

**UNIVERSIDAD PEDAGÓGICA DE EL SALVADOR
FACULTAD DE CIENCIAS ECONÓMICAS**



**“DESARROLLO DE UN SISTEMA DE CONTROL DE ASIGNACIÓN
Y MANTENIMIENTO DE EQUIPO DE INFORMATICA PARA EL
HOSPITAL NACIONAL DOCTOR LUIS EDMUNDO VASQUEZ DE
CHALATENANGO 2009”**

TITULO A OPTAR:

TECNICO EN SISTEMAS DE COMPUTACION

INTEGRANTES:

CHRISTIAN ALFREDO VASQUEZ CARDOZA

JULIO CÉSAR AVALOS BARRERA

AMADEO ERNESTO VELASQUEZ NUÑEZ

SAN SALVADOR, SEPTIEMBRE DE 2009.

INDICE

I.DATOS DE IDENTIFICACIÓN

1.1 Tema	4
1.2 Objetivos	4
1.3 Introducción	5
1.4 Planteamiento del problema	6
1.5 Justificación	7
1.6 Fundamentación teórica	8
1.6.1 Sistemas de información	8
1.6.1.1 Definición de sistema	8
1.6.1.2 Definición de sistema de información	9
1.6.1.3 Tipos y usos de los sistemas de información	12
1.6.1.4 Principales características de los sistemas de información	13
A) Sistemas transaccionales.	13
B) Sistemas de apoyo de las decisiones.	14
C) Sistemas gerenciales	15
D) Sistemas estratégicos.	16
E) Sistemas de información a ejecutivos	17
F) Sistemas de control	19
1.6.1.5 Evolución de los sistemas de información	25
A) Etapa de contagio o expansión.	27
B) Etapa de control o formalización.	28
C) Etapa de integración.	29
D) Etapa de administración de datos.	30
E) Etapa de madurez.	30
1.6.1.6 Ciclo de vida de los sistemas de información	31
1.6.2 Ingeniería de Software	33
1.6.2.1 Etapas del Proceso de Ingeniería de Software	33
1.6.2.2 Modelos de Desarrollo de Software	36

1.6.3 Proceso de desarrollo unificado	39
1.6.3.1 Características del Proceso Unificado	39
1.6.3.2 Etapas de Implementación del Proceso Unificado	41
1.6.4 Tecnologías de aplicaciones	42
1.6.4.1 Definición de Aplicación	42
1.6.4.2 Aplicación Distribuida	42
1.6.4.2.1 Cliente Servidor	43
1.6.4.2.2 Aplicación Multi-capas	47
1.6.4.2.3 Aplicación web	54
1.6.5 ¿Qué es un sistema de inventarios?	58
1.6.6 Software Propietario y Software Libre	59
1.6.6 Tecnologías Utilizadas	66

II MARCO EMPIRICO

2.1 Metodología	69
2.2 Procedimiento para la obtención de datos	69
2.3 Marco Operativo	70
2.3.1 Especificación de requisitos de software	70
2.3.2 Casos de uso	73
2.3.3 Especificaciones de casos de uso	76
2.3.4 Diagrama de clases	84
2.3.5 Diseño de la base de datos	85
2.3.6 Diseño de Interfaz Grafica de Usuario	88
2.4 Nuevas formulaciones teóricas	90
2.5 Bibliografía	91
2.6 Anexos	92

I

DATOS DE

IDENTIFICACIÓN

1.1 TEMA

“Desarrollo de un sistema de control de asignación y mantenimiento de equipo de informática para el Hospital Nacional Doctor Luis Edmundo Vásquez de Chalatenango 2009”

1.2 OBJETIVOS

OBJETIVO GENERAL

Desarrollar una herramienta software que administre datos de los equipos de informática del Hospital Nacional de Chalatenango para un mejor control de asignación y mantenimiento de estos.

OBJETIVOS ESPECIFICOS

- Recolectar información teórica y de campo sobre los sistemas de información que existen para poder desarrollar el marco teórico de este proyecto
- Crear una base de datos y el diseño del sistema para el control de asignación y mantenimiento del equipo de informática.

1.3 INTRODUCCIÓN

Para poder optar al título de Técnico en Sistemas de Computación, se ha desarrollado un sistema de control de asignación y mantenimiento de equipo de informática para el Hospital Nacional Doctor Luis Edmundo Vásquez de Chalatenango, con el cual se pretende que la institución pueda dar mejores prestaciones a sus usuarios a través del uso de esta tecnología.

Este informe se divide en dos partes: en la primera el lector encontrará información teórica sobre lo que son los sistemas de información, los diferentes tipos y usos, así como sus principales características y como ha ido evolucionando con el paso del tiempo y la tecnología. También hemos desarrollado los temas de las tecnologías de Aplicaciones, proceso unificado y el concepto de Sistema de Inventario, con el fin de que el lector pueda comprender desde el punto de vista teórico el funcionamiento de nuestro sistema.

En la segunda parte que se le ha denominado marco empírico se explica cual ha sido la metodología y el proceso para la obtención de los datos. Se muestra información importante sobre la elaboración de la base de datos y algunas ventanas que han sido diseñadas para el fácil manejo del usuario del sistema.

Por último se proponen algunas nuevas formulaciones teóricas como recomendaciones para estudios futuros y las fuentes bibliográficas a las cuales recurrimos para nuestra información.

1.4 PLANTEAMIENTO DEL PROBLEMA

En la actualidad la tecnología ha facilitado el trabajo de las personas en los diferentes campos laborales como la salud, el comercio, entre otros. Pero personas que laboran en estos campos manifiestan su descontento por que los equipos se dañan rápidamente por no tener un buen control de estos.

Partiendo de esta premisa se puede decir que no es ajeno a esta realidad lo que sucede en el Hospital Nacional de Chalatenango, Los equipos son dañados debido a su variable uso y mal cuidado de la persona a la cual se le fue asignado, esto conlleva a que no se de una atención optima a los usuarios que acuden a este centro asistencial, ya que por medio de la tecnología se mantiene un control de los medicamentos, materiales de limpieza, ropa para los pacientes, utilitario y equipo, contabilidad, registro de pacientes, entre otros.

1.5 JUSTIFICACIÓN

Viendo la necesidad en la que se encuentra el Hospital Nacional de Chalatenango, se ha querido implementar un sistema de control de mantenimiento y asignación de equipo de informática para mantener en muy buen estado las diferentes computadoras y sus periféricos, y así prestar un mejor servicio a sus usuarios.

Pero también el sistema ayudará a que se tenga un control de la persona a la que ha sido entregada el equipo, en que departamento se encuentra asignada, el devaluó que ha adquirido en el tiempo de uso, el tiempo transcurrido del último mantenimiento, las piezas que han sido cambiadas o que están defectuosas, todo esto ayudará a que no se generen más gastos de los necesarios en la institución.

1.6 FUNDAMENTACIÓN TEORICA

1.6.1 SISTEMAS DE INFORMACIÓN

1.6.1.1 DEFINICIÓN DE SISTEMA

Un sistema, es un conjunto de partes o elementos organizados y relacionados que interactúan entre sí para lograr un objetivo. Los sistemas reciben (entrada) datos, energía o materia del ambiente y proveen (salida) información, energía o materia.

Un sistema puede ser físico o concreto (una computadora, un televisor, un humano) o puede ser abstracto o conceptual (un software)

Cada sistema existe dentro de otro más grande, por lo tanto un sistema puede estar formado por subsistemas y partes, y a la vez puede ser parte de un súper sistema.

Los sistemas tienen límites o fronteras, que los diferencian del ambiente. Ese límite puede ser físico (el gabinete de una computadora) o conceptual. Si hay algún intercambio entre el sistema y el ambiente a través de ese límite, el sistema es abierto, de lo contrario, el sistema es cerrado.

El ambiente es el medio en externo que envuelve física o conceptualmente a un sistema. El sistema tiene interacción con el ambiente, del cual recibe entradas y al cual se le devuelven salidas. El ambiente también puede ser una amenaza para el sistema



1.6.1.2 DEFINICIÓN DE SISTEMA DE INFORMACIÓN

En términos generales, hablamos de información como un conjunto de datos que están organizados y que tienen un significado. De esta manera, si tomamos datos por separado no tendrían un significado mientras que si los agrupamos en forma organizada, sí. Para ejemplificarlo en forma simple, los datos 2, 0, 0 y 7, no tienen un significado por sí mismos y no conforman una información mientras que si los organizamos significativamente, sí: 2007.

La información es un elemento fundamental en el proceso de la comunicación, ya que tiene un significado para quien la recibe, que la va a comprender si comparte el mismo código que quien la envía. Esto no sólo ocurre en un proceso social sino también en el mundo de la informática.

Durante el siglo pasado, el avance tecnológico y de la informática devino en que actualmente se ligara la información a una verdadera revolución, especialmente concebida por la globalización e Internet, un proceso y un sistema de comunicación que evitan la existencia de barreras entre la confluencia de información desde un punto al otro del planeta.

En cuanto al universo de la computadora, la información es un factor fundamental que se representa a través de símbolos, específicamente en forma de datos binarios. Asimismo, es aquello que maneja un sistema (tanto en la entrada como en el proceso o el resultado de la operación), pudiendo ser la misma en la entrada y en la salida (en cuyo caso el sistema sería uno de flujo de información) o diferente (nos referimos a sistemas de tratamiento de la información).

Un sistema de información es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio.

El equipo computacional: el hardware necesario para que el sistema de información pueda operar.

El recurso humano que interactúa con el Sistema de Información, el cual está formado por las personas que utilizan el sistema.

Un sistema de información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información.

- a) **Entrada de Información:** Es el proceso mediante el cual el Sistema de Información toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos o información que provienen o son tomados de otros sistemas o módulos. Esto último se denomina interfaces automáticas.

Las unidades típicas de entrada de datos a las computadoras son las terminales, las cintas magnéticas, las unidades de diskette, los códigos de barras, los escáner, la voz, los monitores sensibles al tacto, el teclado y el mouse, entre otras.

- b) **Almacenamiento de información:** El almacenamiento es una de las actividades o capacidades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior. Esta información suele ser almacenada en estructuras de información denominadas archivos. La unidad típica de almacenamiento son los discos magnéticos o discos duros, los discos flexibles o diskettes y los discos compactos (CD-ROM).
- c) **Procesamiento de Información:** Es la capacidad del Sistema de Información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o

bien con datos que están almacenados. Esta característica de los sistemas permite la transformación de datos fuente en información que puede ser utilizada para la toma de decisiones, lo que hace posible, entre otras cosas, que un tomador de decisiones genere una proyección financiera a partir de los datos que contiene un estado de resultados o un balance general de un año base.

- d) **Salida de Información:** La salida es la capacidad de un Sistema de Información para sacar la información procesada o bien datos de entrada al exterior. Las unidades típicas de salida son las impresoras, terminales, diskettes, cintas magnéticas, la voz, los graficadores y los plotters, entre otros. Es importante aclarar que la salida de un Sistema de Información puede constituir la entrada a otro Sistema de Información o módulo. En este caso, también existe una interface automática de salida. Por ejemplo, el Sistema de Control de Clientes tiene una interface automática de salida con el Sistema de Contabilidad, ya que genera las pólizas contables de los movimientos procesales de los clientes.

Actividades que realiza un Sistema de Información:

Entradas:

- Datos generales del cliente: nombre, dirección, tipo de cliente, etc.
- Políticas de créditos: límite de crédito, plazo de pago, etc.
- Facturas (interface automático).
- Pagos, depuraciones, etc.

Proceso:

- Cálculo de antigüedad de saldos.
- Cálculo de intereses moratorios.
- Cálculo del saldo de un cliente.

Almacenamiento:

- Movimientos del mes (pagos, depuraciones).
- Catálogo de clientes.
- Facturas.

Salidas:

- Reporte de pagos.
- Estados de cuenta.
- Pólizas contables (interface automática)
- Consultas de saldos en pantalla de una terminal.

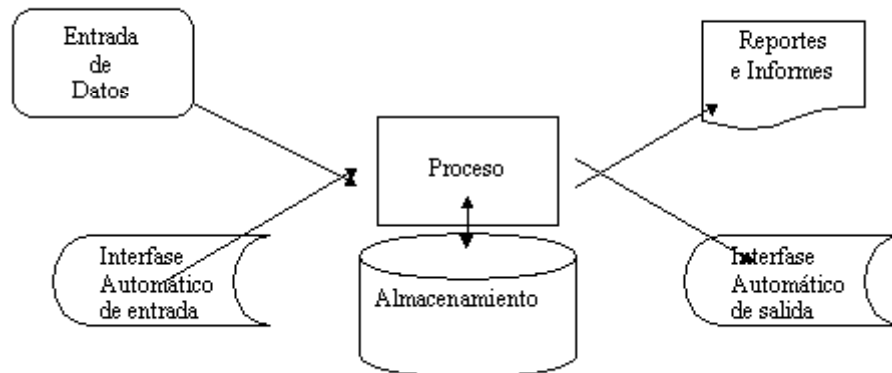
1.6.1.3 TIPOS Y USOS DE LOS SISTEMAS DE INFORMACIÓN

Durante los próximos años, los Sistemas de Información cumplirán tres objetivos básicos dentro de las organizaciones:

Automatización de procesos operativos.

Proporcionar información que sirva de apoyo al proceso de toma de decisiones.

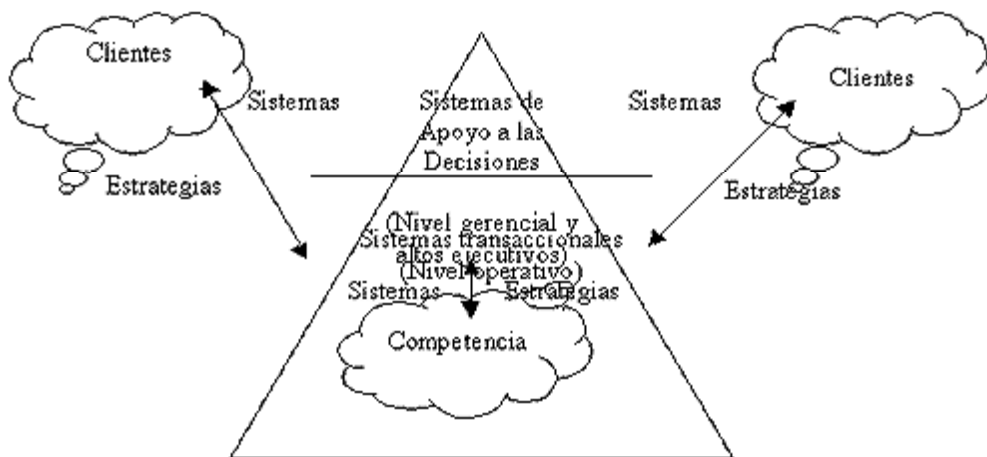
Lograr ventajas competitivas a través de su implantación y uso.



Los Sistemas de Información que logran la automatización de procesos operativos dentro de una organización, son llamados frecuentemente Sistemas Transaccionales, ya que su función primordial consiste en procesar transacciones tales como pagos, cobros, pólizas,

entradas, salidas, etc. Por otra parte, los Sistemas de Información que apoyan el proceso de toma de decisiones son los Sistemas de Soporte a la Toma de Decisiones, Sistemas para la Toma de Decisión de Grupo, Sistemas Expertos de Soporte a la Toma de Decisiones y Sistema de Información para Ejecutivos. El tercer tipo de sistema, de acuerdo con su uso u objetivos que cumplen, es el de los Sistemas Estratégicos, los cuales se desarrollan en las organizaciones con el fin de lograr ventajas competitivas, a través del uso de la tecnología de información.

Los tipos y usos de los Sistemas de Información se muestran a continuación:



1.6.1.4 PRINCIPALES CARACTERÍSTICAS DE LOS SISTEMAS DE INFORMACIÓN:

A) Sistemas Transaccionales.

Sus principales características son:

- A través de éstos suelen lograrse ahorros significativos de mano de obra, debido a que automatizan tareas operativas de la organización.
- Con frecuencia son el primer tipo de Sistemas de Información que se implanta en

las organizaciones. Se empieza apoyando las tareas a nivel operativo de la organización.

- Son intensivos en entrada y salida de información; sus cálculos y procesos suelen ser simples y poco sofisticados.
- Tienen la propiedad de ser recolectores de información, es decir, a través de estos sistemas se cargan las grandes bases de información para su explotación posterior.
- Son fáciles de justificar ante la dirección general, ya que sus beneficios son visibles y palpables.

B) Sistemas de Apoyo de las Decisiones.

Las principales características de estos son:

- Suelen introducirse después de haber implantado los Sistemas Transaccionales más relevantes de la empresa, ya que estos últimos constituyen su plataforma de información.
- La información que generan sirve de apoyo a los mandos intermedios y a la alta administración en el proceso de toma de decisiones.
- Suelen ser intensivos en cálculos y escasos en entradas y salidas de información. Así, por ejemplo, un modelo de planeación financiera requiere poca información de entrada, genera poca información como resultado, pero puede realizar muchos cálculos durante su proceso.
- No suelen ahorrar mano de obra. Debido a ello, la justificación económica para el

desarrollo de estos sistemas es difícil, ya que no se conocen los ingresos del proyecto de inversión.

- Suelen ser Sistemas de Información interactivos y amigables, con altos estándares de diseño gráfico y visual, ya que están dirigidos al usuario final.
- Apoyan la toma de decisiones que, por su misma naturaleza son repetitivos y de decisiones no estructuradas que no suelen repetirse. Por ejemplo, un Sistema de Compra de Materiales que indique cuándo debe hacerse un pedido al proveedor o un Sistema de Simulación de Negocios que apoye la decisión de introducir un nuevo producto al mercado.
- Estos sistemas pueden ser desarrollados directamente por el usuario final sin la participación operativa de los analistas y programadores del área de informática.

Este tipo de sistemas puede incluir la programación de la producción, compra de materiales, flujo de fondos, proyecciones financieras, modelos de simulación de negocios, modelos de inventarios, etc.

C) Sistemas Gerenciales

Estos sistemas son el resultado de interacción colaborativa entre personas, tecnologías y procedimientos -colectivamente llamados sistemas de información- orientados a solucionar problemas empresariales. Los **SIG** o **MIS** (también denominados así por sus siglas en inglés: *Management Information System*) se diferencian de los sistemas de información comunes en que para analizar la información utilizan otros sistemas que se usan en las actividades operacionales de la organización. Académicamente, el término es comúnmente utilizado para referirse al conjunto de los métodos de gestión de la información vinculada a la automatización o apoyo humano de la toma de decisiones (por ejemplo: Sistemas de apoyo a la decisión, Sistemas expertos y Sistemas de información para ejecutivos).

D) Sistemas Estratégicos.

