



**Benemérita
Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación**

Tesis Profesional:

**Sistema de BD para automatizar la gestión de Inventario
y reportes de fallas de equipos de cómputo**

**Que para obtener el grado de
Licenciada en Ciencias de la Computación**

Presenta

Victoria Portillo Martínez

Director de Tesis

MC. Yolanda Moyao Martínez

Puebla, Pue. Noviembre 2012

Capítulo I: Marco Teórico.....	8
1.1 INGENIERIA DE SOFTWARE.....	9
1.2.1 Proceso Unificado de Rational	9
1.2.2 Características Esenciales.	10
1.2.3 Ciclo de Vida	11
1.2.4 Que es Uml.....	12
1.3 MODELOS DE PROCESO DE SOFTWARE.....	14
1.3.1 Proceso Unificado de Desarrollo del Software.....	14
1.3.2 Modelo Incremental	16
1.4 BASES DE DATOS.....	17
1.4.1 Introducción y definición de Bases de Datos	17
1.4.2 DBMS	18
1.5 MODELO DE BASES DE DATOS	18
1.5.1 Modelo Entidad- Relación (E-R).....	19
1.5.2 Modelo Relacional	20
1.5.2.1 Llaves Primarias y Foráneas	20
1.6 NORMALIZACIÓN.....	20
1.6.1 Definición de Normalización	21
1.6.2 Formas Normales	21
1.6.3 Primera Forma Normal	22
1.6.4 Segunda Forma Normal	22
1.6.5 Tercera Forma Normal	22
CAPÍTULO 2: ANÁLISIS Y DISEÑO (ORIENTADO A OBJETOS).....	23
2.1 PLANTEAMIENTO DEL PROBLEMA	24
2.2 OBJETIVO GENERAL.....	24
2.3 REQUERIMIENTOS DEL PROBLEMA	25
2.4 ALCANCES	26
2.5 METODOLOGÍA	29
2.6 ANÁLISIS DEL SISTEMA	30
2.6.1 Diccionario de datos del Modelo	31
2.6.2 Diagrama de Casos de Uso general del Sistema	32
2.6.3 Especificación de Casos de Uso	34
2.6.4 Escenarios Caso de uso Inventario	39
2.6.5 Diagrama de Clases	41
2.7 DISEÑO DEL SISTEMA.....	43

2.7.1 Diagrama de Clases de Diseño	44
2.7.2 Diagrama de Secuencia	45
2.7.3 Diagramas de Colaboración.....	47
Capítulo 3: Diseño de la Base de Datos, implementación y pruebas del sistema	49
3.1. DISEÑO CONCEPTUAL DE LA BASE DE DATOS	50
3.1.1 Modelo Entidad – Relación	51
3.1.2 Descripción de Entidades y Relaciones	52
3.1.3 Diseño Lógico	53
3.1.3.1 Normalización.....	54
3.1.4 IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA	58
3.1.4.1 Modelo Lógico	59
3.1.5 IMPLEMENTACIÓN DE LA INTERFAZ.....	62
3.1.5.1 Interfaz de Inicio de Sesión	63
3.2 INTERFAZ DEL ADMINISTRADOR	66
3.3 INTERFAZ DE USUARIO.....	66
3.4 PRUEBAS DEL SISTEMA	67
3.5 ÁREAS	67
3.6 ALTA DE ARTÍCULOS.....	68
3.7 ARTÍCULOS REGISTRADOS.....	68
3.8 CONSULTA ARTÍCULOS	69
3.9 DATOS CONSULTADOS.....	69
3.10 IMPRIMIR BITÁCORA	70
3.11 BACKUP DE ARTÍCULOS	70
3.12 ÓRDENES DE MANTENIMIENTO	71
3.13 ABRIR REPORTE DE FALLA	71
3.14 ÓRDENES REGISTRADAS	72
3.15 CONSULTAR ÓRDENES POR ATENDER	72
3.16 BITÁCORA ORDEN.....	73
CONCLUSIONES	74
BIBLIOGRAFÍA	75

INTRODUCCIÓN

La falta de un inventario que permita mantener la información concentrada y actualizada para una mejor administración de los activos de una organización educativa puede resultar una tarea compleja, como un acercamiento a la solución de esta necesidad, se plantea el desarrollo de un sistema que tenga como objetivo mantener la información actualizada de la configuración de hardware y software de los recursos de cada sala de cómputo. La organización requiere de un sistema de inventario que sea capaz de realizar por separado las tareas de control de inventario y reportes de fallas tomando en consideración la relación que existe entre ellas.

La automatización de la gestión de inventarios presenta diversos beneficios, como ahorrar tiempo y esfuerzo para llevar a cabo los inventarios físicos, reducir la interrupción en el trabajo de los usuarios, tener el control de cambios en piezas de hardware, optimizar el uso de los recursos de hardware y software en la organización adecuándolos a las necesidades reales y asegurar que los usuarios de la organización vean reportes periódicos que estén enterados de los cambios, en el caso de los procesos de compras conozcan de primera mano información que les ayude en las actividades de compra de activos y toma de decisiones.

El resultado de la gestión de inventarios también contribuye a respaldar las políticas de seguridad en una organización que tiene como fin evitar la instalación de software no licenciado en los laboratorios de cómputo reduciendo el riesgo de incurrir a multas por el uso de software ilegal.

El desarrollo de este trabajo queda documentado de la siguiente forma:

El capítulo uno se centra en el Marco Teórico en el cuál se da una introducción sobre los conceptos fundamentales utilizados durante el desarrollo de este trabajo de tesis que está relacionado con la ingeniería de software, como una metodología de trabajo para desarrollar software con una base técnica apropiada.

El capítulo dos se refiere al Planteamiento del Problema abarcando las características del análisis y diseño orientado a objetos donde se describe el comportamiento de la arquitectura del sistema creando los elementos básicos del (DOO) Diseño Orientado a Objetos. Diagramas de casos de uso, diagramas de estados y los diagramas de clases en UML.

En el capítulo tres se detalla el Diseño de la Base de Datos donde se aplican diversas fases del diseño que sirve como soporte de la base de datos mediante la utilización de procedimientos, técnicas, herramientas como el diseño conceptual, diseño lógico y el proceso de normalización de la base de datos.

Se incluye la Implementación y Pruebas, se visualiza el modelo lógico de la base de datos, la cual será implementado en el lenguaje de programación Php y utilizando el manejador de bases de datos MySQL.

Se muestra las interfaces de usuario de los módulos implementados y se detallan el funcionamiento de cada una de ellos, finalmente tenemos las conclusiones obtenidas con el desarrollo de este trabajo de tesis.

TABLA DE FIGURAS

FIGURA	PÁGINA
CAPÍTULO I	
Figura 1.1 Fases del Ciclo de Vida	12
Figura 1.2 Desarrollo Iterativo Incremental	15
Figura 1.3 Fases del Modelo Incremental	15
Figura 1.4 Elementos gráficos que representan un modelo E-R	19
CAPÍTULO II	
Figura 2.1 Casos de uso para la especificación de requisitos del sistema	32
Figura 2.2 Caso de Uso: Gestión Inventario	33
Figura 2.3 Caso de Uso: Reportes de fallas de Mantenimiento	34
Figura 2.4 Diagrama de Clases, modelo conceptual	42
Figura 2.5 Diagrama de Clases, VOPC	43
Figura 2.6 Clases y Relaciones	44
Figura 2.7 Diagrama de Gestión de Inventario	45
Figura 2.8 Diagrama de Reporte de fallas	46
Figura 2.9 Diagrama Abrir, Atender Reporte	46
Figura 2.10 Diagrama Gestión de Inventario	47
Figura 2.11 Diagrama Asigna Reporte	48
Figura 2.12 Diagrama de Abrir, atender Reporte	48
CAPÍTULO III	
Figura 3.13 Proceso de Diseño de una Base de Datos	50
Figura 3.14 Modelo Entidad relación de clases	51
Figura 3.15 Interfaz de PhpMyAdmin	59

Figura 3.16 Creación de una tabla en PhpMyAdmin	59
Figura 3.17 Conexión de una Base de Datos en Dreamweaver	63
Figura 3.18 Interfaz de inicio de sesión	63
Figura 3.19 Permisos administrativos que permite PhpMyAdmin	64
Figura 3.20 Función MD5 aplicada a la Base de Datos Inventario	65
Figura 3.21 Interfaz Inventario.- Administrador	68
Figura 3.22 Interfaz Inventario.-Usuario	69
Figura 3.23 Interfaz Inventario.- Áreas	69
Figura 3.24 Interfaz inventario.-Registro de Artículos	70
Figura 3.25 Interfaz Inventario.- Artículos Registrados	70
Figura 3.26 Interfaz Inventario.- Consultar Artículos	71
Figura 3.27 Interfaz Inventario.- Resultado Consulta	71
Figura 3.28 Interfaz Inventario.- Vista Preliminar Bitácora	72
Figura 3.29 Interfaz Inventario.- Artículos dados de baja	72
Figura 3.30 Interfaz Orden de Mantenimiento	73
Figura 3.31 Interfaz Orden de Mantenimiento.-Registrar Orden	73
Figura 3.32 Interfaz Orden de Mantenimiento.-Órdenes Registradas	74
Figura 3.33 Interfaz Orden de Mantenimiento.-Consultar Órdenes	74
Figura 3.34 Interfaz Orden de Mantenimiento.- Bitácora	75

TABLAS

1.1 Tabla Formas Normales	21
2.2 Tabla Diccionario de datos del modelo	30
3.3 Tabla Marca	52
3.4 Tabla Modelo	52
3.5 Tabla Área	52

3.6	Tabla Responsable	52
3.7	Tabla Técnico	52
3.8	Tabla Artículos	52
3.9	Tabla Orden	53
3.10	Tabla Marca Normalizada	55
3.11	Tabla Modelo Normalizada	55
3.12	Tabla Área Normalizada	55
3.13	Tabla Responsable Normalizada	56
3.14	Tabla Técnico Normalizada	56
3.15	Tabla Artículos Normalizada	57
3.16	Tabla Orden Normalizada	57
3.17	Tabla Registro de Artículos	60
3.18	Tabla Registro Marcas de equipos de cómputo	60
3.19	Tabla Registro Modelos de equipos de cómputo	60
3.20	Tabla Registro Áreas	61
3.21	Tabla Registro Ordenes de Mantenimiento	61
3.22	Tabla Registro Áreas	61
3.23	Tabla Registro de Técnicos	62
3.24	Tabla Backup de Artículos	62

Capítulo I: Marco Teórico

1.1 INGENIERIA DE SOFTWARE

Según la definición del IEEE, citada por [Lewis 1994] "software es la suma total de los programas de computadora, procedimientos, reglas, la documentación asociada y los datos que pertenecen a un sistema de cómputo". Según el mismo autor, "un producto de software es un producto diseñado para un usuario". En este contexto, la Ingeniería de Software (SE del inglés Software Engineering) es un enfoque sistemático del desarrollo, operación, mantenimiento y retiro del software", que en palabras más llanas, se considera que "la Ingeniería de Software es la rama de la ingeniería que aplica los principios de la ciencia de la computación y las matemáticas para lograr soluciones costo-efectivas (eficaces en costo o económicas) a los problemas de desarrollo de software", es decir, "permite elaborar consistentemente productos correctos, utilizables y costo-efectivos" [1].

El proceso de ingeniería de software se define como "un conjunto de etapas parcialmente ordenadas con la intención de logra un objetivo, en este caso, la obtención de un producto de software de calidad" [Jacobson 1998]. El proceso de desarrollo de software "es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo". Concretamente "define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo" [Jacobson 1998].

El proceso de desarrollo de software requiere por un lado un conjunto de conceptos, una metodología y un lenguaje propio. A este proceso también se le llama el ciclo de vida del software que comprende cuatro grandes fases: concepción, elaboración, construcción y transición. La concepción define el alcance del proyecto y desarrolla un caso de negocio. La elaboración define un plan del proyecto, especifica las características y fundamenta la arquitectura. La construcción crea el producto y la transición transfiere el producto a los usuarios.

1.2 RUP

1. Es una filosofía y prácticas para desarrollos exitosos de software
2. Es un modelo de Proceso y biblioteca de contenidos
3. Es la base de un lenguaje de definición de procesos

1.2.1 Proceso Unificado de Rational

Durante varios años se ha utilizado el modelo tradicional en cascada, demostrando en la práctica que no refleja en la realidad la complejidad inherente al proceso de desarrollo de software. Este problema es derivado de la naturaleza implícita de la estructura de este modelo, definido por una secuencia de grandes etapas que requieren alcanzar hitos que deben ser concluidos antes de continuar con la siguiente fase.

Como una alternativa de solución a este problema, se definieron posteriormente los modelos iterativos e incrementales que trabajan adecuadamente con niveles altos de riesgo, y permiten entregar liberaciones de software en etapas tempranas; tal es el caso del Proceso Unificado propuesto por IBM, que incluye prácticas claves y aspectos relacionados a la planeación estratégica y administración de riesgos;

y actualmente guían de forma natural el proceso de desarrollo de software complejo por lo que ha sido considerado como un estándar el desarrollo de software en las empresas.

El proceso unificado conocido como RUP, es un modelo de software que permite el desarrollo de software a gran escala, mediante un proceso continuo de pruebas y retroalimentación, garantizando el cumplimiento de ciertos estándares de calidad. Aunque con el inconveniente de generar mayor complejidad en los controles de administración del mismo. Sin embargo, los beneficios obtenidos recompensan el esfuerzo invertido en este aspecto.

El proceso de desarrollo constituye un marco metodológico que define en términos de metas estratégicas, objetivos, actividades y artefactos (documentación) requerido en cada fase de desarrollo. Esto permite enfocar esfuerzo de los recursos humanos en términos de habilidades, competencias y capacidades a asumir roles específicos con responsabilidades bien definidas.

1.2.2 Características Esenciales.

Desarrollo iterativo del software:

- Permite comprender los requerimientos que hacen crecer el sistema
- Sigue un modelo que busca las tareas más riesgosas, reduciendo así los riesgos del proyecto.

Administración de requerimientos:

- Describe como se obtienen, organizan, documentan los requerimientos
- Captar y comunicar los requerimientos de la organización
- Documentar las decisiones
- Uso de arquitecturas basadas en componentes:
- Se basa en diseñar una arquitectura que sea flexible, fácil de modificar, comprensible y que se fundamenta en la reutilización de sus componentes.

Modelado visual del software:

- Modela visualmente la organización
- Permite analizar la consistencia entre los componentes, el diseño y su implementación
- Verificar calidad del software

Principios de Desarrollo

El Rup está basado en 5 principios:

Adaptar el proceso: El proceso deberá adaptarse a las características propias del proyecto u organización. El tamaño del mismo, así como su tipo o las regulaciones que lo condicionen, influirán en su diseño específico, aunque se debe tener en cuenta el alcance del proyecto.

Balancear prioridades: Debe encontrarse un balance que satisfaga los deseos de todos.

Demostrar valor iterativamente: Los proyectos se entregan en etapas iteradas. En cada iteración se analiza la opinión, la estabilidad y calidad del producto, y se refina la dirección del proyecto así como también los riesgos involucrados.

Elevar el nivel de abstracción: Este principio dominante motiva el uso de conceptos reutilizables tales como patrón del software, lenguajes de cuarta generación (SQL, lenguajes de consulta), o esquemas (frameworks).

Esto previene a los ingenieros de software ir directamente de los requisitos a la codificación de software a la medida del cliente. Un nivel alto de abstracción también permite discusiones sobre diversos niveles arquitectónicos. Éstos se pueden acompañar por las representaciones visuales de la arquitectura, por ejemplo con UML.

Enfocarse en la calidad: El control de calidad debe en todos los aspectos de la producción. El aseguramiento de la calidad forma parte del proceso de desarrollo y no de un grupo independiente. [15]

1.2.3 Ciclo de Vida

En lo que se refiere al ciclo de vida del RUP, es una implementación del desarrollo en espiral, este divide el proceso de desarrollo del software en ciclos, estos ciclos son la clave que tiene el modelo para crear un proyecto de buena calidad. Fue creado ensamblando los elementos en secuencias semi ordenadas. El ciclo de vida organiza las tareas en fases e iteraciones.

- El ciclo de vida del software se divide en secuencias de ciclos de desarrollo, donde el producto de un ciclo de desarrollo es la generación de un producto de software
- Cada ciclo es una sucesión de fases.

El RUP divide el proceso de desarrollo en ciclos, teniendo un producto final al culminar cada una de ellos, estos a la vez se dividen en fases y donde se debe tomar una decisión importante:

Incepción: se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos.

Elaboración: se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos

Construcción: se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario

Transición: se instala el producto en el cliente y se entrena a los usuarios. Surgen nuevos requisitos a ser analizados.

Mantenimiento: Una vez instalado el producto, el usuario realiza requerimientos de ajuste, esto se hace de acuerdo a solicitudes generadas como consecuencia del interactuar con el producto.

Características: Forma disciplinada de asignar tareas y responsabilidades (quién hace qué cuándo y cómo)

- Pretende implementar las mejores prácticas en Ingeniería de Software
- Desarrollo iterativo
- Administración de requisitos uso de arquitectura basada en componentes

- Modelado visual del software
- Verificación de la calidad del software

Fases: Cada fase concluye con un hito bien definido donde deben tomarse ciertas decisiones.

- Se establece la oportunidad y el alcance del proyecto.
- Se identifican todas las entidades externas con las que trata (actores) y se define la interacción a un alto nivel de abstracción: (identificar todos los casos de uso y describir algunos en detalle).[2]

En la **Figura 1.1** se muestran las fases del ciclo de vida Rup.

