



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE ESTUDIOS SUPERIORES  
ARAGON**



**“AUTOMATIZACIÓN DE FORMATOS PARA VALES DE LA DIVISIÓN  
DE INGENIERÍAS CIVIL Y GEOMÁTICA”**

**T E S I S  
QUE PARA OBTENER EL TITULO  
DE  
INGENIERO EN COMPUTACION**

**PRESENTAN:**

**MEDINA QUINTERO FERNANDO**

**VELASCO MARTINEZ PEDRO**

**DIRECTOR DE TESIS: M.I. JUAN CARLOS ROA BEIZA**



Universidad Nacional  
Autónoma de México



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

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

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

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

## **AGRADECIMIENTOS**

Esta tesis, si bien ha requerido de esfuerzo y mucha dedicación por parte de los autores y su director de tesis (Ing. Juan Carlos Roa Beiza), no hubiese sido posible su finalización, sin la cooperación desinteresada de todas y cada una de las personas que a continuación citaré y muchas de las cuales han sido un apoyo fundamental en momentos decisivos de mi vida.

Detrás de la preparación de este trabajo de tesis, está la dedicación de quienes tuvieron el trabajo de revisar este escrito. A *Flores Díaz Imelda de la Luz, Vázquez Morales Rodolfo, Pérez Gastaldi Juan y García Guzmán Enrique*, mis profesores y ahora mi honorable jurado, para todos ellos mi más profundo agradecimiento ya que con sus observaciones y comentarios encaminaron este trabajo a un mejor termino.

Quiero dar gracias a *Dios*, por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente y por haber puesto en mi camino a todas esas personas que han sido una bendición en mi vida.

A mis padres, por su amor, comprensión y apoyo sin condición ni medida. A *Boni* Mi Padre, por guiarme, ser mi ejemplo e inspiración el mejor héroe que pude desear y el mejor consejero que puedo tener. A *Tere* mi bella Madre por sus cuidados, atenciones, su amor incondicional logran motivarme a salir siempre adelante. *Armando* Mi Hermano la otra parte de mí que siempre llevo conmigo, mi primer mejor amigo en la vida y mi cómplice de infancia los quiero tanto!

A mi familia por estar conmigo en todo momento, y más aún, por haberme dejado entre tantas cosas la mejor herencia que pude haber recibido, “La educación y ahora una carrera”, las mejores armas para enfrentar este camino profesionalmente hablando. Día a día, agradezco a Dios el haberme dado la familia que tengo, a cada paso que doy los llevo conmigo esperando hacerlos sentir orgullosos. Gracias infinitas..

A mi compañero y entrañable amigo Pedro por haber aceptado este reto conmigo, llegar al punto que estamos ahora y darnos cuenta que para los retos y obstáculos en la vida, solo hay que animarse a luchar por lo que se quiere.

A la UNAM, la Facultad de Ingeniería de Aragón, El palacio de Minería, CCH-Vallejo y todas las instituciones relacionadas con esta Máxima Casa de Estudios en las que tuve oportunidad de estar, por darme conocimientos, el ambiente y la oportunidad de convivir con gente tan diversa y admirable entre profesores y alumnos, me dieron ánimos y ambición de prepararme y siempre aspirar a más.

También quisiera hacer parte de mi agradecimiento a todas y cada una de las personas que al paso de mi vida han dejado una huella muy grande y ahora son parte de mí, en general a toda esa gente linda que con sus sonrisas y su luz llenan de vida mis días, que me enseñaron a ver las cosas en perspectiva, gracias por sus oraciones y todos sus buenos deseos.

A todos mi mayor reconocimiento y gratitud.

*Fernando Medina Quintero*

## **OBJETIVO**

Implementación de un sistema de automatización para la generación de vales de la División de Ingeniería Civil y Geomática, con el fin de facilitar la tarea y tener un mayor control de las transacciones tanto de almacén como de compra, mediante una interfaz gráfica orientada hacia el usuario final.

En las divisiones de la Facultad de Ingeniería para poder llevar a cabo una orden de compra y almacén se requiere acudir al departamento de compras con previa elaboración manual del formato correspondiente y posteriormente se acude con el personal del almacén para hacer entrega del formato el cual contiene el pedido de cada jefe de departamento.

Con el fin de tener un control del manejo de la información, el sistema deberá ser capaz de almacenar, generar el formato de compras y almacén según sea el caso, mostrando los datos requeridos dependiendo del perfil del usuario final.

El sistema deberá tener una interfaz amigable para su uso, de tal modo que permita visualizar los módulos que correspondan desde diferentes pantallas pero desde una misma interfaz, como son las administraciones de: generación de vales, consultas, altas, bajas y reportes.



## **DEFINICIÓN DEL PROBLEMA**

Hoy en día el tiempo es un factor preponderante en lo que respecta a la calidad en el servicio que brindan múltiples instituciones, por lo que resulta necesario el automatizar ciertas tareas para dar rapidez a los trámites, asimismo es de vital importancia el acceso así como la disponibilidad de la información para la toma de decisiones.

En la investigación de campo se ha encontrado que los usuarios del almacén se quejan constantemente de la ineficiencia del proceso de generación del vale correspondiente a la operación, asimismo se pone de manifiesto la necesidad de los jefes de departamento de tener acceso a información estadística en cualquier momento y desde cualquier lugar para la toma de decisiones. En más de la mitad de ocasiones los usuarios del almacén tienen que esperar entre 30 y 40 minutos para obtener un vale de almacén, de la misma forma pasa con la obtención de un vale de compra, asimismo el sujetarse a la disponibilidad del personal encargado de la elaboración de dicha tarea. Por otro lado si algún jefe de departamento requiere información referente a los pedidos de algún usuario en especial o datos estadísticos de la división en general es necesario acudir con la persona encargada de dicha tarea y de manera manual hacer un conteo de los datos requeridos.

El usuario debe ser capaz de generar sus propios vales con el formato requerido en cualquier momento y desde el lugar en el que se encuentre para facilitar dicho trámite, de igual forma contar con herramientas que faciliten el llenado de dichos formatos.

Las personas para las cuales les sea útil la información estadística deberían poder tener acceso a reportes gráficos y numéricos para realizar un análisis del consumo en áreas o individuos para valorar desempeño y tomar decisiones.

A partir de un sistema que genere, almacene, organice y difunda información de una manera sencilla, de modo que satisfaga las necesidades de los diferentes usuarios, para que de esta manera se conjunte un sistema de

difusión de información, sencillo, accesible desde cualquier computadora conectada a la red local de la división o desde cualquier punto de la red.

Se pretende que el sistema cubra los siguientes puntos:

- Que toda la información se encuentre estructurada, siempre accesible y sirva de apoyo para otras áreas de la división.
- Información accesible desde diversos sitios geográficamente distantes de la División de Ingenierías Civil y Geomática.
- Actualización de la información en tiempo real con diferentes plataformas y equipos.
- Acceso a la información que permita aumentar la rapidez y la fiabilidad para la toma de decisiones.
- Un ambiente amigable y sin complicaciones de configuración.

## **RESULTADOS ESPERADOS**

- Implantar un sistema de control de vales de almacén y de compra, base de datos, para el control de las transacciones de los diferentes vales generados.
- Que el jefe de departamento pueda realizar consultas, generación de vales tanto de almacén como de compra, así como altas, bajas, cambios y movimientos en tiempo real.
- Que los usuarios finales puedan realizar generación de vales de almacén y compra.
- Monitorear el movimiento diario de altas, bajas y reportes que se hagan diariamente o por períodos.
- Poder obtener reportes diarios, semanales y mensuales de los movimientos hechos por el personal jerárquicamente.
- Poder descargar los resultados en forma gráfica, texto o ambas.

## **METODOLOGÍA A UTILIZAR**

Con base en las necesidades especificadas para el proyecto, nuestro sistema será implementado para un ambiente de trabajo en red. Permitirá accesos al sistema desde cualquier punto de la red y desde afuera mediante enlaces VPN será consistente y permitirá la disponibilidad de la información en tiempo real y en cualquier equipo autorizado.

Dando un enfoque Top-Down el proceso que llevaremos a cabo va a permitir que el problema planteado anteriormente se pueda descomponer en niveles o pasos sucesivos de refinamiento, esto se logrará mediante la conexión entre las sucesivas etapas de estructuración de modo que exista una relación entre ellas mediante entradas y salidas de información.

Para implementar nuestro sistema usaremos la metodología de programación modular (Top-Down).

La implementación de la base de datos será con base en un modelo entidad relación normalizada que nos permitirá manejar los datos en forma óptima.

## **INVENTARIO DE MATERIAS**

- COMPUTADORAS Y PROGRAMACIÓN
- ESTRUCTURAS DE DATOS
- BASES DE DATOS
- INGENIERIA DE PROGRAMACIÓN
- COSTOS Y EVALUACIÓN DE PROYECTOS
- REDES DE COMPUTADORAS
- INVESTIGACIÓN DE OPERACIONES
- SISTEMAS OPERATIVOS

## INDICE TEMÁTICO

1. Planteamiento del Problema .....	2
1.1. Introducción	
1.2. Proceso operativo de la obtención de insumos en el almacén	
1.3. Análisis de la problemática dentro de los procesos de requerimientos	
1.4. Requerimientos del sistema de vales, software y hardware	
2. Marco teórico.....	32
2.1. Base de datos relacionales	
2.2. Características, ventajas y desventajas de arquitectura de red	
2.3. Características, ventajas y desventajas de aplicaciones WEB	
2.4. Características, ventajas y desventajas de aplicaciones PHP	
2.5. Características, ventajas y desventajas de aplicaciones Apache	
2.6. Características, ventajas y desventajas de aplicaciones MySQL	
2.7. Características, ventajas y desventajas de aplicaciones HTML Java Script	
2.8. Características, ventajas y desventajas de aplicaciones cliente servidor	
3. Análisis de planteamiento del problema.....	96
3.1. Contexto situacional del problema	
3.2. Requerimientos generales y particulares de información	
3.3. Recopilación y análisis de información actual	
3.4. Identificación de los posibles módulos de sistema	
3.5. Comparación de las herramientas de solución	
4. Implementación y personalización del sistema .....	126
4.1. Elección de la metodología de desarrollo	
4.2. Diagramación	
4.2.1. Diagrama de flujo y procesos	
4.2.2. Diagramación de entidad-relación	
4.2.3. Diccionario de Datos	
4.2.4. Normalización	
4.3. Diseño y Construcción del back-end	
4.4. Diseño y construcción del front-end	
4.5. Integración y pruebas del sistema	
4.6. Obtención de reportes	

## Capítulo 1

### Planteamiento del Problema

## INTRODUCCION

Nos asombra la velocidad con que avanzan hoy en día las instituciones, en sus técnicas de administración. Esta aceleración ha sido un paso obligado por la creciente demanda masiva de artículos para un consumidor cada vez más exigente en la calidad y rapidez en la obtención de dichos artículos.

En la administración, con el mismo paso acelerado está el desarrollo intelectual y cultural del hombre, está su capacitación como dirigente profesional y está su propio bienestar para integrarse al ritmo que sigue la prosperidad de la dependencia de la cual es dirigente.

La función del almacén se integra en el concepto de gestión. La gestión se define como la administración y dirección de una empresa atendiendo a una serie de procedimientos y reglas que, mediante la coordinación y organización de los recursos disponibles, persigue cumplir los objetivos prefijados de la manera más eficientemente posible. En el caso de la gestión de almacén su objetivo no es solamente el citado de salvaguarda de los materiales allí depositados sino, muy especialmente, la de actuar como regulador entre los ritmos de salida de unas fases (ejemplo: compras o salidas de líneas de producción) y los de entrada de las siguientes (ejemplo: entradas en líneas de producción o ventas).

Los inventarios o stocks, definidos como una provisión de materiales con el objeto de facilitar la continuidad y la satisfacción de los pedidos de consumidores y clientes, se presentan prácticamente en cualquier organización, en particular, en las empresas industriales, sean éstas pequeñas, medianas o grandes.

El almacén es una unidad de servicio en la estructura orgánica y funcional de una empresa comercial o industrial, con objetivos bien definidos de resguardo, custodia, control y abastecimiento de materiales y productos.

## PLANTEAMIENTO DEL PROBLEMA

La manera de organizar y administrar el departamento de almacenes depende de varios factores, tales como el tamaño, plan de organización de la institución, el grado de centralización deseado y la variedad de productos. Sin embargo, para proporcionar un servicio eficiente, las siguientes funciones son comunes a todo tipo de almacenes:

- recepción de materiales en el almacén
- registro de entradas y salidas del almacén
- almacenamiento de materiales
- mantenimiento de materiales y del almacén
- despacho de materiales
- coordinación del almacén con los departamentos de control de inventarios y de contabilidad

El almacén de la división de ingenierías civil y geomáticas es el encargado de surtir de material a los jefes pertenecientes a esta división. Su gama de productos incluye desde material de oficina cintas hojas hasta material de limpieza jabón etc.

Para las empresas como las instituciones, es de suma importancia tener en sus almacenes un control ordenado de todas las operaciones que ahí se suscitan, como la recepción, entrega, control del material; el tener vigilados los inventarios, tanto las relaciones internas (contabilidad, entrega de material), así como las externas (proveedores, compras de material), esto facilitara y dará un ambiente transparente, optimizando el funcionamiento del mismo.

El almacenista lleva acabo el control de la entrada y salida de material basado en los vales en papel que recibe por cada transacción. Estos vales a su vez son expedidos por el personal de la secretaria académica y estos son sellados como método de validación.

El jefe de almacén recibe el vale y permite la transacción de surtir el material. Así como validar si hay en existencia si no para re abastecer el almacén.

## INVENTARIOS

Los inventarios presentan una proporción de los activos/artículos de un almacén u o empresa, y estos son almacenados esperando su uso. El control y la constante actualización de un inventario es un problema muy común en un almacén, es por eso que los procesos administrativos de compra u obtención de materiales que reduzcan el tiempo de la transacción representaran un ahorro de dinero para la empresa.

Al manejarse una variedad importante de productos, se requiere de un sistema que automatizara el control de sus inventarios con el fin de conocer de manera oportuna y veraz las cantidades de los materiales disponibles de los cuales podrá disponer para surtir a los profesores.

Por otra parte, la acumulación de materiales en los almacenes crea la necesidad de cuanto y cuando es necesario hacer una compra de materiales para abastecer el almacén. La gestión de la información indicara los momentos justos de reabastecer el almacén.

El sistema generador de vales de compra y de almacén permitirá a la comunidad (jefes de área de la división), del área de ingeniería civil y geomática, realizar sus solicitudes de materiales que serán proporcionadas por el área de almacén así como al jefe de almacén ver qué porcentaje de material se tiene para surtir evitando la escasez del mismo.

Este sistema de vales debe caracterizarse por permitir al usuario un fácil acceso, sin importar su ubicación geográfica o características del equipo (sistema operativo, navegador versión software, etc.). En este aspecto, las características de disponibilidad y accesibilidad, son propias de las aplicaciones web que las convierte en la solución ideal para el proceso de solicitud de vales, lo cual beneficiaria al ahorro de personal y el tiempo de ejecución de las transacciones por el importante número de profesores de estas áreas que utilizan el servicio de almacén.

## PLANTEAMIENTO DEL PROBLEMA

Se plantea un sistema con base en una PC (servidor) con accesos remoto, para automatizar los procesos de captura y elaboración de vales que se realizan como parte de la obtención o compra de material del almacén.

Si bien algunas características del sistema que actualmente se maneja dependen netamente de la presencia, del personal de almacén y secretarias, también es cierto que existen ciertas cualidades deseables para toda aplicación independientemente de la presencia del personal. De dichas características, cabe mencionar 2 destacables: facilidad manejo y accesibilidad.

La facilidad de manejo ayuda a la iteración del usuario con el servicio, esto es, que la aplicación web de este tipo debe ser lógica con una interface amigable y sencilla de utilizar, de tal manera que los usuarios puedan realizar la transacción ellos mismos de manera eficaz.

Todo sistema para solicitud de vales debe estar pensado en base a la seguridad de la información manejada, evitando saqueos o pérdida de material. Por consiguiente el sistema debe incluir un método de autenticación y validación para disminuir riesgos.

## **DEFINICIÓN.**

Al inicio de este proyecto LA DIVISIÓN DE INGENIERÍAS CIVIL Y GEOMÁTICAS utilizaba un sistema generador de vales utilizando hojas y papel para solicitar el material. Dicho sistema presenta ciertos problemas debidos principalmente a que depende tanto del personal académico y del encargado de almacén que genera los vales y la persona que sella las hojas autorizando las transacciones para ir con el almacenista y obtener el material. Este servicio se presta únicamente a los jefes de cada área puesto que estos hacen un estimado del material que necesitaran al mes, y así repartir el material.

Debido a la carga de trabajo y a la alta demanda de material en LA DIVISIÓN DE INGENIERÍAS CIVIL Y GEOMÁTICAS, se ve en la imperiosa necesidad de



mejorar sustancialmente el tiempo de respuesta y la eficacia del proceso de elaboración de vales que son necesarios para llevar a cabo el proceso de obtención de materiales de almacén.

El proyecto de Tesis del sistema generador de vales de la división resulta valioso pues constituye un claro ejemplo que demuestra que las aplicaciones Web son una alternativa totalmente idónea, al tiempo que permite vislumbrar los beneficios como accesibilidad, facilidad de manejo y seguridad que son características de estas tecnologías Web.

En base a lo anterior es debemos considerar a los usuarios como no expertos en informática y la consulta para estos podrá realizarse de manera local o remota para no depender de los intermediarios, mediante una interfaz amigable e intuitiva que deberá soportarse bajo un sistema operativo de fácil manejo.

### **OBJETIVO**

El proyecto de tesis tiene como meta la implementación de una aplicación web cliente servidor basada en PHP, MySQL y Java para la elaboración de vales para el área del almacén de la división de ingenierías civil y geománticas.

También el objetivo es instaurar un sistema de almacenamiento y difusión de información que le dé al almacén ventajas sobre el anterior tales como:

El sistema generador de vales resultante en esta tesis será desarrollado teniendo en consideración los siguientes objetivos específicos:

- Crear un sistema capaz de dar servicio a los profesores que se conecten al sistema simultáneamente. Para ello se utilizan aplicaciones cliente servidor para reducir el gasto de los recursos del servidor de tal manera que soporte múltiples accesos.

## PLANTEAMIENTO DEL PROBLEMA

- Crear una interfaz amigable y de baja utilización del ancho de banda en sus comunicaciones. Lo cual se logra por las aplicaciones cliente-servidor.
- Crear una aplicación altamente accesible que ofrezca compatibilidad con las últimas versiones de los principales navegadores Web.
- Crear una aplicación capaz de presentar la información en formato PDF **Portable Document Format (Formato de Documento Portátil)** para su impresión. Esto se lograra creando una interfaz gráfica sencilla, que cuenta con áreas definidas en las que el usuario puede visualizar la información de los diversos materiales del almacén, terminado el proceso la información sea actualiza en la base de datos para llevar un conteo del material disponible en el almacén en tiempo real.

### **1.2 PROCESO OPERATIVO DE LA OBTENCIÓN DE INSUMOS EN EL ALMACÉN**

Para la obtención de insumos que se encuentran en el almacén perteneciente a la División de Ingenierías Civil y Geomática es necesario pasar por tres etapas básicas o flujos de trabajo que cubren todos los aspectos relacionados con cualquier petición hacía el almacén; además de que el proceso de adquisición y su administración requieren de algunos pasos previos esenciales para el procedimiento de compras externo que son los siguientes:

- Determinación de la existencia de una necesidad.
- Precisar el tipo de artículo que se necesita, así como la cantidad que se debe pedir.
- Tener y seleccionar algunas fuentes posibles de abastecimiento.
- Analizar las opciones y hacer la petición del pedido.
- Dar seguimiento a la orden.
- Recibir e inspeccionar los artículos pedidos.

- Revisar la factura y pago al proveedor del almacén.
- Llevar los registros apropiados del abastecimiento externo del almacén.

Ahora se presentan las etapas que cubren las peticiones internas hacia el almacén de la División de Ingenierías Civil y Geomática:

### **Flujo Físico**

Dada las peticiones previas o recolectadas con anterioridad y en vista de que en el almacén en ciertas ocasiones no se tiene todo o algunos insumos requeridos por los administrativos de la División de Ingenierías Civil y Geomática es necesario por tanto realizar encargos a un proveedor externo el cual en su vehículo se dirige al almacén donde con previa solicitud externa realiza el proceso de descarga de la mercancía donde se inicia el desarrollo de clasificación de los diferentes insumos solicitados para que de esa forma sea mucho más fácil la localización de los mismos cuando sean requeridos por algún solicitante de la División de Ingenierías Civil y Geomática.

### **Flujo de Información**

El área de administración de materiales (almacén) requiere de una gran diversidad de procedimientos estandarizados de operación que habrán de utilizarse en las tareas diarias normales esto es esencial ya que se manejan una gran cantidad de artículos y por ende de un gran valor monetario, por otra parte, la necesidad de mantener un control adecuado puesto que las grandes consecuencias de operaciones no satisfactorias redundarán en resultados desfavorables.

Para evitar esto es necesario generar una relación de la carga enviada de los diferentes solicitantes de la División de Ingenierías Civil y Geomática, esta parte se realiza de manera interna en el almacén de insumos.

Cuando la mercancía llega a los solicitantes, la relación de peticiones y la documentación (vale de almacén) se imprimen y se anexan para llevar una relación de quién y qué se pide más del almacén, todo esto vía papelería.

Se tienen entonces una asignación de envíos para la División de Ingenierías Civil y Geomática de la cual se hace entrega posterior a cada solicitante

## PLANTEAMIENTO DEL PROBLEMA

mediante Jefatura con documentación requerida y se notifica cualquier incidencia, haciendo una notificación de destino de los envíos que se recibirán. Si llegase a existir alguna incidencia durante el proceso se realiza una notificación hacia el almacén y se procede a ver la forma en que se va a resolver dicho suceso. La información que se entrega se archiva en carpetas separándolas por año y mes.

### **Flujo Administrativo**

Cada mes en fechas señaladas previamente en un calendario es que se hace saber cuándo es posible hacer peticiones de insumos hacia el almacén, cada jefe de departamento está consciente de ello.

Si se tiene una petición que llevarse a cabo se recopila la información requerida la cual toma un promedio de hasta 40 minutos para obtener un vale de almacén, de ahí se tiene que sujetar a la disponibilidad del personal encargado de efectuar dicha tarea.

Con la firma y sello de recibido hacia la División de Ingenierías Civil y Geomática es que se hace entrega en Jefatura de los insumos adquiridos donde posteriormente se hará llegar mediante el Departamento correspondiente a los diferentes solicitantes.

### **Determinando la existencia de una necesidad**

Teniendo en cuenta los flujos anteriores toda petición dirigida hacia el almacén debe tener su origen al reconocerse la existencia definitiva de una necesidad de algún solicitante de la División de Ingenierías Civil y Geomática. Las personas encargadas y responsables, en su mayoría las secretarías de los jefes de departamento, deberán saber cuáles son las necesidades individuales de su departamento correspondiente: qué se necesita, cuánto se necesita y cuándo se tendrá tal necesidad. Esto habrá de conducir a una requisición al departamento de almacén. Ocasionalmente, estas demandas se podrán satisfacer transfiriendo material sobrante de otros departamentos. Tarde o temprano, por supuesto, surgirá la necesidad de efectuar un vale de almacén para obtener el material que se va a necesitar.

Para saber el origen de las peticiones al almacén se tiene asignado un número de departamento de tal forma que mediante dicho código se tiene identificado a

los departamentos que componen a la División de Ingenierías Civil y Geomática.

### **Ordenes de emergencia y urgentes**

Con poca frecuencia se reciben peticiones que indican ser "urgentes" aunque no siempre se pueden suprimir las órdenes urgentes pero pueden surgir emergencia que justifiquen su empleo, aunque generalmente esto no es muy común ya que como se mencionó con anterioridad para evitar este tipo de sucesos existe una calendarización que precisamente su objetivo es este, el de evitar dichas conglomeraciones al almacén.

Sin embargo, también existen algunas órdenes denominadas urgentes que no pueden justificarse por ningún motivo. Consisten de aquellas requisiciones que surgen como consecuencia de:

- Un deficiente control de inventarios.
- Una mala planeación de la producción o presupuestos erróneos.
- Una aparente falta de confianza en la capacidad del departamento de compras de poder conseguir el material para la planta de una manera oportuna.
- La simple costumbre de anotar a las órdenes el término de "urgente".

### **Descripción precisa del artículo deseado**

No será posible esperar que el almacén adquiera artículos en forma apropiada si no conoce con precisión qué es lo que desean los departamentos usuarios. Es por esta razón que se considera algo básico el que se tenga una descripción precisa de la necesidad, el artículo, el bien o el servicio que se solicita.

El encargado del almacén podrá cuestionar una especificación si él pensara que la División pudiera salir beneficiada con alguna modificación a la solicitud. Un caso evidente es aquél en el cual existe escasez de un producto solicitado y se considera que un sustituto representa la única alternativa razonable. Puesto que las futuras condiciones del mercado constituyen un elemento vital, tendrá sentido el que exista un elevado grado de interacción entre el departamento de

## PLANTEAMIENTO DEL PROBLEMA

compras y los demás grupos de personas quienes hacen las especificaciones en las primeras etapas en las cuales se precisan las necesidades.

En el mejor de los casos, una descripción inexacta podrá provocar pérdida del tiempo; pero en ocasiones podrá tener serias consecuencias financieras provocando interrupciones en el abastecimiento, resentimientos entre personas y la pérdida de confianza y respeto en el proveedor.

Como ya se ha señalado al inicio de este punto en la parte del Flujo Físico donde se especifica la necesidad de la adquisición previa externa, es necesario que el departamento de almacén se enteré con exactitud de las especificaciones, antes de que se envíe la orden al proveedor, la necesidad de una verificación final se considera determinante.

Tal verificación no habrá de ser factible cuando el personal del departamento de almacén no está familiarizado con el producto o servicio solicitado. Cualquier duda respecto a la corrección de una requisición deberá turnarse de regreso a quien hizo las peticiones, no estando facultado el departamento de almacén para resolver tal duda de una manera unilateral.

Para ello deberá existir una gran uniformidad de términos para describir los artículos deseados. No puede pasar inadvertida la importancia que tiene el utilizar la nomenclatura apropiada como un medio para evitar todo tipo de confusiones. La manera más efectiva para lograr esta uniformidad consiste en mantener en el almacén un archivo o listado de los artículos comúnmente comprados.

Tal listado se lleva de la siguiente forma al tenerse un catálogo general que incluye todos los artículos utilizados que se mantienen en inventario. Esos catálogos se llevan en forma de hojas sueltas, índices de tarjetas y últimamente mediante un archivo de hoja de cálculo donde solo se presenta el material que supuestamente existe dentro del almacén, ya que no se tiene contemplado un conteo de los artículos para saber las existencias de productos solicitados.

### **Flujo de requisición para el almacén**

Como mínimo, es necesario que se formulen dos copias del vale: una de ellas la conservará el departamento que hace la petición y la otra se deberá enviar al departamento de almacén. En el caso de algunos artículos especiales, como podrían ser los materiales de plomería que se mantienen en inventario, se pueden incluir diferentes artículos (hasta diecisiete) en una sola petición, condicionado esto a la existencia del artículo en almacén y con entrega a una misma fecha. Esto simplificará el manejo de los registros, puesto que se adquirirán artículos específicos y no de uno en uno.

Se considera de suma importancia para el departamento de almacén sepa quién tiene la autoridad para presentar vales de almacén. Bajo ningún concepto aceptará aquél requisiciones de personas que no estén específicamente autorizadas para elaborar tales peticiones.

Será necesario que se revisen cuidadosamente todas las requisiciones antes de proseguir. La cantidad solicitada deberá basarse en necesidades previstas y deberá cotejarse con lo que se consideran lotes óptimos en el almacén. La fecha de entrega solicitada deberá conceder el suficiente tiempo para que se obtengan cotizaciones y muestras, si fueran necesarias, así como para colocar el pedido y obtener la entrega de los productos. Si no se concediera tiempo suficiente, o si para poder cumplir con la fecha estipulada se tuviera que incurrir en costos adicionales, esto deberá comunicarse de inmediato a la persona que presenta la requisición.

El procedimiento que aplica el departamento de almacén para manejar las requisiciones al recibirlas, se considera de tanta importancia que amerita citar un ejemplo:

Estando en el período de hacer validos los requerimientos de insumos por parte de cada jefe de departamento se procede a recabar los datos de los insumos que se van a necesitar, dicho período es de un día solamente de cada mes, posteriormente se elabora un vale de almacén el cual se encuentra disponible desde la página de la División de Ingenierías Civil y Geomática para su descarga y posterior llenado por la persona que mecanografía la orden (generalmente la secretaria del Jefe de Departamento).

## PLANTEAMIENTO DEL PROBLEMA

Elaborado entonces se procede a enviar a la persona encargada de la recopilación de vales para su próxima entrega al departamento de almacén, anotando en el formato la fecha en que se realiza la petición por parte del usuario con las firmas que autorizan dicho vale.

Dicho vale presenta la característica de poner el número solicitado por cada insumo requerido así como las observaciones pertinentes si es que ese fuera el caso de o de los productos solicitados hacia el almacén.

Posteriormente al entregarse al encargado de recibir dichos vales en almacén éste último pone una fecha tentativa para su entrega a la persona encargada de recibir y llevar los vales al almacén.

Ya que se tienen los productos, si es que este fuera el caso, se entregan a la Jefatura de División de Ingenierías Civil para su entrega personal al departamento que hizo el pedido, en caso contrario de faltar algún insumo requerido se hace la observación al solicitante el cual sabrá qué medidas tomar.

Gracias a la lista de materiales se tiene una forma más rápida de notificar al departamento de almacén respecto a las necesidades de los usuarios de esta forma se puede determinar la cantidad total de aquel o aquellos insumos faltantes que se necesitan para cubrir las cantidades que el stock necesita.

El empleo de una lista de materiales representa un medio para simplificar el proceso de requisiciones cuando están involucradas grandes cantidades de artículos frecuentemente utilizados.

### **1.3 ANÁLISIS DE LA PROBLEMÁTICA DENTRO DE LOS PROCESOS DE REQUERIMIENTOS.**

#### **Introducción**

Un almacén es un lugar o espacio físico para el almacenaje de bienes. Los almacenes son usados por fabricantes, importadores, exportadores, comerciales, transportistas, clientes, etc.



### **Tipos de productos almacenados**

En un almacén se depositan las materias primas, el producto semi-terminado o el producto terminado a la espera de ser transferido al siguiente eslabón de la cadena de suministro. Se puede también encontrar embalajes, piezas de recambio, piezas de mantenimiento según las decisiones de la empresa. Sirve como centro regulador del flujo de mercancías entre la disponibilidad y la necesidad de fabricantes, comerciantes y consumidores.

### **Infraestructuras.**

Normalmente son construcciones grandes y planas en las zonas industriales de las ciudades. Están equipados con:

- Muelles de carga para cargar y descargar camiones; o algunas veces son cargados directamente de vías de tren, aeropuertos o puertos marítimos,
- Material de almacenamiento, adaptado al producto y su rotación de inventario, como estantes, racks, etc.

### **Material de manipulación.**

A menudo disponen de carretillas elevadoras frontales, apiladoras o transpalets para la manipulación de mercancías que son generalmente depositadas en estantes estandarizados. Según el tipo de mercancía almacenado se puede también encontrar máquinas más específicas como puente grúa o grúas. Algunos almacenes están completamente automatizados, sin contar apenas con trabajadores en su interior. En estos casos, la manipulación de mercancía se realiza con máquinas coordinadas por controladores programables y ordenadores con el software apropiado.

Este tipo de almacenes automatizados, se pueden emplear para mercancías de temperatura controlada en los cuales la disponibilidad de espacio es menor debido al alto coste que la refrigeración supone para la empresa. También se utilizan para aquellas materias o mercancías que debido a su peligrosidad en el manipulado, o su elevada rotación del inventario rentabilizan el elevado coste que supone la puesta en marcha de este tipo de instalaciones.

### **Procesos y sistema.**

Los procesos comunes de un almacén son:

- Recepción.
- Colocación en stock.
- Preparación de pedido.
- Expedición.
- Gestión del stock (inventarios, ubicación...)

El sistema necesario para la gestión de un almacén es un SGA - Sistema de Gestión de Almacén.

Se denomina así a los programas informáticos destinados a gestionar la operativa de un almacén. Proviene de la traducción del término inglés "WMS" (Warehouse Management System).

### **Funcionamiento.**

Un SGA posee dos tipos básicos de mecanismos de optimización, uno dedicado a optimizar el espacio de almacenaje, mediante una adecuada gestión de ubicaciones y otro destinado a optimizar los movimientos o flujos de material, bien sean éstos realizados por máquinas o por personas.

Además, puede integrar mecanismos de cross docking, para tratar aquellos casos en los que el material pasa por el almacén tan sólo para el proceso de distribución, con lo que no se almacena, sino que simplemente se distribuye, trasladándose el material de los muelles de entrada del almacén a los de salida, asignando automáticamente el material recibido de los proveedores a los pedidos de los clientes.

### **Inventario Físico**

El inventario es una estadística física o conteo de los materiales existentes, para confrontarla con la existencia anotada en los ficheros de existencias o en el banco de datos sobre materiales.

El inventario físico se efectúa periódicamente, casi siempre en el cierre del período fiscal de la empresa, para efecto de balance contable. En esa ocasión, el inventario se hace en toda la empresa; en la bodega, las secciones, en el depósito, entre otras. El inventario físico es importante por las siguientes razones:

- Permite verificar las diferencias entre los registros de existencias y las existencias físicas (cantidad real en existencia).
- Permite verificar las diferencias entre las existencias físicas contables, en valores monetarios.
- Proporciona la aproximación del valor total de las existencias (contables), para efectos de balances, cuando el inventario se realiza próximo al cierre del ejercicio fiscal.

### **Codificación de materiales.**

Cuando la cantidad de artículos es muy grande, se hace casi imposible identificarlos por sus respectivos nombres, marcas, tamaños, etc.

Para facilitar la administración de los materiales se deben clasificar los artículos con base en un sistema racional, que permita procedimientos de almacenaje adecuados, de la bodega y control eficiente de las existencias. Se da el nombre de clasificación de artículos a:

**Catalogación:** Significa inventario de todos los artículos existentes sin omitir ninguna. La catalogación permite la presentación conjunta de todo los artículos proporcionando una idea general de la colección.

**Simplificación:** Significa la reducción de la gran diversidad de artículos empleados con una misma finalidad, cuando existen dos o más piezas para un mismo fin, se recomienda la simplificación que favorece la normalización.

**Especificación:** significa la descripción detallada de un artículo, como sus medidas, formato, tamaño, peso, etc. Cuando mayor es la especificación, se contará con más informaciones sobre los artículos y menos dudas con respecto de su composición y características. La especificación facilita las compras del artículo, pues permite dar al proveedor una idea precisa del material que se comprara.

**Normalización:** Indica la manera en que el material debe ser utilizado en sus diversas aplicaciones. La palabra deriva de normas, que son las recetas sobre el uso de los materiales.

**Estandarización:** significa establecer idénticos estándares de peso, medidas y formatos para los materiales de modo que no existan muchas variaciones entre ellos. La estandarización hace que, por ejemplo, los tornillos sean de tal o cual

## PLANTEAMIENTO DEL PROBLEMA

especificación, con lo cual se evita que cientos de tornillos diferentes entre innecesariamente en existencias.

### **Clasificación y Codificación de los Materiales.**

Así clasificar un material es agruparlo de acuerdo con su dimensión, forma, peso, tipo, características, utilización etc. La clasificación debe hacerse de tal modo que cada género de material ocupe un lugar específico, que facilite su identificación y localización de la bodega

La codificación es una consecuencia de la clasificación de los artículos. Codificar significa representar cada artículo por medio de un código que contiene las informaciones necesarias y suficientes, por medio de números y letras. Los sistemas de codificación más usadas son: código alfabético, numérico y alfanumérico.

En nuestro caso cabe mencionar que las funciones del almacén son:

- Generación de vales de compra.
- Generación de vales de almacén.
- Entrega de mercancía.
- Inventario.

A continuación se mencionan los problemas a los que se enfrenta dicho almacén con sus respectivas soluciones.

**a) El tiempo de demora de la solicitud de los vales es el principal problema que se presenta el cual se genera a causa de diferentes circunstancias, las cuales se desglosan a continuación:**

- Acudir a las oficinas a realizar algún trámite.

Para realizar cualquiera de los dos trámites ya sea el de compra o almacén se requiere trasladarse personalmente a las instalaciones de la División de Ingeniería Civil y Geomática, con previa elaboración del formato de solicitud

requisitado para el departamento de compras, y posteriormente acudir con el personal del almacén haciendo entrega del formato previamente autorizado para llevar a cabo la solicitud.

Dicho proceso puede durar de 30 a 40 minutos dependiendo de la carga de trabajo o la rapidez para obtener la información clave (existencia del insumo) ya que el personal que atiende necesita validar la solicitud, registrar y entregar un acuse donde se autorice la operación que se llevó a cabo.

- En ocasiones no se encuentra el personal ya sea el de compras o el del almacén.

Esto sucede muy a menudo, debido a que el personal salió por unos minutos o el personal designado al servicio se ausenta por alguna emergencia personal y no se encuentra quien pueda desempeñar dicha función por el momento lo cual hace que el solicitante espere al personal para realizar el trámite o en dado caso realizarlo un día posterior.

Con la ayuda del sistema el proceso será más rápido y sencillo, sobre todo porque no se requiere el traslado del usuario hasta las oficinas, todo se hará por medio de una interfaz gráfica utilizando cualquier explorador, a través de la red local o desde cualquier punto de la red global.

El sistema simplificaría la espera del usuario manteniendo una base en donde su pedido tendrá datos pre-cargados y en el cual la foliación se le asignará en el momento del término de llenado de la solicitud ya sea de compra o almacén.

Otra ventaja es la facilidad de operación, lo cual ayudará a que el personal encargado de validar las operaciones solicitadas pueda revisar en tiempo real las solicitudes pendientes y de esa manera no se retrase el tiempo de respuesta al usuario.

Logrando así la generación de los vales en tiempo y forma sin necesidad de acudir a las oficinas de la División de Ingeniería Civil y Geomática, los vales se

## PLANTEAMIENTO DEL PROBLEMA

generarán dependiendo de la hora de solicitud; puesto que el sistema seguirá funcionando después de horas laborales y en estos casos la respuesta o validación se efectuará al siguiente día laboral dentro de las primeras horas.

### **b) La administración del inventario es otro problema al cual se le requiere dar solución.**

La problemática comienza cuando no se tienen actualizados los datos de la existencia de los insumos. Lo cual deriva en la ineficacia para atender la solicitud del usuario, ya sean los siguientes casos:

- Insuficiencia del existente en el almacén: Cuando el usuario solicita el producto y no se encuentra el inventario debidamente actualizado se tiene faltante de algún insumo requerido y no se puede atender satisfactoriamente dicha solicitud.
- Exceso de algún insumo existente en el almacén: Si al momento de realizarse la solicitud de compra no se encuentra con el inventario debidamente actualizado, se puede incurrir en el exceso de algún insumo lo cual afecta en el ámbito económico ya que se compra de más y se descuida lo que se necesita en el almacén.

En los dos casos anteriores el sistema reflejaría en tiempo real la existencia exacta del material que se encuentra en el almacén, con esto los vales de compra se expedirán solamente para las necesidades reales.

Cabe mencionar que el sistema permitirá consultar al solicitante la existencia de cada insumo y así él pueda detectar sus necesidades más urgentes y mantener al día el funcionamiento del almacén

Otras consecuencias que originan un mal servicio y afectan la actualización del inventario.

- Baja calidad y caducidad de los insumos: Resulta frecuente que en la solicitud no se especifican las características del producto según las

necesidades del usuario, a lo que al momento de comprar no se obtenga lo deseado. También suele ocurrir que se compra algún material pero este no se utiliza y llega la fecha de caducidad, provocando una pérdida tanto material como económica ya que se tendría que volver a realizar la compra de dicho insumo.

En este caso el sistema clasificaría los insumos ya sea por el tipo, forma, tamaño, cantidad, caducidad etc. Según se requiera especificar con cada producto y así obtener lo requerido por el usuario

- Desorden de los insumos del almacén. Es recurrente que la ubicación de algunos productos no sea la misma o que el encargado del almacén desconozca la ubicación y con eso se retrase la entrega o la realización del inventario por lo que no es eficiente el desarrollo diario del almacén.

En el sistema se encontrará rápidamente la información con solo buscar el tipo de producto o dato clave por el cual se encontrará la ubicación del insumo en el almacén, así como la descripción y clasificación del producto.

El sistema permitirá acceder a toda la información del producto y así clasificar, contabilizar, ubicar, etc. En otras palabras llevar el control de todo el funcionamiento del almacén desde el ordenador

- Robo: El robo de los insumos no permite tener la existencia real del almacén. El robo consiste en un problema tanto por la pérdida del producto así como la inexactitud de la información en el inventario dando lugar a los casos que se enlistaron anteriormente.

La captura de la información en el sistema permitirá hacer revisiones periódicas aun cuando no exista solicitud, y para el caso que señalamos (robo); permite detectar los faltantes en el almacén, para poder realizar el mantenimiento adecuado del almacén así como los reportes que requiera el jefe de departamento

## PLANTEAMIENTO DEL PROBLEMA

Derivado del hecho de no tener el inventario actualizado surge el problema de:

- Dificultad en la toma de decisiones para realizar la compra del material faltante y para realizar los reportes que se soliciten.

De esa manera con el sistema se le puede dar información certera al usuario de la existencia del material con el que se dispone en el almacén y así del mismo modo para la propia administración interna de la División de Ingeniería Civil y Geomática para realizar los reportes cuando el jefe de departamento los solicite.

PROBLEMÁTICA	AFECTACION
Lento	El tiempo de elaboración y aprobación del vale puede tardar hasta 50 min.
Falta de disponibilidad	Depende de la presencia del personal.
No es seguro	No se tiene de un método de seguridad como autenticación de usuario.
Procesos son excesivos	Muchos procedimientos y personas involucradas en el proceso de obtención de vales.

## 1.4 REQUERIMIENTOS DEL SISTEMA DE VALES, SOFTWARE Y HARDWARE

### 1.4.1 Sistema Operativo

Un sistema operativo es un conjunto de programas cuya función es la de administrar recursos, de manera principal el hardware de una computadora. Las principales tareas que lleva a cabo un sistema operativo son las siguientes:



- Gestión de recursos, administra hardware y redes informáticas así como dispositivos periféricos del sistema.
- Interfaz de usuario, es la parte que permite la comunicación con del usuario con el sistema operativo.
- Gestión de archivos, controla la creación, borrado, acceso a archivos de datos y de programas. Mantiene el registro de la ubicación física de estos.
- Gestión de procesos, controla que programas tienen acceso a que recursos de la computadora así como por cuanto tiempo. Otorga prioridades de ejecución dependiendo del tipo de programa y privilegios asignados.
- Servicios de soporte, incluyen la actualización de versiones, seguridad, controladores de hardware, corrección de errores.

De forma gráfica podemos ver el nivel que ocupa un sistema operativo en una computadora. Se observa su función principal que es la de ser una interfaz entre el hardware y una aplicación. Vea figura 1.4.1



Fig. 1.4.1 Interacción del sistema operativo con los componentes de una computadora

Se evaluaron diferentes opciones y se llegó a estas versiones.

Los sistemas operativos pueden ser clasificados de la siguiente forma:

## PLANTEAMIENTO DEL PROBLEMA

- **Multiusuario:** Permite que dos o más usuarios utilicen sus programas al mismo tiempo. Algunos sistemas operativos permiten a centenares o millares de usuarios al mismo tiempo.
- **Multiprocesador:** soporta el abrir un mismo programa en más de una CPU.
- **Multitarea:** Permite que varios programas se ejecuten al mismo tiempo.
- **Multitramo:** Permite que diversas partes de un solo programa funcionen al mismo tiempo.
- **Tiempo Real:** Responde a las entradas inmediatamente. Los sistemas operativos como DOS y UNIX, no funcionan en tiempo real.

