En la parte inferior a los botones de Bombas y Compresores podrá seleccionar el equipo específico que quiera consultar. Pulsando dos veces el botón derecho del ratón.
5. Pulse en el campo de nombre donde aparece el nombre del equipo seleccionado previamente.
6. Posteriormente, oprima el botón Continuar y visualizará más información adicional.
7. Una vez hecho esto, podrá imprimir la información del equipo. Antes de esto asegúrese, que tiene instalada la impresora correcta.

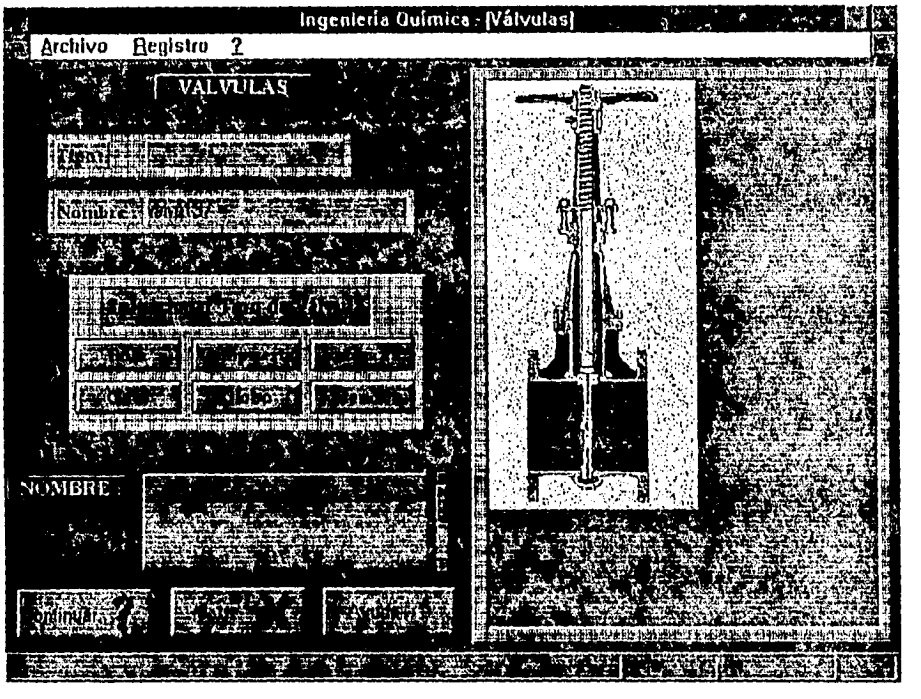
### 4.2.2.2.3 Modulo Válvulas

#### 4.2.2.2.3.1 Objetivo

Ver los tipos, características, usos, especificaciones, así como su fotografía. Además se puede imprimir la información del equipo seleccionado.

#### 4.2.2.2.3.2 Iniciando

Una vez ya dentro del módulo verá la siguiente pantalla :




La cual se divide en las siguientes partes:

Menús .
Está formado por :

*Archivo* a través del cual se tiene acceso al submenu de :
- *Imprimir.* Imprime la información del equipo que se visualiza en pantalla.
- *Impresora.* Especifica y selecciona el tipo de impresora deseado.
- *Salir.* Sale del submodulo y regresa al módulo principal.

*Registro* a través del cual se tiene acceso al submenu de :
- *Primero.* Se desplaza al primer registro de la base de datos.

- *Ultimo.* Se desplaza al último registro de la base de datos.
- *Anterior.* Se desplaza un registro anterior en la base de datos.
- *Siguiente.* Se desplaza un registro hacia adelante en la base de datos.

*?* a través del cual se tiene acceso al submenu de :
- *Indice.* Proporciona información para el manejo adecuado del sistema.
- *Uso de la ayuda.* Explica como utilizar la ayuda en linea.
- *Información Técnica.* Información detallada de los equipos que conforman el sistema.

Botones.
Este módulo tiene los siguientes botones :
- *Bola.* Busca y presenta en pantalla todas las válvulas de bola existentes en la base de datos.
- *Compuerta.* Busca y presenta en pantalla todas las válvulas de compuerta existentes en la base de datos.
- *Esféricas.* Busca y presenta en pantalla todas las válvulas esféricas existentes en la base de datos.
- *Gato.* Busca y presenta en pantalla todas las válvulas de gato existentes en la base de datos.
- *Globo.* Busca y presenta en pantalla todas las válvulas de globo existentes en la base de datos.
- *Retención.* Busca y presenta en pantalla todas las válvulas de retención existentes en la base de datos.
- *Continuar.* Del equipo seleccionado muestra información adicional.
- *Salir.* Sale de la aplicación y regresa a Windows.
- *Ayuda.* Proporciona información para el manejo adecuado del sistema.

Teclas rápidas.
Este módulo tiene las siguientes teclas rápidas:
- *Alt + A.* Abre el menú de Archivo.
- *Alt + ?.* Abre el menú de ayuda.
- *Alt + R.* Abre el menú de registro.
- *Alt + C.* Del equipo seleccionado muestra información adicional.
- *Alt + S.* Sale de la aplicación y regresa a Windows.
- *Alt + Y.* Proporciona información para el manejo adecuado del sistema.
- *Ctrl + F11.* Se desplaza al primer registro de la base de datos.
- *Ctrl + F12.* Se desplaza al último registro de la base de datos.
- *F11.* Se desplaza un registro anterior en la base de datos.
- *F12.* Se desplaza un registro hacia adelante en la base de datos.
- *F9.* Sale de la aplicación y regresa a Windows.
- *F1.* Proporciona información para el manejo adecuado del sistema.

Mensajes.
Al ejecutar cualquier método de salida del sistema aparecerá un mensaje, para confirmar dicha acción.
Al imprimir, aparecerá un mensaje de "¿Quiere imprimir ahora?", en este punto deberá tener la impresora lista.

**Procedimiento.**

1. Entre al módulo de Válvulas.
2. Con las teclas rápidas *F11*, *F12*, *Ctrl+F11* o *Ctrl+F12* puede ver todos los equipos en la base de datos.
3. Si desea algún tipo de válvula, selecciónela entre las diferentes opciones.
4. En la parte inferior a los botones de válvulas podrá seleccionar el equipo específico que quiera consultar. Pulsando dos veces el botón derecho del ratón.
5. Pulse en el campo de nombre donde aparece el nombre del equipo seleccionado previamente.
6. Posteriormente, oprima el botón Continuar y visualizará más información adicional.
7. Una vez hecho esto, podrá imprimir la información del equipo. Antes de esto asegúrese, que tiene instalada la impresora correcta.

## 5. CONCLUSIONES.

El grado de avanze tecnologico en la computación actualmente es muy alto, y la Facutad de Química, explicitamente el Laboratorio de Ingeniería Química no debe quedarse rezagado.

El Laboratorio de Ingeniería Química al utilizar el sistema "Bombas, compresores y válvulas" obtendrá las siguientes ventajas:

1. Es practicamente imposible,ya sea por espacio o por recursos, tener todos los equipos de Bombas, compresores y válvulas que se necesitan para que el alumno pueda conocerlos y aprender de ellos, el sistema permite estar actualizado con la información más reciente de los equipos.

2. El sistema puede almacenar un numero muy grande de información de diferentes equipos, y podrá consultarse de manera inmediata.

3. Un programa tan amigable permite desarrollar en el alumno de Ingeniería Química, la inquietud, ya sea por desarrollar o solo el gusto por conocer los diferentes equipos, o por que no crear un sistema para otros equipos y laboratorios.

4. La inversión inicial es muy pequeña, ya que solo se requiere de un equipo de computo necesario, y gente que sea capaz de continuar con esta labor de información tecnológica.

Cabe hacer mención que el sistema no pretende enseñar como diseñar y utilizar las bombas, compresores y válvulas, solo es una semilla para que futuras generaciones desarrollen y perfeccionen programas que tengan calidad.

El laboratorio debe tener las herramientas adecuadas para que futuras generaciones pueden competir con exito en su vida profesional, y una preparación adecuada da al alumno la confianza y seguridad que necesitan para poder obtener mejor oportunidades.

# TESIS SIN PAGINACION

# COMPLETA LA INFORMACION

# ANEXO I

## Información teórica de "Bombas, Compresores y Válvulas"

## CONTENIDO

## 1. BOMBAS COMPRESORES Y VALVULAS.

La bomba es la forma de máquina más remota que sustituyó la energía natural por el esfuerzo muscular, para satisfacer las necesidades del hombre.

Las primeras bombas de las que se tiene conocimiento, son conocidas de diversas formas, dependiendo de la manera en que se registró su descripción, como las ruedas persas, ruedas de agua o norias. Todos estos dispositivos son ruedas bajo el agua que contenían cubetas que se llenaban con agua cuando se sumergían en una corriente y que automáticamente se vaciaban en un colector a medida que se llevaban al punto más alto de la rueda en movimiento.

La más conocida de aquellas bombas, el tornillo de Arquimedes, aún persiste en los tiempos modernos. Todavía se manufactura para aplicaciones de baja carga, en donde el líquido se carga con basura u otros sólidos.

La tendencia en la industria de procesos químicos es construir plantas cada vez más grandes con equipo de un solo componente, más grande y confiable.

La confiabilidad del equipo rotatorio siempre se debe definir en términos de la duración esperada de la planta y el tiempo de amortización requerido para producir utilidades al propietario.

El corazón de muchos procesos y el que más problemas puede ocasionar es el compresor. Cuando se selecciona un tipo de compresor, es indispensable contar con todas las condiciones del proceso para su examen.

En cuanto a las válvulas podemos decir que en un proceso, se emplean un gran número de válvulas, de tamaños y formas muy diferentes. A pesar, de la amplia variedad de diseños, todas tienen un fin principal que es común: disminuir o detener el flujo de un fluido.

### 1.1 Bombas

Los fluidos se mueven a través de tuberías, aparatos o a la atmósfera, por medio de bombas, ventiladores, soplantes y compresores. Estos dispositivos aumentan la energía mecánica del fluido. El aumento de energía puede emplearse para incrementar la velocidad, la presión o la altura del fluido.

Una gran variedad de tipos y tamaños de bombas han sido desarrolladas para satisfacer las muchas condiciones especiales necesarias en sistemas de plantas químicas.

#### 1.1.1 Concepto

La función primaria de las bombas no es la transportación, si no la adición de energía al fluido. Este incremento de energía es usado para aumentar la presión, la velocidad o elevar un fluido.

Las características de una bomba que debemos conocer son:

- Capacidad. Es la cantidad de fluido descargado por unidad de tiempo.
- Incremento en presión. Definido para las bombas como carga, que es la energía dada al fluido por unidad de peso y se obtiene dividiendo el incremento en presión entre el peso específico del fluido.
- Potencia: La energía consumida por la máquina por unidad de tiempo.
- Eficiencia: La energía suministrada al fluido entre la energía suministrada a la máquina.

### 1.1.2 Clasificación

Las bombas pueden clasificarse en base de las aplicaciones a que están destinadas, los materiales con que se construyen, los líquidos que mueven y aún su orientación en el espacio.

Todas las bombas pueden dividirse en dos grandes categorías:

- Dinámicas: En las cuales se añade energía continuamente, para incrementar las velocidades de los fluidos dentro de la máquina a valores mayores de los que existen en la descarga, de manera que la subsecuente reducción en velocidad dentro, o más allá de la bomba, produce un incremento en la presión.
- Desplazamiento Positivo: En las cuales se agrega energía periódicamente mediante la aplicación de fuerza, lo que resulta en un incremento directo en presión, hasta el valor requerido para desplazar el fluido a través de válvulas ó aberturas en la línea de descarga.

Las bombas de desplazamiento positivo, a su vez se dividen en:

- Reciprocantes
- Rotatorias

dependiendo de la naturaleza del movimiento de los miembros que producen la presión.

Y las bombas dinámicas se dividen en:

- Centrífugas

Estas bombas dividen en:

- Bombas centrífugas de flujo radial: En estas bombas el líquido entra por el centro del impulsor y ocurre radialmente a la periferia.
- Bombas centrífugas de flujo mixto: En estas bombas el líquido entra axialmente y descarga en dos direcciones axial y radial.
- Bombas centrífugas de flujo axial: En estas bombas el líquido entra axialmente y descarga axialmente.

Las bombas centrífugas se subdividen en tipos de acuerdo al número de pasos:

- Bombas de un paso: Una bomba en la cual la carga total está desarrollada por un impulsor.
- Bombas de paso múltiple: Una bomba que tiene dos ó más impulsores actuando en serie en una misma flecha en la carcaza.

**Tabla 1.1 Clasificación de las bombas**

| Tipo de bomba | Clase | Tipo | |
|---|---|---|---|
| Reciprocante | Pistón | Doble acción | Simple |
| | | | Doble |
| | Embolo | Simple acción | Simple, doble |
| | | Doble acción | Triple, múltiple |
| | Diafragma | Simple | |
| | | Múltiple | |
| Rotatorias | Rotor Simple | Aspas | |
| | | Pistón | |
| | | Miembro flexible | |
| | | Tornillo | |
| | Rotor Múltiple | Engranes | |
| | | Lóbulos | |
| | | Balancines | |
| | | Tornillos | |
| Centrífugas | Flujo Radial | Succión simple | Autocebantes |
| | | | Cebadas |
| | Flujo Mixto | Doble succión | Un paso |
| | | | Multipaso |
| | Flujo Axial | Simple succión | Un paso |
| | | | Multipaso |

### 1.1.3 Bombas Centrífugas

La bomba centrífuga desarrolla su presión por fuerza centrífuga en el paso del líquido a través de la bomba y es generalmente aplicable a alta capacidad, baja a media instalaciones de cabecera.