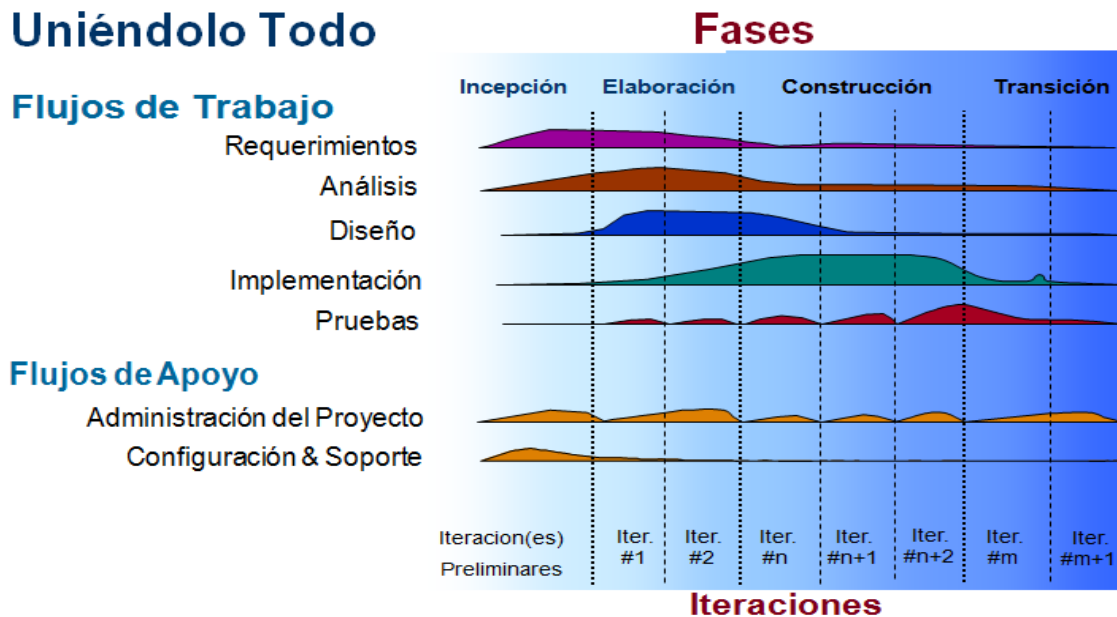


Figura 1.1 Fases del ciclo de vida Rup

1.2.4 Que es Uml

UML fue creado a principios de 1990, por tres líderes del mundo del modelado de objetos: Grady Booch, James Rumbaugh e Ivar Jacobson. Su meta fue unificar los tres mejores lenguajes para modelado del momento: el método Booch, Object Modeling Technique (OMT), y Object Oriented Software Engineering (OOSE, mejor conocida por la introducción del análisis de casos de uso). UML es ahora un Estándar del Object Management Group (OMG). El UML (Unified Modeling Language – Lenguaje Unificado de Modelación) es el lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. [14]

El UML se debe utilizar para modelar visualmente:

- La interacción del sistema con el mundo externo
- El comportamiento del sistema
- La estructura del sistema
- La arquitectura del sistema
- Los componentes del sistema

Ofrece una forma estándar de modelar sistemas software, pudiendo utilizarse:

- Con cualquier proceso de desarrollo
- A lo largo de todo el ciclo de vida
- Con distintas tecnologías de implementación

Para un óptimo aprovechamiento, debe ser usado en un proceso:

- Guiado por casos de uso
- Centrado en la arquitectura
- Iterativo e incremental

UML es una familia de notaciones, útiles para describir distintos aspectos de un sistema:

- Estático. Describe los elementos del sistema y sus relaciones.
- Dinámico. Describe el comportamiento del sistema a lo largo del tiempo.
- Casos de Uso. Desde el punto de vista del usuario.

La Modelación Visual (MV)

Se puede usar para:

- Administrar la complejidad
- Capturar / representar los procesos del negocio
- Comunicar los resultados de los procesos de análisis y diseño
- Definir la arquitectura de software
- Promover la reutilización

UML emplea dos modelos que son: El Modelo Estático y el Modelo Dinámico.

El Modelo Estático.-Los diagramas del modelo estático construyen y documentan aspectos estáticos de un sistema, reflejan lo básico, establecen el esqueleto de un sistema de software, crean una representación de los principales elementos del dominio del problema y comprenden los siguientes diagramas:

Diagrama de Casos de Uso. El Diagrama de casos de uso, muestra un conjunto de casos de uso y actores, así como sus relaciones. Son la vista de casos de uso estática de un sistema.

Diagrama de Clases. El Diagrama de clases, muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Es la vista de diseño estática de un sistema.

Diagrama de Objetos. El Diagrama de objetos, muestra un conjunto de objetos y sus relaciones, es decir. Las instancias de los elementos de un diagrama de clases.

Diagrama de Componentes. Diagrama de componentes, muestra la organización y dependencia entre un conjunto de componentes. Cubren la vista de implementación estática de un sistema. [15]

Diagrama de Secuencia. Diagrama de secuencia, muestra la interacción entre los objetos, incluyendo los mensajes que pueden ser enviados entre ellos. Son la vista dinámica de un sistema.

Diagrama de Colaboración. Diagrama de colaboración, funciona de manera semejante al de secuencia, solo que el de secuencia se preocupa por la ordenación de los mensajes, mientras este resalta la organización estructural de los objetos.

Diagrama de Estados. Diagrama de estados, muestra una máquina de estados que representa una serie de transiciones, eventos y actividades.

Diagrama de Actividad. Diagrama de actividades, un tipo especial de diagrama de estados que muestra el flujo de actividades dentro de un sistema.

Otros elementos importantes en UML son: la notación de paquetes, son mecanismos de extensión de UML que provee de una forma estándar de modelar conceptos e ideas no anticipadas en el estándar original tales como comentario en UML, estereotipos, valores etiquetados y condiciones.

1.3 MODELOS DE PROCESO DE SOFTWARE

Podemos definir un modelo de proceso de software como una representación abstracta de alto nivel de un proceso de software. Cada modelo es una descripción del modelo de software que se presenta desde una perspectiva particular. Alternativamente, a veces se usan los términos ciclo de vida y modelo del ciclo de vida. Cada modelo describe una sucesión de fases y un encadenamiento entre ellas. Según las fases y el modo en que se produzca es como se obtienen diferentes modelos de proceso. Un modelo es más adecuado que otro para desarrollar un proyecto dependiendo de un conjunto de características de éste. [6]

1.3.1 Proceso Unificado de Desarrollo del Software

Es el Proceso de desarrollo basado en el Lenguaje Unificado de Modelado y que es iterativo, centrado en la arquitectura y dirigido por los casos de uso y los riesgos, con un ciclo de vida iterativo e incremental.

Dirigido por Casos de Uso

Un caso de uso es un fragmento de funcionalidad del sistema que proporciona un resultado de valor a un usuario. Los casos de uso modelan los requerimientos funcionales del sistema.

Todos los casos de uso juntos constituyen el modelo de casos de uso.

Los casos de uso también guían el proceso de desarrollo (diseño, implementación, y prueba). Basándose en los casos de uso los desarrolladores crean una serie de modelos de diseño e implementación que llevan a cabo los casos de uso. De este modo los casos de uso no solo inician el proceso de desarrollo sino que le proporcionan un hilo conductor, avanza a través de una serie de flujos de trabajo que parten de los casos de uso.

Centrado en la Arquitectura

La arquitectura de un sistema software se describe mediante diferentes vistas del sistema en construcción.

El concepto de arquitectura software incluye los aspectos estáticos y dinámicos más significativos del sistema.

La arquitectura es una vista del diseño completo con las características más importantes resaltadas, dejando los detalles de lado.

Arquitectura: Conjunto de decisiones significativas acerca de la organización de un sistema software, la selección de los elementos estructurales a partir de los cuales se compone el sistema, las interfaces entre ellos, su comportamiento, sus colaboraciones, y su composición.

Iterativo e Incremental

Es práctico dividir el esfuerzo de desarrollo de un proyecto de software en partes más pequeñas o mini proyectos. Cada mini proyecto es una iteración que resulta en un incremento.

Las iteraciones hacen referencia a pasos en el flujo de trabajo, y los incrementos a crecimientos en el producto.

Las iteraciones deben estar controladas. Esto significa que deben seleccionarse y ejecutarse de una forma planificada.

Los desarrolladores basan la selección de lo que implementarán en cada iteración en dos cosas: el conjunto de casos de uso que amplían la funcionalidad, y en los riesgos más importantes que deben mitigarse. **Ver Figura 1.2**

Beneficios del enfoque iterativo

- La iteración controlada reduce el riesgo a los costes de un solo incremento
- Reduce el riesgo de retrasos en el calendario atacando los riesgos más importantes.
- Acelera el desarrollo. Los trabajadores trabajan de manera más eficiente al obtener resultados a corto plazo.

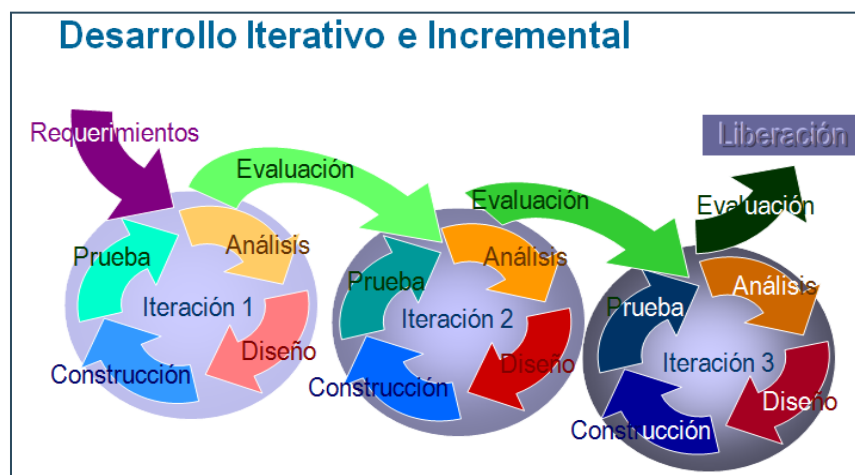


Figura 1.2 Desarrollo Iterativo Incremental

Donde cada iteración es un ciclo de desarrollo que termina con la liberación de un subconjunto del producto final. Cada iteración pasa a través de todos los flujos de trabajo del desarrollo de software.

- Toma de requerimiento
- Análisis & Diseño
- Implementación
- Pruebas
- Documentación
- Cada liberación iterativa es una “pieza” evaluada y documentada del sistema total.

1.3.2 Modelo Incremental

El modelo incremental es un proceso que se divide en 4 partes: Análisis, Diseño, Código y Prueba. Sin embargo, para la producción del Software, se usa el principio de trabajo en cadena o “Pipeline”, utilizado en muchas otras formas de programación. Con esto se mantiene al cliente en constante contacto con los resultados obtenidos en cada incremento. Es el mismo cliente el que incluye o desecha elementos al final de cada incremento a fin de que el software se adapte mejor a sus necesidades reales.

El Modelo Incremental es particularmente útil cuando no se cuenta con una dotación de personal suficiente. Los primeros pasos los pueden realizar un grupo reducido de personas y en cada incremento se añadirá personal, de ser necesario. Por otro lado los incrementos se pueden planear para gestionar riesgos técnicos.

Un modelo incremental lleva a pensar en un desarrollo modular, con entregas parciales del producto Software denominados “incrementos” del sistema, que son escogidos en base a prioridades predefinidas de algún modo. El modelo permite una implementación con refinamientos sucesivos (ampliación y/o mejora). Con cada incremento se agrega nueva funcionalidad o se cubren nuevos requisitos o bien se mejora la versión previamente implementada del producto software. [6]. Ver figura 1.3

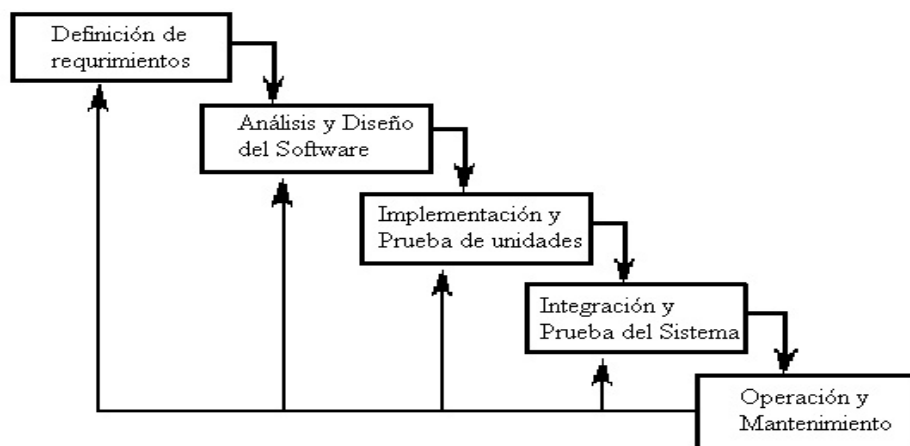


Figura 1.3 Fases del Modelo Incremental

Características

- Se evitan proyectos largos y se entrega “algo de valor” a los usuarios con cierta frecuencia.
- El usuario se involucre más.
- Difícil de evaluar el costo total.

Ventajas:

- Con un paradigma incremental se reduce el tiempo de desarrollo inicial, ya que se implementa la funcionalidad parcial.
- También provee un impacto ventajoso frente al cliente, que es la entrega temprana de partes operativas del Software.
- El modelo proporciona todas las ventajas del modelo en cascada realimentado, reduciendo sus desventajas sólo al ámbito de cada incremento.

Desventajas:

- El modelo Incremental no es recomendable para casos de sistemas de tiempo real, de alto nivel de seguridad, de procesamiento distribuido, y/o de alto índice de riesgos.
- Requiere de mucha planeación, tanto administrativa como técnica.

1.4 BASES DE DATOS

La base de datos es una serie de datos organizados y relacionados entre sí, que permiten el acceso a ellos y un conjunto de programas que manipulen ese conjunto de datos de manera automática.

1.4.1 Introducción y definición de Bases de Datos

Las aplicaciones informáticas de los años sesenta acostumbraban a darse totalmente por lotes (Batch) y estaban pensadas para una tarea muy específica relacionada con muy pocas entidades tipo.

Cada aplicación (una o varias cadenas de programas) utilizaba ficheros de movimientos para actualizar (creando una copia nueva) y/o para consultar uno o dos ficheros maestros o, excepcionalmente, más de dos. Cada programa trataba como máximo un fichero maestro, que solía estar sobre cinta magnética y en consecuencia, se trabajaba con acceso secuencial.

Cada vez que se le quería añadir una aplicación que requería el uso de algunos de los datos que ya existían y de otros nuevos, se diseñaba un fichero nuevo con todos los datos necesarios (algo que provocaba redundancia) para evitar que los programas tuviesen que leer muchos ficheros.

El acceso on-line y la utilización eficiente de las interrelaciones exigían estructuras físicas que diesen un acceso rápido, como por ejemplo los índices, las multilistas, las técnicas de hashing, etc.

Estos conjuntos de ficheros interrelacionados, con estructuras complejas y compartidos por varios procesos de forma simultánea (unos on-line y otros por lotes), recibieron al principio el nombre de Data Banks, y después, a inicios de los años setenta, el de Data Bases. Aquí los denominamos bases de datos (BD).

Definición 1:"Colección de datos interrelacionados almacenados en conjunto sin redundancias perjudiciales o innecesarias; su finalidad es servir a una o más aplicaciones de la mejor forma posible; los datos se almacenan de modo que resulten independientes de los programas que los usan; se emplean métodos bien determinados para incluir nuevos datos y para modificar o extraer los datos almacenados". [1]

Definición 2:"Colección integrada y generalizada de datos, estructurada atendiendo a las relaciones naturales de modo que suministre todos los caminos de acceso necesarios a cada unidad de datos con objeto de poder atender todas las necesidades de los diferentes usuarios". [1]

1.4.2 DBMS

Es la sigla en inglés de Sistema de Administración de Bases de Datos, que corresponde al software que maneja todos los accesos a la Base de Datos, es decir cada solicitud de acceso de un usuario al DBMS es interpretada e inspeccionadas las correspondencias, generando, a continuación, una respuesta coherente a las necesidades de la pregunta. La interfaz con el usuario es el límite de acceso que tiene un Usuario común a la Base, todo lo que está bajo este límite es transparente (desconocido) para él.

Funciones:

Debe permitir al administrador de la base especificar los datos que la integran, su estructura y las relaciones que existen entre ellos, las reglas de integridad semántica, los controles a efectuar antes de autorizar el acceso a la base, etc., así como las características de tipo físico y las vistas lógicas de los usuarios. Esta función la realiza el lenguaje de definición de datos (DDL), propio del DBMS, y debe ser capaz de definir las estructuras de datos a los tres niveles (nivel externo, nivel lógico global o conceptual y nivel interno).

A nivel interno se define:

- Espacio reservado para la base de datos (volúmenes, cilindros y pistas)
- Longitud de los campos
- Modo de representación de los datos (binario, decimal, alfanumérico, etc.)
- Caminos de acceso como punteros e índices.

A nivel externo y conceptual, la función de descripción proporciona los instrumentos para la definición de entidades, su identificación, atributos, interrelaciones entre ellas, autorizaciones de acceso, restricciones de integridad, etc.

1.5 MODELO DE BASES DE DATOS

Para describir la estructura de una base de datos es necesario definir el concepto de modelo de datos. Un modelo de datos es una colección de herramientas conceptuales para describir datos, relacionales entre ellos, semántica asociada a los datos y restricciones de consistencia.

Modelos lógicos basados en objetos. Utilizados para describir datos en el nivel conceptual y de visión. Se caracterizan por proporcionar una capacidad de estructuración flexible y permiten especificar restricciones de datos explícitamente (ejemplo: modelo entidad-relación, modelo orientado a objetos)

Modelos lógicos basados en registros. Utilizados para describir datos en el nivel conceptual y físico, se utiliza para especificar la estructura lógica de la base de datos. Estos modelos se llaman así porque la base de datos está estructurada en registros de formato fijo de varios tipos [2]

1.5.1 Modelo Entidad- Relación (E-R)

El modelo entidad-relación es el modelo conceptual más utilizado para el diseño conceptual de bases de datos. Fue introducido por Peter Chen en 1976. El modelo entidad-relación está formado por un conjunto de conceptos que permiten describir la realidad mediante un conjunto de representaciones gráficas y lingüísticas.

Originalmente, el modelo entidad-relación sólo incluía los conceptos de entidad, relación y atributo. Más tarde, se añadieron otros conceptos, como los atributos compuestos y las jerarquías de generalización, en lo que se ha denominado modelo entidad-relación extendido, como lo muestra la **Figura 1.4**.

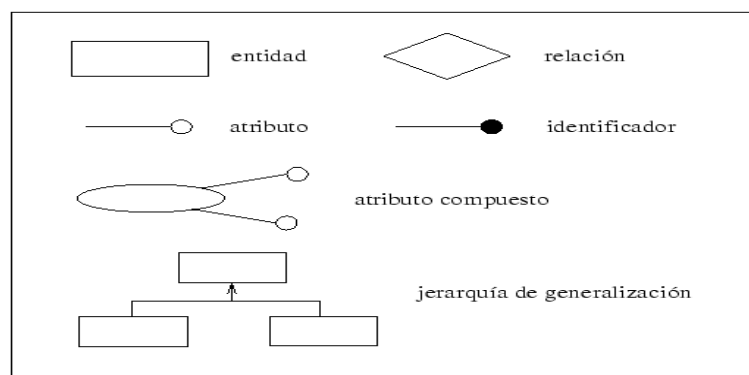


Figura 1.4 Elementos gráficos que representan un modelo E-R

Entidad Cualquier tipo de objeto o concepto sobre el que se recoge información: cosa, persona, concepto abstracto o suceso. Por ejemplo: coches, casas, empleados, clientes, empresas, oficios, diseños de productos, conciertos, excursiones, etc. Las entidades se representan gráficamente mediante rectángulos y su nombre aparece en el interior. Un nombre de entidad sólo puede aparecer una vez en el esquema conceptual.

Relación (interrelación) Es una correspondencia o asociación entre dos o más entidades. Cada relación tiene un nombre que describe su función. Las relaciones se representan gráficamente mediante rombos y su nombre aparece en el interior.

Las entidades que están involucradas en una determinada relación se denominan entidades participantes. El número de participantes en una relación es lo que se denomina grado de la relación. Por lo tanto, una relación en la que participan dos entidades es una relación binaria; si son tres las entidades participantes, la relación es ternaria; etc.

1.5.2 Modelo Relacional

UML no es una metodología, sino más bien es un lenguaje, una notación que permite visualizar, especificar, construir y documentar el modelado de sistemas; sea cual fuere el ciclo de vida elegido para el análisis, diseño e implementación del mismo.

