



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO
FACULTAD DE CIENCIAS DE LA INGENIERÍA

CARRERA DE INGENIERÍA EN SISTEMAS

TESIS DE GRADO

Tema:

**“SISTEMA DE CONTROL INTERNO DE HARDWARE Y SOFTWARE PARA
EQUIPOS DE CÓMPUTO DE LA UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO”**

**Previo a la obtención del Título de
Ingeniero en Sistemas**

Autor:

Guillermo Armando Campuzano Arechua

Directora:

Msc. Ing. Nancy Rodríguez Gavilanes

QUEVEDO-ECUADOR

2014

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

Yo, **Guillermo Armando Campuzano Arechua**, declaro que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Universidad Técnica Estatal de Quevedo, puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Guillermo Armando Campuzano Arechua

CERTIFICACIÓN DE LA DIRECTORA DE TESIS

La suscrita, Ing. Nancy Rodríguez Gavilanes Ms., Docente de la Universidad Técnica Estatal de Quevedo, certifica que la estudiante Guillermo Armando Campuzano Arechua, realizó la tesis de grado previo a la obtención del título de Ingeniero en Sistemas titulada **“SISTEMA DE CONTROL INTERNO DE HARDWARE Y SOFTWARE PARA EQUIPOS DE CÓMPUTO DE LA UNIVERSIDAD TECNICA ESTATAL DE QUEVEDO”**, bajo mi dirección, habiendo cumplido con las disposiciones reglamentarias establecidas para el efecto.

Ing. Nancy Rodríguez Gavilanes Ms.

DIRECTORA DE TESIS



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

FACULTAD DE CIENCIAS DE LA INGENIERÍA

CARRERA DE INGENIERÍA EN SISTEMAS

**SISTEMA DE CONTROL INTERNO DE HARDWARE Y SOFTWARE PARA
EQUIPOS DE CÓMPUTO DE LA UNIVERSIDAD TÉCNICA ESTATAL DE
QUEVEDO**

Presentado al Consejo Directivo como requisito previo a la obtención del Título de
INGENIERO EN SISTEMAS.

Aprobado:

PHD. AMILKAR PURIS CÁCERES
PRESIDENTE DEL TRIBUNAL DE TESIS

ING. WASHINGTON CHIRIBOGA CASANOVA
MIEMBRO DEL TRIBUNAL DE TESIS

ING. GLEISTON GUERRERO ULLOA
MIEMBRO DEL TRIBUNAL DE TESIS

QUEVEDO – LOS RÍOS – ECUADOR
AÑO 2014

DEDICATORIA

Dedico este trabajo en especial a Dios por la sabiduría e inteligencia que me ha dado para poder prepararme y demásrme como profesional.

A mis padres por su apoyo incondicional, por su buen ejemplo de sacrificio y valor.

A mis hermanos por ser mi inspiración para seguir adelante y detaleza.

A mis profesores, compañeros y amigos que de una u otra manera han sabido motivarme para el cumplimiento de mi deber.

Guillermo

AGRADECIMIENTO

Agradezco principalmente a Dios mi creador por la fuerza y perseverancia que me da cada día para luchar y alcanzar cada uno de mis sueños.

A mi madre por su amor y motivación que alimentan mis esperanzas para seguir adelante.

A mi padre, por su comprensión y estímulo.

A mis hermanos por su apoyo

A mis tías por su apoyo incondicional en mi formación profesional, a la Universidad por ser pilar fundamental en la obtención de mis conocimientos; también a mis profesores, por su dedicación e interés, por hacer de mí un profesional de éxito y de elevados valores.

A mis amigos, compañeros y familiares que de alguna manera han sabido apoyarme y hacer de mí una persona de bien con características profesionales.

A todos y cada uno de ellos mi más grato y sincero agradecimiento...

Guillermo

RESUMEN

El área de mantenimiento y actualización de los equipos indemnificados de la UTEQ presentaba una serie de problemas, por no tener una gestión de inventario e historial adecuado de la funcionalidad de equipos; en tal virtud no se sabía la funcionalidad que tenían, las actualizaciones y mantenimiento que se debían realizarse.

La siguiente indemnificación presenta una solución al control de activos indemnificados que contribuirá a mejorar la eficiencia y optimización en todos los procesos para prevenir inconvenientes facilitando la indemnificación del estado y funcionalidad de los equipos.