En el sistema de vales, la elección del sistema operativo resulta fundamental para poder lograr un funcionamiento óptimo. Se ocupará Debian GNU/Linux en su versión 4.0 "Etch".

### **Debian GNU/Linux 4.0**

GNU/Linux (Linux) es uno de los términos empleados para referirse al sistema operativo libre similar a UNIX que usualmente utiliza herramientas de sistema GNU. A las variantes de esta unión de programas y tecnologías, se las denomina distribuciones. Su objetivo consiste en ofrecer ediciones que cumplan con las necesidades de determinado grupo de usuarios. Algunas de ellas son especialmente conocidas por su uso en servidores y supercomputadoras. No obstante, es posible instalarlo en una amplia variedad de hardware como computadoras de escritorio y portátiles.

Debian es una distribución de GNU/Linux, desarrollada por más de mil voluntarios alrededor del mundo, que colaboran a través de Internet. Estos mantienen un sistema operativo GNU basado en software libre precompilado y empaquetado en un formato sencillo en múltiples arquitecturas y en múltiples núcleos. Cuenta con una amplia colección de software disponible, así como un grupo de herramientas que facilitan el proceso de instalación y actualización de software. Ofrece diversas ventajas como son:

- Disponibilidad en diversas arquitecturas (como lo son; i386, AMD64 (x86,x64), alpha, sparc, arm, armel, powerpc, hppa, ia64, mips, s390, m86k).
- Seguridad (cuenta con herramientas de evaluación de vulnerabilidad remota, herramientas de revisión de redes, infraestructura de claves públicas, antivirus, firewall).
- Estabilidad, dada la naturaleza del sistema operativo al ser GNU/Linux un clon de UNIX, se cuenta con la ventaja de ser un sistema multiusuario y multitarea.
- Los paquetes son de código abierto por lo que pueden modificarse de acuerdo a las necesidades particulares del usuario.
- El manejo de la memoria evita que los errores de aplicación detengan el funcionamiento del núcleo del sistema.
- Soporta diversos entornos gráficos (GNOME, KDE, XFCE, etc.)
- Uso de formatos estándar, estos permiten la interoperabilidad más alta entre sistemas evitando incompatibilidades
- Rápida solución de fallos de seguridad, debido a que se cuenta con una comunidad que trabaja de manera conjunta ante este tipo de errores.

Como requerimientos del sistema operativo se debe tener al menos 44 MB de memoria, así como 500 MB de espacio de disco duro

#### **1.4.2 Servidor Web**

Un servidor Web es un programa que implementa el protocolo HTTP (HyperText Transfer Protocol). Éste está diseñado para transferir hipertextos, páginas Web o páginas HTML (HyperText Markup Language). Es un programa que se ejecuta continuamente en un ordenador (también se emplea el término para referirse al equipo que lo ejecuta), manteniéndose a la espera de peticiones por parte de un cliente (un navegador Web) y que responde a estas peticiones adecuadamente, mediante una página Web que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error. Se pueden utilizar varias tecnologías en el servidor para aumentar su potencia

## PLANTEAMIENTO DEL PROBLEMA

más allá de su capacidad de entregar páginas HTML; éstas incluyen scripts CGI, seguridad SSL y aplicaciones del lado del servidor.

En el sistema de vales se ocupará el servidor Web Apache 2.0.49.

### **Servidor HTTP Apache 2.0.49**

Es un servidor Web de código abierto multiplataforma que implementa el protocolo HTTP/ 1.1 , entre sus ventajas podemos destacar las siguientes;

- Modular, consta de un núcleo y un conjunto de módulos que aportan funcionalidad adicional al servidor Web como por ejemplo comunicaciones vías seguras, autenticación contra LDAP, soporte para páginas dinámicas en Perl, PHP, Python, etc. Esto permite el poder personalizar el servidor dependiendo los requerimientos del usuario.
- Código abierto, esto lo hace extensible ya que siendo que su API es de dominio público, pueden realizarse mejoras e innovaciones sobre su desarrollo por lo que es posible crear funcionalidades nuevas y particulares a los requerimientos del usuario.
- Portable (Unix, BSD, GNU/Linux, Windows, Macintosh y otras).
- Eficiente, muestra ventajas en pruebas de rendimiento respecto a otros servidores Web. Consume menos recursos que otros servidores Web, dado que su código se encuentra en constante optimización para su desempeño,
- Puede configurarse de manera sencilla ya que cuenta con archivos de configuración específicos.
- Soporte, cuenta con soporte del Apache Group el cual es un grupo de usuarios dedicados y compañías que distribuyen de manera comercial el software.
- Es de fácil administración ya que se cuenta con interfaces gráficas así como un conjunto de herramientas desarrolladas con este propósito.

### **1.4.3 Navegador Web**

Un navegador Web es un programa que permite la visualización de la información de una página Web, éste interpreta código HTML generalmente. La función principal de un navegador es visualizar documentos de texto, posiblemente con recursos multimedia. Los navegadores más modernos soportan los estándares HTML y XHTML publicados por el World Wide Web Consortium.

#### **Mozilla Firefox 3.5.6**

Es un navegador Web libre desarrollado por la corporación Mozilla, la fundación Mozilla y un gran número de voluntarios externos. Es un navegador multiplataforma que se encuentra disponible para GNU/Linux, Microsoft Windows, MAC OS X. Su código fuente es software libre.

Las características que incluye Mozilla Firefox son la navegación por pestañas corrector ortográfico, marcadores, bloqueador de ventanas emergentes, atajos del teclado, soporte para motores de búsqueda y un gestor de descargas. Firefox proporciona un entorno para los desarrolladores Web, en el que se puede utilizar herramientas incorporadas, como la Consola de errores o el Inspector DOM o extensiones, como Firebug. Es compatible con varios estándares Web, incluyendo HTML, XML, XHTML, SVG, CSS y demás.

Firefox utiliza el sistema SSL/TLS para proteger la comunicación con los servidores Web, utilizando fuerte criptografía cuando se utiliza el protocolo HTTPS. También proporciona apoyo a las tarjetas inteligentes para fines de autenticación. Cuenta con una protección antiphishing que alerta al usuario cuando está accediendo a una página que contiene código malicioso, troyanos, virus y programas espías, a su vez cuenta con antimalware e integración con el antivirus.

Firefox cuenta con un excelente rendimiento y eficiencia en todos los sistemas operativos, a la vez reduce el consumo de memoria de la aplicación y mejora la

velocidad de ejecución de las páginas dinámicas, por estas razones Firefox será usado en el sistema de vales.

### **1.4.4 Desarrollo WEB**

Podemos definir el desarrollo Web como el conjunto de tecnologías del lado del servidor y del cliente que involucran una combinación de procesos de bases de datos con el uso de un navegador en Internet a fin de realizar determinadas tareas o mostrar información. Los lenguajes de programación más usados para el desarrollo Web son ASP.NET, PHP y JSP,

#### **PHP 4.0**

PHP significa Preprocesador de Hipertexto y es un lenguaje de programación interpretado, diseñado con el propósito de creación de páginas Web dinámicas. Una página Web dinámica es aquella que permite la interacción del usuario con un sitio Web.

Es aquí donde intervienen las CGI's (Interfaces Comunes de Puerta de Enlace), estos son programas ejecutables en el servidor cuyo código se ejecuta en el servidor en vez de la computadora del cliente. Las CGI's permiten la interacción en línea con bases de datos y permiten la transferencia bidireccional de información entre el cliente y el servidor. Los CGI's pueden escribirse en diversos lenguajes.

PHP puede ser desplegado en la mayoría de los servidores Web y en casi todos los sistemas operativos y plataformas, su sintaxis es similar a lenguajes de programación estructurada como Perl y C. Permite la conexión a diferentes tipos de servidores de bases de datos como son MySQL, Postgress, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite.

PHP es una alternativa a las tecnologías de Microsoft ASP y ASP.NET, a Cold Fusion, a JSP/Java, y a CGI/Perl, es software libre por lo que no se está forzado a pagar algún tipo de actualización, puede ser adaptado a las

necesidades particulares del usuario, es expansible ya que cuenta con una gran cantidad de módulos disponibles y una variedad de extensiones de código.

#### **1.4.5 Sistema de Gestión de Base de Datos**

Los sistemas de gestión de bases de datos o SGBD (en inglés database management system, abreviado DBMS) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Estos proveen facilidades para la manipulación de grandes volúmenes de datos, consistencia y seguridad.

- **Abstracción**, provee la abstracción física de los datos, haciendo transparentes al usuario detalles acerca del almacenamiento físico de los datos. La independencia de los datos que consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- **Consistencia**. En aquellos casos en los que no se ha logrado eliminar la redundancia, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.
- **Seguridad**. Los SGBD deben garantizar que la información se encuentra segura de permisos a usuarios y grupos de usuarios.

En la manipulación de una base de datos, los SGBD deben incluir un control de concurrencia, o sea, deben permitir a varios usuarios tener acceso "simultáneo" a la base de datos. Controlar la concurrencia implica que si varios usuarios acceden a la base de datos, la actualización de los datos se haga de forma controlada para que no haya problemas.

El SGBD que se ocupará es el relacional, este representa a la base de datos como una colección de tablas. Estas bases de datos suelen utilizar SQL como lenguaje de consultas de alto nivel.

## PLANTEAMIENTO DEL PROBLEMA

El SGBD que se ocupa para el sistema de vales es MySQL el cual es un SGDB relacional, multihilo y multiusuario, es libre por lo que se tienen grandes ventajas en el aspecto de personalización para el desempeño e integración con herramientas de desarrollo libre como PHP.

### **MySQL**

MySQL es un sistema de administración de bases de datos relacionales Open Source y licencia pública, de nivel empresarial y múltiples subprocesos. Está desarrollado en ANSI C. MySQL puede controlar bases de datos, quién puede utilizarlas y cómo son manipuladas. Registra las acciones y las ejecuciones de manera continua en segundo plano. Esto lo hace diferente a la mayoría de los SGDB. Existe una variedad de usos para un SGDB, estos pueden ir desde sistemas de ayuda hasta aplicaciones para sitios Web.

Existen varias APIs que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, incluyendo C, C++, C#, Pascal, Delphi, Eiffel, Smalltalk, Java, Lisp, Perl, PHP, Python, Ruby, Gambas, REALbasic (Mac), (x)Harbour (Eagle1), FreeBASIC, y Tcl; cada uno de estos utiliza una API específica. También existe un interfaz ODBC, llamado MyODBC que permite a cualquier lenguaje de programación que soporte ODBC comunicarse con las bases de datos MySQL.

MySQL es muy utilizado en aplicaciones Web, como Drupal o phpBB, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación Web está muy ligada a PHP, que a menudo aparece en combinación con MySQL.

En aplicaciones Web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones.

MySQL funciona sobre múltiples plataformas, incluyendo: AIX, BSD, FreeBSD, HP-UX, GNU/Linux, Mac OS X, NetBSD, Novell Netware, OpenBSD, OS/2



Warp, QNX ,SGI IRIX, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64, eBD, Windows y OpenVMS.

Las siguientes características son implementadas únicamente por MySQL:

- Múltiples motores de almacenamiento (MyISAM, Merge, InnoDB, BDB, Memory/heap, MySQL Cluster, Federated, Archive, CSV, Blackhole y Example en 5.x), permitiendo al usuario escoger la que sea más adecuada para cada tabla de la base de datos.
- Agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo.

Se eligió MySQL dado que su diseño está orientado a la velocidad, así como a soportar un gran número de transacciones sin afectar esto al desempeño de la aplicación.

En resumen, los requerimientos mínimos para la implementación con relación al software son:

**Requerimientos en Servidor:**

- Debian GNU/Linux
- Apache
- MySQL
- PHP

**Requerimientos en Cliente:**

- Mozilla Firefox.

**Con relación al hardware mínimo en servidor es:**

- Intel Pentium Core 2 Duo 2.2 GHz
- GB de memoria RAM
- 160 GB de disco duro
- DVD
- Monitor

## PLANTEAMIENTO DEL PROBLEMA

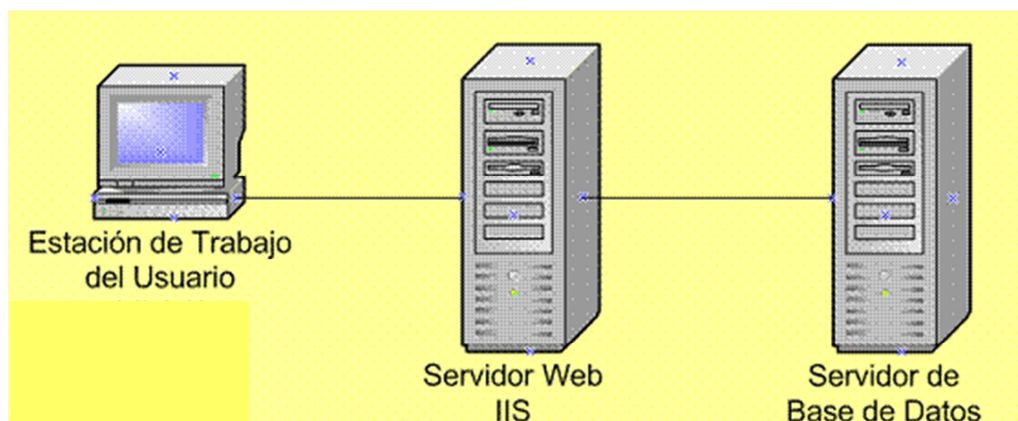
- Teclado
- Mouse

TABLA DE SOFTWARE
Debian GNU/Linux
Apache
MySQL
PHP
Mozilla Firefox

El sistema es una solución a los problemas que tiene la División de Ingeniería Civil y Geomántica en el proceso de generación de vales y de obtención de materiales del almacén.

- Ya que el sistema genera el vale mediante una selección de un catálogo, y la posterior impresión del mismo.
- Como método de seguridad no necesitaría una firma de validación ya que el sistema cuenta con un método de autenticación mediante un usuario y contraseña, evitando con estar sujeto a la disponibilidad del personal que autoriza.

Logrando así una reducción considerable en los tiempos que lleva realizar este proceso así como también una reducción del personal involucrado.



## **2.1 BASE DE DATOS RELACIONALES**

### **Introducción**

Una base de datos (DB por sus siglas en inglés) puede definirse como la serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los Sistemas de Información de una empresa o negocio en particular. Las bases de datos proporcionan la infraestructura requerida para los sistemas de apoyo a la toma de decisiones y para los sistemas de información.

La base de datos nos permite representar parte del mundo real y dependiendo de cómo se vincule, da lugar a una base de datos jerárquica de red o relacional. El modelo relacional se impone sobre el modelo jerárquico y de red debido a su sencillez, pues emplea como modelo de datos las tablas y por el carácter declarativo de su lenguaje de consulta y manipulación.

El modelo de datos relacional está basado en el concepto de relación pues es un modelo en el cual no existe una jerarquía entre los campos de datos de un registro, por lo que cada campo puede ser usado como llave. Una relación es un conjunto de  $n$  tuplas o duplas.

Cada tupla representa un conjunto de valores relacionados permanentemente y pueden representar tanto entidades como relaciones. El número de duplas en una relación es la cardinalidad de la relación.

Las relaciones también pueden ser vistas como tablas, en las que cada tupla corresponde a una fila de la tabla. Los nombres de las columnas de las tablas son los atributos.

Principales características de las bases de datos relacionales:

- Están basadas en el modelo de relaciones, es decir, una tabla bidimensional constituida por líneas (tuplas) y columnas (atributos).

## MARCO TEÓRICO

- Cada tabla tiene un nombre que la identifica unívocamente.
- Las entradas en las tablas tienen un solo valor (son atómicos); no se admiten valores múltiples, por lo tanto la intersección de un renglón con una columna tiene un solo valor y nunca un conjunto de valores.
- Todas las entradas de cualquier columna son de un solo tipo.
- Cada columna posee un nombre único.
- El orden de las columnas no es de importancia para la tabla.
- Cada atributo tiene un dominio, que es una descripción física y lógica de valores permitidos.
- No existen 2 filas en la tabla que sean idénticas.
- La información en las bases de datos son representados como datos explícitos, no existen apuntadores o ligas entre las tablas.
- Las tablas poseen una llave primaria o foránea.
- No redundancia o redundancia controlada.
- Existe consistencia en los datos.
- Contienen reglas de integridad de datos.
- Tener algún tipo de relación entre tablas.

Los cuatro componentes principales de una base de datos son: lo datos a manejar, el hardware, el software y el personal encargado del manejo del sistema. A continuación se describen de manera breve cada uno de ellos:

- **Datos.**- Los datos de una base de datos, deben recoger toda la información necesaria, sin que exista repetición de la misma dentro de la base.
- **Hardware.**-Es el soporte físico que permite almacenar la información de la base de datos.
- **Software.**- el software permite interactuar con la base de datos de manera eficiente.
- **Personal.**- Las bases de datos son manipuladas por personal con diferentes perfiles, entre ellos podemos mencionar:

**Administradores del DBMS** (DBA por sus siglas en inglés). El DBA es el encargado de la administración de la base de datos, es decir, el manejo y control de la misma. Son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

**Administradores de la base** (DBO por sus siglas en ingles) El DBO tiene como principales tareas: diseñar la base de datos, definir las estructuras de almacenamiento y métodos de acceso (tablas) modificar el diseño y organización física de la base y una de las más importante es el de otorgar permisos de acceso a usuarios dentro de la base.

**Programadores de aplicaciones o desarrolladores.** Los programadores de aplicaciones se encargan de diseñar y programar las aplicaciones necesarias para la utilización de la base de datos.

**Usuarios.** Los usuarios son aquellos que utilizan el sistema de base de datos por medio de una interfaz gráfica e interactúan con los mismos, dependiendo de los permisos que les hayan sido otorgados.

### **Sistema manejador de base de datos (DBMS)**

Desde un punto de vista técnico, un sistema manejador de base de datos es un conjunto coordinado de programas, procedimientos, lenguajes, etc., que suministran, a los distintos usuarios de la base, los medios necesarios para describir y manipular los datos almacenados en la base.

El DBMS se compone de un lenguaje de definición de datos, un lenguaje de manipulación de datos y de un lenguaje de control de datos, los cuales se explican a continuación:

### **Lenguaje de definición de datos** (DDL por sus siglas en ingles)

Permite establecer un modelo de base de datos por medio de una serie de definiciones que se expresan en un lenguaje especial, el resultado se almacena en un archivo llamado diccionario de datos.

### **Lenguaje manipulador de datos** (DML por sus siglas en ingles)

Se refiere a una serie de expresiones que permiten manipular los datos. Entre las operaciones más comunes podemos mencionar: insertar, recuperar, eliminar o modificar los datos. Existen básicamente dos tipos de lenguajes de manipulación de datos:

**Lenguajes con procedimientos o procedurales**, manipulan la base de datos registro a registro, en ellos se especifica qué operaciones se deben realizar para obtener los datos resultado.

**Lenguajes sin procedimiento o no procedurales**: son aquellos que operan sobre conjuntos de registros, en los cuales se especifica qué datos deben obtenerse sin decir cómo hacerlo. El lenguaje no procedural más utilizado es el SQL (Structured Query Language) que de hecho, es un estándar y es el lenguaje de los DBMS relacionales.

***Lenguaje de control de datos (DCL por sus siglas en inglés)***:

Contiene elementos útiles para trabajar en un entorno multiusuario, en el que es importante la protección de los datos, la seguridad de las tablas y el establecimiento de restricciones en el acceso, así como de elementos para coordinar el proceso de compartir los datos por parte de usuarios concurrentes, asegurando que no interfieren unos con otros.

### ***Funciones de un DBMS***

La función de un DBMS no se limita solo a permitir mediante la definición y manipulación de datos el dialogo entre usuarios y la base de datos, si no también brinda mecanismos para controlar la concurrencia de usuarios, la seguridad e integridad de la base de datos (incluso después de que haya ocurrido una falla en el sistema, ya sea está provocada por software o hardware).

Las funciones principales de una DBMS se listan a continuación:

- Crear y organizar la base de datos
- Establecer y mantener las trayectorias de uso de la base de datos de tal forma que los datos puedan ser accedidos rápidamente.
- Manejar los datos de acuerdo a las peticiones de los usuarios
- Registrar el uso de las bases de datos.

### Arquitectura de un DBMS

Existen diferentes niveles de abstracción para simplificar la interacción de los usuarios con el sistema, éstos son: el nivel interno, el conceptual y el externo los cuales se muestran en la figura 2.1.1.

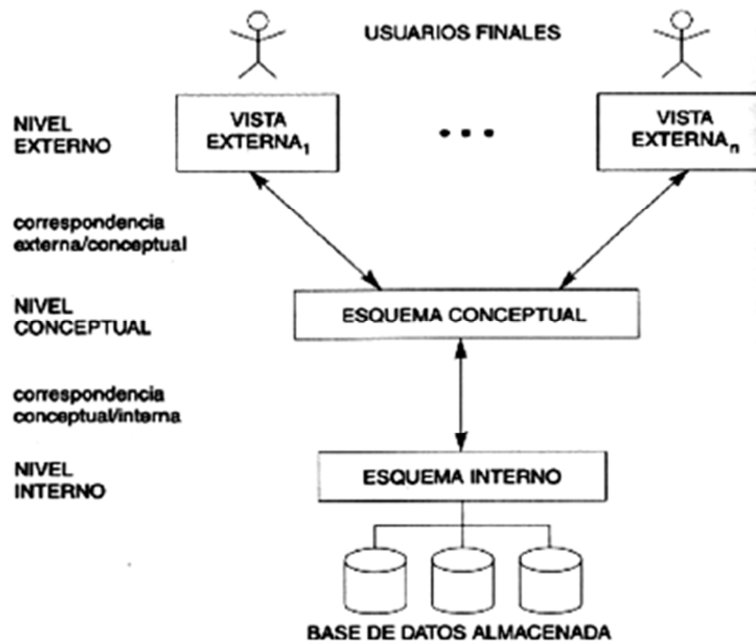


Figura 2.1.1 Niveles de abstracción de un DBMS.

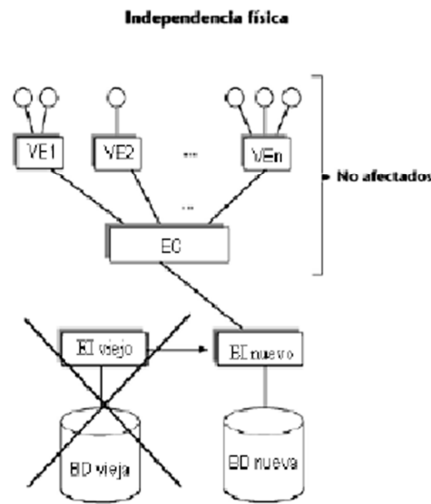
En la figura 2.1.1 podemos observar la relación existente entre los niveles de abstracción de un DBMS a continuación explicaremos la función que tienen cada uno de ellos.

**Nivel interno.**- es la representación del nivel más bajo de abstracción, en éste se describe a detalle la forma de cómo se almacenan los datos en los dispositivos de almacenamiento. Responde a las cuestiones de rendimiento (espacio y tiempo) planteadas al hacer el diseño físico de la base de datos y al ajustarlo posteriormente a nuevas necesidades.

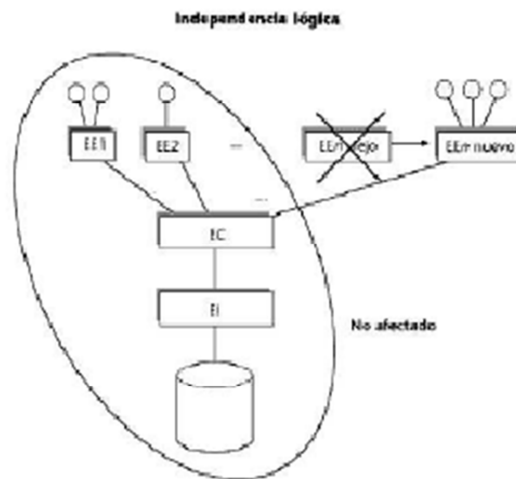
**Nivel conceptual.**- Es el siguiente nivel de abstracción, describe que datos son almacenados realmente en la base de datos y las relaciones que existen entre los mismos, describe la base de datos completa en términos de su estructura de diseño.

**Nivel externo.**- es el nivel más alto de abstracción, es lo que el usuario final puede visualizar del sistema terminado, muestra solo una parte de la base de datos al usuario acreditado para verla.

Los niveles de abstracción anteriormente descritos proporcionan también la independencia de los datos, dicha independencia puede ser de dos tipos: la física (figura 2.1.2) y la lógica (figura 2.1.3).



**Figura 2.1.2** Hay independencia física cuando los cambios en la organización física de la B.D. no afectan al mundo exterior (es decir, los programas de usuarios o los usuarios directos).



**Figura 2.1.3** Hay independencia lógica cuando los usuarios (programas de aplicación o usuarios directos) no se ven afectados por los cambios en el nivel lógico.

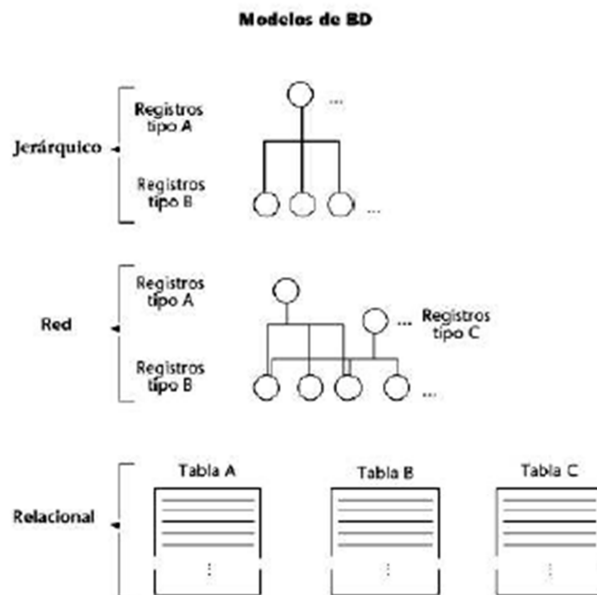
### **Modelo de datos**

El modelo de datos es una colección de herramientas conceptuales para describir los datos, las relaciones que existen entre ellos, la semántica



asociada a los datos y restricciones de consistencia. En otras palabras es el conjunto de componentes o herramientas conceptuales que un SGBD proporciona para modelar.

Existen fundamentalmente tres alternativas disponibles para diseñar las bases de datos: el modelo jerárquico, el modelo de red y el modelo relacional, vea figura 2.1.4.



**Figura 2.1.4 Modelos de las B.D.**

**Modelo jerárquico:** este modelo se conforma por registros interrelacionados en forma de árboles, los cuales pueden representar dos tipos de relaciones entre los datos: relaciones de uno a uno y de uno a muchos.

**Modelo de red:** este modelo permite la representación de muchos a muchos, de tal forma que cualquier registro dentro de la base de datos puede tener varias ocurrencias superiores a él.

### **Modelo Entidad-Relación**

Los diagramas o modelos entidad-relación (denominado por su siglas, ERD “Entity Relationship Diagram”) son una herramienta para el modelado de datos de un sistema de información. Estos modelos expresan entidades relevantes para un sistema de información, sus interrelaciones y propiedades.

## Cardinalidad De Las Relaciones

El diseño de relaciones entre las tablas de una base de datos puede ser la siguiente:

- **Relaciones de uno a uno:** una instancia de la entidad A se relaciona con una y solamente una de la entidad B.
- **Relaciones de uno a muchos:** cada instancia de la entidad A se relaciona con varias instancias de la entidad B.
- **Relaciones de muchos a muchos:** cualquier instancia de la entidad A se relaciona con cualquier instancia de la entidad B

## Álgebra Relacional

Es un conjunto de operaciones que describen paso a paso cómo computar una respuesta sobre las relaciones, tal y como éstas son definidas en el modelo relacional. Describe el aspecto de la manipulación de datos. Estas operaciones se usan como una representación intermedia de una consulta a una base de datos y, debido a sus propiedades algebraicas, sirven para obtener una versión optimizada y eficiente de dicha consulta.

**Tuplas:** Son las "filas de una tabla" en el lenguaje usual de bases de datos.

**Unión compatible:** Una unión es compatible entre dos relaciones, si ellas poseen el mismo grado y su dominio son los mismos de izquierda a derecha.

**Grado:** Número de atributos.

## Operaciones

**Selección ( $\sigma$ ):** Una condición puede ser una combinación booleana, donde se pueden usar operadores como:  $\wedge$ ,  $\vee$ , combinándolos con operadores  $<$ ,  $>$ ,  $\leq$ ,  $\geq$ ,  $=$  y  $\neq$ .

**Proyección ( $\Pi$ ):** Permite extraer columnas (atributos) de una relación, dando como resultado un subconjunto vertical de atributos de la relación, esto es:

$\Pi_{A_1, A_2, \dots, A_n}(R)$ , donde  $A_1, A_2, \dots, A_n$  son atributos de la relación R.

**Producto cartesiano ( $\times$ ):** El producto cartesiano de dos relaciones se escribe como  $R \times S$ . Entrega una relación cuyo esquema corresponde a una

combinación de todas las tuplas de R con cada una de las tuplas de S, y sus atributos corresponden a los de R seguidos por los de S.

**Unión ( $\cup$ ):** Se denota como  $R \cup S$ . Regresa el conjunto de tuplas que se encuentran en R, o en S, o en ambas R y S; deben ser uniones compatibles.

**Diferencia ( $-$ ):** La diferencia de dos relaciones, R y S denotada por:  $R - S$ . Entrega todas aquellas tuplas que están en R, pero no en S. R y S deben ser uniones compatibles.

Estas operaciones son fundamentales en el sentido en que:

- Todas las demás operaciones pueden ser expresadas como una combinación de éstas.
- Ninguna de estas operaciones pueden ser omitidas sin que con ello se pierda información.

### **Cálculo Relacional**

El Cálculo relacional es un lenguaje de consulta que describe la respuesta deseada sobre una Base de datos sin especificar como obtenerla, a diferencia del Álgebra relacional que es de tipo procedural, el cálculo relacional es de tipo declarativo; pero siempre ambos métodos logran los mismos resultados.

### **Relación Con Lógica De Primer Orden**

Podemos definir una formula con base a combinaciones de fórmulas atómicas:

- Una formula atómica es una combinación de variables (tipo tupla o tipo dominio, según corresponda) y atributos o constantes, gracias al uso de operadores como  $<, >, \leq, \geq, =$  y  $\neq$ .
- También es una formula atómica *variable  $\in$  Relación*.
- Las combinaciones de fórmulas atómicas se generan a partir del uso de operadores como NOT ( $\neg$ ), AND ( $\wedge$ ), OR ( $\vee$ ),  $\rightarrow$ .
- Los cuantificadores  $\exists, \forall$  limitan una variable.

### **Cálculo Relacional Basado En Tuplas (TRC).**

Una consulta en TRC es de la forma  $\{T \mid \varphi(T)\}$  ; donde T es una variable tipo tupla y  $\varphi(T)$  es una fórmula que describe a T. El resultado de esta consulta, son todas las tuplas t para las cuales la fórmula es verdadera.

**Variable de tipo tupla** : Una variable tipo tupla T es una variable capaz de tomar cualquier valor tupla que pertenece a una relación (o tabla)

**Sintaxis de consulta en TRC:** La sintaxis es definida a partir de la lógica de primer orden. Donde la variable a utilizar es de tipo tupla. Una variable es libre en una fórmula (o subfórmula) si la (sub) fórmula no contiene ninguna ocurrencia de cuantificadores que la limiten. En una consulta en TRC de la forma:  $\{T \mid \varphi(T)\}$ , T es la única variable libre.

### **Cálculo Relacional Basado En Dominios (DRC)**

Está constituido con los mismos operadores que el cálculo relacional de tuplas pero no hay tuplas sino variables dominio. Las expresiones del cálculo relacional de dominios son de la forma  $\{ (x,y,z,\dots) \mid P(x, y, z,\dots) \}$ . Donde x, y, z representan las variables de dominio, P representa una fórmula compuesta de átomos . Los átomos del cálculo relacional de dominios tienen una de las siguientes formas:

- $(x, y, z) \in r$ , donde r es una relación con n atributos y x , y, z .son variables de dominio o constantes.
- $x \theta y$ , donde x e y son variables de dominio y  $\theta$  es un operador de comparación aritmética ( $>$ ,  $<$ ,  $=$ ,  $\neq$ ). Es necesario que los atributos x e y, tengan dominios cuyos miembros puedan compararse mediante  $\theta$ .
- $x \theta c$ , donde x es una variable de dominio,  $\theta$  es un operador de comparación y c es una constante en el dominio del atributo x.

## **Normalización**

El modelo conceptual de datos obtenido mediante la técnica de entidad-relación se refina y convierte en un modelo lógico relacional, utilizando la normalización, lo que ofrece como resultado el conjunto de tablas a implantar en la base de datos. Su finalidad es reducir las inconsistencias y redundancias de los datos, facilitar el mantenimiento y evitar las anomalías en las manipulaciones de datos, su objetivo es obtener un modelo lógico normalizado que represente las entidades normalizadas y las interrelaciones existentes entre ellas.

Existen varias reglas de normalización sin embargo, es suficiente con garantizar que se cumplan las tres primeras formas normales:

**Primera Forma Normal (FN1):** Una relación R satisface la primera forma normal (FN1) si cumple las condiciones siguientes:

- En las celdas de la tabla bidimensional debe haber valores individuales.
- Todas las entradas en cualquier columna deben ser del mismo tipo.
- Cada columna debe tener un nombre único y no importa el orden de las columnas en la tabla.
- No puede haber dos renglones idénticos
- Todos sus valores son atómicos

**Segunda forma normal (FN2):** Una relación R satisface la FN2 sí; y sólo sí, satisface las reglas de FN1 y todos sus atributos no-llave dependen por completo del atributo llave, es decir, cada tabla que tiene un atributo único como llave, está en FN2.

**Tercera forma normal (FN3):** Una relación está en tercera forma normal si está en la segunda forma normal y no tiene dependencias transitivas.

## 2.2 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE ARQUITECTURA DE RED

Una red de computadoras consta de dos o más computadoras conectadas entre sí que permiten compartir recursos, servicios e información, la cual suele consistir en archivos o datos. A través de compartir información y recursos en una red, los usuarios de los sistemas informáticos de una organización podrán hacer un mejor uso de los mismos, mejorando de este modo el rendimiento global de la organización. Gracias a la instalación de redes se puede:

Compartir programas y archivos

- Compartir recursos
- Compartir bases de datos
- Utilizar software de red
- Creación de grupos de trabajo
- Gestión centralizada
- Acceso a más de un sistema operativo

### COMPONENTES DE UNA RED:

- **Servidor:** Éste ejecuta el sistema operativo de red y ofrece los servicios de red a las estaciones de trabajo.
- **Estaciones de trabajo:** Cuando una computadora se conecta a una red, la primera se convierte en un nodo de la última y se puede tratar como una estación de trabajo o cliente.
- **Tarjetas de interfaz de red:** Toda computadora que se conecta a una red necesita de una tarjeta de interfaz de red que soporte un esquema de red específico, como Ethernet, ArcNet o Token Ring.
- **Sistema de cableado:** El sistema de la red está constituido por el cable utilizado para conectar entre sí el servidor y las estaciones de trabajo.
- **Recursos y periféricos compartidos:** Entre los recursos compartidos se incluyen los dispositivos de almacenamiento ligados al servidor, las unidades de discos ópticos, las impresoras, los trazadores y el resto de equipos que puedan ser utilizados por cualquiera en la red.

### Dispositivos De Interconexión De Redes

- **Repeater (Repetidor):** Dispositivo no inteligente que permite la interconexión de equipos y genera señales en la red para llegar más lejos. Actúa sobre el nivel más bajo de la jerarquía de protocolos y normalmente funciona a la misma velocidad de transmisión que las redes que conecta.
- **Hub (Concentrador):** Es un repetidor multipuerto.
- **Bridge (Puente):** Dispositivo que conecta dos redes a nivel capa de enlace, con el mismo protocolo y medio de transmisión.
- **Router (Ruteadores):** Dispositivo utilizado para interconectar redes que usan el mismo protocolo, sobre el mismo o diferente medio físico de comunicación. Pueden usarse para ligar geográficamente redes dispersas al mismo tiempo.
- **Gateway (Pasarelas):** Dispositivo que interconecta sistema de redes diferentes en tiempo real entre un usuario y una de las varias estaciones remotas.
- **Host (Estaciones de trabajo)** Dispositivo con un nombre único que ofrece servicios a otros ordenadores conectados en la red.
- **Switch (Conmutador)** Dispositivo que trabaja a nivel capa de enlace de datos, como los bridges, permite interconexiones de múltiples segmentos físicos de la red local en una sola de gran tamaño.

### Tipos de Redes

**Local Área Network (LAN, Redes de Área Local)** es un sistema de comunicaciones de alta velocidad que conecta microcomputadoras o PC y/o periféricos que se encuentran cercanos, por lo general dentro del mismo edificio. Una LAN consta de hardware y software de red y sirve para conectar las que están aisladas. Los programas y utilidades que componen el software de la LAN, hacen de puente de unión entre el usuario y el núcleo central de la computadora.

### **Ventajas y desventajas**

- Restricción mínima y máxima en la extensión del alcance
- Restricción máxima de velocidad (decenas o cientos de megabits por segundo )
- Restricción en número de nodos (de treinta a cincuenta).
- Capacidad de conexión con otras redes locales o de área extensa.
- Propiedad privada.
- Presenta baja tasa de error en transmisiones de datos
- Permite la integración en la misma red de una gran cantidad de dispositivos
- La fibra óptica ofrece un medio seguro, confiable y libre de ruido.

**Metropolitan Area Network (MAN, Redes de Área Metropolitana)** es un sistema de interconexión de equipos informáticos distribuidos en una zona que abarca diversos edificios, por medios pertenecientes a la misma organización propietaria de los equipos. Este tipo de redes se utiliza normalmente para interconectar redes de área local.

### Ventajas y desventajas

- Integración de varios puntos en el mismo enlace
- Posibilidad de incremento hacia otros puntos para integración de la misma red
- Transmisión multimedia
- Servicio de banda ancha
- Permite superar los quinientos nodos de red
- Permite la distancia entre nodos de varios kilómetros
- Restricción máxima en la extensión del alcance
- Tiempos de acceso mínimos y servicios síncronos para aplicaciones en tiempo real
- La fibra óptica ofrece un medio seguro, confiable y libre de ruido.



**Wide Area Network (WAN, Redes de Área Extensa):** Es un sistema de interconexión de equipos geográficamente dispersos, incluso a nivel continental. El sistema de conexión normalmente involucra a redes públicas.

**Ventajas y desventajas**

- Pueden utilizar un software especializado para incluir mini o macrocomputadoras como elementos de red.
- No se limita al espacio geográfico para establecer comunicación
- Puede utilizar enlaces a satélites
- La fibra óptica ofrece un medio seguro, confiable y libre de ruido.
- Los equipos deben poseer gran capacidad de memoria para un acceso rápido
- Poca seguridad en los equipos (virus y troyanos entre otros)

**Tabla de clasificación de redes.**

TIPO	DISTANCIA	ALCANCE
LAN	10 m	Habitación
	100 m	Edificio
	1 km	Localidad
MAN	50 km	Ciudad
WAN	100 km	País
	1000 km	Continente
INTERNET	10,000 km	Planeta

**Arquitectura De Red**

Se define por su topología, el método de acceso a la red y protocolos de comunicación. Se llama topología de una Red al patrón de conexión entre sus nodos, es decir, a la forma en que están interconectados los distintos nodos que la forman. Los criterios a la hora de elegir una topología, en general, buscan que eviten el coste del encaminamiento (necesidad de elegir los

caminos más simples entre el nodo y los demás), dejando en segundo plano factores como la renta mínima, el coste mínimo, etc. Otro criterio determinante es la tolerancia a fallos o facilidad de localización de éstos. También tenemos que tener en cuenta la facilidad de instalación y reconfiguración de la Red. Existen distintos tipos de topología:

Hay dos clases generales de topología utilizadas en Redes de Área Local: Topología tipo Bus y Topología tipo Anillo. A partir de ellas derivan otras que reciben nombres distintos dependiendo de las técnicas que se utilicen para acceder a la Red o para aumentar su tamaño.

**Topología en bus:** Una Red en forma de Bus o Canal de difusión es un camino de comunicación bidireccional con puntos de terminación bien definidos. Cuando una estación transmite, la señal se propaga a ambos lados del emisor hacia todas las estaciones conectadas al Bus hasta llegar a las terminaciones del mismo. Así, cuando una estación transmite su mensaje alcanza a todas las estaciones, por esto el Bus recibe el nombre de canal de difusión.

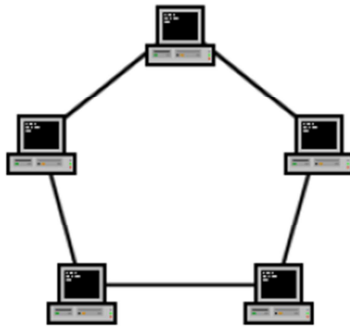
La topología Bus actúa como medio pasivo y por lo tanto, en caso de extender la longitud de la red, el mensaje no debe ser regenerado por repetidores (los cuales deben ser muy fiables para mantener el funcionamiento de la red). En este tipo de topología cualquier ruptura en el cable impide la operación normal y es muy difícil de detectar (ver figura 2.2.1). Por el contrario, el fallo de cualquier nodo no impide que la red siga funcionando normalmente, lo que permite añadir o quitar nodos a la red sin interrumpir su funcionamiento.



**Figura 2.2.1 Topología de bus.**

Una variación de la topología en Bus es la de árbol, en la cual el Bus se extiende en más de una dirección facilitando el cableado central al que se le añaden varios cables complementarios. La técnica que se emplea para hacer llegar la señal a todos los nodos es utilizar dos frecuencias distintas para recibir y transmitir. Las características descritas para el Bus siguen siendo válidas para el árbol.

**Topología en anillo:** Esta se caracteriza por un camino unidireccional cerrado que conecta todos los nodos, figura 2.2.2. Dependiendo del control de acceso al medio, se dan nombres distintos a esta topología: Bucle; se utiliza para designar aquellos anillos en los que el control de acceso está centralizado (una de las estaciones se encarga de controlar el acceso a la red). Anillo; se utiliza cuando el control de acceso está distribuido por toda la red. Como las características de uno y otro tipo de la red son prácticamente las mismas, se utiliza el término anillo para las dos.

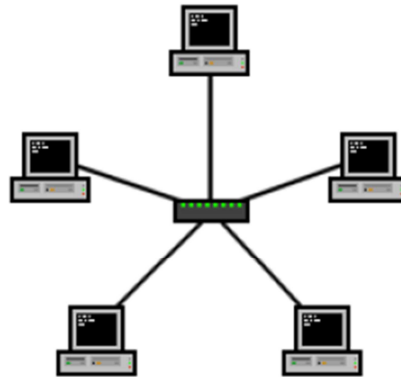


**Figura 2.2.2 Topología de anillo.**

En cuanto a fiabilidad, presenta características similares al Bus: la avería de una estación puede aislarse fácilmente, pero una avería en el cable inutiliza la red. Sin embargo, un problema de este tipo es más fácil de localizar, ya que el cable se encuentra físicamente dividido por las estaciones.

**Topología estrella:** La topología en estrella se caracteriza por tener todos sus nodos conectados a un controlador central. Todas las transacciones pasan a

través del nodo central, siendo éste el encargado de gestionar y controlar todas las comunicaciones (ver figura 2.2.3). Por este motivo, el fallo de un nodo en particular es fácil de detectar y no daña el resto de la red, pero un fallo en el nodo central desactiva la red completa.



**Figura 2.2.3 Topología de estrella.**

### **La Suite TCP/IP**

Internet es un conglomerado muy amplio y extenso en el que se encuentran computadoras con sistemas operativos incompatibles, redes más pequeñas y distintos servicios con su propio conjunto de protocolos para la comunicación. Ante tanta diversidad resulta necesario establecer un conjunto de reglas comunes para la comunicación entre estos diferentes elementos y que además optimice la utilización de recursos tan distantes. Este papel lo tiene el protocolo TCP/IP. Éste también puede usarse como protocolo de comunicación en las redes privadas intranet y extranet.