Sus principales características son:

- Su función primordial no es apoyar la automatización de procesos operativos ni proporcionar información para apoyar la toma de decisiones.
- Suelen desarrollarse in house, es decir, dentro de la organización, por lo tanto no pueden adaptarse fácilmente a paquetes disponibles en el mercado.
- Típicamente su forma de desarrollo es a base de incrementos y a través de su evolución dentro de la organización. Se inicia con un proceso o función en particular y a partir de ahí se van agregando nuevas funciones o procesos.
- Su función es lograr ventajas que los competidores no posean, tales como ventajas en costos y servicios diferenciados con clientes y proveedores. En este contexto, los Sistema Estratégicos son creadores de barreras de entrada al negocio. Por ejemplo, el uso de cajeros automáticos en los bancos en un Sistema Estratégico, ya que brinda ventaja sobre un banco que no posee tal servicio. Si un banco nuevo decide abrir sus puerta al público, tendrá que dar este servicio para tener un nivel similar al de sus competidores.
- Apoyan el proceso de innovación de productos y proceso dentro de la empresa debido a que buscan ventajas respecto a los competidores y una forma de hacerlo en innovando o creando productos y procesos.

Un ejemplo de estos Sistemas de Información dentro de la empresa puede ser un sistema MRP (Manufacturing Resoure Planning) enfocado a reducir sustancialmente el desperdicio en el proceso productivo, o bien, un Centro de Información que proporcione todo tipo de información; como situación de créditos, embarques, tiempos de entrega, etc. En este contexto los ejemplos anteriores constituyen un Sistema de Información Estratégico si y

sólo sí, apoyan o dan forma a la estructura competitiva de la empresa.

Por último, es importante aclarar que algunos autores consideran un cuarto tipo de sistemas de información denominado Sistemas Personales de Información, el cual está enfocado a incrementar la productividad de sus usuarios.

E) Sistemas de información a ejecutivos

Un SIE (EIS por sus siglas en inglés) es un sistema de información para directivos que permite automatizar la labor de obtener los datos más importantes de una organización, resumirlos y presentarlos de la forma más comprensible posible, provee al ejecutivo acceso fácil a información interna y externa al negocio con el fin de dar seguimiento a los factores críticos del éxito.

Los SIE (EIS) se enfocan primordialmente a proporcionar información de la situación actual de la compañía y dejan en un plano secundario la visualización o proyección de esta información en escenarios futuros.

En un entorno característico de sistemas de información, el sistema consolida y administra muchas de las funciones de información diarias en relación con las áreas de oficina, administrativas, financieras y cualquier otra índole que el ejecutivo requiera.

Los SIE se construyen generalmente mediante la integración de software diseñado para operar conjuntamente con la infraestructura y las aplicaciones de información existentes en la institución.

El sistema debiera ofrecer informes y análisis de la información en tiempo real a toda la organización, debe incluir cuadros, gráficas e informes fáciles de leer, sobre todo información intuitiva que permita a los administradores realizar el seguimiento de indicadores críticos.

Los SIE debieran proporcionar acceso a la administración a categorías claves de datos relevantes, como son los datos internos creados por la organización, datos globales de la institución, datos externos (incluida información acerca de la competencia) y datos mundiales (con el uso de fuentes como Internet).

El uso de SIE ha permitido a muchas organizaciones comparar datos claves con los de sus competidores.

Es recomendable que los SIE incluyan una variedad de informes ordinarios incorporados, y las herramientas del sistema debieran permitir a los usuarios de administración crear informes especialmente diseñados que puedan enviarse a otra estación de trabajo o directamente a una impresora.

Características de un EIS

Un buen sistema de información para ejecutivos presenta información en forma de gráficos, columnas y textos.

La capacidad para hacer gráficos se necesita para facilitar en el análisis rápido de las condiciones y tendencias corrientes; las tablas presentan mayor detalle y permiten el análisis de variaciones; la información de textos añade interpretaciones y detalles de los datos.

Las principales características de los sistemas de información para ejecutivos (EIS) son las siguientes:

- Están diseñados para cubrir las necesidades específicas y particulares de la alta administración de la empresa.
- Extraen, filtran, comprimen y dan seguimiento a información crítica del negocio.

- Implica que los ejecutivos puedan interactuar en forma directa con el sistema sin el apoyo o auxilio de intermediarios.
- Es un sistema desarrollado con altos estándares en sus interfaces hombre-maquina, caracterizado por gráficas de alta calidad, información tabular y en forma de texto.
- Pueden acceder a información que se encuentra en línea, extrayéndose en forma directa de las bases de datos de la organización.
- El sistema está soportado por elementos especializados de hardware, tales como monitores o videos de alta resolución y sensibles al tacto, ratón e impresoras con tecnología avanzada.

F) Sistemas de control

Los **sistemas de control** según la Teoría Cibernética se aplican en esencia para los organismos vivos, las máquinas y las organizaciones. Estos sistemas fueron relacionados por primera vez en 1948 por Norbert Wiener en su obra *Cibernética y Sociedad* con aplicación en la teoría de los mecanismos de control. Un sistema de control está definido como *un conjunto de componentes que pueden regular su propia conducta o la de otro sistema con el fin de lograr un funcionamiento predeterminado, de modo que se reduzcan las probabilidades de fallos y se obtengan los resultados buscados*. Hoy en día los procesos de control son síntomas del proceso industrial que estamos viviendo. Estos sistemas se usan típicamente en sustituir un trabajador pasivo que controla una determinado sistema (ya sea eléctrico, mecánico, etc.) con una posibilidad nula o casi nula de error, y un grado de eficiencia mucho más grande que el de un trabajador.

Los sistemas de control deben conseguir los siguientes objetivos:

- a) Ser estables y robustos frente a perturbaciones y errores en los modelos.

- b) Ser eficiente según un criterio preestablecido evitando comportamientos bruscos e irreales.

Necesidades de la supervisión de procesos

- ✓ Limitaciones de la visualización de los sistemas de adquisición y control.
- ✓ Control vs Monitorización
- ✓ Control software. Cierre de lazo de control.
- ✓ Recoger, almacenar y visualizar información.
- ✓ Minería de datos

Clasificación de los sistemas de control según su comportamiento

Definiciones

Supervisión: acto de observar el trabajo o tareas de otro (individuo o máquina) que puede no conocer el tema en profundidad.

1. **Sistema de control de lazo abierto:** Es aquel sistema en que solo actúa el proceso sobre la señal de entrada y da como resultado una señal de salida independiente a la señal de entrada, pero basada en la primera. Esto significa que no hay retroalimentación hacia el controlador para que éste pueda ajustar la acción de control. Es decir, la señal de salida no se convierte en señal de entrada para el controlador. Ejemplo 1: el llenado de un tanque usando una manguera de jardín. Mientras que la llave siga abierta, el agua fluirá. La altura del agua en el tanque no puede hacer que la llave se cierre y por tanto no nos sirve para un proceso que necesite de un control de contenido o concentración. Ejemplo 2: Al hacer una tostada, lo que hacemos es controlar el tiempo de tostado de ella misma entrando una variable (en este caso el grado de tostado que queremos). En definitiva, el que nosotros introducimos como parámetro es el tiempo.

Estos sistemas se caracterizan por:

- ✓ Sencillos y de fácil concepto.

- ✓ Nada asegura su estabilidad ante una perturbación.
- ✓ La salida no se compara con la entrada.
- ✓ Es afectado por las perturbaciones. Éstas pueden ser tangibles o intangibles.
- ✓ La precisión depende de la previa calibración del sistema.

2. **Sistema de control de lazo cerrado:** Son los sistemas en los que la acción de control está en función de la señal de salida. Los sistemas de circuito cerrado usan la retroalimentación desde un resultado final para ajustar la acción de control en consecuencia. El control en lazo cerrado es imprescindible cuando se da alguna de las siguientes circunstancias:

- ✓ Cuando un proceso no es posible de regular por el hombre.
- ✓ Una producción a gran escala que exige grandes instalaciones y el hombre no es capaz de manejar.
- ✓ Vigilar un proceso es especialmente duro en algunos casos y requiere una atención que el hombre puede perder fácilmente por cansancio o despiste, con los consiguientes riesgos que ello pueda ocasionar al trabajador y al proceso.

Sus características son:

- ✓ Complejos, pero amplios en cantidad de parámetros.
- ✓ La salida se compara con la entrada y la afecta para el control del sistema.
- ✓ Estos sistemas se caracterizan por su propiedad de retroalimentación.
- ✓ Más estable a perturbaciones y variaciones internas.

Un ejemplo de un sistema de control de lazo cerrado sería el termotanque de agua que utilizamos para bañarnos. Otro ejemplo sería un regulador de nivel de gran sensibilidad de un depósito. El movimiento de la boya produce más o menos obstrucción en un chorro de aire o gas a baja presión. Esto se traduce en cambios de presión que afectan a la membrana de la válvula de paso, haciendo que se abra más cuanto más cerca se encuentre del nivel máximo.

Tipos de sistemas de control

Los sistemas de control son agrupados de la siguiente forma:

1. **Hechos por el hombre.** Como los sistemas eléctricos o electrónicos que están permanentemente capturando señales de estado del sistema bajo su control y que al detectar una desviación de los parámetros pre-establecidos del funcionamiento normal del sistema, actúan mediante sensores y actuadores, para llevar al sistema de vuelta a sus condiciones operacionales normales de funcionamiento. Un claro ejemplo de este será un termostato, el cual capta consecutivamente señales de temperatura. En el momento en que la temperatura desciende o aumenta y sale del rango, este actúa encendiendo un sistema de refrigeración o de calefacción.

1.1 Por su **causalidad** pueden ser: **causales** y **no causales**. Un sistema es causal si existe una **relación de causalidad** entre las salidas y las entradas del sistema, más explícitamente, entre la salida y los valores futuros de la entrada.

1.2 Según el número de **entradas** y **salidas** del sistema, se denominan:

1.2.1 De **una entrada y una salida** o **SISO** (single input, single output).

1.2.2 De **una entrada y múltiples salidas** o **SIMO** (single input, múltiple output).

1.2.3 De **múltiples entradas y una salida** o **MISO** (múltiple input, single output).

1.2.4 De **múltiples entradas y múltiples salidas** o **MIMO** (múltiple input, múltiple output).

1.3 Según la ecuación que define el sistema, se denomina:

1.3.1 **Lineal**, si la **ecuación diferencial** que lo define es **lineal**.

1.3.2 **No lineal**, si la **ecuación diferencial** que lo define es **no lineal**.

1.4 Las señales o variables de los sistemas dinámicos son **función del tiempo**. Y de acuerdo con ello estos sistemas son:

1.4.1 De **tiempo continuo**, si el modelo del sistema es una **ecuación diferencial**, y por tanto el tiempo se considera **infinitamente divisible**. Las variables de **tiempo continuo** se denominan también **analógicas**.

1.4.2 De **tiempo discreto**, si el sistema está definido por una **ecuación por**

diferencias. El tiempo se considera dividido en **períodos de valor constante**. Los valores de las variables son **digitales** (sistemas binario, hexadecimal, etc.), y su valor solo se conoce en cada período.

1.4.3 De **eventos discretos**, si el sistema evoluciona de acuerdo con variables cuyo valor se conoce al producirse un determinado evento.

1.5 Según la relación entre las variables de los sistemas, diremos que:

1.5.1 Dos sistemas están **acoplados**, cuando las variables de uno de ellos están relacionadas con las del otro sistema.

1.5.2 Dos sistemas están **desacoplados**, si las variables de ambos sistemas no tienen ninguna relación.

1.6 En función de la evolución de las variables de un sistema en el tiempo y el espacio, pueden ser:

1.6.1 **Estacionarios**, cuando sus variables son constantes **en el tiempo y en el espacio**.

1.6.2 **No estacionarios**, cuando sus variables no son constantes **en el tiempo o en el espacio**.

1.7 Según sea la respuesta del sistema (valor de la salida) respecto a la variación de la entrada del sistema:

1.7.1 El sistema se considera **estable** cuando ante una variación muy rápida de la entrada se produce una respuesta **acotada** de la salida.

1.7.2 El sistema se considera **inestable** cuando ante una entrada igual a la anteriormente se produce una respuesta **no acotada** de la salida.

1.8 Si se comparan o no, la entrada y la salida de un sistema, para controlar esta última, el sistema se denomina:

1.8.1 Sistema en **anillo abierto**, cuando la salida para ser controlada, no se compara con el valor de la señal de entrada o **señal de referencia**.

1.8.2 Sistema en **anillo cerrado**, cuando la salida para ser controlada, se compara con la **señal de referencia**. La señal de salida que es llevada junto a la señal de entrada, para ser comparada, se denomina señal de **feedback** o de **retroalimentación**.

1.9 Según la posibilidad de predecir el comportamiento de un sistema, es decir su

respuesta, se clasifican en:

1.9.1 Sistema **determinista**, cuando su comportamiento futuro es predecible dentro de unos límites de tolerancia.

1.9.2 Sistema **estocástico**, si es imposible predecir el comportamiento futuro. Las variables del sistema se denominan **aleatorias**.

2. **Naturales, incluyendo sistemas biológicos.** Por ejemplo, los movimientos corporales humanos como el acto de indicar un objeto que incluye como componentes del sistema de control biológico los ojos, el brazo, la mano, el dedo y el cerebro del hombre. En la entrada se procesa el movimiento y la salida es la dirección hacia la cual se hace referencia.

3. **Cuyos componentes están unos hechos por el hombre y los otros son naturales.** Se encuentra el sistema de control de un hombre que conduce su vehículo. Éste sistema está compuesto por los ojos, las manos, el cerebro y el vehículo. La entrada se manifiesta en el rumbo que el conductor debe seguir sobre la vía y la salida es la dirección actual del automóvil. Otro ejemplo puede ser las decisiones que toma un político antes de unas elecciones. Éste sistema está compuesto por ojos, cerebro, oídos, boca. La entrada se manifiesta en las promesas que anuncia el político y la salida es el grado de aceptación de la propuesta por parte de la población.

4. Un sistema de control puede ser neumático, eléctrico, mecánico o de cualquier tipo, su función es recibir entradas y coordinar una o varias respuestas según su lazo de control (para lo que está programado).

5. Control Predictivo, son los sistemas de control que trabajan con un sistema predictivo, y no activo como el tradicional (ejecutan la solución al problema el momento antes que este suceda). De esta manera, mejoran la eficiencia del proceso.

Características de un sistema de control

1. Señal de Corriente de Entrada: Considerada como estímulo aplicado a un sistema desde una fuente de energía externa con el propósito de que el sistema produzca una respuesta específica.
2. Señal de Corriente de Salida: Respuesta obtenida por el sistema que puede o no relacionarse con la respuesta que implicaba la entrada.
3. Variable Manipulada: Es el elemento al cual se le modifica su magnitud, para lograr la respuesta deseada. Es decir, se manipula la entrada del proceso.
4. Variable Controlada: Es el elemento que se desea controlar. Se puede decir que es la salida del proceso.
5. Conversión: Mediante receptores se generan las variaciones o cambios que se producen en la variable.
6. Variaciones Externas: Son los factores que influyen en la acción de producir un cambio de orden correctivo.
7. Fuente de Energía: Es la que entrega la energía necesaria para generar cualquier tipo de actividad dentro del sistema.
8. Retroalimentación: La retroalimentación es una característica importante de los sistemas de control de lazo cerrado. Es una relación secuencial de causas y efectos entre las variables del sistema. Dependiendo de la acción correctiva que tome el sistema, este puede apoyar o no una decisión, cuando en el sistema se produce un retorno se dice que hay una retroalimentación negativa; si el sistema apoya la decisión inicial se dice que hay una retroalimentación positiva.

1.6.1.5 EVOLUCIÓN DE LOS SISTEMAS DE INFORMACIÓN

De la sección anterior se desprende la evolución que tienen los Sistemas de Información en las organizaciones. Con frecuencia se implantan en forma inicial los Sistemas Transaccionales y, posteriormente, se introducen los Sistemas de Apoyo a las Decisiones. Por último, se desarrollan los Sistemas Estratégicos que dan forma a la estructura competitiva de la empresa.

En la década de los setenta, Richard Nolan, un conocido autor y profesor de la Escuela de Negocios de Harvard, desarrolló una teoría que impactó el proceso de planeación de los recursos y las actividades de la informática.

Según Nolan, la función de la Informática en las organizaciones evoluciona a través de ciertas etapas de crecimiento, las cuales se explican a continuación:

- Comienza con la adquisición de la primera computadora y normalmente se justifica por el ahorro de mano de obra y el exceso de papeles.
- Las aplicaciones típicas que se implantan son los Sistemas Transaccionales tales como nóminas o contabilidad.
- El pequeño Departamento de Sistemas depende en la mayoría de los casos del área de contabilidad.
- El tipo de administración empleada es escaso y la función de los sistemas suele ser manejada por un administrador que no posee una preparación formal en el área de computación.
- El personal que labora en este pequeño departamento consta a lo sumo de un operador y/o un programador. Este último podrá estar bajo el régimen de honorarios, o bien, puede recibirse el soporte de algún fabricante local de programas de aplicación.
- En esta etapa es importante estar consciente de la resistencia al cambio del personal y usuario (ciberfobia) que están involucrados en los primeros sistemas que se desarrollan, ya que estos sistemas son importantes en el ahorro de mano de obra.
- Esta etapa termina con la implantación exitosa del primer Sistema de Información.

Cabe recalcar que algunas organizaciones pueden vivir varias etapas de inicio en las que la resistencia al cambio por parte de los primeros usuarios involucrados aborta el intento de introducir la computadora a la empresa.

A) Etapa de contagio o expansión.

Los aspectos sobresalientes que permiten diagnosticar rápido que una empresa se encuentra en esta etapa son:

- Se inicia con la implantación exitosa del primer Sistema de Información en la organización. Como consecuencia de lo anterior, el primer ejecutivo usuario se transforma en el paradigma o persona que se habrá que imitar.
- Las aplicaciones que con frecuencia se implantan en esta etapa son el resto de los Sistemas Transaccionales no desarrollados en la etapa de inicio, tales como facturación, inventarios, control de pedidos de clientes y proveedores, cheques, etc.
- El pequeño departamento es promovido a una categoría superior, donde depende de la Gerencia Administrativa o Contraloría.
- El tipo de administración empleado está orientado hacia la venta de aplicaciones a todos los usuarios de la organización; en este punto suele contratarse a un especialista de la función con preparación académica en el área de sistemas.
- Se inicia la contratación de personal especializado y nacen puestos tales como analista de sistemas, analista-programador, programador de sistemas, jefe de desarrollo, jefe de soporte técnico, etc.
- Las aplicaciones desarrolladas carecen de interfaces automáticas entre ellas, de tal forma que las salidas que produce un sistema se tienen que alimentar en forma manual a otro sistema, con la consecuente irritación de los usuarios.

- Los gastos por concepto de sistemas empiezan a crecer en forma importante, lo que marca la pauta para iniciar la racionalización en el uso de los recursos computacionales dentro de la empresa. Este problema y el inicio de su solución marcan el paso a la siguiente etapa.

B) Etapa de control o formalización.

Para identificar a una empresa que transita por esta etapa es necesario considerar los siguientes elementos:

- Esta etapa de evolución de la Informática dentro de las empresas se inicia con la necesidad de controlar el uso de los recursos computacionales a través de las técnicas de presupuestación base cero (partiendo de que no se tienen nada) y la implantación de sistemas de cargos a usuarios (por el servicio que se presta).
- Las aplicaciones están orientadas a facilitar el control de las operaciones del negocio para hacerlas más eficaces, tales como sistemas para control de flujo de fondos, control de órdenes de compra a proveedores, control de inventarios, control y manejo de proyectos, etc.
- El departamento de sistemas de la empresa suele ubicarse en una posición gerencial, dependiendo del organigrama de la Dirección de Administración o Finanzas.
- El tipo de administración empleado dentro del área de Informática se orienta al control administrativo y a la justificación económica de las aplicaciones a desarrollar. Nace la necesidad de establecer criterios para las prioridades en el desarrollo de nuevas aplicaciones. La cartera de aplicaciones pendientes por desarrollar empieza a crecer.
- En esta etapa se inician el desarrollo y la implantación de estándares de trabajo

dentro del departamento, tales como: estándares de documentación, control de proyectos, desarrollo y diseño de sistemas, auditoría de sistemas y programación.