UML es de reciente aparición y al ser no propietario, es usado y refinado por muchas empresas, grupos de investigadores y desarrolladores a nivel mundial.

Se cree que los conceptos orientados a objetos y bases de datos relacionales son mutuamente inconsistentes y no se pueden combinar. De hecho lo contrario! A través de la utilización flexible de una base de datos relacional orientada a objetos (OO) proporciona una buena aplicación. El mismo modelo se puede utilizar para desarrollar un código de programación y la estructura de base de datos relacional.

1.5.2.1 Llaves Primarias y Foráneas

Llaves Foráneas

Una base de datos relacional normalmente contiene más de una relación, para poder representar distintos tipos de hechos que suceden en el mundo real. Por ejemplo, podríamos tener una pequeña base de datos que contuviese dos relaciones: una denominada EMPLEADOS, que almacenaría datos de los empleados de una empresa, y otra con el nombre DESPACHOS, que almacenaría los datos de los despachos que tiene la empresa.

Debemos considerar también que entre los distintos hechos que se dan en el mundo real pueden existir lazos o vínculos. Por ejemplo, los empleados que trabajan para una empresa pueden estar vinculados con los despachos de la empresa, porque a cada empleado se le asigna un despacho concreto para trabajar.

Llaves Primarias

A continuación estudiaremos con detalle las reglas de integridad del modelo relacional, reglas que todo SGBD relacional debe obligar a cumplir.

-Regla de integridad de unicidad de la clave primaria

La regla de integridad de unicidad está relacionada con la definición de clave primaria. Concretamente, establece que toda clave primaria que se elija para una relación no debe tener valores repetidos.

1.6 NORMALIZACIÓN

La normalización se podría definir de manera sencilla como el proceso de organizar y estructurar los datos de forma que se minimice la redundancia. Este proceso se basa en un conjunto de pautas, las reglas de normalización, que disminuyen el riesgo de tener un diseño de BD defectuoso. Estas reglas se aplican al modelo relacional a partir del modelo Entidad-Relación.

1.6.1 Definición de Normalización

La normalización de datos es un procedimiento que asegura que un modelo de datos se ajusta a algunos estándares útiles. Para los datos y los modelos entidad-relación, estos estándares se han definido para minimizar la duplicación de datos, proporcionar la flexibilidad necesaria para soportar requisitos funcionales y para permitir que el modelo se estructure de una amplia variedad de diseños alternativos de bases de datos.

Para comprobar que un modelo entidad-relación tenga todas sus entidades claramente identificadas, tiene que estar completamente normalizada.

La normalización es un conjunto de reglas que sirven para ayudar a los diseñadores a desarrollar un esquema que minimice los problemas de lógica. Cada regla está basada en la que le antecede. La normalización se adoptó porque el viejo estilo de poner todos los datos en un solo lugar, como un archivo o una tabla de la base de datos, era ineficiente y conducía a errores de lógica cuando se trataba de manipular los datos

Otra ventaja de la normalización de base de datos es el consumo de espacio. Una base de datos normalizada ocupa menos espacio en disco que una no normalizada. Hay menos repetición de datos, lo que tiene como consecuencia un mucho menor uso de espacio en disco. El proceso de normalización tiene un nombre y una serie de reglas para cada fase. Esto puede parecer un poco confuso al principio, pero poco a poco se va entendiendo el proceso, así como las razones para hacerlo de esta manera.

1.6.2 Formas Normales

Existen básicamente tres niveles de normalización: Primera Forma Normal (1NF), Segunda Forma Normal (2NF) y Tercera Forma Normal (3NF). Cada una de estas formas tiene sus propias reglas. Cuando una base de datos se conforma a un nivel, se considera normalizada a esa forma de normalización. No siempre es una buena idea tener una base de datos conformada en el nivel más alto de normalización, puede llevar a un nivel de complejidad que pudiera ser evitado si estuviera en un nivel más bajo de normalización.

En la **Tabla 1.1** se describe brevemente en qué consiste cada una de las reglas, y posteriormente se explican con más detalle. [1]

Regla	Descripción
Primera Forma Normal (1FN)	Incluye la eliminación de todos los grupos repetidos.
Segunda Forma Normal (2FN)	Asegura que todas las columnas que no son llave sean completamente dependientes de la llave primaria (PK).
Tercera Forma Normal (3FN)	Elimina cualquier dependencia transitiva. Una dependencia transitiva es aquella en la cual las columnas que no son llave son dependientes de otras columnas que tampoco son llave.

Tabla 1.1 Formas Normales

1.6.3 Primera Forma Normal

La regla de la Primera Forma Normal establece que las columnas repetidas deben eliminarse y colocarse en tablas separadas. Poner la base de datos en la Primera Forma Normal resuelve el problema de los encabezados de columna múltiples. Muy a menudo, los diseñadores de bases de datos inexpertos harán algo similar a la tabla no normalizada.

Una y otra vez, crearán columnas que representen los mismos datos. La normalización ayuda a clarificar la base de datos y a organizarla en partes más pequeñas y más fáciles de entender.

En lugar de tener que entender una tabla gigantesca y monolítica que tiene muchos diferentes aspectos, sólo tenemos que entender los objetos pequeños y más tangibles, así como las relaciones que guardan con otros objetos también pequeños.[1]

1.6.4 Segunda Forma Normal

La regla de la Segunda Forma Normal establece que todas las dependencias parciales se deben eliminar y separar dentro de sus propias tablas. Una dependencia parcial es un término que describe a aquellos datos que no dependen de la llave primaria de la tabla para identificarlos. [1]

1.6.5 Tercera Forma Normal

Una tabla está normalizada en esta forma si todas las columnas que no son llave son funcionalmente dependientes por completo de la llave primaria y no hay dependencias transitivas. Comentamos anteriormente que una dependencia transitiva es aquella en la cual existen columnas que no son llave que dependen de otras columnas que tampoco son llave.[1]

CAPÍTULO 2: ANÁLISIS Y DISEÑO (ORIENTADO A OBJETOS)

2.1 PLANTEAMIENTO DEL PROBLEMA

Actualmente se lleva a cabo un control manual de la información en lo que se refiere a actividades realizadas por el personal administrativo del plantel que son los responsables de actualizar el inventario. El departamento de almacén es el encargado de registrar equipos y materiales de computación (stock), transferencia de equipos y materiales, lo que ha dificultado la obtención de reportes del inventario y las actividades realizadas diariamente.

Toda esta información es difícil de manejar, a la vez que causa una pérdida de tiempo debido a que se encuentra archivada en una gran cantidad de documentos y a la hora de presentar informes se torna un proceso muy complicado.

Los inconvenientes que se generan con el actual manejo de la información son los siguientes:

1. No se cuenta con una Base de Datos que permita controlar la información de manera adecuada por lo que el proceso para acceder a la información existente es lento y conflictivo.
2. No se tiene un registro actualizado de los equipos de cómputo y dispositivos que se encuentran en el plantel.
3. No se tienen reportes periódicos de los materiales que se han dado de baja.
4. No se cuenta con un control de licencias instaladas y de los equipos en las que están instaladas.
5. No existe un seguimiento del estado de los equipos que presentan daños físicos.
6. El proceso de solicitud de orden de trabajo/mantenimiento para llevar a cabo un diagnóstico de los equipos que muestran algún daño físico es tardado ya que la solicitud se realiza de manera manual o por fax.
7. El manejo del inventario físico es poco práctico y no está automatizado. Para controlar la salida y entrada de todo el hardware, ya sea en uso o desuso, se debe de contar con herramientas orientadas a esta labor.
8. No es posible la generación de reportes sobre el trabajo realizado por el personal de soporte técnico
9. Cada computadora al que se le brinda mantenimiento, ya sea correctivo o preventivo, no posee su respectivo control de mantenimiento, lista de fallas recurrentes y reportes de soluciones, es decir, un historial de la misma.

2.2 OBJETIVO GENERAL

Desarrollar un sistema para automatizar la gestión de todas las operaciones de inventarios de laboratorios de Cómputo y Reportes de Fallas, el sistema será capaz de realizar por separado las tareas de inventario y reportes de fallas con el fin de facilitar el trabajo de las personas que manejan esta información.

Objetivos Específicos

Modelar y seleccionar una Base de Datos que nos permita administrar toda la información.

- Crear un módulo para el manejo de Inventario que se encargue de interactuar con el manejador de base de datos para poder realizar consultas sobre los materiales existentes, actualizar, modificar o eliminar información.
- Crear un módulo para la generación de Órdenes de mantenimiento solicitadas por el usuario.
- Elaborar la especificación de requerimientos del proyecto.
- Realizar el análisis de los requerimientos
- Diseñar la base de datos.
- Diseñar la aplicación basada en Web.
- Implementar los diseños realizados.
- Realizar pruebas del sistema.

2.3 REQUERIMIENTOS DEL PROBLEMA

En esta etapa se identificarán los diferentes roles que podrán cumplir cada uno de los usuarios de la aplicación.

Los usuarios juegan roles importantes en cada intercambio de información con el sistema, es muy importante tener identificado el rol de cada usuario, ya que serán ellos los que entregarán su conformidad con respecto al proceso en el que participan.

Luego de cada rol se identificarán las tareas que deberá soportar la aplicación, se deben definir escenarios que son descripciones narrativas de cómo la aplicación será utilizada.

Cada usuario deberá explicar textualmente los escenarios que describen su tarea.

Se representa la interacción entre el usuario y el sistema, agrupando las tareas representadas en los escenarios existentes.

Requerimientos Funcionales

El sistema permitirá consultar, crear, editar e imprimir la información registrada de los equipos, de cómputo.

- Registrará las características técnicas del hardware que compone cada equipo.
- Tendrá el control de licencias de software.
- Registrará las órdenes de mantenimiento realizadas por el usuario al personal de Soporte Técnico.
- Generará una bitácora la cual contendrá todas las Órdenes de Mantenimiento finalizadas.
- Creará un Backup de artículos eliminados.

- El manejo del sistema será diseñado para soportar diferentes tipos de usuarios, los cuales podrán realizar actividades dependiendo de los servicios que les sean asignados.

Requerimientos no Funcionales

- El software debe estar desarrollado (modelado y construido) en una herramienta (Rational Rose).
- Uso de la herramienta UML.
- El desarrollo del sistema en lenguaje de programación PHP y utilizando el manejador de bases de datos MySQL

2.4 ALCANCES

El alcance de este proyecto será que los sistemas de inventario y reportes podrán trabajar por separado. Aprovechando la relación que existe entre ellos, para facilitar el uso del sistema de reportes que es lo que más se utiliza con más frecuencia.

Dados los requerimientos de los usuarios del sistema, se solicitó que los sistemas estuvieran manejando la misma base de datos, pero que a la vez mantuvieran autonomía en el manejo.

Módulos del Sistema

Después de analizar las actividades y requerimientos del problema, se ha decidido desarrollar los siguientes módulos.

- Registro de usuarios
- Manejo de Inventario
- Órdenes de mantenimiento
- Almacén
- Compras

1.- Módulo Inventario

La funcionalidad de la información de inventario y reportes de fallas de equipo de cómputo que realiza el usuario parecen funcionar para dos tareas independientes una de la otra, pero no es así, ya que los equipos o dispositivos a las que se les brinda el servicio, son las mismas que se encuentran en el inventario. Es importante observar que ambos sistemas requieren de la misma información para poder trabajar adecuadamente.

El sistema inventario tiene dos funciones:

- 1.- Mantener el Control del Inventario de los Equipos, Dispositivos, Hardware y Software que hay en los laboratorios de cómputo del plantel.
- 2.-Registrar las Órdenes de Mantenimiento generadas por el usuario.

Los elementos que pertenecen al inventario son: Dispositivos, hardware, software, computadoras completas, discos duros, memorias RAM etc.

Computadoras Completas

Son los equipos que se encuentran en cada área del inventario.

- Cada equipo tiene un nombre de modelo propio (fecha)
- Cada equipo está asociado a una marca dentro de dicha tabla
- Cada equipo tiene un No. de Serie.
- Cada equipo muestra su descripción general (No. Equipo, monitor, teclado, mouse)
- Cada equipo muestra sus características Técnicas como: (Disco duro, Memoria, Procesador, Motherboard, Tarjeta Red)
- Cada equipo tiene asociado el Software que tiene instalado (Sistema operativo, Programas etc.)
- Cada equipo muestra el estado actual en el que se encuentra.
- Cada equipo está asociado a un área o departamento.
- Cada equipo tiene asignado un responsable.
- Cada equipo tiene asignado una solicitud de mantenimiento.

Dispositivos

Son los artículos como: Impresoras, proyectores, escáneres, reguladores que se encuentran en el inventario.

- Cada dispositivo tiene asociado un nombre y No. de Serie
- Cada dispositivo tiene asociado una marca y modelo
- Cada dispositivo tiene un código
- Cada dispositivo está asociado a una área o departamento

Software

Son los programas instalados en cada uno de los equipos.

- El software tiene asociado un nombre y No. de serie.
- El software tiene asociado un No. Licencia
- El software tiene asociada una versión.
- El software tiene asociado el No. equipo en el que está instalado
- El software tiene asociada la fecha de compra.
- software tiene asociada la existencia

De tal forma que el sistema tiene las siguientes funciones:

- Agregar, eliminar y actualizar equipos
- Agregar, eliminar y actualizar piezas de hardware
- Agregar, eliminar y actualizar dispositivos
- Agregar, eliminar y actualizar existencias
- Agregar, eliminar y actualizar software
- Mostrar listas de equipos, existencias, piezas, accesorios y software
- Obtener el estado de una existencia específica
- Obtener las ubicaciones de todo el plantel

2.-Reporte de Fallas.

Desempeña una labor fundamental dentro de las funciones del sistema de Inventario, además que es el módulo más utilizado permitirá registrar todas las solicitudes de soporte técnico para ser atendidas por un técnico.

Módulo Reporte de Mantenimiento.

Permitirá el registro de las órdenes de mantenimiento hechas por usuario.

- Un usuario registrado previamente en el sistema puede emitir una solicitud para soporte técnico.
- Una solicitud se realiza a equipos completos y dispositivos.
- Las solicitudes se encolan y solo el responsable de soporte técnico puede asignarlas y decidir cuáles son las que se atenderán primero y les asigna al técnico para que las atienda. Una solicitud desencadena un mantenimiento. Cuando los técnicos resuelven el problema, entonces el administrador cambia el estado del mantenimiento a “Finalizado”. Con esto se genera una Bitácora de Reportes.
- Las Bitácoras no pueden ser eliminadas por razones de seguridad y para guardar siempre el registro de cada máquina o dispositivo al que se realizó el mantenimiento.
- Cada equipo, pieza de hardware o dispositivo tiene un estado y dependiendo de su funcionamiento puede cambiar.

Existen dos posibles estados para un equipo:

1. Funcionando
2. Dañado

La realización y desarrollo de un sistema brindará beneficios tales como:

1. Ahorrar tiempo y esfuerzo para llevar a cabo los inventarios físicos.

2. Reducir la interrupción en el trabajo de los usuarios en la actividad diaria.
3. Tener el registro de la configuración de los equipos de cómputo al momento de analizar la ocurrencia de un incidente.
4. Facilitar las labores de auditoría proporcionando una visión general de los componentes.
5. Mejorar los procesos de compra y toma de decisiones.
6. Reducir el riesgo de incurrir en multas y/o procesos legales por el uso de software ilegal y controlar cambios en piezas hardware.
7. También se pueden definir algunas sugerencias para el proceso de gestión de inventarios, asegurar que los usuarios de la institución vean reportes periódicos para que estén enterados de los cambios y en el caso de los procesos de compras, conozcan de primera mano información que les ayude en las actividades de compra.
8. El resultado de la gestión de inventarios contribuye a respaldar las políticas de seguridad del proceso en una organización que tienen como fin evitar la instalación de software sin licencia en los laboratorios de cómputo.

Limitaciones del Sistema

Cabe observar que como parte de este trabajo de tesis solo se contempla el desarrollo de los tres primeros módulos del sistema, debido a restricciones de tiempo y al plazo que se requiere para completar el proyecto.

2.5 METODOLOGÍA

Se utilizó la metodología de desarrollo iterativo e incremental (Rup) que consiste en construir sistemas haciendo aproximaciones que se acercan progresivamente a la solución ideal. El desarrollo procede como una serie de liberaciones que evolucionan hacia el sistema final. Utilizando una notación estandarizada Uml.

Beneficios:

- Reducción de riesgos basados en retroalimentación temprana
- Mayor flexibilidad para integrar los requerimientos
- Aumenta la calidad del software.

Características:

- Dirigido por casos de uso
- Centrado en la arquitectura
- Ciclo de vida iterativo e incremental. Cada ciclo es una sucesión de fases (iniciación, elaboración, construcción, transición).

Inicio.-Establece el caso del negocio para desarrollo del sistema. En esta fase se definen las principales funciones del sistema para los usuarios más importantes y se estiman los costos del producto.

Elaboración.- Análisis del dominio del problema, establecimiento de la base arquitectónica y evaluación de riesgos preliminares. En esta fase se especifica en detalle la mayoría de los casos de uso y se diseña la arquitectura del sistema, la cual especifica en forma de vistas todos los modelos del sistema.

Construcción.-Desarrollo incremental de un producto de software completo que éste listo para hacer transición hacia la comunidad de usuarios.

- Se construye el producto
- Se utilizan la mayor parte de los recursos
- Al finalizar se cubren la mayor parte de los casos de uso
- Se realiza la primera entrega a los usuarios

Transición.-Pasando el producto de software a la comunidad de usuarios.

- El producto existe en su primera versión
- Algunos usuarios prueban el producto
- A su vez cada fase puede tener varias iteraciones cada una con sus 5 Flujos de trabajo.

Requerimientos

- Análisis y diseño
- Implementación
- Pruebas
- Documentación

2.6 ANÁLISIS DEL SISTEMA

En esta etapa se presentó la conceptualización del dominio del problema, se usó la documentación disponible (descripción del problema y entrevistas con los usuarios etc.)

Se tomó como base el modelo de CU y el modelo conceptual de clases

Se definieron clases de entidad, límite y control VOPC para cada caso de uso, la cuales representan las conexiones entre las clases incluyendo las relaciones.

Boundary.- Límites del sistema (pantallas, reportes)

Entity.-Información del sistema (modelo conceptual)

Control.-Coordinación del flujo de los CU (flujo de eventos)

Se utilizaron escenarios de CU para ordenar, dirigir y ejecutar el proceso, a través de diagramas de secuencia y colaboración.

Para completar el análisis de CU se utilizaron las tarjetas CRC que nos permitieron descubrir clases.