Los diferentes protocolos de la suite TCP/IP trabajan conjuntamente para proporcionar el transporte de datos dentro de Internet (o Intranet). En otras palabras, hacen posible que accedamos a los distintos servicios de la Red. Estos servicios incluyen, transmisión de correo electrónico, transferencia de archivos, grupos de noticias, acceso a la World Wide Web, etc.

Hay dos clases de protocolos dentro de la suite TCP/IP que son los protocolos a nivel de red y protocolos a nivel de aplicación, los cuales se describen a continuación:

### **Protocolos a Nivel de Red**

Estos protocolos se encargan de controlar los mecanismos de transferencia de datos. Normalmente son invisibles para el usuario y operan por debajo de la superficie del sistema. Dentro de estos protocolos tenemos:

**TCP:** Controla la división de la información en unidades individuales de datos (llamadas paquetes) para que estos paquetes sean encaminados de la forma más eficiente hacia su punto de destino. En dicho punto, TCP se encargará de re ensamblar dichos paquetes para reconstruir el archivo o mensaje que se envió. Por ejemplo, cuando se nos envía un archivo HTML desde un servidor Web, el protocolo de control de transmisión en ese servidor divide el archivo en uno o más paquetes, numera dichos paquetes y se los pasa al protocolo IP. Aunque cada paquete tenga la misma dirección IP de destino, puede seguir una ruta diferente a través de la red. Del otro lado (el programa cliente en nuestra computadora), TCP reconstruye los paquetes individuales y espera hasta que hayan llegado todos para presentárnoslos como un solo archivo.

**IP:** Se encarga de repartir los paquetes de información enviados entre la computadora local y las computadoras remotas. Esto lo hace etiquetando los paquetes con una serie de información, entre la que cabe destacar las direcciones IP de las dos computadoras. Basándose en esta información, IP garantiza que los datos se encaminarán al destino correcto. Los paquetes recorrerán la red hasta su destino (que puede estar en el otro extremo del planeta) por el camino más corto posible gracias a unos dispositivos denominados ruteadores.

### **Protocolos a Nivel de Aplicación**

Estos protocolos son visibles para el usuario en alguna medida. Por ejemplo, el protocolo FTP (File Transfer Protocol) es visible para el usuario. El usuario solicita una conexión a otra computadora para transferir un archivo, la conexión se establece, y comienza la transferencia. Durante dicha transferencia, es visible parte del intercambio entre la máquina del usuario y la máquina remota.

## Funcionamiento de TCP/IP

TCP/IP opera a través del uso de una pila. Dicha pila es la suma total de todos los protocolos necesarios para completar una transferencia de datos entre dos máquinas (así como el camino que siguen los datos para dejar una máquina o entrar en la otra). La pila está dividida en capas, como se ilustra en la figura siguiente:

EQUIPO SERVIDOR O CLIENTE	
Capa de Aplicaciones	Cuando un usuario inicia una transferencia de datos, esta capa pasa la solicitud a la Capa de Transporte.
Capa de Transporte	La Capa de Transporte añade una cabecera y pasa los datos a la Capa de Red.
Capa de Red	En la Capa de Red, se añaden las direcciones IP de origen y destino para el enrutamiento de datos
Capa de Enlace de Datos	Ejecuta un control de errores sobre el flujo de datos entre los protocolos anteriores y la Capa Física.
Capa Física	Ingresa o egresa los datos a través del medio físico, que puede ser Ethernet vía coaxial, PPP vía módem, etc.

Después de que los datos han pasado a través de este proceso, viajan a su destino en otra máquina de la red. Allí, el proceso se ejecuta al revés (los datos entran por la capa física y recorren la pila hacia arriba)

### Direcciones IP

Una dirección IP consiste en cuatro números separados por puntos, estando cada uno de ellos en el rango de 0 a 254. Por ejemplo, una dirección IP válida sería 193.146.85.34. Cada uno de los números decimales representa una cadena de ocho dígitos binarios. De este modo, la dirección anterior sería realmente la cadena de ceros y unos:

11000001.10010010.01010101.00100010

La versión actual del protocolo IP (la versión 4 o IPv4) define de esta forma direcciones de 32 bits, lo que quiere decir que hay 2<sup>32</sup> (4.294.967.296) direcciones IPv4 disponibles.

## 2.3 CARACTERÍSTICAS VENTAJAS Y DESVENTAJAS DE APLICACIONES WEB.

En el mundo de la ingeniería de software se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web (HTML, JavaScript, Java, php, etc.) en la que se confía la ejecución al navegador.

Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales.

Una aplicación Web es un conjunto de páginas Web estáticas y dinámicas. Una *página Web estática* es aquella que no cambia cuando un usuario la solicita: el servidor Web envía la página al navegador Web solicitante sin modificarla. Por el contrario, el servidor modifica las *páginas Web dinámicas* antes de enviarlas al navegador solicitante. La naturaleza cambiante de este tipo de página es la que le da el nombre de dinámica.

Cabe mencionar que una página Web puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones, como por ejemplo rellenar y enviar formularios.

### **Características**

Las aplicaciones web generan dinámicamente una serie de páginas en un formato estándar, como HTML o XHTML, que son soportados por los navegadores web comunes. Se utilizan lenguajes que a su vez serán interpretados por el navegador, tales como JavaScript, para añadir elementos dinámicos a la interfaz de usuario. Generalmente cada página web se envía al

cliente como un documento estático, pero la secuencia de páginas ofrece al usuario una experiencia interactiva. Durante la sesión, el navegador web interpreta y muestra en pantalla las páginas, actuando como cliente para cualquier aplicación Web.

### **Estructura de las aplicaciones Web**

Aunque existen muchas variaciones posibles, una aplicación Web está normalmente estructurada como una aplicación de tres-capas. En su forma más común, el navegador Web ofrece la primera capa y un motor capaz de usar alguna tecnología Web dinámica (ejemplo: PHP, Java) constituye la capa de en medio. Por último, una base de datos constituye la tercera y última capa.

### **Procesamiento de páginas Web estáticas**

Un sitio Web estático consta de un conjunto de páginas y de archivos HTML relacionados alojados en un equipo que ejecuta un servidor Web.

Un servidor Web es un software que suministra páginas Web en respuesta a las peticiones de los navegadores Web. La petición de una página se genera cuando el usuario hace clic en un vínculo de una página Web, cuando introduce una URL en el cuadro de texto Dirección del navegador.

Cuando el servidor Web recibe una petición de una página estática, el servidor lee la solicitud, localiza la página y la envía al navegador solicitante, como se muestra en la figura 2.3.1



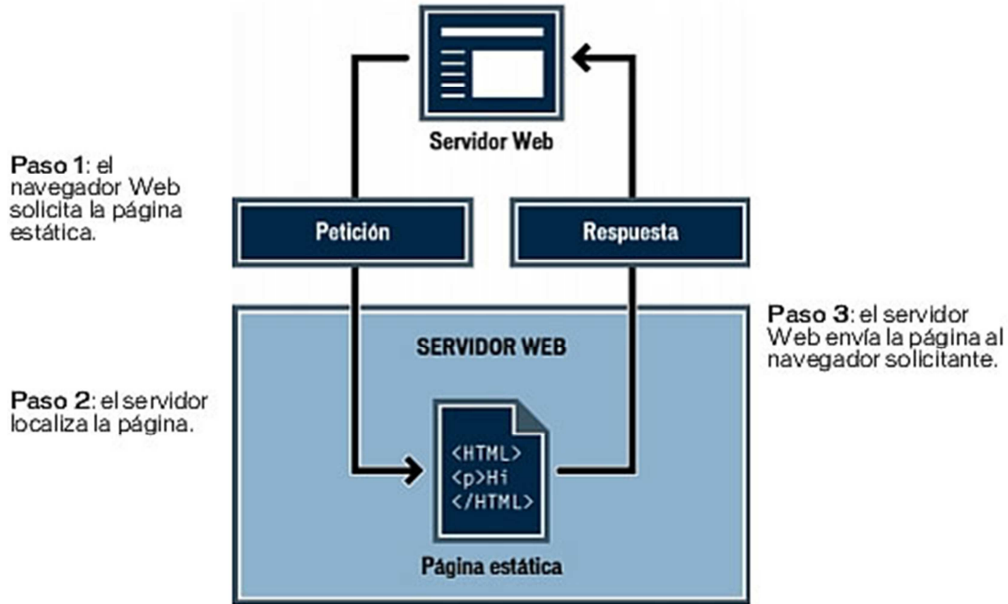


Figura 2.3.1 Procesamiento de una página WEB estática.

### Procesamiento de páginas dinámicas

Cuando un servidor Web recibe una petición para mostrar una página Web estática, el servidor la envía directamente al navegador que la solicita. Cuando el servidor Web recibe una petición para mostrar una página dinámica, sin embargo, reacciona de distinta forma: transfiere la página a un software especial encargado de finalizar la página. Este software especial se denomina servidor de aplicaciones.

El servidor de aplicaciones lee el código de la página, finaliza la página en función de las instrucciones del código y elimina el código de la página. El resultado es una página estática que el servidor de aplicaciones devuelve al servidor Web, que a su vez la envía al navegador solicitante. Lo único que el navegador recibe cuando llega la página es código HTML puro. A continuación se incluye una vista de este proceso en la figura 2.3.2

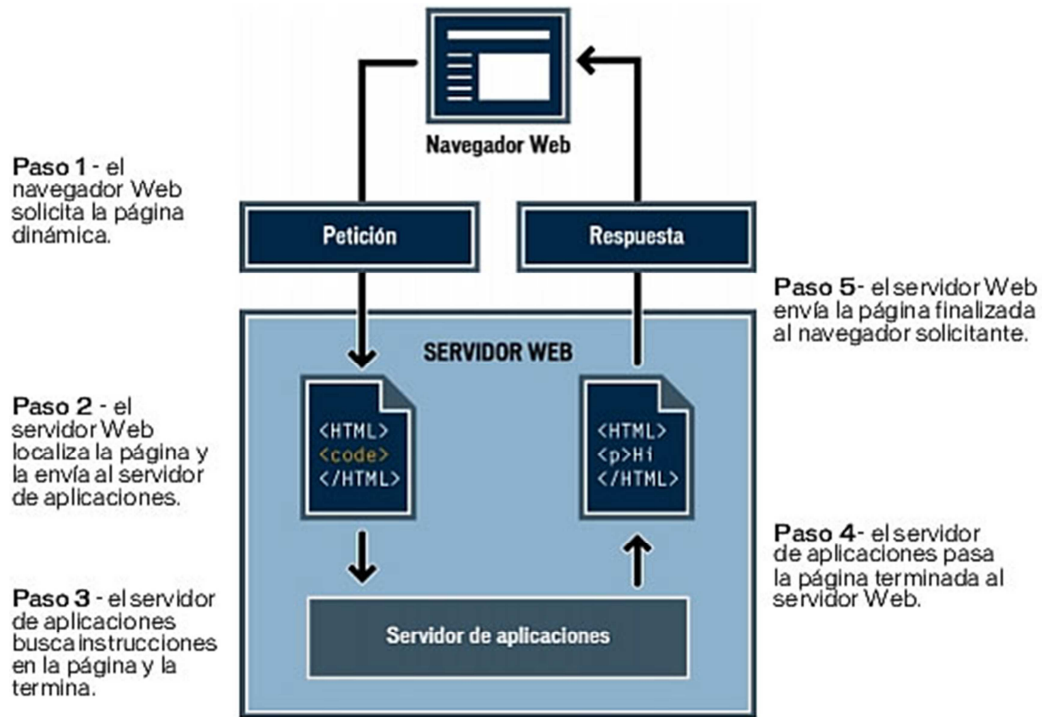


Figura 2.3.2 Procesamiento de una página WEB dinámica.

### Acceso a una base de datos

Un servidor de aplicaciones le permite trabajar con recursos del lado del servidor, como las bases de datos. Por ejemplo, una página dinámica puede indicar al servidor de aplicaciones que extraiga datos de una base de datos y los inserte en el código HTML de la página

A continuación en la figura 2.3.3 se ofrece una ilustración del proceso de consulta de base de datos y de devolución de los datos al navegador.

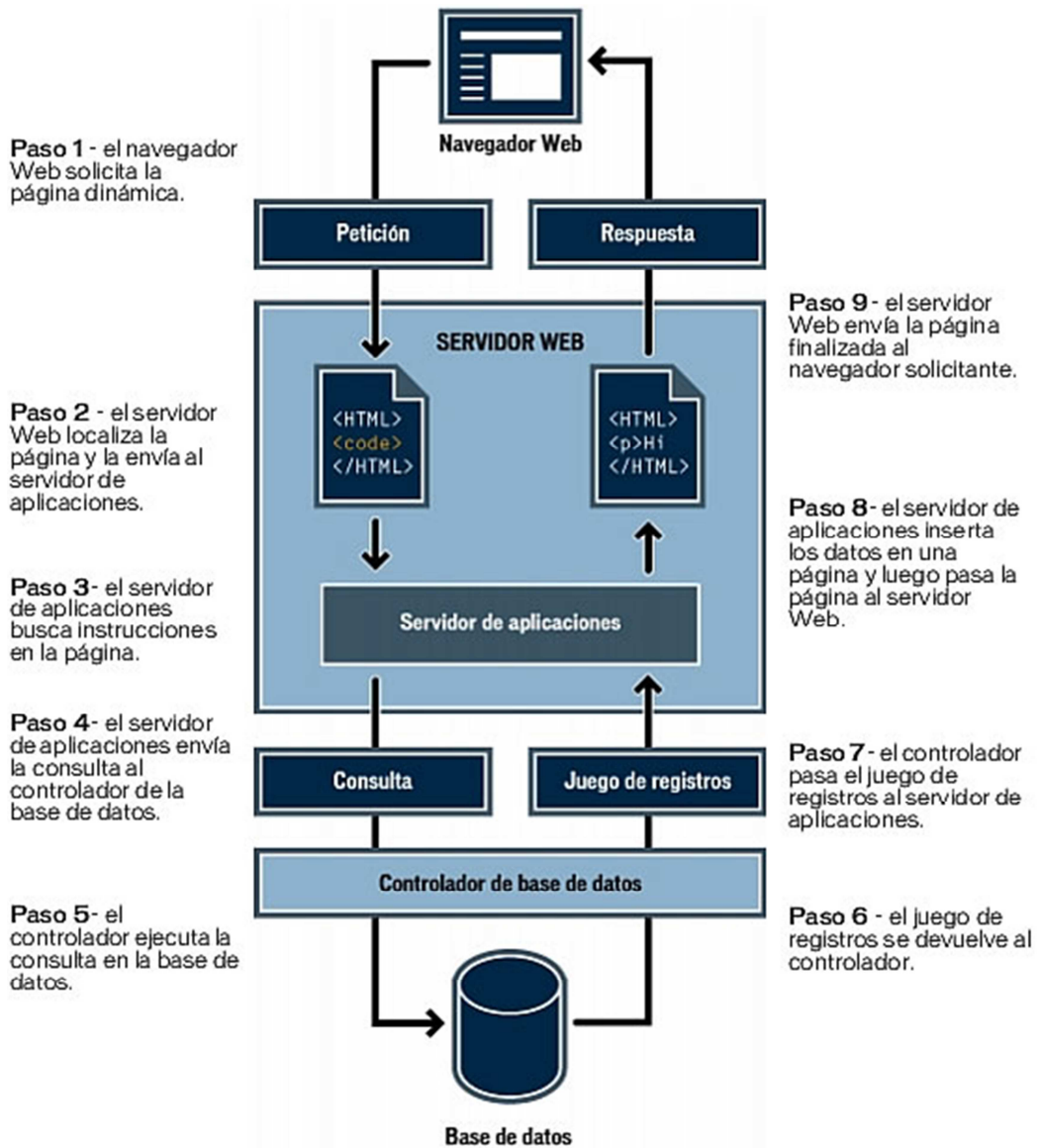


Figura 2.3.3 Proceso del Accesos a una Base de Datos

Puede utilizar prácticamente cualquier base de datos con su aplicación Web, siempre y cuando se haya instalado el controlador de base de datos correcto en el servidor.

## **Ventajas**

Una ventaja significativa es que las aplicaciones web funcionan igual independientemente de la versión del sistema operativo instalado en el cliente. En vez de crear clientes para Windows, Mac OS X, Linux, y otros sistemas operativos, la aplicación web se escribe una vez y se ejecuta igual en todas partes. Entre otras ventajas podemos destacar las siguientes:

- Ahorra tiempo: Se pueden realizar tareas sencillas sin necesidad de descargar ni instalar ningún programa.
- No hay problemas de compatibilidad: Basta tener un navegador mínimamente actualizado para poder utilizarlas.
- No se necesita Instalar nada (No depende de algún Software), por lo general.
- Brindan privacidad con acceso (usuario y contraseña) para acceder a tus datos.
- No necesitas actualizarlo.
- Recuperas tu contraseña perdida con un email alternativo.
- No ocupan espacio en nuestro disco duro.
- Actualizaciones inmediatas: Como el software lo gestiona el propio desarrollador, cuando nos conectamos estamos usando siempre la última versión que haya lanzado.
- Consumo de recursos bajo: Dado que la aplicación no se encuentra en nuestro ordenador, muchas de las tareas que realiza el software no consumen recursos nuestros porque se realizan desde otro ordenador.
- Multiplataforma: Se pueden usar desde cualquier sistema operativo porque sólo es necesario tener un navegador.
- La disponibilidad suele ser alta porque el servicio se ofrece desde múltiples localizaciones para asegurar la continuidad del mismo.
- Los virus no dañan los datos porque éstos están guardados en el servidor de la aplicación.

### Desventajas

Habitualmente ofrecen menos funcionalidades que las aplicaciones de escritorio. Se debe a que las funcionalidades que se pueden realizar desde un navegador son más limitadas que las que se pueden realizar desde el sistema operativo.

- Si no se dispone de conexión a Internet no se pueden usar.
- La disponibilidad depende de un tercero, el proveedor de la aplicación.
- Depende de una conexión a Internet permanente (generalmente) y una conexión promedio para una óptima navegación.
- La seguridad de los datos, depende de la seguridad de la aplicación Web y del servidor donde este alojado.
- Está limitado el espacio donde se guarda la información (solo si necesitas guardar videos, imágenes, audio).
- Poco soporte a varios Idiomas (generalmente).
- Dependen mucho de la configuración de la privacidad del navegador para trabajar al 100% (JavaScript, Cookies, Flash).
- La estabilidad de la aplicación está sujeta al número de visitas en un mismo lapso de tiempo (sobre carga del servidor).

## 2.4 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE APLICACIONES PHP

**PHP Hypertext Pre-processor (PHP)** es un lenguaje de programación interpretado, diseñado para la creación de páginas web dinámicas, usado principalmente en interpretación del lado del servidor y que actualmente puede ser utilizado desde una interfaz de línea de comandos.

### Características de PHP

Diseñado especialmente para aplicaciones web PHP, similar al ASP de Microsoft o el JSP de Sun, puede ser incrustado dentro de código HTML el cual se ejecuta en un servidor web, tomando como entrada el código hecho en PHP y dando como salida la creación de páginas web.

## MARCO TEÓRICO

La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas. No es un lenguaje de marcas como podría ser HTML, XML o WML. Está más cercano a JavaScript o a C.

A diferencia de Java o JavaScript que se ejecutan en el navegador, PHP se ejecuta del lado del servidor por eso nos permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos. El programa PHP es ejecutado en el servidor y el resultado es enviado al navegador. El resultado es normalmente una página HTML pero también podría ser una página WML (Wap).

Al ser un lenguaje libre dispone de una gran cantidad de características lo convierten en la herramienta ideal para la creación de páginas web dinámicas:

- Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.
- Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación.
- Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.
- Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.

- El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.
- Con PHP se puede hacer cualquier cosa que podemos realizar con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas.

Es un lenguaje middleware cuyos programas se llaman scripts los cuales están incrustados dentro de la página HTML. El término scripts se refiere a archivos en lenguajes que son almacenados de manera casi idéntica a cuando son escritos y que son traducidos en el momento de su ejecución.

Hay tres áreas principales donde se usan scripts PHP:

- **Server-side scripting.** Este es el campo más tradicional y el principal foco. Se necesitan tres cosas para hacer este trabajo. El intérprete PHP (CGI ó módulo), un servidor web y un navegador web. Tiene que ejecutar el servidor web con PHP instalado. Puede acceder a la salida del programa PHP con un navegador web, visualización de la página PHP a través del servidor. Todos estos se pueden ejecutar en el equipo de origen si son sólo la experimentación con programación PHP.
- **Secuencias de comandos de línea de comandos.** Se puede crear un script PHP para ejecutarlo sin ningún tipo de servidor o el navegador. Sólo necesita el intérprete PHP para usarlo de esta manera. Este tipo de uso es ideal para scripts ejecutados regularmente desde Cron (en \* nix o Linux) el cual sirve para automatizar tareas, comúnmente para realizar copias de seguridad de archivos o el Programador de tareas (en Windows). Estos scripts también pueden ser utilizados para tareas simples de procesamiento de texto.
- **Escribir aplicaciones de escritorio.** PHP no es probablemente el mejor lenguaje para crear una aplicación de escritorio con una interfaz gráfica de usuario, pero si se conoce bien PHP, y quisiera utilizar algunas

características avanzadas de PHP en sus aplicaciones cliente también se puede usar PHP-GTK para escribir dichos programas. Se tiene la habilidad de escribir aplicaciones multiplataforma de esta manera. PHP-GTK es una extensión de PHP, no disponible en la distribución principal.

### **Modo de operación del PHP**

- El Navegador realiza una petición al servidor (se escribe la URL).
- Después el servidor ejecuta el código PHP solicitado y retorna el código HTML generado al Navegador.
- Por último el Navegador muestra la respuesta del servidor.

Este tipo de iteración permite algunas operaciones complejas como conexiones a bases de datos o ejecución de complejos programas. PHP además de soportar un número masivo de bases de datos, también nos ofrece una gran variedad de opciones que nos permiten desarrollar múltiples funcionalidades que van desde enviar un e-mail, subir un archivo (upload), crear una imagen en tiempo de ejecución, interactuar con diversos protocolos de comunicación, interactuar con documentos XML, autenticación, creación dinámica de documentos PDF, entre muchas otras cosas.

Las principales características de PHP son: su rapidez; su facilidad de aprendizaje; su soporte multiplataforma tanto de diversos Sistemas Operativos, como servidores HTTP y de bases de datos; y el hecho de que se distribuye de forma gratuita bajo una licencia abierta. Ver figura 2.4.1





Figura 2.4.1 Funcionamiento de PHP

### Ventajas de PHP

PHP puede ser usado en todos los principales sistemas operativos, incluyendo Linux, muchas variantes de Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS, y probablemente otras. PHP también soporta la mayoría de los servidores web de hoy en día. Esto incluye Apache, Microsoft Internet Information Server, Personal Web Server, los servidores Netscape e iPlanet, Oreilly Website Pro Server, Caudium, Xitami, OmniHTTPd, y muchos otros. Para la mayoría de los servidores, PHP tiene un módulo, para los otros que soporten el estándar CGI, PHP puede funcionar como un procesador CGI.

Así que con PHP, se tiene la libertad de elegir el sistema operativo y un servidor web. Además, también tiene la opción de usar programación procedural o programación orientada a objetos, o una mezcla de ellos. Aunque no todas las características estándar de programación orientada a objetos se implementan en PHP 4, bibliotecas de código y aplicaciones de gran tamaño (incluyendo la biblioteca PEAR) están escritas íntegramente usando

programación orientada a objetos. PHP 5, determina las deficiencias de programación orientada a objetos relacionados con PHP 4, e introduce un modelo de objetos completo.

Con PHP no se limitan a la salida HTML, pues entre sus habilidades se incluyen: creación de imágenes, archivos PDF y películas Flash (usando libswf y Ming) sobre la marcha. También se puede producir fácilmente un texto, como XHTML y archivos XML. PHP puede autogenerar estos archivos y guardarlos en el sistema de archivos, en lugar de imprimir a cabo, la formación de un servidor caché para el contenido dinámico.

Al ser uno de los productos más representativos y utilizados del Open Source, el cual es un movimiento global, en el cual el código fuente del software se hace disponible a cualquier persona que pueda examinar su funcionamiento interno para hacer sugerencias y cambios.

PHP 5 es, al momento de redactar parte de este capítulo, la versión más estable liberada. El equipo de desarrollo de PHP incluye docenas de programadores, así como otras docenas de personas trabajando en proyectos relacionados con PHP como PEAR y el proyecto de documentación.

### **Seguridad**

Se puede acceder a ficheros, ejecutar comandos y abrir comunicaciones de red en el servidor. Todas estas características hacen que lo que se ejecute en el servidor Web sea seguro por defecto.

PHP ha sido diseñado específicamente para ser un lenguaje más seguro para escribir programas CGI, Perl o C y con la correcta selección de las opciones de configuración de tiempo de compilación y ejecución se consigue la exacta combinación de libertad y seguridad que se necesita. Ya que existen diferentes modos de utilizar PHP, existe también una multitud de opciones de configuración que permiten controlar su funcionamiento. Una gran selección de opciones garantiza que se pueda usar PHP para diferentes aplicaciones, pero también significa que existen combinaciones de estas opciones y configuraciones del servidor que producen instalaciones inseguras.

El sistema debe poseer protecciones contra ataques. PHP provee diferentes niveles de seguridad, estos pueden ser configurados desde el archivo \*.ini

### **Velocidad**

No solo la velocidad de ejecución, la cual es importante, sino además no crear demoras en la máquina. Por esta razón no debe requerir demasiados recursos de sistema. PHP se integra muy bien junto a otro software, especialmente bajo ambientes Unix, cuando se configura como módulo de Apache, está listo para ser utilizado.

### **Estabilidad**

La velocidad no sirve de mucho si el sistema se cae cada cierta cantidad de ejecuciones. Ninguna aplicación es 100% libre de bugs, pero teniendo de respaldo una increíble comunidad de programadores y usuarios es mucho más difícil para los bugs sobrevivir. PHP utiliza su propio sistema de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.

### **Simplicidad**

Se les debe permitir a los programadores generar código productivamente en el menor tiempo posible. Usuarios con experiencia en C y C++ podrán utilizar PHP rápidamente.

Por último pero no menos importante, tenemos muchas otras extensiones muy interesantes, las funciones del motor de búsquedas, funciones para pasarelas de IRC, utilidades de compresión (gzip, bz2), conversión de calendarios, traducción.

### **Desventajas de PHP**

- El manejo de errores no es tan sofisticado como Cold Fusion o ASP.
- No existe IDE o Debugger, aunque una IDE puede no ser importante para la mayoría de los programadores.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- PHP consume más recursos (y por tanto es ligeramente más lento) llamando y ejecutando una función que ejecutando código que

encuentra en línea, embebido en el script. Por eso no tiene sentido usar funciones salvo que efectivamente se vaya a utilizar esa porción de código varias veces.

## **2.5 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE APACHE**

### **Servidor Web**

Un servidor Web es un programa que implementa el protocolo HTTP (HyperText Transfer Protocol). Está diseñado para transferir páginas HTML (HyperText Markup Language). Se ejecuta continuamente en un ordenador (también se emplea el término para referirse al equipo que lo ejecuta), manteniéndose a la espera de peticiones por parte de un cliente (un navegador Web) y que responde a estas peticiones adecuadamente, mediante una página Web que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error.

Se pueden utilizar varias tecnologías en el servidor para aumentar su potencia más allá de su capacidad de entregar páginas HTML, éstas incluyen scripts CGI, seguridad SSL y aplicaciones del lado del servidor.

Un servidor Web aporta diversas ventajas importantes como son:

- Es un sistema abierto al cual se puede acceder y realizar transacciones desde cualquier punto de la red.
- La información entregada puede ser de diversos tipos (datos, documentos, multimedia, etc. Generalmente esta es de solo lectura por lo que el usuario no puede modificar o alterar las fuentes originales.
- Dado que el servidor Web es de fácil acceso, se hace posible difundir información de forma instantánea mediante su almacenamiento en el servidor.

Se pueden almacenar aplicaciones en el servidor, lo cual representa diversas ventajas ya que al ejecutarse en el servidor, el cliente no requiere de algún tipo

de capacidad añadida, por lo que un cliente equipado con un navegador Web puede acceder sin dificultad alguna hacia la aplicación.

Un servidor Web es un programa que sirve para atender y responder a las diferentes peticiones de los navegadores, proporcionando los recursos que soliciten usando el protocolo HTTP o el protocolo HTTPS (la versión cifrada y autenticada). Un servidor web básico cuenta con un esquema de funcionamiento muy simple, basado en ejecutar infinitamente el siguiente ciclo:

1. Espera peticiones en el puerto TCP indicado (el estándar por defecto para HTTP es el 80)
2. Recibe una petición
3. Busca el recurso
4. Envía el recurso utilizando la misma conexión por la que recibió petición
5. Vuelve al segundo punto

Un servidor web que siga el esquema anterior cumplirá todos los requisitos básicos de los servidores HTTP, aunque sólo podrá servir páginas estáticas. Vea Fig. 2.5.1.

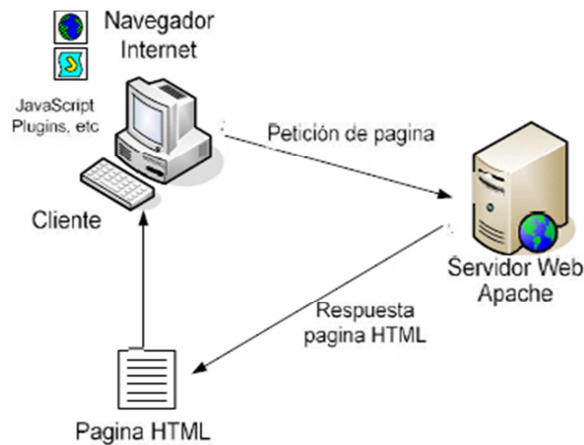


Fig. 2.5.1 Esquema de servidor Web de contenido estático

A partir del anterior esquema se han diseñado y desarrollado todos los servidores de HTTP que existen, variando sólo el tipo de peticiones (páginas estáticas, CGI's, etc.) que pueden atender.

Uno de los aspectos fundamentales del servidor Web es el nivel de soporte que ofrece para servir contenido dinámico. Puesto que la mayor parte del contenido Web que se sirve no viene de páginas estáticas, sino que se genera de forma dinámica, y esta tendencia se mueve claramente a la alza, el soporte para contenido de tipo dinámico que ofrece un servidor web es uno de los puntos críticos en la elección. Vea Fig. 2.5.2

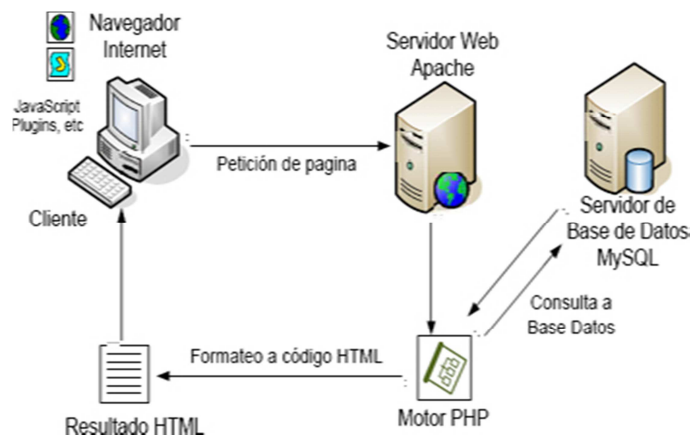


Fig. 2.5.2 Esquema de servidor Web de contenido dinámico

### Servidor Web Apache

Es un servidor Web de código abierto multiplataforma que implementa el protocolo HTTP, su objetivo es ofrecer un servidor Web confiable, eficiente extensible y de código fuente abierto. Es el servidor Web con mayor uso de la red desde hace mucho tiempo (según la lista de la Netcraft el 60% de los servidores Web de Internet son Apache). A continuación se muestra una descripción general de la arquitectura del servidor Apache. Vea Fig. 2.5.3

## Arquitectura de Apache

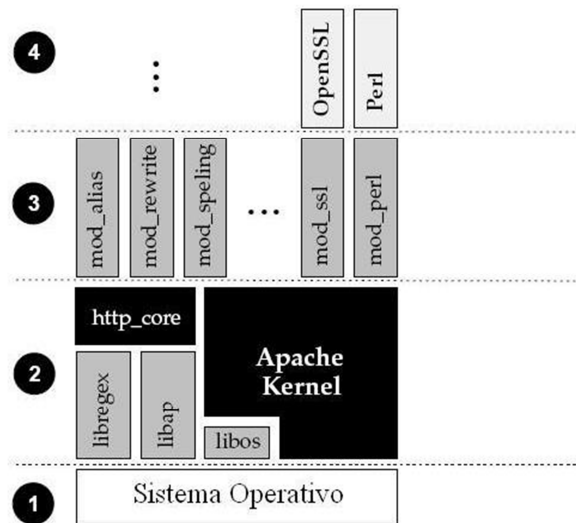


Fig. 2.5.3 Arquitectura de Apache

- **Funcionalidad de plataforma**, en esta capa se encuentra el sistema operativo, Apache es prácticamente universal por lo que es soportado por la gran mayoría de los sistemas operativos.
- **Funcionalidad básica**, en esta capa se sitúa el módulo http\_core que es el encargado junto con el Kernel del sistema, del funcionamiento básico del servidor, así como de proporcionar la API a la capa superior. Contiene librerías de código reusable (libap), y una librería que implementa parseo de expresiones regulares y abstracción del sistema operativo (libos)
- **Funcionalidad opcional**, en esta capa se encuentra el conjunto de módulos que otorgan funciones adicionales al servidor, como son módulos que permitan la compresión de contenido, funciones de seguridad, funcionalidades de caché, de autenticación, de comunicaciones seguras, etc.
- **Librerías de módulos externos**, en esta se albergan las distintas librerías de las cuales hacen uso módulos externos o desarrollados por terceros.

La razón principal por la que Apache es el servidor Web más usado es debido a las grandes ventajas que ofrece tanto a usuarios como a desarrolladores. Entre estas podemos mencionar las siguientes:

### **Eficiencia y estabilidad**

La eficiencia de Apache es una de sus grandes ventajas, ya que su código fuente (Apache se encuentra desarrollado en lenguaje C) se encuentra optimizado para consumir la menor cantidad de recursos del sistema. Esto le permite lograr un gran desempeño frente a gran cantidad de operaciones, por lo que lo convierte en una aplicación sumamente estable. Un ejemplo de esto es que actualmente el 90% de los servidores de alta disponibilidad en la red Internet confían en Apache.

### **Código Abierto**

Apache es de código abierto, existe por lo tanto una transparencia en la forma en que se lleva a cabo su desarrollo. El API de Apache es de dominio público por lo que pueden realizarse mejoras y adición de nuevos módulos al servidor, así como su adaptación a necesidades particulares. Las actualizaciones son constantes ya que existe una comunidad de desarrolladores a nivel global que mantiene y desarrolla el proyecto. Esto contribuye a su estabilidad ya que al aparecer errores en el sistema, no es necesario esperar a que un grupo particular de desarrolladores resuelva la falla, sino que esta se encuentra abierta para poder ser resuelta por un grupo global de los mismos.

### **Multiplataforma**

Apache corre prácticamente en la mayoría de las plataformas (UNIX, BSD, GNU/Linux, Windows, Macintosh, entre otras). Es una aplicación sumamente portable y configurable en diferentes ambientes.

### **Extensibilidad**

Apache es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache. Actualmente existen



muchos módulos para Apache que son adaptables a éste. Desde módulos que permiten generar contenido dinámico (con PHP, Java, Perl, Python), monitoreo del rendimiento, comunicaciones seguras por SSL, creación de servidores virtuales por IP. Otra cuestión importante es que cualquiera que posea una experiencia decente en la programación de C o Perl puede escribir un módulo para realizar una función determinada.

### **Configurable**

Apache puede configurarse de manera sencilla ya que cuenta con archivos de configuración específicos. Es posible configurar el servidor para que ejecute un determinado script cuando ocurra un error en concreto. Tiene una gran facilidad en su configuración para la creación y gestión de logs. Apache permite la creación de logs de errores configurados por el administrador.

### **Sencillez en la administración**

La administración es uno de los elementos principales de cualquier servidor. Apache cuenta con un conjunto de archivos de configuración ampliamente documentados, estos se encuentran en formato ASCII y pueden ser editados con cualquier editor de texto. Si se necesita realizar alguna actualización resulta sencillo respaldar la configuración del servidor. Se puede administrar de forma remota ya que puede operarse y configurarse desde la línea de comandos., así como por medio de interfaces gráficas y un conjunto de herramientas desarrolladas con este propósito.

### **Portabilidad y soporte**

Uno de los grandes beneficios de la comunidad de código abierto es que el soporte es mayor que en otro tipo de software en el mercado. Se cuenta con el Apache Group el cual es un grupo de usuarios dedicados al desarrollo de Apache así como compañías que distribuyen de manera comercial el software.

### **Desventajas de Apache**

- Un fallo en el sistema puede hacer que el servidor deje de funcionar, se puede tener un servidor de redundancia si el servicio que se brinda es de alta criticidad
- Una configuración errónea puede hacer que no funcione adecuadamente el servidor, se debe de tener la documentación necesaria respecto a los archivos de configuración esto para poder ajustar correctamente algunos parámetros del servidor
- El o los servidores donde se encuentre instalado Apache se ven afectados ante problemas de seguridad.

### **2.6 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE MySQL.**

MySQL es un sistema de administración de bases de datos relacionales de gran robustez, con un excelente desempeño, diseñado para soportar sistemas de producción de misión crítica con altas cargas de procesamiento. Es además, el manejador de base de datos más popular entre los producidos bajo los términos del Open Source, aunque existe una versión comercial distribuida y soportada por la compañía MySQL AB.

#### **Características**

Entre las características más importantes de MySQL se puede destacar:

- Comandos SQL anidados y soporte a tablas derivadas. Esto permite el uso de las cláusulas SQL EXISTS, IN, NOT EXISTS, NOT IN y UNION.
- Internacionalización. MySQL soporta UNICODE, admitiendo con esto el desarrollo de aplicaciones sin importar la plataforma o el lenguaje del país donde se genere. También permite el manejo interno de mensajes de error.

- Almacenamiento dinámico de comandos SQL. Esto hace que MySQL tenga un mejor desempeño que los demás manejadores de bases de datos. MySQL almacena en un espacio de memoria los resultados más recientes de comandos, lo cual permite un acceso inmediato a futuras solicitudes que impliquen datos procesados con anterioridad.
- Manejo transaccional seguro. InnoDB es un tipo de tabla que soporta transacciones, lo que permite decidir la confirmación o el rechazo de un conjunto de operaciones efectuadas con anterioridad en la base de datos. Asimismo soporta operaciones de revisión del estado, las cuales son utilizadas para transferir una base de datos a un determinado estado después de una falla seria del sistema.
- Conectividad para clientes. Los clientes pueden conectarse al servidor MySQL mediante TCP/IP o sockets de UNIX; utilizando aplicaciones hechas con C, C++, Java, Perl o PHP. Es importante mencionar que MySQL puede conectarse directamente a otros manejadores de bases de datos utilizando ODBC.
- Escalabilidad. MySQL puede manejar bases de datos con 50 millones de registros y más de 60,000 tablas. Soporta 64 índices por tabla y cada índice puede estar compuesto de 15 columnas o partes de ellas, con una longitud máxima de 1,000 bytes. Los índices pueden utilizar prefijos de columnas de tipo CHAR, VARCHAR, BLOB o TEXT.
- Seguridad. La utilización de privilegios y contraseñas, hace de MySQL un gestor de bases de datos relacionales flexible y muy seguro, todas las claves viajan encriptadas a través de la Web. Mediante la dirección IP del cliente, MySQL puede restringir y administrar el acceso a los recursos del servidor.
- Integridad. Para revisar, optimizar y reparar tablas u otros objetos de una base de datos, MySQL posee comandos y mecanismos internos.
- Procesamiento. MySQL puede manipular el multiprocesamiento distribuido en uno o más CPU's.
- Tipos de datos. Los tipos de datos que puede haber en un campo, se pueden agrupar en tres grandes grupos: numéricos, fecha y hora, y cadenas (caracteres).

A continuación se describen con más detalle algunas de las características mencionadas del manejador de bases de datos MySQL:

### **Ventajas**

MySQL fue originalmente desarrollada para manejar bases de datos muy grandes en forma más rápida que otras soluciones de bases de datos disponibles y ha estado en uso intenso en sistemas de producción de alta demanda por años. Está constantemente en desarrollo, lo que ofrece un rico y muy actualizado conjunto de funciones y características, así también ofrece interfaces con muchos tipos ODBC para exportar de otras aplicaciones como MS Access, DBase, Fox, etc.

- Rapidez
  - El código de MySQL fue escrito teniendo como meta principal un excelente desempeño.
  - Posee un sistema de direccionamiento de memoria rápido basado en **threads** (subprocesos o hilos).
  - Tablas en memoria que son usadas como tablas temporales.
  
- Escalabilidad
  - Soporte a grandes bases de datos. Se usa MySQL Server con bases de datos que contienen 50 millones de registros. También existen usuarios que usan MySQL Server con 60.000 tablas y cerca de 5.000.000 de registros.
  - Se permiten hasta 64 índices por tabla (32 antes de MySQL 4.1.2). Cada índice puede consistir desde 1 hasta 16 columnas o partes de columnas. El máximo ancho de límite son 1000 bytes (500 antes de MySQL 4.1.2). Un índice puede usar prefijos de una columna para los tipos de columna CHAR, VARCHAR, BLOB, o TEXT.

- Seguridad
  - Posee un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está encriptado cuando se conecta con un servidor.
  
- Conectividad
  - Los clientes pueden conectarse con el servidor MySQL usando sockets TCP/IP en cualquier plataforma. En sistemas Windows de la familia NT (NT, 2000, XP, ó 2003). los clientes pueden usar **named pipes** (conexión de software temporal entre dos programas o comandos) para la conexión. En sistemas Unix, los clientes se pueden conectar usando ficheros socket Unix.
  - En MySQL 5.0, los servidores Windows soportan conexiones con memoria compartida si se inicializan con la opción **shared-memory** (memoria compartida). Los clientes se pueden conectar a través de memoria compartida usando la opción `protocol=memory`.
  - La interfaz para el conector ODBC (MyODBC) proporciona a MySQL soporte para programas clientes que usen conexiones ODBC (Open Database Connectivity). Por ejemplo, puede usar MS Access para conectar al servidor MySQL. Los clientes pueden ejecutarse en Windows o Unix. El código fuente de MyODBC está disponible. Todas las funciones para ODBC 2.5 están soportadas, así como muchas otras.
  - La interfaz para el conector J MySQL proporciona soporte para clientes Java que usen conexiones JDBC. Estos clientes pueden ejecutarse en Windows o Unix. El código fuente para el conector J está disponible.
  
- Portabilidad
  - Escrito en C y en C++
  - Probado con un amplio rango de compiladores diferentes
  - Funciona en diferentes plataformas.

- Usa GNU Automake, Autoconf, y Libtool para portabilidad.
- APIs (Interfaz de Programación de Aplicaciones) disponibles para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, y Tcl.
- Uso completo de multi-hilo mediante hilos del kernel. Pueden usarse fácilmente múltiple CPU's si están disponibles.
- Relativamente sencillo de añadir otro sistema de almacenamiento. Esto es útil si se desea añadir una interfaz SQL para una base de datos propia.
- Las funciones SQL están implementadas usando una librería altamente optimizada y deben ser tan rápidas como sea posible. Normalmente no hay reserva de memoria tras toda la inicialización para consultas.
- El servidor está disponible como un programa separado para usar en un entorno de red cliente/servidor. También está disponible como biblioteca y puede ser incrustado (linkado) en aplicaciones autónomas. Dichas aplicaciones pueden usarse por sí mismas o en entornos donde no hay red disponible.