- Se integra a la organización del departamento de sistemas, personal con habilidades administrativas y preparadas técnicamente.
- Se inicia el desarrollo de interfaces automáticas entre los diferentes sistemas.

C) Etapa de integración.

Las características de esta etapa son las siguientes:

- La integración de los datos y de los sistemas surge como un resultado directo de la centralización del departamento de sistemas bajo una sola estructura administrativa.
- Las nuevas tecnologías relacionadas con base de datos, sistemas administradores de bases de datos y lenguajes de cuarta generación, hicieron posible la integración.
- En esta etapa surge la primera hoja electrónica de cálculo comercial y los usuarios inician haciendo sus propias aplicaciones. Esta herramienta ayudó mucho a que los usuarios hicieran su propio trabajo y no tuvieran que esperar a que sus propuestas de sistemas fueran cumplidas.
- El costo del equipo y del software disminuyó por lo cual estuvo al alcance de más usuarios.
- En forma paralela a los cambios tecnológicos, cambió el rol del usuario y del departamento de Sistemas de Información. El departamento de sistemas evolucionó hacia una estructura descentralizada, permitiendo al usuario utilizar herramientas para el desarrollo de sistemas.

- Los usuarios y el departamento de sistema iniciaron el desarrollo de nuevos sistemas, reemplazando los sistemas antiguos, en beneficio de la organización.

D) Etapa de administración de datos.

Entre las características que destacan en esta etapa están las siguientes:

- El departamento de Sistemas de Información reconoce que la información es un recurso muy valioso que debe estar accesible para todos los usuarios.
- Para poder cumplir con lo anterior resulta necesario administrar los datos en forma apropiada, es decir, almacenarlos y mantenerlos en forma adecuada para que los usuarios puedan utilizar y compartir este recurso.
- El usuario de la información adquiere la responsabilidad de la integridad de la misma y debe manejar niveles de acceso diferentes.

E) Etapa de madurez.

Entre los aspectos sobresalientes que indican que una empresa se encuentra en esta etapa, se incluyen los siguientes:

- Al llegar a esta etapa, la Informática dentro de la organización se encuentra definida como una función básica y se ubica en los primeros niveles del organigrama (dirección).
- Los sistemas que se desarrollan son Sistemas de Manufactura Integrados por Computadora, Sistemas Basados en el Conocimiento y Sistemas Expertos, Sistemas de Soporte a las Decisiones, Sistemas Estratégicos y, en general, aplicaciones que proporcionan información para las decisiones de alta administración y aplicaciones de carácter estratégico.

En esta etapa se tienen las aplicaciones desarrolladas en la tecnología de base de datos y se logra la integración de redes de comunicaciones con terminales en lugares remotos, a través del uso de recursos computacionales

1.6.1.6 CICLO DE VIDA DE LOS SISTEMAS DE INFORMACIÓN

Un *sistema de información* es el conjunto de recursos que permiten recoger, gestionar, controlar y difundir la información de toda una empresa u organización.

Desde los años setenta, los sistemas de bases de datos han ido reemplazando a los sistemas de ficheros en los sistemas de información de las empresas. Al mismo tiempo, se ha ido reconociendo la gran importancia que tienen los datos que éstas manejan, convirtiéndose en uno de sus recursos más importantes. Esto ha hecho que muchas empresas tengan departamentos que se encarguen de gestionar toda su información, que estará almacenada en una base de datos. Aparecen los papeles de *administrador de datos* y *administrador de la base de datos*, que son las personas encargadas de supervisar y controlar todas las actividades relacionadas con los datos de la empresa y con el ciclo de vida de las aplicaciones de bases de datos, respectivamente.

Un sistema de información está formado por los siguientes componentes:

- La base de datos.
- El SGBD.
- Los programas de aplicación.
- Los dispositivos físicos (ordenadores, dispositivos de almacenamiento, etc.).
- El personal que utiliza y que desarrolla el sistema.

La base de datos es un componente fundamental de un sistema de información. El ciclo de vida de un sistema de información está ligado al ciclo de vida del sistema de base de datos sobre el que se apoya. Al ciclo de vida de los sistemas de información también se le denomina *ciclo de vida de desarrollo del software*. Las etapas típicas del ciclo de vida de

desarrollo del software son: planificación, recolección y análisis de los requisitos, diseño (incluyendo el diseño de la base de datos), creación de prototipos, implementación, prueba, conversión y mantenimiento. Este ciclo de vida hace énfasis en la identificación de las funciones que realiza la empresa y en el desarrollo de las aplicaciones que lleven a cabo estas funciones. Se dice que el ciclo de vida de desarrollo del software sigue un enfoque orientado a funciones, ya que los sistemas se ven desde el punto de vista de las funciones que llevan a cabo. Por esta razón, el análisis estructurado hace énfasis en los diagramas de flujo de datos, siguiendo el movimiento de los datos a través de una secuencia de transformaciones, y refinando éstas a través de una serie de niveles. Lo mismo ocurre en el diseño estructurado, que ve a un sistema como una función que se descompone sucesivamente en niveles o sub-funciones.

Concentrándose en las funciones se infravaloran los datos y, en especial, la *estructura* de los datos que son manipulados por las funciones. El resultado es que estos sistemas tienen valor durante poco tiempo en relación con las necesidades de los usuarios a largo plazo. Esto sucede debido a que al poco tiempo de haber instalado un sistema, las funciones implementadas son en realidad un subconjunto de las funciones que los usuarios realmente desean. Casi inmediatamente, los usuarios descubren una gran variedad de servicios adicionales que quisieran incorporar al sistema. Estas necesidades causan problemas a los sistemas obtenidos con un diseño orientado a funciones, puesto que este diseño puede requerir una revisión importante para acomodar las funciones adicionales.

En contraste, el enfoque orientado a datos centra el foco de atención en el análisis de los datos utilizados por las funciones. Esto tiene dos ventajas. La primera es que los datos son una parte considerablemente más estable que las funciones. La segunda ventaja es que la propia estructura de un esquema de base de datos requiere de un análisis sofisticado de los datos y de sus relaciones. Una vez que se haya construido un esquema para la base de datos que sea lógico, podrían diseñarse tantas funciones como fuera necesario para sacar provecho del mismo. Sin embargo, sin un esquema tal, la base de datos sólo podría ser útil para una única aplicación. Por lo tanto, el enfoque orientado a funciones puede ser bueno para el desarrollo a corto plazo, pero pierde su valor real a largo plazo. Usando un enfoque

orientado a datos, los datos pasan a ser los cimientos sobre los cuales se puede construir una gran variedad de funciones diferentes.

Por lo tanto, en este capítulo se van a estudiar cada una de las etapas del ciclo de vida de desarrollo del software desde la perspectiva del desarrollo de una aplicación de bases de datos, siguiendo un enfoque orientado a datos.

1.6.2 INGENIERIA DE SOFTWARE

Ingeniería de software es la disciplina o área de la informática que ofrece métodos y técnicas para desarrollar y mantener software de calidad.

1.6.2.1 ETAPAS DEL PROCESO DE INGENIERIA DE SOFTWARE

La ingeniería de software requiere llevar a cabo numerosas tareas, dentro de etapas como las siguientes:

A) Análisis de requisitos

Extraer los requisitos de un producto de software es la primera etapa para crearlo. Mientras que los clientes piensan que ellos saben lo que el software tiene que hacer, se requiere de habilidad y experiencia en la ingeniería de software para reconocer requisitos incompletos, ambiguos o contradictorios. El resultado del análisis de requisitos con el cliente se plasma en el documento ERS, Especificación de Requerimientos del Sistema, cuya estructura puede venir definida por varios estándares, tales como CMM-I. Asimismo, se define un diagrama de Entidad/Relación, en el que se plasman las principales entidades que participarán en el desarrollo del software.

La captura, análisis y especificación de requisitos (incluso pruebas de ellos), es una parte crucial; de esta etapa depende en gran medida el logro de los objetivos finales. Se han ideado modelos y diversos procesos de trabajo para estos fines. Aunque aún no está formalizada, ya se habla de la Ingeniería de Requisitos.

La IEEE Std. 830-1998 normaliza la creación de las Especificaciones de Requisitos Software (Software Requirements Specification).

B) Especificación

La Especificación de Requerimientos describe el comportamiento esperado en el software una vez desarrollado.

Gran parte del éxito de un proyecto de software radicará en la identificación de las necesidades del negocio (definidas por la alta dirección), así como la interacción con los usuarios funcionales para la recolección, clasificación, identificación, priorización y especificación de los requerimientos del software.

Entre las técnicas utilizadas para la especificación de requerimientos se encuentra:

- Casos de Uso

C) Arquitectura

La integración de infraestructura, desarrollo de aplicaciones, bases de datos y herramientas gerenciales, requieren de capacidad y liderazgo para poder ser conceptualizados y proyectados a futuro, solucionando los problemas de hoy. El rol en el cual se delegan todas estas actividades es el del Arquitecto.

El Arquitecto de Software es la persona que añade valor a los procesos de negocios gracias a su valioso aporte de soluciones tecnológicas.

La Arquitectura de Sistemas en general, es una actividad de planeación, ya sea a nivel de infraestructura de red y hardware, o de Software.

La Arquitectura de Software consiste en el diseño de componentes de una aplicación (entidades del negocio), generalmente utilizando patrones de arquitectura. El diseño

arquitectónico debe permitir visualizar la interacción entre las entidades del negocio y además poder ser validado, por ejemplo por medio de diagramas de secuencia.

Un diseño arquitectónico describe en generar el cómo se construirá una aplicación de software. Para ello se documenta utilizando diagramas, por ejemplo:

- Diagramas de clases
- Diagramas de base de datos

Siendo estos los mínimos necesarios para describir la arquitectura de un proyecto que iniciará a ser codificado.

D) Programación

Reducir un diseño a código puede ser la parte más obvia del trabajo de ingeniería de software, pero no necesariamente es la que demanda mayor trabajo y ni la más complicada. La complejidad y la duración de esta etapa está íntimamente relacionada al o a los lenguajes de programación utilizados, así como al diseño previamente realizado.

E) Prueba

Consiste en comprobar que el software realice correctamente las tareas indicadas en la especificación del problema. Una técnica de prueba es probar por separado cada módulo del software, y luego probarlo de forma integral, para así llegar al objetivo. Se considera una buena práctica el que las pruebas sean efectuadas por alguien distinto al desarrollador que la programó, idealmente un área de pruebas; sin perjuicio de lo anterior el programador debe hacer sus propias pruebas. En general hay dos grandes formas de organizar un área de pruebas, la primera es que esté compuesta por personal inexperto y que desconozca el tema de pruebas, de esta forma se evalúa que la documentación entregada sea de calidad, que los procesos descritos son tan claros que cualquiera puede entenderlos y el software hace las cosas tal y como están descritas. El segundo enfoque es tener un área de pruebas conformada por programadores con experiencia, personas que saben sin mayores

indicaciones en qué condiciones puede fallar una aplicación y que pueden poner atención en detalles que personal inexperto no consideraría.

1.6.2.2 MODELOS DE DESARROLLO DE SOFTWARE

La ingeniería de software tiene varios modelos o paradigmas de desarrollo en los cuales se puede apoyar para la realización de software, de los cuales podemos destacar a éstos por ser los más utilizados y los más completos:

- **Modelo en cascada o Clásico (modelo tradicional)**

Es el enfoque metodológico que ordena rigurosamente las etapas del **ciclo de vida del software**, de tal forma que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior.

Un ejemplo de una metodología de desarrollo en cascada es:

1. Análisis de requisitos
2. Diseño del Sistema
3. Diseño del Programa
4. Codificación
5. Pruebas
6. Implantación
7. Mantenimiento

De esta forma, cualquier error de diseño detectado en la etapa de prueba conduce necesariamente al rediseño y nueva programación del código afectado, aumentando los costes del desarrollo. La palabra *cascada* sugiere, mediante la metáfora de la fuerza de la gravedad, el esfuerzo necesario para introducir un cambio en las fases más avanzadas de un proyecto.

- **Modelo en espiral (modelo evolutivo)**

Es un modelo de 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.

- **Modelo de prototipos**

El **desarrollo con prototipación**, también llamado **modelo de prototipos** que pertenece a los **modelos de desarrollo evolutivo**, se inicia con la definición de los objetivos globales para el software, luego se identifican los requisitos conocidos y las áreas del esquema en donde es necesaria más definición. Entonces se plantea con rapidez una iteración de construcción de prototipos y se presenta el modelado (en forma de un diseño rápido).

El diseño rápido se centra en una representación de aquellos aspectos del software que serán visibles para el cliente o el usuario final (por ejemplo, la configuración de la interfaz con el usuario y el formato de los despliegues de salida). El diseño rápido conduce a la construcción de un prototipo, el cual es evaluado por el cliente o el usuario para una retroalimentación; gracias a ésta se refinan los requisitos del software que se desarrollará. La iteración ocurre cuando el prototipo se ajusta para satisfacer las necesidades del cliente.

Esto permite que al mismo tiempo el desarrollador entienda mejor lo que se debe hacer y el cliente vea resultados a corto plazo.

- **Desarrollo por etapas**

El modelo de desarrollo de software por etapas es similar al Modelo de prototipos ya que se muestra al cliente el software en diferentes estados sucesivos de desarrollo, se diferencia

en que las especificaciones no son conocidas en detalle al inicio del proyecto y por tanto se van desarrollando simultáneamente con las diferentes versiones del código.

Pueden distinguirse las siguientes fases:

- Especificación conceptual
- Análisis de requerimientos
- Diseño inicial
- Diseño detallado, codificación, depuración y liberación

Estas diferentes fases se van repitiendo en cada etapa del diseño.

- **Desarrollo iterativo y creciente o Iterativo e Incremental**

Desarrollo iterativo y creciente (o incremental) es un proceso de desarrollo de software, creado en respuesta a las debilidades del modelo tradicional de cascada.

Para apoyar el desarrollo de proyectos por medio de este modelo se han creado frameworks (entornos de trabajo), de los cuales los dos más famosos son el Rational Unified Process y el Dynamic Systems Development Method (Método de desarrollo de sistemas dinámicos). El desarrollo incremental e iterativo es también una parte esencial de un tipo de programación conocido como Extreme Programming y los demás frameworks de desarrollo rápido de software.

- **RAD (Rapid Application Development)**

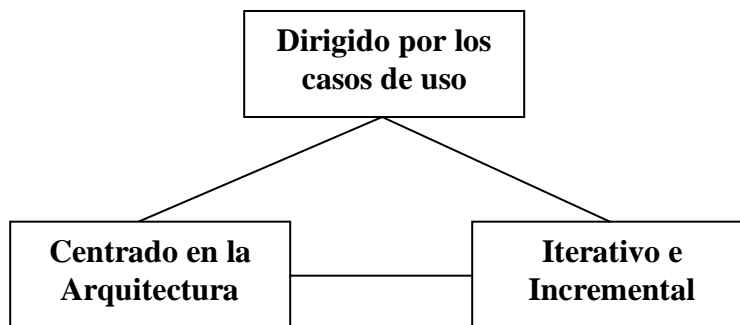
El Desarrollo rápido de aplicaciones o **Rapid Application Development (RAD)** es un proceso de desarrollo de software (en inglés, software development process), desarrollado inicialmente por James Martin en 1980. El método comprende el desarrollo iterativo, la construcción de prototipos y el uso de utilidades CASE (Computer Aided Software Engineering). Tradicionalmente, el desarrollo rápido de aplicaciones tiende a englobar también la usabilidad, utilidad y la rapidez de ejecución.

Hoy en día se suele utilizar para referirnos al desarrollo rápido de GUIs tal como Glade, o IDEs de desarrollo completas. Algunas de las plataformas más conocidas son Visual Studio, Delphi, Foxpro o Anjuta.

1.6.3 PROCESO DE DESARROLLO UNIFICADO

El Proceso Unificado es un proceso de desarrollo de software. Un proceso de desarrollo de software es un conjunto de actividades necesarias para transformar los requerimientos del usuario en un sistema de software.

1.6.3.1 CARACTERISTICAS DE LOS PROCESOS UNIFICADOS



a) Manejado por casos de uso:

Un caso de uso representa una pieza de funcionalidad en el sistema que le devuelve al usuario un resultado de valor. Los casos de uso sirven para capturar requerimientos funcionales. Todos los casos de uso de un sistema conforman el modelo de casos de uso.

Se dice que el Proceso Unificado está manejado por casos de uso porque el desarrollador creará modelos de implementación y diseño que harán efectivos a los casos de uso planteados en la etapa de análisis. Cada modelo del proceso tendrá una correspondencia con el modelo de casos de uso.

b) Centrado en la arquitectura:

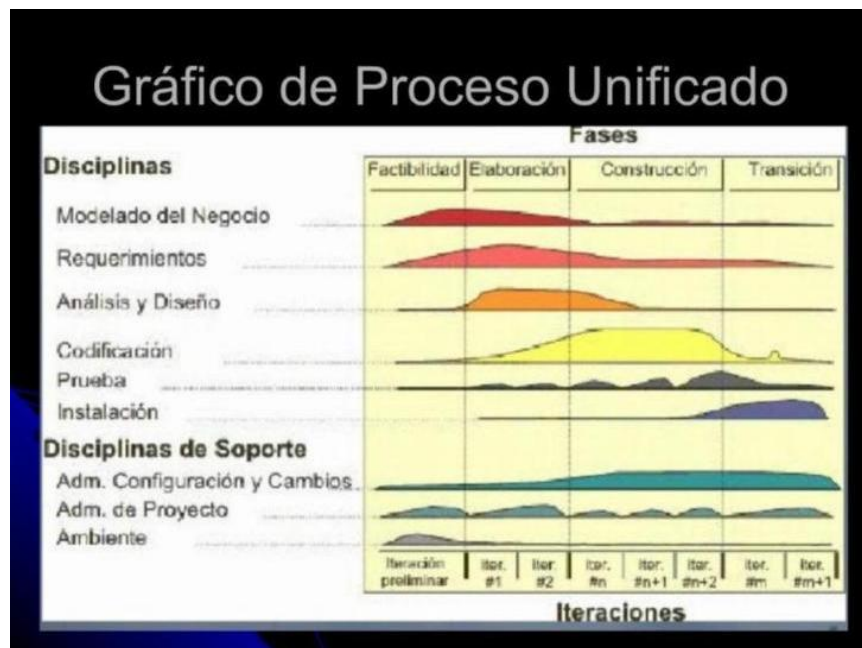
A pesar de que es cierto que los casos de uso manejan el proceso, éstos no lo hacen en soledad, sino que son desarrollados en conjunto con la arquitectura del sistema. Los casos de uso manejan la arquitectura del sistema y la arquitectura del sistema influencia en la selección de los casos de uso. En consecuencia, tanto la arquitectura del sistema como los casos de uso maduran a medida que el ciclo de vida continua.

c) Iterativo e incremental:

Para el PU la vida de un sistema se encuentra dividida en **ciclos**. Cada ciclo termina con un lanzamiento de diferentes modelos del producto.