2.6.1 Diccionario de datos del Modelo

En la **tabla 2.2** muestra la descripción de la información que se almacenó en la Base de Datos

Palabra	Definición
Inventario	Información almacenada de los artículos registrados y números de serie.
Orden Mtto.	Orden de trabajo para diagnóstico de equipo que presenta algún daño.
Artículo	Los componentes que se registrarán en el sistema.
Trabajo Solicitado	Solicitudes de mantenimiento de equipo
Usuario	Contempla los datos personales de cada usuario registrado en el sistema.
Soporte Técnico	Lo conforman los técnicos que atenderán las órdenes de mantenimiento.
Software	Son los programas y licencias de software instalados en las computadoras
Hardware	Lo conforman memoria, CPU, tarjetas de red, discos duros
Equipo	Monitor, CPU, teclado, mouse.
Backup	Alberga los sucesos referentes al respaldo de los datos.
Estado	Dañado, funcionando.
Dispositivo	Proyector, impresora, escáner, regulador.
Área	Primaria, secundaria, preparatoria.
Reporte de Mtto.	Bitácora que registra las órdenes de mantenimiento.
Responsable	Lo conforman los encargados de cada laboratorio de cómputo.
Técnico	Son los responsables de cada área del departamento de soporte.
Estado de la Orden	La orden de Mtto. puede estar en Espera, Atendida o Finalizada

Tabla 2.2 Diccionario de Datos

2.6.2 Diagrama de Casos de Uso general del Sistema

En la **Figura 2.1** se muestra el diagrama general de casos de uso del sistema de Inventario. El objetivo principal del diagrama de casos de uso es visualizar que las funciones del sistema a las que puede acceder cada actor, organizando el comportamiento del sistema y muestra las relaciones entre cada caso de uso.

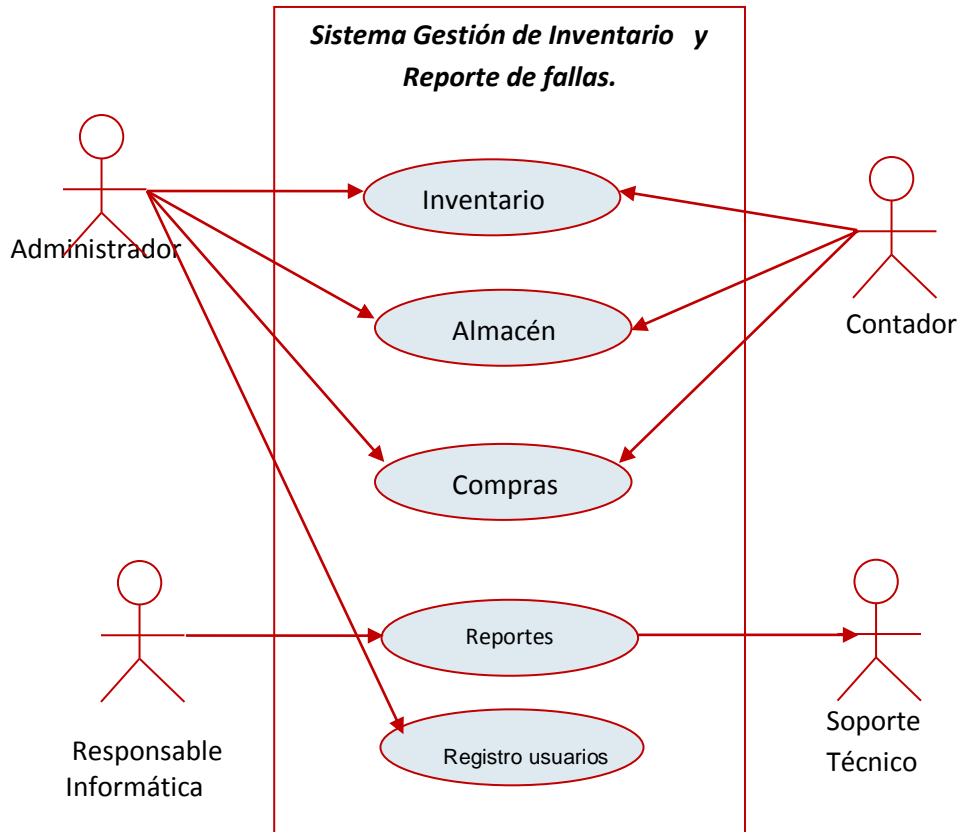


Figura 2.1 Diagrama de casos de uso para la especificación de requisitos del sistema

Descripción de Actores

Nombre de Actor: Administrador

Definición: Es el encargado del sistema. Tendrá todos los permisos y libertad de acceso por el sistema.

- El administrador es el encargado de manipular la información contenida en el sistema.
- Tiene acceso a toda la información del sistema y es el único que puede modificar todo lo que requiera.

Nombre del Actor: Encargado de Informática

Definición: Es el encargado de los laboratorios de cómputo y su responsabilidad es reportar las fallas de los equipos o dispositivos que ahí se encuentran.

- El encargado de informática se comunicara con el técnico para informar que existe algún problema con los equipos o dispositivos del laboratorio de cómputo.

Nombre del Actor: Técnico

Definición: Es el responsable de gestionar los reportes de mantenimiento.

- El técnico de acuerdo a su área de trabajo se clasificará de la siguiente manera:
1.- Técnico (soporte Hardware, Redes)

Nombre del Actor: Técnico mantenimiento Hardware/software

Definición: Es el responsable de atender órdenes de trabajo.

- El técnico resolverá fallas relacionadas con el hardware y configuración de los equipos.
- El técnico es el encargado de atender, consultar, cerrar reportes.

Diagrama de caso de uso Gestión Inventario

El siguiente diagrama muestra de manera detallada las operaciones primordiales que podrá realizar el sistema de inventario, el usuario deberá logearse y acceder a ellas de acuerdo a sus privilegios de usuario. Alta, Baja, Modificación, Impresión. **Ver figura 2.2**

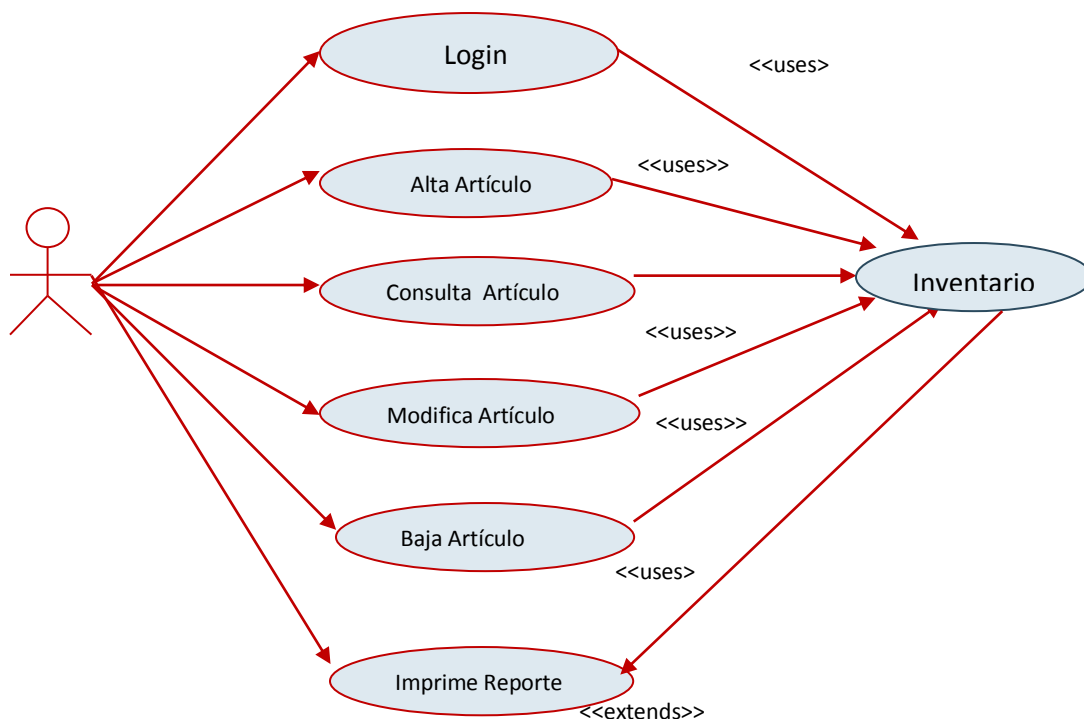


Figura 2.2 Caso de Uso: Gestión Inventario

Diagrama de Casos de Uso Reportes de Mantenimiento.

A continuación se muestra el diagrama de Casos de Uso General del sistema de Reportes; El cuál se puede observar como el usuario levanta el reporte, posteriormente un encargado del área de mantenimiento puede consultar los reportes para atenderlos. **Ver figura 2.3**

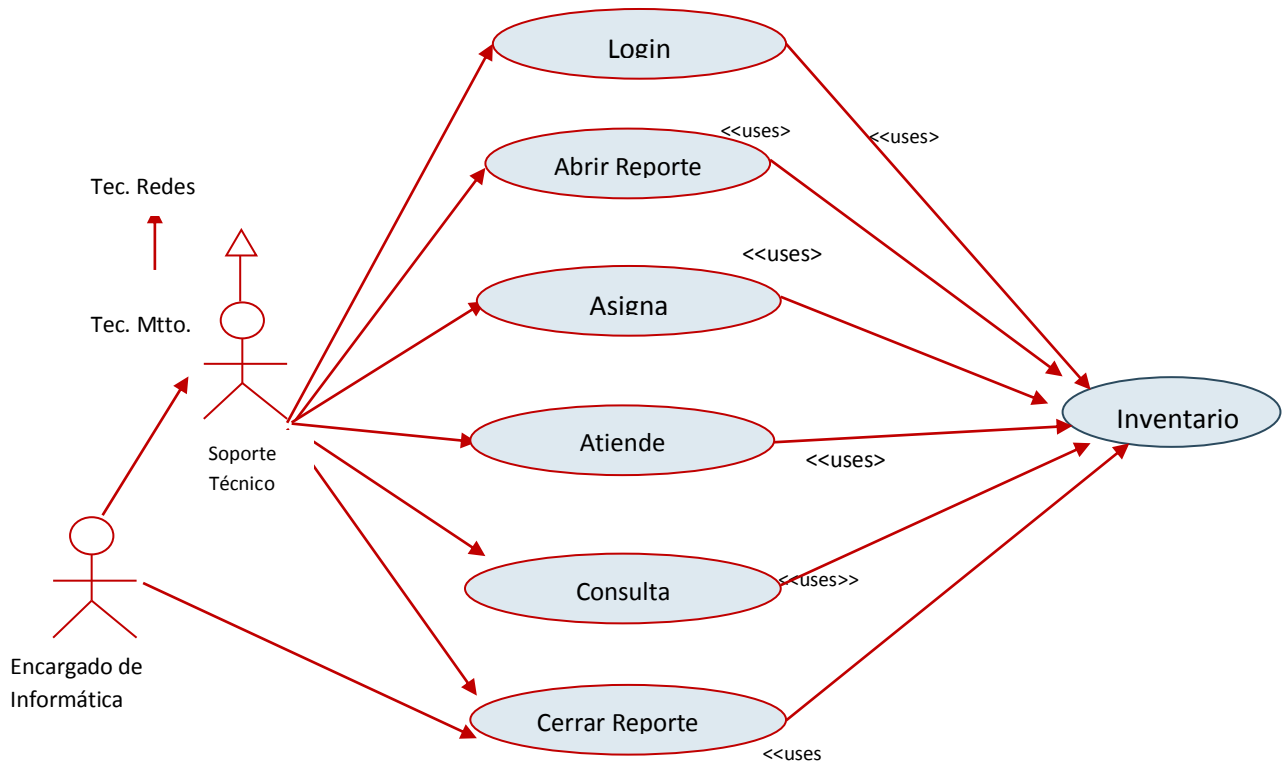


Figura 2.3 Caso de Uso: Reportes de fallas de Mantenimiento

2.6.3 Especificación de Casos de Uso

La especificación de Casos de Uso es una narración que describe el rol desempeñado por los actores en su interacción con el sistema. Para el sistema de inventario y reporte de fallas la especificación de Casos de Uso se realizó a través de los flujos de datos los cuales ayudan a interpretar y esclarecer los Casos de Uso.

El primer documento que se realizó es el modelo de casos de uso, este modelo de la especificación permite definir detalladamente las funciones que ofrece el sistema y que funcionalidad puede utilizar cada usuario.

Diagrama de Casos de Uso.- Permite definir los actores y usuarios del sistema y especificar que funcionalidades tendrá cada actor.

Caso de Uso: Inventario

A través de este caso de uso el actor puede realizar cualquiera de las siguientes operaciones sobre el stock.

1.-Alta Artículo

El usuario administrador es el encargado de registrar los equipos de cómputo y dispositivos que se van adquiriendo en un periodo de tiempo dentro del inventario.

2.-Consulta Artículo

Las consultas son fundamentales en el sistema ya que los usuarios que requieran conocer la existencia o la ubicación física de algún dispositivo o equipo de cómputo podrán realizar las consultas necesarias.

3.-Modificar Artículo

El usuario administrador será el único responsable de la modificación del inventario ya que es aquí donde se registra que material será dado de baja, reemplazado o asignado a algún laboratorio de cómputo.

4.-Dar de Baja Artículo

El usuario administrador es el encargado de dar de baja los equipos o dispositivos que ya no funcionen o que sean obsoletos.

Caso de Uso: Reporte de Mantenimiento.

A través de este caso de uso el usuario podrá realizar las siguientes operaciones:

Alta de Reporte

Para abrir un reporte el usuario se comunica al Departamento de Soporte, donde es atendido por un técnico, el cual pide al usuario los datos necesarios para levantar el reporte de falla y lo guarda en la Base de Datos.

Consulta de Reporte

Los técnicos podrán consultar los reportes de mantenimiento solicitados por los usuarios de los diferentes laboratorios de cómputo en caso de que haya surgido alguna falla de algún dispositivo o equipo de cómputo.

Cerrar Reporte

Finalmente el técnico acude con el usuario y si acepta que está satisfecho con el servicio que se realizó, el técnico introduce su Login y su password y cierra el reporte.

Flujo de Datos.-Gestión de Inventario

1.-Caso de uso Alta de Artículo
1.-El actor selecciona la opción de crear un nuevo artículo en el sistema.
3.- El actor llena el formulario de alta de artículo.
2.-El sistema muestra un formulario vacío para ser llenado con los datos de Artículos
4. El sistema almacena todos los datos registrados en el formulario
5.-El sistema da de alta el nuevo artículo en el sistema
6.- El sistema informa que el artículo se ha registrado
7.- El sistema regresa al caso de uso artículo
Posibles errores y alternativas
4.-El sistema detecta algún dato incorrecto en el formulario
5.-El sistema avisa al usuario que el dato introducido no es correcto
6.-El sistema regresa al paso 3
Posibles errores y alternativas

2.-Caso de uso Consulta de Artículo
1.-El actor necesita consultar algún campo de un artículo del sistema.
2.-El actor selecciona el artículo que desea consultar.
3.-El sistema muestra un formulario con los datos del artículo seleccionado
4.-El sistema vuelve al caso de uso artículos
Posibles errores y alternativas

3.-Caso de Uso Modificar Artículo

1.-El actor requiere modificar algún campo de un artículo del sistema.

3.-El actor edita el formulario con los datos que quiere modificar.

2.- El sistema muestra un formulario con los datos del artículo seleccionado.

4.- El sistema comprueba que todos los datos del formulario sean correctos.

5.- El sistema modifica los datos del artículo.

6.- El sistema informa que el artículo se ha modificado satisfactoriamente.

Posibles errores y alternativas

4.-El sistema detecta algún dato incorrecto en el sistema.

5.-El sistema avisa que ha surgido un error.

4.-Caso de Uso Baja de Artículo

El actor requiere dar de baja algún artículo del sistema.

1.- El actor selecciona el artículo que desea dar de eliminar.

2.- El actor indica que desea eliminar el artículo seleccionado.

4.- El actor confirma que desea eliminar definitivamente el artículo.

2.-El sistema pide confirmación al usuario antes de eliminar el artículo

4.- El sistema elimina definitivamente el artículo del sistema

5.- El sistema vuelve al caso de uso artículo

Posibles errores y alternativas

4.-El actor decide anular la operación de baja.

Flujo de Datos

1.-Reportes de Fallas

1.-El actor deberá estar registrado previamente en el sistema para poder realizar cualquier operación deseada.

2.- El actor según la opción que elija el usuario, el sistema comienza el caso de uso correspondiente.

3.-El actor selecciona la opción que desea

4.-El sistema muestra un formulario que le permitirá al actor realizar la operación deseada sobre el caso de uso Reportes de Mantenimiento.

Posibles errores y alternativas

2.-Caso de uso Abrir Reporte

1.-El actor selecciona la opción Reporte de la pantalla.

3.- El actor pide los datos al usuario para dar de alta el servicio a realizar.

2.-El sistema muestra un formulario vacío para ser llenado con los datos del usuario

4.- El sistema comprueba que todos los datos en el formulario sean correctos

5.- El sistema da de alta la nueva orden o solicitud de mantenimiento

6.- El sistema regresa al caso de uso abrir reporte

Posibles errores y alternativas

4.-El sistema detecta algún dato falta de completar en el formulario

5.-El sistema avisa al usuario que el dato introducido no es correcto

6.-El sistema regresa al paso 2

3.- Caso de Uso Cerrar Reporte
1.- El actor finaliza la solicitud de trabajo
2.-El actor corrobora con el responsable del área que el trabajo sea satisfactorio.
3.-El sistema muestra formulario con el estado del mantenimiento:(iniciado, en espera, finalizado)
4.-El sistema cierra el reporte
Posibles errores y alternativas
5.-Sin la autorización del responsable de soporte, el sistema no podrá cerrar el reporte.

2.6.4 Escenarios Caso de uso Inventario

Numeración: 1.1
Título: Alta Artículo
Precondiciones: No. de elementos que se adquieren, descripción marca, modelo no. Serie.
Quien lo comienza: El personal de Administrativo
Quien lo finaliza: El personal de Administrativo
Post condiciones:
Descripción: Se describe como artículos: Todos los dispositivos de hardware, equipos de cómputo impresoras, proyectores, escáneres, accesorios etc. Para que puedan ser etiquetados con su número de serie.

Numeración: 1.2
Título: Consulta artículo
Precondiciones
Quien lo comienza: Cualquier usuario que este registrado en la base de datos
Quien lo finaliza: El usuario que inicio
Post condiciones:
Descripción
Cualquier usuario registrado que requiera consultar la existencia, ubicación física, No. De serie o estado de un equipo o dispositivo, si fue dado de baja.

Numeración: 1.3
Título: Modificar artículo
Precondiciones: El material debe estar registrado.
Quien lo comienza: El personal Administrativo
Quien lo finaliza: El personal Administrativo
Post condiciones:
Descripción: Los datos de los artículos que están almacenados en la base de datos Inventario pueden ser modificados (No. Serie, marca, modelo, capacidad, ubicación etc).

Numeración: 1.4
Título: Baja de artículos
Precondiciones: .-Estar registrado en la base de datos y tener la orden de baja proporcionada por el técnico a cargo.
Quien lo comienza: El personal de Administrativo
Quien lo finaliza: Administrativo
Descripción: Se dará de baja un equipo o dispositivo cuando este obsoleto, averiado o deje de funcionar. El responsable de mantenimiento autorizará y determinará si se da de baja

Escenarios Caso de uso reportes de mantenimiento

Numeración: 1.1
Título: Abrir reportes
Precondiciones: Estar registrado en el sistema
Quien lo comienza: Técnico
Quien lo finaliza: Técnico
Post condiciones:
Descripción: El técnico recibe los datos del usuario para llenar el formulario de reportes de orden de trabajo. Los datos del formulario a llenar son:(Depto. o área, nombre del usuario, Fecha de solicitud, trabajo solicitado, fecha para atención, área, status de la solicitud).