- Diseño Multicapa

El diseño de MySQL Server es multi capa, con módulos independientes. Algunos de los últimos módulos se listan a continuación con una indicación de lo bien probados que están:

- Replicación. Hay grandes grupos de servidores usando replicación en producción, con buenos resultados. Se trabaja para mejorar características de replicación en MySQL 5.x.
- InnoDB tablas. El motor de almacenamiento transaccional InnoDB es usado en grandes sistemas de producción con alta carga de trabajo.
- BDB tablas. El código Berkeley DB es muy estable, pero todavía se está trabajando para mejorarlo con el interfaz del motor de almacenamiento transaccional BDB en MySQL Server.

- MyODBC 3.51. Usa ODBC SDK 3.51 y es usado en sistemas de producción ampliamente. Algunas cuestiones surgidas parecen ser cuestión de las aplicaciones que lo usan e independientes del controlador ODBC o la base de datos subyacente.

### **Desventajas**

Como ya se ha comentado MySQL surgió como una necesidad de un grupo de personas sobre un manejador de bases de datos rápido, por lo que sus desarrolladores fueron implementando únicamente lo que precisaban, intentando hacerlo funcionar de forma óptima. Es por ello que, aunque MySQL se incluye en el grupo de sistemas de bases de datos relacionales, carece de algunas de sus principales características y aunque algunas de ellas ya se han incluido en las versiones más recientes no se puede asegurar la estabilidad y buen funcionamiento de estas:

- Subconsultas
- Triggers
- Procedimientos almacenados
- Transacciones
- Integridad referencial
- Índices.

A continuación se presenta una breve descripción de cada una de las características mencionadas como desventajas.

- Subconsultas
  - Tal vez ésta sea una de las características que más hacía falta, aunque gran parte de las veces que se necesitan, es posible reescribirlas de manera que no sean necesarias.
- Triggers

- Aunque éstos ya se incluyen en la última versión, no podemos hablar de una gran estabilidad ya que los triggers reducen de forma significativa el rendimiento de la base de datos, incluso en aquellas consultas que no los activan.
- Procedimientos almacenados
  - Los procedimientos almacenados fueron implementados hasta la versión 5, la cual aún presenta algunos problemas de estabilidad.
- Transacciones
  - A partir de las últimas versiones ya hay soporte para transacciones, aunque no por defecto (se ha de activar un modo especial).
- Integridad referencial
  - Aunque admite la declaración de claves foráneas en la creación de tablas, internamente no las trata de forma diferente al resto de los campos.
- Índices
  - MySQL utiliza a menudo índices para realizar operaciones de clasificación rápidamente sin examinar todas las filas.
  - Los índices agilizan la recuperación pero hace más lentas las operaciones de insertar, borrar, así como las actualizaciones de valores en las columnas indexadas (es decir la mayor parte de las operaciones que implican escritura) porque la escritura no afecta sólo a la fila de datos, sino a menudo también a los índices.
  - Los índices se actualizan cada vez que se modifica la columna o columnas que utiliza. Por ello no es aconsejable usar como índices columnas en las que serán frecuentes operaciones de escritura (INSERT, UPDATE, DELETE).



- Si tiene muchos índices, los ficheros pueden alcanzar el tamaño máximo de fichero antes que el de datos.

## **2.7 Características, ventajas y desventajas de aplicaciones HTML Java Script**

Introducción.

HTML, siglas de HyperText Markup Language (Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas Web. El lenguaje HTML puede ser creado y editado con cualquier editor de textos básico, como puede ser [Gedit](#) en Linux, el Bloc de Notas de Windows, o cualquier otro editor que admita texto sin formato como GNU Emacs, Microsoft Wordpad, [TextPad](#), entre otros.

Existen además, otros editores para la realización de sitios Web con características WYSIWYG (What You See Is What You Get, o en español: “lo que ves es lo que obtienes”). Estos editores permiten ver el resultado de lo que se está editando en tiempo real, a medida que se va desarrollando el documento. Ahora bien, esto no significa una manera distinta de realizar sitios Web, sino que una forma un tanto más simple ya que estos programas, además de tener la opción de trabajar con la vista preliminar, tiene su propia sección HTML la cual va generando todo el código a medida que se va trabajando. Algunos ejemplos de editores WYSIWIG son Macromedia Dreamweaver, o Microsoft FrontPage.

Combinar estos dos métodos resulta muy interesante, ya que de alguna manera se ayudan entre sí. Por ejemplo; si se edita todo en HTML y de pronto se olvida algún código o etiqueta, simplemente me dirijo al editor visual o WYSIWYG y se continúa ahí la edición, o viceversa, ya que hay casos en que sale más rápido y fácil escribir directamente el código de alguna característica que queramos adherirle al sitio, que buscar la opción en el programa mismo.

HTML es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. Se

escribe en forma de "etiquetas", rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo JavaScript), el cual puede afectar el comportamiento de navegadores Web y otros procesadores de HTML.

HTML también es usado para referirse al contenido del tipo de MIME text/html o todavía más ampliamente como un término genérico para el HTML, ya sea en forma de XML (como XHTML 1.0 y posteriores) o en forma directamente de SGML (como HTML 4.01 y anteriores).

Por convención, los archivos de formato HTML usan la extensión .htm o .html.

Elementos.

Los elementos son la estructura básica de HTML. Los elementos tienen dos propiedades básicas: atributos y contenido. Cada atributo y contenido tiene ciertas restricciones para que se considere válido al documento HTML. Un elemento generalmente tiene una etiqueta de inicio por ejemplo (<nombre-de-elemento>) y una etiqueta de cierre (</nombre-de-elemento>). Los atributos del elemento están contenidos en la etiqueta de inicio y el contenido está ubicado entre las dos etiquetas por ejemplo:

(<nombre-de-elemento atributo="valor">Contenido</nombre-de-elemento>).

A continuación se listan varios tipos de elementos de marcado usados en HTML.

El marcado estructural. Describe el propósito del texto. Por ejemplo, <h2>Golf</h2> establece a "Golf" como un encabezamiento de segundo nivel, el cual se mostraría en un navegador de una manera similar al título "Marcado HTML" al principio de esta sección. El marcado estructural no define cómo se verá el elemento, pero la mayoría de los navegadores Web han estandarizado el formato de los elementos. Un formato específico puede ser aplicado al texto por medio de hojas de estilo en cascada.

El marcado de presentación. Describe la apariencia del texto, sin importar su función. Por ejemplo, <b>negrita</b> indica que los navegadores Web visuales

deben mostrar el texto en negrita, pero no indica qué deben hacer los navegadores web que muestran el contenido de otra manera.

El marcado hipertextual. Se utiliza para enlazar partes del documento con otros documentos o con otras partes del mismo documento. Para crear un enlace es necesario utilizar la etiqueta de ancla <a> junto con el atributo href, que establecerá la dirección URL a la que apunta el enlace.

Atributos.

La mayoría de los atributos de un elemento son pares nombre-valor, separados por un signo de igual "=" y escritos en la etiqueta de comienzo de un elemento, después del nombre de éste. El valor puede estar rodeado por comillas dobles o simples, aunque ciertos tipos de valores pueden estar sin comillas en HTML (pero no en XHTML). De todas maneras, dejar los valores sin comillas es considerado poco seguro. En contraste con los pares nombre-elemento, hay algunos atributos que afectan al elemento simplemente por su presencia.

Frames o marcos.

Los frames, nos permiten partir la ventana del navegador en diferentes áreas. Cada una de estas áreas es independiente y han de ser codificadas con archivos HTML también independientes. Como resultado, cada frame o marco contiene las propiedades específicas que le indiquemos en el código HTML a presentar en ese espacio. Así mismo, y dado que cada marco es independiente, tendrán sus propias barras de desplazamiento, horizontales y verticales, por separado.

Existen en la Web muchas páginas que contienen frames y seguro que todos hemos tenido la ocasión de conocer algunas. Se suelen utilizar para colocar en una parte de la ventana una barra de navegación, que generalmente se encuentra fija y permite el acceso a cualquier zona de la página Web. Una de las principales ventajas de la programación con frames viene derivada de la independencia de los distintos frames, pues podemos navegar por los

contenidos de nuestro Web con la barra de navegación siempre visible, y sin que se tenga que recargar en cada una de las páginas que vamos visitando.

Ventajas de los frames.

- La navegación de la página será más rápida. Aunque la primera carga de la página sería igual, en sucesivas impresiones de páginas ya tendremos algunos marcos guardados, que no tendrían que volverse a descargar.
- Crear páginas del sitio sería más rápido. Como no tenemos que incluir partes de código como la barra de navegación, título, etc. crear nuevas páginas sería un proceso mucho más rápido.
- Partes de la página (como la barra de navegación) se mantienen fijas y eso puede ser bueno, para que el usuario no las pierda nunca de vista.
- Estas mismas partes visibles constantemente, si contienen enlaces, pueden servir muy bien para mejorar la navegación por el sitio.
- Mantienen una identidad del sitio donde se navega, pues los elementos fijos conservan la imagen siempre visible.

Desventajas de los frames.

- QUITAN espacio en la pantalla. El espacio ocupado por los frames fijos se pierde a la hora de hacer páginas nuevas, porque ya está utilizado. En definiciones de pantalla pequeña o dispositivos como Palms, este problema se hace más patente.
- Fuerzan al visitante a entrar por la declaración de frames. Si no lo hacen así, sólo se vería una página interior sin los recuadros. Estos recuadros podrían ser insuficientes para una buena navegación por los contenidos y podrían no conservar una buena imagen corporativa.
- La promoción de la página sería, en principio, más limitada. Esto es debido a que sólo se debería promocionar la portada, pues si se promocionan páginas interiores, podría darse el caso de que los visitantes entrasen por ellas en lugar de por la portada, creándose el problema descrito en el punto anterior.

- A mucha gente le disgustan pues no se sienten libres en la navegación, pues entienden que esas partes fijas están limitando su movilidad por la Web. Este efecto se hace más patente si la página con frames tiene enlaces a otras páginas Web fuera del sitio y, al pulsar un enlace, se muestra la página nueva con los marcos de la página que tiene frames.
- Algunos navegadores no los soportan. Esto no es muy habitual, pero si estamos haciendo una página que queramos que sea totalmente accesible deberíamos considerarlo importante.
- Los favoritos no funcionan correctamente en muchos casos. Si queremos incluir un favorito a una página de un frame que no sea la portada podemos encontrar problemas.
- Puede que el botón de atrás del navegador no se comporte como deseamos.
- Si se quiere actualizar más de un frame con la pulsación de un enlace se deberá utilizar JavaScript. Además los scripts se pueden complicar bastante cuando se tienen que comunicar varios frames entre sí.

#### Códigos HTML básicos.

- `<html>`: define el inicio del documento HTML, le indica al navegador que lo que viene a continuación debe ser interpretado como código HTML.
- `<script>`: incrusta un script en una web, o se llama a uno mediante `src="uri del script"` Se recomienda incluir el tipo MIME en el atributo `type`, en el caso de JavaScript `text/javascript`.
- `<head>`: define la cabecera del documento HTML, esta cabecera suele contener información sobre el documento que no se muestra directamente al usuario. Como por ejemplo el título de la ventana del navegador. Dentro de la cabecera `<head>` podemos encontrar: vea figura 2.7.1