Cada ciclo consiste de cuatro **fases**: concepción, elaboración, construcción y transición.

Cada fase se encuentra subdividida en **iteraciones**. Al final de cada fase se produce un “**milestone**” o punto de revisión. Una iteración es un mini proyecto que concluye con la entrega de algún documento o implementación interna. Dentro de cada iteración se realizan actividades de captura de requerimientos, análisis, diseño, implementación y prueba.



1.6.3.2 ETAPAS DE IMPLEMENTACIÓN DEL PROCESO UNIFICADO

La obtención de requerimientos de usuario: Los requerimientos para un sistema de software determinan lo que hará el sistema y definen las restricciones de su operación e implementación.

Realización de casos de uso: un **caso de uso** es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software. Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico. Normalmente, en los casos de usos se evita el empleo de jergas técnicas, prefiriendo en su lugar un lenguaje más cercano al usuario final. En ocasiones, se utiliza a usuarios sin experiencia junto a los analistas para el desarrollo de casos de uso.

Modelado de las clases: Los diagramas de clases son diagramas de estructura estática que muestran las clases del sistema y sus interrelaciones (incluyendo herencia, agregación, asociación, etc). Los diagramas de clase son el pilar básico del modelado con UML, siendo utilizados tanto para mostrar lo que el sistema puede hacer (análisis), como para mostrar cómo puede ser construido (diseño). El diagrama de clases de más alto nivel (*main class diagram*), será lógicamente un dibujo de los paquetes que componen el sistema. A su vez cada paquete tendrá un *main class diagram* que muestra las clases del paquete

Las clases se documentan con una descripción de lo que hacen, sus métodos y sus atributos. Las relaciones entre clases se documentan con una descripción de su propósito, su cardinalidad (cuantos objetos intervienen en la relación) y su opcionalidad (cuando un objeto es opcional el que intervenga en una relación).

Diseño y creación de la Base de Datos: Una base de datos se puede definir como un conjunto de información relacionada que se encuentra agrupada ó estructurada.

Desde el punto de vista de la informática, la base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.

Codificación del sistema: es el momento en que el programador comienza a crear las pantallas que contendrá el sistema en un lenguaje de programación específico, este puede ser HTML, JAVA, etc.

Pruebas: Es la etapa en que se comienza a realizar las pruebas necesarias al sistema, corregir los errores que contenga antes de su resultado final.

1.6.4 TECNOLOGÍAS DE APLICACIONES

1.6.4.1 DEFINICIÓN DE APLICACIÓN

En informática, una aplicación es un tipo de programa informático diseñado para facilitar al usuario la realización de un determinado tipo de trabajo. ...

Programa o conjunto de programas diseñados para realizar funciones directamente para un usuario. Las aplicaciones necesitan de un sistema operativo para poder funcionar. Existen varios tipos de aplicaciones: procesadores de texto, bases de datos, hojas de cálculo, correo electrónico, etc.

1.6.4.2 APLICACIÓN DISTRIBUIDA

¿A qué se refiere la distribución?

La distribución se refiere a la construcción de software por partes, a las cuales le son asignadas una serie de responsabilidades dentro de un sistema.

Con la tecnología más avanzada, se define el modelo de aplicaciones distribuidas, una estructura base válida para cualquier aplicación final multipuesto.

Una aplicación distribuida se basa en una **red de servidores**, cada uno con una copia de la aplicación que se comporta de forma idéntica ante el usuario.

La información se reparte inteligentemente entre los "nodos de aplicación" sin que se produzca centralización.

Un usuario puede trabajar desde cualquier puesto, permitiendo traslado de personal sin necesidad de operaciones administrativas.

Seguridad

No hay piezas únicas; un modelo sin piedras angulares asegura la disponibilidad de la aplicación cuando alguno de sus componentes cae.

Si un servidor cae, los puestos que lo usan pueden temporalmente utilizar otro cualquiera de la red, prácticamente en igualdad de condiciones.

Dimensiones

Una aplicación distribuida no tiene límites. Desde dos servidores, hasta los necesarios para cubrir una actividad internacional.

La potencia se distribuye entre todos, este es el principio que hace superior a un equipo de trabajo.

El carácter de aplicación distribuida es una de las características comunes de nuestro modelo de aplicación.

1.6.4.2.1 CLIENTE SERVIDOR

Esta arquitectura consiste básicamente en que un programa -el cliente- que realiza peticiones a otro programa -el servidor- que le da respuesta.

Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa.

Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

Una disposición muy común son los **sistemas multicapa** en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema.

La arquitectura cliente-servidor sustituye a la arquitectura monolítica en la que no hay distribución, tanto a nivel físico como a nivel lógico.

CARACTERÍSTICAS DE LAS APLICACIONES CLIENTE SERVIDOR

CLIENTE

En la arquitectura Cliente Servidor el remitente de una solicitud es conocido como **CLIENTE**. Sus características son:

- 1- Es quien inicia solicitudes o peticiones, tienen por tanto un papel activo en la comunicación (dispositivo **maestro** o **amo**).
- 2- Espera y recibe las respuestas del servidor.
- 3- Por lo general, puede conectarse a varios servidores a la vez.
- 4- Normalmente interactúa directamente con los usuarios finales mediante una interfaz gráfica de usuario.

SERVIDOR

En los sistemas Cliente Servidor el receptor de la solicitud enviada por el Cliente se conoce como **SERVIDOR**. Sus características son:

- 1- Al iniciarse esperan a que lleguen las solicitudes de los clientes, desempeñan entonces un papel pasivo en la comunicación (dispositivo **esclavo**).
- 2- Tras la recepción de una solicitud, la procesan y luego envían la respuesta al cliente.
- 3- Por lo general, aceptan conexiones desde un gran número de clientes (en ciertos casos el número máximo de peticiones puede estar limitado).
- 4- No es frecuente que interactúen directamente con los usuarios finales.

COMPARACIÓN DE LA ARQUITECTURA C/S CON OTRAS ARQUITECTURAS DE RED

a) Comparación con las redes de pares

Las **redes de pares**, también conocidas como redes **par-a-par** o **peer-to-peer** (abreviado con las siglas **P2P**) son otro tipo de arquitectura de red; en ellas cada nodo o elemento del sistema puede actuar al mismo tiempo como cliente y como servidor; cada nodo tiene, por tanto, las responsabilidades y estados de ambos elementos.

b) Comparación con la arquitectura Cliente-Cola-Cliente

Si bien la clásica arquitectura C/S requiere uno de los puntos terminales de comunicación para actuar como un servidor, que puede ser algo más difícil de aplicar, la arquitectura Cliente-Cola-Cliente habilita a todos los nodos para actuar como clientes simples, mientras que el servidor actúa como una cola que va capturando las peticiones de los clientes (un proceso que debe pasar sus peticiones a otro, lo hace a través de una cola, por ejemplo, una consulta a una base de datos, entonces, el segundo proceso conecta con la base de datos, elabora la petición, la pasa a la base de datos, etc.). Esta arquitectura permite simplificar en gran medida la implementación de software. la arquitectura P2P originalmente se basó en el concepto "Cliente-Cola-Cliente".

EJEMPLO PRACTICO DE APLICACIÓN CLIENTE SERVIDOR



La tecnología de diseño de aplicaciones según el modelo cliente-servidor es la más utilizada en el desarrollo de programas flexibles, seguros y mantenibles. Puede llevarse a cabo según diferentes estructuras.

1.6.4.2 APLICACIÓN MULTI-CAPAS

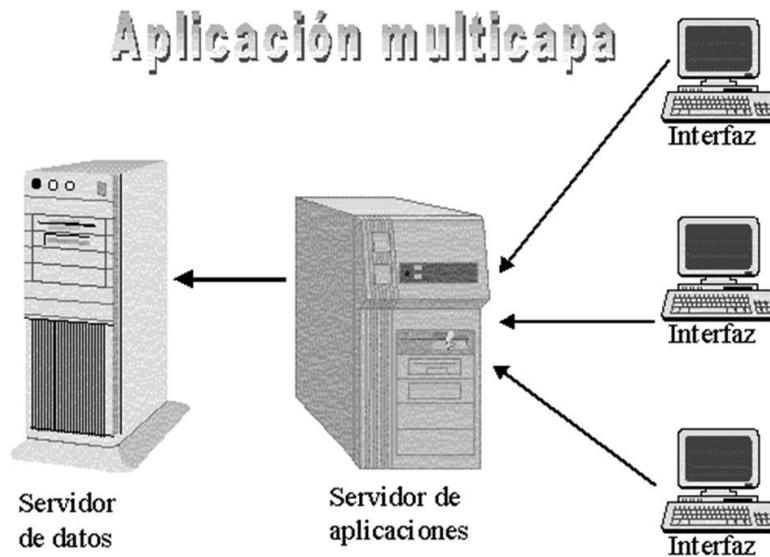
La arquitectura cliente/servidor genérica tiene dos tipos de nodos en la red: clientes y servidores. Consecuentemente, estas arquitecturas genéricas se refieren a veces como arquitecturas de dos niveles o **dos capas**.

Algunas redes disponen de tres tipos de nodos:

1. Clientes que interactúan con los usuarios finales.
2. Servidores de aplicación que procesan los datos para los clientes.

3. Servidores de la base de datos que almacenan los datos para los servidores de aplicación.

Esta configuración se llama una arquitectura de tres-capas.



Ventajas de las arquitecturas n-capas

La ventaja fundamental de una arquitectura **n-capas** comparado con una arquitectura de dos niveles (o una tres-capas con una de dos niveles) es que separa hacia fuera el proceso, eso ocurre para mejorar el balance la carga en los diversos servidores; es más escalable.

Centralización del control: 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 poner al día datos u otros recursos (mejor que en las redes P2P).

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).

Fácil mantenimiento: al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio (o se afectarán mínimamente). Esta independencia de los cambios también se conoce como encapsulación.

Existen tecnologías, suficientemente desarrolladas, diseñadas para el paradigma de C/S que aseguran la seguridad en las transacciones, la amigabilidad del interfaz, y la facilidad de empleo.

Desventajas de las arquitecturas de la n-capas

Pone más carga en la red, debido a una mayor cantidad de tráfico de la red.

Es mucho más difícil programar y probar el software que en arquitectura de dos niveles porque tienen que comunicarse más dispositivos para terminar la transacción de un usuario.

La congestión del tráfico ha sido siempre un problema en el paradigma de C/S. Cuando una gran cantidad de clientes envían peticiones simultáneas al mismo servidor, puede ser que cause muchos problemas para éste (a mayor número de clientes, más problemas para el servidor). Al contrario, en las redes P2P como cada nodo en la red hace también de servidor, cuantos más nodos hay, mejor es el ancho de banda que se tiene.

El paradigma de C/S clásico no tiene la robustez de una red P2P. Cuando un servidor está *caído*, las peticiones de los clientes no pueden ser satisfechas. En la mayor parte de redes P2P, los recursos están generalmente distribuidos en varios nodos de la red. Aunque algunos salgan o abandonen la descarga; otros pueden todavía acabar de descargar consiguiendo datos del resto de los nodos en la red.

El software y el hardware de un servidor son generalmente muy determinantes. Un hardware regular de un ordenador personal puede no poder servir a cierta cantidad de

clientes. Normalmente se necesita software y hardware específico, sobre todo en el lado del servidor, para satisfacer el trabajo. Por supuesto, esto aumentará el coste.

El cliente no dispone de los recursos que puedan existir en el servidor. Por ejemplo, si la aplicación es una Web, no podemos escribir en el disco duro del cliente o imprimir directamente sobre las impresoras sin sacar antes la ventana previa de impresión de los navegadores.

Dirección

Los métodos de dirección en ambientes del servidor de cliente se pueden describir como los siguientes:

Dirección del proceso de la máquina: la dirección se divide como proceso-máquina. Por lo tanto 56-453 indicaría el proceso 56 en la computadora 453.

Servidor de nombres: los servidores de nombres tienen un índice de todos los nombres y direcciones de servidores en el dominio relevante.

Localización de Paquetes: Los mensajes de difusión se envían a todas las computadoras en el sistema distribuido para determinar la dirección de la computadora de la destinación.

Comerciante: Un comerciante es un sistema que pone en un índice todos los servicios disponibles en un sistema distribuido. Una computadora que requiere un servicio particular comprobará con el servicio que negocia para saber si existe la dirección de una computadora que proporciona tal servicio.

Cooperación cliente-servidor

Chained Server

En esta clasificación de servidores se encuentran aquellos que de vez en cuando se comportan como clientes, ya sea de manera efímera o por mucho tiempo.

Temporal

Son servidores que por motivos de gestión o para ejecutar cierta parte del proceso de transacción, se comportan como clientes ya sea para solicitar la información requerida al servidor correspondiente o simplemente como un enlace hacia el "destino" de la petición. Sólo realiza la transacción solicitada y vuelve a su estado normal una vez terminada la misma.

Largo Plazo

Aquellos en los cuales realiza varias transacciones del por mucho más tiempo que la anterior, y se orienta más a la aplicación para la que fue dirigida. ejem: servidor. de WEB, etc.

Múltiple Server

Para que un proceso se realice de la mejor manera, es preferible utilizar terminales distintos realizando la misma tarea, a centralizar los recursos y que con más hardware/software se realice la misma tarea. Con la ejecución de múltiples servidores el procesamiento es más rápido, el tiempo de respuesta es descentralizado y se incrementa la confiabilidad.

Cooperación de procesos paralelos

El mismo proceso se ejecuta simultáneamente (sistemas redundantes).

Cooperación de base de datos

Si se requiere de cierta información ya existente, por qué crearla de nuevo, simplemente interactúa y aprovecha la información ya creada.

Organización del Código

La organización del código, su estructuración según el modelo de orientación al objeto, es fundamental si queremos conectar los diferentes componentes de forma rápida y fiable. Por lo tanto, el diseño de los elementos del proyecto (la solución en el caso de .NET) es algo DETERMINANTE.

1) La clase de acceso a los datos

En una aplicación de este tipo siempre tendremos una clase principal que será la encargada de proveer todos los servicios de acceso a los datos. En ningún otro sitio del programa se debe acceder directamente a las bases de datos. Este componente encapsula, aísla los datos del resto de la aplicación.

2) Un programa de pruebas (en modo consola o en modo visual) que permita verificar el funcionamiento de la clase de acceso a los datos.

En este programa, mediante un menú, programaremos las funciones que nos permitan llamar a cada uno de los métodos y propiedades de la clase de acceso a datos. Además de un buen banco de pruebas, esta aplicación constituye un elemento fundamental en la documentación de la misma, permite ver con claridad la forma de utilizar cada uno de sus elementos.

Además, cuando se trabaja en grupo, el encargado de cada componente debe generar un proyecto de pruebas sencillo que le permita depurarlo y ponerlo en marcha antes de juntarlo con el resto de módulos, y servirá de clarificación para el resto de compañeros de trabajo.

3) Clases “satélite” de la aplicación

Una aplicación de acceso a datos puede trabajar directamente con objetos de tipo DataSet, pero en muchas ocasiones es preferible el diseño de clases específicas para las entidades que viajan entre cliente y servidor.

Estas clases suelen coincidir con las entidades del modelo de la base de datos. Las relaciones 1 – N de la base de datos se implementan habitualmente como ArrayList dentro de la clase.

Generalmente, queda mucho mejor documentado y claro un esquema que trabaje con clases

en lugar de un esquema que trabaje con DataSets. Si trabajamos con DataSets, para cada caso deberemos documentar el modelo entidad/relación del mismo.

4) El prototipo visual

Si la aplicación va a realizarse en el entorno de Aplicaciones para Windows, debemos realizar un prototipo funcional, que permita ver cómo quedará el diseño final, sin implementar las funciones de acceso a datos. Únicamente ha de permitir la navegación sobre los diferentes escenarios que tendremos.

Según el tipo de aplicación, puede ser necesario el desarrollo de varios prototipos, dirigidos a diferentes perfiles de usuario.

En ningún caso se debe mezclar este prototipo con la gestión de los datos. La gestión de los datos es única, y los prototipos utilizan aquellas partes de la clase de gestión que sea necesario.

5) Los datos globales de la aplicación cliente

En la aplicación cliente tendremos una serie de datos globales que van a ser utilizados en todo el programa. La conexión con la clase de acceso a datos es un ejemplo. Dado que programamos en un entorno de orientación al objeto, NO TENEMOS VARIABLES GLOBALES. Con este modelo, la clase de inicio debería tener toda la información global, y pasarla a cada uno de los elementos que se van activando. Para evitar esta situación, construiremos una clase ESTÁTICA en el proyecto, donde almacenaremos aquellos datos que deban ser utilizados globalmente.

6) La estructura de proyectos

Para separar correctamente las distintas partes del proyecto, en nuestra solución deberemos tener como mínimo los siguientes elementos:

- **Una biblioteca de clases.** Contendrá la clase de acceso a datos y las clases satélite (si las hay).
- **Un proyecto de pruebas.** Permitirá probar cada método/propiedad de la clase de datos (no importa si es en modo consola).
- **Un proyecto cliente.** Será la aplicación que utilizarán los usuarios para trabajar (visual). Si la aplicación es local, con esto será suficiente. Si la aplicación trabaja con remoting, deberemos implementar además un nuevo proyecto que contenga el programa de activación del objeto remoto. Será el programa servidor. Si la aplicación trabaja con Web services, deberemos implementar un proyecto Web Service, que incruste un objeto del tipo de acceso a datos, y que publique los métodos web correspondientes. Estos métodos web deberían estar diseñados polimórficamente iguales (i.e. con el mismo nombre y parámetros) que los de la clase de acceso a datos. Si la aplicación es una página web, tendremos un proyecto Web, que incrustará un objeto del tipo de acceso a datos, y que implementará toda la funcionalidad. Este será realmente el proyecto cliente, ya que en las máquinas cliente sólo tendremos el recurso del navegador.

En este tutorial veremos paso a paso la generación de cada uno de estos componentes, con las consideraciones de diseño importantes a tener en cuenta.

1.6.4.2.3 APLICACIÓN WEB

En la ingeniería 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, 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. Existen aplicaciones como los webmails, wikis, weblogs, tiendas en línea que son ejemplos bien conocidos de aplicaciones web.

Es importante 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, participar en juegos diversos y acceder a gestores de base de datos de todo tipo.

Antecedentes

En los primeros tiempos de la computación cliente-servidor, cada aplicación tenía su propio programa cliente que servía como interfaz de usuario que tenía que ser instalado por separado en cada ordenador personal de cada usuario. El cliente realizaba peticiones a otro programa -el servidor- que le daba respuesta. Una mejora en el servidor, como parte de la aplicación, requería normalmente una mejora de los clientes instalados en cada ordenador personal, añadiendo un coste de soporte técnico y disminuyendo la productividad.

A diferencia de lo anterior, las aplicaciones web generan dinámicamente una serie de páginas en un formato estándar, como HTML o XHTML, que soportan por los navegadores web comunes. Se utilizan lenguajes interpretados en el lado del cliente, tales como JavaScript, para añadir elementos dinámicos a la interfaz de usuario. Generalmente cada página web en particular 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.