Numeración: 1.2
Título: Consulta Reporte
Precondiciones: Estar registrado en el sistema
Quien lo comienza: Cualquier usuario que este registrado.
Quien lo finaliza: Técnico
Post condiciones: Técnico
Descripción Cualquier usuario registrado que requiera consultar los reportes de orden de trabajo ya sea por realizar, pendientes, o finalizados.

Numeración: 1.5
Título: Cerrar reporte
Precondiciones: .-Estar registrado, haber concluido con el usuario.
Quien lo comienza: Técnico
Quien lo finaliza: Técnico
Post condiciones:
Descripción: Para poder cerrar el reporte el técnico deberá haber concluido el trabajo solicitado y haber preguntado al usuario si está satisfecho con el servicio que se realizó, de no ser así la orden de trabajo quedará pendiente.

2.6.5 Diagrama de Clases

El diagrama de clases es el diagrama principal para el análisis y diseño, los diagramas se utilizan generalmente para facilitar el entendimiento de largas cantidades de datos y relación entre diferente partes de los datos. En el **diagrama 2.4** representa el modelo conceptual de clases. [15]

Perspectiva Conceptual

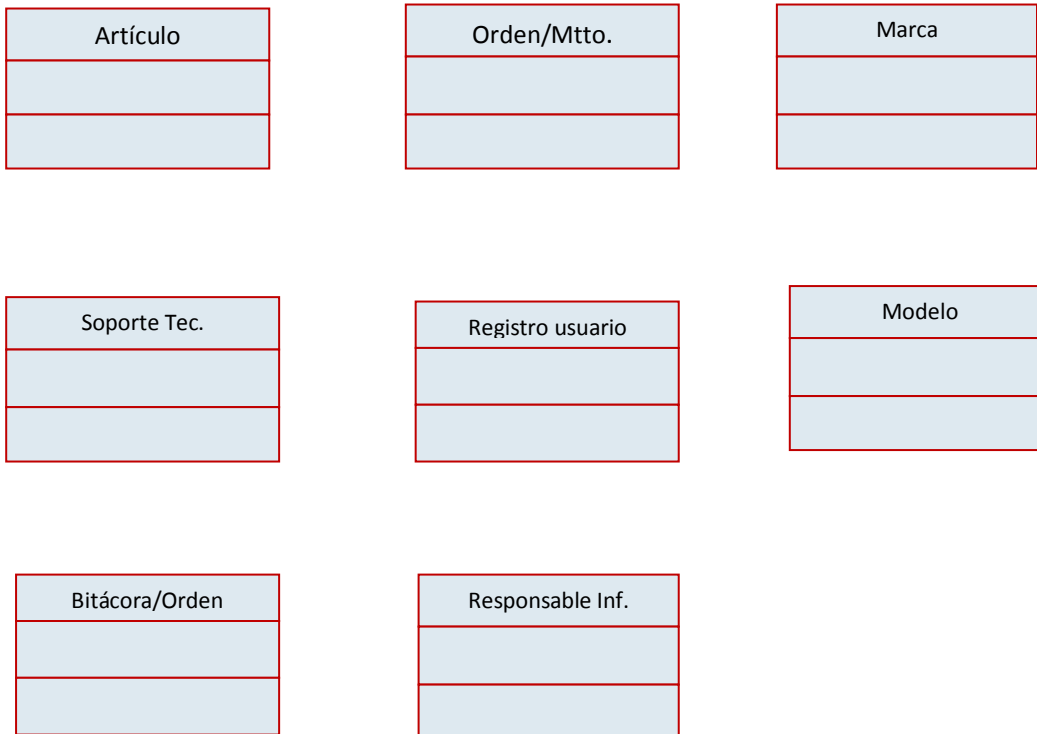
Es la que se usa para hacer el Modelo Conceptual; sirve para modelar conceptos puros o abstracciones, como clases, sin importar como sean estas clases. Los diagramas hechos desde esta perspectiva normalmente son muy sencillos y no tienen detalle alguno más que para nombrar las clases.

Perspectiva de Especificación

Es la que se usa para hacer el Modelo de Análisis, se define lo que deben hacer pero no como lo deben hacer.

Perspectiva de Implementación

Es la que se usa para hacer el Modelo de Diseño, los diagramas creados desde esta perspectiva deben ser muy detallados y complejos.



2.4 Diagrama de Clases, Modelo Conceptual

Diagrama de Clases VOPC

Las clases **VOPC** son el mecanismo que el UML provee para extender la notación cada clase de análisis debe tener un estereotipo. **Ver diagrama 2.5**

Estereotipos comunes

- **Clase de límite.- boundary** (pantallas o interfaces de usuario, reportes, otros sistemas)
- **Clase de entidad.- entity** (su información es independiente a sus alrededores, contiene información del sistema)
- **Clase de control.-** (sirve como intermediario entre las clases límite y control, coordina la secuencia de los flujos de eventos enviando mensajes a los objetos.) [14]





2.5 Diagrama de Clases VOPC

2.7 DISEÑO DEL SISTEMA

El diseño del sistema es el corazón de cualquier idea que se quiere implantar, si tenemos un diseño bien pensado no tendremos problemas al momento de desarrollarlo, contiene todos los puntos contemplados en el análisis anteriormente realizado, también se puede tomar como un momento de reflexión donde se pueden agregar nuevas características y siempre enfocándose a la pregunta que hace todo esto posible. ¿Qué se va a hacer?, si tenemos bien definido el ¿Qué?, entonces la siguiente pregunta será mucho más sencilla de responder.

Al desarrollar éste sistema se tuvo una entrevista con personas que están directamente involucradas en el entorno.

2.7.1 Diagrama de Clases de Diseño

Tomando en cuenta las relaciones que existen entre las distintas clases, la **Figura 2.6** nos brinda un enfoque más particular y detallado de cada una de las clases.

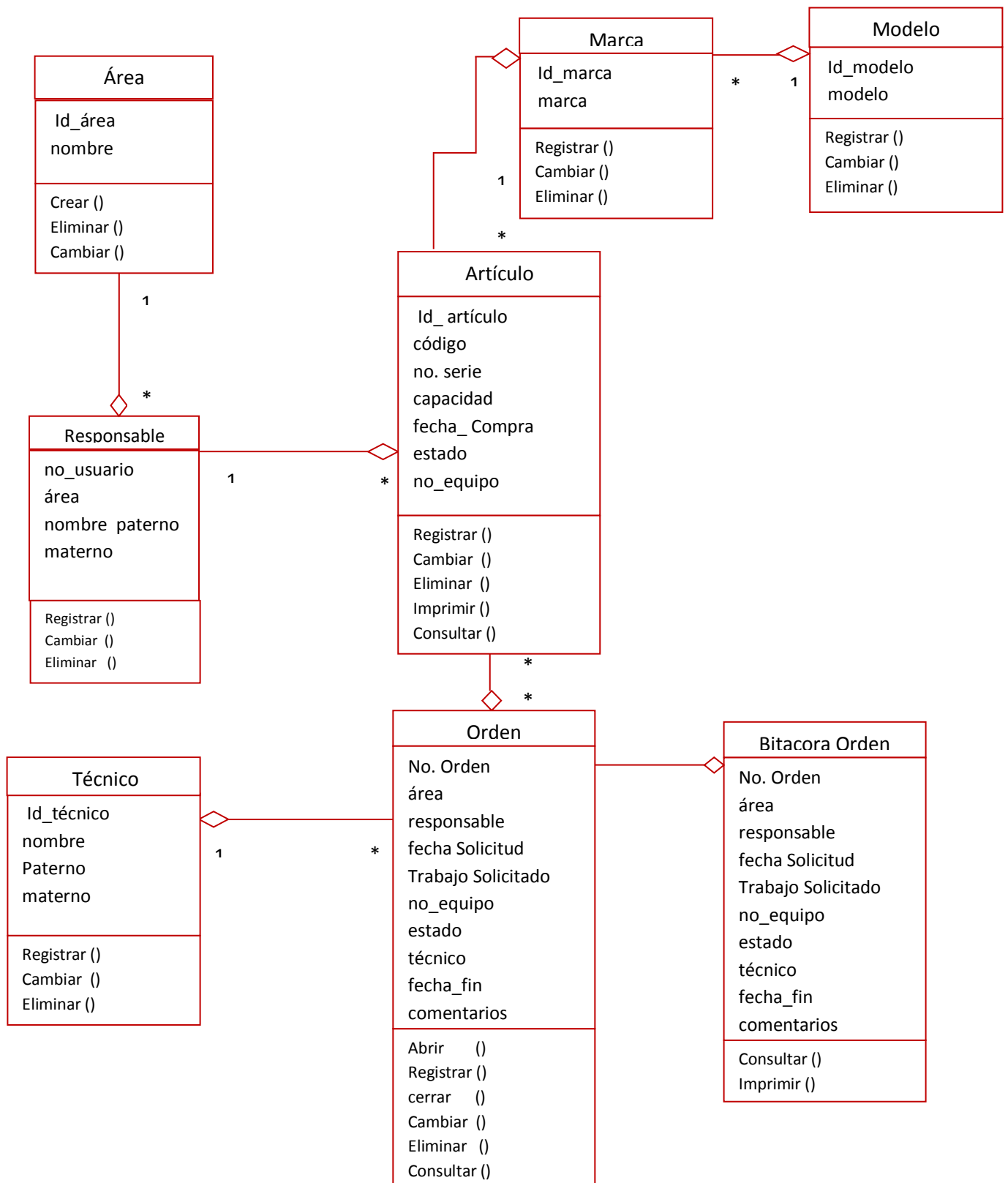
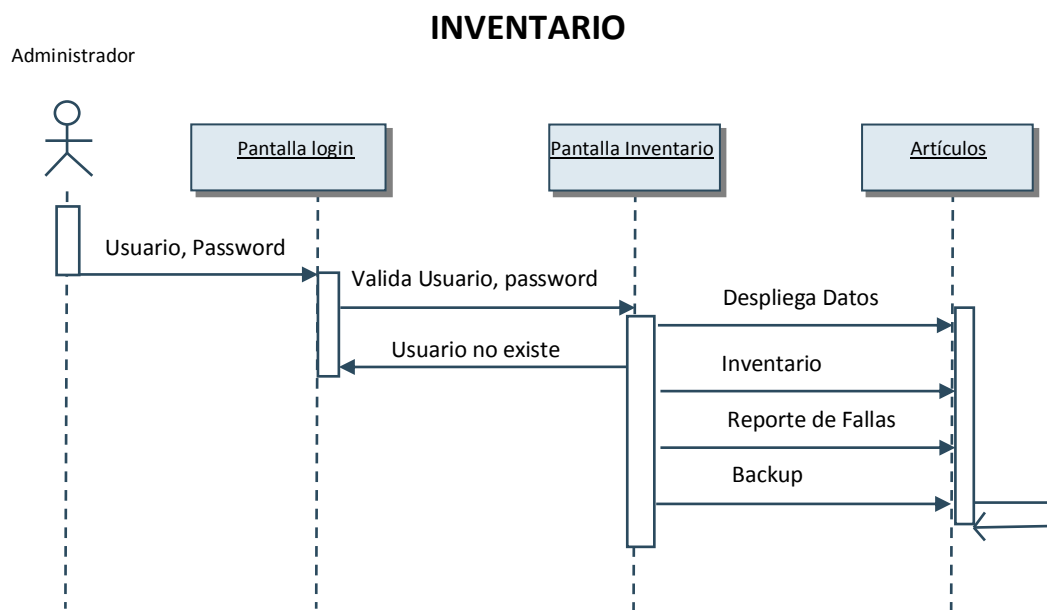


Figura 2.6 Clases y relaciones

2.7.2 Diagrama de Secuencia

Es un diagrama que resalta la interacción de un conjunto de objetos enfocando su atención en el orden cronológico del envío de mensajes enviados entre estos. Este diagrama permite dar detalle a los casos de uso y depende de la formulación de dichos casos de uso.

- **Objetos o Actores**
- **Líneas de vida.**-Indica la vida de un objeto durante la interacción, es una línea vertical.
- **Focos de Control.**-Muestra el periodo de tiempo en el cuál el objeto se encuentra desarrollado en alguna operación.
- **Mensajes.**-Este se realiza entre objetos y se representa con la línea horizontal con la flecha pueden ser asíncrono de llamada a procedimiento o concurrentes.

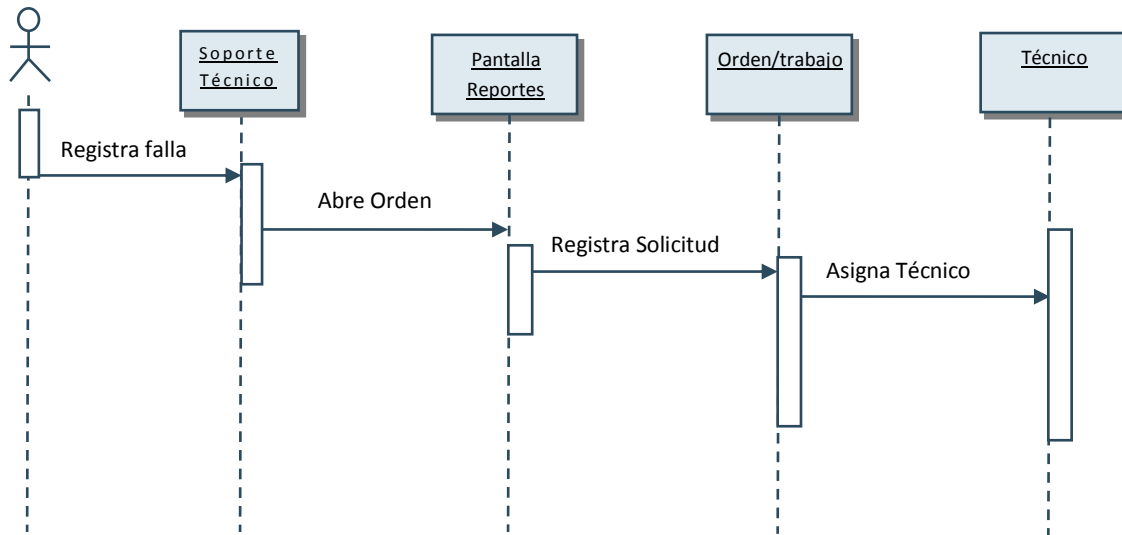


2.7 Diagrama de Gestión de Inventario

En la **Figura 2.7** se puede visualizar el flujo temporal del caso de uso iniciar sesión además de los posibles accesos o servicios.

REGISTRA REPORTE DE FALLA

Responsable
Informática

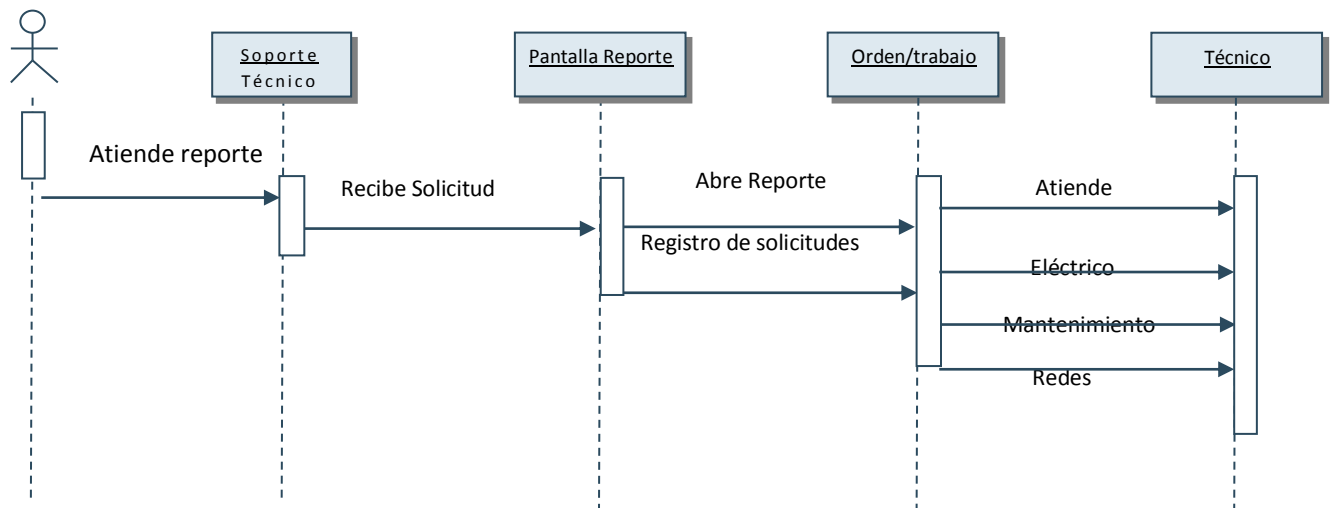


2.8 Diagrama de Reporte de Fallas

El encargado de soporte técnico inicia sesión correctamente y puede continuar con las opciones, registrar una nueva orden como se muestra en la **Figura 2.8**

ATENDER REPORTE

Encargado
Informática



2.9 Diagrama Abrir, Atender Reporte

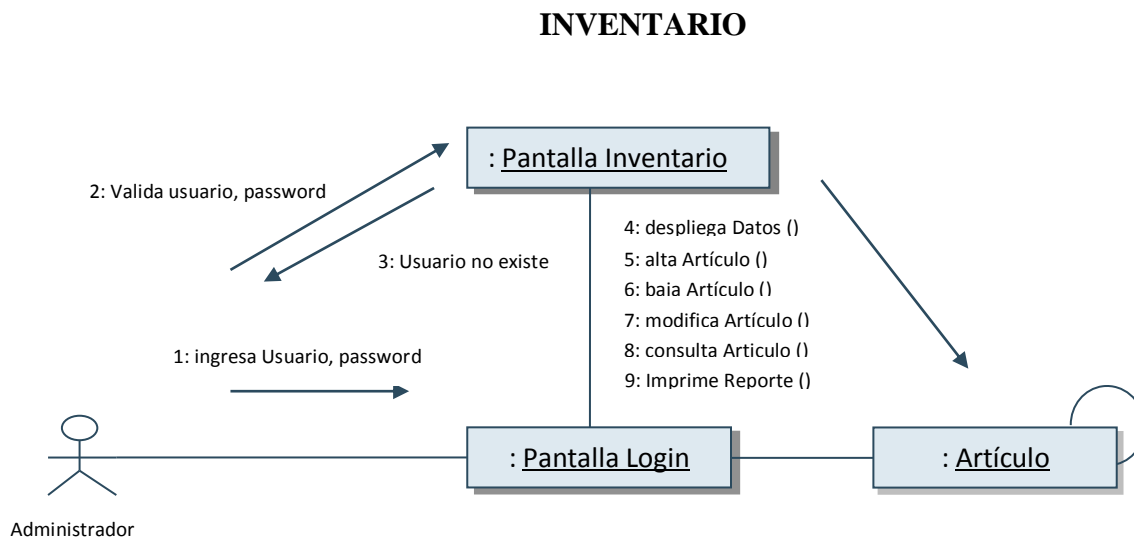
El técnico abre las solicitudes hechas por el usuario y toma la que se le ha asignado de acuerdo al tipo de falla como muestra la **Figura 2.9**

2.7.3 Diagramas de Colaboración

Muestra la colaboración entre objetos para realizar una tarea durante el uso de mensajes entre ellos enfatiza la estructura organizacional estableciendo un orden para dichos mensajes sin importar el orden cronológico, enumera los mensajes para establecer una secuencia.