Las partes básicas de una bomba centrífuga son:

**Tabla 1.2 Partes básicas de una bomba centrífuga.**

| Parte | Propósito |
|---|---|
| Propulsor | Imparte velocidad al líquido. |
| Cubierta | Da dirección al flujo del propulsor y convierte esta energía de la velocidad en energía de presión. |
| Eje | Transmite potencia de la polea impulsora al propulsor. |
| Caja empaquetada | Este es un instrumento de estrangulamiento de fuga que podría ocurrir al punto de entrada del eje en la cubierta. |
| a) Empaque | Este es el más común instrumento de estrangulamiento de fuga entre la parte interior y exterior de la cubierta. |
| b) Cuello | Para posición y ajuste de la presión de empaque. |
| c) Sellos | Proporcionan Pasaje para distribuir el medio de sellamiento uniformemente alrededor de la parte de la cubierta que pasa a través de la caja empaquetada. |
| d) Sello mecánico | Proporciona un arreglo de sello mecánico que tiene lugar de empaquetamiento. Básicamente este tiene una superficie rotatoria con la cubierta y la cara estacionaria. El minúsculo orificio entre las dos caras evita la fuga de líquido fuera y la entrada de aire. |
| Cubierta de camisa | Protege la cubierta donde este pasa a través de la caja de empaquetamiento. Usualmente usada en bombas con empaque, pero frecuentemente eliminada si los sellos mecánicos son empleados. |
| Anillos de desgaste | Guardan recirculación interna bajo un mínimo. En tipos pequeños únicamente un anillo es usado en la cubierta y en tamaños grandes, arreglos de anillos son usados en la cubierta y en el propulsor. |
| Platos de desgaste | Con propulsores de tipo abierto ó cerrado ejecutan el mismo propósito como los anillos de desgaste con claros radiales. |
| Conexiones | Exactamente localizan la cubierta e impulsan las entradas radial y de empuje. |
| Armazón | Para amontonar la unidad rígidamente y soportar las conexiones. En la mayoría de las bombas de succión simple esta es una pieza separada. En muchas bombas de succión dobles, el soporte es a través de la pieza fundida como parte de la cubierta. |
| Acoplador | Conecta la bomba a la polea impulsora |

Los tres tipos comunes de *propulsores* que imparten la principal energía al líquido para aplicaciones de proceso son:

- Completamente cerrados.-Usados para alta cabeza y aplicaciones de alta presión.
- Semicerrados. Usados para aplicaciones de propósitos generales, tienen tipos de aspas abiertas a la entrada para romper las partículas suspendidas y evitar estancamientos.
- Abiertos. Usados para bajas cabezas, aplicaciones de sólidos suspendidos, muy pequeños flujos.

Las aspas radiales se encuentran en la parte trasera del anillo de refuerzo ó plato del propulsor para reducir la presión en la caja de empaquetamiento, y evitar los sólidos suspendidos de la entrada del lado trasero y que posiblemente cause estancamiento.

Las aspas de bombeo son retrolavadas en forma relativa a la rotación del propulsor.

La *cubierta* puede ser construida de un amplia variedad e metales, tan bien con corresponda al material del propulsor. Opera a presiones alrededor de los 5000 psi. Sin embargo, la aplicación del proceso usual está en el intervalo de 75 psi a 899 psi.

La remoción de las partes de la cubierta es necesario para acceso al propulsor y frecuentemente a los empaques y sellos.

Se debe tener cuidado en la selección del material del *eje*. Este debe ser resistente a la corrosión de los fluidos del proceso. Para algunos diseños es preferible usar cubiertas de camisa de material resistente a la corrosión sobre el material estructural de la cubierta. Estas camisas pueden ser metales, cerámica, neopreno, etc.

Las *conexiones* deben ser adecuadas para el manejo de las entradas de cubierta con uso excesivo, proveyendo lubricación.

Los *empaques* metálicos ó suaves en una cubierta de empaquetamiento son satisfactorios para sistemas de fluidos no corrosivos o baja presión. Los empaques especiales tales como teflón ó sellos mecánicos son comúnmente usados para fluidos corrosivos.

El *sello* mecánico simple está hecho de un elemento rotatorio fijo a la cubierta y un elemento estacionario fijo a la bomba.

El doble sello es para problemas de sellamiento severos, donde la fuga de salida al ambiente no puede ser tolerada y debe ser controlada.

**Tabla 1.3 Requerimientos para instalaciones de sellos mecánicos.**

| Función | Descripción | Observaciones |
|---|---|---|
| Enfriamiento | Enchaquetamiento de agua<br>Caja de empaquetamiento | El líquido debe quedarse en la caja de empaquetamiento |
| Enfriamiento | Plato | Eficiente para enfriar el contacto con las caras. |
| Lubricación | Final | Buena bajo vacío, seca sellos. |
| Lubricación | Circulación | Buen enfriamiento de caras de contacto. |
| Flushing | Sello interno | Bueno para líquidos volátiles, soluciones que tienden a cristalizar, vapor. |
| Flushing | Sello externo | Calentamiento para evitar solidificación. |
| Extinción | Unicamente sellos externos | Para líquidos corrosivos y oxidantes, para altas temperaturas. |
| Venteo y drenaje | Sello interno | Función de seguridad, para venteo a flare, drenaje. |

Las bombas centrífugas tienen las siguientes características:

- Amplia capacidad, presión e intervalo de fluidos característicos.
- Fácilmente adaptado a motor directo, V-belt u otro motor.
- Relativamente pequeños requerimientos de área.
- Bajo costo.
- Dificultad para obtener flujos muy pequeños a moderados a altas presiones.
- Desarrolla condiciones turbulentas en fluidos.
- Tipo turbina:
    a) Ofrece muy alta cabeza a flujos pequeños
    b) Arrastre de agua con el vapor por si mismas
    c) Limitado para fluidos no abrasivos, limpios con propiedades físicas limitadas
    d) El espacio libre pueden ser un problema en el ensamble y mantenimiento.

Bombas en serie.

Algunas veces es ventajoso ó económico usar dos ó más bombas en serie para alcanzar la presión de descarga deseada. En este caso la capacidad está limitada por la capacidad más pequeña de alguna de las bombas (si ellas son diferentes) a su velocidad de operación. La presión de descarga total de la última bomba es la suma de las presiones de descarga individuales de las bombas. Para bombas idénticas, la capacidad es la de una bombá, y la presión de descarga de la última bomba es la suma de las cabezas individuales de cada bomba actuando como una unidad simple.

Bombas en paralelo.

Las bombas son operadas en paralelo para dividir la carga entre dos ó mas bombas más pequeñas que una simple más larga, o para proveer capacidad adicional en un sistema.

**Tabla 1.4 Tipo de selección basada en el manejo de líquidos.**

| Líquido | Tipo de bomba básico | Tipo de propulsor |
|---|---|---|
| Agua y otros líquidos claros no corrosivos a temperaturas moderadas ó frías. | Simple ó doble succión. | Cerrado, excepto para muy pequeñas capacidades |
| Agua por abajo de 250 °F | Simple ó doble succión. | Cerrado, excepto para muy pequeñas capacidades. |
| Hidrocarburos, calientes | Simple succión | Cerrado con entradas largas. |
| Corrosivos: | | |
| Medianamente ácidos ó alcalinos | Simple ó doble succión. | |
| Fuertemente ácidos ó alcalinos | Simple ó doble succión con succión simple, probablemente menos caro si es disponible para el intervalo. | Cerrado, excepto para muy pequeñas capacidades ó donde el líquido tiene a formar escala en superficies de partes movibles. |
| Corrosivos calientes | Simple succión. | |

| | | |
|---|---|---|
| Agua con sólidos en suspensión: | | |
| Abrasivos finos | Si todas las partículas pasan a través de una pantalla o malla 1/8", bombas lineales de neopreno son disponibles, lo cual dará muchas veces la vida de bombas de metal, proveyendo acción no química ó excesiva temperatura que deteriorará el neopreno. Especiales compuestos de neopreno pueden ser aplicados para proporcionar resistencia a ciertos químicos. | Abierto, que permite mejor aplicación del neopreno, excepto en tamaños grandes. También hecho en tipos cerrados. |
| Abrasivos gruesos | Simple succión. No disponible para amplios rangos de operación, esto es, pequeñas capacidades no tan fácilmente obtenidas. | Cerrado. |
| Sólidos pulposos | Simple succión. Doble succión únicamente usada en concentraciones de sólidos muy ligeras. | Cerrado. |

Las bombas centrífugas, presentan las siguientes ventajas:

- Son aparatos giratorios
- No tienen órganos articulados, y los mecanismos de acoplamiento son muy sencillos
- La impulsión del accionador es bastante sencilla
- El gasto puede ser constante, sin requerir regulador
- Se adaptan con facilidad a muchas circunstancias
- El costo es relativamente bajo
- El espacio requerido es pequeño, comparado con el de otras
- El peso es pequeño, y por lo tanto la cimentación también
- Su mantenimiento se reduce a renovar aceite de chumaceras, empaques del prensa
- estopas y el número de refacciones a cambiar es pequeño

Y las desventajas son:

- Baja eficiencia a gastos pequeños y grandes cargas
- Las bombas con pequeña descarga solo pueden manejar líquidos sin sólidos en suspensión
- Problemas de cebado

Las materiales que componen las partes de una bomba son:

Los factores principales que indican el material que debe usarse para construir la bomba, son las condiciones de servicio y la naturaleza del líquido a manejar.

Algunas de las condiciones de servicio que afectan la selección de los materiales, son las siguientes:

- Resistencia a la corrosión
- Acción electroquímica
- Abrasividad de sólidos en suspensión
- Temperatura de Bombeo
- Carga por paso (afecta tanto la velocidad periférica del impulsor como la velocidad del líquido en los conductos)
- Presión de operación
- Adaptabilidad del material para diseños estructurales particulares
- Factor de carga y duración esperada

Las *carcazas* pueden estar hechas de cualquiera de los siguientes materiales:

- Fierro fundido. Si las presiones no exceden 1100 psi ni las temperaturas exceden 350 °F como máximo.
- Acero Fundido ó Forjado: Si la presión de descarga ó la temperatura de bombeo impiden usar fierro fundido. A muy bajas temperaturas el acero fundido es más resistente a la tensión que el fierro fundido.
- Bronce. Si el líquido bombeado es medianamente corrosivo.
- Acero Inoxidable ( y otras aleaciones). Si el líquido bombeado es corrosivo ó excesivamente abrasivo.

Los *impulsores* de bronce se prefieren para manejar líquidos normales por las siguientes razones:

- Es más fácil de fundir
- Es más fácil de maquinar
- Proporciona superficies más lisas
- No se enmohece

Sin embargo, no deben usarse impulsores de bronce con carcazas de fierro fundido si el líquido manejado es un electrolito fuerte. Tales líquidos requieren materiales ferrosos.

Debido a que el bronce se expande aproximadamente 40% más que el acero, raramente se usa para accesorios de bombas si la temperatura del líquido bombeado excede 250°F.

Se usa bronce por las mismas razones que en los impulsores. También pueden usarse anillos de fierro fundido, acero fundido ó acero inoxidable, si la dureza u otras propiedades no se pueden tener con bronce.

Las *flechas*, están fabricadas, por lo general de acero de horno de reverbero. Si se piensan tener altos esfuerzos, se recomiendan aceros aleados de alta resistencia a la tensión. Con líquidos corrosivos se recomiendan acero inoxidable, bronce al fósforo ó metal monel.

Las *mangas de Flecha* son usualmente de bronce, si este no es satisfactorio por su baja resistencia a la abrasión, puede usarse acero inoxidable. El fierro fundido es poco usado.

Los *prensaestopas* son normalmente de bronce aunque también el fierro fundido ó el acero pueden usarse en bombas que tengan accesorios de fierro solamente.

La siguiente tabla incluye algunos de los servicios más comunes, el tipo de empaque recomendado y la temperatura máxima que soporta este.

**Tabla 1.5 Servicios más comunes y tipo de empaque recomendado.**

| Tipo de Servicio | Descripción de Empaque | Temperatura °C (máx.) |
|---|---|---|
| Agua, aceite, álcalis, ácidos débiles, servicio general. | Asbestos blancos trenzados en forma cuadrada, lubricante: aceite mineral y grafito. | 175 |
| Kerosina, gasolina y casi todos los solventes. | Asbestos blancos, trenzados en forma cuadrada, lubricante: resistente a solventes. | 175 |
| Aplicaciones a altas presiones y temperaturas. Para flechas que giran a altas velocidades. | Asbestos blancos trenzados en forma de listoncillos, lubricantes y grafito con alto punto de fusión en un 20% en peso. | 400 |
| Productos corrosivos exceptuando ácidos fuertes. | Asbestos blancos trenzados en forma de listoncillos. | 300 |
| Productos corrosivos incluyendo ácidos fuertes. | Igual que el anterior, exceptuando asbestos azules del Africa. | 300 |

Básicamente un *sello* mecánico está formado por dos superficies en contacto: una giratoria sobre la flecha, la hecha de metal y otra estacionaria sujeta al prensaestopa, hecha de grafito.

Existen dos tipos principales de sellos mecánicos:

- Desbalanceados
- Balanceados

Su diferencia estriba en la forma de las superficies de contacto. En el primer caso, hay un desbalanceo hidráulico debido a que la presión del prensaestopa se ejerce íntegra en toda la superficie del elemento rotatorio, en el segundo caso la presión del prensaestopa se ejerce en ambos sentidos de esta área, dando por resultado un balanceo hidráulico estático.

Ambos tipos existen en forma simple ó doble. El tipo simple, ya sea desbalanceado ó balanceado, se lubrica con el líquido bombeado. El tipo doble, desbalanceado ó balanceado, consta de dos sellos colocados dentro de una cámara de sello en donde se añade aceite como lubricante. La ventaja de un sello doble, es que no escapa el líquido bombeado a la atmósfera, sin embargo es mucho más caro y requiere más a tensión al instalarlo y operarlo.

El sello simple desbalanceado, está restringido a bajas presiones por parte del prensaestopa.

El sello simple balanceado se usa ampliamente en las refinerías e industrias químicas en general. Con la excepción de los empaques, todas las partes internas del sello están hechas de acero al cromo y grafito.