Interfaz

Las interfaces web tienen ciertas limitaciones en las funcionalidades que se ofrecen al usuario. Hay funcionalidades comunes en las aplicaciones de escritorio como dibujar en la pantalla o arrastrar-y-soltar que no están soportadas por las tecnologías web estándar. Los desarrolladores web generalmente utilizan lenguajes interpretados o script en el lado del cliente para añadir más funcionalidades, especialmente para ofrecer una experiencia interactiva que no requiera recargar la página cada vez (lo que suele resultar molesto a los usuarios). Recientemente se han desarrollado tecnologías para coordinar estos lenguajes con tecnologías en el lado del servidor, como por ejemplo PHP.

Como ejemplo, AJAX, es una técnica de desarrollo web que usa una combinación de varias tecnologías.

Consideraciones Técnicas

Una ventaja significativa es que las aplicaciones web deberían funcionar igual independientemente de la versión del sistema operativo instalado en el cliente. En vez de crear clientes para Windows, Mac OS X, GNU/Linux, y otros sistemas operativos, la aplicación web se escribe una vez y se ejecuta igual en todas partes. Sin embargo, hay aplicaciones inconsistentes escritas con HTML, CSS, DOM y otras especificaciones para navegadores web que pueden causar problemas en el desarrollo y soporte de las aplicaciones web. Adicionalmente, la posibilidad de los usuarios de personalizar muchas de las características de la interfaz (tamaño y color de fuentes, tipos de fuentes, inhabilitar JavaScript) puede interferir con la consistencia de la aplicación web.

Otra aproximación es utilizar Adobe Flash Player o Java applets para desarrollar parte o toda la interfaz de usuario. Como casi todos los navegadores incluyen soporte para estas tecnologías (usualmente por medio de plug-ins), las aplicaciones basadas en Flash o Java pueden ser implementadas con aproximadamente la misma facilidad. Dado que ignoran las configuraciones de los navegadores, estas tecnologías permiten más control sobre la

interfaz, aunque las incompatibilidades entre implementaciones Flash o Java puedan crear nuevas complicaciones. Por las similitudes con una arquitectura cliente-servidor, con un cliente "no ligero", existen discrepancias sobre el hecho de llamar a estos sistemas "aplicaciones web"; un término alternativo es "Aplicación Enriquecida de Internet".

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 Servlets o ASP, ASP.NET, CGI, ColdFusion, embPerl, Python (programming language) o Ruby on Rails) constituye la capa de en medio. Por último, una base de datos constituye la tercera y última capa.

El navegador web manda peticiones a la capa de en medio que ofrece servicios valiéndose de consultas y actualizaciones a la base de datos y a su vez proporciona una interfaz de usuario.

Uso empresarial

Una estrategia que está emergiendo para las empresas proveedoras de software consiste en proveer acceso vía web al software. Para aplicaciones previamente distribuidas, como las aplicaciones de escritorio, se puede optar por desarrollar una aplicación totalmente nueva o simplemente por adaptar la aplicación para ser usada con una interfaz web. Estos últimos programas permiten al usuario pagar una cuota mensual o anual para usar la aplicación, sin necesidad de instalarla en el ordenador del usuario. Las compañías que siguen esta estrategia se denominan Proveedores de Aplicaciones de Servicio (ASP por sus siglas en inglés), un modelo de negocio que está atrayendo la atención de la industria del software.

Lenguajes De Programación

Existen numerosos lenguajes de programación empleados para el desarrollo de Aplicaciones Web, entre los que destacan:

PHP

ASP/ASP.NET

Java, con sus tecnologías Java Servlets y JavaServer Pages (JSP)

Perl

Ruby

Python

HTML

XML

ASP no es un lenguaje de programación en sí mismo, sino una arquitectura de desarrollo web en la que se pueden usar por debajo distintos lenguajes (por ejemplo VB.NET o C# para ASP.NET o VBScript/JScript para ASP).

Ejemplos de aplicaciones web

eyeOS por Equipo eyeOS. Escritorio virtual

Flickr por Ludicorp. Administrador de fotografías.

Gmail por Google. Correo electrónico.

SugarCRM CRM

1.6.5 ¿QUÉ ES UN SISTEMA DE INVENTARIOS?

Un sistema de inventarios es un conjunto de normas, métodos y procedimientos aplicados de manera sistemática para planificar y controlar los materiales y productos que se emplean en una organización. Este sistema puede ser manual o automatizado. Para el control de los costos, elemento clave de la administración de cualquier empresa, existen sistemas que permiten estimar los costos de las mercancías que son adquiridas y luego procesadas o vendidas.

1.6.6 SOFTWARE PROPIETARIO Y SOFTWARE LIBRE

El término **software propietario, privativo o de código cerrado** hace referencia a cualquier programa informático en el que los usuarios tienen limitadas las posibilidades de usarlo, modificarlo o redistribuirlo (con o sin modificaciones), o cuyo código fuente no está disponible o el acceso a éste se encuentra restringido.

En el software de código cerrado una persona física o jurídica (compañía, corporación, fundación, etc.) posee los derechos de autor sobre un software lo que le da la posibilidad de controlar y restringir los derechos del usuario sobre su programa. Esto implica por lo general que el usuario sólo tiene derecho a ejecutar el software y no dispone de acceso a su código fuente o aún teniendo acceso a él no tiene derecho a modificarlo ni distribuirlo.

De esta manera, un software sigue siendo de código cerrado aún si el código fuente es hecho público, cuando se mantiene la reserva de derechos sobre el uso, modificación o distribución (por ejemplo, la versión comercial de SSH o el programa de licencias shared source de Microsoft).

Software libre, (en inglés *free software*, aunque en realidad esta denominación también puede significar gratis, y no necesariamente libre, por lo que se utiliza el hispanismo **libre software** también en inglés) es la denominación del software que respeta la libertad de los usuarios sobre su producto adquirido y, por tanto, una vez obtenido puede ser usado, copiado, estudiado, cambiado y redistribuido libremente. Según la *Free Software Foundation*, el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software; de modo más preciso, se refiere a cuatro libertades de los usuarios del software: la libertad de usar el programa, con cualquier propósito; de estudiar el funcionamiento del programa, y adaptarlo a las necesidades; de distribuir copias, con lo cual se puede ayudar a otros, y de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie (para la segunda y última libertad mencionadas, el acceso al código fuente es un requisito previo).

El software libre suele estar disponible gratuitamente, o al precio de costo de la distribución a través de otros medios; sin embargo no es obligatorio que sea así, por lo tanto no hay que asociar software libre a "software gratuito" (denominado usualmente freeware), ya que, conservando su carácter de libre, puede ser distribuido comercialmente ("software comercial"). Análogamente, el "software gratis" o "gratuito" incluye en ocasiones el código fuente; no obstante, este tipo de software *no es libre* en el mismo sentido que el software libre, a menos que se garanticen los derechos de modificación y redistribución de dichas versiones modificadas del programa.

CUADRO COMPARATIVO DE LAS VENTAJAS DE UN SOFTWARE PROPIETARIO Y UN SOFTWARE LIBRE

SOFTWARE PROPIETARIO	SOFTWARE LIBRE
<p><i>1. Propiedad y decisión de uso del software por parte de la empresa</i></p> <p>El desarrollo de la mayoría de software requiere importantes inversiones para su estudio y desarrollo. Este esfuerzo, de no ser protegido se haría en balde, puesto que la competencia se podría apropiar inmediatamente del producto una vez finalizado, para sus propios fines. Esto garantiza al productor ser compensado por la inversión, fomentando así el continuo desarrollo.</p> <p><i>2. Soporte para todo tipo de hardware</i></p> <p>Refiriéndonos por supuesto solo al mercado</p>	<p>1. Económico (más de mil millones de euros en licencias de Microsoft en España anuales)</p> <p>El bajo o nulo coste de los productos libres permiten proporcionar a las PYMES servicios y ampliar sus infraestructuras sin que se vean mermados sus intentos de crecimiento por no poder hacer frente al pago de grandes cantidades en licencias.</p> <p>La práctica totalidad de los concursos para desarrollo de software para la administración pública pasan por compatibilizar con productos de la factoría de Microsoft, por lo que garantiza la perpetuación e ingresos hacia Microsoft y</p>

<p>del sistema operativo mayoritario, que es Microsoft Windows, y no al resto de sistemas operativos de tipo Unix, que es minoritario. Se da, que el actual dominio de mercado invita a los fabricantes de dispositivos para ordenadores personales a producir drivers o hardware solo compatible con Windows. Por lo que la elección del sistema operativo de Microsoft tiene garantizado un soporte de hardware seguro.</p> <p>Aunque siendo justos, habría que decir que dicho soporte por parte de los fabricantes se limita generalmente al sistema operativo actual. Las siguientes versiones de sistemas operativos no se suelen soportar, a fin de vender nuevos dispositivos.</p> <p>3. Mejor acabado de la mayoría de aplicaciones</p> <p>El desarrollador de software propietario, generalmente, da un mejor acabado a las aplicaciones en cuestiones, tanto de estética, como de usabilidad de la aplicación. Aunque muchas aplicaciones de software libre tienen un excelente acabado, aquí se nota de forma especial el cuidado y el esfuerzo del desarrollador de software propietario. Al fin y al cabo gana dinero directamente con su producto y debe</p>	<p>no favorece a las empresas locales que pudieran ofrecer productos equivalentes.</p> <p>Además de la reducción de costes por uso de software libre. ¿Qué podrían aportar esas inversiones si los beneficiados fuesen empresas del propio estado, en lugar de mandar a una compañía extranjera esas enormes cantidades de dinero?</p> <p>2. Libertad de uso y redistribución</p> <p>Las licencias de software libre existentes permiten la instalación del software tantas veces y en tantas máquinas como el usuario desee.</p> <p>3. Independencia tecnológica</p> <p>El acceso al código fuente permite el desarrollo de nuevos productos sin la necesidad de desarrollar todo el proceso partiendo de cero. El secretismo tecnológico es uno de los grandes frenos y desequilibrios existentes para el desarrollo en el modelo de propiedad intelectual.</p> <p>4. Fomento de la libre competencia al basarse en servicios y no licencias</p> <p>Uno de los modelos de negocio que genera</p>
--	---

<p>demostrar por qué su producto es una mejor opción.</p>	<p>el software libre es la contratación de servicios de atención al cliente. Este sistema permite que las compañías que den el servicio compitan en igualdad de condiciones al no poseer la propiedad del producto del cual dan el servicio.</p>
<p>4. Las aplicaciones número uno son propietarias</p>	<p>Esto, además, produce un cambio que redundará en una mayor atención al cliente y contratación de empleados, en contraposición a sistemas mayoritariamente sostenidos por la venta de licencias y desatención del cliente.</p>
<p>Obviamente, el actual dominio de mercado no sólo interesa a los fabricantes de hardware, sino que también a los de software. Algunas de las aplicaciones más demandadas son, según ámbitos: Microsoft Office, Nero Burning Rom, 3DStudio, etc...</p>	<p>5. Soporte y compatibilidad a largo plazo</p>
<p>5. El ocio para ordenadores personales está destinado al mercado propietario</p>	<p>Este punto, más que una ventaja del software libre es una desventaja del software propietario, por lo que la elección de software libre evita este problema. Al vendedor, una vez ha alcanzado el máximo de ventas que puede realizar de un producto, no le interesa que sus clientes continúen con él. La opción es sacar un nuevo producto, producir software que emplee nuevas tecnologías solo para éste y no dar soporte para la resolución de fallos al anterior, tratando de hacerlo obsoleto por todos los medios, pese a que este pudiera cubrir perfectamente las necesidades de muchos de sus usuarios.</p>
<p>Los desarrolladores de juegos tienen sus miras en el mercado más lucrativo, que es el de las consolas y en última instancia el de los ordenadores personales. Además, en estos últimos la práctica totalidad de títulos benefician a Microsoft Windows.</p>	<p>6. Menor necesidad de técnicos especializados</p>
<p>6. Menor necesidad de técnicos especializados</p>	<p>El mejor acabado de la mayoría de sistemas de software propietario y los estándares de facto actuales permiten una simplificación del tratamiento de dichos sistemas, dando acceso a su uso y administración, a técnicos</p>

que requieren una menor formación, reduciendo costes de mantenimiento.

Esto ciertamente es una ventaja de cara a la implantación de dichos sistemas, aunque no hay que olvidar que el menor conocimiento del sistema redonda en la imposibilidad de rodear las herramientas de simple manejo para solventar errores o ampliar las posibilidades de las mismas.

7. Mayor mercado laboral actual

Como muchas de las ventajas restantes se deben al dominio del mercado, no hay que ignorar que cualquier trabajo relacionado con la informática pasará, en casi todos los casos, por conocer herramientas de software propietario.

8. Mejor protección de las obras con copyright

Las obras protegidas por copyright se ven beneficiadas por mecanismos anti copia, como el DRM y otras medidas, que palian o dificultan en cierto grado la piratería. El fenómeno de la piratería, está en constante debate sobre a quién perjudica y quién beneficia realmente (véanse cuáles son las empresas que mayores beneficios han

Véanse diferentes versiones de Windows que dejan de ser soportadas por Microsoft o software de grabación que no admite nuevos modelos de grabadoras ópticas sin una actualización, aún cuando la grabadora nueva emplee el mismo mecanismo de grabación que la antigua.

6. Formatos estándar

Los formatos estándar permiten una interoperabilidad más alta entre sistemas, evitando incompatibilidades. Los estándares de facto son válidos en ocasiones para lograr una alta interoperabilidad si se omite el hecho que estos exigen el pago de royalties a terceros y por razones de mercado expuestas en el anterior punto no interesa que se perpetúen mucho tiempo.

Los formatos estándares afectan a todos los niveles. Un ejemplo lo estamos viendo en los documentos emitidos por las administraciones públicas en distintos formatos y versiones, que producen retrasos y dificultades en el acceso adecuado a la información para las mismas administraciones y para sus usuarios. Algunas administraciones de la unión europea ya están dando el paso a formatos

obtenido, a pesar de ser las más pirateadas y ofreciendo la competencia en ocasiones mejores productos), aunque eso, por supuesto, es otra discusión que merece ser tratada aparte.

9. Unificación de productos

Una de las ventajas más destacables del software propietario es la toma de decisiones centralizada que se hace en torno a una línea de productos, haciendo que no se desvíe de la idea principal y generando productos funcionales y altamente compatibles.

Aquí, el software libre tiene una clara desventaja, al ser producido y tomadas las decisiones por un exceso de grupos y organismos descentralizados que trabajan en líneas paralelas y no llegan muchas veces a acuerdos entre ellos. Esto ocasiona que en algunas ocasiones haya un gran caos a programadores y usuarios finales que no saben que vías tomar. Además genera productos cuya compatibilidad deja bastante que desear.

abiertos como ODF (Open Document Format).

7. Sistemas sin puertas traseras y más seguros

El acceso al código fuente permite que tanto hackers como empresas de seguridad de todo el mundo puedan auditar los programas, por lo que la existencia de puertas traseras es ilógica ya que se pondría en evidencia y contraviene el interés de la comunidad que es la que lo genera.

8. Corrección más rápida y eficiente de fallos

El funcionamiento e interés conjunto de la comunidad ha demostrado solucionar más rápidamente los fallos de seguridad en el software libre, algo que desgraciadamente en el software propietario es más difícil y costoso. Cuando se notifica a las empresas propietarias del software, éstas niegan inicialmente la existencia de dichos fallos por cuestiones de imagen y cuando finalmente admiten la existencia de esos bugs tardan meses hasta proporcionar los parches de seguridad.

9. Métodos simples y unificados de gestión de software

Actualmente la mayoría de distribuciones de Linux incorporan alguno de los sistemas que unifican el método de instalación de programas, librerías, etc. por parte de los usuarios. Esto llega a simplificar hasta el grado de marcar o desmarcar una casilla para la gestión del software, y permiten el acceso a las miles de aplicaciones existentes de forma segura y gratuita a la par que evitan tener que recurrir a páginas web de dudosa ética desde las que los usuarios instalan sin saberlo spyware o virus informáticos en sus sistemas. Este sistema de acceso y gestión del software se hace prácticamente utópico si se extrapola al mercado propietario.

10. Sistema en expansión

Las ventajas especialmente económicas que aportan las soluciones libres a muchas empresas y las aportaciones de la comunidad han permitido un constante crecimiento del software libre, hasta superar en ocasiones como en el de los servidores web, al mercado propietario.

	<p>El software libre ya no es una promesa, es una realidad y se utiliza en sistemas de producción por algunas de las empresas tecnológicas más importantes como IBM, SUN Microsystems, Google, Hewlett-Packard, etc. Paradójicamente, incluso Microsoft, que posee sus propias herramientas, emplea GNU Linux en muchos de sus servidores. Podemos augurar sin lugar a dudas un futuro crecimiento de su empleo y una consolidación bien merecida.</p>
--	--

1.6.7 TECNOLOGIAS UTILIZADAS

- **JAVA SE**

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. Su principal destino era la ejecución de aplicaciones vía Internet de manera segura y que fuera de manera local, no remota como lo hace CGI.

Java SE O Standard Edition: Anteriormente conocido como J2SE. Versión de Java Estandar. Dispone de algunas APIs más usadas e imprescindibles.

- **MySQL**

MySQL es la base de datos open source más popular y, posiblemente, mejor del mundo. Su continuo desarrollo y su creciente popularidad está haciendo de MySQL un competidor cada vez más directo de gigantes en la materia de las bases de datos como Oracle. Es también un sistema de administración de bases de datos (*Database Management System, DBMS*) para bases de datos relacionales. Así, MySQL no es más que una aplicación que permite gestionar archivos llamados de bases de datos o, lo que es lo mismo, u

- **NETBEANS**

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados *módulos*. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

II

MARCO

EMPIRICO

2.1 METODOLOGIA

Para poder desarrollar el proyecto que se ha recurrido a un estudio y análisis de la situación actual en la que se encuentra el Hospital Nacional de Chalatenango, partiendo de los resultados obtenidos se creará un sistema que permita tener un mejor control en cuanto al mantenimiento, asignación y estado en el que se encuentran los equipos informáticos. Para el desarrollo de este proyecto se ha tomado como base el Proceso Unificado, que parte con la obtención de requerimientos de usuario, realización de casos de uso, el modelado de las clases, diseño de la Base de Datos, creación de Base de Datos, codificación del sistema y pruebas.

2.2 PORCEDIMIENTO PARA LA OBTENCIÓN DE DATOS

Para la recolección de los datos se utilizó a la técnica de la entrevista (Ver nexos A), realizada a la encargada del área de informática del hospital, para conocer más acerca del entorno en el que trabajará el sistema y las características técnicas de este, de acuerdo a los recursos existentes. Además para la obtención de recolección de datos se realizó la observación directa de las áreas involucradas en el sistema. Por último recurrimos a la obtención de documentos (Ver Nexos B y C) facilitados por la encargada del departamento de Informática.