El diagrama presenta los siguientes elementos:

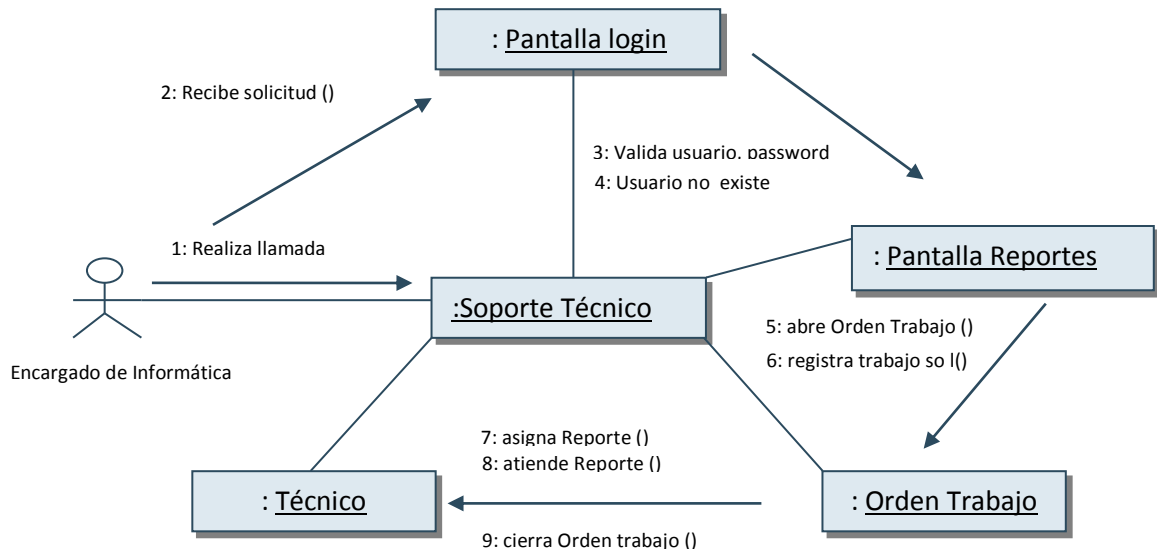
- **Objetos.-** Pueden representarse en diferentes formas como los objetos comunes, activos, multiobjetos compuestos.
- **Enlaces.-** Es una instancia a una asociación de un diagrama de clases, representa la unión de dos instancias indicando alguna forma de navegación y visibilidad.
- **Numeraciones.-** Determina la secuencia en que ocurren estos mensajes representado mediante un número.
- **Mensajes.-** Pueden ser al mismo objeto, indicar la construcción y destrucción de objetos, condicionales o excluyentes.
- **Interacciones.-** Indica la cantidad de repeticiones que ejecutará el mensaje.



2.10 Diagrama Gestión de Inventario

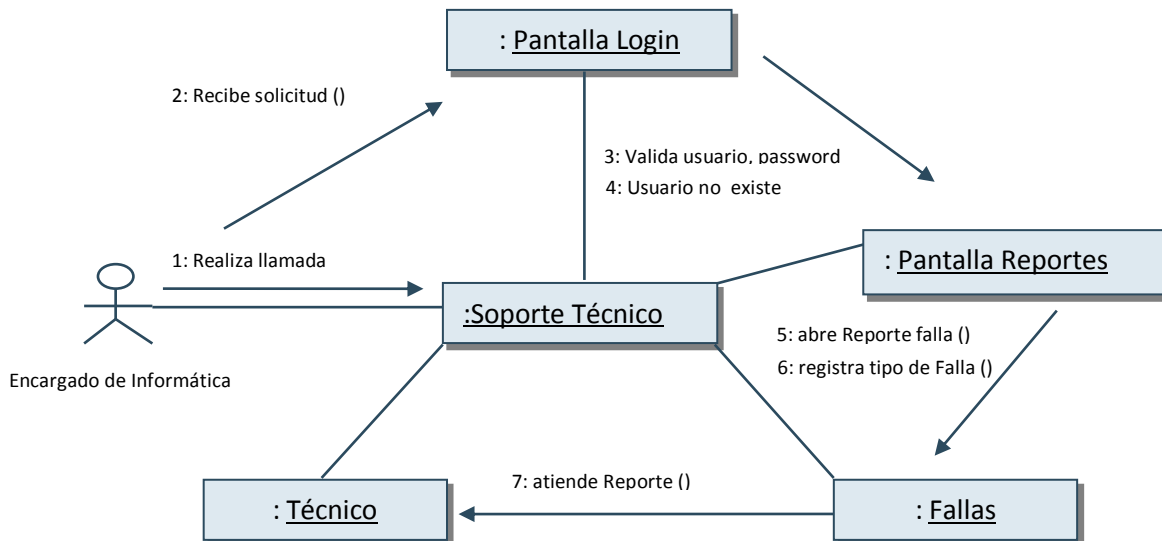
El **diagrama 2.10** muestra las diferentes opciones que tiene el administrador para poder gestionar el inventario.

REPORTE DE FALLAS



2.11 Diagrama Asigna Reporte

ATENDER REPORTE



2.12 Diagrama de Abrir, Atender Reporte

Capítulo 3: Diseño de la Base de Datos, implementación y pruebas del sistema

3.1. DISEÑO CONCEPTUAL DE LA BASE DE DATOS

El modelo o Diseño Conceptual es la primera fase del desarrollo de la BD. Se subdivide en 2 etapas: Análisis de Requisitos y generación del esquema conceptual. Hay tres tipos de diseño en el proceso de modelamiento de datos: Modelo Conceptual, Modelo Lógico y Modelo Físico, en la **Figura 3.13** se aprecia el Modelamiento de los Datos.

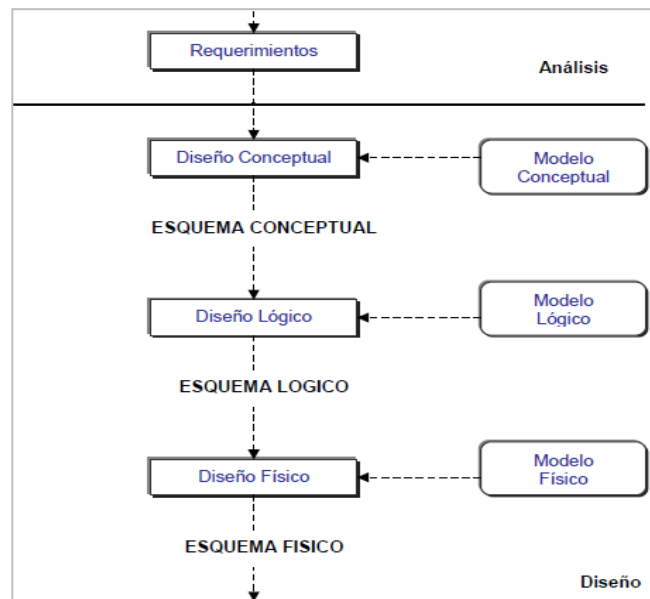


Figura 3.13 Proceso de Diseño de una Base de Datos

Un modelo conceptual es un lenguaje que se utiliza para describir esquemas conceptuales. El objetivo del diseño conceptual es describir el contenido de información de la base de datos y no las estructuras de almacenamiento que se necesitarán para manejar esta información.

El modelo conceptual de datos que se va a utilizar es el modelo Entidad-Relación (ER) su propósito es simplificar el diseño de bases de datos a partir de descripciones textuales de los requerimientos.

Pasos para la construcción del Modelo Conceptual:

- Identificar los tipos de Entidad
- Identificar los tipos de Relación
- Determinar los dominios de los atributos
- Crear Modelo ER.
- Identificar clave principal, candidata y alternativa.

En la **Figura 3.14** E-R están contenidas las siguientes entidades: Artículos, Marca, Modelo, Responsable, Orden de trabajo, Bitácora, Técnico

3.1.1 Modelo Entidad – Relación

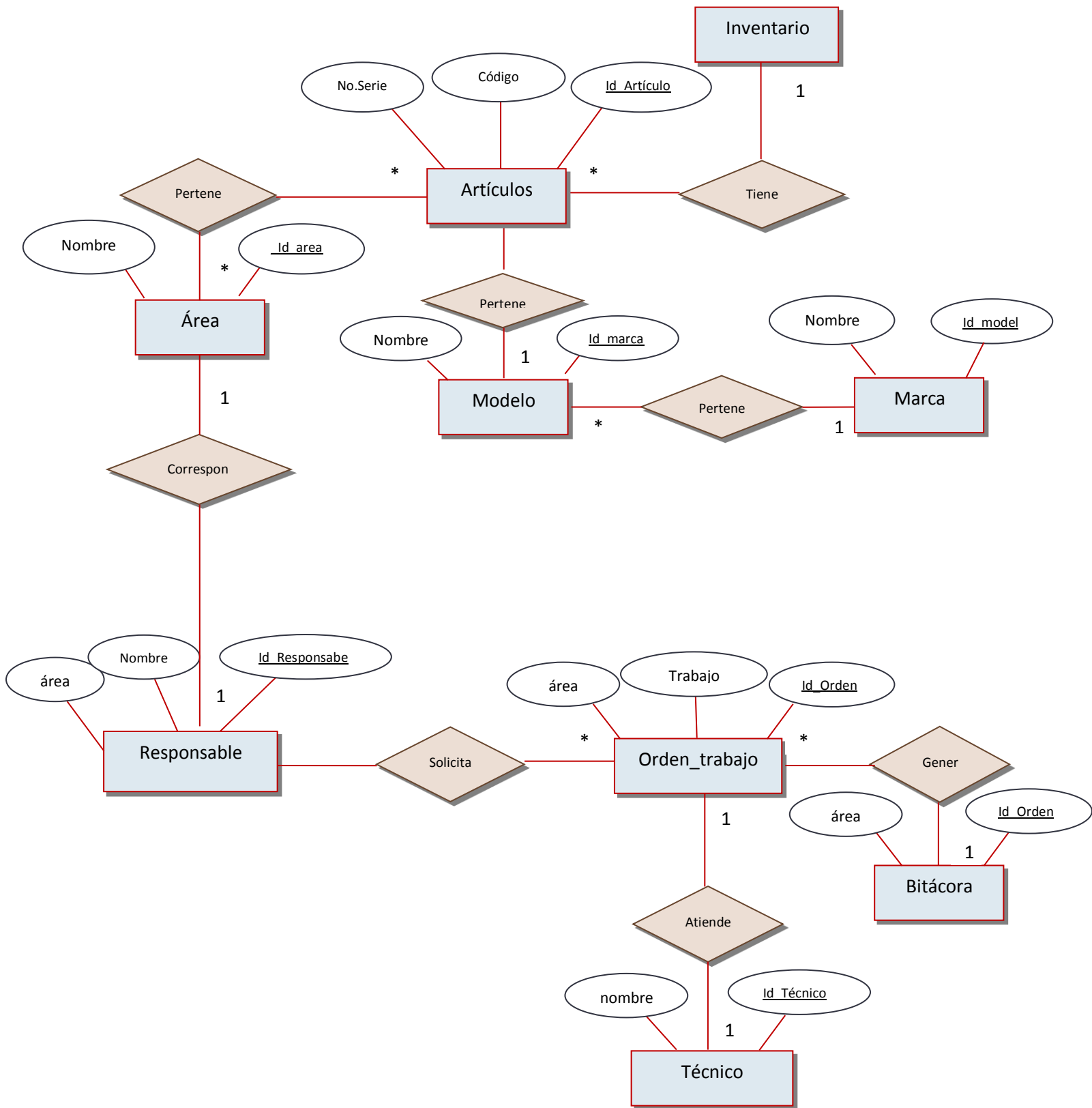


Figura 3.14 Modelo Entidad Relación de Clases

3.1.2 Descripción de Entidades y Relaciones

Las tablas que forman el modelo entidad relación muestran las llaves primarias y foráneas las cuales denotamos con **PK** y **FK**.

MARCA	
<u>Id marca</u> PK	nombre

Tabla 3.3 Marca

MODELO	
<u>Id modelo</u> PK	nombre

Tabla 3.4 Modelo

ÁREA	
<u>Id área</u> PK	nombre

Tabla 3.5 Área

RESPONSABLE				
<u>Id artículo</u> PK	<u>área</u> FK	nombre	paterno	materno

Tabla 3.6 Responsable

TÉCNICO				
<u>Id artículo</u> PK	<u>área</u> FK	nombre	paterno	materno

Tabla 3.7 Técnico

ARTICULOS									
<u>Id artículo</u>	<u>marca</u>	<u>modelo</u>	<u>área</u>	código	no_serie	capacidad	estado	no_equipo	fecha_com
PK	FK	FK	FK						

Tabla 3.8 Artículos

ORDEN							
<u>No_orden</u>	<u>área</u>	<u>técnico</u>	no_equipo	estado	Fecha_sol	Fecha_fin	comentarios
PK	FK	FK					

Tabla 3.9 Orden

3.1.3 Diseño Lógico

A partir del esquema conceptual y teniendo en cuenta los requisitos de proceso y de entorno, se elabora el esquema lógico que se apoya en el modelo relacional el cual es el que soportará el SGBD.

El modelo relacional es el más utilizado para interpretar el modelo ER y traducirlo a una BD, donde la idea principal es el uso de las relaciones.

La manera de migrar un modelo ER a un relacional puede ser interpretada como una o más tablas estructuradas en registros y campos que se pueden vincular entre sí por un campo en común.

Las ventajas de utilizar el Modelo Relacional son:

- 1.-Garantizar las herramientas para evitar la duplicidad de registros a través de campos claves o llaves
- 2.-Garantizar la integridad referencial, así al eliminar un registro, elimina todos los registros dependientes.
- 3.-Favorece la normalización por ser más comprensible y aplicable.

Existen reglas básicas para convertir un esquema en el modelo ER al relacional y son las siguientes:

- 1.-Cada una de las entidades será el nombre de la tabla.
- 2.-Cada uno de los atributos de las entidades, forman parte de los campos que componen la tabla.
- 3.-En caso en que las relaciones contengan una cardinalidad M: N, entonces esta relación se convierte en una tabla que contiene las llaves de cada una de las entidades.
- 4.-En caso en que las relaciones tengan una cardinalidad 1: M, entonces la llave de la entidad de la izquierda pasa a formar parte de los atributos de la entidad de la derecha hay hacemos la conversión a tablas.
- 5.-En caso en que las relaciones tengan una cardinalidad M: 1, entonces la llave de la entidad de la derecha pasa a formar parte de los atributos de la entidad de la izquierda, y hacemos la conversión a tablas.

6.-Normalización.

3.1.3.1 Normalización

La Normalización es una serie de reglas a las relaciones obtenidas tras el paso del Modelo Entidad Relación al Modelo Relacional, este proceso es necesario para:

- Evitar la redundancia de los datos.
- Evitar problemas de actualización de los datos en las tablas.
- Proteger la integridad de los datos.

La migración del Modelo ER al Modelo Relacional, implica un filtrado que se realiza con las formas Normales.

Primera Forma Normal

Las tablas se encuentran en primera forma Normal si cumple con las siguientes características:

- Todos los atributos son atómicos.
- Un atributo es atómico si los elementos del dominio son indivisibles, mínimos.
- La tabla contiene una llave primaria única. La llave primaria no contiene atributos nulos.
- No debe existir variación en el número de columnas.
- Los campos no llave deben identificarse por la llave (Dependencia Funcional).
- Debe Existir una independencia del orden tanto de las filas como de las columnas, es decir, si los datos cambian de orden no deben cambiar sus significados.

Con ello se eliminan la posible duplicidad de los datos poniendo en orden y forma el modelo relacional.

Segunda forma Normal.

Las tablas se encuentran en segunda forma Normal si cumple con las siguientes características:

- Si las tablas cumplen la primera forma Normal.
- Todos los atributos que no son clave principal, deben depender únicamente de la clave principal, en otras palabras, no deben existir dependencias parciales.

Tercera forma Normal.

Las tablas se encuentran en tercera forma normal si cumple con las siguientes características:

- Si las tablas cumplen la primera forma Normal y la segunda forma Normal.
- Si no existe ninguna dependencia transitiva funcional entre los atributos que no son clave.

Tomando en cuenta las definiciones anteriores justificaremos la normalización las tablas de la base de datos.

Tabla Marca se encuentra en primera forma normal, puesto que no se repite algún atributo dentro de la tabla, como lo indica la primera forma normal, cada elemento es atómico. Por otra parte, tampoco tiene múltiples valores. Con esta justificación, se da por hecho que la tabla está en primera forma normal (1FN). Si la clave principal está formada por un solo atributo y está en (1FN) automáticamente estaría en (2FN) [2]. **Ver Tabla 3.10**

MARCA	
<u>Id_marca</u> PK	nombre

Tabla 3.10 Muestra la tabla Marca normalizada

Tabla Marca está en primera forma ya que el valor que contiene el atributo es único, como lo indica la primera forma normal, cada elemento es atómico. Con esta justificación, se da por hecho que la tabla está en primera forma normal (1FN). Si la clave principal está formada por un solo atributo y está en (1FN) automáticamente estaría en (2FN) **Ver Tabla 3.11**

MODELO	
<u>Id_modelo</u> PK	Nombre

Tabla 3.11 Muestra la tabla Modelo normalizada

Tabla Marca está en primera forma ya que el valor que contiene el atributo es único, como lo indica la primera forma normal, cada elemento es atómico. Con esta justificación, se da por hecho que la tabla está en primera forma normal (1FN). Si la clave principal está formada por un solo atributo y está en (1FN) automáticamente estaría en (2FN) **Ver Tabla 3.12**

AREA	
<u>Id_área</u> PK	nombre

Tabla 3.12 Muestra la tabla Área normalizada

Tabla Responsable tiene los atributos nombre, paterno, materno los cuales se encuentran en (1FN) ya que el valor que contienen sus atributos es único, la tabla se encuentra en (2FN) ya que los atributos dependen de la llave primaria, la llave foránea id_área depende de la llave primaria de la tabla área. La tabla no cuenta con dependencias transitivas por lo que la tabla responsable se encuentra normalizada hasta la (3FN) **Ver Tabla 3.13**

RESPONSABLE				
<u>Id responsable</u>	<u>Id área</u>	nombre	paterno	materno
PK	FK			

Tabla 3.13 Muestra la tabla Responsable normalizada

Tabla Técnico tiene los atributos nombre, paterno, materno los cuales se encuentran en (1FN) ya que el valor que contienen sus atributos es único, la tabla se encuentra en (2FN) ya que los atributos dependen de la llave primaria, la llave foránea id_área depende de la llave primaria de la tabla área. La tabla no cuenta con dependencias transitivas por lo que la tabla responsable se encuentra normalizada hasta la (3FN) **Ver Tabla 3.14**

TÉCNICO				
<u>Id técnico</u>	<u>Id área</u>	nombre	paterno	materno
PK	FK			

Tabla 3.14 Muestra la tabla Técnico normalizada

Tabla Artículos está en la primera forma normal, puesto que no se repite algún atributo dentro de ella, como la forma normal lo dice, cada elemento es atómico. Con esta justificación, se da por hecho que la tabla está en primera forma normal (1FN).