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML
2 <html>
3   <head>
4     <title>Example</title>
5     <link href="screen.css" rel="sty
6   </head>
7   <body>
8     <h1>
9     <a href="/">Header</a>
10    </h1>
11    <ul id="nav">
12      <li>
13        <a href="one/">One</a>
14      </li>
15      <li>
16        <a href="two/">Two</a>
17      </li>

```

Figura 2.7.1 Ejemplo de código HTML con coloreado de sintaxis.

- <title>: define el título de la página. Por lo general, el título aparece en la barra de título encima de la ventana
- <link>: para vincular el sitio a hojas de estilo o iconos Por ejemplo:<link rel="stylesheet" href="/style.css" type="text/css">
- <style>: para colocar el estilo interno de la página; ya sea usando [CSS](#), u otros lenguajes similares. No es necesario colocarlo si se va a vincular a un archivo externo usando la etiqueta <link>
- <meta>: para metadatos como la autoría o la licencia, incluso para indicar parámetros http (mediante http-equiv="") cuando no se pueden modificar por no estar disponible la configuración o por dificultades con server-side scripting.
- <body>: define el contenido principal o cuerpo del documento. Esta es la parte del documento html que se muestra en el navegador; dentro de esta etiqueta pueden definirse propiedades comunes a toda la página, como color de fondo y márgenes. Dentro del cuerpo <body> podemos encontrar numerosas etiquetas.

A continuación se indican algunas a modo de ejemplo:

- <h1> a <h6>: encabezados o títulos del documento con diferente relevancia.
- <table>: define una tabla
- <tr>: fila de una tabla

- `<td>`: columna de de una tabla
- `<a>`: Hipervínculo o enlace, dentro o fuera del sitio web. Debe definirse el parámetro de pasada por medio del atributo *href*. Por ejemplo: `<a href="http://www.wikipedia.org">Wikipedia</a>` se representa como Wikipedia)
- `<div>`: división de la página. Se recomienda, junto con *css*, en vez de `<table>` cuando se desea alinear contenido
- `<img>`: imagen. Requiere del atributo *src*, que indica la ruta en la que se encuentra la imagen. Por ejemplo: ``. Es conveniente, por accesibilidad, poner un atributo *alt*="texto alternativo".
- `<li><ol><ul>`: Etiquetas para listas.
- `<b>`: texto en negrita (*Etiqueta desaprobada. Se recomienda usar la etiqueta <strong>*)
- `<i>`: texto en cursiva (*Etiqueta desaprobada. Se recomienda usar la etiqueta <em>*)
- `<s>`: texto tachado (*Etiqueta desaprobada. Se recomienda usar la etiqueta <del>*)
- `<u>`: texto subrayado

La mayoría de etiquetas deben cerrarse como se abren, pero con una barra ("/") tal como se muestra en los siguientes ejemplos:

- `<table><tr><td>Contenido de una celda<\/"td"><\/"tr"><\/"table">`
- `<script>Código de un script integrado en la página<\/script>`.

Ventajas de HTML:

- Es el lenguaje de formateo para los navegadores Web.
- Es fácil de entender y utilizar.
- Su uso es muy extendido.

Desventajas de HTML:

- No tiene semántica. Uso de etiquetas con nombres diferentes.
- El contenido no puede ser reconocido ni procesado por programas.
- Tiene un costoso mantenimiento de las páginas.

## MARCO TEÓRICO

- No tiene estándares comunes.
- Solo tiene hiperenlaces simples (XML puede tener de 1 a n enlaces).  
Xlink.

En resumen, el HTML es un lenguaje muy fácil de comprender y muy utilizado para la presentación de la información, pero esta no se puede procesar ni almacenar, ya que no permite su manipulación por un programa debido a su anarquía.

### JavaScript.

JavaScript es muy fácil de aprender para quien ya conoce lenguajes similares como el C++ o Java, pero, dada su simplicidad sintáctica y su manejabilidad, no es tampoco difícil para quien se acerca por primera vez a este lenguaje. Sin embargo, esto puede ser un arma de doble filo porque la simplicidad se basa en una disponibilidad de objetos limitada, por lo que algunos procedimientos, aparentemente muy sencillos, requieren *scripts* bastantes complejos.

La característica principal de JavaScript, de hecho, es la de ser un *lenguaje de scripting*, pero, sobre todo, la de ser el lenguaje de scripting por excelencia y, sin lugar a dudas, el más usado. Esta particularidad conlleva una notable serie de ventajas y desventajas según el uso que se le deba dar y teniendo en cuenta la relación que se establece entre el mecanismo cliente-servidor. Para explicar con pocas palabras dicha relación, podemos decir que el servidor envía los datos al cliente y estos datos pueden llegar en dos formatos diversos: en formato texto (o ASCII) o en formato binario (o código máquina: El cliente sabe cómo comprender sólo el formato binario (es decir, la secuencia de 1 y 0), por lo que si los datos llegan en este formato son inmediatamente ejecutables (y, por desgracia, no dejan abierta la posibilidad de efectuar controles), mientras que si el formato es diverso tienen que ser interpretados y traducidos al formato binario y, por tanto, el cliente necesitará un filtro, o mejor dicho, un intérprete que sepa leer estos datos y los pueda traducir al binario. Los datos en formato texto son visibles al usuario como simples combinaciones de caracteres y de palabras y son, por tanto, fáciles de manipular, si bien



requieran más tiempo para su interpretación a causa de los distintos pasos y de las transformaciones a las que deben someterse para que el cliente pueda comprenderlos: Los datos en formato binario, sin embargo, son difíciles de comprender por el usuario, pero inmediatamente ejecutables por el cliente ya que no requieren fases intermedias.

Es un lenguaje de programación compacto y orientado al objeto destinado al desarrollo de aplicaciones Internet que actúa a modo de complemento del HTML. El primer navegador que soportó JavaScript fue Netscape Navigator 2.0. El código de las aplicaciones Java Script se integra en el mismo archivo con el texto en formato HTML y es interpretado por el cliente (navegador). Actualmente los únicos navegadores que soportan JavaScript son Netscape Navigator(a partir de la versión 2.0) y Microsoft Internet Explorer (a partir de la versión 3.0, denominándolo Jscript).

También afectan a JavaScript las incompatibilidades: los navegadores 4.0 soportan nuevas especificaciones de JavaScript que no soportan los navegadores anteriores. Además el mismo código JavaScript en Navigator no se comporta de manera idéntica en Internet Explorer.

Sintaxis: JavaScript hereda principalmente de Java su sintaxis y estructura, aunque el estilo de programación puede diferir bastante.

JavaScript está destinado para la implementación de programas pequeños, es ideal para tareas repetitivas y de control de eventos. Por ejemplo, se puede utilizar para calcular el valor de una variable dependiente del valor de un campo de formulario. Cada vez que cambie el valor de dicho campo, el programa JavaScript encargado de controlar dicho evento se invoca y se calcula el nuevo valor de la variable.

Lenguaje Interpretado.

JavaScript es un lenguaje interpretado. Los archivos con el código se ejecutan directamente.

Los lenguajes interpretados ofrecen varias ventajas; entre ellas que son más sencillos y fáciles de aprender. Además, normalmente es más fácil desarrollar,

modificar y depurar programas para lenguajes interpretados porque no es necesario recompilar el código cada vez que se realiza un cambio.

Los programas para lenguajes interpretados suelen requerir un mayor tiempo de ejecución que los programas compilados. En el caso de JavaScript esto no supone un problema, ya que, con los anchos de bandas actuales, la velocidad de ejecución es despreciable frente a la velocidad de transmisión. Los scripts escritos en JavaScript se compilan en formato Bytecode (como los programas Java) y evalúan mientras se está transmitiendo el script.

Lenguaje no extensible.

JavaScript en comparación con Java, no es totalmente extensible. El modelo de JavaScript está formado por un número limitado de objetos básicos, propiedades, métodos y tipos de datos, que son suficientes para desarrollar aplicaciones cliente-servidor.

### **Ventajas de JavaScript.**

- JavaScript no requiere un tiempo de compilación, los scripts se pueden desarrollar en un periodo de tiempo relativamente corto. A esto podemos añadirle las características de interfaz como, por ejemplo, cuadro de diálogo, formularios y otros elementos GUI (Interfaz Gráfico de Usuario), son gestionados por el navegador y por el código HTML. Por lo tanto los programadores que utilizan JavaScript no se deben preocupar en crear o controlar dichos elementos en sus aplicaciones.
- Aunque JavaScript tiene muchas similitudes con Java, no incluye la sintaxis y reglas complejas de Java.
- Como WWW es independiente de la plataforma hardware o sistema operativo, los programas escritos en JavaScript también lo son, siempre y cuando exista un navegador con soporte JavaScript para la plataforma en cuestión.
- Los programas JavaScript tienden a ser pequeños y compactos no requieren mucha memoria ni tiempo adicional de transmisión. Además,

al incluirse dentro de las mismas páginas HTML se reduce el número de accesos independientes a la red.

- El lenguaje de scripting es seguro y fiable porque es claro y hay que interpretarlo, por lo que puede ser filtrado; para el mismo JavaScript, la seguridad es casi total.
- El código JavaScript se ejecuta en el cliente por lo que el servidor no es solicitado más de lo debido; un script ejecutado en el servidor, sin embargo, sometería a éste a dura prueba y los servidores de capacidades más limitadas podrían resentir de una continua solicitud por un mayor número de usuarios.

### **Desventajas de JavaScript.**

- En las primeras versiones de Netscape Navigator con soporte JavaScript el número de métodos integrados era extremadamente reducido. En la versión actual, Navigator 4.0, se ha incrementado significativamente el número de métodos, pero sigue siendo insuficiente para gestionar documentos y ventanas.
- El código de JavaScript debe incluirse dentro del mismo documento HTML, y no es posible ocultar el código fuente y evitar la copia y reutilización de éste.
- Los script tienen capacidades limitadas, por razones de seguridad, por lo cual no es posible hacer todo con JavaScript, sino que es necesario usarlo conjuntamente con otros lenguajes evolucionados, posiblemente más seguros, como Java. Dicha limitación es aún más evidente si queremos operar en el hardware del ordenador, como, por ejemplo, la fijación en automático de la resolución vídeo o la impresión de un documento.
- Un problema importante es que el código es visible y puede ser leído por cualquiera, incluso si está protegido con las leyes del copyright. Esto, que en mi opinión es una ventaja, representa el precio que tiene que pagar quien quiere utilizar el Web: la cuestión de los derechos de autor ha asistido a una verdadera revolución con la llegada de Internet

(citamos, como ejemplo más representativo, el MP3). La tutela que proporcionan las leyes actuales resulta débil e inadecuada, por lo que la única solución es tomarse las cosas con filosofía.

- El código del script debe descargarse completamente antes de poderse ejecutar y ésta es la otra cara de la moneda de lo que hemos dicho anteriormente: si los datos que un script utiliza son muchos (por ejemplo, una recopilación de citas que se mostrara de manera casual), el tiempo que tardará en descargarse será muy largo, mientras que la interrogación de la misma base de datos en el servidor sería más rápida.

## 2.8 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE APLICACIONES CLIENTE / SERVIDOR

Las aplicaciones cliente/servidor basan su lógica de comunicación en la arquitectura o modelo cliente/servidor, éste modelo es el procesamiento cooperativo de la información por medio de un conjunto de componentes, en el cual múltiples clientes, distribuidos geográficamente, solicitan requerimientos a uno o más servidores centrales. Vea Fig. 2.8.1

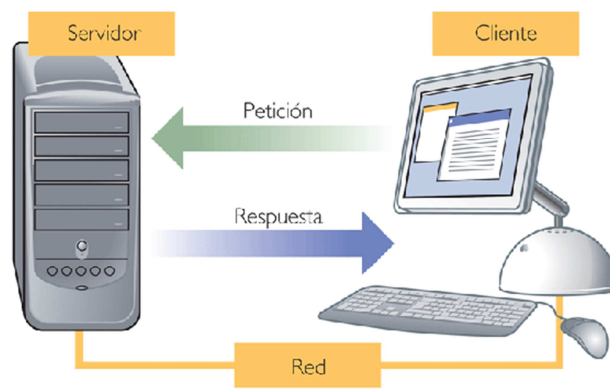


Fig. 2.8.1 Modelo Cliente/Servidor

Éste modelo proporciona al usuario final un acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso. Los usuarios invocan la parte del cliente de la aplicación, ésta construye una solicitud para ese servicio y realiza un envío por el canal de comunicación

hacia el servidor, el servidor recibe la solicitud de servicio, procesa la información y devuelve los resultados en forma de respuesta, generalmente un servidor puede tratar múltiples peticiones (múltiples clientes) al mismo tiempo (ver figura 2.8.1). Para que exista comunicación entre los clientes y los servidores debe existir un canal de comunicación que proporcione los mecanismos básicos de transporte y direccionamiento.

### **Características del modelo Cliente/Servidor**

- Se establece una relación entre procesos distintos, los cuales pueden ser ejecutados en la misma máquina o en máquinas distribuidas a lo largo de la red.
- La relación establecida puede ser de muchos a uno, en la que un servidor puede dar servicio a muchos clientes, regulando su acceso a recursos compartidos.
- Un sistema de servidores realiza múltiples funciones al mismo tiempo, esto representa una imagen de un solo sistema en los clientes.
- Las plataformas de software y hardware entre clientes y servidores son independientes. Esto es una de las grandes ventajas de la arquitectura ya que se puede contar con clientes y servidores de forma independiente a su plataforma
- El concepto de escalabilidad tanto horizontal como vertical es aplicable a cualquier sistema cliente/servidor. La escalabilidad horizontal permite agregar más componentes activos al sistema sin afectar de manera significativa el rendimiento de éste. La escalabilidad vertical permite mejorar las características del servidor o agregar múltiples servidores.

El modelo cliente/servidor es un modelo basado en la idea del servicio, en el que el cliente es un proceso que consume servicios y el servidor un proceso proveedor de los mismos.

## **Cliente**

Es un proceso que permite al usuario formular los requerimientos hacia el servidor, se le conoce con el término front-end. Este normalmente maneja todas las funciones relacionadas con la manipulación y despliegue de información, por lo que se encuentra desarrollado sobre plataformas que permiten construir interfaces gráficas de usuarios, así como la capacidad de acceder a los servicios distribuidos en cualquier parte de la red.

Sus funciones pueden resumirse en:

- Administración de la interfaz de usuario
- Interacción con el usuario
- Procesamiento de la lógica de la aplicación, así como validaciones locales
- Generación de requerimientos a las bases de datos
- Recepción de resultados del servidor
- Presentación de resultados

## **Servidor**

Es el proceso encargado de atender a múltiples clientes que realizan peticiones de algún recurso administrado por él. A este proceso se le conoce con el término de back-end. El servidor generalmente maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de los datos.

Las funciones del servidor pueden resumirse en:

- Aceptar los requerimientos de bases de datos que realizan los clientes
- Procesar requerimientos de bases de datos
- Formatear datos para transmitirlos a los clientes

- Procesar la lógica de la aplicación y realizar validaciones en la base de datos middleware.

Podemos definir middleware como la interfaz que provee la conectividad entre aplicaciones cliente/servidor y entre aplicaciones y bases de datos. Es una capa de software que permite a los desarrolladores omitir el manejo de detalles de bajo nivel de diversos protocolos de comunicación, sistemas operativos y arquitecturas de bases de datos. En esta capa pueden encontrarse; API's, mensajes de red, accesos a bases de datos entre otras.

### **Clasificación de modelos Cliente/Servidor**

Se tiene una primera clasificación de acuerdo a los conceptos de Cliente/Servidor ligero y pesado. En esta se toman en cuenta los grados de libertad que brinda el modelo cliente/servidor para balancear los procesos entre los niveles de presentación, aplicación y base de datos.

#### **Cliente Pesado/Servidor Ligero**

- La mayor parte de la aplicación corre en el lado del cliente
- El servidor exporta los datos en bruto (sin formato)
- Los clientes saben la organización de los datos en el servidor

En esta podemos encontrar a los servidores de archivos

#### **Cliente Ligero/Servidor Pesado**

- La mayor parte de la aplicación corre del lado del servidor
- El servidor exporta métodos que operan sobre los datos
- El cliente no consume mayores recursos que la interfaz gráfica

En esta última clasificación podemos encontrar al servidor Web, los servidores de transacciones, servidores de correo, etc. Vea Fig. 2.8.2

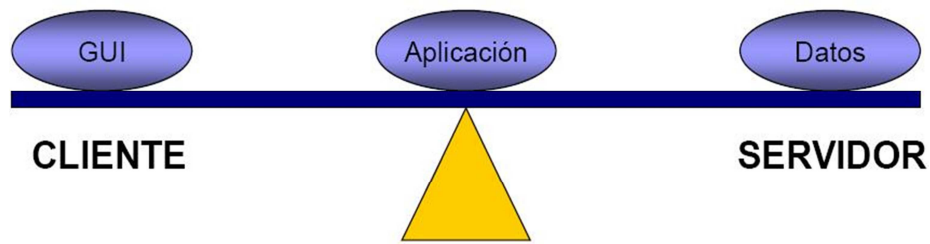


Fig. 2.8.2 Cliente Ligero/Servidor Pesado

En una segunda clasificación podemos hablar de capas o planos (tier). La cual es una clasificación orientada a diferenciar la manera en que las prestaciones funcionales de la aplicación serán asignadas, así como su proporción, esto para ambas entidades cliente y servidor.

Dichas prestaciones funcionales pueden agruparse en tres componentes; interfaz de usuario, lógica de negocios y datos, cada uno de los cuales corresponden a una capa. De esta forma tenemos aplicaciones en una sola capa, dos capas (two-tier), tres capas (three-tier) y multi capa (multi-tier).

### **Aplicación Cliente/Servidor una capa**

Las aplicaciones cliente/servidor de una capa agrupan toda la lógica en una sola entidad, (lógica de presentación, lógica de aplicación y fuente de datos), la aplicación se comporta como cliente y servidor simultáneamente.

### **Aplicación Cliente/Servidor dos capas**

Generalmente las aplicaciones Cliente/Servidor están basadas en modelos de dos capas. El modelo de dos capas consta de un cliente, que mantiene la lógica de la aplicación y un servidor que administra las peticiones a la base de datos. Éste tipo de aplicaciones son generalmente utilizadas cuando se requiere poco procesamiento de datos, además de que la organización tiene una base de datos centralizada en un solo servidor.



### Aplicación Cliente/Servidor tres capas

Las aplicaciones en tres capas es un modelo que evolucionó del modelo de dos capas, para éste modelo se agregó una capa intermedia entre el cliente y el servidor de base de datos. La capa que se agregó consiste en un servidor de aplicaciones que contiene la lógica de la aplicación (lógica de negocios). Generalmente éste tipo de aplicaciones se utiliza cuando se requiere mucho procesamiento de datos en la aplicación, además de que los procesos no se encuentren relativamente relacionados con los datos, a su vez se requiera aislar la tecnología de base de datos para que sea fácil realizar cambios. Vea Fig. 2.8.3

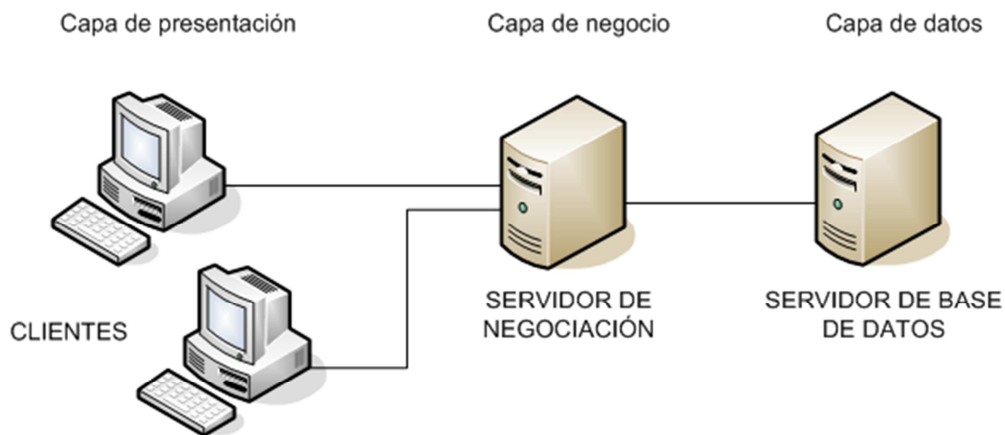


Fig. 2.8.3 Aplicación Cliente/Servidor de tres capas

### Aplicación Cliente/Servidor multicapa

En las aplicaciones Cliente/Servidor multicapa, la lógica de presentación, la lógica de la aplicación y el servidor de base de datos, además de estar separados, pueden ser subdivididos aún más. Éste modelo ha predominado en el desarrollo de aplicaciones multiplataforma.

### Ventajas de aplicaciones Cliente/Servidor

- Los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado

no pueda dañar el sistema. Esta centralización también facilita la tarea de la actualización de información.

- Escalabilidad, se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).
- El desarrollo del software puede ser realizado de manera paralela, lo que implica menos tiempo de desarrollo.
- Mejora el rendimiento. Los datos son almacenados y usados donde son generados, lo cual permite distribuir la complejidad del sistema en los diferentes sitios de la red, optimizando la labor de cada uno de ellos.
- Mantenimiento y soporte sencillo.
- Rendimiento, el tráfico en los canales de comunicación se reduce. Los clientes únicamente se comunican con otros módulos lo estrictamente necesario, obtiene los datos solicitados y cierra la conexión, por lo que deja la red libre para realizar más conexiones.

### **Desventajas de aplicaciones Cliente/Servidor**

- Costo elevado. El costo de la administración es elevado debido a la complejidad técnica del servidor.
- El servidor es el eslabón débil. El servidor es el único eslabón débil en las aplicaciones cliente/servidor, debido a que toda la red está construida en torno a él.
- Manejo óptimo de errores. Se deben de tener estrategias en el manejo de errores para mantener la consistencia de los datos y de las aplicaciones.
- Las distancias en el canal de comunicación puede contribuir a la velocidad de respuesta de la información.

### 3.1 CONTEXTO SITUACIONAL DEL PROBLEMA

En esta parte se describirá el proceso general que se lleva a cabo para la obtención de insumos que se encuentran alojados en el almacén, además del proceso que se necesita para obtener insumos externos, es decir mediante compras; ambos procesos se llevan a cabo mediante un vale o ficha que se hará entrega posterior al encargado del almacén y/o de compras según sea el caso.

A continuación se lleva a cabo la descripción de la obtención de insumos en el almacén, mediante el diagrama general del proceso Solicitud hacia el Almacén, que se muestra en la figura 3.1.1, el usuario realiza una petición de insumos al almacén y espera una resolución satisfactoria que cubra sus necesidades.

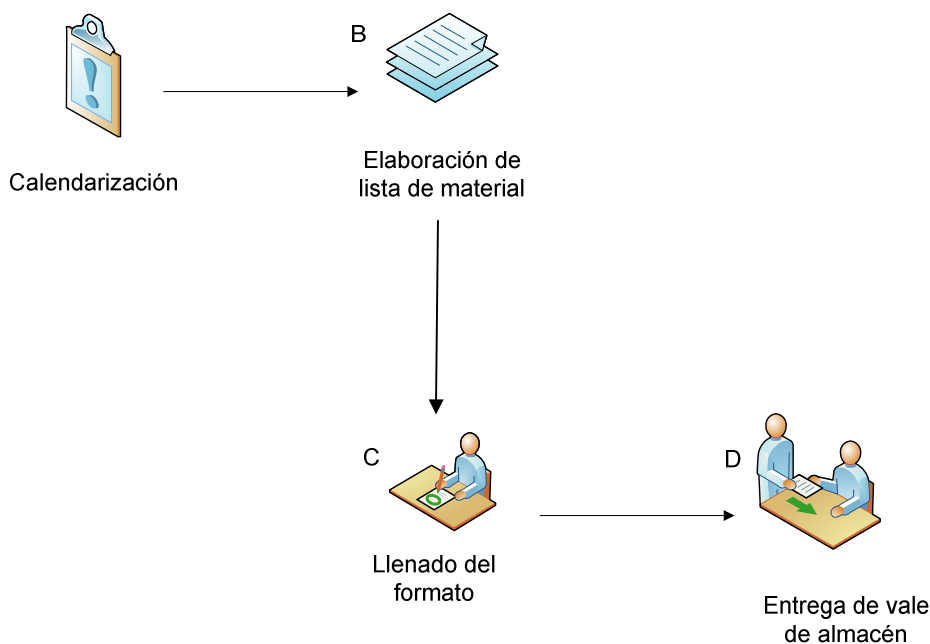


Figura 3.1.1 Diagrama de proceso de Solicitud hacia el Almacén

- A. El proceso inicia mediante una calendarización de peticiones, esto es que, en determinada fecha se pueden hacer pedidos al almacén, no está disponible en todo momento.

Durante este proceso suelen presentarse inconvenientes tales como:

- B. Existe una única fecha en que se puede realizar dichos pedidos, el usuario por tanto elabora una lista de material que necesita.
- C. Se procede entonces a llenar un formato el cual está elaborado en hoja de cálculo con los requerimientos necesarios con un máximo de hasta diecisiete artículos. En caso de necesitar más del total de insumos por vale se procede a llenar tantos vales como sean necesarios, aunque generalmente esto último no sucede.
- D. El proceso finaliza cuando teniendo la solicitud elaborada se procede a entregar el vale de almacén al encargado de reunir dichos vales para que se entreguen al almacén de manera conjunta. Antes de llevar los vales se pregunta por última vez a los usuarios si es todo lo que van a necesitar, pues ya no hay segunda vuelta.

En el diagrama general de proceso de Entrega de insumos de Almacén, que se muestra en la figura 3.1.2, el usuario recibe el material pedido.

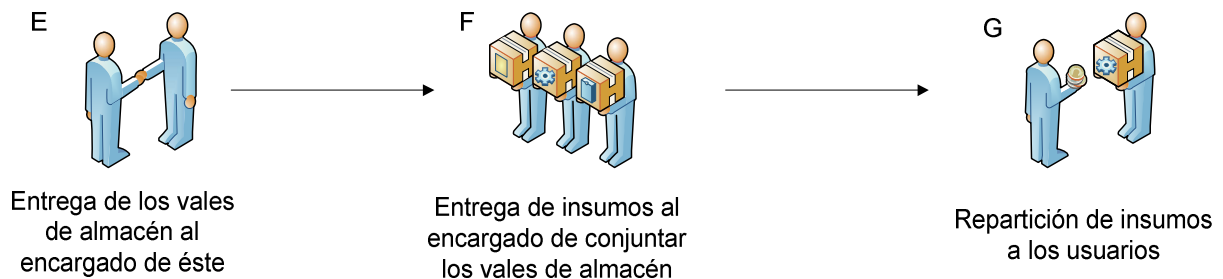


Figura 3.1.2 Diagrama de proceso de Entrega hacia el Usuario

- E. El proceso inicia cuando se hace entrega de los vales de almacén al encargado del mismo, quien al cabo de un par de semanas tendrá que abastecer las peticiones hechas por parte de los usuarios de la División de Ingenierías Civil y Geomática.

Durante este proceso suelen presentarse dificultades tales como:

- F. Posteriormente se hace entrega al encargado de recibir los vales de almacén aquellos insumos que fueron obtenidos del almacén, ya que no se sabe a ciencia cierta si todo lo que se pidió será entregado pues no se tiene un conteo del material en stock.
- G. El proceso finaliza cuando se procede con la repartición de los insumos a cada usuario indicándoles qué es lo que pudo faltar en su pedido si ese fuese el caso.

A continuación se lleva a cabo la descripción de la obtención de insumos externos (vales de compra) mediante el diagrama general del proceso Solicitud de Compras, que se muestra en la figura 3.1.3, el usuario realiza una petición de insumos para comprar y espera una resolución satisfactoria que cubra sus necesidades.

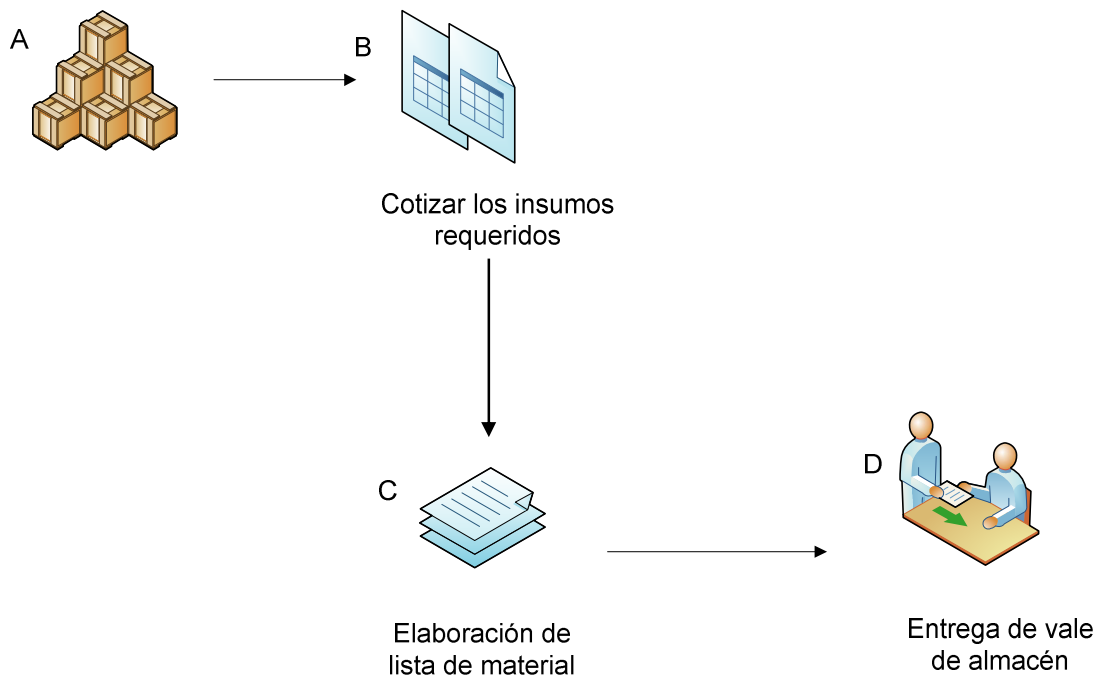


Figura 3.1.3 Diagrama de proceso de Entrega hacia el Usuario

## ANÁLISIS DE PLANTEAMIENTO DEL PROBLEMA

- A. El proceso inicia cuando el usuario sabe que algunos productos que requiere de forma imperativa no están o mejor dicho no los maneja el almacén de insumos.
- B. Sabiendo lo que se va a necesitar el usuario procede a elaborar una cotización del o los material(es) requerido(s), asimismo indicando en el vale el o los lugar(es) de dónde se obtienen dichos precios.
- C. Se procede entonces a llenar un formato el cual está elaborado en hoja de cálculo con los insumos que se desean adquirir con un máximo de hasta quince artículos. En caso de necesitar más del total de insumos por vale se procede a llenar tantos vales como sean necesarios, aunque generalmente esto último no sucede.
- D. El proceso finaliza cuando con la solicitud elaborada se procede a entregar el vale de compras al encargado de reunirlos, que es el mismo que se encarga de reunir los vales para almacén. Sabiendo que aquí no hay fecha calendarizada para éstos no es necesario reunir cierta cantidad de vales de compra.

En el diagrama general de proceso de Entrega de insumos de Compra, que se muestra en la figura 3.1.4, el usuario recibe el material pedido externo.

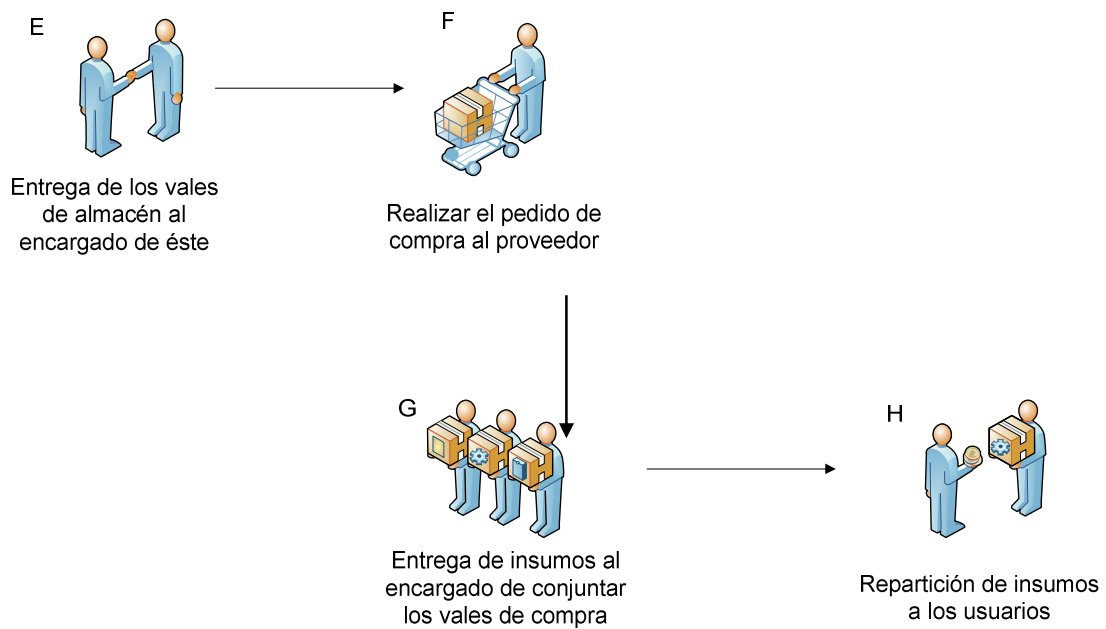


Figura 3.1.4 Diagrama de proceso de Entrega hacia el Usuario

- E. El proceso inicia cuando se hace entrega del o los vale(s) de compra al encargado de ese departamento.
- F. Éste último hará dicho pedido al proveedor que se indicó en la cotización previa.

Durante este proceso suelen presentarse dificultades tales como:

- G. El tiempo de entrega de estos insumos será función del proveedor externo, aunando el tiempo en que se hace notificación al encargado del envío y recibo de vales el cual labora en la División de Ingenierías Civil y Geomática para que se le haga entrega a éste.
- H. El proceso finaliza cuando se notifica y hace entrega de los insumos de compra requeridos al usuario que hizo la petición.

### **3.2 REQUERIMIENTOS GENERALES Y PARTICULARES DE LA INFORMACIÓN**

#### **Requerimientos generales**

- Para la implementación del sistema generador de vales se pretende que este funcione en un equipo de cómputo central (servidor) al que se puedan conectar los jefes de la división y el jefe del almacén para realizar sus consultas.
- En todas las áreas se debe tener facilidad de acceso a este sistema.
- El sistema deberá ser desarrollado con herramientas de distribución gratuita, con la más alta tecnología en cuanto a manejo y presentación de la información.
- El sistema tendrá accesos a través de cualquier navegador comercial, donde el usuario pueda acceder desde cualquier ubicación.

## ANÁLISIS DE PLANTEAMIENTO DEL PROBLEMA

- El método de autenticación de acceso al sistema será mediante usuario y contraseña.
- Los vales serán generados como documentos PDF.
- La base de datos estará localizada en el servidor central.
- Esta base de datos solo podrá ser administrada por el personal autorizado.
- La información que contenga debe ser verdadera, estar al día y debe ser necesaria para realizar los procesos que dependen de ella.
- Debe ser segura.
- Mediante las consultas se podrá acceder a la información sobre los materiales que se manejan en el almacén.
- Es de vital importancia que el sistema cuente con catálogos para que los usuarios solo tengan que seleccionar el material evitando errores de dedo.
- La interfaz permitirá al usuario elegir el tipo de vale que desea generar ya sea de compra de material o de solicitud.
- Las ventanas de salida permitirán mostrar resultados de las transacciones realizadas.

### **Requerimientos Particulares**

- Los requerimientos particulares del sistema son:
- Debe ser un sistema robusto, capaz de gestionar diferentes módulos, datos, usuarios y tipo de servicios que se manejan.



## REQUERIMIENTOS GENERALES Y PARTICULARES DE LA INFORMACIÓN

- Que posea una base de datos de usuarios y materiales.
- Debe tener la capacidad de soportar incrementos en su base de datos.
- Que genere en automático la papelería que se requiera.
- Los jefes de la división y el personal de almacén podrán consultar el sistema.
- Debe permitir la implementación de parámetros de gestión de servicios según materiales.
- Debe permitir seleccionar los materiales que se desean adquirir.
- Debe permitir realizar en tiempo real y de manera interactiva, procesos de creación actualización y accesos a todos los servicios del sistema
- Debe incluir herramientas para la generación de reportes orientados a usuarios finales.
- Debe disponer de un lenguaje estándar de consulta de bases de datos (MySQL), que permita realizar consultas independientemente de la interface de consulta de usuarios finales.
- Debe tener permisos de acceso y seguridad que regulen y gestionen el acceso a datos, procesos y funciones según el material y el tipo de operación que se lleve en el almacén.
- Debe realizar operaciones de ingreso, almacenamiento, edición, salida y visualización de datos.
- Debe garantizar condiciones de integridad de datos en procesos concurrentes en el contexto del sistema (edición simultánea de vales).
- Debe tener método de autenticación única para cada usuario a través de un nombre de usuario y contraseña válidos para el sistema.

## ANÁLISIS DE PLANTEAMIENTO DEL PROBLEMA

- Debe permitir una navegación simple y lógica.
- Debe permitir configurar valores por defecto y preferencias según el tipo de usuario.

### **Requerimientos del administrador**

Para la lista de usuarios

- Dar de alta un usuario
  - Nombre de Usuario
  - RFC
  - Área
- Modificar o borrar los datos de algún usuario según su cargo.
- Acceso a una lista de módulos con sus propiedades.
  - Se podrá acceder a una lista de materiales con opción a seleccionar

Dadas las especificaciones dadas anteriormente el sistema contará con lo siguiente:

### **Interfaz de Usuario**

Dado que es el medio con que el usuario puede comunicarse con una máquina, esta proveerá:

- Puesta en marcha y apagado.
- Control de las funciones manipulables del equipo.
- Comunicación con otros sistemas.
- Información de estado.
- Configuración de la propia interfaz y entorno.
- Control de acceso.
- Sistema intuitivo

### **Presentación de la información**

- Debe permitir gestionar administrar las configuraciones de los vales tales como: materiales, número de artículos, área solicitante, presupuestos y proveedores.
- Debe permitir establecer configuraciones de las interfaces en términos de:
  - Selección de campos
  - Orden de los materiales
- Debe soportar las operaciones de selección de materiales
- Debe soportar los medios de envío y distribución digital.
- Debe permitir al usuario especificar los elementos de imprimir y guardar vale.
- Debe permitir a los usuarios seleccionar y manipular las opciones para su impresión.

### **Circulación**

Se encargará de establecer los parámetros de control y gestión de transacciones, solicitud o compra segundos patrones basados en tipos de transacciones y tipos de usuarios.

### **Adquisición y control de Existencias**

Gestionará los procesos de compras y solicitud de material a través de identificadores únicos (número de folio, número de pedido, nombre de proveedor) persistentes e invocables a través de las opciones de sistema:

## ANÁLISIS DE PLANTEAMIENTO DEL PROBLEMA

- Solicitud de Compra
- Vale de salida de almacén

### **Seguridad, autenticación y autorización**

El sistema será capaz de realizar lo siguiente:

- Soportar la gestión de acceso y seguridad según niveles y funciones específicas.
- Deberá soportar el acceso consistente a todos los espacios, servicios y funciones a través de un único punto de autenticación (login).
- Tendrá la posibilidad de establecer permisos de accesos específicos según los perfiles de usuarios, ya sea de solicitud o compra de material.
- Permitirá al almacén configurar accesos a funciones y base de datos.
- Soportará modelos de control basados en la combinación de roles y atributos (un único usuario puede tener múltiples roles sin necesidad de múltiples identificadores).

### **Proceso de instalación**

- Deberá contemplar un proceso de instalación y parametrización hasta lograr condiciones de implementación completa del sistema.
- Se describirán los conocimientos y habilidades técnicas que se requieren para instalar e implementar el sistema.
- Se describirá un modelo de responsabilidades sugeridas para el proveedor y el almacén en el proceso de instalación e implementación.

### **Presentación de contenidos**

- Permitirá gestionar y parametrizar aspectos relativos a la disposición (layout) apariencia (look and feel), encabezamiento HTML (etiquetas meta del header HTML).

### **Impresión y descarga de datos**

- Debe soportar y proveer medios de envío de registros.
- Debe permitir la visualización de todos los campos número materiales usuario y tipo de vale, para ser impreso o almacenado.
- Debe permitir al usuario seleccionar el formato, para imprimir o guardar un registro para su control personal.

### **Gestión de Materiales**

- Debe permitir la identificar a los usuarios a partir de dispositivos manuales (ejemplo, teclado).
- Debe permitir hacer reportes según patrones de materiales.

### **Proceso de recuperación de datos**

- El sistema dispondrá de capacidades recuperar información en caso de algún altercado (falla eléctrica) sin necesidad de volver a cargar la selección de material.
- Tendrá capacidad para establecer procesos de respaldo de datos automáticos.

### 3.3 RECOPIACIÓN Y ANÁLISIS DE INFORMACIÓN ACTUAL

La recopilación y análisis de la información nos ayuda a conseguir que se cumpla con el objetivo planeado en el desarrollo del sistema.

Las razones principales por las cuales es necesario reunir información acerca del manejo del almacén son:

- Nadie tiene el conocimiento del manejo del almacén como las personas que laboran en él.
- Necesitamos conocer el funcionamiento y las necesidades que se tienen, para desarrollar el nuevo sistema y que cumpla con los requisitos.

Los usuarios que en nuestro caso son los jefes de departamento y el personal del almacén son nuestras principales fuentes, ya que éstos hacen uso directo de la información y son capaces de poder suministrar u obtener información necesaria para ser utilizada.

#### **Administración del material del almacén.**

Actualmente se hace uso de los siguientes formatos para la obtención de la información, los cuales seguiremos utilizando cuando el sistema comience a entrar en funcionamiento, ya que las funciones para las que se desarrollará el sistema principalmente son el control del inventario y la generación de reportes:

- Vale de salida de almacén.

Contiene número de folio, fecha de recepción y fecha de entrega, los datos del solicitante, del material solicitado, nombre y firma del personal que autoriza.

Ver figura 3.3.1



<b>Nombre del campo</b>	<b>Tipo de dato</b>
Folio	Entero consecutivo
Fechas	Fecha
RFC	Cadena de caracteres
Nombre del solicitante	Cadena de caracteres
Clave y nombre del departamento	Cadena de caracteres
Número	Entero
Código	Cadena de caracteres
Descripción	Cadena de caracteres
Unidad de medida	Entero
Cantidad (solicitada y entregada)	Entero
Observaciones	Cadena de caracteres

Figura 3.3.2 Tabla de datos (vale de salida de almacén).

- Vale de solicitud interna de compra.

Contiene número de folio, fecha, los datos del solicitante, del material solicitado, del proveedor, del área de adquisiciones, área de presupuesto, nombre y firma del personal que autoriza. Ver figura 3.3.3





## ANÁLISIS DE PLANTEAMIENTO DEL PROBLEMA

<b>OBSERVACIONES:</b>		
VO. BO. RESPONSABLE DE BIENES Y SUMINISTROS	VO. BO. SUFICIENCIA	AUTORIZACIÓN SECRETARIO O JEFE DE UNIDAD
<u>ING. JORGE OCHOA MORENO</u> NOMBRE Y FIRMA	<u>C.P. DAVID TÉLIZ ESTRADA</u> NOMBRE Y FIRMA	<u>ING. LUIS JIMÉNEZ ESCOBAR</u> NOMBRE Y FIRMA

Figura 3.3.3 Vale de solicitud interna de compra

Esta información es capturada en un libro de Excel, si el material ya se encuentra únicamente se agrega al consecutivo en existencia; en caso contrario se registra por primera vez y es agregado al control del inventario, para que el usuario tenga conocimiento de su existencia en una futura consulta.

Como se mencionó anteriormente el almacén continuará funcionando como hasta ahora, los vales se seguirán llenando por los usuarios solo que gracias al sistema se llevará un control por medio de la base de datos, en la cual se harán consultas sobre el material disponible y el solicitado por el personal. Y así poder realizar reportes; incluso se puede prever la falta de insumos ya que se pueden tomar tendencias de lo más solicitado en cierto tiempo y adelantarse al pedido del mismo sin necesidad de saber que ya no se cuenta con dicho insumo.

Los tipos de datos que se manejan en el llenado de los vales se muestran en la tabla siguiente:

Nombre del campo	Tipo de dato
Folio	Entero consecutivo
Fechas	Fecha
Área del solicitante	Cadena de caracteres
Nombre del solicitante	Cadena de caracteres
Autorización del responsable del área	Cadena de caracteres
Teléfono	Entero

Número	Entero
Cantidad	Cadena de caracteres
Descripción	Cadena de caracteres
Unidad	Entero
Precio por unidad	Cadena de caracteres
Importe	Cadena de caracteres
Proveedor	Cadena de caracteres
Anexos	Cadena de caracteres
Con cargo a	Cadena de caracteres
Área de adquisiciones	Cadena de caracteres
Área de presupuesto	Cadena de caracteres

**Figura 3.3.4 Tabla de datos (vale de solicitud interna de compra).**

### **3.4 IDENTIFICACION DE LOS POSIBLES MODULOS DEL SISTEMA**

El sistema generador de vales de almacén se definirá por los siguientes módulos para su funcionamiento operacional: vea figura 3.4.1

- Módulo de Circulación.
- Módulo Catalogación
- Módulo de Directorio.
- Módulo de Materiales.
- Módulo de Consulta.
- Módulo de Administración.
- Módulo de Identificación de Usuario

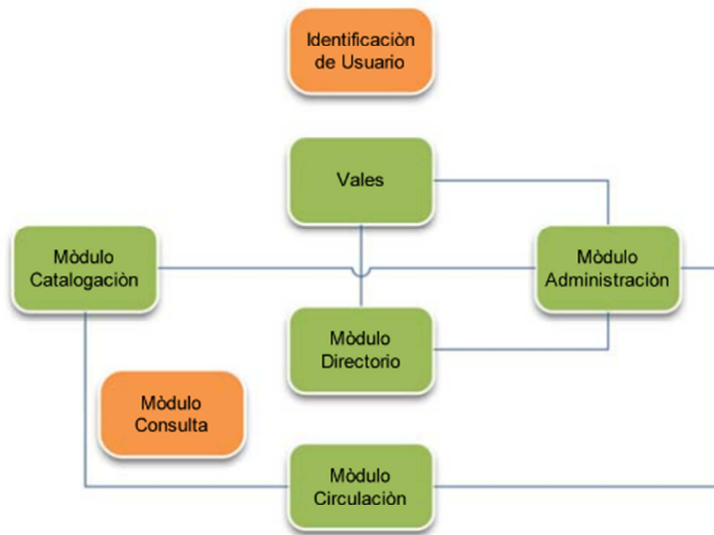


Figura 3.4.1 Diagrama de bloques.

En la Figura 3.4.1 se observa la interacción de los módulos entre sí, cumpliendo el objetivo de funcionamiento del sistema.

### Módulo de circulación

En el módulo de circulación se llevaran a cabo todos los movimientos del material catalogado en la figura 3.4.2 se puede observar el diagrama de bloques del módulo a continuación.



Figura 3.4. 2 Diagrama módulo de circulación

### **Solicitud interna de compra**

Este se refiere cuando el usuario con una lista ya definida por los niveles de materiales en existencia en el almacén. Genera un vale con la lista de materiales y los precios por pieza para surtir el almacén.

### **Vale de salida de almacén**

En esta parte se tendrá la lista de productos disponibles en el almacén, donde se pueden seleccionar los artículos, ver el pedido o generar el vale para su impresión.

### **Información del material y del usuario**

Se refiere a si el material está disponible para poder disponer, si no está en existencia.

La información del usuario es: el área del usuario.

### **Módulo de catalogación**

En esta parte se tendrá una lista de productos disponibles en el almacén donde se puede seleccionar un *máximo de 17*. Los materiales del almacén están organizados alfabéticamente para mayor comodidad.

Si en el almacén se requiere la generación de un vale, bastara con trabajar un nivel de catalogación con los siguientes elementos:

- Nombre del solicitante
- RFC del solicitante
- Cantidad
- Descripción de los bienes

## ANÁLISIS DE PLANTEAMIENTO DEL PROBLEMA

- Unidad de medida
- Precio por unidad
- Código
- Solicitada
- Entrega
- Observaciones

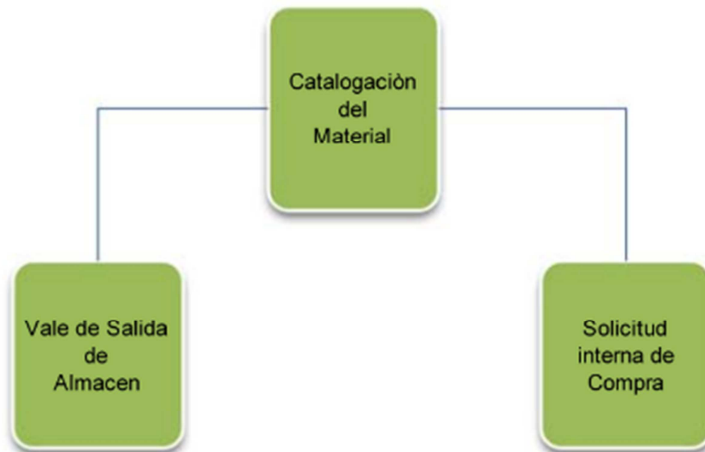


Figura 3.4.3 Diagrama módulo de catalogación

En la figura 3.4.3 se observa que se generan dos tipos de vales uno de solicitud interna de compra y la de salida de almacén, ambos cuentan con el mismo catálogo uno es para la compra y abastecer el material y la otra es para la solicitud de los usuarios.

## **Módulo de Directorio**

En este módulo se guardara toda la informaron del usuario del sistema generador de vales de almacén, estos datos son:

- Información personal
  - Nombre del solicitante
  - Teléfono
  - RFC
- Información de la institución
  - Área del solicitante
- Información de vales
  - Vale salida de almacén
  - Solicitud interna de compra

### Información de almacén

- Unidad
- Situación
- Precio por unidad
- Vale autorizado

Información general

- Datos de facturación



Figura 3.4.4 Diagrama moduló directorio

**Módulo de Consulta**

Este se refiere a la selección de materiales por parte del usuario. Este módulo facilitara a los usuarios la selección de material y les evitara errores de dedo.

**Módulo de Administración**

Se pretende utilizar para guardar los parámetros de configuración del sistema así como también catálogos para todos los demás módulos como se muestra en la figura 3.4.5.

Los catálogos pueden ser para el módulo de directorio .módulo de circulación y modulo Información de vales.





Figura 3.4.5 Diagrama moduló Administración

### 3.5 COMPARACIÓN Y UBICACIÓN DE LAS HERRAMIENTAS DE SOLUCIÓN

La necesidad de acceso a la información provoca cambios en diversos rubros, tecnológicos, administrativos, sociales, entre otros. Estos cambios tienen repercusión sobre toda la organización, es así como surge la necesidad de utilizar nuevas tecnologías destinadas a mejorar la administración de nuestra institución. Para tal motivo se desarrollará un sistema capaz de llevar de una manera más eficiente el control del inventario del almacén.

Hoy en día existen diversas opciones tecnológicas que se pueden elegir para implementar este sistema, en nuestro caso se desarrollará para que sea accesible vía Web, esto debido a que la Web no solo se limita a presentar textos y enlaces, sino que también puede ofrecernos imágenes, videos, sonido, etcétera.

Para establecer un sistema vía Web se requiere del siguiente software:

- Un servidor Web.
- Un manejador de base de datos (DBMS).
- Un lenguaje de programación, para del lado del servidor y para ejecutarse del lado del cliente.

- Un navegador de Internet.

Un servidor Web es un programa que implementa el protocolo HTTP (hypertext transfer protocol). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas Web o páginas HTML (hypertext markup language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos. Un servidor Web se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP que solemos conocer como navegador. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita.

El sistema manejador de bases de datos (DBMS) es la porción más importante del software de un sistema de base de datos. Un DBMS es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de alguna tarea específica.

Las funciones principales de un DBMS son:

- Crear y organizar la Base de Datos.
- Establecer y mantener las trayectorias de acceso a la base de datos de tal forma que los datos puedan ser acusados rápidamente.
- Manejar los datos de acuerdo a las peticiones de los usuarios.
- Registrar el uso de las bases de datos.
- Interacción con el manejador de archivos. Esto a través de las sentencias en DML, al comando del sistema de archivos. Así el manejador de base de datos es el responsable del verdadero almacenamiento de los datos.
- Respaldo y recuperación. Consiste en contar con mecanismos implantados que permitan la recuperación fácilmente de los datos en caso de ocurrir fallas en el sistema de base de datos.
- Control de concurrencia. Consiste en controlar la interacción entre los usuarios concurrentes para no afectar la inconsistencia de los datos.
- Seguridad e integridad. Consiste en contar con mecanismos que permitan el control de la consistencia de los datos evitando que éstos se vean perjudicados por cambios no autorizados o previstos.

Se requiere de un lenguaje de programación que nos permita crear páginas dinámicas que sean reconocidas, interpretadas y ejecutadas por el propio servidor. Las páginas dinámicas del servidor se suelen escribir en el mismo archivo HTML, mezclado con el código HTML. Cuando una página es solicitada por parte de un cliente, el servidor ejecuta los scripts y se genera una página resultado, que solamente contiene código HTML. Este resultado final es el que se envía al cliente y puede ser interpretado sin lugar a errores ni incompatibilidades, puesto que sólo contiene HTML. Para escribir páginas dinámicas de servidor existen varios lenguajes: Common Gateway Interface (CGI) comúnmente escritos en Perl, Active Server Pages (ASP), Hipertext Preprocesor (PHP), y Java Server Pages (JSP).

Las ventajas de este tipo de programación son que el cliente no puede ver los scripts, ya que se ejecutan y transforman en HTML antes de enviarlos. Además son independientes del navegador del usuario, ya que el código que reciben es HTML fácilmente interpretable.

Otro lenguaje que se requiere es uno que se ejecute en el cliente (navegador de Internet), para ello se utilizan dos lenguajes de programación principalmente: Javascript y Visual Basic Script (VBScript). Las páginas del cliente son muy dependientes del sistema donde se están ejecutando y esa es su principal desventaja, ya que cada navegador tiene sus propias características, incluso cada versión, y lo que puede funcionar en un navegador puede no funcionar en otro. Como ventaja se puede decir que estas páginas descargan al servidor algunos trabajos, ofrecen respuestas inmediatas a las acciones del usuario y permiten la utilización de algunos recursos de la máquina local.

Antes de seleccionar un conjunto de productos para desarrollar este sistema debemos considerar algunos aspectos como: seguridad, costo, compatibilidad y rendimiento.

Teniendo como plataforma seleccionaremos Linux. En cuanto al costo, se pretende que el sistema no resulte costoso en cuanto a compra de licencias por el uso de productos para su implementación, motivo por el cual se usará software libre, reduciendo así los costos.

Partiendo del hecho de que como plataforma se usara Linux y software libre para el servidor Web, para la base de datos y lenguaje de programación, se debe tener en cuenta la compatibilidad y rendimiento entre estas herramientas.

A continuación se comparan algunas tecnologías con el fin de encontrar la mejor conjunción entre el servidor Web, la base de datos y los lenguajes de programación.

Comenzando con los manejadores de base de datos, tenemos: MySQL y PostgreSQL.

- MySQL: Su principal objetivo de diseño fue la velocidad. Otra característica importante es que consume muy pocos recursos, tanto de CPU como de memoria.
- PostgreSQL: Postgres intenta ser un sistema de bases de datos de mayor nivel que MySQL, a la altura de Oracle, Sybase o Interbase.

En la tabla 3.5.1 observamos las ventajas que tienen ambos productos:

MySQL	PostgreSQL
Mayor rendimiento. Mayor velocidad tanto al conectar con el servidor como al servir selects y demás.	Por su arquitectura de diseño, escala muy bien al aumentar el número de CPU's y la cantidad de RAM.
Mejores utilidades de administración (backup, recuperación de errores, etc.).	Soporta transacciones y desde la versión 7.0, claves ajenas (con comprobaciones de integridad referencial).
Aunque se cuelgue, no suele perder información ni corromper los datos.	Tiene mejor soporte para triggers y procedimientos en el servidor.
Mejor integración con PHP	Soporta un subconjunto de SQL92 mayor que el que soporta MySQL. Además, tiene ciertas características orientadas a objetos.
No hay límites en el tamaño de los registros.	No disponible
Mejor control de acceso, en el sentido de qué usuarios tienen acceso a qué tablas y con qué permisos.	No disponible
MySQL se comporta mejor que Postgres a la hora de modificar o añadir campos a una tabla.	No disponible

Tabla 3.5.1 Ventajas de MySQL y PostgreSQL.

En la tabla 3.5.2, tenemos las desventajas que presentan cada uno de los manejadores:

MySQL	PostgreSQL
No soporta transacciones, "roll-backs" ni subselects.	Consume Bastantes más recursos y carga más el sistema.
No considera las claves ajenas. Ignora la integridad referencial, dejándola en manos del programador de la aplicación.	Límite del tamaño de cada fila de las tablas a 8k (se puede ampliar a 32k recompilando, pero con un coste añadido en el rendimiento).
No disponible	Es de 2 a 3 veces más lenta que MySQL
No disponible	Menos funciones en PHP.

Tabla 3.5.2 Desventajas de MySQL y PostgreSQL.

En cuanto a los lenguajes de programación del lado del servidor, se tiene 3 opciones: PHP, ASP, JSP. A continuación se comparan estos lenguajes:

Característica	PHP	ASP	JSP
Plataforma	Independiente de Plataforma y servidor (Unix, Windows, Linux)	Se trata de un sistema propietario que es usado nativamente sólo por Microsoft Internet Information Server (IIS). Esto limita su disponibilidad a servidores basados en Win32.	Independiente de Plataforma y servidor. (Unix, Windows, Linux)
Seguridad	Instalado sobre servidores Unix o Linux, es más veloz y seguro. Además, PHP permite configurar el servidor de modo que se permita o rechacen diferentes usos, lo que puede hacer al lenguaje seguro dependiendo de las necesidades	Debido que solo funciona bajo Windows, es más susceptible a sufrir caídas, e infecciones de virus	Las páginas JSP son compiladas en Servlets por lo que actúan como una puerta a todos los servicios Java de Servidor y librerías Java para aplicaciones http ayudando a proteger el sistema contra las caídas
Performance (Rendimiento)	En el caso de estar montado sobre un servidor Unix o Linux, es más rápido dado que se ejecuta en un único espacio de memoria	Son demasiadas las comunicaciones entre componentes COM que se realizan entre todas las tecnologías implicadas en una página ASP. Consumiendo así mas recursos	Los componentes JSP son reusables en distintas plataformas (UNIX, Windows), también ayuda en el manejo de la memoria protegiendo contra fallos de memoria
Soporte	Recibe contribuciones de diversos desarrolladores	ASP es específica de Microsoft que desarrolla sus procesos internamente.	El API JSP se beneficia de la extendida comunidad JAVA existente.

Tabla 3.5.3 Comparación entre PHP, ASP y JSP.

Una vez comparado los manejadores de base de datos, observamos que MySQL resulta la mejor opción ya que no consume demasiados recursos, es más seguro y tiene un mejor rendimiento.

En cuanto a los lenguajes de programación observamos que PHP y JSP son buenas opciones ya que ASP, solo trabaja en Windows, motivo por el cual queda descartado, por otro lado considerando que MySQL tiene una mejor interacción con PHP que JSP, PHP resulta la mejor opción.

PHP puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo Linux, muchas variantes Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS y probablemente alguno más.

PHP soporta la mayoría de servidores web de hoy en día, incluyendo Apache, Microsoft Internet Information Server, Personal Web Server, Netscape e iPlanet, O'Reilly Website Pro server, Caudium, Xitami, OmniHTTPd y muchos otros. PHP tiene módulos disponibles para la mayoría de los servidores.