Para decidir si es más conveniente usar empaque ó sello mecánico, es necesario conocer las siguientes ventajas y desventajas:

**Tabla 1.6 Ventajas y desventajas de los empaques y sellos.**

| Sellos Mecánicos | Empaques |
|---|---|
| Ventajas: | Ventajas: |
| Sello positivo | Menos caros |
| Menos atención de mantenimiento | Amplia variedad en existencia |
| Limita la exposición a productos inflamables ó tóxicos | Puede reempacarse en el mismo lugar |
| | |
| Desventajas: | Desventajas: |
| Más caros | Mantenimiento inadecuado daña flecha y manga de flecha |
| Reparaciones más costosas | Es aconsejable atención frecuente |
| Reparación requiere desmantelamiento de la bomba | Cierto goteo, inevitable |
| Falla supone parar totalmente | Presenta problemas de toxicidad y fuego (según los líquidos) |
| Limitaciones de presión y temperatura | |
| Fuga considerable si falla | |

**Tabla 1.7 Materiales que componen la bomba estándar.**

| | |
|---|---|
| Carcaza | Fierro Fundido |
| Cubierta de Succión | Fierro Fundido |
| Prensaestopa | Bronce |
| Tuerca de Prensaestopa | Latón |
| Flecha | Acero al Carbón |
| Impulsor | Bronce |
| Anillo de desgaste de cubierta | Bronce |
| Manga de Flecha | Bronce |
| Deflector | Bronce |
| Tuerca del Impulsor | 11.5 - 13% Cromo |
| Roldana del Impulsor | 11.5 - 13% Cromo |
| Soporte Cubierta | Fierro Fundido |
| Cuña del Impulsor | Acero Inoxidable |
| Juntas | 2 mm. Durable |
| Pernaje | Acero de Bajo Carbón |
| Junta do la manga | Aluminio |
| Cople | Flexible |
| Base | Acero estructural |

**Tabla 1.β Selección de los materiales para la bomba de acuerdo al tipo de fluido bombeado.**

| Líquido | Condiciones del líquido | S. G. | Selección del Material |
|---|---|---|---|
| Aceite de Alquitrán de Hulla | ------ | ------ | B, C, 8, 9, 10, 11 |
| Acetona | ------ | 0.79 | B, C |
| Acido acético | Concentrado frío | 1.05 | 8, 9, 10, 11, 12 |
| Acido Clorhídrico | Concentrado | 1.19 (28%) | 11, 12 |
| Acido Fluorhídrico | Solución acuosa | ------ | A, 14 |
| Acido Nítrico | Conc. hirviendo | 1.50 | 6, 7, 10, 12 |
| Acido Sulfúrico | 77% Frío | 1.69 - 1.84 | C, 10, 11, 12 |
| Ac. Sulfúrico oleum | Fumante | 1.92 - 1.94 | 3, 10, 11 |
| Agua destilada | Alta pureza | 1.00 | A, 8 |
| Agua fresca | ------ | 1.00 | B |
| Alcoholes | ------ | ------ | A, B |
| Amoniaco acuoso | ------ | ------ | C |
| Gasolina | ------ | 0.68 - 0.75 | B, C |
| Hidróxido de Potasio (hidróxido de sodio) | Solución Acuosa | ------ | C, 5, 8, 9, 10, 11, 13, 14, 15 |
| Salmuera CaCl2 | pH 8 | ------ | C |
| Salmuera NaCl | menor de 3% fría | ------ | A, C, 13 |
| Salmuera, agua do mar | ------ | ------ | A, B, C |
| Carbono de sodio | Sol acuosa fría | ------ | C |

A: Bomba construida toda en bronce

B: Bomba con accesorios de bronce
C: Bomba construida toda en fierro fundido

### 1.1.4 Bombas Reciprocantes

Las bombas reciprocantes (alternativas) se utilizan en numerosas aplicaciones que exceden la capacidad de las bombas centrífugas ó rotatorias. Algunos servicios se podrían efectuar con una centrífuga ó rotatoria, pero a expensas de un aumento en los requisitos de potencia y de mantenimiento.

Una bomba reciprocante es de desplazamiento positivo, es decir, recibe un volumen fijo de líquido en condiciones casi de succión, lo comprime a la presión de descarga y lo expulsa por la boquilla de descarga. En estas bombas se logra por movimiento alternativo de un pistón, émbolo ó diafragma.

La bomba reciprocante no es cinética como la centrífuga y no requiere velocidad para producir presión, pues se pueden obtener presiones altas a bajas velocidades.

Las aplicaciones típicas de las bombas reciprocantes son:

- Carga de glicoles
- Carga de aminas
- Petróleo pobre
- Inyección de agua salada
- Eliminación de agua salada
- Evitadores de reventones
- Sistemas de oleoductos y gasoductos
- Sistema hidráulico
- Producción de fertilizantes
- Limpieza
- Tambores deshidratadores
- Pruebas hidrostáticas
- Pastas aguadas
- Dosificación
- Homogeneización

Desventajas de las bombas reciprocantes.

Las bombas reciprocantes tienen ciertas desventajas y la más común es el flujo a pulsaciones.

En la mayoría de las aplicaciones los costos inicial y de mantenimiento de las bombas reciprocantes serán mayores que para las centrífugas ó las rotatorias. La empaquetadura típica en una bomba de potencia dura menos de tres meses, o sea mucho menos que un sello mecánico en un eje rotatorio.

La bomba de acción directa tiene baja eficiencia térmica cuando se le impulsa con un gas como el vapor de agua. La eficiencia mecánica es alta, pero, debido a que no tiene ningún componente, como un volante, para almacenar energía, el gas motor debe

permanecer a la plena presión de entrada en el cilindro durante toda la carrera, al final de la carrera se expande el gas hacia el tubo de escape, pero no efectúa ningún trabajo durante la expansión.

Las bombas reciprocantes, por lo general, se clasifican por sus características:

- Extremo de impulsión, es decir, potencia ó acción directa.
- Orientación de la línea de centros del elemento de bombeo, es decir, horizontal ó vertical.
- Número de carreras de descarga por ciclo de cada biela, es decir, acción sencilla ó doble acción.
- Configuración del elemento de bombeo: pistón, émbolo ó diafragma.
- Número de varillas ó bielas de mando, es decir, simplex, duplex ó multiplex.

Las bombas reciprocantes no succionan líquidos, reducen solamente la presión en la cámara de succión y la presión externa empuja al líquido en la bomba.

La selección adecuada y mantenimiento de los empaques es un factor que afecta la eficiencia. Los empaques de asbesto impregnado con aceite y grafito son muy usados en los estoperos de vapor y líquido. Para altas temperaturas y cuando el fluido actúa como solvente del aceite, se usan empaques metálicos, estos empaques están disponibles con ó sin grafito.

El fluido pasa desde que se abren las válvulas de descarga hasta que se cierran cerca del final de la carrera, cuando el pistón pasa y regresa. Durante parte del ciclo de bombeo, el flujo es cero, sin embargo, el flujo desde la descarga puede ser constante dependiendo del diseño de la bomba. Las bombas de doble acción siempre tienen flujo en la línea de descarga.

La capacidad de flujo de una bomba reciprocante varía directamente con la velocidad. Diseños existentes tienen velocidades entre 20 y 200 carreras por minuto. Estas bombas pueden desarrollar muy altas presiones (10,000 psi y mayores) para muy bajas ó altas capacidades.

Las bombas reciprocantes son particularmente buenas para bombear fluidos viscosos, porque la alta velocidad de corte actuando sobre las paredes del cilindro sirve como empaque adicional. Este tipo de bombas tiene como desventajas el alto costo inicial, el tamaño y el mantenimiento costoso que debe dárseles. Otra desventaja adicional es que líquidos con sólidos abrasivos no pueden ser bombeados con una bomba reciprocante porque desgasta grandemente las superficies internas de la bomba.

- Bombas movidas por vapor. El flujo producido por estas bombas es constante hasta el final de la carrera, en donde el pistón del líquido (pistón que mueve al fluido manejado) se detiene y regresa.

  Las ventajas de estas bombas son su buen rendimiento en un intervalo amplio de condiciones de funcionamiento y su flexibilidad de capacidad, carga y velocidad. Las desventajas son su alto costo inicial, el grande espacio que ocupan, su ruidoso funcionamiento y la mayor atención que requieren.

Las bombas simplex de doble efecto son apropiadas para el servicio de abastecimiento de agua para alimentar las calderas de vapor. Las bombas duplex de doble efecto son mejores para bombear alquitranes, aceites y otros líquidos muy viscosos.

- Bombas Reciprocantes de Pistón. Este tipo de bomba confiere energía a un sistema de fluido por medio de un pistón actuando sobre una cantidad definida de fluido.

  El mencionado pistón puede ser movido por vapores ó mediante un motor eléctrico.

  Por cada carrera del pistón una cantidad definida y constante de fluido es descargada por la bomba.

  En una bomba reciprocante cuando el pistón entra en el cilindro (succión de líquido), descarga de fluido cesa, consecuentemente, el líquido es movido en pulsaciones.

  En una bomba de doble acción se usa el cilindro en ambos lados y el líquido manejado es casi el mismo para cada movimiento.

- Las bombas Reciprocantes de Embolo poseen las mismas cualidades, características y principios de funcionamiento que las bombas de pistón, difiriendo únicamente en que el émbolo es un aditamento cerrado que se mueve al través del cilindro en el cual están los empaques fijos.

- Bombas Reciprocantes de Diafragma. En este tipo de bombas un disco flexible que se mueve en el centro y su periferia es fija al cuerpo de la bomba.

  El diafragma puede ser de metal delgado, plástico flexible ó hule, y este diafragma puede conectarse al eje de un cigüeñal reciprocante. El diafragma es flexionado por aceite a presión de un cilindro movido por pistón, ó puede ser accionado por aire comprimido admitido y expulsado por un ciclo automático.

  Estas bombas se usan para gastos elevados de líquidos, ya sea claros ó conteniendo sólidos. También son apropiados para pulpas gruesas, drenajes, lodos y soluciones ácidas y alcalinas, así como mezclas de agua con sólidos que puedan ocasionar erosión.

- Bombas Reciprocantes de Proporcionado. Se trata de una bomba de pistón cuya transmisión está arreglada de tal forma que permite variar la carrera desde cero hasta su máxima carrera de diseño.

### 1.1.5 Bombas Rotatorias

Hay diferentes tipos de bombas rotatorias de desplazamiento positivo.

La mayoría de este tipo son capaces de manejar únicamente una solución limpia esencialmente libre de sólidos. Los diseños usan partes de plástico ó neopreno para que el dispositivo de presión puede manejar algunas partículas suspendidas. En

general, estas bombas manejan materiales de una amplia gama de viscosidad (por arriba de 500,000 SSU), y pueden desarrollar quite a altas presiones (sobre 1000 psi).

Estas bombas son de bajo costo y requieren poco espacio.

Algunas pueden ser rotadas en alguna dirección, tienen claros cerrados, requieren protección de relevo a sobre presión en la descarga, y tienen baja eficiencia volumétrica.

### 1.2 Compresores.

La compresión de aire y otros gases consume una gran cantidad de energía en las industrias de procesos químicos. En la producción de gases industriales como oxígeno, nitrógeno y helio y en la licuefacción de gas natural, la potencia para compresión es de más del 80% de la energía total requerida.

Al aumentar adiabáticamente la presión de un fluido compresible, aumenta también la temperatura del mismo, este aumento de temperatura tiene algunos inconvenientes. Como consecuencia del aumento de volumen especifico de un fluido con la temperatura, el trabajo necesario para comprimir un Kg. del mismo, es mayor que si la compresión fuese isotérmica. Las temperaturas excesivas crean problemas con los lubricantes, cajas prensa estopas, y materiales de construcción; además el fluido puede no soportar temperaturas no elevadas sin descomponerse.

Por otras parte como en los compresores reales existe fricción, el calor generado pro la misma es absorbido por el gas y llegan a alcanzarse temperaturas muy superiores a la adiabática. Los compresores, por consiguiente, están provistos de una camisa de refrigeración por la que se hace circular agua u otro refrigerante. En los compresores pequeños refrigerados, la temperatura del gas a la salida puede ser próxima a la de la entrada, consiguiéndose así una compresión isotérmica. En los muy pequeños, es suficiente la refrigeración por aire, que se consigue mediante las aletas externas de que esta provisto el cilintro.

#### 1.2.1 Concepto.

Los compresores son aparatos para compresión y movimientos de gases utilizados en las plantas de la industria de procesos químicos suelen ser complejos, construidos con precisión y costosos. Por ello, su selección, operación y mantenimiento deben ser cuidadosos.

#### 1.2.2 Clasificación.

Estos se clasifican más convenientemente teniendo en cuenta el intervalo de diferencia de presión que puede producir. Según este criterio el orden creciente es ventiladores, sopladores y compresores.

### 1.2.3 Ventiladores.

Los ventiladores de gran tamaño son generalmente centrífugos. El principio de funcionamiento es fundamentalmente el mismo que el de las bombas centrifugas. Los rodetes de un ventilador se montan dentro de una carcaza construida dentro de una lamina de metal ligero. La holgura es grande y las cargas de salidas pequeñas de 12 a 150 cm. de agua. Una diferencia de bombas y aparatos de flujos de gases radica en el efecto de la presión y temperatura sobre la densidad del gas que entra al sistema .

### 1.2.4 Compresores Centrífugos.

En un compresor centrifugo se produce la presión al aumentar la velocidad del gas que pasa por el impulsor y luego, al recuperarla en forma controlada para producir el flujo y presión deseados . Estos compresores pueden ser unitarios, salvo que el flujo sea muy grande o que las necesidades del proceso exijan otra cosa.

La mayor parte de los impulsores para la industria son del tipo de inclinación hacia atrás o inversa que permite mejor control por que su curva de rendimiento tiene mayor pendiente. La velocidad en las puntas de un impulsor convencional, suelen ser de 800 a 900 ft/seg.

Esto significa que el impulsor podrá producir alrededor de 9500 ft de carga, lo que depende del gas que se comprima. Si se requieren valores más altos, se emplean compresores de etapas múltiples .

Según sea el sistema para el proceso, se necesitan diversos controles contra oscilación para evitar que el compresor llegue al valor en el cual se produce. Por lo general, de debe incluir un factor de seguridad de 5 a 10 % para los controles automáticos.

Para el proceso, el compresor centrifugo tiene la ventaja de que envía gas libre de aceite y de que no hay piezas que se desgasten en la corriente del compresor.

Hay disponibles varios tipos de sellos de extremo. A continuación se mencionan tipos de sellos y sus limites normales de presión

**Tabla 1.9 Tipos de sellos y sus límites normales de presión.**

| Tipo de Sello | Presión Aprox. (psig) |
|---|---|
| Laberinto | 15 |
| Anillo de Carbón | 100 |
| Contacto mecánico | 500 |
| Película de aceite | 3000 o mayor |

La ventaja del sello de laberinto es que es del tipo de holgura sin piezas con rozamiento y es el más sencillo de todos. También se utiliza entre las etapas de los compresores de etapas múltiples. Su desventaja es la gran cantidad de fugas que permite, las cuales no se pueden permitir con gases costosos como el nitrógeno y el oxígeno.

Los sellos de anillo o carbón se utilizan salvo cuando el gas esta limpio o hay un medio amortiguador limpio que incluya un lubricante. Son de menor costo que los sellos de película de aceite o de contacto mecánico y tienen la ventaja de que impiden las fugas externas del gas comprimido.

En el sello de contacto mecánico hay una película de aceite que se mantiene entre sus caras estacionarias y giratoria. Tiene la ventaja de que minimiza el paso de aceite hacia el lado del gas. Su desventaja es una posible perdida de la película de aceite, lo cual puede ocasionar serios daños en las cara pareadas.

En el sello de película de aceite, como en el de contacto mecánico, se emplea la película para sellar el gas comprimido de la atmósfera. La desventaja es que necesitan controles complicados, bombas adicionales, un enfriador y filtro del aceite de sello.

Las carcazas de los compresores pueden ser del tipo dividido o partido, horizontal o vertical , con respecto al eje. Para el mantenimiento es más fácil el acceso al rotor con las carcaza dividida horizontalmente que con la que esta en forma vertical. Sin embargo, la de tipo horizontal tiene capacidad limitada de presión debido a la gran superficie de sellamiento en la unión. Cuando se utiliza carcaza dividida en sentido vertical se debe dejar espacio para sacar la carcaza interna y el rotor.

Ventajas de un compresor centrifugo:

En el intervalo de 2,000 a 200,000 $ft^3$ / min., y según sea la relación de presión, este compresor es económico por que se puede instalar una sola unidad.

Ofrece una variación bastante amplia en el flujo con un cambio pequeño en la carcaza.