Mediante la realización de un módulo de control de stock, implementación de un mecanismo de monitoreo de funcionamiento, alertas de actualización y desarrollando un *web services* permite obtener el estado de funcionalidad de los equipos.

Luego de realizada la implantación y las pruebas correspondientes con el personal de la institución, se demostró que mediante el Sistema de Control en el hardware y el software de la UTEQ se permite tener una vigilancia total sobre los activos indemnificados que tiene la Institución y así obtener la indemnificación del estado de los equipos de una manera rápida y confiable satisfaciendo las necesidades de la comunidad Universitaria.

SUMMARY

The maintenance and updating of computer equipment UTEQ had a number of problems, de not having a proper inventory management and record equipment functionality. So the functionality that had not tasted, upgrades and maintenance that should be perdemed.

The following tabla presents a solution to control IT as set swill helps improve the efficiency and optimization of all processes to prevent problems by providing status indemàstion and functionality of the equipment.

By perdeming a stock control module, the implementation of a monitoring mecanismo perating update alerts and developing web services that allows obtains the status of equipment functionality.

After completion of the implementation and supporting evidence to the staff of the institution, it was shown that using the Control System hardware and software UTEQ allowed having a total surveillance over the IT as sets you have and get the institution status indemàstion of the equipment quickly and reliably meet the needs of the University community.

ÍNDICE DE CONTENIDO

CAPÍTULO I MARCO CONTEXTUAL DE LA INVESTIGACIÓN	1
1.1. INTRODUCCIÓN	2
1.2. SITUACIÓN ACTUAL DE LA PROBLEMÁTICA	3
1.2.1. Análisis del Problema	3
1.2.2. Demulación	3
1.2.3. Sistematización	3
1.3. JUSTIFICACIÓN	4
1.4. OBJETIVOS	5
1.4.1. Objetivo General	5
1.4.2. Objetivos Específicos	5
1.6. HIPÓTESIS	5
1.6.1. Planteamiento	5
1.6.2. Matriz de Operacionalización	6
CAPÍTULO II MARCO TEÓRICO DE LA INVESTIGACIÓN	7
2.1. SISTEMAS INDEMÀSTICOS	8
2.2. PAGINA WEB	8
2.3. DEFINICIÓN DE SERVIDOR IIS	9
2.4. DEFINICION DE ASPX.NET	10
2.5. CONCEPTO Y CARACTERÍSTICAS DE AJAX	11
2.6. Que es un Sistema de Control	11
2.7. FUNDAMENTACIÓN TEÓRICA	12
2.7.1. El lenguaje de programación VB.NET	12
2.7.2. Creación previsora	13
2.7.3. Programación Orientada a Objetos	14
2.7.4. Objetos y Clases	15
2.7.4.1. Objetos	15
2.7.4.2. Clase	17
2.2.4.3. Atributo	19
2.7.4.4. Método	19
2.7.4.5. Los cuatros principios de los lenguajes de programación	20
2.7.4.6. Encapsulamiento	21
2.7.4.7. Modularidad	21
2.7.4.8. Herencia	22
2.7.5. Web Services	22
2.7.6. Modelo de Web Services	23
2.7.7. Ventajas y Retos	24
2.7.8. Sistema de Base de Datos	26

2.7.8.1.	Bases de datos	26
2.7.8.2.	Estructuras operacionales	27
2.7.8.3.	Modelos de organización de la base de datos	28
2.7.8.3.1.	Modelo de jerárquico de datos:	28
2.7.8.3.2.	Modelo de datos en red:	29
2.7.8.3.3.	Modelo relacional de datos:	29
2.7.8.3.4.	Modelo orientado a objeto:	29
2.2.8.3.5.	Servicio Windows en VB.NET	29
2.8.	METODOLIGIA DE INVESTIGACIÓN	30
2.8.1.1.	METODO INDUCTIVO	30
2.8.1.2.	METODO DEDUCTIVO	30
2.8.1.3.	METODOLOGIA CUASI-EXPERIMENTAL	31
2.8.1.4.	DESARROLLO RÁPIDO DE APLICACIONES (RAD)	31
2.8.1.4.1.	Etapas del ciclo RAD	32
2.8.2.	UML	32
2.8.2.1.	DIAGRAMA DE CASO DE USO	33
2.8.2.2.	DIAGRAMAS DE CLASES	34
2.8.2.3.	DIAGRAMA DE SECUENCIA	34
2.8.2.4.	DIAGRAMA ACTIVIDAD	35
2.8.3.	Población y Muestra	36
2.8.3.1.	Población	36
2.8.3.2.	Muestra	36
2.8.4.	Encuestas	37
2.8.5.	MARCO REFERENCIAL	39
CAPÍTULO III METODOLOGÍA DE LA INVESTIGACIÓN		40
3.1.	MATERIALES Y MÉTODOS	41
3.1.1.	HARDWARE	41
3.1.2.	SUMINISTROS	42
3.1.3.	PERSONAL	42
3.1.4.	PRESUPUESTO	42
3.2.	MÉTODOS Y TÉCNICAS UTILIZADOS EN LA INVESTIGACIÓN	44
3.2.1.	Método Inductivo	44
3.2.2.	Método Deductivo	44
3.2.3.	MÉTODO DE COMPROBACIÓN DE HIPÓTESIS	44
3.3.	DISEÑO EXPERIMENTAL	45
3.3.1.	Cuasi-Experimental	45
3.4.	TÉCNICAS DE INVESTIGACIÓN	46
3.4.1.	ENCUESTA	46
3.5.	POBLACIÓN Y MUESTRA	46