2.3 MARCO OPERATIVO

2.3.1 ESPECIFICACION DE REQUISITOS DE SOFTWARE

HOSPITAL NACIONAL DE CHALATENANGO DR. LUIS EDMUNDO VASQUEZ.
SISTEMA DE CONTROL Y MANTENIMIENTO DE EQUIPO INFORMATICO ESPECIFICACION DE REQUISITOS DE SOFTWARE VERSION: 2.0

DESCRIPCIÓN GLOBAL

1 Perspectiva del Sistema

El sistema de control y mantenimiento de equipos informáticos interactuará inicialmente con el departamento de servicio informático quien será el único ente en controlar la alimentación del sistema, y como segunda instancia la empresa encargada de los mantenimientos a dichos equipos, para la emisión de reportes al departamento de servicio informático y la dirección..

2 Funciones del Sistema

El sistema tendrá funciones como altas y bajas de equipos informáticos, control de mantenimiento correctivo y preventivo, periféricos, hardware y un formulario de servicio...

- **Altas y Bajas:** está relacionado con el registro de equipos informáticos existentes, los datos técnicos de cada equipo, intervalos de mantenimiento a cada equipo.
- **Mantenimientos Correctivo:** Podremos almacenar los problemas que posee algún equipo y los cambios que se le han efectuado a nivel de hardware.
- **Mantenimiento Preventivo:** Permite controlar el tiempo que se realizo el ultimo mantenimiento, controla y avisa de cuando le toca hacer el próximo, controla la empresa q realizo el mantenimiento y el costo del mismo.
- **Periféricos:** almacenará el estado de cada periférico (mouse, teclado, monitor etc.). La condición de este es que se actualizará manualmente el cambio de alguno de estos periféricos.
- **Hardware:** se podrá saber con que tipo de hardware cuentan las diferentes maquinas.
- **Formulario de Servicio:** Podrá mostrar lo que se realizo en el mantenimiento tanto preventivo como correctivo.

3. Alojamiento del Sistema

El sistema se encontrara en el Departamento de Informática, (pendiente)

REQUISITOS ESPECIFICOS

1. Requisitos Funcionales

1. REQ01 Registro del Equipo: El usuario podrá registrar los equipos informáticos y guardarlos mediante el sistema en cuestión, los campos de estos registros deberán ser, como mínimo:
 - Datos generales: marca, modelo, año de adquisición, tipo de computadora (Laptop, de escritorio), Procesador, Motherboard, Disco duro, Memoria RAM.
 - Datos de Impresor: Numero de inventario, marca, modelo, serie, tipo de conector, tipo de impresora, estado, observación.
 - Datos de Monitor: Numero de inventario, marca, modelo, serie, tipo de monitor, tamaño de monitor, estado, procedencia del equipo y observación.
 - Datos del Teclado: Numero de inventario, marca, modelo, serie, idioma, tipo de conectores, estado, procedencia de equipo, observación.
 - Datos de regulador de voltaje: Numero de inventario, marca, modelo, serie, tipo, potencia, estado, procedencia del equipo y tipo de mantenimiento.
 - Datos de Programas Instalados: Tipo de software, nombre, versión, idioma, tipo de licencia, estado (vigente o expirado).
2. REQ02 Visibilidad de las descripciones del equipo: El usuario podrá ver las descripciones con las que dispone determinado equipo para poder realizar la operación correspondiente de acuerdo a ello.
3. REQ03 Selección de descripciones: Se podrá obtener la descripción de los equipos almacenados en la base de datos mediante consultas generales y personalizadas, tales como consulta por departamento, encargado, por fecha de adquisición, código de inventario.
4. REQ04 Registro de Mantenimiento: Permite almacenar los datos de los diferentes mantenimientos que se le realizan a los equipos, los datos mínimos a almacenar son los siguientes: Tipo de mantenimiento (correctivo o preventivo) fecha, Nombre de la empresa que realizó el mantenimiento, tareas de mantenimiento realizadas como: cambio de periféricos, limpieza, cambio de piezas defectuosas.
5. REQ5: Control de Mantenimiento: Alertara al usuario de forma automática cuando a un equipo se le aproxime o se le haya vencido la fecha de mantenimiento.
6. REQ6: Consulta de mantenimiento: Permite al usuario consultar el tipo de mantenimiento que ha sido aplicado y el estado en el que se encuentra cada equipo informático.
7. REQ7: Generación de ficha individual de equipo e inventario de software: Permite obtener de forma impresa los datos generales de cada equipo informático con el que cuenta el hospital, según el formato establecido por el ministerio de salud (ver anexo B).

2. Requisitos de Interfaces

1. REQ8 Interfaces del usuario: Se podrá comunicar con el usuario para aprovechar los requisitos del sistema, el usuario indicará al sistema las operaciones que debe realizar e introducirá los datos que el sistema le pida.
2. REQ9: Interfaces de Software: El sistema se construirá con una sola aplicación que estará instalada en un equipo único. Con una interfaz gráfica de fácil manejo. Para ingresar cada usuario tendrá un nombre y clave registrados en el equipo.
3. REQ9A: Interfaz de Cliente Servidor: El sistema que se implementará será de tipo cliente servidor, cuya base de datos estará instalada en un Servidor Local.

3. Requisitos de Rendimiento

1. REQ10 Tiempo de respuesta: La respuesta que dará el sistema con respecto a la petición del usuario deberá ser en tiempo real.

4. Requisitos de desarrollo

1. REQ11 Proceso de Desarrollo: El proceso de desarrollo de software para este sistema será el Proceso Unificado de Desarrollo de Software, que consiste en cuatro fases que son: iniciación, elaboración, construcción e implementación y cuatro disciplinas a saber: análisis, diseño, codificación y prueba, mismas que nos ayudarán a simplificar la planeación de actividades.
2. REQ12 La base de datos en que se implementara el sistema es una base de datos relacionar MySQL porque no tiene ningún costo para la empresa.
3. REQ13 El lenguaje en el que se desarrollara el sistema es JAVA.SE por su capacidad de escalabilidad y porque también no contiene ningún costo para la empresa.

5. Atributos

1. REQ14 Portabilidad: El sistema no contará con el acceso desde otras computadoras ya que esta aplicación no estará instalada en una red.
2. REQ15 Mantenibilidad: El sistema deberá ser diseñado para que su mantenimiento sea fácil, y de esta manera pueda ser ampliado y corregido en caso de ser necesario.
3. REQ16 Disponibilidad: El sistema tendrá disponibilidad a la hora que sea conveniente para el usuario.

<p>Elaborado por:</p> <ol style="list-style-type: none"> 1. Amadeo Ernesto Velásquez Núñez. 2. Christian Alfredo Vásquez Cardoza. 3. Julio César Avalos Barrera.
<p>Autorizado por Jefe De Departamento de Informática: Licenciada Sandra de Manzano.</p>
<p>Autorizado por Director del Hospital: Dr.</p>

2.3.2 CASOS DE USOS

Diagrama de casos de usos

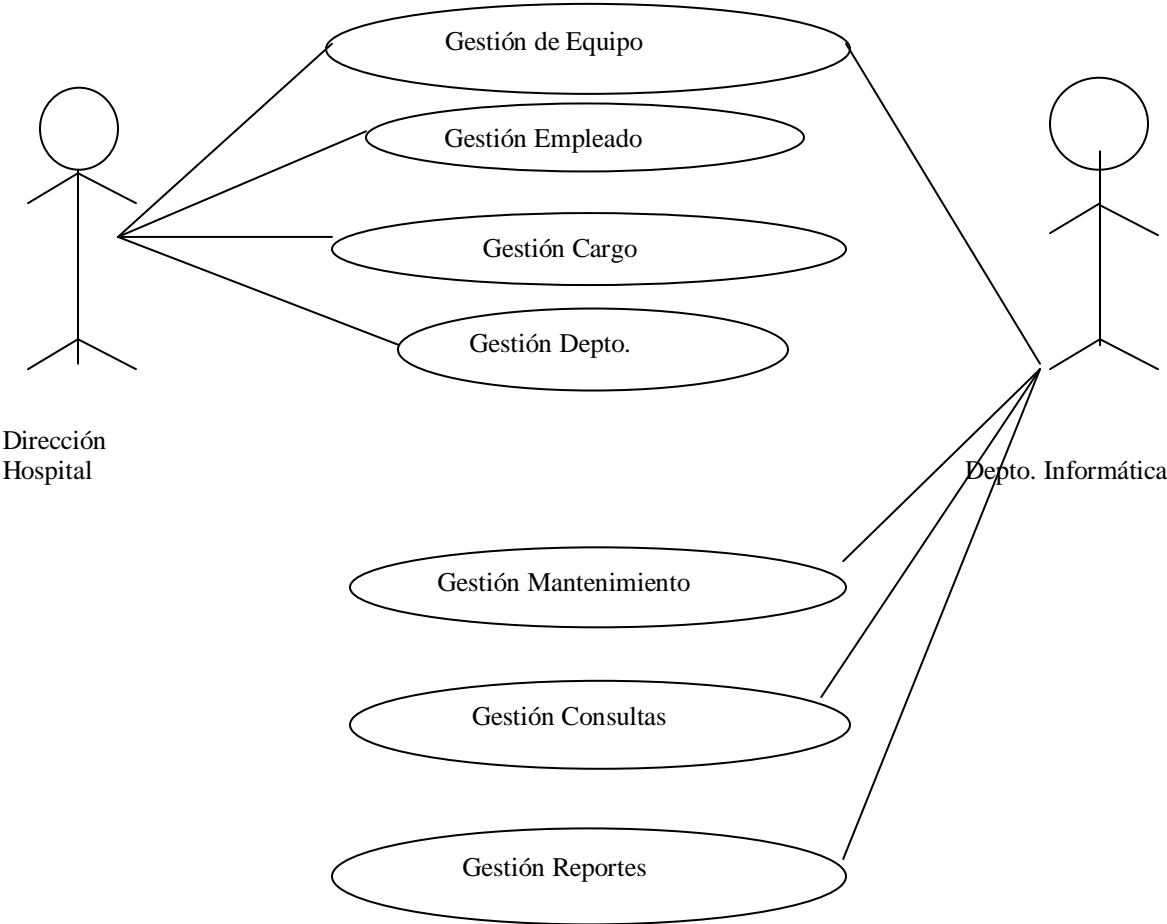


Diagrama de Gestión de Equipo

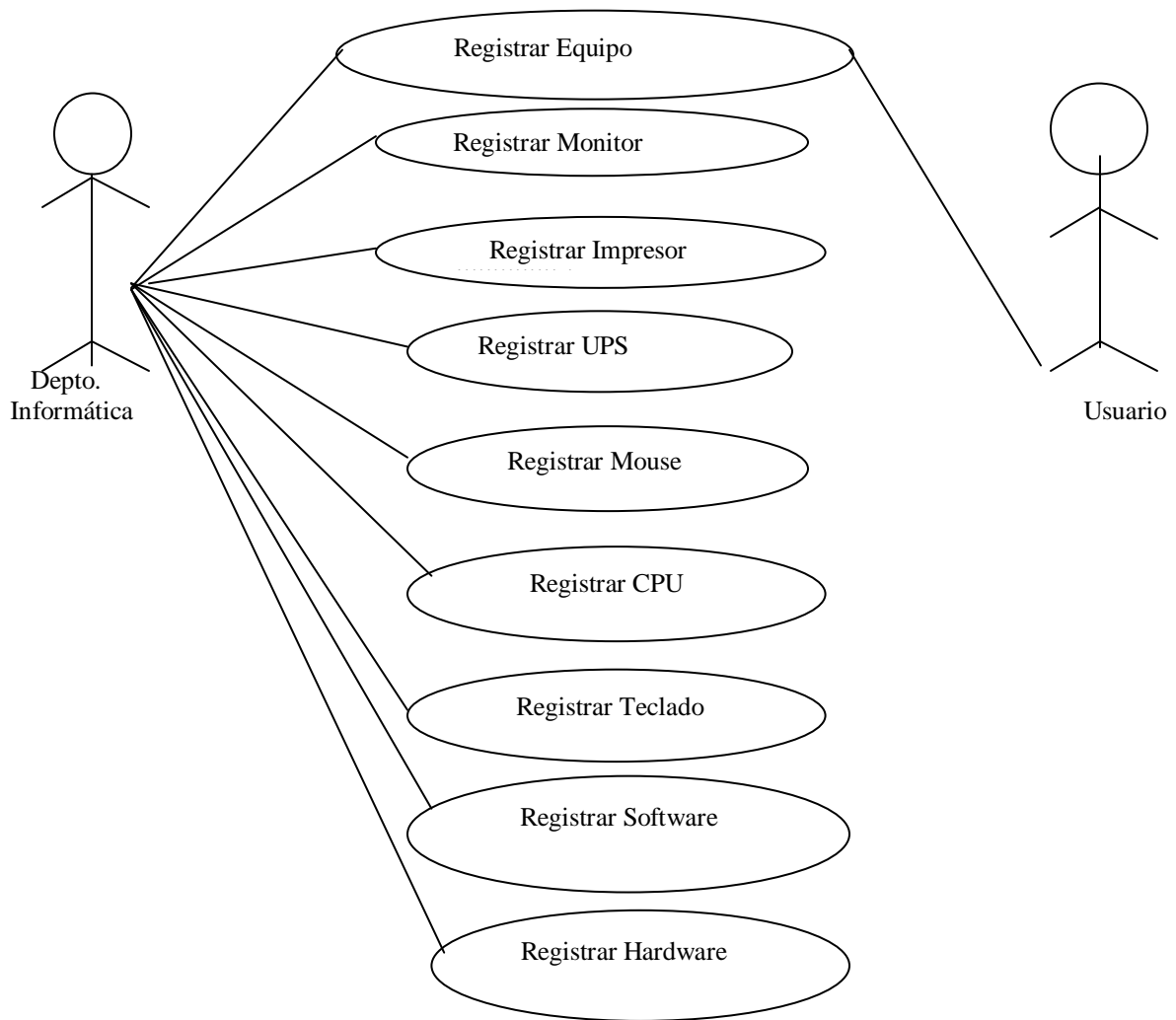
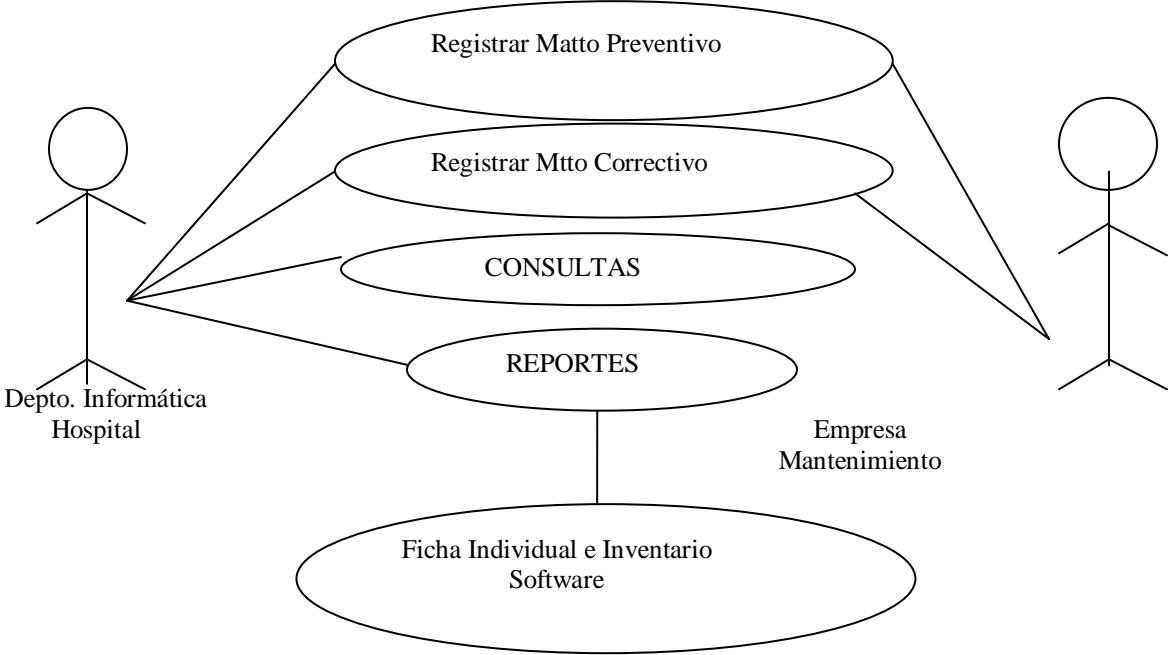


Diagrama de Gestión de Mantenimiento



2.3.3 ESPECIFICACIONES DE CASOS DE USOS

Nombre:	Realizar ingresos
Versión:	0.1
Autor:	Ernesto Velásquez, Christian Vásquez, Julio Avalos
Actores:	Distribución
Precondiciones:	El pedido realizado haya sido recibido, y revisado.
Flujo normal:	<ol style="list-style-type: none"> 1. Se establece el numero de ingreso de manera automatica y secuencial 2. Se ingresa el numero de factura 3. Se ingresa el proveedor. 4. Se ingresa fecha de compra 5. se selecciona tipo de compra 6. Se ingresan los productos comprados <ol style="list-style-type: none"> 1. Ingresar producto 2. Seleccionar si aplica IVA 3. Ingresar cantidad del producto 4. Ingresar precio unitario del producto 5. Ingresar descuento 6. Calcular subtotal (precio * cantidad)-descuento 7. ingresar otros gastos 8. Calcular total descuentos 9. Calcular total IVA 10. Calcular subtotal compra 11. Calcular total compra 12. actualizar inventario de productos
Flujo alternativo	
Post-condiciones	Se registra el ingreso y se actualiza el inventario de productos.

Nombre:	Gestión de Equipo
Versión:	0.1
Autores:	Ernesto Velásquez, Christian Vásquez, Julio Avalos
Actores:	Departamento de Informática del Hospital.
Descripción:	<p>Se establece el ingreso de un nuevo equipo informático para ser controlado desde el sistema, pudiéndolo controlar de forma generalizada y también a través de cada una de sus partes o hardware que este posea.</p> <p>Logrando verificar la asignación de dicho equipo controlando el cuidado del mismo y responsabilizando al empleado de su herramienta de trabajo, controlando también el departamento en el que esté ubicado y el cargo del empleado que lo posee.</p> <p>A la vez controlamos el estado del equipo por medio de los correspondientes mantenimientos que este requiera en el transcurso de su existencia.</p> <p>Generamos reportes y consultas que se requieran necesarias para verificar alguna modificación o simplemente presentar un informe que lo requiera la dirección del hospital o la autoridad competente en este caso Ministerio de Salud Pública ente controlador de instituciones públicas.</p>

Nombre:	Registro de Equipo
Versión:	0.1
Autores:	Ernesto Velásquez, Christian Vásquez, Julio Avalos
Actores:	Departamento de Informática del Hospital.
Precondiciones:	El equipo este registrado en el Activo fijo del Hospital.
Flujo normal:	<ol style="list-style-type: none"> 1. Se registra el equipo asignándole un código para formulación de un inventario. 2. El equipo es asignado a un empleado. 3. Selecciona el cargo que este empleado posee. 4. Selecciona el departamento al cual pertenece el empleado.
Flujo alternativo	<ul style="list-style-type: none"> • Si al equipo se le compra un hardware nuevo solo se registra el hardware dentro del equipo que ya existe. • Si el empleado cambia de cargo se puede modificar el cargo que fue asignado al principio. • Si el empleado cambia de departamento se puede modificar el departamento que fue asignado al principio. • Si el equipo es asignado a otra persona se puede modificar el nombre del empleado.
Post-condiciones	Se actualizan las modificaciones hechas al equipo.