La ausencia de piezas rozantes en la corriente de compresión permite trabajar un largo tiempo entre intervalos de mantenimiento.
Se pueden obtener grandes volúmenes en un lugar de tamaño pequeño.
Cuando se genera suficiente vapor en el proceso, un compresor centrifugo será adecuado para moverlo con una turbina de vapor de conexión directa.
Su característica es un flujo suave y libre de pulsaciones.

Desventajas :

Los centrífugos son sensibles al peso molecular del gas que se comprime. Los cambios imprevistos en el peso molecular pueden hacer que las presiones de descarga sean muy altas o muy bajas.
Se necesitan velocidades muy altas en las puntas para producir la presión .
Un aumento pequeño en la caída de presión en el sistema de proceso puede ocasionar reducciones muy grandes en el volumen del compresor.
Se requiere de un complicado sistema para aceite lubricante, y aceite para sellos.

### 1.2.5 Compresores de desplazamiento positivo.

Estos compresores se pueden dividir en rotatorios y reciprocantes para las aplicaciones más comunes en un proceso. Al contrario de los centrífugos, son de capacidad constante y tienen presiones de descarga variables.

Los compresores reciprocantes funcionan con el principio adiabático mediante el cual se introduce el gas en el cilindro por las válvulas de entrada se retiene y comprime en el cilindro y sale por las válvulas de descarga, en contra de la presión de descarga. Estos compresores rara vez se emplean como unidades individuales, salvo que el proceso requiera funcionamiento intermitente.

Los compresores reciprocantes tienen piezas en contacto como los anillos de los pistones con las paredes del cilindro, resortes y placas o discos de válvulas que se acoplan con sus asientos y entre la empaquetadura y la biela.

Estos pueden ser del tipo lubricado o sin lubricar. Si el proceso lo permite, es preferible tener un compresor lubricado, por que las piezas durarán más. Hay que tener cuidado de no lubricar en exceso, por que la carbonización del aceite en las válvulas puede ocasionar adherencias y sobrecalentamiento.

Los problemas más graves en los compresores con cilindros lubricados son las suciedad y la humedad, pues destruyen la película de aceite dentro del cilindro.

En los compresores sin lubricación, la mugre suele ser el problema más serio. En estos compresores, los anillos del pistón y del desgaste se suelen hacer con materiales rellenos con teflón, bronce, vidrio o carbón, según sea el gas que se comprima.

### 1.2.6 Compresores rotatorios de desplazamiento positivo.

Hay varios tipos de compresores rotatorios de desplazamiento positivo, entre ellos están el de tipo de soplador con lóbulos, el tipo de espiral rotatorio, el diseño de ayuda de agua y de aspas deslizables. Todos tiene el mismo tipo de curva de rendimiento que el compresor reciprocante, es decir, son de capacidad fija con contra presión variable. Los compresores rotatorios se prestan más para las unidades motrices de velocidad variable, como las turbinas de vapor, que los compresores reciprocantes.

El diseño de anillo de agua tiene la ventaja de que el gas no hace contacto con las partes rotatorias metálicas. Los aspectos críticos son la presión de vapor del gas de entrada, comparada con la presión de vapor del líquido que forma el anillo de agua y el aumento de temperatura en el mismo.

El compresor de aspas deslizables es muy compacto, pero tiene las misma ventaja que el reciprocante por que se necesitan piezas con rozamiento en la corriente de gas, y la perdida del lubricante puede ocasionar sobrecalentamiento del cilindro.

Los compresores de espiral y de lóbulos rotatorios ofrecen la ventaja de que el aire no contiene aceite por que no hay contacto con ninguna parte en la zona de compresión.

Su diseño rotatorio les de una capacidad mayor que la del compresor reciprocante y sin problemas de pulsaciones.

### 1.3 Válvulas

#### 1.3.1 Concepto

Las válvulas son aquellas que se utilizan para controlar la velocidad de flujo o para evitar el flujo del fluido. El diseño básico de la válvula es el que indica su uso.

Las válvulas sirven para oponer una restricción al flujo de fluidos, por lo que siempre hay una caída de presión relacionada con el flujo en una válvula. La reducción en la presión ocurre por las pérdidas de energía por fricción en el fluido de proceso.

#### 1.3.2 Clasificación

El tipo de válvula es determinado por el tipo de servicio que proporciona la válvula y las condiciones de funcionamiento.

Las válvulas se clasifican de acuerdo a su funcionamiento en:

- Válvulas de cierre o bloqueo
- Válvulas de estrangulación
- Válvulas de retención

Y de acuerdo al tipo de servicio en:

- Líquidos
- Gases
- Líquidos con gases
- Líquidos con sólidos
- Gases con sólidos
- Vapores
- Con corrosión o sin corrosión
- Con erosión o sin erosión

La decisión más importante para seleccionar una válvula es la de elegir los materiales para el cuerpo, las guarniciones y los asientos. Por supuesto, también son parte del proceso de selección el tamaño de las válvulas y de los operadores.

El fluido que pasa por una válvula puede dañar el cuerpo y las guarniciones por erosión o por corrosión. La corrosión química por halógenos puede ser muy severa, por ejemplo los cloruros atacan y penetran en la película protectora del acero inoxidable. También pueden ocurrir otras tres clases de corrosión menos comunes:

- La corrosión galvánica que ocurre cuando dos materiales diferentes están muy separados de la serie galvánica y está presente un electrolito fuerte.

- La corrosión intergranular que ocurre en los límites de los granos de los aceros inoxidables austeníticos (todos los aceros inoxidables de la Serie 300), cuando se calientan en la gama de 850 °F a 1650 °F.
- Las grietas por corrosión con esfuerzo, que ocurren, a veces como resultado de esfuerzos residuales internos ó de esfuerzos externos en el sistema de tubería.

Las válvulas hechas en su totalidad con resinas termoplásticas y las válvulas metálicas con las piezas, que tienen contacto con el fluido, revestidas con plástico se han vuelto muy comunes en servicios corrosivos.

La mejor selección de válvulas es la que producirá mínima caída de presión.

### 1.3.3 Válvula de Aguja

Las válvulas de aguja son, básicamente, válvulas de globo que tienen machos cónicos similares a agujas que ajustan con precisión en sus asientos. Al abrirlas, el vástago gira y se mueve hacia afuera. Por lo general, se utilizan como válvulas para instrumentos o en sistemas hidráulicos, aunque no para altas temperaturas.

Los extremos suelen ser roscados y sus tamaños van de 1/8 a 1 in.

Estas válvulas tienen las siguientes características:

- Gama de tamaño desde 1/8 hasta 1 in.
- Capacidad de presión hasta 10 000 psi
- Capacidad de temperatura criogénica hasta 500 °F
- Materiales de construcción: Bronce, hierro, acero, acero inoxidable y aleaciones especiales
- Servicio: Estrangulación suave y cierre con líquidos limpios

Las válvulas de aguja permiten estrangulación exacta de volúmenes pequeños

Esta válvula consiste en un macho o aguja cónico o conformado que tiene movimiento alterno en el cuerpo. La estrangulación se produce porque el orificio anular formado por el macho y el asiento varía según la posición del macho.

Las válvulas de aguja se pueden utilizar para controlar el paso de aire o líquidos hidráulicos para accionamiento. También se utilizan en los tubos para instrumentos a fin de reducir las pulsaciones de presión.

### 1.3.4 Válvula de Bola

Las válvulas de bola, básicamente, son válvulas de macho modificadas.
La bola tiene un orificio que se une con el cuerpo en la posición abierta.
La válvula de bola básica es una adaptación de la válvula de macho. Tiene una bola con un orificio en un eje geométrico para conectar las partes de entrada y de salida del cuerpo. En la posición abierta, el flujo es rectilíneo y para cerrarla, se gira la bola 90°.

Las válvulas de bola son de tipo venturi, orificio reducido y orificio completo.

Esta válvula, igual que la de macho, también es de 1/4 de vuelta. Los tamaños pequeños se hacen girar con una palanca, en los grandes se utiliza una unidad de engranes.

Además, las válvulas se clasifican como de bola flotante ó montada en muñón.

En la válvula de bola flotante, la presión en la tubería también aumenta la eficacia del sello, sin embargo, las bajas presiones diferenciales ocasionan problemas.

Para las bolas de montaje en muñones, la posición de la bola se fija con guías superior e inferior con cojinetes y la presión en la tubería mueve a los asientos contra la bola.

Estas válvulas se utilizan en forma principal para servicio de corte y no son satisfactorias para estrangulación. Son rápidas para operarlas, de mantenimiento fácil, no requieren lubricación, producen cierre hermético con baja torsión y su caída de presión es función del tamaño del orificio.

Tienen la desventaja que cuando están cerradas, atrapan algo de líquido entre el asiento y el orificio de bola, lo cual es indeseable en muchos casos.

Se pueden emplear para vapor, agua, aceite, gas, aire, fluidos corrosivos, pastas aguadas y materiales pulverizados secos.

Los principales elementos de esta válvula son:

- Cuerpo
- Asiento
- Bola

Hay dos tipos principales de cuerpos para válvulas de bola:

- Entrada superior
- Cuerpo dividido

En el de entrada superior, la bola y los asientos se instalan por la parte superior.

En el de cuerpo dividido, la bola y asientos se instalan desde los extremos.

Las bolas tienen orificios completos, de venturi y de superficie reducida. El orificio completo es igual al diámetro interior de la tubería. El orificio de venturi tiene superficies reducidas y hay flujo de venturi dentro del cuerpo.

Los extremos del cuerpo suelen ser con soldadura de enchufe, con brida ó roscados.

Las válvulas de bola, igual que las de macho, pueden ser de orificios múltiples y se pueden utilizar en lugar de dos ó tres válvulas rectilíneas, lo cual simplifica la tubería.

### 1.3.5 Válvula de Compuerta

Las válvulas de compuerta no se recomiendan para servicios de estrangulación, porque la compuerta y el sello tienden a sufrir erosión rápida cuando restringen la circulación y producen turbulencia con la compuerta parcialmente abierta.

Cuando la válvula está abierta del todo, se leva por completo la compuerta fuera del conducto del flujo, por lo cual el fluido pasa en línea recta por un conducto que suele tener el mismo diámetro que la tubería.

Las características principales del servicio de las válvulas de compuerta incluyen:

- Cierre completo sin estrangulación
- Operación poco frecuente
- Mínima resistencia a la circulación

Los principales elementos estructurales de la válvula de compuerta son:

- Volante
- Vástago
- Bonete
- Compuerta
- Asientos
- Cuerpo

Existen diferentes tipos de vástagos:

- Vástago no elevable, con rosca interna, tiene ventajas cuando hay poca altura.
- Vástago elevable con rosca externa que requiere más espacio libre, pero impide que la rosca esté en contacto con los fluidos del proceso.
- Vástago elevable con rosca interna, que expone la rosca del vástago a los líquidos del proceso, por tanto, no se debe usar con líquidos corrosivos.

También están disponibles diferentes tipos de bonetes:

- Bonetes con rosca interna ó externa para válvulas pequeñas y servicio a baja presión.
- Bonetes con unión para válvulas pequeñas donde se necesita mantenimiento frecuente.
- Bonetes con brida y atornillados para válvulas grandes y servicio a presión y temperatura altas.
- Bonetes con abrazadera en válvulas para presión moderada, donde se necesita limpieza frecuente.
- Bonetes sellados de presión para servicio con altas presiones y temperaturas.
- Bonetes con sello de pestaña para altas presiones y temperaturas.
- Bonetes con cierre de obturador para presión y temperaturas altas.

Los asientos de las válvulas de compuerta pueden ser integrales con el cuerpo ó ser de construcción anular.

Las fugas por las válvulas de compuerta pueden ocurrir en ambos extremos en donde se conectan a la tubería (cuando la válvula está abierta), en la unión entre el bonete y el cuerpo, en el vástago, y corriente abajo de la compuerta cuando la válvula está cerrada. Se pueden proveer sellos para evitar las fugas al exterior ó corriente abajo cuando está cerrada la válvula. Estos sellos pueden ser de metal a metal, metal en contacto con un material elástico, ó metal en contacto con un inserto elástico colocado en el cara del metal.

Esta válvula no se destina para servicio de estrangulación porque la compuerta y el asiento se erosionan con rapidez en cualquier posición que no sea la de apertura ó cierre totales. Cuando se abre ligeramente la válvula en un servicio de estrangulación, el disco y el asiento quedan sometidos a esfuerzos que causarían deformación y erosión que, afín de cuentas impedirán un cierre hermético.

### 1.3.6 Válvula de Diafragma

Las válvulas de diafragma se utilizan en servicios para corte y estrangulación y desempeñan una serie de servicios importantes para el control de líquidos.

La presión debe ser baja (no mayor de 150 psig) porque el diafragma no puede resistir grandes fuerzas.

La rotura del diafragma puede ocasionar un serio problema y grandes daños cuando se manejan productos químicos corrosivos.

Estas válvulas se fabrican con muy diversos materiales como hierro fundido, hierro dúctil, acero fundido, acero inoxidable y aleaciones resistentes a la corrosión.

Los tamaños normales son entre 1/8 y 24 in.

En este tipo de válvula los líquidos no pueden tener contacto con las piezas de trabajo en donde ocasionarían corrosión y fallas en servicio.

Cuando se abre la válvula, se eleva el diafragma fuera de la trayectoria de flujo y el líquido tiene un flujo suave y sin obstrucciones. Cuando se cierra la válvula, el diafragma asienta con rigidez contra un vertedero ó zona circular en el fondo de la válvula.

Las aplicaciones principales de las válvulas de diafragma son para bajas presiones y con pastas aguadas que obstruirían ó corroerían las piezas funcionales de la mayor parte de otros tipos de válvulas.

Estas válvulas no requieren empaquetadura en el vástago. Su duración depende de las presiones, temperaturas y la frecuencia de las aperturas y cierres.

Los componentes principales de este tipo de válvula son:

- Cuerpo
- Bonete
- Diafragma flexible

Los dos tipos generales de cuerpos son:

- El rectilíneo
- El vertedero ó Saunders.

La válvula de vertedero ó Saunders es preferible para estrangulación y también produce cierre hermético.

Los vástagos de las válvulas de diafragma no son giratorios, los diafragmas solo se mueven hacia arriba ó abajo con ayuda de un pistón de compresión, el cual, a su vez, se mueve con un brazo de palanca o un vástago giratorio.

Los extremos de la válvula pueden ser roscados, con brida, soldados a tope, con soldadura de enchufe ó con roscas macho higiénicas.

Hay muchos tipos de cuerpos para las válvulas de diafragma, los cuales son:

- De sumidero
- De circulación rectilínea

La primera es la que más se utiliza porque ofrece cierre hermético y una carrera corta que permite el uso de materiales más duros y menos flexibles.

El segundo tipo tiene uso limitado, porque hay pocos elastómeros de suficiente flexibilidad para soportar la carrera larga que se necesita.

La válvula de diafragma es adecuada para productos viscosos, pastas aguadas ó líquidos corrosivos.