Analizando la tabla también está en (2FN) ya que, además de estar en primera forma normal, todos los atributos dependen de la llave primaria, la llave foránea id_marca está relacionada con la llave primaria de la tabla marca, Id_modelo está relacionada con la llave primaria de la tabla modelo, Id_área está relacionada con la llave primaria de la tabla área.

Se encuentra en (3FN) por qué no existen dependencias transitivas todos los atributos no llave dependen de manera no transitiva de la llave primaria, no dependen de ninguna otra columna que no sea llave. **Ver Tabla 3.15**

ARTICULOS									
<u>Id artículo</u>	<u>Id marca</u>	<u>Id modelo</u>	<u>Id área</u>	código	no_serie	capacidad	estado	fecha_com	no_equipo
PK	FK	FK	FK						

Tabla 3.15 Muestra la tabla Artículos normalizada

Tabla Orden está en la primera forma normal, puesto que los atributos contienen valores atómicos (1FN).

La tabla se encuentra en (2FN), ya que, además de estar en primera forma normal, todos los atributos dependen de la llave primaria, la llave foránea id_responsable está relacionada con la llave primaria de la tabla responsable, Id_técnico está relacionada con la llave primaria de la tabla técnico.

Se encuentra en (3FN) por qué no existen dependencias transitivas todos los atributos no llave dependen de manera no transitiva de la llave primaria, no dependen de ninguna otra columna que no sea llave. **Ver Tabla 3.16**

Orden							
<u>Id orden</u>	<u>Id responsable</u>	<u>Id técnico</u>	trabajo_sol	estado	fecha_sol	Fecha_fin	no_equipo
PK	FK	FK					

Tabla 3.16 Muestra la tabla Orden normalizada

3.1.4 IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA

Se presentan las herramientas que facilitaron el desarrollo del sistema así como la implementación de las interfaces, estas herramientas son: PHP, MySQL y PhpMyAdmin para instalar un servidor local, y administrar bases de datos, estas herramientas son gratuitas y ofrecen opciones para desarrollar aplicaciones web y base de datos. [3]

MySQL

MySQL es un sistema de gestión de base de datos relacional, con licencia GLP de GNU. Su diseño multihilo le permite soportar gran carga de forma muy eficiente. Este gestor de bases de datos es uno de los más usados en Internet, debido a su gran rapidez y facilidad de uso [3]

Características de MySQL

Las principales características de este gestor de bases de datos son las siguientes:

1. Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
2. Gran portabilidad entre sistemas.
3. Soporta hasta 32 índices por tabla.
4. Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.

PHP

Funcionalidades primordiales en PHP se consideran:

- Gestión de bases de datos: El lenguaje PHP ofrece interfaces para el acceso a la mayoría de las bases de datos posibles en sistemas Microsoft. Gestión de archivos: mediante operaciones de creación, borrado, modificación, además de ofrecer transferencia de archivos.
- Tratamiento de imágenes: mediante funciones de automatización de formato, envío de lotes de imágenes y funciones de graficado.

Ventajas de PHP

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan
- en la actualidad, destaca su conectividad con MySQL
- Leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML.
- Permite las técnicas de Programación Orientada a Objetos. [7]

3.1.4.1 Modelo Lógico

La implementación de la base de datos se realizó con PhpMyAdmin el cual es un servidor independiente de la plataforma, software libre, que consiste principalmente en la BD MySQL, el servidor Web Apache y los intérpretes para los lenguajes de Script: Php y Perl. Actualmente WAMP SERVER 2.0 se encuentra disponible para Windows de manera gratuita en internet.

PhpMyAdmin facilita la creación de la base de datos con sus tablas correspondientes. Ver **Figura 3.15**

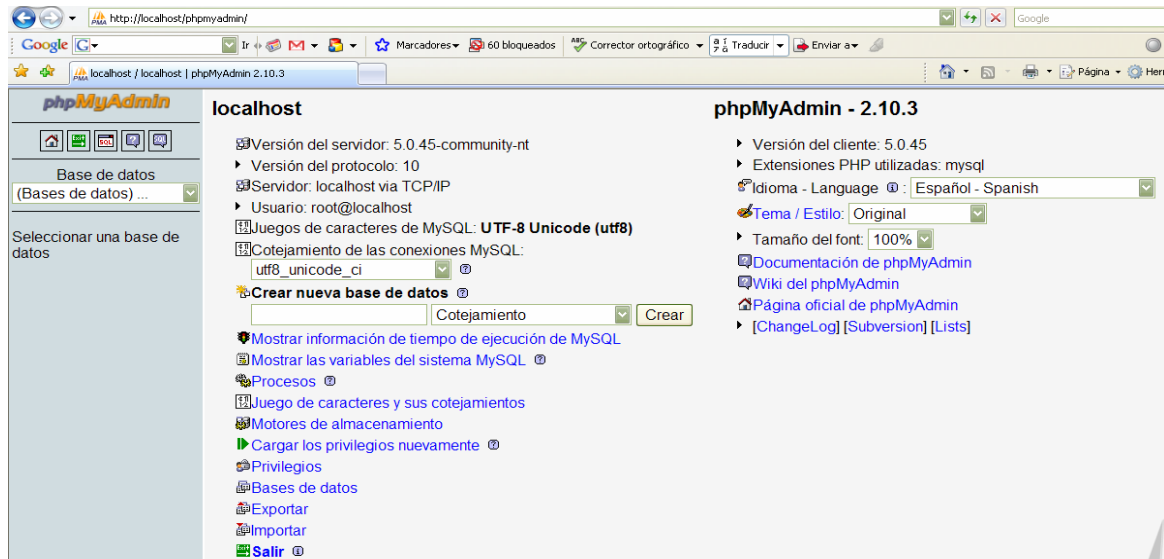


Figura 3.15 Interfaz de PhpMyAdmin

Una vez creada nuestra base de datos, continuamos con la creación de las tablas en las cuales vamos a almacenar la información correspondiente para cada entidad que se requiera. Ver **Figura 3.16**

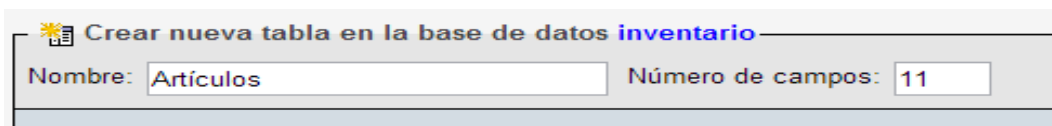


Figura 3.16 Creación de una tabla

Después de crear la tabla, se seleccionan los campos que lleva la misma, y los tipos de valor que llevara cada campo, es decir se dan las propiedades que la tabla requiere, una vez hecho esto el programa muestra un reporte de los valores que tiene cada tabla.

Artículos

Mantiene el registro de las piezas que componen un equipo de cómputo. **Ver Tabla 3.17**

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<input type="checkbox"/>	<u>id_articulo</u>	int(30)			No	None	auto_increment
<input type="checkbox"/>	codigo	int(15)			No	None	
<input type="checkbox"/>	no_serie	int(20)			No	None	
<input type="checkbox"/>	capacidad	text	latin1_swedish_ci		No	None	
<input type="checkbox"/>	estado	text	latin1_swedish_ci		No	None	
<input type="checkbox"/>	fecha_comp	date			No	None	
<input type="checkbox"/>	no_equipo	int(5)			No	None	
<input type="checkbox"/>	id_marca	int(11)			No	None	
<input type="checkbox"/>	id_modelo	int(11)			No	None	
<input type="checkbox"/>	id_area	int(11)			No	None	

Tabla 3.17 Registro de Artículos

Marca

La **Tabla 3.18** muestra el registro de nombres de marcas disponibles en los laboratorios de cómputo.

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<input type="checkbox"/>	<u>id_marca</u>	int(5)			No	None	auto_increment
<input type="checkbox"/>	nombre	text	latin1_swedish_ci		No	None	

Tabla 3.18 Registro Marcas de Equipos de Cómputo

Modelo

La **Tabla 3.19** muestra el registro de nombres de marcas disponibles en los laboratorios de cómputo.

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<input type="checkbox"/>	<u>id_modelo</u>	int(11)			No	None	auto_increment
<input type="checkbox"/>	nombre	text	latin1_swedish_ci		No	None	

Tabla 3.19 Registro Modelos de Equipos de Cómputo

Área

Mantiene el registro de las diferentes areas que existen dentro del plantel. Ver **Tabla 3.20**

Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<u>id_area</u>	int(11)			No	None	auto_increment
nombre_area	varchar(30)	latin1_swedish_ci		No	None	

Tabla 3.20 Registro de Áreas

Orden

La **Tabla 3.21** muestra las solicitudes de órdenes de mantenimiento realizadas por los responsables de los laboratorios de cómputo del plantel al departamento de Soporte Técnico.

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<input type="checkbox"/>	<u>id_orden</u>	int(5)			No	None	auto_increment
<input type="checkbox"/>	trabajo_sol	text	latin1_swedish_ci		No	None	
<input type="checkbox"/>	fecha_sol	date			No	None	
<input type="checkbox"/>	fecha_fin	date			No	None	
<input type="checkbox"/>	estado	text	latin1_swedish_ci		No	None	
<input type="checkbox"/>	no_equipo	int(5)			No	None	
<input type="checkbox"/>	id_responsable	int(5)			No	None	
<input type="checkbox"/>	id_tecnico	int(5)			No	None	

Tabla 3.21 Registro de Ordenes de Mantenimiento

Responsable

Mantiene el registro de los responsables de cada área en el plantel como se puede mostrar en la **Tabla 3.22**

Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<u>id_responsable</u>	int(11)			No	None	auto_increment
nombre	varchar(30)	latin1_swedish_ci		No	None	
paterno	varchar(30)	latin1_swedish_ci		No	None	
materno	varchar(30)	latin1_swedish_ci		No	None	
id_area	int(11)			No	None	

Tabla 3.22 Registro de Áreas

Técnico

La **Tabla 3.23** muestra el registro de los técnicos del departamento de Soporte Técnico

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<input type="checkbox"/>	id_tecnico	int(5)			No	None	auto_increment
<input type="checkbox"/>	nombre	text	latin1_swedish_ci		No	None	
<input type="checkbox"/>	paterno	text	latin1_swedish_ci		No	None	
<input type="checkbox"/>	materno	text	latin1_swedish_ci		No	None	
<input type="checkbox"/>	id_area	int(5)			No	None	

Tabla 3.23 Registro de los Técnicos

BACKUP

La **Tabla 3.24** muestra la tabla el Backup de los artículos que se han dado de baja.

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<input type="checkbox"/>	id_articulo	int(30)			No	None	auto_increment
<input type="checkbox"/>	codigo	int(15)			No	None	
<input type="checkbox"/>	no_serie	int(20)			No	None	
<input type="checkbox"/>	capacidad	text	latin1_swedish_ci		No	None	
<input type="checkbox"/>	estado	text	latin1_swedish_ci		No	None	
<input type="checkbox"/>	fecha_comp	date			No	None	
<input type="checkbox"/>	no_equipo	int(5)			No	None	
<input type="checkbox"/>	Id_marca	int(11)			No	None	
<input type="checkbox"/>	Id_modelo	int(11)			No	None	
<input type="checkbox"/>	Id_area	int(11)			No	None	

Tabla 3.24 Backup de Artículos

3.1.5 IMPLEMENTACIÓN DE LA INTERFAZ

Una vez creada la base de datos con sus respectivas tablas se requirió conectar nuestra base de datos MySQL con el servidor local, se utilizó Dreamweaver para establecer la conexión y crear el sitio. **Ver Figura 3.17**

Ventajas de Dreamweaver:

- Dreamweaver es una herramienta de diseño y edición de sitios

- Herramientas de creación de contenido HTML
- Permite la conexión a Bases de Datos como MySQL y Access.
- Filtrar y mostrar el contenido utilizando tecnología de Script como: ASP, JSP y PHP entre otras.

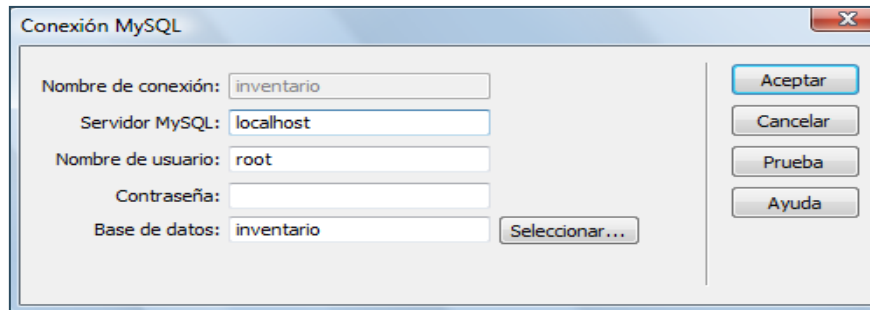


Figura 3.17 Conexión de la Base de Datos con Dreamweaver

3.1.5.1 Interfaz de Inicio de Sesión

En la **Figura 3.18** muestra la pantalla principal donde el usuario podrá ingresar de acuerdo al rol que desempeña en el plantel para poder acceder al sistema.

Se lleva a cabo la autenticación de sesiones las cuales son Administrador y Usuario.

El Administrador tiene los privilegios de ingresar, modificar y eliminar usuarios y artículos mientras que el usuario solo puede consultar artículos y órdenes de mantenimiento e imprimir reportes de artículos y bitácora de mantenimiento.



Figura 3.18 Interfaz de inicio de sesión

Una medida de seguridad del sistema es el control de acceso, se utilizaron mecanismos a través de los cuales se intenta asegurar que solamente las personas autorizadas pueden acceder a los recursos del sistema de información.

Con los siguientes privilegios.

- 1.- Leer escribir, ejecutar
- 2.-Seleccionar, insertar, actualizar
- 3.-Referenciar, indexar

Estas autorizaciones definen que accesos podrán realizar los usuarios sobre los objetos y que otros no.

PhpMyAdmin permite los permisos administrativos (Implementadas por primitivas Grant, Revoke, Own) que permiten la modificación de autorizaciones. **Ver Fig. 3.19**

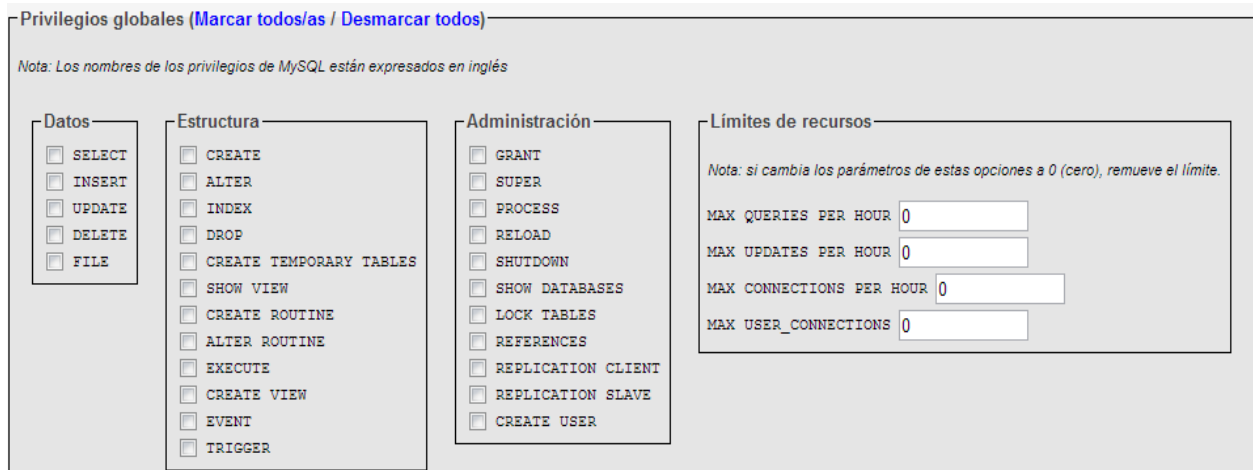


Figura 3.19 Permisos administrativos que permite PhpMyAdmin

Otra medida de seguridad para proteger datos y contraseñas de los usuarios es la encriptación de datos a través de algoritmos. Los datos se codifican a través de algún algoritmo. Un usuario no autorizado tendrá problemas para descifrar los datos codificados. Las funciones que se utilizaron para mejorar la seguridad del sistema fueron las funciones HASH y la función MD5. **Ver Figura 3.20**

Función Hash.-Una función Hash es un método para generar claves o llaves que representen de manera unívoca a un documento o conjunto de datos los más conocidos son MD5 y SHA-1.

Es una función matemática compleja unidireccional que aplicada a un conjunto de caracteres de cualquier longitud obtiene un pequeño resultado, de largo fijo, y cualquier cambio al conjunto de caracteres origen, producirá un cambio en el resultado.

Mediante estas funciones puedo controlar que mi información no haya sido alterada, o por error o intencionalmente durante la transmisión o durante el periodo que permaneció almacenada.

Función MD5.- En Php se utiliza la función MD5 que es una función Hash irreversible (de un solo sentido) es decir se encripta el password tecleado por el usuario y es imposible que partiendo de la cadena encriptada se vuelva a la contraseña origen.

```
<html><head><title></title>

<script language="JavaScript" src="file:///C:/wamp/logs/md5.js"></script>
<script language="JavaScript">
numero = Math.random().toString();

function calculaMD5() {
var pw = document.forms["login"].elements["password"].value
pw += numero
return calculaMD5(pw)
}

function enviaMD5(hash) {
document.forms["login"].elements["cifrado"].value = hash;
document.forms["login"].elements["numero"].value = numero;
document.forms["login"].submit();
}
</script>
</head>

<body>
<form action="file:///C:/wamp/logs/auth.php" method="POST" name="login">
Usuario: <input type="Text" name="usuario"><br>
Password: <input type="Password" name="password"><br>
<input type="Hidden" name="cifrado" value="">
<input type="Hidden" name="numero" value="">
<input type="Submit" value=" Login " onClick="enviaMD5(calculaMD5())">
</form>
```

Figura 3.20 Función MD5 aplicada a la BD Inventario

3.2 INTERFAZ DEL ADMINISTRADOR

La **Figura 3.21** muestra como el administrador puede seleccionar entre los diferentes módulos del sistema.

Administrador.-Permite ingresar a nuevos usuarios o administradores así como también se podrá eliminar o modificar información registrada previamente.

Inventario.-El sistema permitirá al administrador dar de alta nuevos artículos o bien modificar o eliminar artículos ya registrados. El sistema también permitirá consultar el estado de los artículos y la fecha de ingreso la cual es indispensable para la toma de decisiones cuando se lleve a cabo la actualización de los equipos en los laboratorios de cómputo.

Artículos dados de baja.-El sistema almacenará los datos de los artículos que se han dado de baja para crear una bitácora de artículos que de acuerdo a su estado “funcionando” podrán reubicarse en otra área



Figura 3.21 Interfaz Inventario.- Administrador

3.3 INTERFAZ DE USUARIO

La **Figura 3.22** muestra las áreas Primaria, Secundaria, Preparatoria, a las que el usuario puede ingresar para realizar consultas del inventario así como también puede imprimir reportes.

El sistema permite al usuario ingresar al módulo Órdenes de Mantenimiento para consultar el estado de las solicitudes (Espera, Atendidas, o Finalizadas). El sistema permite al usuario interactuar con los dos módulos de manera conjunta “Inventario” y “órdenes de Mantenimiento” ya que los artículos registrados en el inventario son los mismos a los que se les proporcionara el servicio de soporte técnico.