Nombre:	Registro de Monitor
Versión:	0.1
Autores:	Ernesto Velásquez, Christian Vásquez, Julio Avalos
Actores:	Departamento de Informática del Hospital.
Precondiciones:	Que el Monitor haya sido registrado en equipo.
Flujo normal:	<ol style="list-style-type: none"> 1. Se registra monitor en el sistema con su código para la formulación de un inventario. 2. Se registra el modelo del monitor. 3. Se registra la serie del monitor. 4. Se registra el Tipo del monitor. 5. Se registra el tamaño del monitor. 6. Se selecciona el estado en el que se encuentra el monitor. 7. Se selecciona la procedencia del monitor. 8. Se registra la fecha de adquisición. 9. Se registra el precio al cual fue adquirido si es por compra. 10. Se registran las observaciones si las hay en cuanto al monitor.
Post-condiciones	Se actualizan las modificaciones hechas al monitor.

Nombre:	Registro de Ups
Versión:	0.1
Autores:	Ernesto Velásquez, Christian Vásquez, Julio Avalos
Actores:	Departamento de Informática del Hospital.
Precondiciones:	Que el Ups halla sido registrado en equipo.
Flujo normal:	<ol style="list-style-type: none"> 1. Se registra el Ups asignado a un equipo con su código para la formulación de un inventario. 2. Se registra la fecha de adquisición. 3. Se registra el precio al cual fue adquirido si es por compra. 4. Se realizan las observaciones pertinentes si las hubieran.
Post-condiciones	Se actualizan las modificaciones hechas al Ups.

Nombre:	Registro de Mouse
Versión:	0.1
Autores:	Ernesto Velásquez, Christian Vásquez, Julio Avalos
Actores:	Departamento de Informática del Hospital.
Precondiciones:	Que el Mouse haya sido registrado en equipo.
Flujo normal:	<ol style="list-style-type: none"> 1. Se registra mouse en el sistema con su código para la formulación de un inventario. 2. Se registra la marca del mouse. 3. Se registra el modelo mouse. 4. Se registra la serie del mouse. 5. Se registra el tipo de mouse. 6. Se selecciona el estado en el que se encuentra el mouse. 7. Se realizan las observaciones pertinentes si las hubieran.
Post-condiciones	Se actualizan las modificaciones hechas al mouse.

Nombre:	Registro de CPU
Versión:	0.1
Autores:	Ernesto Velásquez, Christian Vásquez, Julio Avalos
Actores:	Departamento de Informática del Hospital.
Precondiciones:	Que el CPU haya sido registrado en equipo.
Flujo normal:	<ol style="list-style-type: none"> 1. Se registra el CPU en el sistema con su código para la formulación de un inventario. 2. Se registra la marca del CPU. 3. Se registra el modelo del CPU. 4. Se registra la serie del CPU. 5. Se registra el procesador del CPU. 6. Se registra la velocidad del CPU. 7. Se registra el disco duro del CPU. 8. Se registra la memoria RAM del CPU. 9. Se detallan las observaciones del CPU si este le fuera necesario.
Post-condiciones	Se actualizan las modificaciones hechas al CPU.

Nombre:	Registro de Impresor
Versión:	0.1
Autores:	Ernesto Velásquez, Christian Vásquez, Julio Avalos
Actores:	Departamento de Informática del Hospital.
Precondiciones:	Que el Impresor haya sido registrado en equipo.
Flujo normal:	<ol style="list-style-type: none"> 1. Se registra Impresor en el sistema con su código para la formulación de un inventario. 2. Se registra el modelo del Impresor. 3. Se registra la serie del Impresor. 4. Se registra el Tipo del Impresor. 5. Se registra la velocidad del Impresor. 6. Se selecciona el estado en el que se encuentra el Impresor. 7. Se selecciona la procedencia del Impresor. 8. Se registra la fecha de adquisición. 9. Se registra el precio al cual fue adquirido si es por compra. 10. Se registran las observaciones si las hay en cuanto al Impresor.
Post-condiciones	Se actualizan las modificaciones hechas al Impresor.

Nombre:	Registro de Teclado
Versión:	0.1
Autores:	Ernesto Velásquez, Christian Vásquez, Julio Avalos
Actores:	Departamento de Informática del Hospital.
Precondiciones:	Que el teclado halla sido registrado en equipo.
Flujo normal:	<ol style="list-style-type: none"> 1. Se registra teclado en el sistema con su código para la formulación de un inventario. 2. Se registra el modelo del teclado. 3. Se registra la serie del teclado. 4. Se registra el Tipo del teclado. 5. Se registra el idioma del teclado. 6. Se selecciona el estado en el que se encuentra el teclado. 7. Se selecciona la procedencia del teclado. 8. Se registra la fecha de adquisición. 9. Se registra el precio al cual fue adquirido si es por compra. 10. Se registran las observaciones si las hay en cuanto al teclado.

Nombre:	Gestión de Empleado
Versión:	0.1
Autores:	Ernesto Velásquez, Christian Vásquez, Julio Avalos
Actores:	Departamento de Informática del Hospital.
Descripción:	Administración del registro de empleados que laboran en la institución, en donde se podrán realizar las acciones de ingresar nuevo registro de empleado, modificar sus datos, eliminar registro de empleado del sistema y listar los empleados.

Nombre:	Gestión de Cargo
Versión:	0.1
Autores:	Ernesto Velásquez, Christian Vásquez, Julio Avalos
Actores:	Departamento de Informática del Hospital.
Descripción:	Administración del registro de Cargo de los empleados que laboran en la institución, donde se podrán realizar las acciones siguientes: Agregar nuevo Cargo, modificar sus datos y eliminar un registro de cargo del sistema y listar los Cargos.

Nombre:	Gestión de Departamento
Versión:	0.1
Autores:	Ernesto Velásquez, Christian Vásquez, Julio Avalos
Actores:	Departamento de Informática del Hospital.
Descripción:	Administración de registro Departamento de la Institución, donde se podrán realizar las acciones siguientes: Agregar un nuevo Departamento, modificar sus datos y eliminar un registro de departamento del sistema y listar los departamentos.

Nombre:	Gestión de Mantenimiento
Versión:	0.1
Autores:	Ernesto Velásquez, Christian Vásquez, Julio Avalos
Actores:	Departamento de Informática del Hospital.
Descripción:	Administración de registro mantenimiento de los equipos de la Institución, donde se podrán agregar los mantenimientos realizados a un equipo en específico, el tipo de mantenimiento realizado a este, se podrá verificar el estado de sus periféricos del equipo, la empresa que lo realizo, la fecha del nuevo mantenimiento y su costo.

Nombre:	Gestión de Consultas
Versión:	0.1
Autores:	Ernesto Velásquez, Christian Vásquez, Julio Avalos
Actores:	Departamento de Informática del Hospital.
Descripción:	Esta gestión está definida con el fin de poder ver (consultar) los datos de cada equipo, y el estado también de sus periféricos o hardware de cada equipo.

Nombre:	Gestión de Reportes
Versión:	0.1
Autores:	Ernesto Velásquez, Christian Vásquez, Julio Avalos
Actores:	Departamento de Informática del Hospital.
Descripción:	Consiste en llegar a la entrega total de cada especificación registrada, tipos de mantenimiento asignaciones de equipos, a que departamento se le asigna, etc.

Nombre:	Registrar Mantenimiento Preventivo
Versión:	0.1
Autores:	Ernesto Velásquez, Christian Vásquez, Julio Avalos
Actores:	Departamento de Informática del Hospital.
Descripción:	Se Aplica este mantenimiento según un criterio de uso de cualquier equipo informático o de condiciones previstas con anticipación, establecidas de acuerdo al uso que este teniendo, o de la antigüedad de algún Equipo para su mejor funcionalidad.

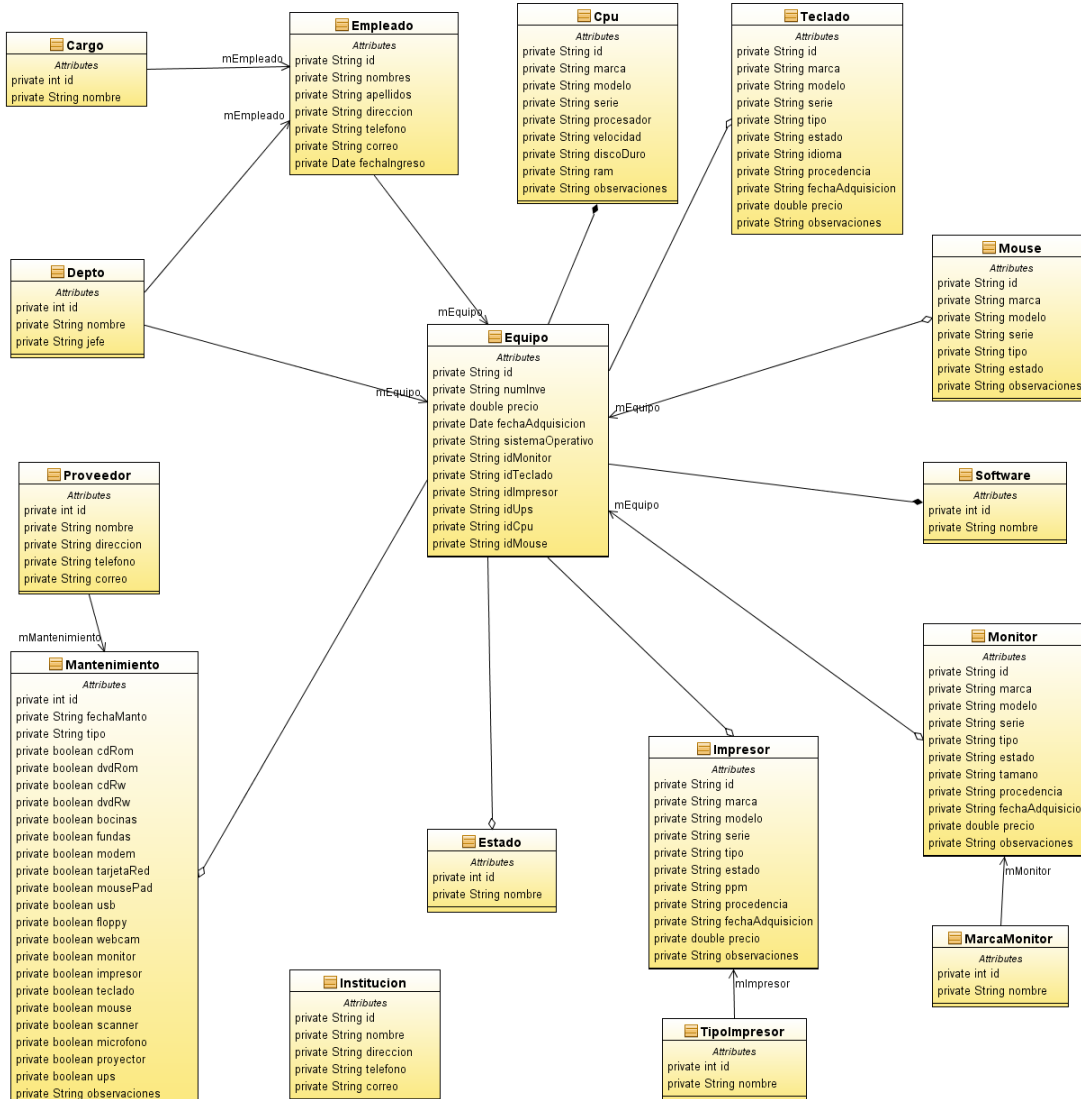
Nombre:	Registrar Mantenimiento Correctivo
Versión:	0.1
Autores:	Ernesto Velásquez, Christian Vásquez, Julio Avalos
Actores:	Departamento de Informática del Hospital.
Descripción:	Se Aplica este mantenimiento cuando un equipo pierde casi en su totalidad su funcionamiento por el fallo de alguna pieza interna o externa de la computadora podemos aplicar el mantenimiento reparando la pieza o sustituyéndola por una nueva en el equipo.

Nombre:	Consultas
Versión:	0.1
Autores:	Ernesto Velásquez, Christian Vásquez, Julio Avalos
Actores:	Departamento de Informática del Hospital.
Descripción:	Se puede tener acceso a las consultas de los 2 tipos de mantenimiento realizados a los diferentes equipos que el hospital posee.

Nombre:	Reportes
Versión:	0.1
Autores:	Ernesto Velásquez, Christian Vásquez, Julio Avalos
Actores:	Departamento de Informática del Hospital.
Descripción:	Mediante los diferentes mantenimientos efectuados podemos generar un reporte a la dirección del hospital o al Ministerio de Salud de cada equipo donde verificaríamos su condición y a la vez podríamos ver la posibilidad de sustituirlo o desfasarlo.

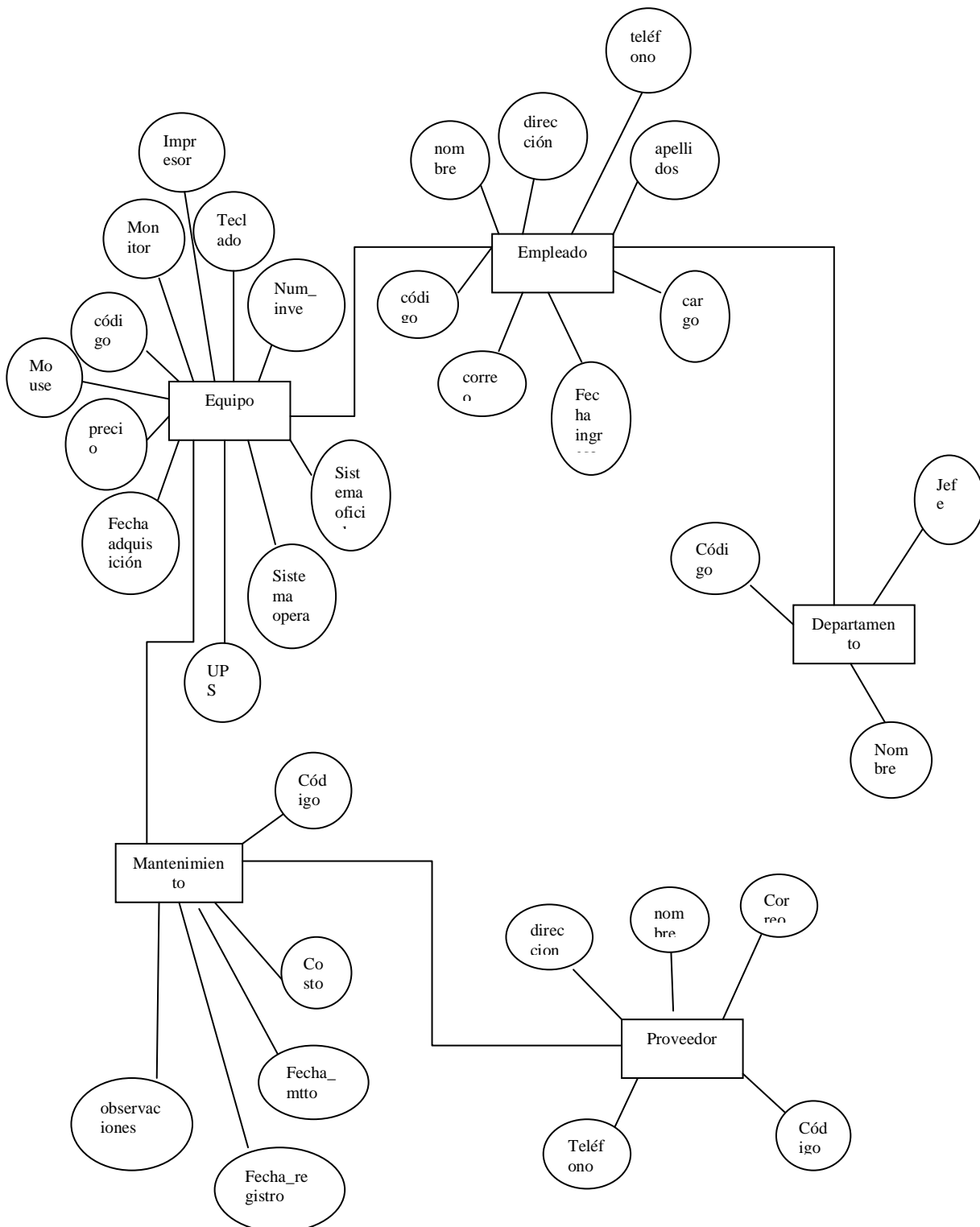
Nombre:	Ficha Individual e Inventario de Software
Versión:	0.1
Autores:	Ernesto Velásquez, Christian Vásquez, Julio Avalos
Actores:	Departamento de Informática del Hospital.
Descripción:	Esta es la expresión del sistema en la que podemos ver o imprimir un resultado total de un equipo y sus periféricos logrando así una satisfacción en el control de cada equipo inventariado.