### 1.3.7 Válvula de Globo

La construcción interna de esta válvula consiste en un disco ó macho que se mueve dentro del cuerpo de la válvula y acopla con un asiento para el cierre.

La caída de presión suele ser grande y para minimizarla, muchos fabricantes ofrecen válvulas en Y y en ángulo.

Las principales características de los servicios de las válvulas de globo incluyen operación frecuente, estrangulación al grado deseado de cualquier flujo, cierre positivo para gases y aire, y alta resistencia y caída tolerable de presión en la línea.

Estas válvulas tienen las siguientes características:

- Gama de tamaño desde 1/2 in hasta 30 in.
- Capacidad de presión hasta 2 500 psi.
- Capacidad de temperatura hasta 1 000 °F.
- Materiales de construcción: Bronce, hierro, hierro fundido, acero forjado, acero fundido, acero inoxidable, latón y aleaciones resistentes a la corrosión.
- Servicio: Estrangulación y cierre con líquidos limpios.

Los principales elementos estructurales de la válvula de globo son:

- Volante
- Vástago
- Bonete
- Asientos
- Disco
- Cuerpo

Para éste tipo de válvula existen diferentes tipos de vástagos:

- *Vástago elevable con rosca interna*, el cual no debe utilizarse en tuberías que manejan material corrosivo porque las roscas del vástago sólo tienen protección parcial.
- *Vástago elevable con rosca externa.*
- *Vástago deslizable* para apertura y cierre rápidos.

También existen diferentes tipos de bonetes:

- *Bonetes de rosca interna y externa*, para válvulas pequeñas, cuando existen bajas temperaturas y presiones.
- *Bonete de unión* para válvulas pequeñas, cuando se requiere desarmarlas con frecuencia.
- *Bonete con brida* atornillado para válvulas grandes y presiones o temperaturas altas.
- *Bonete sellado a presión* para servicio a temperaturas y presiones altas.

Los asientos de las válvulas de globo pueden ser fundidos integrales ó anillos de asiento reemplazables que se fijan con tornillos ó en alguna otra forma.

La principal diferencia entre la válvula de compuerta y de globo es que la primera previene la corriente abajo del elemento de control.

Las válvulas de globo de operación manual tienen un disco ó un macho que acoplan con un anillo de asiento metálico. El disco puede ser todo de metal ó tener un inserto elástico. El sellamiento elástico se hace al oprimir una superficie metálica contra una de caucho (hule) ó plástico. Cuando el servicio no es severo ó la presión no es alta, este tipo de sello produce un cierre hermético y es deseable para líquidos que contienen partículas de sólidos.

Las válvulas de globo son usadas primordialmente para regular el paso de un flujo. Debido a su diseño nos pueden proporcionar un cierre hermético en aquellos casos en que sea necesario y al mismo tiempo nos permiten un continuo abrir y cerrar, según el caso lo requiera.

El uso principal de este tipo de válvula es cortar ó regular el flujo del líquido.

Debido a su alta caída de presión, aun cuando se encuentren totalmente abiertas no deben ser usadas cuando se requiera un flujo continuo.

Se utilizan para regular gases y/o líquidos, aún cuando éstos tengan partículas en suspensión, ya que debido al diseño de la válvula no dañan los asientos ni el disco.

Esta válvula puede utilizarse en: Compresoras, Bombas, Condensadores de vapor, líneas de vapor, Generadores de turbina, Plantas termoeléctricas, instrumentación, líneas de derivación, muestreo. En Refinerías para unidades de alcalización, plantas catalíticas, torres de destilación, etc.

### 1.3.8 Válvula de Macho

Dado que el flujo por la válvula es suave e ininterrumpido, hay poca turbulencia dentro de ella y, por tanto, la caída de presión es baja. Las ventajas principales de las válvulas de macho son acción rápida, operación sencilla, espacio mínimo para instalación y cierre hermético cuando tienen macho cónico.

Hay dos tipos principales de válvulas de macho:

- Lubricados para evitar las fugas entre la superficie del macho y el asiento en el cuerpo y reducir la fricción durante la rotación
- Los no lubricados en que el macho tiene un revestimiento que elimina la necesidad de la lubricación.

Los principales servicios de las válvulas de macho incluyen apertura ó cierre total sin estrangulación, tienen mínima resistencia al flujo, son para operación frecuente y tienen poca caída de presión.

Los componentes básicos de esta válvula son:

- Cuerpo
- Macho
- Tapa

Los dos tipos de válvulas de macho son:

- Circulación rectilínea
- Orificios múltiples.

El primero es cónico ó cilíndrico y tiene un conducto por el cual circula el líquido, en la posición abierta, la cavidad en el macho conecta los extremos de entrada y salida de la válvula y permite flujo lineal. Y en el segundo los orificios son de diferentes diseños, como sigue:

- Orificio redondo completo. Tiene una abertura para toda la cavidad en el macho y el cuerpo.
- Orificio rectangular. Tiene orificios de tamaño completo, por lo general rectangulares y con una apertura mínima del 70% del tamaño de la tubería.
- Orificio venturi. Tiene aberturas redondas ó rectangulares con superficie reducida y con flujo de venturi en el cuerpo.
- Orificio de rombo. La abertura del macho es en forma de rombo.

Una característica importante de la válvula de macho es su fácil adaptación al tipo de orificios múltiples.

En las válvulas de un solo orificio el macho gira 90° desde la posición de apertura total a la de cierre total y se le denomina, a veces, válvula de un cuarto de vuelta.

Las válvulas con orificios múltiples pueden girar hasta 270° .

La válvula de macho es muy compacta y requiere un mínimo de espacio. Por ello, pesa menos que las válvulas de compuerta ó de globo.

Las limitaciones básicas de las válvulas de macho son los problemas con la torsión y los asentamientos.

En las válvulas con macho lubricado, se inyecta lubricante a presión para evitar las fugas de líquidos entre la cara del macho y el cuerpo. Además, se reduce la fricción al girar el macho. El macho tiene ranuras que permiten que el lubricante forme un sello. La presión del lubricante eleva el macho y facilita la operación. Las válvulas macho lubricadas se utilizan mucho en la industria petrolera. Este tipo de válvula tiene cierre hermético e inhibe la corrosión. La ventaja de estas válvulas es la operación rápida. Cuando no se puede permitir la contaminación de los productos se utilizan válvulas sin lubricación.

Las válvulas con macho no lubricado están disponibles en los tipos:

- Elevable
- Con camisa de elastómeros ó revestimiento para el macho.

Para operar la válvula elevable se levanta en forma mecánica el macho para facilitar la rotación.

Para la segunda la camisa ó revestimiento de elastómero se aplica como revestimiento ó por inmersión en el cuerpo ordenado por completo el macho. Las ventajas principales de las válvulas con macho no lubricado son el cierre hermético, operación rápida, ausencia de problemas de lubricación y amplia gama de temperaturas.

Estas válvulas tienen extremos de rosca, con brida y soldados.

Estas válvulas tienen las siguientes características:

- Gama de tamaño hasta 30 in.
- Capacidad de presión hasta 5000 psi.
- Capacidad de temperatura hasta 600 °F.
- Materiales de construcción: Hierro, acero, acero inoxidable y diversas aleaciones resistentes a la corrosión. Disponibles con camisa complota de caucho ó plástico.
- Servicio: Cierre (estrangulación en algunos tipos).

El uso principal de las válvulas de macho es en servicio de corte y sin estrangulación.

Las válvulas con orificio en V se utilizan en servicio con pasta aguada ó pulpa, en las industrias química y del papel.

El problema más serio con las válvulas de macho es accionarlas después de que han estado en la misma posición un tiempo largo sin moverlas. En algunos tipos se utiliza un sistema de elevación para levantar el macho antes de hacerlo girar.

Las válvulas de macho son uno de los tipos ideales para manejar corrientes con alto contenido de sólidos, incluso pastas aguadas muy espesas. Las válvulas lubricadas se pueden utilizar para estrangulación aunque puede ocurrir abrasión si hay sólidos en el material que circula. Las válvulas no lubricadas no suelen ser aptas para estrangulación, salvo con caídas pequeñas de presión, por el peligro de contracción y aplastamiento de la camisa.

### 1.3.9 Válvula de Mariposa

Estas válvulas son sencillas, ligeras y de bajo costo. El diseño abierto de flujo rectilíneo evita la acumulación de sólidos y produce baja caída de presión. Su operación es fácil y rápida con una manija. La regulación del flujo se efectúa con un disco de válvula que sella contra un asiento.

Las válvulas de mariposa son del tipo de oscilación total ó con asientos para el cierre.

Las válvulas para baja presión suelen tener revestimiento completo con caucho ó material similar, cuando cierran el disco comprime el revestimiento en los 360°.

Los cuerpos de las válvulas de mariposa de tamaño pequeño son de extremos roscados y se atornillan a la tubería. Las válvulas de tamaño más grande están destinadas para instalación entre un par de bridas y se clasifican como tipo de disco y tipo de orejas.

Las válvulas de mariposa son adecuadas para servicio de paso y cierre ó de estrangulación y tiene bajas pérdidas por fricción del líquido.

Son de acción rápida, porque 1/4 de vuelta del vástago moverá al disco de la posición de apertura a la de cierre total. Las válvulas con revestimiento están limitadas a presiones no mayores de 150 psi y todas ellas tienen limitaciones para temperaturas debido al material del asiento y el sello.

Algunos tipos de válvulas tienen asientos duros con sellos anulares alrededor del disco; otros, pueden tener asiento blanco y disco descentrado. Es difícil estrangular con una válvula de mariposa entre las posiciones de 60° de apertura y apertura total.

Estas válvulas tienen las siguientes características:

- Gama de tamaño desde 2 hasta 2 ft. ó más.
- Capacidad de presión hasta 2 000 psi.
- Capacidad de temperatura hasta 2 000 °F.
- Materiales de construcción: Las camisas pueden ser de plástico, caucho ó cerámica.
- Servicio: Estrangulación (cierre sólo con asientos ó tipos especiales), líquidos limpios y pastas aguadas.

Los principales elementos estructurales de la válvula de globo son:

- Disco (llamado también aspa, chapaleta u hoja)
- Eje
- Cuerpo con empaquetadura y cojinetes para sellamiento y soporte.

Los dos cuerpos disponibles son el de anillo macizo ó de placa y oblea y el de carrete. El cuerpo de anillo macizo se atornilla entre las bridas de tubo y requiere poco espacio.

Los discos circulares cierran con el disco paralelo al diámetro del cuerpo. Los discos de tipo elíptico cierran entre 10° y 15° fuera del diámetro del cuerpo. Los discos circulares pueden girar 360° y se requiera cierta holgura entre el disco y el cuerpo.

Las válvulas con poca caída de presión y baja torsión (llamadas a veces ligeras) tienen disco y eje delgados para máxima capacidad de flujo. Las válvulas gruesas son para caídas grandes de presión y tienen disco reforzado y eje más grueso para manejar la alta torsión requerida.

Por lo general, las válvulas de mariposa se fabrican con una camisa completa, sello de asiento y superficies de asiento para la junta en la brida hechas con elastómero.

En el control de flujo, las ventajas de las válvulas de mariposa incluyen la facilidad de mantenimiento sin necesidad de herramientas ó equipo especiales y sin necesidad de paros de la planta.

Las partes esenciales que integran una válvula de mariposa son:

- Cuerpo de la válvula
- Rodamientos inferior y superior (bushing)
- Asiento
- Empaque de la flecha

- Disco
- Flecha
- Tornillos de la flecha

*Cuerpo* de una sola pieza, es un anillo sólido producto de fundición que está maquinado con una exactitud, tal que no existan grandes fugas en el espacio existente entre este y el disco.

Los materiales de construcción estándar son hierro fundido, bronce, aluminio, acero al carbón, acero inoxidable, y algunos otros metales para usos muy específicos.

*Disco* convencional, el disco es de material fundido, generalmente tiene el borde maquinado y al filo se le da un tratamiento especial para que ocasione la mínima turbulencia al fluido.

En el eje se emplea material suficiente para mantener la rigidez y resistencia del disco y sus componentes.

En algunos casos el disco puede girar solo hacia un lado, en cambio en otros puede girar para cualquier lado indistintamente, lo cual representa una ventaja al evitar que se pueda atascar la válvula de mariposa.

El disco cierra en posición paralela al cuerpo, basta un giro de solo 90° para que el disco pase de abierto a cerrado ó viceversa.

Cuando se manejan sustancias corrosivas los bordes del disco están cromados, para que así la protección sea más completa.

Los materiales de construcción comúnmente usados son los siguientes: bronce, hierro fundido, aluminio, acero inoxidable, monel.

Disco de diseño de cola de pescado.
El disco del material fundido se hace en los mismos metales que el anterior.

Se puede usar en cualquier válvula de mariposa de asiento metálico, para trabajo pesado.

La función de la cola de pescado en el disco es retardar el flujo en la parte inferior de la tubería, resultando así un buen control.

Los *rodamientos* ó bushings sirven como chumaceras para facilitar el movimiento de la flecha y evitar los desgastes. Los materiales usados son: bronce, bronce recubierto de teflón en válvulas de 14" ó 20", en válvulas más pequeñas se usa una fibra de vidrio con teflón, otros materiales que también se emplean son acero inoxidable y monel.

El *asiento* ó forro del cuerpo de la válvula es una parte que no siempre está presente, en aquellas válvulas de asiento metálico es el mismo cuerpo de la válvula el que sirve de asiento al disco.

El asiento es de un material elástico, los materiales comúnmente usados son: neopreno, hycar, hule natural, silicón, hypalon, poliuretanos, nitrilos, teflón y otros.

En otro aspecto, los bordes del asiento embonan en las ranuras del cuerpo de la válvula asegurando en esta forma su posición sin necesidad de fijarlo con cemento ó cualquier otro pegamento.

Para reemplazarlo basta tan solo presionar los bordes hacia abajo, habiendo quitado previamente el disco y la flecha, presionándolo hasta retirarlo completamente.

Otra de las funciones del reborde del asiento elástico de la válvula es, su utilidad como empaque entre la cara de la brida y la cara del cuerpo, eliminando por consiguiente el empleo de empaques en las bridas.

La *flecha* es una pieza, generalmente torneada.
Los materiales empleados para su construcción son monel, acero al carbón, acero inoxidable y otros. Cuando se emplea acero al carbón suele recubrírsele de un forro plástico.

Monel es una aleación de níquel y cobre, 60% de Ni el resto de Cu.

Para evitar el movimiento entre la flecha y el disco se fijan entre ambos por medio de unas pequeñas espigas que se introducen en unos orificios practicados para tal propósito sobre el eje del disco.

Para dar un soporte adecuado a la flecha se usan unas guías interiores de acero inoxidable forrado de teflón.

Se le ponen también a la flecha una serie de empaques y sellos para evitar posibles fugas de fluidos del proceso, o que se pueda mezclar ese fluido con el lubricante, en los casos en que este es empleado.