Con PHP no se encuentra limitado a resultados en HTML. Entre las habilidades de PHP se incluyen: creación de imágenes, archivos PDF y películas Flash (usando libswf y Ming) sobre la marcha. También puede presentar otros resultados, como XHTML y archivos XML. PHP puede autogenerar estos archivos y almacenarlos en el sistema de archivos en vez de presentarlos en la pantalla.

Quizá la característica más potente y destacable de PHP es su soporte para una gran cantidad de bases de datos. Escribir la interfaz vía Web para una base de datos es una tarea simple con PHP. Los siguientes productos están soportados actualmente:

Adabas D	Ingres	Oracle (OCI7 and OCI8)
dBase	InterBase	Ovrimos
Empress	FrontBase	PostgreSQL
FilePro (read-only)	mSQL	Solid
Hyperwave	Direct MS-SQL	Sybase
IBM DB2	MySQL	Velocis
Informix	ODBC	Unix dbm

Tabla 3.5.4 Productos soportados por PHP.

PHP también cuenta con soporte para comunicarse con otros servicios usando protocolos tales como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (en Windows) y muchos otros. También se pueden crear sockets puros. PHP soporta WDDX para el intercambio de datos entre lenguajes de programación en web. Y hablando de interconexión, PHP puede utilizar objetos Java de forma transparente como objetos PHP Y la extensión de CORBA puede ser utilizada para acceder a objetos remotos.

En cuanto al lenguaje que se ejecuta del lado del cliente debemos considerar cual es más compatible con los navegadores de Internet más utilizados. Debido a que Visual Basic Script (VBScript) solo es compatible con Internet Explorer queda descartado ya que reduce las opciones en cuanto al funcionamiento de la aplicación, pues forzaría a los usuarios a utilizar un navegador en específico, mientras que Javascript es compatible con más navegadores, como son Internet Explorer, Netscape, Opera, entre otros, es por ello que se utilizará Javascript como lenguaje para ejecutarse del lado del cliente.

En cuanto al servidor Web, de entre las opciones que se mencionaron anteriormente, el servidor Apache es uno de los mejores servidores de Web utilizados en la red Internet desde hace mucho tiempo, únicamente le hace competencia un servidor de Microsoft, el Internet Information Server (IIS). Por lo que este servidor es uno de los mayores triunfos del software libre.

Es un servidor Web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos (HTTP 1.1)

## ANÁLISIS DE PLANTEAMIENTO DEL PROBLEMA

- Implementa los últimos protocolos.
- Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo y con la API de programación de módulos.
- Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para solución de los mismos.

Es por ello que usaremos Apache como servidor Web.

Una vez que se revisaron las opciones tecnológicas para implementar el sistema se determinó que MySQL junto con Apache y PHP forman un buen equipo.



## Capítulo 4

### Implementación y personalización del sistema

#### 4.1 ELECCIÓN DE LA METODOLOGÍA DE DESARROLLO

Una metodología de desarrollo de software se refiere a un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar.

##### Ventajas de tener una metodología

- Mejora de los procesos de desarrollo
- Todas las personas del proyecto trabajan bajo un marco común.
- Estandarización de conceptos, actividades y nomenclaturas.
- Actividades de desarrollo apoyadas por procedimientos y guías.
- Resultados del desarrollo predecibles
- Uso de herramientas de ingeniería software
- Planificación de actividades en base a un conjunto de tareas definidas y a la experiencia en otros proyectos.
- Recopilación de mejores prácticas para proyectos futuros

Modelo de proceso	Funciona con requisitos y arquitectura no predefinidos	Produce software altamente fiable	Gestión de riesgos	Permite correcciones sobre la marcha	Visión del progreso por el Cliente y el Jefe del proyecto
Cascada	Bajo	Alto	Bajo	Bajo	Bajo
Espiral	Alto	Alto	Alto	Medio	Medio

## **SSADM**

SSADM (Structured Systems Analysis and Design Method) es una metodología de aproximación en cascada para el desarrollo de sistemas de información y que puede ser considerada como la más completa de las metodologías estructuradas, por su garantía contrastada a lo largo de los años por los desarrolladores informáticos. Es la metodología estándar de desarrollo de proyectos del gobierno del Reino Unido.

### **Objetivos**

SSADM se basa en 3 vistas fundamentales:

- **Modelo lógico de datos:** Es el proceso de identificar, modelar y documentar los requerimientos de datos de un sistema de información de negocio. El modelo lógico de datos consiste en la estructura lógica de datos (LDS) y su documentación asociada. LDS representa las entidades y relaciones.
- **Modelo de flujo de datos:** Es el proceso de identificar, modelar y documentar como fluyen los datos a través del sistema de información. El Modelo de flujo de datos consiste en un conjunto de diagramas de flujo de datos (DFD) y su documentación asociada. Los DFD representan los procesos, entidades externas y flujos de datos.
- **Modelo de Eventos de entidad:** Es el proceso de identificar, modelar y documentar como los eventos de negocio que afectan a cada entidad y la secuencia en que ocurren. Un modelo entidad/evento consiste en un conjunto de historia de vida de las entidades y la correspondiente documentación. Es decir, este proceso representa el comportamiento de un sistema dinámico a lo largo del tiempo.

SSADM considera las estructuras de datos con mayor estabilidad que los procesos, y por lo tanto los datos forman la columna vertebral de la metodología. De este modo, SSADM pertenece a la familia de los métodos “orientado a los datos”.

SSADM es una aproximación top-down donde se representa el esquema de alto nivel y se va descomponiendo gradualmente en niveles inferiores. Además,

aplica técnicas individuales así como el modo en que se pasa el control de una a otra una vez que la primera ha finalizado su actividad.

La gestión del proyecto se centra más en la monitorización de la calidad y la completitud de su producto que en la aplicación de las técnicas que lo crean.

SSADM consiste en una arquitectura de especificación de 3 esquemas, considerando 3 áreas del sistema de información:

- Capa externa mediante la que los usuarios interactúan con el sistema
- Diseño interno
- Modelo conceptual que representa los requerimientos de negocio y sobre los que se basa el diseño interno.

SSADM, cubre las tres fases fundamentales del ciclo de vida del desarrollo software: Estudio de viabilidad, Análisis y Diseño como se observa en la figura 4.1.1. Aunque no está diseñada para realizar el mantenimiento de las aplicaciones, lo que hace es suministrar la documentación completa y precisa para poder mantener el sistema fácilmente.

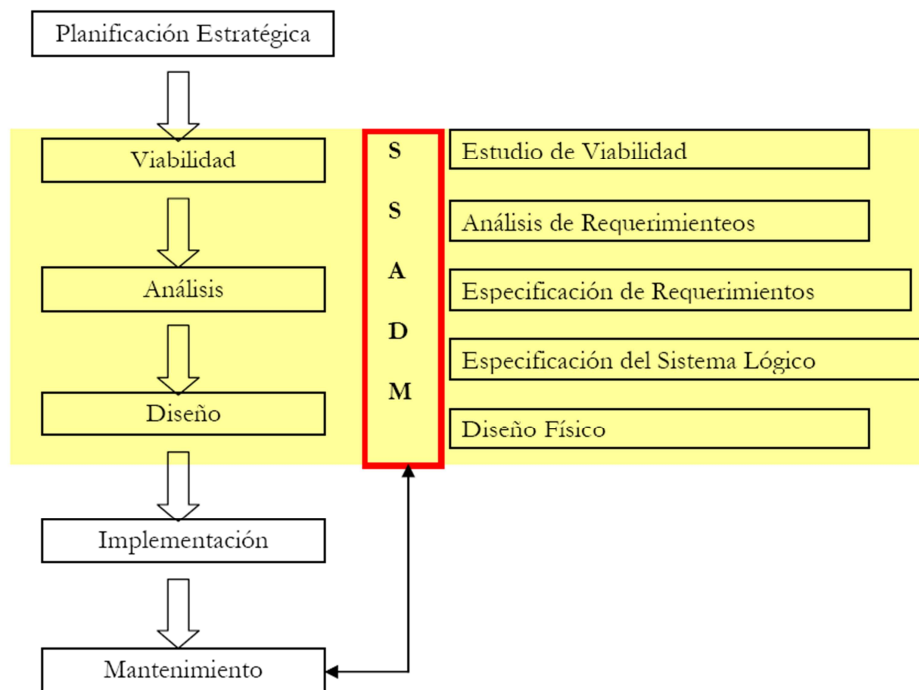


Fig. 4.1.1 Alcance de SSADM

## Estructura

En la Figura 4.1.2 se puede observar con un mayor nivel de detalle las distintas tareas que contienen cada una de las tres fases del ciclo de vida de SSADM.

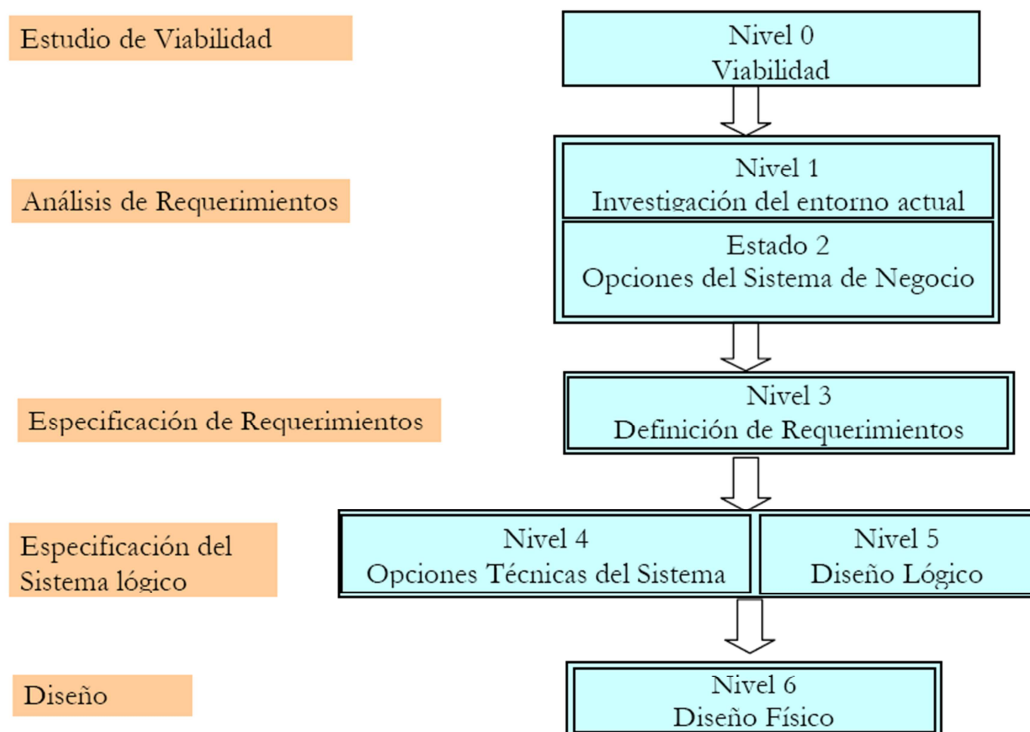


Figura. 4.1.2 Estructura de SSADM.

Estudio de Viabilidad: Durante esta fase se realizan las siguientes actividades:

- Se define el alcance del sistema de información propuesto.
- Se investigan las diversas opciones para el desarrollo del proyecto.
- Se selecciona una de las opciones realizando un análisis coste-beneficio.
- Se decide continuar con el proyecto o abandonar el proyecto en caso de encontrar riesgos que impidan su viabilidad.

Investigación del entorno actual: Durante esta actividad se realizan las siguientes tareas:

- En muchas ocasiones el sistema de información puede ser una sustitución una ampliación de uno ya existente.
- En esta situación, se realiza un análisis completo de requerimientos mediante la modelización del sistema actual con el objeto de extraer los problemas existentes y las nuevas necesidades.
- El análisis del sistema actual detecta los puntos débiles que se deben cubrir.

Opciones de negocio del sistema: Durante esta actividad se realizan las siguientes tareas:

- Al final del nivel 1, deberíamos tener un conjunto de requerimientos claro para poder examinar en esta tarea y proponer las distintas opciones de solución.
- Aunque se deben considerar los aspectos técnicos y físicos de implementación, lo más importante en esta tarea es definir las soluciones de negocio, pero no el entorno tecnológico.

Definición de requerimientos: Durante esta tarea se realiza:

- La transformación de los requerimientos en las especificaciones de lo que se quiere del sistema.
- Se aplican las técnicas de modelado y se pasa del análisis al diseño.

Opciones de diseño técnico:

- De la actividad anterior se obtiene la información necesaria para proponer las alternativas técnicas de implementación.
- Se especificarán las distintas opciones de hardware, software y de plataforma de desarrollo para poder seleccionar la más adecuada.
- Esta tarea se realiza en paralelo con el diseño lógico.

Diseño lógico:

- Se obtiene el proceso de diseño lógico independientemente del entorno técnico particular.

- El diseño resultante podrá ser implementado en diversas plataformas.
- Puede ser un modelo de cómo el sistema satisfará los requerimientos del usuario.

Diseño físico:

- La información obtenida en la fase anterior se usa para trasladar al diseño físico basado en el entorno técnico seleccionado.
- Se utilizan las técnicas de selección del entorno, ya que SSADM simplemente se limita a suministrar unas directrices genéricas.

### **Técnicas**

Las técnicas que usa SSADM [Longworth 1986] son:

- Diagramas de flujos de datos (DFD): Para representar gráficamente los flujos de información en el sistema considerado, así como los flujos de información con el exterior, es decir, con otros subsistemas o sistemas.
- Diagramas de datos lógicos (Vista Datos):
  - Estructura lógica de datos (LDS): Se basa en el modelo entidad/relación de CHEN, en el que se dibuja el diagrama inicial de E/R y se ajusta posteriormente convirtiendo las relaciones en entidades compuestas e identificando nuevas relaciones.
  - Diagramas entidad/relación (ER): Para representar mediante estructuras de datos el mundo real.

Diagramas de comportamiento de entidad (Vista Eventos):

- Diagramas Historia de vida de entidades (ELH) Para representar todo el proceso de creación, actualización y desaparición de una entidad a lo largo del tiempo.
- Diagramas de correspondencia (ECD): Para relacionar el número máximo de ocurrencias de una entidad que puede intervenir en una ocurrencia de relación.

## MERISE

### Objetivos

MERISE es una metodología de la Administración francesa, creada por iniciativa del Ministerio de Industria Francés y desarrollada por Tardieu, Rochfeld y Colleti (Tardieu et al 1983). Para el Ministerio de Industria Francés.

MERISE introduce dos ciclos complementarios: ciclo de abstracción y ciclo de decisión.

El ciclo de abstracción se basa en tres niveles:

- Nivel conceptual: Es donde se define el “qué” es decir, los objetivos y limitaciones. En este nivel se realiza un tratamiento de los datos según el modelo conceptual de datos y los procesos según el modelo conceptual de procesos.
- Nivel Organizativo: Es donde se define la organización adecuada que hay que implantar para alcanzar los objetivos asignados y se realiza un tratamiento de los datos y el modelo organizativo de tratamientos para la realización de los procesos.
- Nivel físico u operativo: Se realiza la integración de los medios técnicos necesarios para el proyecto, utiliza el modelo físico de datos para los datos y el modelo operativo de tratamiento para los procesos.

En la tabla 4.1.3 se define un modelo de datos y un modelo de procesos para cada nivel de abstracción.

NIVELES	DATOS	TRATAMIENTOS
CONCEPTUAL	Modelo Conceptual de Datos	Modelo Conceptual de Tratamientos
ORGANIZATIVO	Modelo Lógico de Datos	Modelo Organizativo de Tratamientos
FÍSICO	Modelo Físico de Datos	Modelo Operativo de Tratamientos

Figura 4.1.3 Arquitectura MERISE

MERISE, al igual que SSADM, cubre las cuatro fases fundamentales del ciclo de vida del desarrollo software: Estudio preliminar, Análisis, Diseño e Implementación (Gabay 1989) como se puede observar en la Figura 4.1.4. De la misma forma, aunque no está diseñada para realizar el mantenimiento de las aplicaciones, lo que hace es suministrar la documentación completa y precisa para poder mantener el sistema fácilmente.

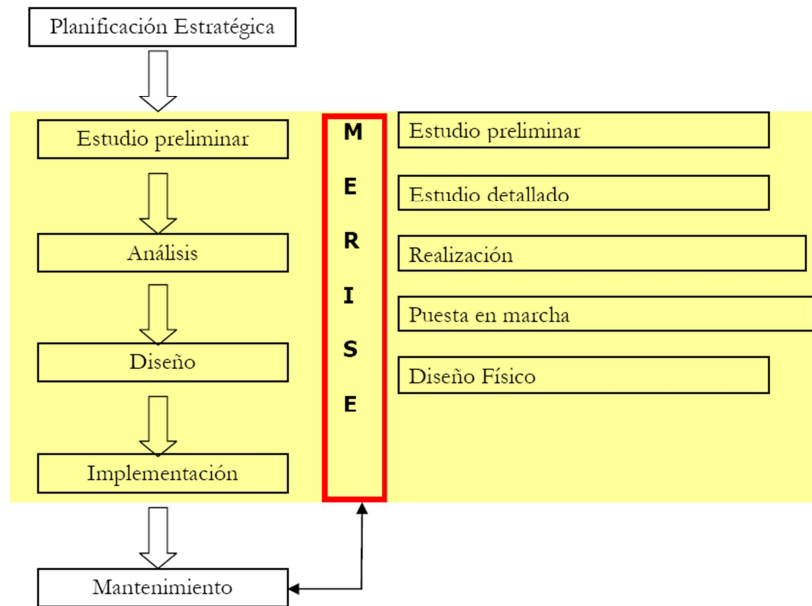


Figura 4.1.4: Ámbito y alcance de MERISE

### Estructura

Fases de la Metodología:

- Estudio preliminar: En esta fase se analiza la situación existente y la propuesta de una solución global atendiendo a los criterios de gestión, de la organización y decisiones adoptadas por el comité directivo del proyecto.
- Estudio detallado: En esta fase se define a nivel funcional la solución
- Implementación: Es la fase donde se describe el entorno técnico con la distribución de los datos en los ficheros y los tratamientos en módulos de programas, también se construyen los programas con su codificación y pruebas.



- Realización y puesta en marcha: En esta fase se realiza la instalación, formación del personal, así como la implantación de los medios técnicos y organizativos y la recepción por parte del usuario.

### **Técnicas**

En el desarrollo de un proyecto son 3 los grupos de trabajo que intervienen:

- Comité director: Su función es fijar los objetivos y tomar las decisiones importantes en el desarrollo del mismo, interviniendo al final de cada etapa para realizar el control y seguimiento del proyecto y aprobando los informes de las mismas. Este equipo de trabajo estará formado por los directivos de cada área afectada, los responsables de los servicios implicados, el responsable del servicio de informática y el jefe del proyecto.
- Comité de usuarios: Su función es realizar un seguimiento sistematizado para la comprobación a lo largo del desarrollo de los cumplimientos de las actividades asignadas, mediante la comprobación de los diseños de las pantallas, validación de informes y de listados, resolución de problemas, etc. Estará compuesto por los responsables de los servicios afectados.
- Equipo de desarrollo: Su misión es elaborar los informes y documentación contemplados en cada una de las fases de desarrollo, así como la realización de los análisis, programación, pruebas, instalación y puesta en marcha. Este equipo estará formado por el jefe del proyecto y los analistas y programadores asignados al mismo, así como el representante del grupo de usuarios.

Además de estos grupos de trabajo, MERISE utiliza las técnicas:

- DFD (Diagramas de flujo de datos): Para la representación gráfica de la organización identificando los flujos de información entre los diferentes actores.
- Modelo conceptual de datos (Modelo entidad/relación): Para la representación mediante estructuras del mundo real, pudiendo optar por cualquier tipo de base de datos.

- Modelo lógico de datos: Para adaptar el modelo conceptual de datos al sistema gestor de base de datos elegido.
- Modelo conceptual de tratamientos: Para representar las acciones a realizar sobre los datos con el objeto de obtener los resultados previstos. Para ello, utiliza las redes Petri.
- Modelo organizativo de tratamientos: Para describir cómo se va a organizar la ejecución de las mismas.
- Modelo operacional de tratamientos: Se parte de la descomposición de los procedimientos y fases anteriores y se obtienen los procedimientos manuales, las fases en tiempo real, las fases en tiempo diferido donde se incluyen las entradas, tratamientos y listados.

### **Espiral**

El desarrollo en espiral es un modelo del ciclo de vida del software definido por primera vez por Barry Boehm en 1988, utilizado generalmente en la Ingeniería de software. Las actividades de este modelo se conforman en una espiral, en la que cada bucle o iteración representa un conjunto de actividades. Las actividades no están fijadas a priori, sino que las siguientes se eligen en función del análisis de riesgo, comenzando por el bucle interior.

### **Introducción**

La Ingeniería de software, se vale y establece de una serie de modelos que establecen y muestran las distintas etapas y estados por los que pasa un producto de software, desde su concepción inicial, pasando por su desarrollo, puesta en marcha y posterior mantenimiento, hasta la retirada del producto. A estos modelos se les denomina «modelos de ciclo de vida del software». El primer modelo concebido fue el de Royce, más comunmente conocido como *desarrollo en cascada* o *desarrollo lineal secuencial*. Este modelo establece que las diversas actividades que se van realizando al desarrollar un producto de software se suceden de forma lineal.

Boehm, autor de diversos artículos de ingeniería del software; modelos de estimación de esfuerzo y tiempo que se consume en hacer productos de

software; y Modelos del Ciclo de Vida; ideó y promulgó un modelo desde un enfoque distinto al tradicional en Cascada: El Modelo Evolutivo Espiral. Su Modelo de Ciclo de Vida en Espiral tiene en cuenta fuertemente el riesgo que aparece a la hora de desarrollar software. Para ello, se comienza mirando las posibles opciones de desarrollo, se opta por la de riesgo más asumible y se hace un ciclo de la espiral. Si el cliente quiere seguir haciendo mejoras en el software, se vuelve a evaluar las distintas nuevas opciones y riesgos y se realiza otra vuelta de la espiral, así hasta que llegue un momento en el que el producto de software desarrollado sea aceptado y no necesite seguir mejorándose con otro nuevo ciclo.

Este modelo fue propuesto por Boehm en 1988. Básicamente consiste en una serie de ciclos que se repiten en forma de espiral, comenzando desde el centro. Se suele interpretar como que dentro de cada ciclo de la espiral se sigue un Modelo en Cascada, pero no necesariamente debe ser así. El Espiral puede verse como un modelo evolutivo que conjuga la naturaleza iterativa del modelo MCP con los aspectos controlados y sistemáticos del Modelo en Cascada, con el agregado de gestión de riesgos.

En cada vuelta o iteración hay que tener en cuenta:

- Los Objetivos: Que necesidad debe cubrir el producto.
- Opciones: Las diferentes formas de conseguir los objetivos de forma exitosa, desde diferentes puntos de vista como pueden ser:
  - Características: experiencia del personal, requisitos a cumplir, etc.
  - Formas de gestión del sistema.
  - Riesgo asumido con cada opción.

Desarrollar y Verificar: Programar y probar el software.

Si el resultado no es el adecuado o se necesita implementar mejoras o funcionalidades

Se planificarán los siguientes pasos y se comienza un nuevo ciclo de la espiral. La espiral tiene una forma de caracola y se dice que mantiene dos dimensiones, la radial y la angular:

- Angular: Indica el avance del proyecto de software dentro de un ciclo.
- Radial: Indica el aumento del coste del proyecto, ya que con cada nueva iteración se pasa más tiempo desarrollando.

Este sistema es muy utilizado en proyectos grandes y complejos como puede ser, por ejemplo, la creación de un Sistema Operativo.

Al ser un modelo del Ciclo de Vida orientado a la gestión de riesgo se dice que uno de los aspectos fundamentales de su éxito radica en que el equipo que lo aplique tenga la necesaria experiencia y habilidad para detectar y catalogar correctamente los riesgos.

### Tareas

Para cada ciclo habrá cuatro actividades:



Figura. 4.1.5 Ciclos del Modelo Espiral

Determinar o fijar objetivos:

- Fijar también los productos definidos a obtener: requerimientos, especificación, manual de usuario.
- Fijar las restricciones.

- Identificación de riesgos del proyecto y estrategias u opciones para evitarlos.
- Hay una cosa que solo se hace una vez: planificación inicial o previa.

Análisis del riesgo:

- Se estudian todos los riesgos potenciales y se seleccionan una o varias alternativas propuestas para reducir o eliminar los riesgos.

Desarrollar, verificar y validar (probar):

- Tareas de la actividad propia y de prueba.
- Análisis de opciones e identificación resolución de riesgos.
- Dependiendo del resultado de la evaluación de los riesgos, se elige un modelo para el desarrollo, el que puede ser cualquiera de los otros existentes, como formal, evolutivo, cascada, etc. Así si por ejemplo si los riesgos en la interfaz de usuario son dominantes, un modelo de desarrollo apropiado podría ser la construcción de prototipos evolutivos. Si los riesgos de protección son la principal consideración, un desarrollo basado en transformaciones formales podría ser el más apropiado.

Planificar:

- Revisamos todo lo hecho, evaluándolo, y con ello decidimos si continuamos con las fases siguientes y planificamos la próxima actividad.

Mecanismos de control:

- La dimensión radial mide el coste.
- La dimensión angular mide el grado de avance del proyecto.

Variaciones del Modelo En Espiral

- Modelo en Espiral Típico de seis regiones.
- Modelo en espiral WIN.

Esta metodología fue regida para la elaboración del sistema, ya que este sistema estará expuesto a futuros requerimientos donde la flexibilidad y la adaptabilidad nos beneficiaran en gran medida.

### Ventajas:

El análisis del riesgo se hace de forma explícita y clara. Une los mejores elementos de los restantes modelos.

- Reduce riesgos del proyecto
- Incorpora objetivos de calidad
- Integra el desarrollo con el mantenimiento, etc.

Además es posible tener en cuenta mejoras y nuevos requerimientos sin romper con la metodología, ya que este ciclo de vida no es rígido ni estático.

### Desventajas:

- Genera mucho tiempo en el desarrollo del sistema
- Modelo costoso
- Requiere experiencia en la identificación de riesgos

Inconvenientes: Planificar un proyecto con esta metodología es a menudo imposible, debido a la incertidumbre en el número de iteraciones que serán necesarias. En este contexto la evaluación de riesgos es de la mayor importancia y, para grandes proyectos, dicha evaluación requiere la intervención de profesionales de gran experiencia.

El IEEE clasifica al desarrollo en espiral como modelo no operativo en sus clasificaciones de MCV.

### **Metodología de diseño estructurado de Yourdon de Marco.**

Esta metodología proporciona una manera para diseñar paso a paso sistemas y programas detallados. Cabe mencionar que unos pasos involucran el análisis,

otros el desarrollo del diseño y otros más la medición y la mejora de la calidad del diseño. La principal herramienta generada en el diseño estructurado es el “diagrama de estructura” donde muestra los componentes de procedimientos del programa, su ordenación jerárquica y los datos conectados a ellos.

El diagrama de estructura es un diagrama de árbol o jerárquico que, en términos generales, define la arquitectura global de un programa que muestra los procedimientos y sus interrelaciones. En dicho diagrama se utilizan bloques básicos, como son cajas que representan los componentes de procedimientos y las flechas que muestran cómo se conectan.

Pasos a seguir:

- Construir el modelo físico.
- Construir el modelo lógico a partir del modelo físico.
- Construir un modelo lógico del nuevo sistema. En términos generales, es construir una especificación estructurada que incluye los diagramas de flujo de datos, un diccionario de datos y las especificaciones de los procesos.
- Crear una familia con los nuevos modelos físicos.
- Estimar los costos y los tiempos para cada modelo.
- Seleccionar un modelo.
- Empaquetar la especificación en subsistemas.

Y a continuación se ejemplificará el proceso de esta metodología:

Trazar el diagrama de flujo de datos.

El objetivo es representar el problema de diseño como el flujo de datos a través de un sistema. Un sistema se compone de procesos que transforman a los datos. Estos procesos y los datos que los enlazan forman los cimientos para definir los componentes del programa.

Trazar el diagrama de estructura.

En este punto se desea representar el diseño del programa como una jerarquía de componentes de procedimiento. El diagrama de estructura se deriva del diagrama de flujo de datos obtenido previamente. El diseño estructurado proporciona dos estrategias de diseño para guiar la transformación respectiva, las cuales son: los análisis de transformación y los análisis de transacción. Estas dos estrategias nos ayudan a dirigir el diseño jerárquico, así como un proceso paso a paso de transformación por cada estrategia.

- Análisis de transformación. Este modelo de flujo de información divide al diagrama de flujo de datos (DFD) en tres partes: la entrada que recibe el nombre de rama aferente; el proceso lógico llamado transformación central; y la salida, denominada rama eferente.
- Análisis de transacción. Este modelo se utiliza cuando se diseñan programas con proceso de transacciones. El diagrama de estructura general para un programa con procesos de transacciones se representa en la parte superior por el módulo de la transacción central y en la parte inferior hay varios módulos de transacciones para cada tipo distinto de transacción.

Evaluación del diseño.

En este punto la medición de la calidad de diseño es fundamental, para ello se utilizan dos técnicas ya conocidas, como son el acoplamiento y la cohesión.

El acoplamiento mide el grado de independencia entre los componentes de los procedimientos (módulos) en el diagrama de estructura.

La cohesión mide la fuerza de las relaciones entre los elementos dentro de un módulo. Lo ideal es tener un bajo acoplamiento y un alto grado de cohesión.

Preparación del diseño para la implantación.

Esta parte también es conocida como empaquetar el diseño. Empaquetar es el proceso de dividir el diseño del programa lógico en unidades físicas de implantación llamadas unidades de carga. De hecho es un diseño físico del programa.



#### 4.2.1 DIAGRAMA DE FLUJO Y PROCESOS

Una vez analizado el sistema y definido con suficiente claridad los diferentes módulos o interfaces entre el sistema y el ambiente interno de los diferentes departamentos, se procede a determinar los elementos internos y externos que intervienen en el sistema. Es decir, los elementos que el sistema recibe del exterior y la información que se genera a través del sistema o respuesta esperada.

En esta sección se analizarán los diagramas de proceso de datos que no es más que una representación gráfica que permite al analista definir entradas, procedimientos y salidas de la información en la organización permitiendo de ésta manera comprender los procedimientos existentes con la finalidad de optimizarlos reflejándolos en el sistema propuesto.

Por otra parte el diagrama de proceso de datos representa el flujo de información y las transformaciones que se aplican a los datos al moverse desde la entrada hasta la salida del sistema. Además de que proporciona un mecanismo para el modelado funcional así como el modelado del flujo de información.

#### **Diagramas De Contexto**

Un sistema o subsistema, tal como existe dentro de la institución pública, puede ser representado gráficamente de diferentes formas. Los diversos modelos gráficos muestran las fronteras del sistema y la información usada dentro del mismo.

Los componentes se describen a continuación y se muestran en la figura 4.2.1.1

**Procesos.-** Un proceso significa que se realizan algunas acciones o grupo de acciones que modifican los datos.

**Entidad Externa.-** Una entidad es una persona, grupo, departamento o cualquier sistema que recibe u origina información o datos pero que no es parte del sistema.

**Flechas.-** Un flujo de datos muestra que es una información desde o hacia un proceso y conecta entidades.

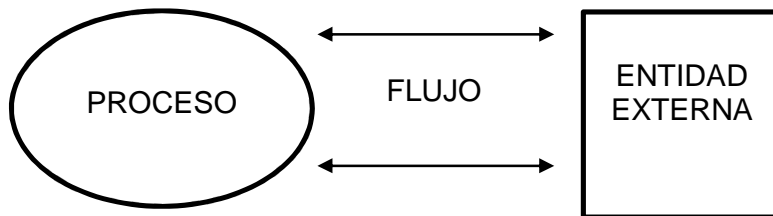


Figura 4.2.1.1 Componentes Básicos para el diagrama de Contexto.

El diagrama de contexto nos permite identificar a las personas o entidades que se comunican con el sistema y recibirá datos del medio ambiente que serán sus entradas necesarias para producir datos que el mundo exterior espera e identificara como salida.

El sistema debe reconocer cada flujo de entrada para identificar que ha ocurrido un evento y cada evento debe generar salidas inmediatas como respuesta o bien almacenar datos, que posteriormente se convertirán en salida o un indicador de un cambio de estado dentro del sistema.

En la figura 4.2.1.2 el diagrama de contexto muestra cómo se conectara nuestro sistema con cada una de las áreas que lo rodea.

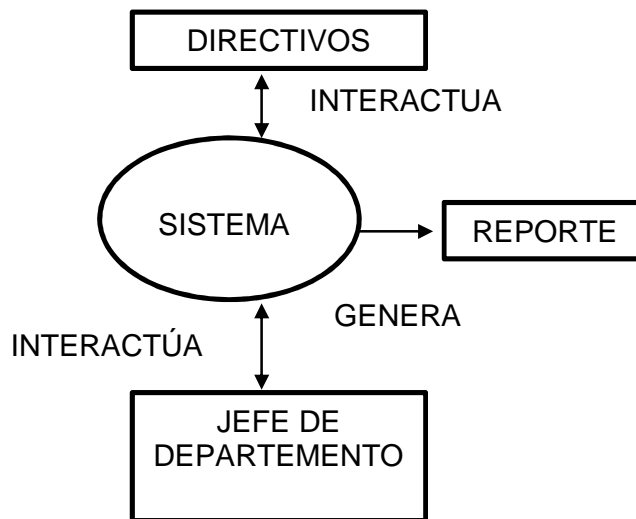


Figura 4.2.1.2 Diagrama de Contexto General a Nivel 0.

En la figura 4.2.1.2 se muestra un panorama de las diferentes entidades que intervienen en el sistema, así como el flujo de información entre ellos, lo cual significa que el sistema de automatización de vales recibirá datos de las entidades, los procesará y emitirá resultados que involucren a otras entidades o a la misma que generó la información.

A continuación se listan algunos de los eventos considerados para el desarrollo del sistema:

- El Jefe de Departamento solicita trámite de alta en el sistema.
- Algún directivo realiza el registro del usuario y productos disponibles en almacén.
- Se realizan reportes puntuales y solicitudes de compra y salida de almacén.

La figura 4.2.1.3 muestra el primer nivel del diagrama de contexto. A través de este nivel podemos visualizar los diferentes módulos que conforman al sistema

y se aprecia que cada módulo atiende una situación determinada y tiene una comunicación con entes bien determinados.

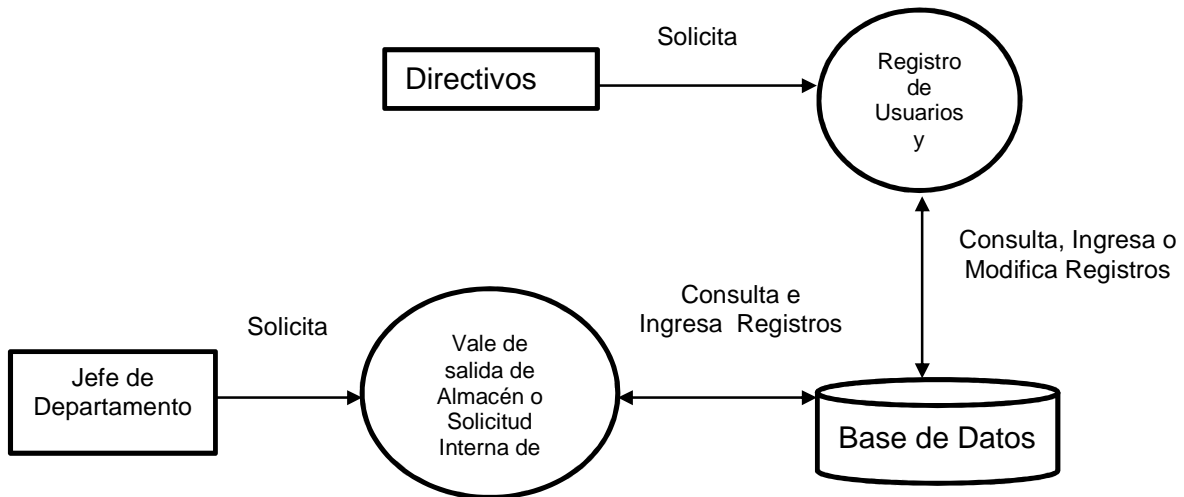


Figura 4.2.1.3 Diagrama de contexto Nivel 1.

En la figura 4.2.1.4 mostramos el proceso que realiza un Jefe de Departamento para solicitar una orden de compra o vale de salida de almacén.

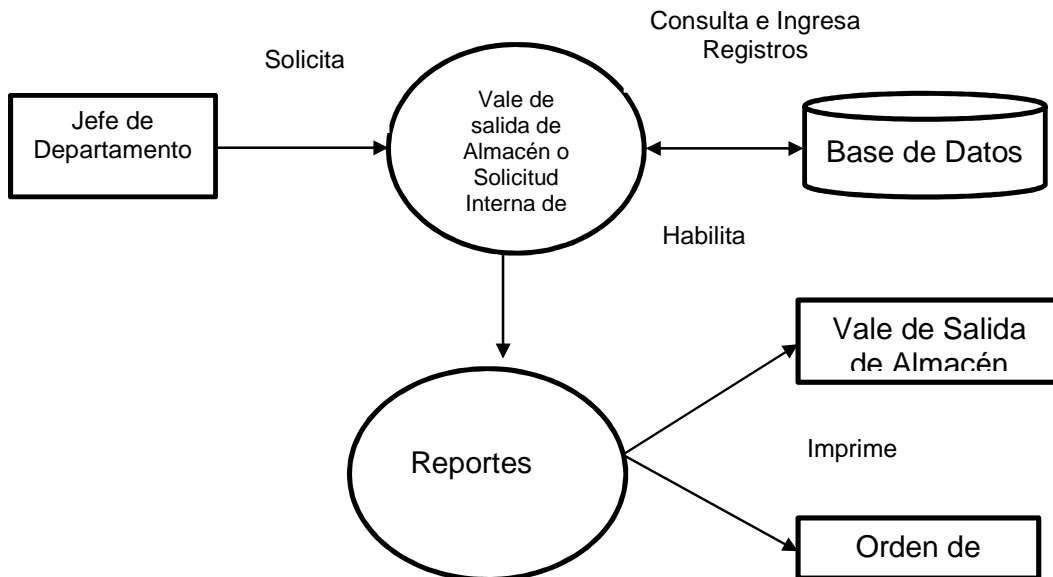


Figura 4.2.1.4 Proceso de solicitud de vale de compra o salida de almacén.

Para que un Jefe de Departamento solicite una orden de compra o vale de salida de almacén realiza los siguientes pasos:

- Ingresa al sistema vía web.
- Escoge la opción deseada
- Registra sus datos.
- Envía su registro a la base de datos.
- Genera los reportes deseados.

Cuando los Jefes de Departamento y productos ya fueron capturados, se procede al registro como lo muestra la figura 4.2.1.5.

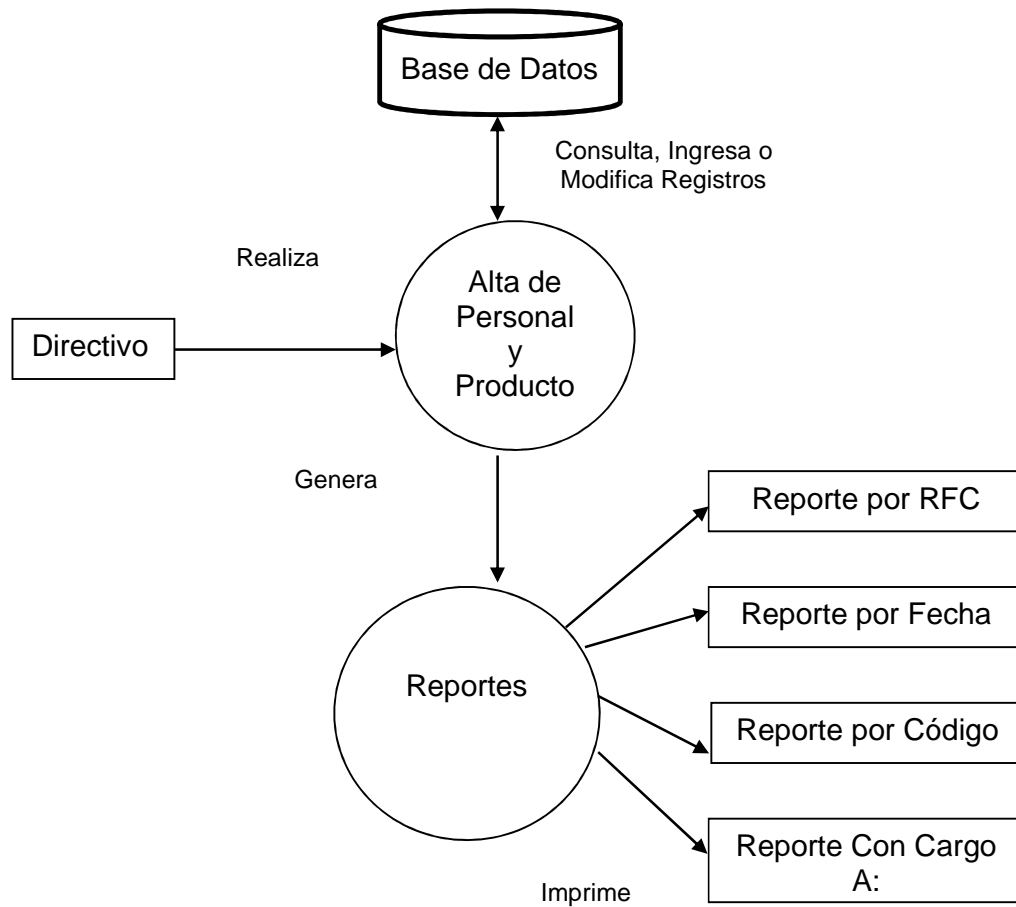


Figura 4.2.1.5 Proceso de registro y generación de reportes. Para el registro de personal y productos disponibles los directivos siguen los siguientes pasos:

- Ingresa al sistema vía web.
- Escoge la opción deseada
- Introduce información necesaria
- Se agrupan datos por categoría
- Genera reportes para control departamental y toma de decisiones.

### **Diagrama De Flujo**

El diagrama de flujo es un esquema para representar gráficamente un algoritmo.

Se basan en la utilización de diversos símbolos para representar operaciones específicas. Se les llama diagramas de flujo porque los símbolos utilizados se conectan por medio de flechas para indicar la secuencia de operación.

Partiendo del menú principal, en la figura 4.2.1.6 se representa gráficamente de forma general los procesos que lleva a cabo nuestro sistema para el alta, y consulta de datos de usuarios y productos dependiendo del tipo de usuario.

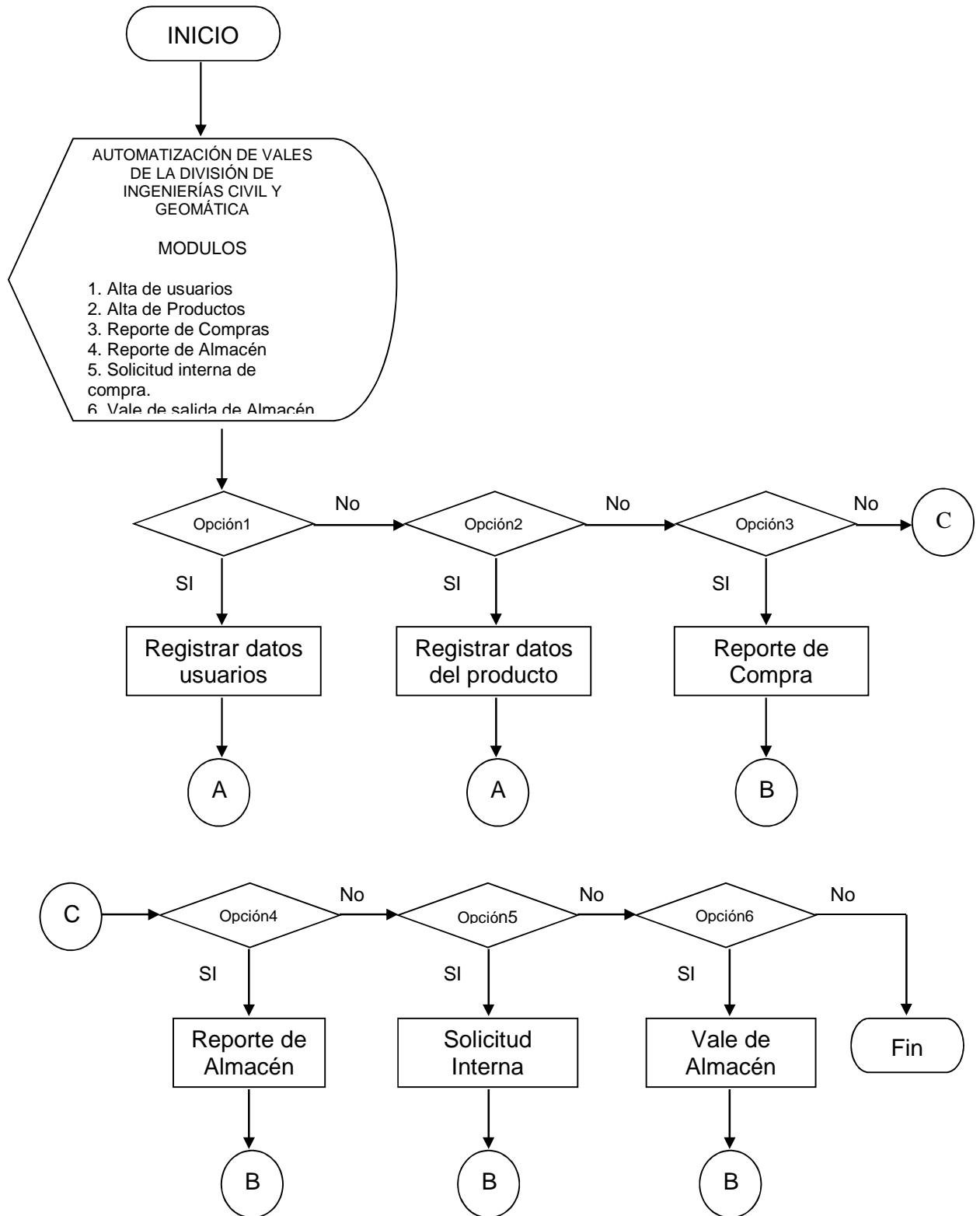


Figura 4.2.1.6 Diagrama de flujo general.

A continuación en las figuras 4.2.1.7 se muestran los pasos que sigue el sistema en ejecución para registrar los datos de un Jefe de Departamento.

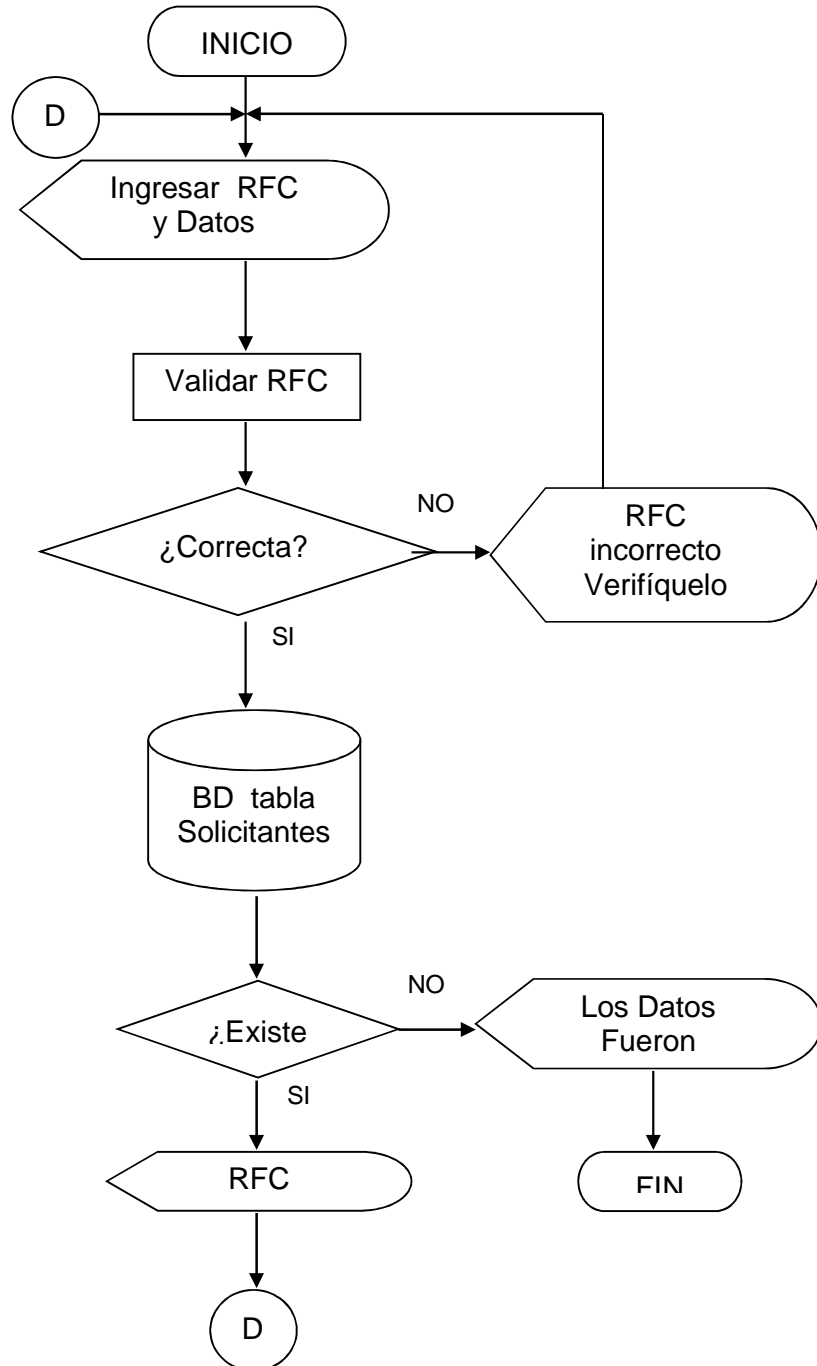


Figura 4.2.1.7. Diagrama de flujo para alta de Jefe de Departamento.



Enseguida mostramos en las figuras 4.2.1.8 el flujo que sigue la información para el registro de productos.

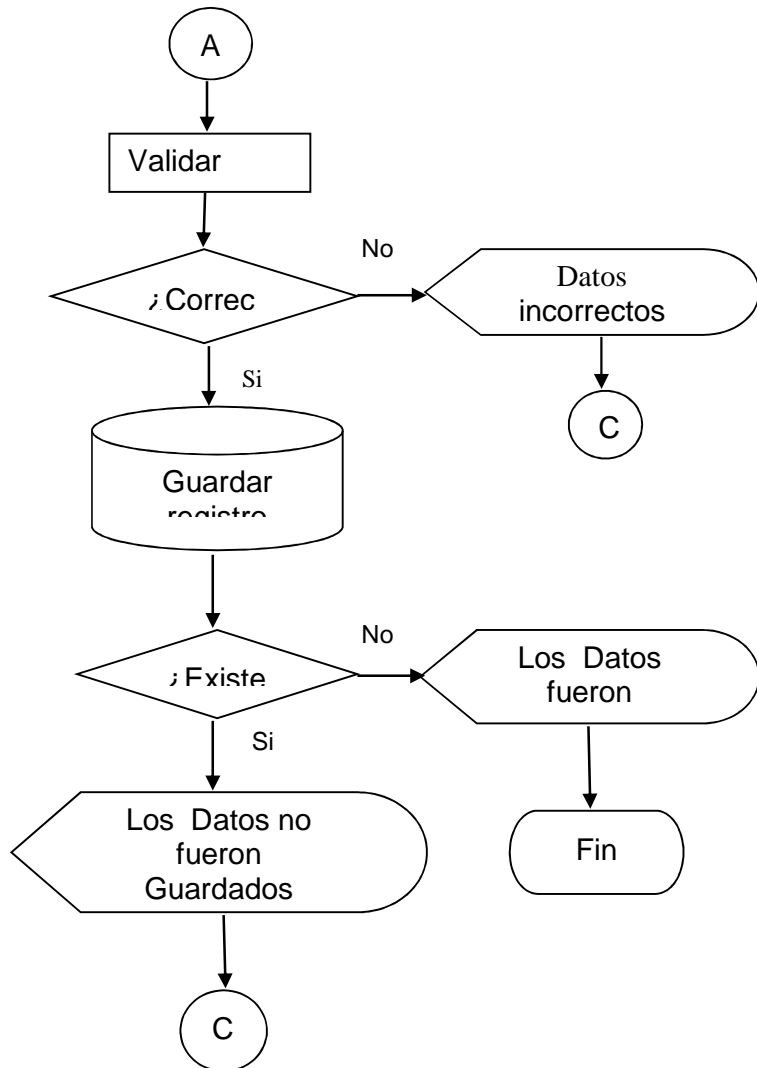
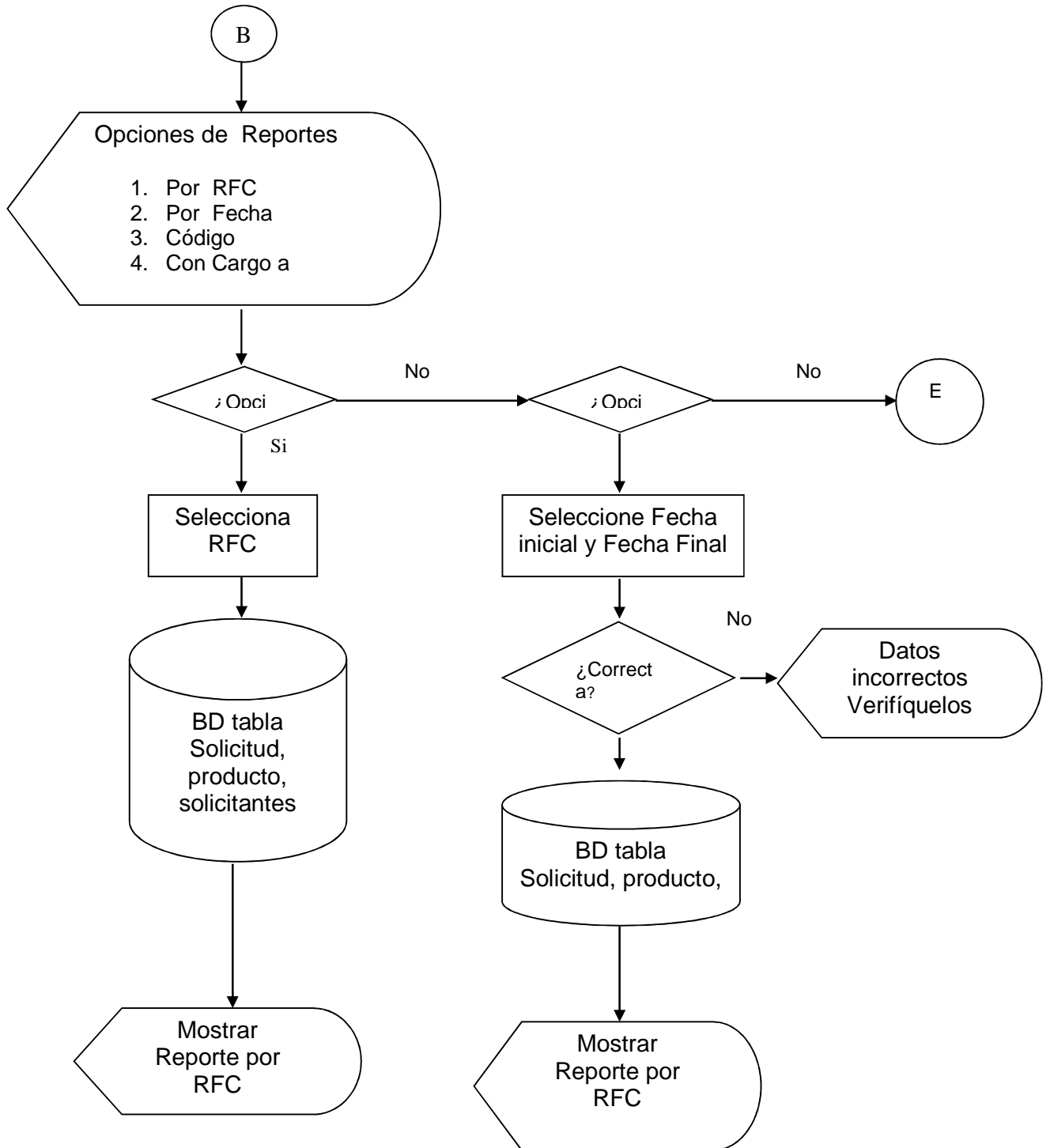


Figura 4.2.1.8. Diagrama de flujo para registro de productos

Por último, en las figuras 4.2.1.9. Se mostrará gráficamente cómo se llevan a cabo los reportes de Compra.



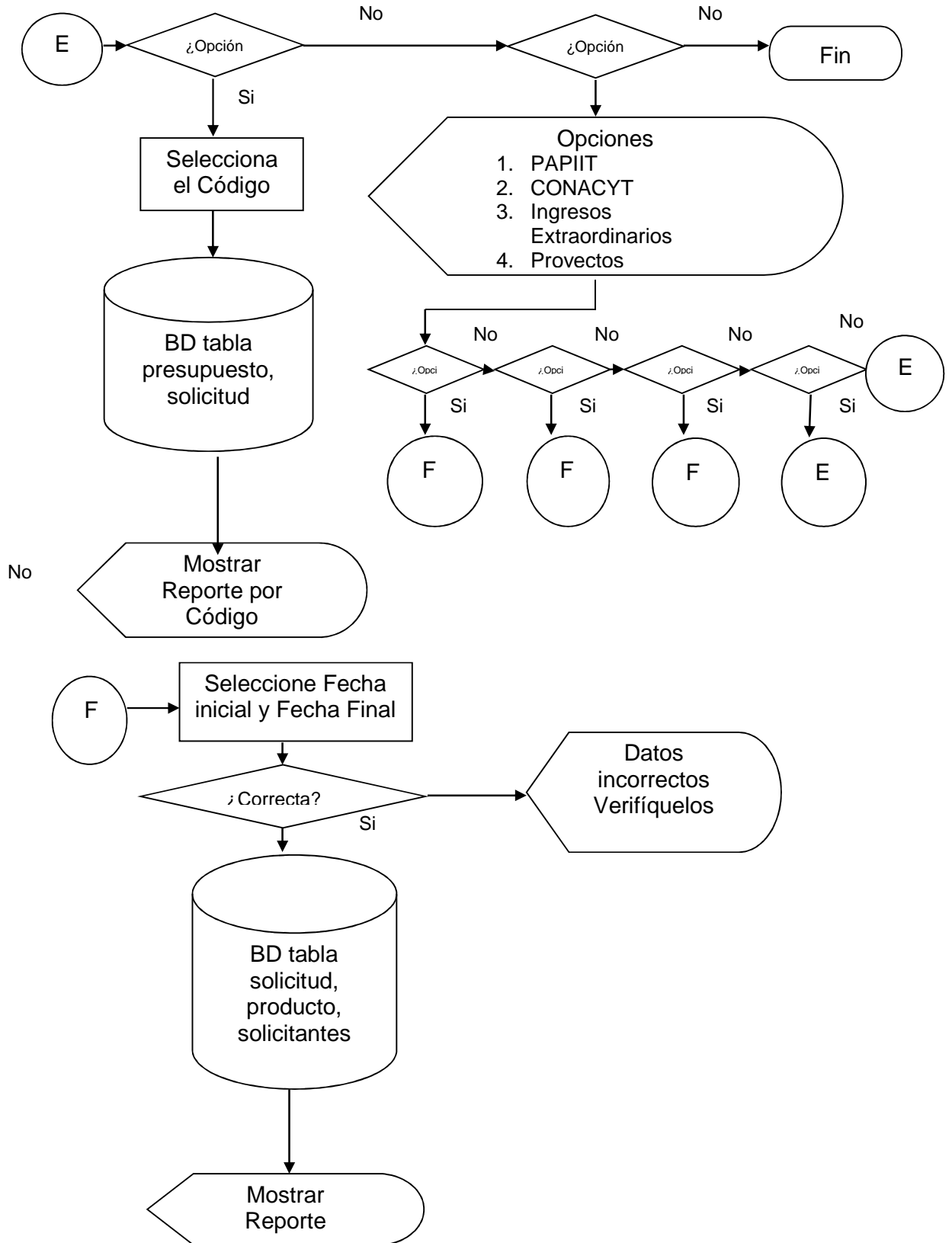


Figura 4.2.1.9. Diagrama de flujo para Reportes de Compra.

## 4.2.2 DIAGRAMACIÓN DE ENTIDAD-RELACIÓN

El diagrama Entidad-Relación (E-R) es un modelo que describe en alto nivel la abstracción de los datos almacenados en un sistema, el cual busca modelar los requerimientos en cuanto al almacenamiento de la información, se basa en una percepción del mundo real, la cual está formada por objetos básicos llamados entidades y las relaciones entre estos objetos, además de las características de estos objetos llamados atributos.

Para ser más preciso: trata sobre modelar los requerimientos de los datos el cual se basa en la funcionalidad deseada para el sistema futuro. Se dice, por tanto que es la representación gráfica de una base de datos por medio de elementos que identifican el problema a resolver.

El generar un diagrama Entidad-Relación es una técnica usada para describir el entendimiento de las necesidades de información que se tiene, está bien definida y ayuda a generar diagramas fáciles de leer y de verificar.

Los objetivos del diagrama Entidad-Relación son los siguientes:

- Describir exactamente la información que requiere un sistema.
- Facilitar la discusión acerca de las entidades a modelar, llegar a un acuerdo
- Ayudar a prevenir errores y evitar malos entendidos con la información.
- Ayuda a generar un documento bien formado del sistema ideal.
- Es la base para el diseño físico de la base de datos.

Los elementos que integran el modelo son:

- **Entidades:** Representan las cosas sobre las cuales se tiene cierto interés en almacenar y éstas pueden ser tangibles (persona o libro) o conceptual (centro de costo). En otras palabras es un objeto que existe y

- se distingue de otros objetos de acuerdo a sus características llamadas atributos.
- **Atributo:** Es una característica que define a la entidad.
- **Instancia:** Es un valor particular que se ajusta a la definición de la entidad. Pueden tomar un conjunto de valores permitidos al que se le conoce como dominio del atributo. Así cada entidad se describe por medio de un conjunto de parejas formadas por el atributo y el valor de dato.
- **Llave primaria:** Identifica en forma única cada instancia de la entidad, por lo que no debe ser nulo y deberá existir en otra entidad como llave foránea.
- **Llave foránea:** Es un identificador de la instancia que determina la relación con otra entidad por medio de su llave primaria, estableciendo de esta forma la asociación entre ellas.
- **Relación:** Es la asociación que existe entre dos a más entidades y la cantidad de entidades en una relación determina el *grado* de la relación.

Lo primero que hay que tener presente a la hora de diseñar una base de datos relacional es el propio concepto de modelo relacional, que organiza los datos en una base de datos como una colección de tablas teniendo presente lo siguiente:

- Cada tabla tiene un nombre que la identifica unívocamente.
- Cada tabla tiene una o más columnas nominadas, que están dispuestas en un orden específico de izquierda a derecha.
- Cada tabla tiene “n” filas, conteniendo cada una un único valor en cada columna. Las filas están desordenadas.

- Todos los valores de una columna determinada tienen el mismo tipo de datos y éstos están extraídos de un conjunto de valores legales llamado el dominio de la columna.

A su vez, las tablas están relacionadas unas con otras por los datos que contienen. El modelo de datos relacional utiliza claves primarias y claves secundarias (externas o foráneas) para representar estas relaciones entre tablas. A la hora de definir las claves primarias y secundarias es necesario tener presente lo siguiente:

- Una clave primaria es una columna o combinación de columnas dentro de una tabla cuyo(s) valor(es) es(son) un valor de clave primaria para alguna otra tabla. Una tabla puede contener más de una clave secundaria, enlazándola a una o más tablas.
- Una combinación clave primaria/clave secundaria crea una relación padre/hijo entre las tablas que las contienen.

Aplicando la teoría antes descrita para los diagramas Entidad-Relación, es posible obtener el siguiente diagrama E-R descrito en la figura 4.2.2.1 en su diseño lógico.

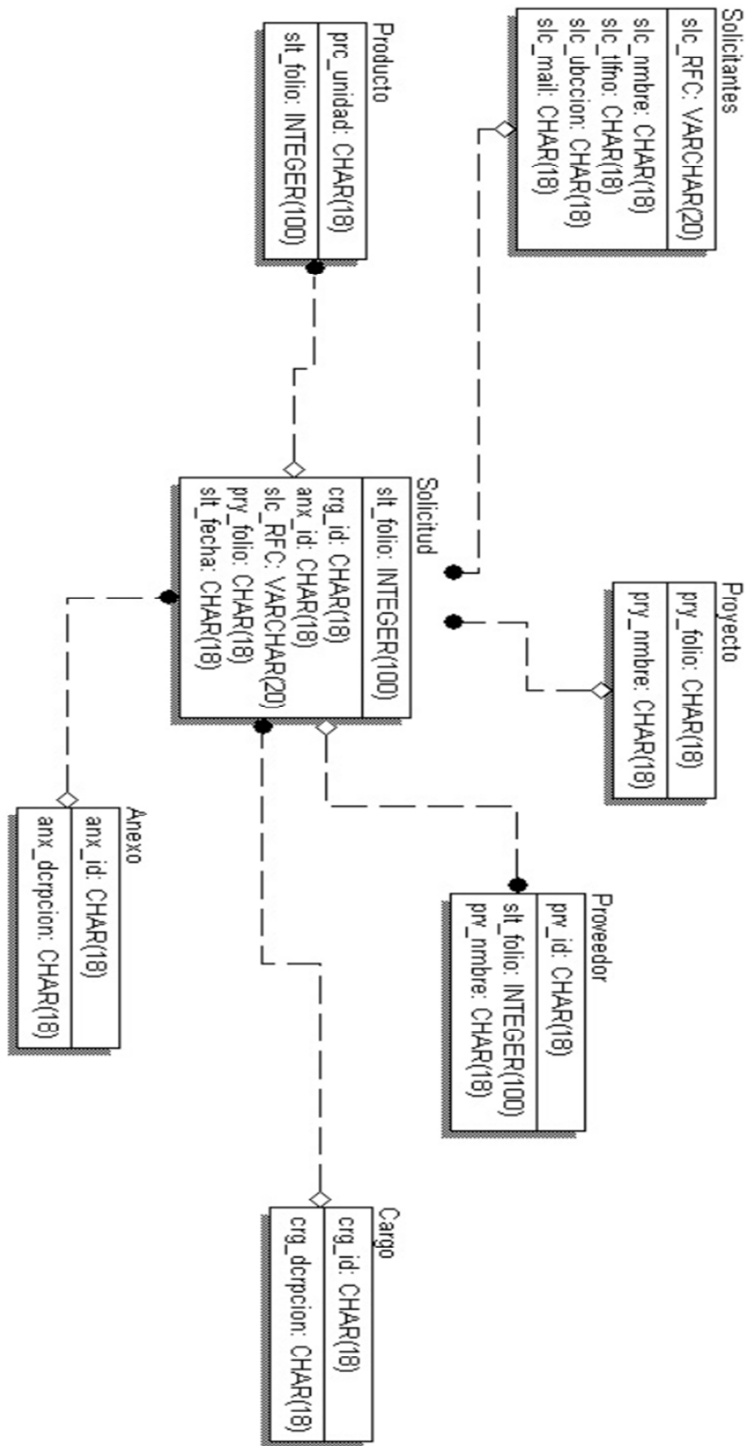


Figura 4.2.2.1 Diagrama Entidad-Relación en su diseño lógico

### 4.2.3 DICCIONARIO DE DATOS

El diccionario de datos es un listado organizado de todos los datos que pertenecen a un sistema. Como su nombre lo sugiere, estos elementos se centran alrededor de los datos y la forma en que están estructurados para satisfacer los requerimientos de los usuarios y las necesidades de la organización.

Sus funciones son:

- Describir el propósito de la Base de Datos.
- Proveer una detallada descripción de cada tabla dentro de la Base de Datos y explicar su propósito.
- Documentar la estructura interna de cada tabla, incluyendo todos sus campos y sus tipos de datos con comentarios, todos los índices y todas las vistas.
- Describir reglas como pueden ser el uso de valores no nulos y únicos.

El contenido de un registro del diccionario, es un conjunto de columnas que contienen:

- Tipo de dato de la columna.
- Acrónimo.
- Longitud del tipo de dato.
- Llave que pueden ser primaria o foránea.
- Acepta nulos.
- Tablas con las que se asocia.
- Tipo de dato
- Descripción.

A continuación se muestran las tablas que conforman el sistema de vales.



Tabla Solicitantes

Nombre de Columna	Tipo de dato y longitud	Opción de columna vacía	Llave	Descripción
slc_RFC	char(15)	No	Primaria	RFC del solicitante
slc_nombre	varchar(50)	Si	No	Nombre del solicitante
slc_dpto	varchar(80)	No	No	Departamento del solicitante
slc_cdgo	int(4)	No	No	Código del departamento
slc_ubccion	char(35)	Si	No	Ubicación del solicitante
slc_tlfno	char(10)	Si	No	Teléfono del solicitante
slc_mail	varchar(35)	Si	No	E-mail del solicitante
slc_pass	varchar(15)	No	No	Password del solicitante
slc_fltro	int(2)	Si	No	Tipo de usuario

Figura. 4.2.3.1

Tabla Solicitud

Nombre de Columna	Tipo de dato y longitud	Opción de columna vacía	Llave	Descripción
slt_folio	int(10)	No	Primaria	Folio de la solicitud
crg_id	char(18)	Si	Foránea	Identificador de cargo
anx_id	int(1)	Si	Foránea	Identificador de anexo
slc_RFC	char(18)	Si	Foránea	RFC del solicitante
pry_folio	int(10)	Si	Foránea	Folio del proyecto
slt_fecha	date	Si	No	Fecha de la solicitud
slt_obsrv	varchar(250)	Si	No	Observaciones de la solicitud
prspsto_id	int(4)	Si	No	Presupuesto de la solicitud

Figura. 4.2.3.2

Tabla Salidaslc

Nombre de Columna	Tipo de dato y longitud	Opción de columna vacía	Llave	Descripción
sld_folio	int(10)	No	Primaria	Folio de salida
slc_RFC	char(18)	Si	Foránea	RFC del solicitante
sld_fecha	date	Si	No	Fecha de salida
prspsto_id	int(4)	Si	No	Presupuesto de la solicitud

Figura. 4.2.3.3

Tabla Proyecto

Nombre de Columna	Tipo de dato y longitud	Opción de columna vacía	Llave	Descripción
pry_folio	int(10)	No	Primaria	Folio del proyecto
pry_nombre	varchar(30)	No	No	Nombre del proyecto

Figura. 4.2.3.4

Tabla Proveedor

Nombre de Columna	Tipo de dato y longitud	Opción de columna vacía	Llave	Descripción
prv_id	int(18)	No	Primaria	Identificador de Proveedor
slt_folio	int(10)	Si	Foránea	Folio de la solicitud
prv_nombre	char(18)	Si	No	Nombre del proveedor

Figura. 4.2.3.5

Tabla Producto

Nombre de Columna	Tipo de dato y longitud	Opción de columna vacía	Llave	Descripción
prc_id	int(10)	No	Primaria	Identificador de producto
prc_unidad	char(18)	No	No	Tipo unidad del producto
slt_folio	int(10)	Si	Foránea	Folio de solicitud
prc_cntdad	int(5)	No	No	Cantidad de producto
prc_dscrip	varchar(50)	No	No	Descripción del producto
prc_prcunidad	int(11)	No	No	Precio de unidad
prc_imprte	int(11)	No	No	Importe del producto
prc_total	int(11)	No	No	Suma total de productos

Figura. 4.2.3.6

Tabla Presupuesto

Nombre de Columna	Tipo de dato y longitud	Opción de columna vacía	Llave	Descripción
prspsto_id	int(4)	No	Primaria	Identificador de presupuesto
prspsto_nombre	varchar(35)	No	No	Nombre de presupuesto

Figura. 4.2.3.7

Tabla Prdsalida

Nombre de Columna	Tipo de dato y longitud	Opción de columna vacía	Llave	Descripción
prd_id	int(18)	No	Primaria	Folio salida
prd_obsrv	varchar(30)	No	No	Observaciones
prd_cntdad	int(5)	No	No	Cantidad
ctl_id	varchar(10)	No	No	Identificador del producto
sld_folio	int(10)	No	No	Folio de salida

Figura. 4.2.3.8

Tabla Catalogo

Nombre de Columna	Tipo de dato y longitud	Opción de columna vacía	Llave	Descripción
ctl_id	varchar(10)	No	Primaria	Identificador del producto
ctl_nombre	varchar(60)	No	No	descripción
ctl_undad	char(10)	No	No	Tipo unidad del producto
ctl_filtro	int(2)	No	No	Filtro

Figura. 4.2.3.9

Tabla Cargo

Nombre de Columna	Tipo de dato y longitud	Opción de columna vacía	Llave	Descripción
crg_id	int(1)	No	Primaria	Identificador de cargo
crg_dcrpcion	char(18)	Si	No	Descripción de cargo

Figura. 4.2.3.10

Tabla Anexo

Nombre de Columna	Tipo de dato y longitud	Opción de columna vacía	Llave	Descripción
anx_id	int(1)	No	Primaria	Identificador de anexo
anx_dcrpcion	char(18)	Si	No	Descripción de anexo

Figura. 4.2.3.11

#### 4.2.4 NORMALIZACIÓN

La normalización es un concepto aplicado a una base de datos relacional, es el proceso mediante el cual se transforman datos complejos a un conjunto de estructuras de datos más pequeñas, que además de ser más simples y estables, son fáciles de mantener; contribuyen a minimizar los problemas de lógica, cada regla está basada en la que le antecede.

Las ventajas que se obtienen al aplicar la normalización son:

Facilidad de uso. Los datos se agrupan en tablas que identifican visiblemente un objeto o una relación.

Flexibilidad. La información necesaria puede obtenerse de las tablas relacionales o relaciones mediante las operaciones de álgebra relacional.

Precisión. Las interrelaciones entre tablas mantienen información diferente relacionada con toda exactitud.

Seguridad. Los controles de acceso para consultar o actualizar información son sencillos de implementar.

Implementación. Las tablas se almacenan físicamente como archivos planos.

Independencia. Los programas no están ligados a las estructuras, pudiendo incrementar la base de datos añadiendo nuevos atributos o nuevas tablas sin afectar los programas que las usan.

Claridad. La representación de la información en tablas simples es clara y sencilla para el usuario.

Gestión. Los lenguajes manipulan la información de manera sencilla al estar los datos basados en el álgebra y el cálculo relacional.

Redundancia. La información no se duplica dentro de las estructuras.

Rendimiento. Se trata sólo la información que se utilizará en cada aplicación.

En la Fig. 4.2.4.1 se muestra la tabla sin normalizar, posteriormente se irá aplicando la normalización.

## IMPLEMENTACIÓN Y PERSONALIZACIÓN DEL SISTEMA

slc_rfc	slc_nombre	slc_dpto	slc_codigo	slc_ubacion
CAFF830622	Fredy Cardea Flores	Depto. de Computo	0	Ciilo 37
SQLR4704047XA	Ing. Rodolfo Solis Ubaldo	Jefe de la Division de Ingenieria Civil y Geomatica	2200	Edificio Direccion
SQLR4704047XA	Ing. Rodolfo Solis Ubaldo	Jefe de la Division de Ingenieria Civil y Geomatica	2200	Edificio Direccion
AETA820507685	Ing. Tanya Itzel Arteaga Ricci	Jefa del Depto. de Computo	2200	Cb. 37 P.B.
MERM7208318F6	Ml. Marco Tulio Mendoza Rosas	Jefe del Depto. de Construccion	2233	7 P.B.
CEGA7805253E8	Arturo Olvera Gorrez	Secretaria Academica	0	Jefatura
EASO620919JK9	Dr. Carlos Agustin Escalante Sandoval	Coordinador de Posgrado	2220	Posgrado 1er. Piso
TEHM600926FE2	Ing. Marcos Trejo Hernandez	Secretario Academico	2201	139 1er. Piso
AETA820507685	Ing. Tanya Itzel Arteaga Ricci	Jefa del Depto. de Computo	2200	Cb. 37 P.B.
EASO620919JK9	Dr. Carlos Agustin Escalante Sandoval	Coordinador de Posgrado	2220	Posgrado 1er. Piso
CAFF830622	Fredy Cardea Flores	Depto. de Computo	0	Ciilo 37

slc_tlfno	slc_mail	slc_pass	slc_filtro	prc_id	prc_unidad	slt_folio	prc_cantidad
5522438593	lebensraum69@gmail.com	ragnarok	3	2	GARRAFA	1	4
56160558	rsolisub@cancun.fi-a.unam.mx	RSolis	2	6	PZA	3	2
56160558	rsolisub@cancun.fi-a.unam.mx	RSolis	2	2	GARRAFA	1	4
56228012	tanya@yahoo.com.mx	tricoli	2	2	GARRAFA	1	4
56228004	rockmar@servidor.unam.mx	MTulio	1	2	GARRAFA	1	4
55507709	arturo_olveragomez@yahoo.com.mx	danika	3	4	PZA	2	2
56228011	caes@servidor.unam.mx	CEscalante	1	3	PZA	1	1
56228006	wolftwain@yahoo.com	MTrejo	2	1	PZA	1	2
56228012	tanya@yahoo.com.mx	tricoli	2	8	PZA	5	1
56228011	caes@servidor.unam.mx	CEscalante	1	8	PZA	5	1
5522438593	lebensraum69@gmail.com	ragnarok	3	1	PZA	1	2

prc_descrip	prc_prcunidad	prc_impnte	prc_total	prv_id	slt_folio	prv_nombre	prspsto_id	prspsto_nombre	org_id	org_descripion
ACIDO	456	1824	2148	1	1	DEPOT	2210	Secretaria Tecnica	3	ingresos ext.
MIN-LAPS	5550	11100	11100	2	1	OFFICE	2210	Secretaria Tecnica	0	presupuesto
ACIDO	456	1824	2148	5	2	DEPOT	2210	Secretaria Tecnica	1	papiit
ACIDO	456	1824	2148	3	2	OFFICE	2235	Departamento de Sanitaria	1	papiit
ACIDO	456	1824	2148	7	4	DEPOT	2220	Posgrado	3	ingresos ext.
LAP-TOPS	5350	10700	57590	7	4	DEPOT	2200	Jefatura	0	presupuesto
COXA	78	78	2148	3	2	OFFICE	2235	Departamento de Sanitaria	1	papiit
POST-IT	123	246	2148	1	1	DEPOT	2236	Departamento de Sistemas	1	papiit
O	1	1	1	8	5	DEPOT	2200	Jefatura	3	ingresos ext.
O	1	1	1	8	5	DEPOT	2200	Jefatura	3	ingresos ext.
POST-IT	123	246	2148	1	1	DEPOT	2210	Secretaria Tecnica	4	otros

Figura. 4.2.4.1 Tabla sin normalizar

### Primera Forma Normal (1FN)

Una tabla está en Primera Forma Normal si y sólo si:

- Todos los atributos son indivisibles, mínimos.
- La tabla contiene una llave primaria.
- La tabla no contiene atributos nulos.

Una columna no puede tener múltiples valores por lo que se eliminan valores repetidos dentro de una BD. La tabla sin normalización de la Fig. 4.2.4.1 quedará conformada en dos tablas que llamaremos Solicitantes (ver tabla 4.2.4.2) y Requisición (ver tabla 4.2.4.3).

## NORMALIZACIÓN

slc_rfc	slc_nombre	slc_dpto	slc_edgo	slc_ubccion	slc_tlfno
CAFF830622	Fredy Cardea Flores	Depto. de Computo	0	Cbilo 37	5522438593
SOUR4704047XA	Ing. Rodolfo Solis Ubaldo	Jefe de la Division de Ingenieria Civil y Geomatica	2200	Edificio Direccion	56160558
TEHM600926RE2	Ing. Marcos Trejo Hernandez	Secretario Academico	2201	139 1er. Piso	56228006
MERM7208318F6	M.I. Marco Tulio Mendoza Rosas	Jefe del Depto. de Construccion	2233	7 P.B.	56228004
EASO620919JK9	Dr. Carlos Agustin Escalante Sandoval	Coordinador de Posgrado	2220	Posgrado 1er. Piso	56228011
OEGA7805253E8	Arturo Olvera Gomez	Secretaria Academica	0	Jefatura	55507709
AETA820507685	Ing. Tanya Itzel Arteaga Ricci	Jefa del Depto. de Computo	2200	Cb. 37 P.B.	56228012

slc_mail	slc_pass	slc_filtro
lebensraum69@gmail.com	ragnarok	3
rsolisub@cancun.fi-a.unam.mx	RSolis	2
wolftwain@yahoo.com	MTrejo	2
rockmar@servidor.unam.mx	MTulio	1
caes@servidor.unam.mx	CEscalante	1
arturo_olveragomez@yahoo.com.mx	danka	3
tanya@yahoo.com.mx	tricci	2

Figura. 4.2.4.2 Solicitantes

slt_folio	prc_id	prc_unidad	slt_folio	prc_cantidad	prc_descrip	prc_prcunidad	prc_imperte	prc_total
1	2	GARRAFA	1	4	ACIDO	456	1824	2148
3	6	PZA	3	2	MINI-LAPS	5550	11100	11100
1	2	GARRAFA	1	4	ACIDO	456	1824	2148
1	2	GARRAFA	1	4	ACIDO	456	1824	2148
1	2	GARRAFA	1	4	ACIDO	456	1824	2148
2	4	PZA	2	2	LAP-TOPS	5350	10700	57590
1	3	PZA	1	1	COSA	78	78	2148
1	1	PZA	1	2	POST-IT	123	246	2148
5	8	PZA	5	1	O	1	1	1
5	8	PZA	5	1	O	1	1	1
1	1	PZA	1	2	POST-IT	123	246	2148

prv_id	slt_folio	prv_nombre	prspsto_id	prspsto_nombre	crge_id	crge_descripcion
1	1	DEPOT	2210	Secretaria Tecnica	3	ingresos ext.
2	1	OFFICE	2210	Secretaria Tecnica	0	presupuesto
5	2	DEPOT	2210	Secretaria Tecnica	1	papiit
3	2	OFFICE	2235	Departamento de Sanitaria	1	papiit
7	4	DEPOT	2220	Posgrado	3	ingresos ext.
7	4	DEPOT	2200	Jefatura	0	presupuesto
3	2	OFFICE	2235	Departamento de Sanitaria	1	papiit
1	1	DEPOT	2236	Departamento de Sistemas	1	papiit
8	5	DEPOT	2200	Jefatura	3	ingresos ext.
8	5	DEPOT	2200	Jefatura	3	ingresos ext.
1	1	DEPOT	2210	Secretaria Tecnica	4	otros

Figura. 4.2.4.3 Requisición

### Segunda Forma Normal (2FN)

Una relación está en esta forma si está en 1FN y si los atributos que no forman parte de ninguna llave dependen de forma completa de la llave principal. En esta regla todas las dependencias parciales deben eliminarse y separarse dentro de sus propias tablas. Una dependencia parcial es un término que describe aquellos datos que no dependen de la llave primaria para

## IMPLEMENTACIÓN Y PERSONALIZACIÓN DEL SISTEMA

identificarlos. La tabla 4.2.4.3 Requisición no se encuentra en 2FN, ya que las columnas referentes al presupuesto son independientes del proveedor, por lo tanto eliminaremos esas columnas y crearemos la tabla de Formato (ver tabla 4.2.4.5).

prc_id	prc_unidad	sit_folio	prc_cntdad	prc_dscrip	prc_prcunidad	prc_imppte	prc_total
1	PZA	1	2	POST-IT	123	246	2148
2	GARRAFA	1	4	ACIDO	456	1824	2148
3	PZA	1	1	COSA	78	78	2148
4	PZA	2	2	LAP-TOPS	5350	10700	57590
5	PZA	2	3	COMPUTADORAS DE ESCRITORIO	15630	46890	57590
6	PZA	3	2	MINI-LAPS	5550	11100	11100
7	PZA	4	4	OIU	9	36	36
8	PZA	5	1	O	1	1	1

Figura. 4.2.4.4 Productos

prv_id	sit_folio	prv_nombre	prspsto_id	prspsto_nombre	crg_id	crg_dscrpcion
1	1	DEPOT	2210	Secretaria Tecnica	3	ingresos ext.
2	1	OFFICE	2210	Secretaria Tecnica	0	presupuesto
5	2	DEPOT	2210	Secretaria Tecnica	1	papiit
3	2	OFFICE	2235	Departamento de Sanitaria	1	papiit
7	4	DEPOT	2220	Posgrado	3	ingresos ext.
7	4	DEPOT	2200	Jefatura	0	presupuesto
3	2	OFFICE	2235	Departamento de Sanitaria	1	papiit
1	1	DEPOT	2236	Departamento de Sistemas	1	papiit
8	5	DEPOT	2200	Jefatura	3	ingresos ext.
8	5	DEPOT	2200	Jefatura	3	ingresos ext.
1	1	DEPOT	2210	Secretaria Tecnica	4	otros

Figura. 4.2.4.5 Formato

### Tercera Forma Normal (3FN)

La tabla se encuentra en esta forma si está 2FN y todas las columnas que no son llave son funcionalmente dependientes por completo de la llave primaria y no hay dependencias transitivas. Una dependencia transitiva es aquella en la cual existen columnas que no son llaves que dependen de otras columnas que tampoco son llaves. Las tabla solicitantes (ver figura 4.2.4.2) y Producto (ver figura 4.2.4.4) se encuentra en 3FN, sin embargo la tabla Formato (ver figura 4.2.4.5) no lo está, ya que el nombre del presupuesto, la descripción del cargo, son independiente del identificador de proveedor. Para normalizarla se crean las tablas Presupuesto, Cargo y Proveedor (ver figuras 4.2.4.5, 4.2.4.6 y 4.2.4.7).



prspsto_id	prspsto_nombre
2210	Secretaría Técnica
2201	Secretaría Académica
2232	Departamento de Hidráulica
2241	Departamento de Topografía
2234	Departamento de Geotecnia
2231	Departamento de Estructuras
2233	Departamento de Construcción
2235	Departamento de Sanitaria
2236	Departamento de Sistemas
2220	Posgrado
2240	Departamento de Geomatica
2207	Laboratorio de Sanitaria
2200	Jefatura

Figura. 4.2.4.6 Presupuesto

crg_id	crg_descpcion
0	presupuesto
1	papiit
2	conacyt
3	ingresos ext.
4	otros

Figura. 4.2.4.7 Cargo

prv_id	slt_folio	prv_nombre
1	1	DEPOT
2	1	OFFICE
3	2	OFFICE
4	2	WAL-MART
5	2	DEPOT
6	3	DEPOT
7	4	DEPOT
8	5	DEPOT

Figura.4.2.4.8 Proveedor

### 4.3 DISEÑO Y CONSTRUCCIÓN DEL BACK END

Para el sistema de vales se utilizará como back end el manejador de bases de datos MySQL. Este se ejecuta en diferentes plataformas, y los códigos binarios están disponibles para la mayoría de ellas. Los códigos binarios son el resultado de la compilación del código fuente. Esta es la forma más fácil de adquirir MySQL.

En Debian con el comando apt-get podemos descargar e instalar MySQL:

```
$ apt-get install mysql-client mysql-server
```

La primera vez que se inicia MySQL se entra con el usuario de root sin contraseña, posteriormente podemos asignarla. Procedemos a crear un usuario de consulta a la base de datos, de la siguiente manera:

```
$ mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 20
Server version: 5.0.67 Source distribution
mysql> CREATE USER 'prueba'@'localhost' IDENTIFIED BY 'prueba';
Query OK, 0 rows affected (0,00 sec)
```

Le asignamos permisos de lectura y escritura:

```
mysql> GRANT SELECT, INSERT ON test.* TO 'prueba'@'localhost'  
IDENTIFIED BY 'prueba';
```

Una vez instalado MySQL, podemos hacer uso del paquete de herramientas gráficas para administración, así como para hacer consultas a la base de datos.

### Creación de la base de datos en MySQL

Para esta tarea se utiliza el comando `CREATE DATABASE`. Vea fig. 4.3.1

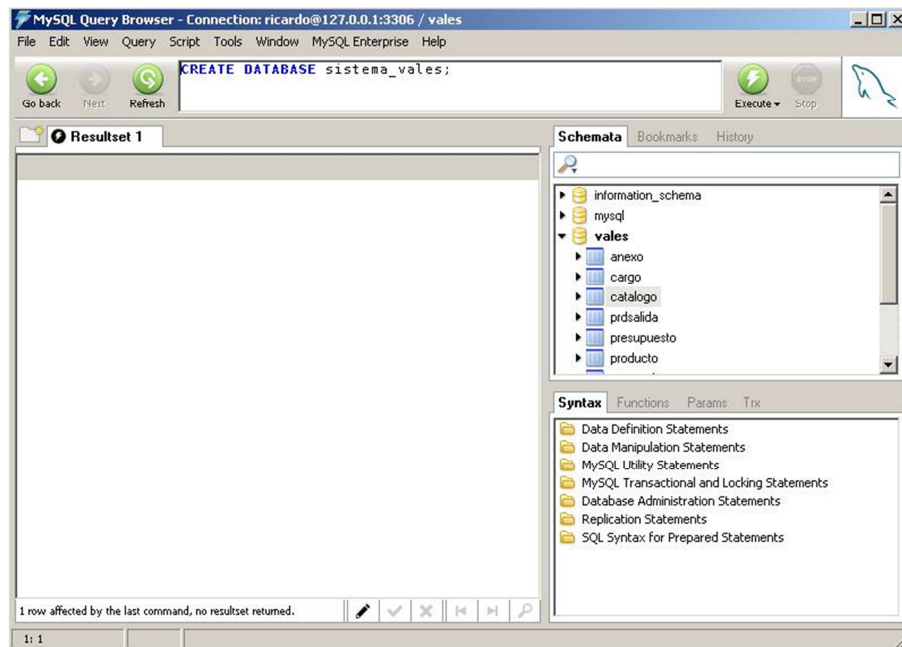


Figura. 4.3.1 Creación de la base de datos

Una vez creada exitosamente la base de datos, se puede comenzar a usar con el comando `USE`. Vea fig. 4.3.2

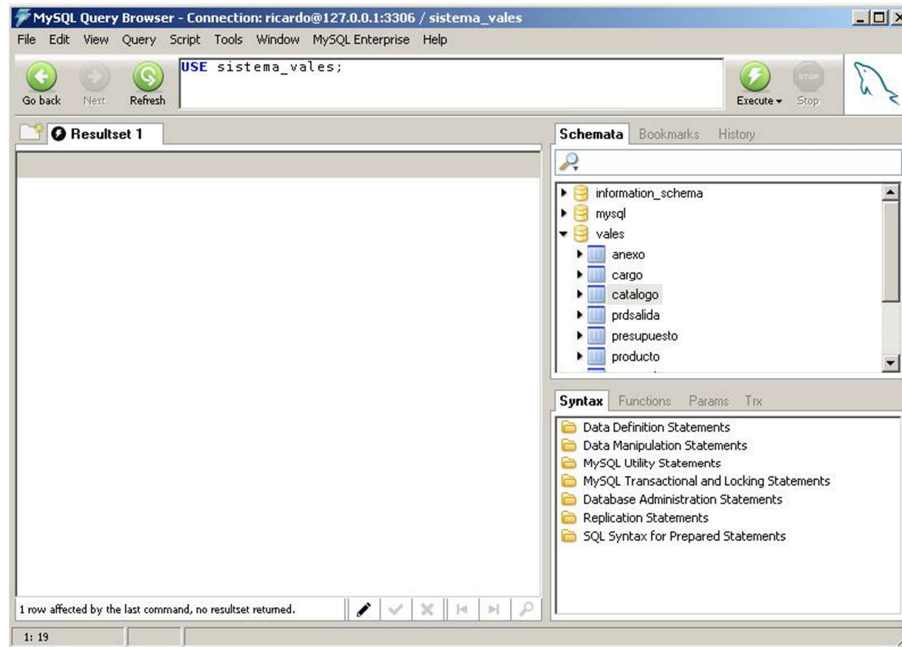


Figura. 4.3.2 Selección de la base de datos a usar

## Creación de tablas en MySQL

Las tablas se crean con el comando `CREATE TABLE` seguido del nombre de la tabla, en la sentencia se describen los tipos de datos en cada campo, la llave primaria y foránea si es el caso, así como si el campo soportará valores nulos.

La sentencia para crear la tabla “salidaslc” es la siguiente:

```
mysql> CREATE TABLE `salidaslc` (
-> `sld_folio` int(10) NOT NULL auto_increment,
-> `slc_RFC` char(18) default NULL,
-> `sld_fecha` date default NULL,
-> `prspsto_id` int(4) default NULL,
-> PRIMARY KEY (`sld_folio`),
-> KEY `slc_RFC` (`slc_RFC`));
```

Puede usarse la herramienta gráfica y seleccionar los tipos de datos para cada campo así como si estos son llave o no. Vea fig. 4.3.3

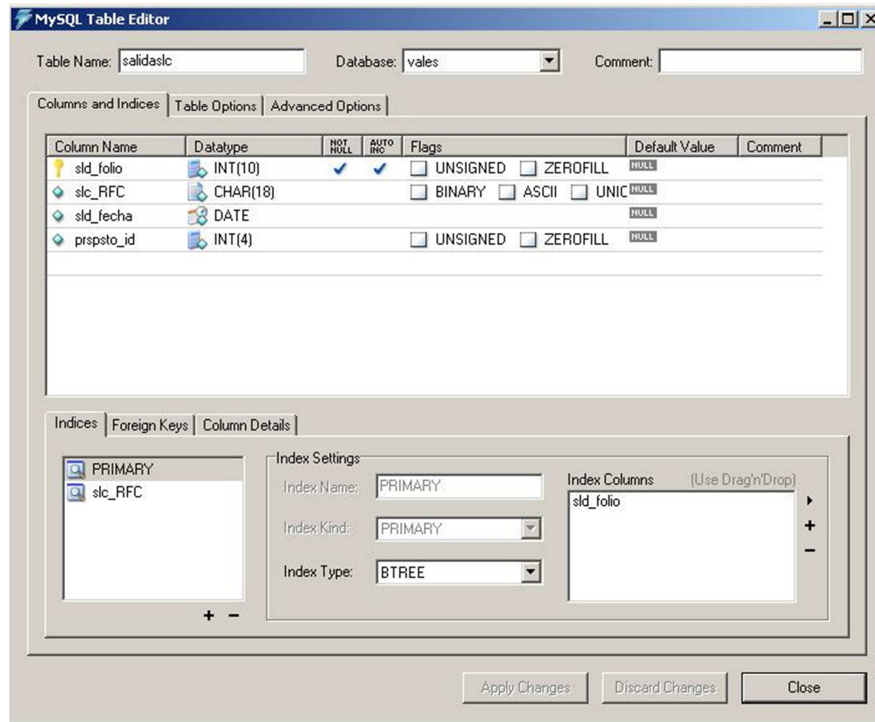


Figura. 4.3.3 Creación de tablas con MySQL Query Browser

Para verificar que se ha creado exitosamente podemos teclear el comando `DESCRIBE <tabla>`. Se mostrará que la tabla ha sido creada, así como una descripción de sus campos y tipos de datos.

```
mysql> DESCRIBE salidaslc;
+-----+-----+-----+-----+-----+-----+
+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
+
| sld_folio     | int(10)      | NO   | PRI | NULL    | auto_increment |
| slc_RFC       | char(18)     | YES  | MUL | NULL    |                |
| sld_fecha     | date         | YES  |     | NULL    |                |
| prspsto_id    | int(4)       | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
+
4 rows in set (0.05 sec)
```

Así mismo con la interfaz gráfica: Vea fig. 4.3.4

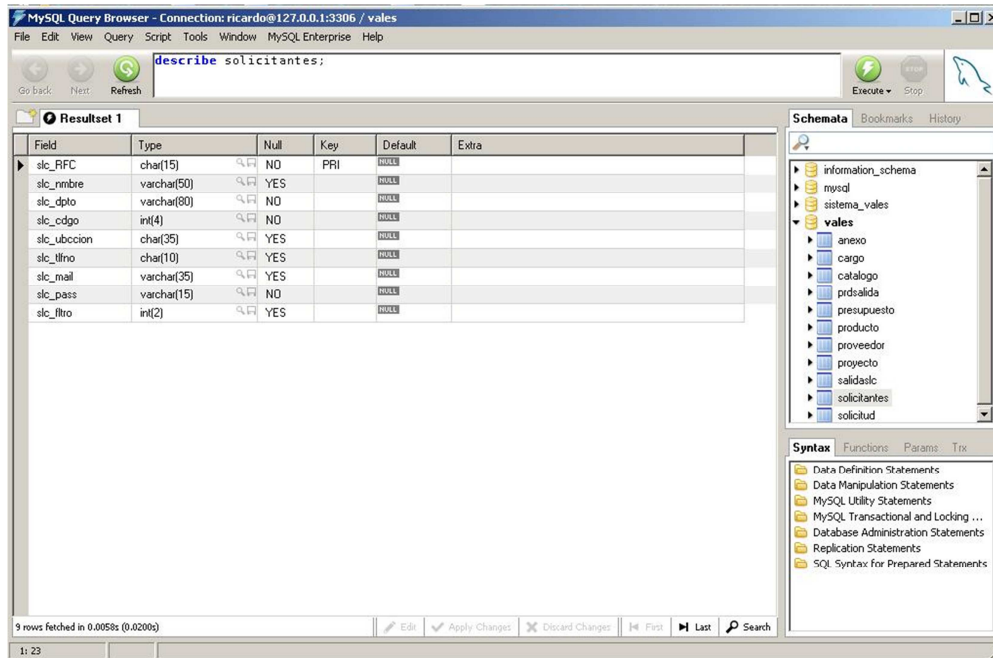


Figura. 4.3.4 Descripción de tabla solicitantes

A continuación se muestra el árbol de tablas de la base de datos, usando el comando show tables. Y se muestra con la interfaz gráfica. Vea figura 4.3.5

```
mysql> SHOW TABLES FROM vales;
```

```
+-----+
| Tables_in_vales |
+-----+
| anexo            |
| cargo           |
| catalogo        |
| prdsalida       |
| presupuesto     |
| producto        |
| proveedor       |
| proyecto        |
| salidaslc       |
| solicitantes    |
| solicitud       |
+-----+
```

```
11 rows in set (0.00 sec)
```

## IMPLEMENTACIÓN Y PERSONALIZACIÓN DEL SISTEMA



Figura. 4.3.5 Árbol de tablas

A continuación se muestra el diagrama del tamaño de las tablas por medio de la interfaz gráfica de MySQL. Vea. fig. 4.3.6

The screenshot shows the 'vales' schema table details in MySQL Administrator. The table is titled 'vales' and contains 11 tables. The data is as follows:

Table Name	Engine	Rows	Data len...	Index length	Update time
proyecto	MyISAM	0	0 B	1 kB	2010-01-30 09:15:34
solicitud	MyISAM	5	220 B	6 kB	2010-01-30 09:15:38
anexo	MyISAM	4	236 B	2 kB	2010-01-30 09:15:38
cargo	MyISAM	5	295 B	2 kB	2010-01-30 09:15:38
producto	MyISAM	8	352 B	3 kB	2010-01-30 09:15:38
presupuesto	MyISAM	13	428 B	2 kB	2010-01-30 09:15:38
proveedor	MyISAM	8	504 B	3 kB	2010-01-30 09:15:38
salidaslc	MyISAM	10	660 B	3 kB	2010-01-30 09:15:37
prdsalida	MyISAM	35	0.9 kB	2 kB	2010-01-30 09:15:36
solicitanes	MyISAM	31	3.4 kB	2 kB	2010-01-30 09:15:37
catalogo	MyISAM	241	12.5 kB	8 kB	2010-01-30 09:15:36

Summary statistics at the bottom of the table view:

- Num. of Tables: 11
- Rows: 360
- Data Len: 19.4 kB
- Index Len: 34 kB

Figura. 4.3.6 Esquema de tablas de la base de datos

## Eliminación de tablas

Si se desea borrar una tabla existente, así como sus datos, tecleamos:

```
DROP TABLE <tabla>;
```

### **Consulta de la base de datos**

La manipulación de datos del lenguaje de consultas estructurado, puede ser resumida en los siguientes cuatro comandos: SELECT, INSERT, UPDATE y DELETE. Existe una serie de reglas comunes de sintaxis para todas las instrucciones. A continuación se muestran algunas de estas:

- Si se utilizará más de una tabla en la cláusula FROM, todos los campos con el mismo nombre, deben ir precedidos por el nombre de la tabla en donde se vayan a utilizar.
- Las instrucciones múltiples de una cláusula WHERE deben conectarse por medio de alguna palabra clave, AND u OR.
- Todas las instrucciones SQL deben tener un verbo, una cláusula FROM o INTO y usualmente una lista de parámetros de nombres de columna.

### **SELECT**

La instrucción SELECT es probablemente la más utilizada de las instrucciones SQL. Ésta devuelve únicamente los datos contenidos en la base de datos. MySQL probablemente procesa este tipo de consulta más rápido que cualquier otra base de datos. A continuación se muestra su sintaxis.

```
SELECT <columna> FROM <tabla> WHERE <condición>;
```

El nombre de columna puede ser de una o más columnas que pertenezcan a una tabla en la cláusula FROM. El orden en que se especifiquen los nombres de las columnas es con el que se mostrarán. Si se utiliza el comodín \* se seleccionarán todas las columnas. En MySQL pueden ocuparse funciones de manejo de cadenas, aritméticas, de fecha, etc. Por ejemplo:

SELECT concat(Nombre, “”,Apellido\_Materno, “ “, Apellido\_Materno) as nombre  
FROM clientes;

SELECT max(Id\_Cliente) as numero\_actual FROM clientes;

En la primera consulta, la función concat, concatena los campos Nombre, Apellido\_Paterno y Apellido\_Paterno. En la segunda selecciona el Id\_Cliente con el valor más grande de la tabla “clientes”. Se muestra otro ejemplo en la fig. 4.3.7

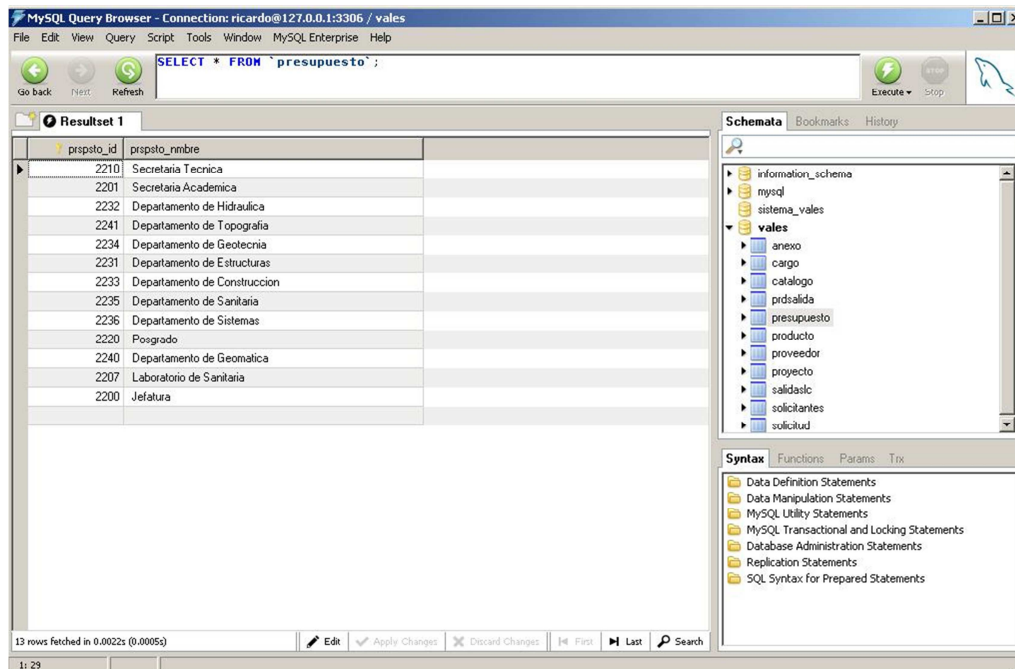


Figura. 4.3.7 Sentencia SELECT

## INSERT

El comando INSERT INTO, añade nuevos registros a una tabla existente. El formato es básicamente el mismo que el de la instrucción SELECT. Un comando INSERT INTO, sería como el siguiente:

INSERT INTO Clientes VALUES(NULL,“Carlos”,“Cleriga”);

El nombre de la tabla en la que se realizará la inserción se pone después de la palabra reservada INTO. Después de la palabra clave VALUES deberá ir la



lista de valores que se quieren insertar. Si se inserta un registro de esta forma, debe tener un valor para cada columna y el orden de los datos deberá concordar con el orden en que se declararon las columnas en la base de datos.

## **UPDATE**

El comando UPDATE permite editar valores de datos existentes. Este comando es muy similar a SELECT, pero permite manipular los valores seleccionados. La sintaxis es la siguiente:

```
UPDATE <tabla> SET <columna> = <valor> WHERE <condición>;
```

Al igual que con todas las instrucciones SQL, el verbo reservado va primero. A éste le sigue el nombre de la tabla que se manipulará. A continuación se emplea la palabra reservada SET y, por último, se coloca la lista de nombres de columna y valores que se desean establecer. Pueden existir valores múltiples después de comando SET. Cada par columna/valor debe estar separado por una coma. La cláusula WHERE designa qué registros serán actualizados.

## **DELETE**

La instrucción DELETE es muy similar a SELECT, la única diferencia es que en lugar de seleccionar registros para verlos, borra esos registros. Debe tenerse precaución al usar esta instrucción y evitar accidentalmente borrar registros. Puede usarse primero la instrucción SELECT y así asegurarse de que solamente se borrará lo deseado. El verbo reservado se escribe primero, enseguida la palabra reservada FROM, seguida del nombre de la tabla y después la cláusula WHERE donde se establece la condición que deben cumplir los registros a borrarse. Si no se especifica la condición se borrarán todos los registros de la tabla:

```
DELETE FROM <tabla> WHERE <condición>;
```

## Respaldo de la base de datos

El respaldo es parte fundamental de la administración de cualquier sistema de bases de datos. Los motivos van desde cualquier incidente físico del equipo, hasta algún daño accidental por parte de algún usuario. Tan importante como el respaldo, es la restauración de los datos. MySQL incluye un par de utilerías de gran beneficio para el respaldo y la restauración de las bases de datos: `mysqldump` y `mysqlimport` respectivamente.

La utilería para respaldar una base de datos en MySQL es `mysqldump`, esta puede ser utilizada desde la línea de comandos y ser invocada por un programa de calendarización como `cron` en UNIX/Linux. La sintaxis para `mysqldump` es la siguiente:

```
$mysqldump <opciones> <base de datos> <tablas>
```

Si no se especifican las opciones o tablas, se respaldará toda la base de datos especificada. Cuando se ejecuta `mysqldump`, en realidad se crea un archivo lleno con las instrucciones `CREATE` o `INSERT` para crear las especificaciones de las tablas y datos.