2.3.4 DIAGRAMA DE CLASES

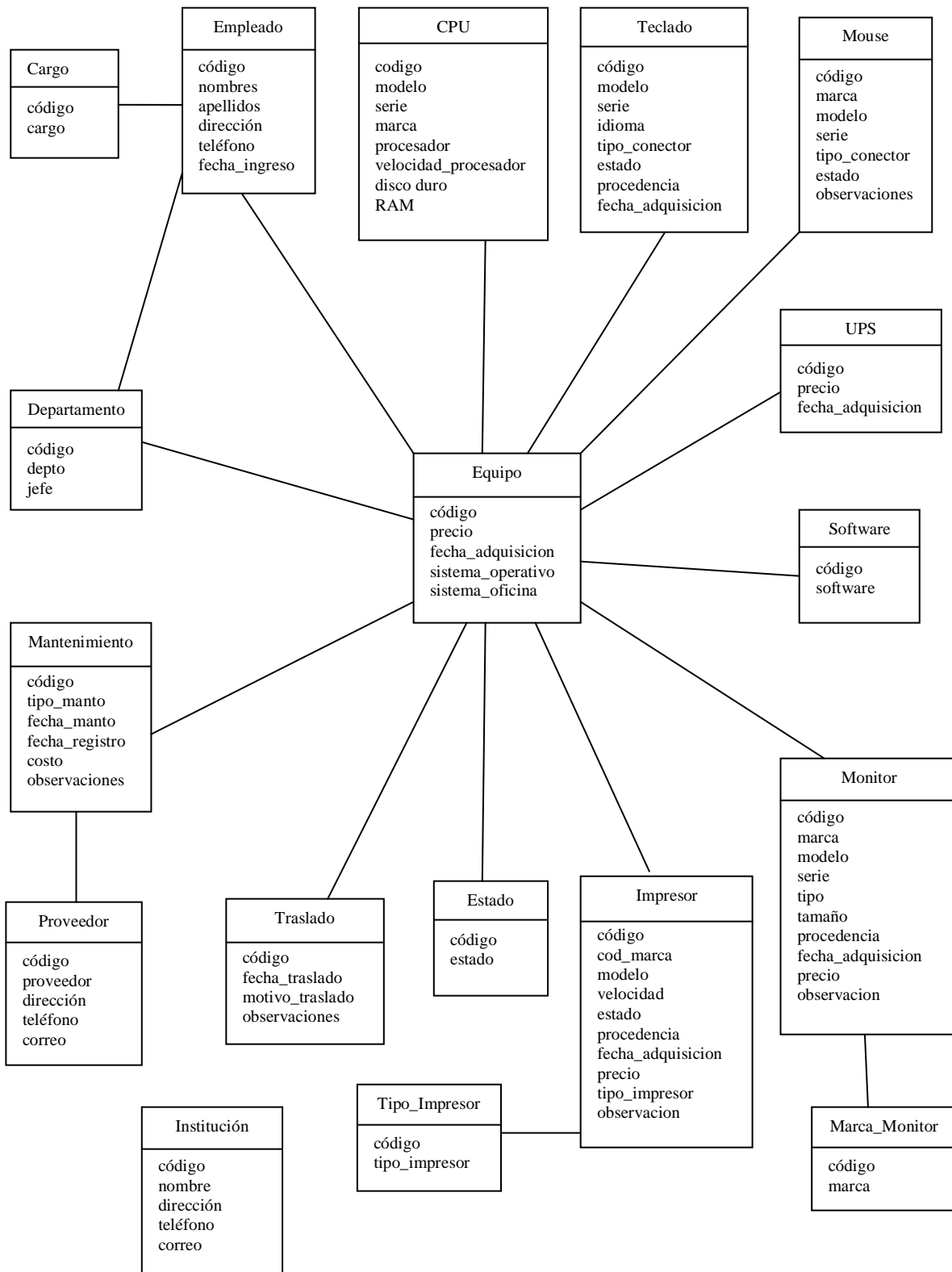


2.3.5 DISEÑO DE LA BASE DE DATOS

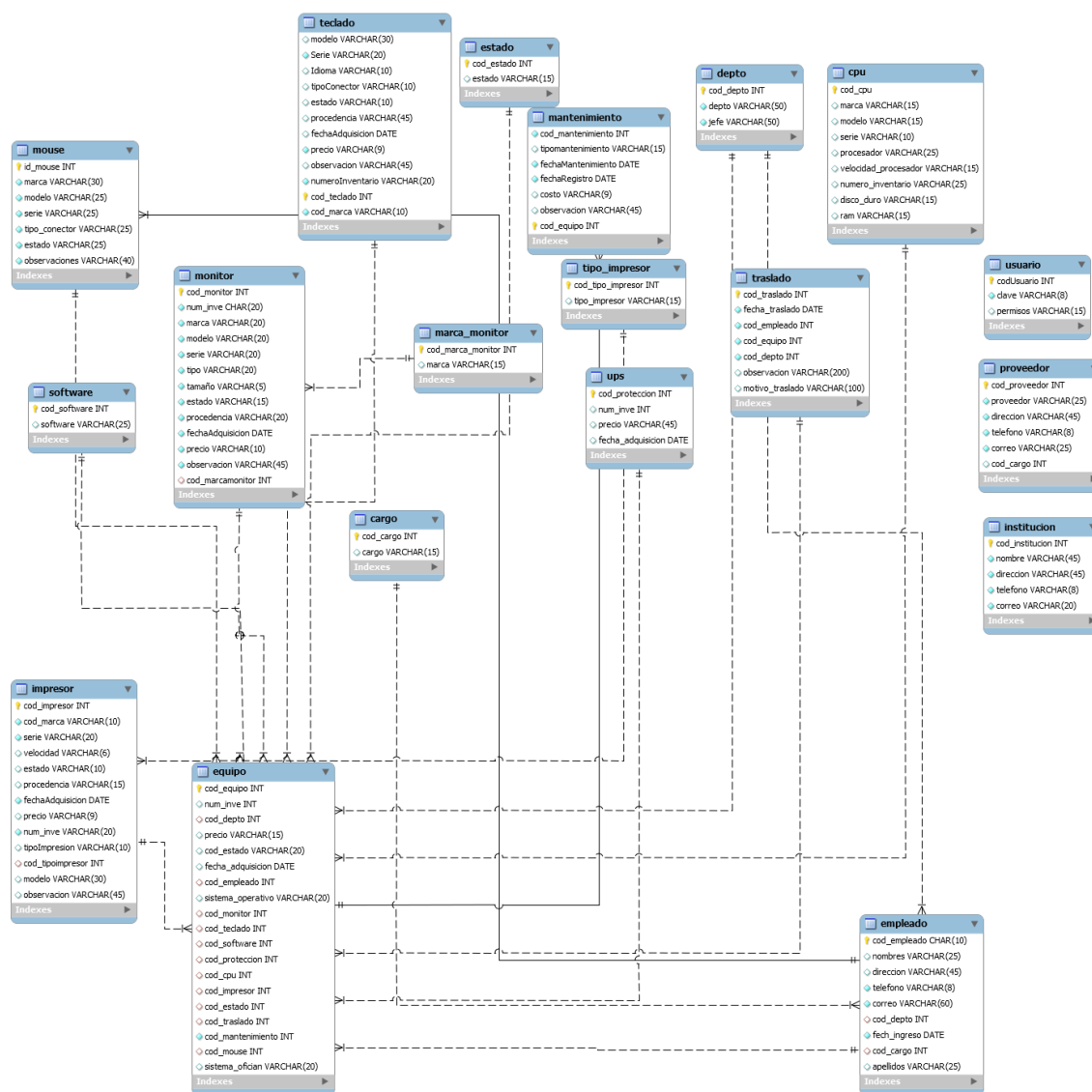
DIAGRAMA CONCEPTUAL (E-R)



DISEÑO LÓGICO

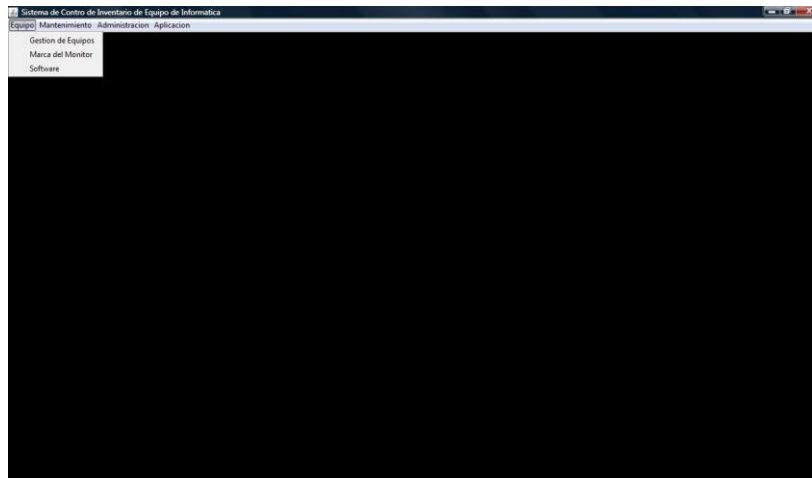


DISEÑO FÍSICO

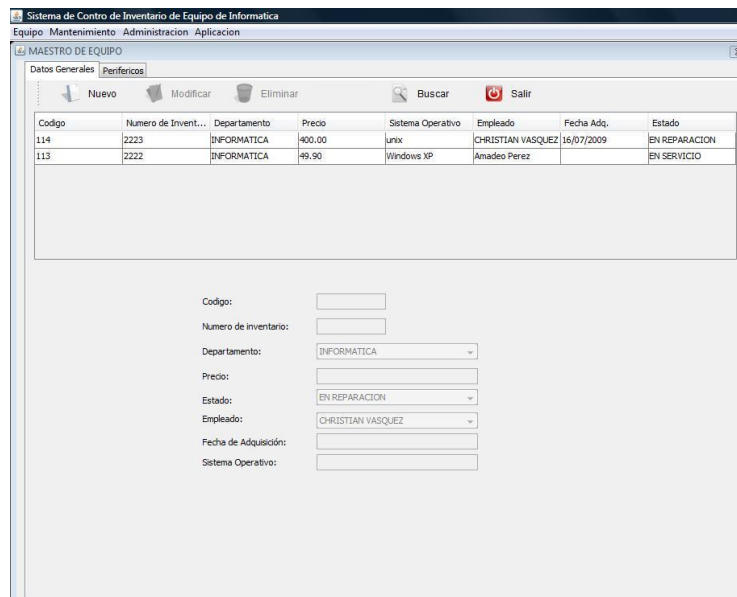


2.3.6 DISEÑO DE INTERFAZ GRAFICA DE USUARIO

Para que los usuarios puedan trabajar de una forma práctica se ha elaborado un diseño gráfico de fácil manejo, a continuación mostramos algunas de las pantallas que se encuentran en nuestro sistema.

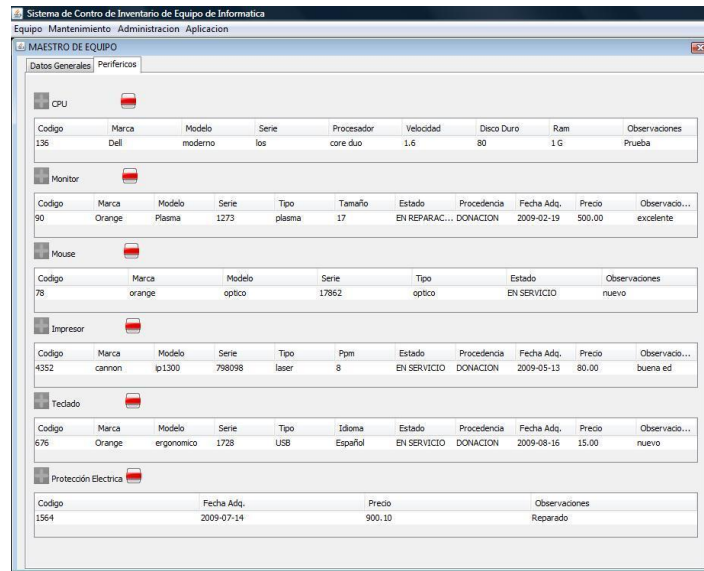


En la pantalla principal el usuario podrá seleccionar a través de un menú las diferentes pantallas para trabajar o consultar la información sobre el equipo informático de la empresa.



En la pantalla de gestión de equipo podrá ingresar, modificar, eliminar o simplemente consultar todos los equipos que posee el Hospital con las características de interés para los

responsables, al igual que los periféricos y el estado en que se encuentran cada uno de ellos.



Además el sistema cuenta con una sección para poder tener el control del mantenimiento de cada uno de los equipos y señalando además la fecha y tipo de mantenimiento dado.

The screenshot shows the 'Mantenimiento' tab with the following fields:

- Código de Equipo:
- Usuario:
- Departamento:
- Tipo:
- Memoria RAM:
- Marca:
- Disco Duro:
- Modelo:
- Procesador:
- Sistema Operativo:
- Numero de Inventario:
- Numero de Serie:

The screenshot shows the 'Mantenimiento' tab with the following sections:

- Descripcion General:**
 - Tipo de Mantenimiento:
 - Fecha Mantenimiento:
- Mantenimiento Equipo:**
 - CD-ROM
 - DVD-ROM
 - Módem
 - USB
 - CD-RW
 - DVD-RW
 - Tarjeta de Red
 - Floppy
 - Bocinas
 - Fundas
 - Mouse Pad
 - WebCam
- Mantenimiento Perifericos:**
 - Monitor
 - Impresor
 - Teclado
 - Scanner
 - Mouse
 - Micrófono
 - Proyector
 - Ups

2.4 NUEVAS FORMULACIONES TEORICAS

Después de haber realizado toda la investigación y una vez comprobada su validez como instrumento de optimización para el control y asignación del equipo de informática en los diversos departamentos de la Institución, el siguiente paso sería optimizar diferentes partes del sistema que permita mejorar sus prestaciones:

Estudio de la incorporación de la función contable de Activo Fijo

Se Observó como el mismo sistema podría también formar parte del departamento de Activo Fijo, de manera que al ir registrando, el precio de adquisición, el cambio de piezas, los años de uso y de otras características propias de la contabilidad pueda deducir el precio actual que posee cada uno de los equipos de la institución. Esto ha sido comprobado empíricamente pero sin ningún estudio riguroso del caso.

Evolución del Sistema

Además de mejorar el sistema, este puede servir como un modelo para la creación de un sistema de inventario propio para los otros departamentos de la institución, con el fin de optimizar las prestaciones a los usuarios del Hospital y tener un mejor control de los bienes que posee este. También puede ser adaptado a la realidad y necesidad de otras empresas que desean tener un mejor control de sus bienes, recomendamos hacer un estudio sobre ello.

2.5 BIBLIOGRAFIA

- Laudon, Jane y Kenneth (2008). Sistemas de información gerencial- Administración de la empresa digital (10ª. Edición). Madrid, España: Pearson Educación- Prentice Hall.
- Senn, James A. (1992). Análisis y Diseño de Sistemas de Información. (2ª. Edición). México: Editorial McGrawHill.
- Kurose J. F.; Ross K. W.(2003). Redes de Computadores. Un enfoque descendente basado en Internet (2ª. Edición), Madrid, España: Addison-Wesley
- Jacobson, Ivar; Booch, Grady; Rumbaugh, James (2000). El Proceso Unificado de Desarrollo de Software (1ª. Edición). Madrid, España: Pearson Addison-Wesley.

Medio Digitales:

(Consultados en el año 2009)

- <http://www.uml.org/>
- <http://www.sun.com/>
- http://www-01.ibm.com/software/rational/?pgel=ibmhzn&cm_re=masthead-_-products-_-sw-rational

2.6 ANEXOS

(ANEXO A)



Universidad Pedagógica de El Salvador
Facultad de Ciencias Económicas
Carrera: Técnico en Sistemas de Computación

ENTREVISTA

Nombre: _____ **Cargo:** _____

Departamento en el que trabaja: _____ **Fecha:** _____

1-¿Cuenta el hospital con sistema computarizado que lleve el control detallado de los equipos?

2-¿Cómo se lleva a cabo el control de los equipos?

3-¿Quiénes se encargan de recolectar la información del estado de los equipos?

4-¿Qué tipo de documentación se ocupa para la recolección de información?

5-¿A qué agentes o instituciones se tiene que dar constancia de esta información?

6-¿A quién se le es asignado el equipo?

7-¿Qué requerimientos solicitaría al tener un sistema computarizado?

8-¿Cuáles serían los entes que manejarían tal sistema?

(ANEXO B)

Consultores de Equipos y Programas

FORMULARIO DE SERVICIO

Usuario:		Tipo de Servicio:	
Área de Trabajo:			

Tipo:		Memoria RAM:	
Marca:		Disco Duro:	
Modelo:		Procesador:	
# de Serie:		Sistema Operativo:	
# de Inventario:		Software de Oficina:	Licencia

CD-ROM <input type="checkbox"/>	DVD-ROM <input type="checkbox"/>	Módem <input type="checkbox"/>	USB <input type="checkbox"/>
CD-RW <input type="checkbox"/>	DVD-RW <input type="checkbox"/>	Tarjeta de Red <input type="checkbox"/>	FLOPPY <input type="checkbox"/>
Bocinas <input type="checkbox"/>	Fundas <input type="checkbox"/>	Mouse Pad <input type="checkbox"/>	WebCam <input type="checkbox"/>

IMPRESOR			UPS		
Tipo:	Matricial <input type="checkbox"/>	Inyección <input type="checkbox"/>	Laser <input type="checkbox"/>	Tipo:	
Marca:				Marca:	
Modelo:				Modelo:	
# Serie:				# Serie:	
# Inventario:				# Inventario:	

SCANNER		OTROS	
Tipo:		Tipo:	
Marca:		Marca:	
Modelo:		Modelo:	
# Serie:		# Serie:	
# Inventario:		# Inventario:	

Observaciones: _____

Fecha de Servicio: Día ____ Mes ____ Año ____

Técnico Responsable	
Nombre:	_____
Firma:	_____

Requió Conforme	
Nombre:	_____
Firma:	_____

Cto. Urbano José Simeón Cañas
D. P. ... # 10, Matanzas

EXHA INDIVIDUAL DE EQUIPO E INVENTARIO DE SOFTWARE

Datos Generales de Identificación

NIVEL T/O: HOSPITAL NAC DR LUIS DE LA ROSA
 DIRECCION: EVASQUEZ
 DEPENDENCIA: UNIDAD FINANCIERA INSTITUCIONAL
 AREA ACTIVO FIJO: RESPONSABLE DEL EQUIPO: JOSE FERNANDO RECHOS

DATOS DE LA RED

NOBRE DEL PC	EN RED	GRUPO DE TRABAJO O DOMINIO	DIRECCION IP	CON ACCESO A INTERNET	Proveedor internet
	SI <input checked="" type="checkbox"/> No <input type="checkbox"/>	MSPAS.008 SV	192.168.20.38	SI <input type="checkbox"/> No <input checked="" type="checkbox"/>	TELECOM

Datos del equipo

NO INV ACTIVO FIJO	Marca	Modelo	Serie	Tipo de Procesador	Velocidad Procesador	Capacidad Disco Duro	Capacidad de Memoria RAM	Estado	Procedencia del equipo (F.F.)	Tipo Mnt.	Observación
(256-153-009-01-00001)	HP COMPAQ	DC-60000	MR-150					BUENO	GOES	PREVENTIVO Y CORRECTIVO	

TECLADO

NO INV ACTIVO FIJO	MARCA	MODELO	SERIE	IDIOMA	TIPO DE CONECTOR	ESTADO	Procedencia del equipo (F.F.)	OBSERVACION

MONITOR

NO INV ACTIVO FIJO	MARCA	MODELO	SERIE	TIPO DE MONITOR	TAMAÑO DE MONITOR	ESTADO	Procedencia del equipo (F.F.)	OBSERVACION

IMPRESOR

NO INV ACTIVO FIJO	MARCA	MODELO	SERIE	TIPO DE IMPRESOR	VELOCIDAD DEL IMPRESOR	B/N, O COLOR	ESTADO	Procedencia del equipo (F.F.)	Tpo Mnt.	OBSERVACION
0256-153-056-01-00021	CANON	FX1M1 #1900	PC V V08730			GRIS	BUENO	GOES	PREVENTIVO CORRECT	

(ANEXO C)

ANEXO D

PRESUPUESTO

GASTOS DE TRASPORTE		\$ 40.00
Gasolina	\$ 40.00	
GASTOS DE ALIMENTACION (4 viajes)		\$ 60.00
3 desayunos por viaje	\$ 6.00	
3 Almuerzos por viaje	\$ 9.00	
GASTOS DE PAPELERIA		\$ 28.15
Fotocopias	\$ 1.15	
Impresiones	\$ 24.00	
Anillados	\$ 3.00	
TIEMPO INVERTIDO EN 6 MESES (3 programadores)		\$ 2,700.00
Pago al mes por cada uno	\$ 150.00	
GASTOS DE EQUIPO UTILIZADO		\$ 1,074.00
1 Computadora Intel Core 2 Duo T5500 1.66 GHz , Memoria 2048 MB, DDR2 PC5300 667MHz, Disco Duro 160 GB - 5400 rpm, 160GB 5400rpm Seagate ST9160821AS	\$ 1,050.00	
3 Memorias USB	\$ 24.00	
GASTOS DE SOFTWARE		\$ -
Java	\$ -	
NetBeen	\$ -	
MySQL	\$ -	
TOTAL DE GASTOS INVERTIDOS		\$ 3,902.15