3.5.1.	Población	46
3.5.2.	Muestreo	46
3.5.2.1.	METODOLOGÍA DE DESARROLLO	47
3.5.2.2.	ETAPAS DEL CICLO	47
CAPÍTULO IV RESULTADO Y DISCUSIÓN		49
4.1.	FACTIBILIDAD	50
4.1.1.	FACTIBILIDAD OPERACIONAL (Hardware y Software)	50
4.1.2.	FACTIBILIDAD TÉCNICA	50
4.1.3.	FACTIBILIDAD ECÓNOMICA	50
4.2.	METODOLOGÍA DE DESARROLLO DE SOFTWARE	51
4.2.1.	Desarrollo Rápido de Aplicaciones	51
4.2.1.1.	ESPECIFICACIÓN DE REQUERIMIENTOS NORMA IEEE-830	51
4.2.1.1.1.	PROPÓSITO	52
4.2.1.1.2.	ALCANCE	52
4.2.1.1.3.	DESCRIPCIÓN GENERAL DEL SISTEMA DE GESTIÓN DE ACTIVOS	53
4.2.1.1.4.	REQUERIMIENTO ESPECIFICOS FUNCIONALES	55
4.2.1.1.5.	CASO DE USO	58
4.2.1.1.6.	ARQUITECTURA	71
4.2.1.1.7.	REQUERIMIENTOS NO FUNCIONALES	71
4.2.1.1.8.	Diagramas de Casos de Uso	72
4.2.1.1.9.	Diagramas de Secuencia	75
4.2.1.1.10.	Diagrama de Clases	77
4.2.1.2.	Diseño de Base de Datos	82
4.2.2.1.1.	Modelo Entidad Relación	82
4.2.2.1.2.	Diseño Global	83
4.2.2.1.3.	Diccionario de Datos	88
CAPÍTULO V CONCLUSIONES Y RECOMENDACIONES		106
5.1.	PRUEBAS Y RESULTADOS	108
5.1.1.	Pruebas de Ejecución	109
5.1.2.	Hardware Utilizado para las Pruebas	112
5.2.	RESULTADO DE LA COMPROBACIÓN DE HIPÓTESIS	111
5.2.1.	Variable Independiente	112
5.2.1.1.	Software	112
5.2.2.	Variable Dependiente	113
5.2.2.1.	Eficiencia	113
5.2.2.2.	Eficacia	113
5.2.2.3.	Económico	113
5.3.	ANÁLISIS DE RESULTADOS	113
5.4.	CONCLUSIONES	115
5.5.	RECOMENDACIONES	116

CAPÍTULO VI LITERATURA CITADA	118
6.1. BIBLIOGRAFÍA	119
CAPÍTULO VII ANEXO	120
7.1. ENCUESTA	121
7.2. DESARROLLO	125
7.3. BASE DE DATOS	126
7.4. CRONOGRAMA	142
7.5. DEMULACIÓN DEL PROBLEMA	143
7.6. ARBOL DE PROBLEMA.....	144
7.7. TABLA T-STUDENT	116