Las principales características de los servicios de las válvulas de mariposa incluyen:

- Apertura total
- Cierre total ó estrangulación
- Operación frecuente
- Cierre positivo para gases ó líquidos
- Baja caída de presión

Las válvulas de contacto de metal con metal ó de giro completo se suelen utilizar en aplicaciones en donde se esperan grandes velocidades y mucha variación en el flujo o en la caída de presión.

Las válvulas con revestimiento ó camisa de elastómero son las que más se utilizan. Están disponibles en tres configuraciones básicas:

- Disco de centramiento vertical
- Disco desplazado en sentido vertical
- Disco en ángulo

Las primeras tienen menor duración debido a la fricción y asentamiento por compresión que ocurren en el asiento en la zona de la protuberancia para el disco.

Los discos desplazados en sentido vertical eliminan este problema pero tienen orificios de menor diámetro, que reducen la eficiencia.

Las válvulas de disco en ángulo eliminan esos dos problemas.

La válvula con revestimiento elástico tiene la ventaja básica de que aíslan el fluido del proceso del cuerpo de la válvula. La válvula de mariposa con camisa es de cierre positivo.

La mayor parte de las válvulas con revestimiento elástico están equipadas con asientos reemplazables que tienen un anillo duro de apoyo para impedir que se contraiga el asiento, también facilita el desmontaje e instalación del asiento.

La válvula con sellamiento de elastómero limitado también es de cierre positivo. Tiene ventajas con respecto a la válvula de revestimiento elástico porque minimiza la cantidad de elastómero requerida para el asiento.

La válvula con cuerpo de placa u oblea está disponible en configuraciones de punteo, orejas, brida sencilla y brida doble. La más común es la de tipo de punteo por la facilidad de deslizar el cuerpo entre las bridas sobre una cuna de tornillos, tiene un solo juego de tornillos y no necesita alineación especial.

El tipo de orejas también es deslizable, pero hay que alinearlo y colocar los tornillos por un lado y otro, y apretarlos a una torsión uniforme.

La válvula de placa, estrecha, de cara con cara es la de mayor uso en las plantas de proceso porque ofrece muchas ventajas cuando se diseña un sistema de tubería, entre ellas ahorro de espacio, ligereza de peso, suspensores y soportes más delgados, confiabilidad y la variedad de tipos de guarniciones y de válvulas.

Aunque las válvulas de mariposa son excelentes válvulas de control, su uso más común es para cierre. Producen cierre hermético a prueba de goteo en casi cualquier aplicación en un proceso, incluso vapor, aire, gases, líquidos, pastas aguadas y sólidos.

El uso principal de estas válvulas es para servicio de corte y estrangulación cuando se manejan grandes volúmenes de gases y líquidos a presiones relativamente bajas.

Son adecuadas en especial para grandes volúmenes de gases ó líquidos a presiones bajas.

Son una buena selección para pastas aguadas ó líquidos con muchos sólidos en suspensión y no permiten la acumulación de sedimentos.

Estas válvulas se han utilizado en la industria de procesos químicos durante muchos años en sistemas de distribución de agua, torres de enfriamiento y tuberías para aire, como válvulas de aislamiento o admisión de bombas y en tuberías para espuma y agua en los sistemas de extinción de incendios.

Sus aplicaciones con productos corrosivos incluyen nitrato de amonio, soluciones salinas, soluciones cáusticas calientes y vapor.

Además, trabajan bien en servicio en procesos secos como polímeros y resinas pulverizados, válvulas de descarga de tolvas y secadoras, en tuberías para transporte neumático y en sistemas de recuperación de polvos.

### 1.3.10 Válvula de Retención

Las válvulas de retención son integrales y se destinan a impedir la inversión del flujo en una tubería. La presión del fluido circulante abre la válvula, el peso del mecanismo de retención y cualquier inversión en el flujo la cierran.

Las válvulas de retención ó check impiden el flujo inverso en las tuberías. Son de funcionamiento automático y se mantienen abiertas por la presión del fluido que circula.

Hay diferentes tipos de válvulas de retención y su selección depende de la temperatura, caída de presión que producen y la limpieza del fluido.

Las válvulas de retención están disponibles en los tipos de:

- Bisagra
- Disco inclinable
- Elevación (disco, pistón ó bola)
- Cierre
- Retención para vapor

La válvula de bisagra tiene una placa ó chapaleta embisagrada en la parte superior, que produce muy poca caída de presión. La placa puede ser un disco de material compuesto cuando el líquido contiene partículas de sólidos, el ruido es indeseable ó si se requiere un cierre hermético. Esta válvula abre con la presión en la tubería pues el flujo en sentido normal hará que el disco oscile y se separe del asiento. Se cierra cuando se reduce la presión y llega a cero; en este caso, el disco queda sujeto contra el anillo de asiento por su propio peso ó por pesos externos conectados a un eje que pasa a través del cuerpo.

La válvula de retención de bisagra (columpio) abre con la presión en la tubería pues el flujo en sentido normal hará que el disco oscile y se separe del asiento. Se cierra cuando se reduce la presión y llega a cero; en este caso, el disco queda sujeto contra el

anillo de asiento por su propio peso ó por pesos externos conectados a un eje que pasa a través del cuerpo.

Esta válvula funciona por gravedad, cosa que se debe tener en cuenta para instalarla.

La válvula de retención de bisagra se utiliza con bajas velocidades de fluido con inversiones de flujo poco frecuentes; en algunos sistemas se utilizan en combinación con válvulas de compuerta. Las principales características de estas válvulas de retención son mínima resistencia al flujo, servicios de baja velocidad y con cambios de dirección poco frecuentes.

La válvula de bisagra con disco dividido es una variante de la válvula anteriormente descrita. Las dos mitades del disco están embisagradas con un pasador y tienen un resorte para mantenerlas cerradas cuando no hay flujo. Esta válvula no funciona por gravedad, lo cual permite más flexibilidad en la instalación. Es adecuada para instalaciones con inversiones frecuentes de la circulación porque, al contrario de la válvula de bisagra, no cierra de golpe ni ocasiona choques de presión.

Las válvulas de retención por elevación, por lo general con cuerpo de globo, funcionan por gravedad y son para instalación horizontal ó vertical, pero no son intercambiables. Estas válvulas requieren caídas de presión bastante elevadas. Se utilizan en servicios con alta presión y en tuberías más pequeñas que las válvulas de bisagra.

Las válvulas de elevación son mejores que las de bisagra en servicios en donde hay frecuentes inversiones, porque el pistón está amortiguado para evitar el golpe del ariete.

Las principales características de estas válvulas de retención son mínima resistencia al flujo, servicios de baja velocidad y con cambios de dirección poco frecuentes.

Los componentes principales de estas válvulas son el cuerpo, disco, pasador oscilante y tapa. Hay dos tipos principales de cuerpo:

- En Y
- Rectilíneo

Las válvulas en Y tienen una abertura alineada con el asiento, que está integrada al cuerpo; esto permite rectificar por esmerilado las válvulas que asientan metal contra metal.

Las válvulas rectilíneas tienen un disco embisagrado en la parte superior, con lo cual la superficie de asentamiento está a un pequeño ángulo, lo cual permite que el disco oscile y se abra con presiones más bajas.

Los discos que se emplean en estas válvulas son metálicos o de composición. Los metálicos están disponibles en configuración para flujo en Y y rectilíneo. Los discos de composición son preferibles para líquidos que contienen partículas extrañas.

Las tapas son:

- Roscada, que es la más económica y sencilla
- Atornillada con junta entre la tapa y el cuerpo.

Los extremos de las válvulas de retención de bisagra pueden ser de rosca, con brida ó soldados.

La válvula de retención de disco inclinable, es muy semejante a la de bisagra. Hay baja resistencia al flujo debido a su diseño rectilíneo. Estas válvulas consisten en una cubierta cilíndrica que tiene un disco pivotado (inclinable ó giratorio).

Esta válvula de retención tiene poca caída de presión a baja velocidad y mayor caída de presión a alta velocidad.

Los componentes principales de la válvula de disco inclinado son:

- El disco
- El eje (varilla) de pivoteo
- Cuerpo

Estas válvulas están disponibles con sello de anillo blando ó de metal con metal. Sus extremos pueden ser bridados.

Estas válvulas se pueden instalar en una tubería horizontal ó en una vertical con flujo ascendente.

Las válvulas de disco oscilante mantienen una baja resistencia a la circulación por su diseño rectilíneo. No cierra de golpe. La placa flota cuando hay pleno flujo y empieza a cerrar por inclinación a un ángulo creciente con la trayectoria de paso cuando se reduce el flujo. El disco inclinable produce menos caída de presión a bajas velocidades y más caída a alta velocidad que una válvula equivalente de bisagra.

La caída de presión en cualquiera de los tipos de válvulas con cuerpo de placa suele ser más grande en los tamaños pequeños que en los de bisagra y menor en los más grandes. Las válvulas con cuerpo de placa ó con brida se deben instalar en tuberías horizontales ó verticales con flujo ascendente, pero nunca con flujo descendente.

La válvula de retención de pistón es un disco con un amortiguador constituido por un pistón y un cilindro que amortiguan cuando ocurre la inversión de flujo. Esta válvula es adecuada para servicios en donde ocurren frecuentes inversiones de flujo y es, quizá, la única que se ha utilizado con éxito en aplicaciones en donde alterna el flujo. La caída de presión es mayor que en cualquiera de los otros tipos.

# ANEXO II

## Listado del Programa

BIBLIOGRAFIA.


1. Ludwing, Ernest E. 1977. Applied Process Design for Chemical and Petrochemical Plants. Vol.1 2a. Ed. Gulf Publishing Company. USA

2. Martínez Tinajero, Augusto René y Seegrove de la Vega, Horacio F. 1969. Bombas Centrífugas. Tesis. Facultad de Química, UNAM.

3. Pérez Camacho, Ricardo. 1982. Diseño de una práctica para el Laboratorio de Ingeniería Química con bombas centrífugas en serie y paralelo. Tesis. Facultad de Química, UNAM.

4. Noriega Gutiérrez, Eduardo. 1973. Selección de Bombas Centrífugas y su aplicación al manejo de líquidos corrosivos. Tesis. Facultad de Química, UNAM.

5. Rebuelta Lacal, Ma. Elena. 1969. Monografía sobre Válvulas de mariposa. Tesis. Facultad de Química, UNAM.

6. Gómez Pimienta, José Manuel. 1968. Estimación del costo de bombas centrífugas en México. Tesis. Facultad de Química, UNAM.

7. Karassik, Igor J., et. al. 1983. Manual de Bombas. Diseño, aplicación, especificaciones, operación y mantenimiento. McGraw-Hill.

8. Cetre, Shankuntala. 1988. Técnicas de Bases de Datos. Estructuración en diseño y administración. Ed. Trillas.

9. Larson, James A. y Freeman, Harvey A. 1981. Tutorial. DataBase Management in the 1980's. Computer Society Press.

10. Date, C.J. 1981. An introduction to DataBase Systems. 3a. Ed. The systems programming series. Addison-Wesley Publishing Company.

11. Mahnke, Hans. 1987. Ingeniería de Software práctico y conciso. Planificación, métodos y técnicas de representación- Datanet, S.A.

12. Mayrhauser, Anneliese Von. 1990. Software Engineering. Methods and Management. Academic Press, Inc. USA.

13. Ullman, Jeffrey D. 1982. Principles of DataBase Systems. 2a. Ed. Science Press.

14. Atre, Shakuntala. 1988. Técnicas de Bases de Datos. Estructuración en Diseño y Administración. Ed. Trillas.

15. McNaughton, Kenneth. 1989. Bombas. Selección, uso y mantenimiento. McGraw-Hill. USA.

16. Greene, Richard W. 1989. Válvulas. Selección, uso y mantenimiento. McGraw-Hill. USA.

17. Richer Vela, Oscar. 1989. Computación. Biblioteca Científica y Tecnológica. Ediciones Ciencia y Técnica, S. A. México.

18. Fairley, Richard E. 1988. Ingeniería de Software. McGraw-Hill. USA.

19. Llorenç Guilera Agüera. 1988. Introducción a la Informática. EDUNSA Ediciones y Distribuciones Universitarias, S.A. Barcelona, España.

20. J. Myers, Glenford. 1983. El arte de probar el Software. Librería "El Ateneo" Editorial. Argentina.

21. Ghezzi, Carlo et. al. 1991. Fundamentals of Software Engineering. Prentice Hall. New Jersey, USA.

22. Van Vlient, Hans. 1993. Software Engineering. Principles and Practice. John Wiley & Sons. New Yor, NY. USA

23. Sommerville, Ian. 1988. Ingeniería del Software. 2a. Edición. Addison-Wesley Iberoamerica. USA

Object : Tesis1

MethodName : Uses

Source :
```
Uses ObjectPal
 MensajeLIB(NombreFicha String, NombrePagina String, NumerodeObjeto String)
 NombreDePagina(Interfaz UIObject) String
 EsLlamadaVerdadero()
 EsLlamadaFalso()
endUses
```

Object : Tesis1

MethodName : Var

Source :
```
Var
NombreInterfaz String
Interfaz     UIObject
Valvula,
Compresor,
AcercaDe,
NombreFicha   Form
tesislib     Library
endVar
```

Object : Tesis1

MethodName : open

Source :
```
method open(var eventInfo Event)
var
 titulo Application
endvar
if eventInfo.isPreFilter()
        then
                ; este código se ejecuta para cada objeto de la ficha

        else
                ; este código se ejecuta sólo para la ficha en sí

titulo.setTitle("Ingeniería Química")
titulo.maximize()

endif
endmethod
```

Object : Tesis1

MethodName : close

Source :
```
method close(var eventInfo Event)
 if eventInfo.isPreFilter() then
        doDefault
 else
  if valvula.isAssigned() then
   valvula.close()
  endif
  showspeedbar()
  removemenu()
```

```
 endif
endmethod
```

Object : Tesis1

MethodName : mouseEnter

Source :

```
method mouseEnter(var eventInfo MouseEvent)
if eventInfo.isPreFilter()  then
                    eventInfo.getTarget(interfaz)
   nombreInterfaz = Interfaz.Name
   if Interfaz.Name <> Self.Name  AND
      (Interfaz.class = "Field" OR Interfaz.class = "MultiRecord" OR
       Interfaz.class = "Button") then
     tesislib.MensajeLib(NombreFicha.Name,tesislib.NombreDePagina(Interfaz),nombreInterfaz)
   endif
endif
endmethod
```

Object : Tesis1

MethodName : mouseExit

Source :

```
method mouseExit(var eventInfo MouseEvent)
if eventInfo.isPreFilter()  then
  eventInfo.getTarget(Interfaz)
  if Interfaz.class <> "Text" AND Interfaz.class <> "Bitmap" then
    message("")
  endif
endif
endmethod
```