```
mysql> mysqldump --opt sistema_vales > backup.sql
```

Con la sentencia anterior se realiza el respaldo de la base de datos.

```
mysql> sistema_vales < backup.sql
```

Y se respalda con el archivo creado.

Otra manera es mediante la interfaz gráfica, con la utilería de Backup. Vea fig. 4.3.8

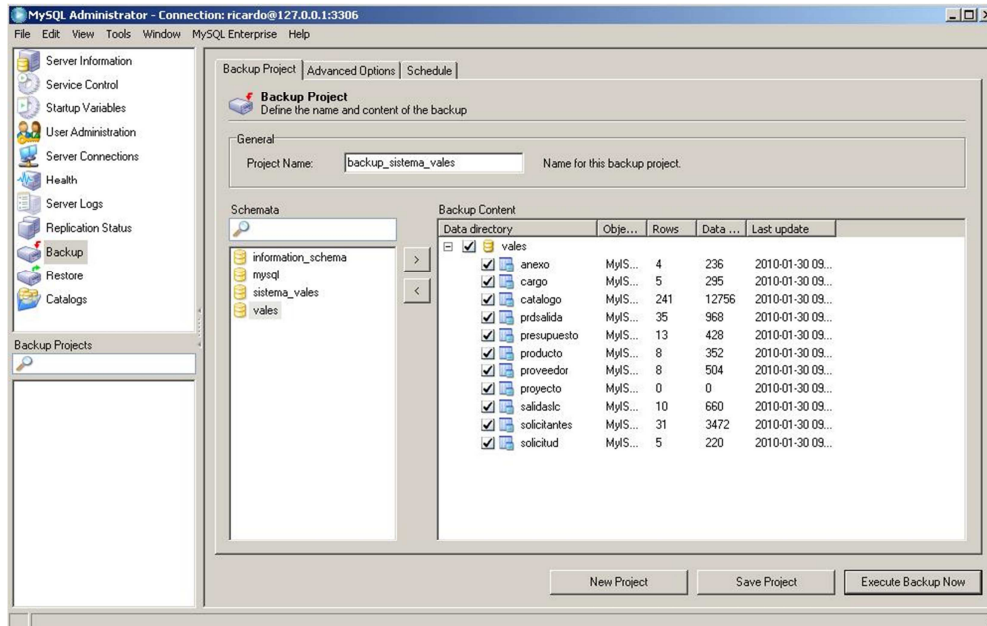


Figura. 4.3.8 Creación de respaldo de la base de datos

La utilidad `mysqlimport` lee algunos formatos de datos, entre ellos los delimitados por tabuladores, y los inserta en la base de datos indicada, la sintaxis es la siguiente:

```
mysqlimport <opciones> <base de datos> <archivo de respaldo>
```

#### 4.4 DISEÑO Y CONSTRUCCIÓN DEL FRONT- END

El sistema de Automatización de Vales está basado en web que integra la parte que sirve para la generación de dichos vales tanto de almacén como de compra además de varios módulos de consulta de reportes, altas y bajas de usuarios y productos.

Este sitio es un portal web que se implementó usando archivos PHP, HTML, Java Script y algunos otros tipos de archivo, distribuidos en cuatro secciones para tener un mejor manejo de lo que se quiere realizar, además de contar con ciertos privilegios los cuales hacen que como usuario no administrador se tenga la única sección que en nuestro caso será la de generar vales de almacén y compra.

En el directorio **principal**, existen los archivos php los cuales contienen código html que servirá para obtener el contenido informativo que se expondrá en cada una de las páginas que conforman el sitio. A cada acceso o liga de los menús de la página principal, le corresponderá un archivo php en este directorio, el cual contendrá toda la información necesaria ya sea texto y/o imágenes necesarias para presentar la información que se desee en cada una de ellas.

En el directorio **css** se encuentran las hojas de estilo, las cuales son muy utilizadas en los desarrollos en web para separar el estilo lógico del estilo físico del sitio, dejando este último en bloques de definición de estilos separados de la estructura del documento. El modo de funcionamiento de las CSS consiste en definir, mediante una sintaxis especial, la forma de presentación que le aplicaremos a:

- Un sitio web entero, de modo que se puede definir la forma de todo el sitio de una sola vez.
- Un documento HTML o página. Se puede definir la forma, en un pequeño trozo de código en la cabecera, a toda la página.
- Una porción del documento, aplicando estilos visibles en un trozo de la página.
- Una etiqueta en concreto, llegando incluso a poder definir varios estilos diferentes para una sola etiqueta. Esto es muy importante ya que ofrece potencia en nuestra programación.

En el directorio **images**, como su nombre lo indica, se incluirán todas las imágenes que se deseen exponer a través del sitio. El objetivo de esta estructura continúa la tendencia de hacer lo más sencillo posible, el mantenimiento y administración de contenidos que se deberá realizar sobre el sitio.

Finalmente, en el directorio **site** se encontrarán los archivos php que conforman la funcionalidad del sitio. En esta sección encontraremos que existe un archivo php para cada uno de las diferentes peticiones del usuario, es decir

archivos de reportes ya sean de compra y/o almacén y sus diferentes variables que existen para el sitio, los cuales le corresponden un acceso o liga de los menús de la página principal, que es en donde se encuentran las funciones necesarias para incluir los archivos generales, las imágenes para conformar cada página completa.

Es así como mediante la interacción de todos estos elementos, se logra la funcionalidad completa del sitio web de la División de Ingenierías Civil y Geomática para su objetivo que es la de automatizar la obtención de vales, así mismo como el obtener reportes de los mismos. El portal comienza con una página de **logueo** como se muestra en la Figura 4.4.1



Figura 4.4.1 Página de inicio del Sistema de Vales de la DICyG

Entrando como usuario administrador se presenta la siguiente pantalla Figura 4.4.2 en la cual se tienen muchas más opciones que solo entrando como usuario de privilegios restringidos en el cual solo se tiene la opción de generación de vales. Para obtener el acceso al sistema mediante el archivo *valida.php* verifica el tipo de usuario que quiere acceder al sistema y además si éste está registrado previamente en la base de datos, el código que lo verifica es el siguiente:

```
if($_POST)
{
    $user = @$_POST['user'];
    $pass = @$_POST['pass'];
```

```

$conexion = mysql_connect("localhost","root","dra96F");
mysql_select_db( "nomb_base",$conexion);
$sql = "SELECT slc_RFC, slc_pass, slc_fltro FROM solicitantes
WHERE slc_RFC='$user' && slc_pass='$pass'";
$result = mysql_query($sql,$conexion);

if(mysql_num_rows($result) > 0)
{
    $result2 = mysql_fetch_array($result);
    if($result2[2] == 1)
    {
        $_SESSION["slc_RFC"] = $result2[0];
        $_SESSION["slc_pass"] = $result2[1];
        $_SESSION["slc_fltro"] = $result2[2];
        $aviso = "";

        header("location:opcionesUsuarios.php");
    }
    else if($result2[2] == 2)
    {
        $_SESSION["slc_RFC"] = $result2[0];
        $_SESSION["slc_pass"] = $result2[1];
        $_SESSION["slc_fltro"] = $result2[2];
        $aviso = "";
        header("location:opciones.php");
    }
}

```



Figura 4.4.2 Página principal del sitio

Por otro lado, las páginas de contenido dinámico son aquellas que involucran consultas hacia una o varias tablas de la base de datos. Estas páginas serán también resultado de la interpretación de archivos PHP que tendrán codificada la funcionalidad necesaria para establecer, si es el caso, los criterios de búsqueda requeridos y con base en ellos, publicar información de la base de datos a través de las consultas apropiadas. En la figura 4.4.3 se muestra por ejemplo, la página de reportes de compra por fecha.

Para el ejemplo, el archivo que hace la búsqueda pertinente a la fecha es el archivo *consultFecha.php* la cual recibe las variables fecha, mes y año y despliega la consulta si existen datos dentro de la base de datos, el código que genera esto es el siguiente:

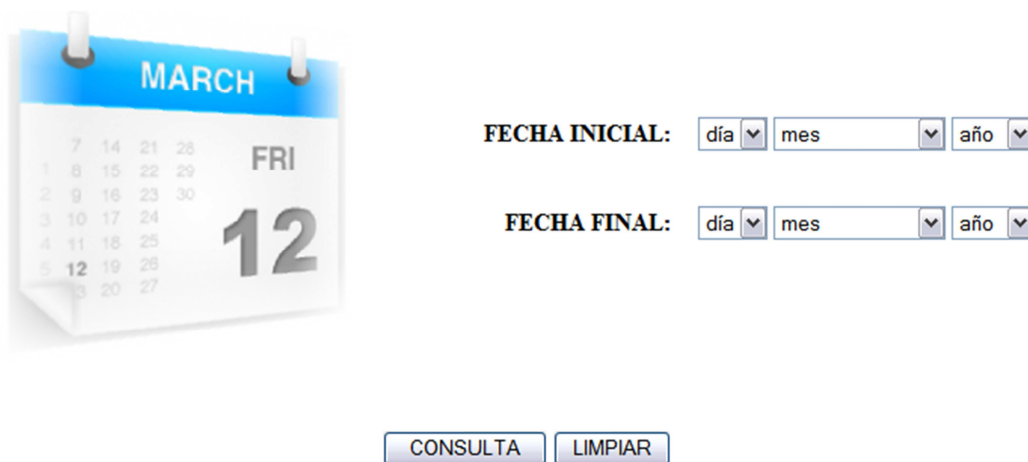
```
if($base=mysql_select_db("nomb_base"))
{
    $fechalni = @ $_POST['fechalni'];
    $meslni = @ $_POST['meslni'];
    $aniolni = @ $_POST['aniolni'];
    $fechaFin = @ $_POST['fechaFin'];
```

```

$mesFin = @ $_POST['mesFin'];
$anioFin = @ $_POST['anioFin'];
$fechaInicial = "$anioIni"."-$mesIni"."-$fechaIni";
$fechaFinal = "$anioFin"."-$mesFin"."-$fechaFin";
$total = 0;
$totalIva = 0;
$fechaInicialNorm = "$fechaIni"."-$mesIni"."-$anioIni";
$fechaFinalNorm = "$fechaIni"."-$mesIni"."-$anioIni";
    
```

```

$sql ="SELECT solicitud.slt_fecha, solicitud.slt_folio, solicitud.prspsto_id,
solicitantes.slc_nombre, solicitantes.slc_dpto, producto.prc_cntdad,
producto.prc_dscrip, producto.prc_prcunidad, producto.prc_imprte,
producto.prc_total FROM solicitud, solicitantes, producto
WHERE slt_fecha >='$fechaInicial'AND slt_fecha <='$fechaFinal' AND
producto.slt_folio = solicitud.slt_folio AND solicitantes.slc_RFC =
solicitud.slc_RFC";
    
```



The image shows a user interface for selecting a date range. On the left, there is a calendar for the month of March, with the date 12th highlighted. To the right of the calendar are two date selection fields. The first field is labeled 'FECHA INICIAL:' and the second is labeled 'FECHA FINAL:'. Each field contains three dropdown menus for selecting the day, month, and year. Below these fields are two buttons: 'CONSULTA' and 'LIMPIAR'.

Figura 4.4.3 Consulta de Compras por Fecha

Finalmente lo esencial del sistema que es el generar vales de almacén y de compra, mediante el menú de Generación de Vales se tienen las dos opciones de tipo de vale que se necesite, donde se debe escoger la que se vaya a generar en la Figura 4.4.4 por ejemplo se muestra la pantalla en donde hay que ingresar los datos de los insumos requeridos teniendo todos los campos obligatorios llenos se procede entonces a **generar el vale de compra** el cual



se presenta en formato PDF con previa elaboración mediante una biblioteca elaborada en PHP.

Para esta parte se tiene un archivo que va comportándose dinámicamente al introducir los datos del usuario y lo más importante que sería el ir actualizando los precios tanto parciales como el total de la cotización, es decir se obtiene su RFC y teléfono este último necesario para la generación del vale pues es requisito que aparezca en cambio el primero es de suma importancia ya que con el RFC se podrán obtener los diversos reportes que se necesitan, esto se logra gracias a Java Script por medio de un par de funciones establecidas en el archivo *solicitudCompra.php* además del envío de los datos hacia la base de datos y casi de forma conjunta la llamada del archivo que genera el vale final en formato PDF, a continuación el código que genera esto, solo por citar un ejemplo:

```
<script language="javascript">
function envio_form(){
    if (document.main.nombre_sol.value.length==0){
        alert("Tiene que elegir un solicitante")
        document.main.nombre_sol.focus()
        return 0;
    }
    if (document.main.prv_nmbre0.value.length==0){
        alert("Tiene que ingresar al menos un proveedor")
        document.main.prv_nmbre0.focus()
        return 0;
    }
    document.main.target = "_blank";
    document.main.id.value="respuesta";
    document.main.action = "valeSolicitudCompraPdf.php"
    document.main.submit();
    document.main.target = "_self";
    document.main.id.value="tablón";
    document.main.action = "enviaValeCompra.php"
    document.main.submit();
}
```



El fragmento de código que hace esto es el siguiente:

```

$j=0;
for($i=0; $i<=14; $i++){
$cant = "cantidad".$i;
$descrip = "prc_dscrip".$i;
$unidad = "prc_unidad".$i;
$precio = "precio".$i;
$importe = "importe".$i;
$cantidad = @ $_POST[$cant];
$descripcion = @ $_POST[$descrip];
$unidadMedida = @ $_POST[$unidad];
#number_format($precio, 3, '.', ',');
$precioUnidad = @ $_POST[$precio];
$importeUnidad = @ $_POST[$importe];

if(!empty($descripcion)){
$j++;
$pdf->Cell(8.5,4.25,".$j,1,0,'C'); //número
$pdf->Cell(18.5,4.25,".$cantidad,1,0,'C'); //cantidad
$pdf->Cell(100.5,4.25,".strtoupper($descripcion),1,0,'L'); // descripción
$pdf->SetFont(".$tipoLetra,",4.5);
$pdf->Cell(9,4.25,".strtoupper($unidadMedida),1,0,'C'); //unidad
$pdf->SetFont(".$tipoLetra,",6);
$pdf->Cell(13,4.25,'$ '.number_format($precioUnidad,2),1,0,'C'); //precio por
unidad
$pdf->Cell(26,4.25,'$ '.number_format($importeUnidad,2),1,1,'C'); // importe
    }
}

//Pinta la tabla con las celdas vacías (las que no se ocuparon)
$limite=15-$j;

```

## IMPLEMENTACIÓN Y PERSONALIZACIÓN DEL SISTEMA

```
for($k=0; $k<$limite; $k++){  
$pdf->Cell(8.5,4.25,"",1,0,'C'); //número  
$pdf->Cell(18.5,4.25,"",1,0,'C'); //cantidad  
$pdf->Cell(100.5,4.25,"",1,0,'C'); // descripción  
$pdf->Cell(9,4.25,"",1,0,'C'); //unidad  
$pdf->Cell(13,4.25,"",1,0,'C'); //precio por unidad  
$pdf->Cell(26,4.25,"",1,1,'C'); // importe  
}
```



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
 FACULTAD DE INGENIERÍA  
 CLAVE: 416  
 SECRETARÍA ADMINISTRATIVA  
 BIENES Y SUMINISTROS  
 SOLICITUD INTERNA DE COMPRA

SELLO

ÁREA SOLICITANTE:	DIVISIÓN DE INGENIERÍAS CIVIL Y GEOMÁTICA		FOLIO:	02	02	2010
NOMBRE DEL SOLICITANTE:	ING. MARCOS TREJO HERNANDEZ		FECHA:	02	02	2010
TELÉFONO:	62228008		ING. RODOLFO SOLÍS UBALDO AUTORIZACIÓN DEL RESPONSABLE DEL ÁREA			
No.	CANTIDAD	DESCRIPCIÓN DE LOS BIENES E INSUMOS REQUERIDOS	UNIDAD	PRECIO POR UNIDAD	IMPORTE	
1	1	TAMBOR TONER IMPRESORA DELL	PZA	\$ 1,500.00	\$ 1,500.00	
2	2	IMPRESORAS LASER-JET HP	PZA	\$ 2,350.00	\$ 4,700.00	
				COSTO TOTAL	\$ 6,200.00	

FECHA COMPROMISO DE ENTREGA: \_\_\_\_\_  
 FECHA DE LIBERACIÓN: \_\_\_\_\_ (Fecha en que el servicio esta disponible para su entrega al usuario).

No.	PROVEEDOR (ES)
1	OFFICE DEPOT
2	OFFICE MAX

ANEXOS:  
 COTIZACIONES  CATÁLOGOS  JUSTIFICACIÓN  OTROS

CON CARGO A:  
 PRESUPUESTO  PAPIT  CONACYT  INGRESOS EXT.   
 OTROS   
 NOMBRE DEL PROYECTO \_\_\_\_\_ No. PROYECTO: \_\_\_\_\_

ESPACIO EXCLUSIVO PARA SER LLENADO POR EL ÁREA DE ADQUISICIONES

DIVISA	TIPO DE CAMBIO	EQUIVALENTE MONEDA NACIONAL
FORMA DE ADQUISICIÓN		
ADJUDICACIÓN DIRECTA	INVITACIÓN A CUANDO MENOS TRES PERSONAS	LICITACIÓN PÚBLICA
SVA	DGPROV	DGPROV
POR MONTOS	SUBCOMITE	SUBCOMITE
POR EXCEPCIÓN		

ESPACIO EXCLUSIVO PARA SER LLENADO POR EL ÁREA DE PRESUPUESTO

CÓDIGO	CUENTA ING. EXT.	PREVIO No.

OBSERVACIONES:

---



---



---

VO. BO. RESPONSABLE DE BIENES Y SUMINISTROS

VO. BO. SUFICIENCIA

AUTORIZACIÓN SECRETARIO O JEFE DE UNIDAD ADMINISTRATIVA

SR. MANUEL VALADEZ HERNÁNDEZ  
 NOMBRE Y FIRMA

C.P. DAVID TÉLLEZ ESTRADA  
 NOMBRE Y FIRMA

ING. LUIS JIMÉNEZ ESCOBAR  
 NOMBRE Y FIRMA

Figura 4.4.5 Generación del Vale de Compra en formato PDF

#### 4.5 INTEGRACIÓN Y PRUEBAS DEL SISTEMA

La integración es una técnica sistemática para construir la estructura del programa, es decir, consiste en poner juntos a los módulos para detectar errores asociados con la interacción, por ello las pruebas constituyen una parte integral y vital del desarrollo del sistema. Se realizan con el propósito de descubrir defectos y se establecen para mejorar la calidad del mismo.

Los criterios que se utilizan para la realización de los casos de pruebas del sistema, permiten detectar los errores y poder así proseguir con la corrección de los mismos. A continuación se lista los diferentes tipos de prueba:

**Prueba de caja negra.** La prueba funcional es un enfoque para probar si las especificaciones del componente probado son utilizadas para derivar casos de prueba. El componente de una caja negra y su comportamiento sólo puede ser determinado al estudiar sus entradas y salidas relacionadas a éstas.

Se centran en los requisitos funcionales del software, el objetivo es construir conjuntos de datos de entrada que consigan ejercitar todos los requisitos funcionales de un programa. En las pruebas de caja negra no nos fijamos en el código del programa sino en la especificación de programación. Uno u otro determinan lo que el programa tiene que hacer.

**Prueba de caja blanca.** Son pruebas que se basan en la lógica del programa, no en la especificación. Se realizan utilizando el conocimiento del funcionamiento interno del código. Las pruebas de caja blanca sólo se pueden realizar por programadores.

Entre las pruebas de caja blanca se encuentran las pruebas de instrucciones, pruebas de decisiones que se refieren a el código del programa, las sentencias de bifurcación condicionales deben tener una coherencia y una cobertura de ramas al 100% como un situación deseable, pero habitualmente es un objetivo excesivamente costoso de alcanzar en su plenitud. También se encuentran las pruebas de cubrimiento y las pruebas de ciclos.

**Prueba de regresión.** Es volver a ejecutar un subconjunto de pruebas que se han llevado a cabo anteriormente para asegurarse de que los cambios no han propagado efectos colaterales no deseados.

- **Prueba de validación.** La validación proporciona una seguridad final de que el software satisface todos los requisitos funcionales de comportamiento y rendimientos. Durante la validación se usan exclusivamente pruebas de caja negra.

**Pruebas de integración.** Es una técnica sistemática para construir la estructura del programa, mientras que al mismo tiempo se llevan a cabo las pruebas para detectar errores asociados con la interacción. Los tipos fundamentales de integración son:

- **Integración incremental.** Se combina el siguiente módulo que se debe probar con el conjunto de módulos que ya han sido probados.
- **Integración no incremental.** Se prueba cada módulo por separado y luego se integran todos de una vez y se prueba el programa completo.
  - **Ascendente.** Se comienza por los módulos hoja.
  - **Descendente.** Se comienza por el módulo raíz.

**Pruebas unitarias o de módulo.** Estas pruebas enfocan el esfuerzo de verificación en las unidades más pequeñas de diseño de SW, es decir, el módulo, probando los caminos de control importantes con el fin de descubrir errores dentro de los límites de éste.

**Pruebas de validación o aceptación.** La información contenida en una sección de la especificación de los requerimientos de software, forma la base para un enfoque de pruebas de validación. La validación del software se logra por medio de las pruebas de caja negra que demuestran la conformidad con los requerimientos del usuario.

**Pruebas ALFA y BETA.** Las pruebas Alfa son llevadas a cabo por el usuario en el lugar del desarrollo, donde el desarrollador participa como observador.

Las pruebas Beta se llevan cabo por los usuarios finales, en el lugar de trabajo de éstos, y el desarrollador generalmente no está presente.

**Pruebas de stress.** Las pruebas de stress ayudan a simular casos en los que el número de clientes o la recuperación masiva de datos de una base de datos aumentan. De este modo es posible evaluar tanto el tiempo de respuesta de un sistema como su capacidad de responder ante esos casos.

Las pruebas que tienen como propósito el validar que el producto de software funcione correctamente son llamadas pruebas limpias o pruebas positivas, la desventaja es que sólo pueden validar el producto específicamente para los casos de uso. Un número finito de pruebas no puede validar que el producto funcione correctamente para todas las situaciones. La presencia de al menos una falla en este tipo de pruebas, es suficiente para mostrar que el producto no funciona correctamente.

### **PRUEBAS DEL SISTEMA**

Se realizaron pruebas de caja negra sobre el módulo de vale de salida de almacén: Al ingresar a éste módulo de la figura 4.5.1, siempre aparecerá una lista de los productos con los que cuenta la unidad, permitiendo elegir uno de ellos y así pasar al siguiente módulo de selección.





Figura 4.5.1 Lista de productos existentes.

Una vez elegidos los productos, aparecerá en una lista desplegable las personas autorizadas para solicitar un vale de salida de almacén; cuya elección llenará automáticamente los registros de RFC y departamento, tal y como se muestra en la figura 4.5.2.

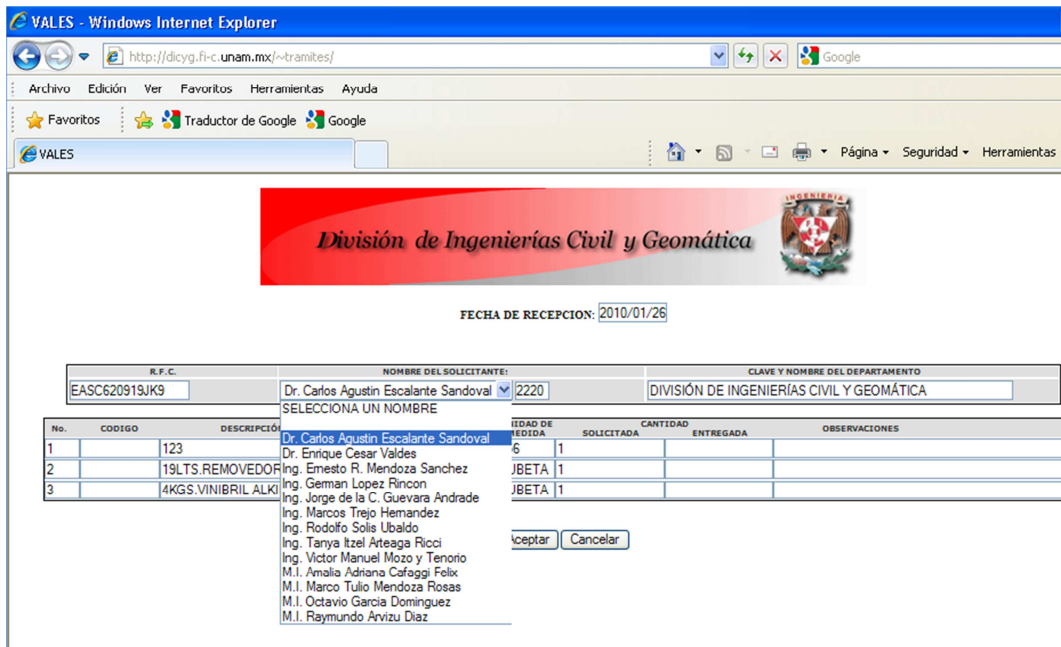



Figura 4.5.2 Selección del solicitante.




**INTRODUCE LOS DATOS**

RFC	<input type="text"/>
Nombre y Apellidos	JUAN CARLOS GALICIA VEGA
Nombre del Departamento	<input type="text"/> as
Ubicación	<input type="text"/>
Teléfono	<input type="text"/>
Correo	<input type="text"/>
Contraseña	<input type="text"/>
Tipo de Usuario	<input type="text"/> todos los privilegios



Mensaje de página web



Tiene que escribir su R.F.C.

Figura 4.5.4 Mensaje solicitando introducir la información requerida.

Pruebas de unidades. Se centra en el proceso de verificación de una unidad o módulo de software, tal como está implementada en el código fuente. La ventana de validación de RFC mostrado en la Figura 4.5.5 provocada por intentar guardar un RFC con clave repetida, puede llevar a un mensaje de error (Figura 4.5.6) con otras validaciones necesarias de los datos.



*División de Ingenierías Civil y Geomática*

**ALTA PARA NUEVOS SOLICITANTES**

INTRODUCE LOS DATOS

RFC	<input type="text" value="GAVJ820516"/>
Nombre y Apellidos	<input type="text" value="FREDY CARDEÑA FLORES"/>
Nombre del Departamento	<input type="text" value="Departamento de Estructuras"/>
Ubicación	<input type="text" value="EDIFICIO B"/>
Teléfono	<input type="text" value="45689745"/>
Correo	<input type="text" value="fredy@hotmail.com"/>
Contraseña	<input type="password" value="*****"/>
Tipo de Usuario	<input type="text" value="2 -- todos los privilegios"/>



Figura 4.5.5 Alta de usuario con RFC repetido.

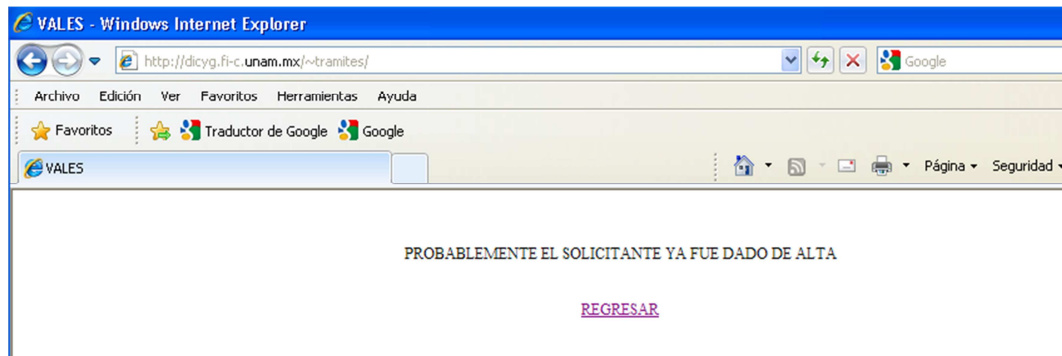


Figura 4.5.6 Mensaje de error por RFC repetido.

## Tipos de Mantenimiento

Un aspecto importante en el diseño y desarrollo de sistemas es el mantenimiento. Una vez acabada la fase de desarrollo y la implantación del sistema, es imprescindible garantizar el mantenimiento del mismo. Los programas deben ser modificados con el tiempo ya que las reglas de negocio pueden variar o es necesario adaptarse al entorno o a los cambios en los sistemas tecnológicos.

Existen diferentes tipos de mantenimiento aplicables al sistema:

- **Mantenimiento Preventivo.** Es la actividad en la cual se realizan cambios a la aplicación para mejorar el mantenimiento futuro, la estabilidad y confiabilidad en la operación. También es útil para proporcionar bases seguras sobre las que podrán implementarse mejoras posteriores.
- **Mantenimiento Correctivo.** Se presenta cuando ocurre un error en la operación del sistema.
- **Mantenimiento Adaptativo.** Se presenta cuando se generan cambios en los requerimientos de tal manera que la especificación sea adaptada a estos nuevos requerimientos, verificando que la nueva implementación cumpla con ellos. Estos requerimientos también pueden ser cambios en las plataformas de hardware o sistemas operativos.
- **Mantenimiento Perfectivo.** Se realiza cuando existe la necesidad de optimización de procesos, sin que cambien forzosamente los requerimientos. La especificación permite entender claramente el impacto de los cambios de manera que estos se implanten confiadamente.

Así podemos ver la importancia que tiene cada uno de estos para el funcionamiento adecuado del sistema, por lo que se debe de crear un plan para ejecutarlos de manera periódica. Principalmente el mantenimiento preventivo, pues es a partir de este que se puede detectar la necesidad de aplicar alguno

de los otros tipos de mantenimiento; como mínimo, el mantenimiento preventivo se debe realizar cada mes, dado el uso continuo que se le dará a un sistema de este tipo.

A pesar de esto, un sistema siempre es propenso a fallar eventualmente, por lo que como parte del mantenimiento, es altamente recomendable que se realicen respaldos de la información contenida en la base, esto previendo que el sistema llegara a tener alguna falla grave, o por algún imprevisto, sea borrado total o parcialmente del servidor en donde se aloja, así la información por lo menos se encontraría en resguardo.

## 4.6 OBTENCIÓN DE REPORTES

Las ventanas con las que contara el sistema generador de vales serán citadas a continuación.

### Para usuarios

#### Ventana de Inicio de sesión

En esta parte el usuario se identifica ante el sistema mediante un nombre de usuario y una contraseña. A continuación se incluye una vista de este proceso en la figura 4.6.1

INVOICE

Sold to: \_\_\_\_\_ Ship to: \_\_\_\_\_

Quantity	Price	Amount

División de Ingenierías Civil y Geomática

Iniciar Sesión

Usuario: mortal

Contraseña: \_\_\_\_\_

Sign in

Figura 4.6.1 Inicio de sesión.

## Ventana de tipo de vales

En esta ventana se ofrece al usuario la opción de seleccionar el tipo de vale que desea elaborar, ya sea de solicitud interna de compra o de vale de salida de almacén, a continuación en la figura 4.6.2 se ofrece una ilustración de esta ventana.



Figura 4.6.2 Tipo de vales.





## IMPLEMENTACIÓN Y PERSONALIZACIÓN DEL SISTEMA

- Proveedor: Se escribe el proveedor de ese material.

En la parte de “anexos” se seleccionara una de las siguientes opciones:

- Cotizaciones
- Catálogos
- Justificación
- Otros

En la sección de “con cargo a” una de las siguientes opciones, con las siguientes observaciones:

- Presupuesto: Se elegirá un código presupuestal
- Ingresos ext: En esta opción se habilita la parte de nombre del proyecto y no. proyecto para que sean llenados.
- Otros: En esta opción se habilita la parte de nombre del proyecto y no. proyecto para que sean llenados.
- Observaciones: Este se llenara en caso de ser requerido.

Este proceso se ejemplifica mediante la figura 4.6.4.

No.		PROVEEDOR(ES)			
1		OFFICE DEPOT			
2		CONOCIDO			

ANEXOS				
<input checked="" type="radio"/> COTIZACIONES	<input type="radio"/> CATALOGOS	<input type="radio"/> JUSTIFICACIÓN	<input type="radio"/> OTROS	

CON CARGO A				
<input checked="" type="radio"/> PRESUPUESTO	<input type="radio"/> PAPIIT	<input type="radio"/> CONACYT	<input type="radio"/> INGRESOS EXT.	<input type="radio"/> OTROS
SELECCIONA UN CÓDIGO				No. PROYECTO
SELECCIONA UN CÓDIGO				
2200 Jefatura				
2201 Secretaria Academica				
2207 Laboratorio de Sanitaria				
2210 Secretaria Tecnica				
2220 Posgrado				
2231 Departamento de Estructuras				
2232 Departamento de Hidraulica				
2233 Departamento de Construccion				
2234 Departamento de Geotecnia				
2235 Departamento de Sanitaria				
2236 Departamento de Sistemas				
2240 Departamento de Geomatica				
2241 Departamento de Topografia				

OBSERVACIONES	

Figura 4.6.4 Proveedores, anexos, cargos y observaciones.



### Para administradores

Además de las ventanas anteriores, los administradores al iniciar sesión en el sistema, manejan las siguientes opciones:

- Alta de usuarios
- Alta de productos
- Reportes de compra
- Reportes de Almacén



Figura 4.6.5 Opciones de administrador.

### Alta de usuarios

En esta sección se deberán llenar los siguientes campos:

- RFC
- Nombre y Apellidos
- Nombre del Departamento: Se selecciona de una lista.
- Ubicación
- Teléfono
- Correo
- Contraseña
- Tipo de Usuario: Se selecciona de una lista.

Esto se ejemplifica a continuación en la figura 4.6.6



*División de Ingenierías Civil y Geomática*

**ALTA PARA NUEVOS SOLICITANTES**

INTRODUCE LOS DATOS

RFC	<input type="text"/>
Nombre y Apellidos	<input type="text"/>
Nombre del Departamento	Coordinación de Ing. Geomatica ▾
Ubicación	<input type="text"/>
Teléfono	<input type="text"/>
Correo	<input type="text"/>
Contraseña	<input type="text"/>
Tipo de Usuario	0 — sin privilegios ▾



Figura 4.6.6 Alta de usuarios.

### Alta de Productos

En esta sección se deberán llenar los siguientes campos:

- Clave del Producto
- Descripción
- Unidad de medida

Esto se muestra a continuación en la figura 4.6.7.



*División de Ingenierías Civil y Geomática*

**ALTA PARA NUEVOS PRODUCTOS**

INTRODUCE LOS DATOS

Clave del Producto	<input type="text"/>
Descripción	<input type="text"/>
Unidad de Medida	<input type="text"/>



Figura 4.6.7 Alta de productos.

## Reportes de compra

De esta sección se puede buscar las transacciones efectuadas por los usuarios, las opciones de consulta a elegir son por medio de las siguientes opciones:

- R.F.C.
- Fecha
- Código
- Con cargo a.

A continuación la figura 4.6.8 con las opciones.



Figura 4.6.8 Reportes de compra

### RFC

Se selecciona el método de consulta seleccionando el RFC del usuario de una lista ya definida.

### Fecha

Esta consulta se realiza indicando el lapso en el cual se desea enterar de las transiciones, indicando “fecha de inicial y fecha final”.

### Código

Se selecciona el código correspondiente al área de la cual se desean saber sus operaciones realizadas.

### Con cargo a

De esta sección se pueden elegir las siguientes opciones:

- PAPIIT
- CONACYT
- INGRESOS EXT
- PROYECTOS

Las opciones se muestran a continuación en la figura 4.6.9




Figura 4.6.9 Reportes de compra con cargo a

Para todas estas opciones correspondientes a la opción "con cargo a" el método de búsqueda es por medio de las siguientes opciones que muestran en la figura 4.6.10 y se citaran a continuación:

- Fecha inicial
- Fecha final
- R.F.C.

---



*División de Ingenierías Civil y Geomática*

CONSULTA CON CARGO A:

FECHA INICIAL: día ▾ mes ▾ año ▾

FECHA FINAL: día ▾ mes ▾ año ▾

R.F.C: SELECCIONA EL R.F.C. ▾




Figura 4.6.10 Opciones de consulta

### Reportes de almacén

Para esta sección se cuenta con las secciones de “fecha” y “código presupuestal” presentándolos a continuación en la figura 4.6.11

---



*División de Ingenierías Civil y Geomática*

CONSULTA POR:



FECHA



CÓDIGO PRESUPUESTAL



Figura 4.6.11 Opciones reportes de almacén.



### Por fecha

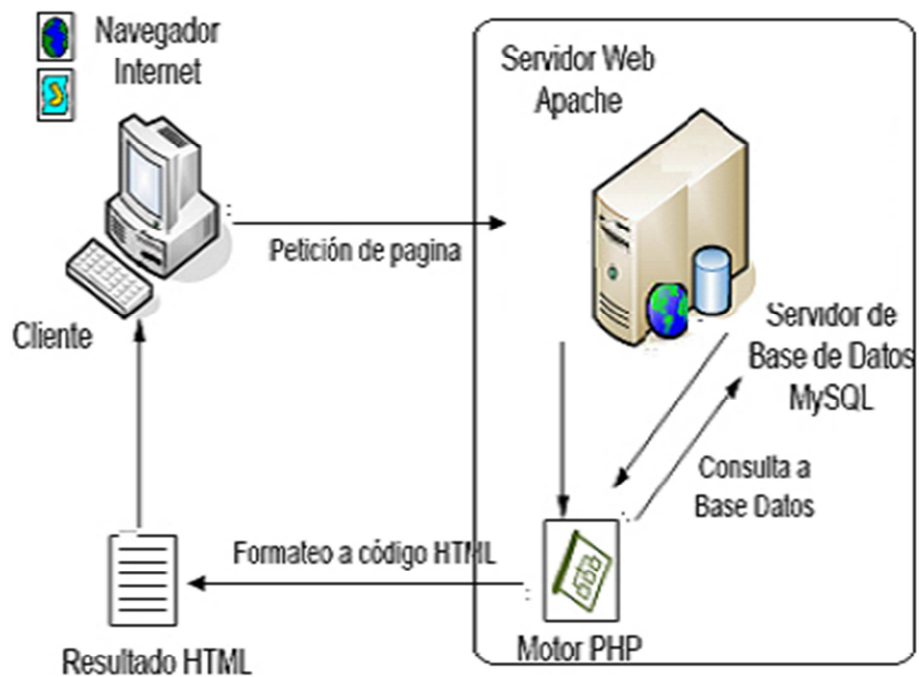
Se requiere de definir los parámetros siguientes para obtener su resultado:

- Fecha inicial
- Fecha final

### Por código presupuestal

Se selecciona el código correspondiente al área de la cual se desean saber sus operaciones realizadas.

El sistema quedara de la siguiente manera:



## CONCLUSIONES

Se cumplieron las expectativas en el desempeño y desarrollo del sistema, ya que el aplicar tecnología de código abierto garantiza que el sistema cumplirá con los requisitos de calidad y compatibilidad de hoy en día. Así mismo se cuenta con el soporte adecuado por parte de la gran comunidad de desarrolladores de código abierto a nivel global.

La aplicación de la arquitectura web, permite la disponibilidad de los datos en línea, así como el fácil mantenimiento y la distribución de las tareas, permitiendo un rendimiento óptimo en la aplicación lo que hace que el sistema sea eficaz y competente a las necesidades de la disponibilidad e integridad de la información.

El proceso ahora con la aplicación web funciona almacenando los datos en una base de datos (BD). Esta BD está formada por un número variable de tablas que contienen columnas y filas, estas tablas se componen del contenido que ha sido previamente cargado en ellas a través de formularios.

En estas tablas llamamos al nombre de cada columna campo. Y a cada fila registro. Ambos en el sistema poseen números de id (identificación) que son únicos para cada uno de ellos.

Las páginas que se generan a partir de esos contenidos son llamadas dinámicas. En este contexto el término dinámico no indica movimiento o animación, sino que hace referencia al hecho de que las páginas dinámicas de un sitio web se generan a partir de una solicitud o consulta que realiza una máquina cliente a un servidor web (en este caso).

Dada la problemática y los puntos seguidos en búsqueda de la solución, los resultados se alcanzaron satisfactoriamente, llevando a conformar un sistema sólido práctico y funcional. Los beneficios obtenidos optimizan la forma de generación de vales disminuyendo el costo humano en ejecución de las tareas que permiten la generación de los mismos.

Para este problema las aplicaciones web resultaron una solución idónea, ya que los problemas de ubicación y accesos son cubiertos perfectamente por estas tecnologías, así como las bases de datos cumplen con almacenar los vales evitando tener pilas de papeles. Pasar de lo manual a lo automático es una gran ventaja además de optimizar el sistema rudimentario, ayuda en la reducción de tiempos en el proceso de generación de vales.

**BIBLIOGRAFÍA:**

Maslakowsky Mark y Butcher Tony, Aprendiendo MySQL en 21 Días. Pearson Educación, México 2001

Apache Web Server Administration & e-commerce Handbook. Scott Hawkins. Prentice Hall, 2001

Tanenbaum, Andrew S. Sistemas Operativos Modernos, México 2009

F. Korth Henry, et al., Fundamentos de Bases de Datos, Mc Graw Hill, 3ra edición

**SITIOS WEB**

<http://es.wikipedia.org/wiki/Almac%C3%A9n>

<http://www.monografias.com/trabajos12/alma/alma.shtml>

[http://es.wikipedia.org/wiki/Aplicaci%C3%B3n\\_web](http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web)

[http://livedocs.adobe.com/dreamweaver/8\\_es/using/wwhelp/wwhimpl/common/](http://livedocs.adobe.com/dreamweaver/8_es/using/wwhelp/wwhimpl/common/)

[html/wwhelp.htm?context=LiveDocs\\_Parts&file=gs\\_12\\_u8.htm](http://www.adobe.com/livehelp/wwhelp/wwhimpl/common/html/wwhelp.htm?context=LiveDocs_Parts&file=gs_12_u8.htm)

[http://es.wikipedia.org/wiki/Servidor\\_HTTP\\_Apache](http://es.wikipedia.org/wiki/Servidor_HTTP_Apache)

<http://www.debian.org/index.es.html>

<http://es.wikipedia.org/wiki/Normalización>

<http://www.lamphowto.com>

<http://dev.mysql.com>