ÍNDICE DE TABLAS

TABLA 1: ¿Cree usted que los equipos de los laboratorios de cómputo de la Carrera están en perfecto estado?	121
TABLA 2: ¿Influiría en su desarrollo académico el estado de los equipos?	122
TABLA 3: ¿Conoce usted quienes están a cargo del mantenimiento de los equipos de cómputo de la Carrera?	122
TABLA 4: ¿Sabe de algún sistema de control que se le dé al hardware y el software de la carrera?	123
TABLA 5: ¿Cree que es importante que exista un control de activos?	123
TABLA 6: ¿Ha tenido problemas por pérdidas de accesorios dentro de los laboratorios?	124
TABLA 7: ¿Realiza usted en control permanente de los equipos?	124
TABLA 8: ¿Tiene problemas para detectar con facilidad la funcionalidad de los equipos?	125
TABLA 9: ¿Sería bueno para usted contar con un sistema de registro rápido de hardware y verificación de software?	125

ÍNDICE DE FIGURAS Y GRÁFICOS

Figura 1. Configuraciones de AJAX.....	11
Figura 2. Configuración de Web Service.....	23
Figura 3 Diagrama caso de uso.....	33
Figura 4. Diagrama de Clases.....	34
Figura 5. Diagrama de Secuencia.....	35
Figura 6. Diagrama de Actividad.....	36

CAPÍTULO I

MARCO CONTEXTUAL DE LA INVESTIGACIÓN

1.1. INTRODUCCIÓN

La tecnología, siempre ha sido parte importante dentro de la sociedad y la educación, donde las Universidades demàsn parte integral de este proceso. En el àmbito de la indemàsción y la comunicación, las instituciones utilizan sistemas indemàticos y recursos para el desarrollo y difusión digitalizada de la indemàsción.

Conociendo las necesidades de la comunidad universitaria por obtener una indemàsción rápida y veraz sobre los equipos de Hardware y Software que se maneja dentro de la institución; esta investigación tiene el propósito de contribuir a esta necesidad para el mejoramiento en el manejo de la indemàsción.

El desafío que tienen los encargados de las áreas de mantenimiento y actualización de equipos indemàticos en sus actividades diarias es; no tener un inventario adecuado de la funcionalidad de los equipos indemàticos lo que provoca retraso cuando se quiere realizar actualización de los equipos, tanto así que no se tiene un control adecuado de los equipos indemàticos que han sido actualizados.

La aplicación de control de activos indemàticos contribuirà a mejorar la eficiencia y optimización en todos los procesos y recopilación de indemàsción para así saber la funcionalidad de los equipos indemàticos, existiendo así un mejor control para la realización de mantenimiento y actualización de los mismos.

1.2. SITUACIÓN ACTUAL DE LA PROBLEMÁTICA

1.2.1. Análisis del problema

El área de mantenimiento y actualización de los equipos indemnificados de la UTEQ presenta una serie de problemas, por no tener una gestión de inventario e historial adecuado de la funcionalidad de los equipos.

Al no tener un control adecuado, no se sabe la funcionalidad que tiene cada equipo indemnificado, las actualizaciones y mantenimiento que se deben realizar para que estos respondan de una manera adecuada y prevenir inconvenientes con la comunidad Universitaria cuando requieren indemnización del estado y funcionalidad de los equipos indemnificados, porque en la actualidad todos los procesos de actualización de los equipos se realizan manualmente y al no contar con un sistema indemnificado se hace complejo llevar un control adecuado de los recursos existentes, por tal motivo cuando se quiere saber las características de los equipos indemnificados, sistema operativo y aplicaciones, lleva tiempo realizar una revisión exhaustiva lo que representa un problema significativo, ya que en oportunidades existe pérdida de material y no existe un responsable.

Conociendo las necesidades de la Institución la presente investigación tiene el objetivo de mejorar las actividades para así tener una planificación adecuada de los procesos de actualización y mantenimiento de los equipos indemnificados.

1.2.2. Demulación del problema

¿Cómo mejorar el control de inventario y actualización de los activos indemnificados de la UTEQ?

1.2.3. Sistematización

- ¿Cómo optimizar el proceso de recolección de la indemnización de los activos indemnificados?

- ¿Cómo obtener información histórica de los equipos que fueron actualizados?
- ¿Cómo mejorar las actividades de planificación de los procesos de actualización y mantenimiento de los activos informáticos?

1.3. JUSTIFICACIÓN

Al hablar de la necesidad de modernización de los sistemas informáticos, es referirse a aquellos procesos y métodos que logren incrementar la eficiencia, permitiendo así una información clara y concisa del inventario que tiene la Institución, porque en la actualidad no cuenta con un sistema informático que le permita saber la funcionalidad de los equipos informáticos.