Object : Tesis1

MethodName : keyPhysical

Source :

```
method keyPhysical(var eventInfo KeyEvent)
var
 laTecla String
 lapagina String
endVar
if eventInfo.isPreFilter()  then
 laPagina = tesislib.NombreDePagina(active)
 laTecla  = eventinfo.vChar()
 switch
  case lapagina = "Principal" :
   if eventInfo.isAltKeyDown() then
    switch
      case laTecla = "B" or laTecla ="b": disableDefault
                Principal.BotonCompresor.pushbutton()
      case laTecla = "V" or laTecla ="v": disableDefault
```

```
                Principal.BotonValvula.pushbutton()
            case laTecla = "Y" or laTecla ="y": disableDefault
                Principal.BotonAyuda.pushbutton()
            case laTecla = "S" or laTecla ="s": disableDefault
                Principal.BotonSalir.pushbutton()
          endswitch
        endif
        if eventInfo.isFromUI() then
          switch
            case laTecla ="VK_F1": disabledefault
              Principal.BotonAyuda.pushbutton()
            case laTecla ="VK_F2": disabledefault
            case laTecla ="VK_F3": disabledefault
            case laTecla ="VK_F4": disabledefault
            case laTecla ="VK_F5": disabledefault
            case laTecla ="VK_F6": disabledefault
            case laTecla ="VK_F7": disableDefault
            case laTecla ="VK_F8": disableDefault
            case laTecla ="VK_F9": disabledefault
              Principal.BotonSalir.pushbutton()
            case laTecla ="VK_F10": disabledefault
          endswitch
        endif
      endswitch
    endif
  endmethod
```

Object : Principal

MethodName : open

Source :

```
method open(var eventInfo Event)
 if not isfile("Tesis1.fsl") then
   msgInfo("ERROR","No estas en el directorio de trabajo")
   close()
 endif

 if not tesislib.open("tesislib.fsl",GlobalToDesktop) then
   msgStop("Error","No pude abrir la Biblioteca")
 endif
   NombreFicha.attach()
         hidespeedbar()
   maximize()
```

```
endmethod
```

Object : Principal

MethodName : arrive

Source :
```
method arrive(var eventInfo MoveEvent)
  MenuPaginaPrincipal()
endmethod
```

Object : Principal

MethodName : menuAction

Source :
```
method menuAction(var eventInfo MenuEvent)
  var
    OpcionMenu string
  endvar
  OpcionMenu = eventInfo.menuChoice()
  switch
    case OpcionMenu = "&Bombas y Compresores"  : BotonCompresor.pushbutton()
    case OpcionMenu = "&Válvula"               : BotonValvula.pushbutton()
    case OpcionMenu = "&Manual de Usuario"     : BotonAyuda.pushbutton()
    case OpcionMenu = "&Información Teórica"   : helpShowIndex("infor.hlp")


    case OpcionMenu = "&Salir"                 : BotonSalir.pushbutton()
  endswitch
endmethod
```

Object : Principal

MethodName : proc

Source :
```
proc MenuPaginaPrincipal()
var
  menuprincipal menu
  sub1,sub2  PopUpMenu
endvar
  sub1.addText("&Bombas y Compresores")
  sub1.addText("&Válvula")
  sub1.addSeparator()
  sub1.addText("&Salir")
```

```
 menuprincipal.addPopUp("&Archivo",sub1)


 sub2.addText("&Manual de Usuario")
 sub2.addText("&Información Teórica")
 menuprincipal.addPopUp("&?",sub2)
 menuprincipal.show()

endproc


proc procsalir()
 if msgQuestion("Salir de la Aplicación","¿Quiere salir de esta aplicación?") = "Yes" then
  removemenu()
  close()
  if valvula.isAssigned() then
   valvula.close()
  endif
  if compresor.isAssigned() then
   compresor.close()
  endif
  exit()
 endif
endproc
```

Object : Principal.BotonCompresor

MethodName : pushButton

Source :

```
method pushButton(var eventInfo Event)
if isAssigned(compresor) then
 compresor.bringToTop()
 compresor.wait()
 compresor.hide()
 maximize()
 principal.moveTo()
else
 tesislib.esLlamadaVerdadero()
 if compresor.open("bombas", WinStyleMaximize) then
  compresor.wait()
  compresor.hide()
  maximize()
  principal.moveTo()
 else
  msgInfo("Error", "No puede abrir la tabla")
  tesisLib.EsLlamadaFalso()
 endif
endif
endmethod
```

Object : Principal.BotonValvula

MethodName : pushButton

Source :
```
method pushButton(var eventInfo Event)
if isAssigned(valvula) then
 valvula.bringToTop()
 valvula.wait()
 valvula.hide()
 maximize()
 principal.moveTo()
else
 tesislib.esLlamadaVerdadero()
 if valvula.open("valvulas", WinStyleMaximize) then
  valvula.wait()
  valvula.hide()
  maximize()
  principal.moveTo()
 else
  msgInfo("Error", "No puede abrir la tabla")
  tesisLib.EsLlamadaFalso()
 endif
endif
endmethod
```

Object : Principal.BotonSalir

MethodName : pushButton

Source :
```
method pushButton(var eventInfo Event)
procsalir()
endmethod
```

Object : Principal.BotonAyuda

MethodName : pushButton

Source :
```
method pushButton(var eventInfo Event)
helpShowIndex("bueno.hlp")
endmethod
```

Object : Explorar

MethodName : Uses

Source :
```
Uses ObjectPal
 MensajeLiB(NombreFicha String, NombrePagina String, NumerodeObjeto String)
 NombreDePagina(Interfaz UIObject) String
 EsLlamadaVerdadero()
 EsLlamadaFalso()
endUses
```

Object : Explorar

MethodName : Var

Source :
```
Var
 NombreInterfaz String
 Interfaz     UIObject
 Intro,
 Informa1,
 AcercaDe,
 NombreFicha   Form
 tesislib     library
endVar
```

Object : Explorar

MethodName : close

Source :
```
method close(var eventInfo Event)
 if eventInfo.isPreFilter()then
        doDefault
 else
  if Informa1.isAssigned() then
   Informa1.close()
  endif
  if Intro.isAssigned() then
   Intro.close()
  endif
 endif
endmethod
```

Object : Explorar

MethodName : arrive

Source :
```
method arrive(var eventInfo MoveEvent)
if eventInfo.isPreFilter()
        then
                ; este código se ejecuta para cada objeto de la ficha
        else
                ; este código se ejecuta sólo para la ficha en sí
   hidespeedbar()
        MenuExplorar()
endif
endmethod
```

Object : Explorar

MethodName : mouseEnter

Source :
```
method mouseEnter(var eventInfo MouseEvent)
if eventInfo.isPreFilter()  then
                    eventInfo.getTargat(Interfaz)
    nombreInterfaz = Interfaz.Name
    if Interfaz.Name <> Self.Name  AND
       (Interfaz.class = "Field" OR Interfaz.class = "MultiRecord" OR
        Interfaz.class = "Button") than
      tesislib.MensajeLib(NombreFicha.Name,tesislib.NombreDePagina(Interfaz),nombreInterfaz)
    endif
endif
endmethod
```

Object : Explorar

MethodName : mouseExit

Source :
```
method mouseExit(var eventInfo MouseEvent)
if eventInfo.isPreFilter()  then
  eventInfo.gettarget(Interfaz)
  if Interfaz.class <> "Text" AND Interfaz.class <> "Bitmap" then
    message("")
  endif
endif
endmethod
```

Object : Explorar

MethodName : keyPhysical

Source :
```
method keyPhysical(var eventInfo KeyEvent)
var
 laTecla String
 lapagina String
endVar
if eventInfo.isPreFilter()  then
    laPagina = tesislib.NombreDePagina(active)
    laTecla  = eventInfo.vChar()
    switch
      case lapagina = "PaginaValvula" :
      if eventInfo.isAltKeyDown() then
        switch
          case laTecla = "C" or laTecla = "c" :disableDefault
                    BotonContinuar1.pushbutton()
          case laTecla = "Y" or laTecla ="y": disableDefault
                    PaginaValvula.BotonAyuda.pushbutton()
          case laTecla = "S" or laTecla ="s": disableDefault
                    PaginaValvula.BotonSalir.pushbutton()
        endswitch
      endif

         if eventInfo.isFromUi() then
      switch
        case laTecla ="VK_F1": disabledefault
                    PaginaValvula.BotonAyuda.pushbutton()
        case laTecla ="VK_F2": disabledefault
        case laTecla ="VK_F3": disabledefault
        case laTecla ="VK_F4": disabledefault
        case laTecla ="VK_F5": disabledefault
        case laTecla ="VK_F6": disabledefault
        case laTecla ="VK_F7": disableDefault
        case laTecla ="VK_F8": disableDefault
```

```
        case laTecla ="VK_F9": disabledefault
          PaginaValvula.BotonSalir.pushbutton()
        case laTecla ="VK_F10": disabledefault
       endswitch
      endif
    endswitch
endif
endmethod
```

Object : Explorar

MethodName : menuAction

Source :

```
method menuAction(var eventInfo MenuEvent)
 var
  OpcionMenu  String
  acercade    Form
 endVar
 if eventInfo.isPreFilter() then
  doDefault
 else
  OpcionMenu = eventInfo.menuChoice()
  switch
   case OpcionMenu = "&Imprimir..."        :Imprime()
```

```
    case OpcionMenu = "Imp&resora..."      :eventInfo.setId(MenuFilePrinterSetup)
    case OpcionMenu = "&Salir"             :BotonSalir.pushButton()
    case OpcionMenu = "&Primero\tCtrl+F11"  :Primero()
    case Opcionmenu = "&Ultimo\tCtrl+F12"   :Ultimo()
    case opcionmenu = "&Siguiente\tF12"     :Siguiente()
    case opcionMenu = "&Anterior\tF11"      :Anterior()
    case OpcionMenu = "&Indice"            :BotonAyuda.pushButton()
    case OpcionMenu = "&Uso de la Ayuda"    :HELPONHELP()
    case OpcionMenu = "&Información Teórica" :helpShowIndex("infor.hlp")
   endswitch
  endif
endmethod
```

Object : Explorar

MethodName : proc

Source :

```
proc MenuExplorar()
 var
  menuprincipal Menu
  sub1, sub2, sub3, sub4 popUpMenu
 endVar

 sub1.addText("&Imprimir...")
 sub1.addText("Imp&resora...")
 sub1.addSeparator()
 sub1.addText("&Salir")
 menuprincipal.addPopUp("&Archivo",sub1)

 sub2.addText("&Primero\tCtrl+F11")
 sub2.addText("&Ultimo\tCtrl+F12")
 sub2.addText("&Anterior\tF11")
 sub2.addText("&Siguiente\tF12")
 menuprincipal.addPopUp("&Registro",sub2)

 sub3.addText("&Indice")
 sub3.addText("&Uso de la Ayuda")
 sub3.addSeparator()
 sub3.addText("&Información Teórica")
 menuprincipal.addPopUp("&?", sub3)

 menuprincipal.show()
endproc

Proc Imprime()
var
 Informe Report
 DatosInforme ReportPrintInfo
endVar
 if msgQuestion("Imprimir"," ¿ Quiere imprimir ahora ?") = "Yes" then
```

```
    message("Preparando impresión...")
    Informe.print("reporte.rsl")
  endif
  maximize()
endproc


proc Termina()
BotonSalir.pushbutton()
endproc

proc Primero()
active.action(DataBegin)
endproc

proc Ultimo()
active.action(DataEnd)
endproc

proc Siguiente()
active.action(DataNextRecord)
endproc

proc Anterior()
active.action(dataPriorRecord)
endproc

proc Insertar()
active.action(DataBegInEdit)
active.action(DataInsertRecord)
endproc
```

Object : PaginaValvula

MethodName : open

Source :
```
method open(var eventInfo Event)
  if not tesislib.Open("tesisLib",GlobalToDesktop) then
   msgInfo("Fallo","No pude abrir TESISLIB.")
   formReturn(False)
  else
   MenuExplorar()
   NombreFicha.attach()
   hidespeedbar()
   maximize()
   tesislib.esllamadaverdadero()
   if intro.open("intro1.fsl") then
    intro.wait()
    intro.close()
    tesislib.esllamadafalso()
          endif

   endif
endmethod
```

Object : PaginaValvula

MethodName : arrive

Source :
```
method arrive(var eventInfo MoveEvent)
  MenuExplorar()
endmethod
```

Object : PaginaValvula.BotonAyuda

MethodName : pushButton

Source :
```
method pushButton(var eventInfo Event)
helpShowIndex("bueno.hlp")
endmethod
```

Object : PaginaValvula.#Cuadro48.BotonCompre

MethodName : pushButton

Source :
```
method pushButton(var eventInfo Event)
var
  qvar Query
  Nombre1 String
endvar
  DelayScreenUpDates(Yes)
  encontrado.visible = Yes
  encontrado.TableName ="XXX.db"
  dmRemoveTable("Consu2.db")

  Nombre1 = "Com.."
  qvar = query

    Bombas.db |Tipo     | Nombre |
          | Com.. | check |

   endquery
  executeQBE(qvar,"Consu2.db")
```

```
encontrado.Tablename = "Consu2.db"
encontrado.visible = Yes


endmethod
```

Object : PaginaValvula.#Cuadro48.BotonBombas

MethodName : pushButton

Source :

```
method pushButton(var eventInfo Event)
var
 qvar Query
endvar

 DelayScreenUpDates(Yes)
 encontrado.visible = Yes
 encontrado.TableName ="XXX.db"
 dmRemoveTable("Consu2.db")

 qvar = query

  bombas.db |Tipo     | Nombre |
        | Bom..   | check  |

  endquery
 executeQBE(qvar,"Consu2.db")


  encontrado.Tablename = "Consu2.db"
  encontrado.visible = Yes

endmethod
```

Object : PaginaValvula.BotonSalir

MethodName : pushButton

Source :

```
method pushButton(var eventInfo Event)
if msgQuestion("Regresar al Menu Principal","¿Quieres regresar al menu principal?") = "Yes" then

  Formreturn(True)
endif
```

```
endmethod
```

Object : PaginaValvula.BotonContinuar1

MethodName : pushButton

Source :
```
method pushButton(var eventInfo Event)
tesislib.esLlamadaVerdadero()
 if Informa1.open("Informa1.fsl") then
   Informa1.wait()
   Informa1.close()
   tesislib.esLlamadaFalso()
   endif
endmethod
```

Object : PaginaValvula.encontrado.Registroencontrado1

MethodName : canArrive

Source :
```
method canArrive(var eventInfo MoveEvent)
var
 ObjetoDestino  UIObject
 ValorDestino   AnyType
 Pedido,
 PedidoTc       TCursor
endVar
 if self.BlankRecord then
  message("No hay nombres para seleccionar")
  eventInfo.setErrorCode(CanNotArrive)
 else
  eventInfo.getDestination(ObjetoDestino)
  VALV.locate(Objetodestino.Name, ObjetoDestino.Value)
  pedido.open("Bombas.db")
  pedidotc.open("bomba1.db")
  pedidotc.empty()
  pedidotc.edit()
  scan pedido for pedido."Nombre" = Objetodestino.value :
    pedidotc.insertRecord()
    pedidotc.copyrecord(pedido)
  endscan
  pedidotc.endEdit()
  pedido.close()
  pedidotc.close()
 endif
endmethod
```