Figura 3.22 Interfaz inventario.-Usuario

3.4 PRUEBAS DEL SISTEMA

El sistema ha sido probado e implantado en varios equipos con diferentes plataformas de manera local, y desde un Hosting gratuito los usuarios han accedido al sistema.

3.5 ÁREAS

La interfaz permite al Administrador seleccionar de las diferentes áreas registrar nuevos artículos, modificar o eliminar artículos registrados previamente. También permite al Administrador realizar consultas o imprimir reportes de artículos registrados. Como muestra la **Figura 3.23**



Figura 3.23 Interfaz Inventario.- Áreas

3.6 ALTA DE ARTÍCULOS

La **Figura 3.24** muestra la interfaz que permite registrar los artículos y una ventana emergente cuando los datos han sido registrados correctamente.



Figura 3.24 Interfaz Inventario.-Registro de Artículos

3.7 ARTÍCULOS REGISTRADOS

La **Figura 3.25** muestra la interfaz de los artículos registrados y los módulos que le permitirán al administrador realizar las consultas o bien realizar una vista preliminar antes de imprimir un reporte.



Figura 3.25 Interfaz inventario.-Artículos registrados

3.8 CONSULTA ARTÍCULOS

La **Figura 3.26** muestra la interfaz que permite al Administrador realizar consultas por “Fecha de Compra” o “estado”

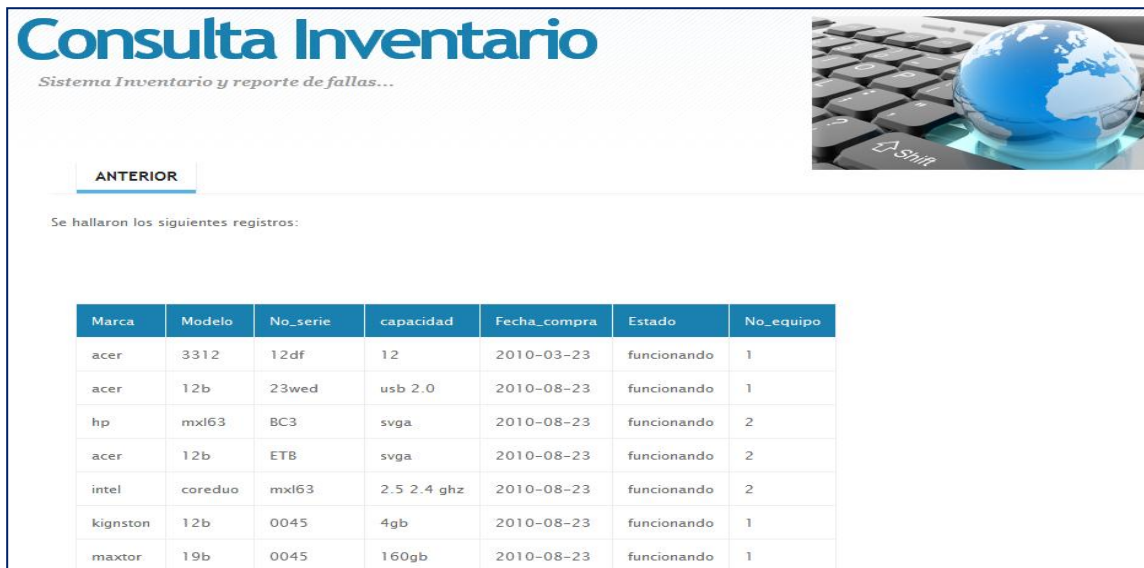


The screenshot shows the 'Consulta Inventario' web interface. At the top left, the title 'Consulta Inventario' is displayed in a large blue font, with the subtitle 'Sistema Inventario y reporte de fallas...' below it. To the right is a decorative image of a blue globe on a keyboard. Below the title is a navigation link labeled 'ANTERIOR'. The main content area contains two search forms. The first form, titled 'Formulario de búsqueda', has a 'Fecha Compra' field with the value '2010' and a 'Buscar' button. The second form has an 'Estado' field with the value 'funcionando' and a 'Buscar' button.

Figura 3.26 Interfaz Inventario.- Consultar Artículo

3.9 DATOS CONSULTADOS

La **Figura 3.27** muestra la interfaz de los datos encontrados de la consulta realizada.



The screenshot shows the 'Consulta Inventario' web interface displaying search results. At the top left, the title 'Consulta Inventario' is displayed in a large blue font, with the subtitle 'Sistema Inventario y reporte de fallas...' below it. To the right is a decorative image of a blue globe on a keyboard. Below the title is a navigation link labeled 'ANTERIOR'. Below the navigation link, the text 'Se hallaron los siguientes registros:' is displayed. Below this text is a table with 7 columns and 8 rows. The columns are: Marca, Modelo, No_serie, capacidad, Fecha_compra, Estado, and No_equipo. The rows contain the following data:

Marca	Modelo	No_serie	capacidad	Fecha_compra	Estado	No_equipo
acer	3312	12df	12	2010-03-23	funcionando	1
acer	12b	23wed	usb 2.0	2010-08-23	funcionando	1
hp	mxl63	BC3	svga	2010-08-23	funcionando	2
acer	12b	ET8	svga	2010-08-23	funcionando	2
intel	coreduo	mxl63	2.5 2.4 ghz	2010-08-23	funcionando	2
kignston	12b	0045	4gb	2010-08-23	funcionando	1
maxtor	19b	0045	160gb	2010-08-23	funcionando	1

Figura 3.27 Interfaz Inventario.- Resultado Consulta

3.10 IMPRIMIR BITÁCORA

La **Figura 3.28** muestra la vista preliminar para continuar con la impresión de la bitácora del inventario.

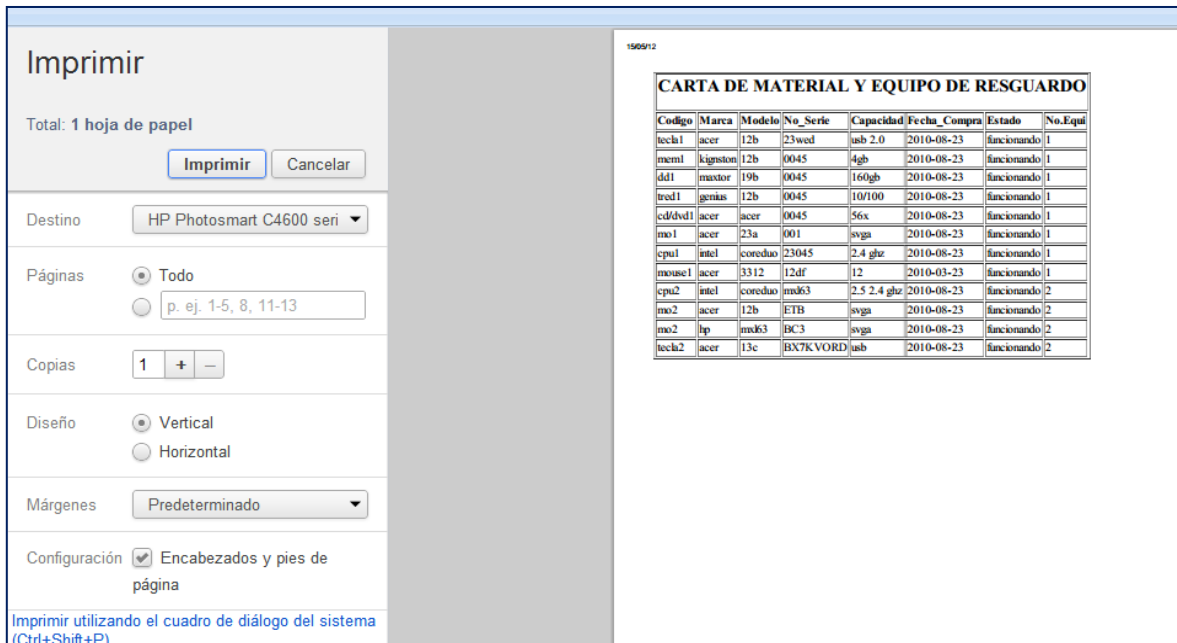


Figura 3.28 Interfaz Inventario.- Vista Preliminar Bitácora

3.11 BACKUP DE ARTÍCULOS

La **Figura 3.29** muestra la interfaz de los artículos que han sido dados de baja y que aún funcionan para poder ser canalizados a otras áreas.



Figura 3.29 Interfaz Inventario.- Artículos dados de baja

3.12 ÓRDENES DE MANTENIMIENTO

La **Figura 3.30** muestra la interfaz principal del módulo Órdenes de Mantenimiento que permite registrar solicitudes de mantenimiento, cambiar el estado de las órdenes (Espera, Atendido o Finalizado) permite consultar los estados de las órdenes e imprimir la bitácora de las órdenes registradas.



Figura 3.30 Interfaz Orden de Mantenimiento

3.13 ABRIR REPORTE DE FALLA

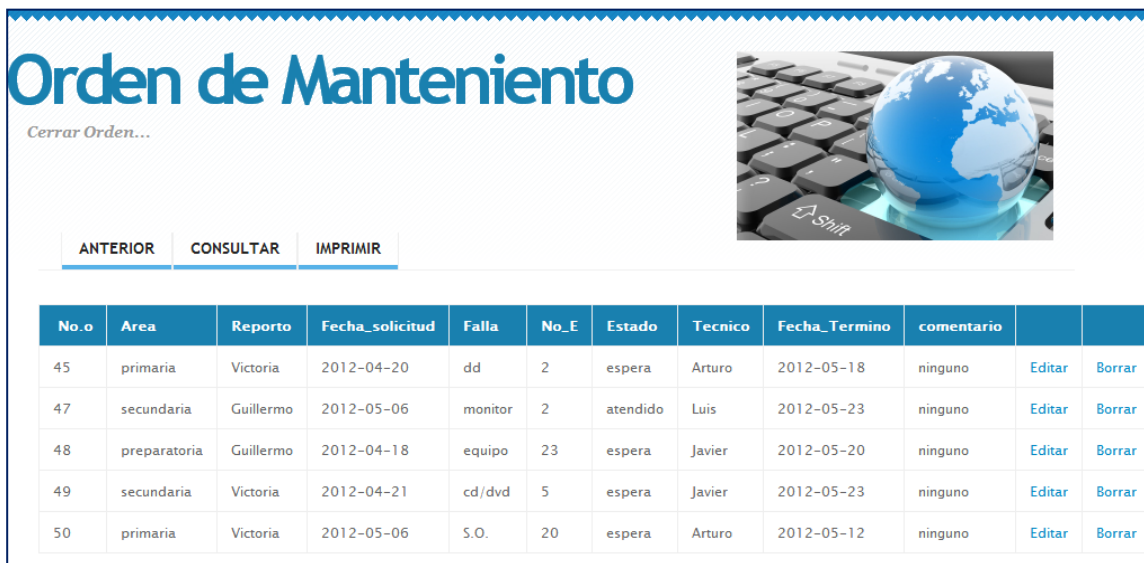
En la **Figura 3.31** muestra la interfaz Abrir Orden que permite registrar las solicitudes de mantenimiento hechas por el usuario y se colocarán en Espera o Atendido dependiendo de las solicitudes que esté atendiendo cada técnico, el administrador deberá hacer uso de la bitácora para saber que técnico está disponible y asignarle un nuevo mantenimiento.

Area:	Primaria
Reporto:	Marcela
Fecha_sol:	2012-05-20
Trabajo_sol:	revisión disco duro
No. Equipo:	3
Estado:	Atendido
Tecnico:	Javier
Fecha_Fin:	2012-05-25
Comentarios:	ninguno

Figura 3.31 Interfaz Orden de Mantenimiento.-Registrar Orden

3.14 ÓRDENES REGISTRADAS

En la **Figura 3.32** muestra la interfaz Cerrar orden que permite modificar el estado de las solicitudes de Atendido a Finalizado permitiendo al administrador concluir la solicitud. La interfaz permitirá eliminar órdenes registradas así como también consultar e imprimir órdenes de mantenimiento.



The screenshot displays the 'Orden de Mantenimiento' interface. At the top, there is a header with the title 'Orden de Mantenimiento' and a sub-header 'Cerrar Orden...'. Below the header, there are three navigation buttons: 'ANTERIOR', 'CONSULTAR', and 'IMPRIMIR'. The main content area features a table with the following data:

No.o	Area	Reporto	Fecha_solicitud	Falla	No_E	Estado	Tecnico	Fecha_Termino	comentario		
45	primaria	Victoria	2012-04-20	dd	2	espera	Arturo	2012-05-18	ninguno	Editar	Borrar
47	secundaria	Guillermo	2012-05-06	monitor	2	atendido	Luis	2012-05-23	ninguno	Editar	Borrar
48	preparatoria	Guillermo	2012-04-18	equipo	23	espera	Javier	2012-05-20	ninguno	Editar	Borrar
49	secundaria	Victoria	2012-04-21	cd/dvd	5	espera	Javier	2012-05-23	ninguno	Editar	Borrar
50	primaria	Victoria	2012-05-06	S.O.	20	espera	Arturo	2012-05-12	ninguno	Editar	Borrar

Figura 3.32 Interfaz Orden de Mantenimiento.-Órdenes Registradas

3.15 CONSULTAR ÓRDENES POR ATENDER

En la **Figura 3.33** se muestra la interfaz que permite realizar consultas por no. de orden y estado de la solicitud.



The screenshot displays the 'Bitacora' interface. At the top, there is a header with the title 'Bitacora' and a sub-header 'Reporte de Fallas ...'. Below the header, there is a navigation button 'ANTERIOR'. The main content area features a search form titled 'Formulario de búsqueda' with the following fields:

Formulario de búsqueda	
No.Orden	45
	45
	Buscar

Figura 3.33 Interfaz Orden de Mantenimiento.-Consultar Órdenes

3.16 BITÁCORA ORDEN

En la **Figura 3.34** se muestra la interfaz de la bitácora de las solicitudes de mantenimiento, donde el administrador y el usuario podrán consultar e imprimir reportes de mantenimiento.



No.o	Area	Reporto	Fecha_sol	Falla	No_E	Estado	Tecnico	Fecha_fin	Comentarios
45	primaria	Victoria	2012-04-20	dd	2	espera	Arturo	2012-05-18	ninguno
47	secundaria	Guillermo	2012-05-06	monitor	2	atendido	Luis	2012-05-23	ninguno
48	preparatoria	Guillermo	2012-04-18	equipo	23	espera	Javier	2012-05-20	ninguno
49	secundaria	Victoria	2012-04-21	cd/dvd	5	espera	Javier	2012-05-23	ninguno
50	primaria	Victoria	2012-05-06	S.O.	20	espera	Arturo	2012-05-12	ninguno

Figura 3.34 Interfaz Orden de Mantenimiento.- Bitácora

CONCLUSIONES

Aun cuando el proceso de desarrollo del sistema de inventario y reporte de fallas de equipo de cómputo no ha concluido en su totalidad, los módulos considerados como parte de este trabajo de tesis ya han sido instalados en computadoras del área de almacén de la institución.

El usuario encargado de esta área cuenta ya con el inventario actualizado permitiendo así conocer la ubicación de los equipos que han sido reubicados a otra área, o han quedado obsoletos y han sido dados de baja, el Backup de artículos ayuda al administrador a tener un respaldo de los equipos o dispositivos que se han dado de baja.

El usuario puede acceder de manera rápida y precisa a la información que desea consultar desde cualquier parte de la institución ya que esta se encuentra disponible en un servidor.

Así como también el usuario puede imprimir reportes generados por el sistema tanto de artículos que están en el inventario como artículos que se han dado de baja permitiéndole así un mejor control y toma de decisiones a la hora de adquirir nuevos equipos o dispositivos.

La gestión automática de inventarios permite a una organización tener una visión detallada de la configuración de hardware y software de los equipos de cómputo pertenecientes a cada área en tiempo real, evitando los desplazamientos hasta el escritorio del usuario.

El Control del Inventario y el registro de órdenes de Mantenimiento actualmente representan metas alcanzadas permitiendo tener un módulo que ayude a administrar sus activos o equipos instalados en cada laboratorio, mediante bitácoras de mantenimiento.

Algo importante que resaltar como resultado del desarrollo de dichos módulos, es que el proceso mismo al haberse apegado a la tecnología UML y empleado como lenguaje de programación a PHP, ha permitido que las clases diseñadas sean útiles para los módulos restantes, reduciendo en gran medida el tiempo de análisis y programación.

Así como también el uso de la metodología RUP que tuvo como objetivo crear un producto final con calidad a diferencia de otras metodologías RUP es el pionero en representar a través de diagramas todo el proceso que involucra desarrollar una aplicación.

El modelo relacional me ayudó a tener un producto de calidad ya que garantiza herramientas para evitar la duplicidad de registros así como también garantiza la integridad referencial, al eliminar un registro elimina todos los registros relacionados dependientes y favorece la normalización por ser más comprensible y aplicable.

Como trabajo futuro se plantea realizar los módulos de Almacén y Órdenes de compra que están directamente ligados a este sistema.

BIBLIOGRAFÍA

- [1] [DATE,1999] Date, C. J. An Introduction to Database Systems. 7th Ed.. Addison-Wesley Publishing Company. 1999.
- [2] [TC,1998] Connolly, Thomas. Database Systems: A Practical Approach to Design, Implementation, and Management. 2nd Ed. Addison-Wesley. 1998.
- [3] [ULLMAN, 1997] Ullman & Widom. A first course in database systems, PrenticeHall, 1997.
- [4] [AS, 1998] Silberschatz, Abraham, et al. Fundamentos de Bases de Datos. Tercera Edición. Mc GrawHill. 1998.
- [5] [HRR, 1999] Rue Rasnussen, Hanne, et al. Enterprise DBA Part 1A: Architecture and Administration. Production 1.0 August 1999.
- [6] [SUN, 20001] Sun Microsystems. Object-Oriented Application Analysis and Design for Java Technology (UML). Revision B.2, April 2001.
- [7] [NK, 1999] Kochhar, Nieena. et al. Introduction to Oracle: SQL and PL/SQL. Production 1.3 July 1999.
- [8] [DAN, 1991] Dan Remenyi Ph. D. Introducing Strategic Information System Planning. NCC Blackwell 1991.
- [9] [LANCE, 1991] Lance Elliot B. Information Systems Strategic Planning. Computer Technology Research Corp. 1991.
- [10] [HALL, 1988] Hall, Vicky J. Information System Analysis. Prentice Hall 1988.
- [11] [VARGAS, 1999] Vargas C., Marcela. Modelamiento de Datos. Conceptos Fundamentales. 1999.
- [12] [ZIKMUND, 1996] Zikmund, W; D'Amico, M. Mercadotecnia, México DF: Editorial Continental S.A. de C.V, páginas: 131-135, 1996.
- [13] [KOTTLER, 1975] Kotler, P; Cox, K. Estudios de mercadotecnia, México: Editorial Dianamercadotecnia local en internacional. 1975.
- [14] El Lenguaje Unificado de Modelado. Manual de Referencia James RumbaughIvar JacobsonGrady BoochPearson Addison Wesley.
- [15] UML gota a gota Martin Fowler Kendall Scott.Pearson Addison Wesley.