Object : PaginaValvula.VALV.Registro9.Nombre1

MethodName : arrive

Source :
```
method arrive(var eventInfo MoveEvent)
var
 ObjetoDestino  UIObject
 ValorDestino   AnyType
 Pedido,
 PedidoTc       TCursor
```

```
endVar
 if self.BlankRecord then
  message("No hay nombres para seleccionar")
  eventInfo.setErrorCode(CanNotArrive)
 else
  eventInfo.getDestination(ObjetoDestino)
  VALV.locate(Objetodestino.Name, ObjetoDestino.Value)
  pedido.open("Bombas.db")
  pedidotc.open("Bomba1.db")
  pedidotc.empty()
  pedidotc.edit()
  scan pedido for pedido."Nombre" = Objetodestino.value :
   pedidotc.insertRecord()
   pedidotc.copyrecord(pedido)
  endscan
  pedidotc.endEdit()
  pedido.close()
  pedidotc.close()
 endif
endmethod
```

Object : Explorar

MethodName : Uses

Source :
```
Uses ObjectPal
 MensajeLIB(NombreFicha String, NombrePagina String, NumerodeObjeto String)
 NombreDePagina(Interfaz UIObject) String
 EsLlamadaVerdadero()
 EsLlamadaFalso()
endUses
```

Object : Explorar

MethodName : Var

Source :
```
Var
 NombreInterfaz String
 Interfaz     UIObject
 Intro,
 Informa,
 AcercaDe,
 NombreFicha   Form
 tesislib     Library
endVar
```

Object : Explorar

MethodName : close

Source :
```
method close(var eventInfo Event)
 if eventInfo.isPreFilter()then
        doDefault
 else
  if Informa.isAssigned() then
   Informa.close()
  endif
  if Intro.isAssigned() then
   Intro.close()
  endif
 endif
endmethod
```

Object : Explorar

MethodName : arrive

Source :
```
method arrive(var eventInfo MoveEvent)
if eventInfo.isPreFilter()
        then
                ; este código se ejecuta para cada objeto de la ficha
        else
                ; este código se ejecuta sólo para la ficha en sí
   hidespeedbar()
        MenuExplorar()
endif
endmethod
```

Object : Explorar

MethodName : mouseEnter

Source :
```
method mouseEnter(var eventInfo MouseEvent)
if eventInfo.isPreFilter()  then
                    eventInfo.getTarget(Interfaz)
   nombreInterfaz = Interfaz.Name
   If Interfaz.Name <> Self.Name  AND
     (Interfaz.class = "Field" OR Interfaz.class = "MultiRecord" OR
      Interfaz.class = "Button") then
     tesislib.MensajeLib(NombreFicha.Name,tesislib.NombreDePagina(Interfaz),nombreInterfaz)
   endif
endif
endmethod
```

Object : Explorar

MethodName : mouseExit

Source :
```
method mouseExit(var eventInfo MouseEvent)
If eventInfo.isProFilter()  then
  eventInfo.gettarget(Interfaz)
  if Interfaz.class <> "Text" AND Interfaz.class <> "Bitmap" then
    message("")
  endif
endif
endmethod
```

Object : Explorar

MethodName : keyPhysical

Source :
```
method keyPhysical(var eventInfo KeyEvent)
var
 laTecla String
 lapagina String
endVar
if eventInfo.isPreFilter()  then
 laPagina = tesislib.NombreDePagina(active)
 laTecla  = eventinfo.vChar()
 switch
  case lapagina = "PaginaValvula" :
   if eventInfo.isAltKeyDown() then
    switch
     case laTecla = "C" or laTecla = "c" :disableDefault
      BotonContinuar.pushbutton()
     case laTecla = "Y" or laTecla ="y": disableDefault
      PaginaValvula.BotonAyuda.pushbutton()
     case laTecla = "S" or laTecla ="s": disableDefault
      PaginaValvula.BotonSalir.pushbutton()
    endswitch
   endif
   if eventInfo.isFromUI() then
    switch
     case laTecla ="VK_F1": disabledefault
      PaginaValvula.BotonAyuda.pushbutton()
     case laTecla ="VK_F2": disabledefault
     case laTecla ="VK_F3": disabledefault
     case laTecla ="VK_F4": disabledefault
     case laTecla ="VK_F5": disabledefault
```

```
        case laTecla ="VK_F6": disabledefault
        case laTecla ="VK_F7": disableDefault
        case laTecla ="VK_F8": disableDefault
        case laTecla ="VK_F9": disabledefault
          PaginaValvula.BotonSalir.pushbutton()
        case laTecla ="VK_F10": disabledefault
       endswitch
      endif
    endswitch
  endif
endmethod
```

Object : Explorar

MethodName : menuAction

Source :

```
method menuAction(var eventInfo MenuEvent)
 var
  OpcionMenu  String
  acercade   Form
 endVar
 if eventInfo.isPreFilter() then
  doDefault
 else
  OpcionMenu = eventInfo.menuChoice()
  switch
   case OpcionMenu = "&Imprimir..."        Imprime()
```

```
    case OpcionMenu = "Imp&resora..."      :eventInfo.setId(MenuFilePrinterSetup)
    case OpcionMenu = "&Salir"             :BotonSalir.pushButton()
    case OpcionMenu = "&Primero\tCtrl+F11" :Primero()
    case Opcionmenu = "&Ultimo\tCtrl+F12"  :Ultimo()
    case opcionmenu = "&Siguiente\tF12"    :Siguiente()
    case opcionMenu = "&Anterior\tF11"     :Anterior()

    case OpcionMenu = "&Indice"            :BotonAyuda.pushButton()
    case OpcionMenu = "&Uso de la Ayuda"   :HELPONHELP()
    case OpcionMenu = "&Información Teórica" :helpShowIndex("infor.hlp")

   endswitch
   endif
  endmethod
```

Object : Explorar

MothodName : proc

Source :

```
proc MenuExplorar()
 var
  menuprincipal Menu
  sub1, sub2, sub3, sub4 popUpMenu
 endVar

 sub1.addText("&Imprimir...")
 sub1.addText("Imp&resora...")
 sub1.addSeparator()
 sub1.addText("&Salir")
 menuprincipal.addPopUp("&Archivo",sub1)

 sub2.addText("&Primero\tCtrl+F11")
 sub2.addText("&Ultimo\tCtrl+F12")
 sub2.addText("&Anterior\tF11")
 sub2.addText("&Siguiente\tF12")
 menuprincipal.addPopUp("&Registro",sub2)

 sub3.addText("&Indice")
 sub3.addText("&Uso de la Ayuda")
 sub3.addSeparator()
 sub3.addText("&Información Teórica")
 menuprincipal.addPopUp("&?", sub3)

 menuprincipal.show()
endproc

proc Imprime()
var
 Informe Report
```

```
 DatosInforme ReportPrintInfo
endVar
 if msgQuestion("Imprimir"," ¿ Quiere imprimir ahora ?") = "Yes" then
    message("Preparando impresión...")
    Informe.print("reporte.rsl")
 endif
 maximize()
endproc


proc Termina()
BotonSalir.pushbutton()
endproc

proc Primero()
active.action(DataBegin)
endproc

proc Ultimo()
 active.action(DataEnd)
endproc

proc Siguiente()
 active.action(DataNextRecord)
endproc

proc Anterior()
 active.action(dataPriorRecord)
endproc

proc Insertar()
 active.action(DataBeginEdit)
 active.action(DataInsertRecord)
endproc
```

Object : PaginaValvula

MethodName : open

Source :
```
method open(var eventInfo Event)
 if not tesislib.Open("tesisLib",GlobalToDesktop) then
   msgInfo("Fallo","No pude abrir TESISLIB.")
   formReturn(False)
 else
  MenuExplorar()
  NombreFicha.attach()
  hidespeedbar()
  maximize()
  tesislib.esllamadaverdadero()
  if intro.open("intro.fsl") then
    intro.wait()
    intro.close()
    tesislib.esllamadafalso()
          endif
 endif
endmethod
```

Object : PaginaValvula

MethodName : arrive

Source :
```
method arrive(var eventInfo MoveEvent)
 MenuExplorar()
endmethod
```

Object : PaginaValvula.BotonAyuda

MethodName : pushButton

Source :
```
method pushButton(var eventInfo Event)
helpShowIndex("bueno.hlp")
endmethod
```

Object : PaginaValvula.#Cuadro17.Retencion

MethodName : pushButton

Source :
```
method pushButton(var eventInfo Event)
var
 qvar Query
 Nombre1 String
endvar

 DelayScreenUpDates(Yes)
 encontrado.visible = Yes
 encontrado.TableName ="XXX.db"
 dmRemoveTable("Consu1.db")
 Nombre1 = "Re.."
 qvar = query

  valv.db |Tipo     | Nombre |
       | ~Nombre1 | check  |

  endquery
```

```
  executeQBE(qvar,"Consu1.db")
  encontrado.Tablename = "Consu1.db"
  encontrado.visible = Yes

endmethod
```

Object : PaginaValvula.#Cuadro17.Gato

MethodName : pushButton

Source :

```
method pushButton(var eventInfo Event)
var
  qvar Query
  Nombre1 String
endvar

  DelayScreenUpDates(Yes)
  encontrado.visible = Yes
  encontrado.TableName ="XXX.db"
  dmRemoveTable("Consu1.db")
  Nombre1 = "Ga.."

  qvar = query

   valv.db |Tipo     | Nombre |
        | ~Nombre1 | check  |

   endquery
  executeQBE(qvar,"Consu1.db")
  encontrado.Tablename = "Consu1.db"
  encontrado.visible = Yes


endmethod
```

Object : PaginaValvula.#Cuadro17.Esfericas

MethodName : pushButton

Source :

```
method pushButton(var eventInfo Event)
var
  qvar Query
  Nombre1, Cadena1, Clasi1, Esta1 String
```

```
endvar

 DelayScreenUpDates(Yes)
 encontrado.visible = Yes
 encontrado.TableName ="XXX.db"
 dmRemoveTable("Consu1.db")
 Nombre1 = "Esf.."
 qvar = query

  valv.db |Tipo     | Nombre |
       | ~Nombre1 | check  |

  endquery
 executeQBE(qvar,"Consu1.db")
 encontrado.Tablename = "Consu1.db"
 encontrado.visible = Yes

endmethod
```

Object : PaginaValvula.#Cuadro17.Bola

MethodName : pushButton

Source :

```
method pushButton(var eventInfo Event)
var
 qvar Query

 Nombre1  String

endvar

 DelayScreenUpDates(Yes)
 encontrado.visible = Yes
 encontrado.TableName ="XXX.db"
 dmRemoveTable("Consu1.db")


 Nombre1 = "Bo.."

 qvar = query

  valv.db |Tipo     | Nombre |
       | ~Nombre1 | check  |

  endquery
 executeQBE(qvar,"Consu1.db")
 encontrado.Tablename = "Consu1.db"
 encontrado.visible = Yes
endmethod
```

Object : PaginaValvula.#Cuadro17.#Cuadro34.Globo

MethodName : pushButton

Source :

```
method pushButton(var eventInfo Event)
var
```

```
qvar Query
Nombre1 String
endvar

DelayScreenUpDates(Yes)
encontrado.visible = Yes
encontrado.TableName ="XXX.db"
dmRemoveTable("Consu1.db")


Nombre1 = "Gl.."
qvar = query

 valv.db |Tipo     | Nombre |
      | ~Nombre1 | check  |

 endquery
executeQBE(qvar,"Consu1.db")
encontrado.Tablename = "Consu1.db"
encontrado.visible = Yes


endmethod
```

Object : PaginaValvula.#Cuadro17.#Cuadro26.Compuerta

MethodName : pushButton

Source :

```
method pushButton(var eventInfo Event)
var
 qvar Query
 Nombre1 String
endvar

 DelayScreenUpDates(Yes)
 encontrado.visible = Yes
 encontrado.TableName ="XXX.db"
 dmRemoveTable("Consu1.db")

 Nombre1 = "Com.."
 qvar = query

  valv.db |Tipo     | Nombre |
       | ~Nombre1 | check  |

  endquery
 executeQBE(qvar,"Consu1.db")
 encontrado.Tablename = "Consu1.db"
 encontrado.visible = Yes


endmethod
```

Object : PaginaValvula.BotonSalir

MethodName : pushButton

Source :

```
method pushButton(var eventInfo Event)
```

```
if msgQuestion("Regresar al Menu Principal","¿Quieres regresar al menu principal?") = "Yes" then

  Formreturn(True)
endif
endmethod
```

Object : PaginaValvula.BotonContinuar

MethodName : pushButton

Source :
```
method pushButton(var eventInfo Event)
tesislib.esLlamadaVerdadero()
 if  Informa.open("Informa.fsl") then
    Informa.wait()
    Informa.close()
    tesislib.esLlamadaFalso()
    endif
endmethod
```

Object : PaginaValvula.encontrado.Registroencontrado

MethodName : canArrive

Source :
```
method canArrive(var eventInfo MoveEvent)
var
 ObjetoDestino  UIObject
 ValorDestino   AnyType
 Pedido,PedidoTc TCursor
endVar
 if self.BlankRecord then
  message("No hay nombres para seleccionar")
  eventInfo.setErrorCode(CanNotArrive)
 else
  eventInfo.getDestination(ObjetoDestino)
  VALV.locate(Objetodestino.Name, ObjetoDestino.Value)
  pedido.open("Valv.db")
  pedidotc.open("informa2")
  pedidotc.empty()
  pedidotc.edit()
  scan pedido for pedido."Nombre" = Objetodestino.value :
    pedidotc.insertRecord()
    pedidotc.copyrecord(pedido)
  endscan
  pedidotc.endEdit()
  pedido.close()
  pedidotc.close()
 endif
endmethod
```

Object : PaginaValvula.VALV.Registro9.Nombre

MethodName : arrive

Source :
```
method arrive(var eventInfo MoveEvent)
var
 ObjetoDestino  UIObject
 ValorDestino   AnyType
 Pedido,
```

```
 PedidoTc     TCursor
endVar
 if self.BlankRecord then
   message("No hay nombres para seleccionar")
   eventInfo.setErrorCode(CanNotArrive)
 else
   eventInfo.getDestination(ObjetoDestino)
   VALV.locate(Objetodestino.Name, ObjetoDestino.Value)
   pedido.open("Valv.db")
   pedidotc.open("Informa2.db")
   pedidotc.empty()
   pedidotc.edit()
   scan pedido for pedido."Nombre" = Objetodestino.value :
     pedidotc.insertRecord()
     pedidotc.copyrecord(pedido)
   endscan
   pedidotc.endEdit()
   pedido.close()
   pedidotc.close()
 endif
endmethod
```