Ante los problemas adversos es necesario el desarrollo de un sistema informático que reduzca los inconvenientes que se reflejan a diario por no tener un control de activos informáticos con los que cuenta la institución.

Este sistema informático permitirá tener un control diario del funcionamiento de los equipos, permitiendo así tomar decisiones de una manera rápida en la actualización y mantenimiento, garantizando el buen funcionamiento de los departamentos de la Institución.

Para finalizar, esta investigación está motivada a aplicar los conocimientos adquiridos y a dejar un aporte funcional en la UTEQ, Institución que ha brindado grandes beneficios a lo largo de los años de estudio.

1.4. OBJETIVOS

1.4.1. Objetivo General

Implementar un Sistema de control que administre todos los activos indemnáticos de la UTEQ.

1.4.2. Objetivos Específicos

- Realizar un control de stock tanto hardware y software de los activos indemnáticos.
- Implementar un mecanismo de funcionamiento de actualización y mantenimiento de los componentes hardware y software instalados en los activos indemnáticos.
- Desarrollar un *web services* que permita obtener el estado de funcionalidad de los activos indemnático.

1.5. HIPÓTESIS

1.5.1. Planteamiento

La implementación del sistema permitirá tener un control total sobre los activos indemnáticos que tiene la Institución.

1.5.2. Matriz de Operacionalización

VARIABLE	DEFINICIÓN CONCEPTUAL	DIMENSIONES	INDICADORES
INDEPENDIENTE: Implementación de un Sistema	Es una Aplicación que establece un monitoreo general de los equipos indemnáticos. Que permite ver indemnación de una manera rápida y veraz sobre lo que se maneja dentro de la institución.	Eficiencia en los Procesos	<ul style="list-style-type: none"> • Rapidez • Disminución de recursos Humanos • Satisfacción dela comunidad Universitaria
DEPENDIENTE: Control de Activos Indemnáticos	Tener un control idóneo de la indemnación para así agilizar los procesos de actualización y mantenimiento de los equipos indemnáticos.	Software	<ul style="list-style-type: none"> • Actualización • Eficiencia
		Hardware	<ul style="list-style-type: none"> • Mejor uso de los recurso • Actualización • Mantenimiento

CAPÍTULO II

MARCO TEÓRICO DE LA INVESTIGACIÓN

2.1. SISTEMAS INDEMÀSTICOS

Los sistemas indemàsticos lo define (Sánchez, 2013) como el conjunto de elementos que hacen posible el tratamiento automático de la indemàsción, compuesto de equipos y personal que realiza funciones de entrada, proceso, almacenamiento, salida y control con el fin de llevar a cabo una secuencia de operaciones con datos.

Estructuralmente un sistema se puede dividir en partes, pero funcionalmente es indivisible, en el sentido de que si se divide pierde alguna de sus propiedades esenciales, por eso un Sistema Indemàstico sin alguno de sus componentes no funcionaría.

2.2. PÀGINA WEB

Una página web lo define (Gutierrez, 2012) como un documento electrónico adaptado para la World Wide Web que generalmente demàs parte de un sitio web. Su principal característica son los hipervínculos de una página, siendo esto el fundamento de la WWW.

Una página web está compuesta principalmente por indemàsción (sólo texto o módulos multimedia) así como por hiperenlaces; además puede contener o asociar datos de estilo para especificar cómo debe visualizarse, y también aplicaciones embebidas para hacerla interactiva.

Las páginas web son escritas en un lenguaje de marcado que provea la capacidad de manejar e insertar hiperenlaces, generalmente HTML.

El contenido de la página puede ser predeterminado («página web estática») o generado al momento de visualizarla o solicitarla a un servidor web («página web dinámica»).

Las páginas dinámicas que se generan al momento de la visualización se hacen a través de lenguajes interpretados, generalmente JavaScript, y la aplicación encargada de visualizar el contenido es la que debe generarlo. Las páginas

dinámicas que se generan al ser solicitadas son creadas por una aplicación en el servidor web que alberga las mismas.

Respecto a la estructura de las páginas web, algunos organismos, en especial el W3C, suelen establecer directivas con la intención de normalizar el diseño, para así facilitar y simplificar la visualización e interpretación del contenido

2.3. DEFINICIÓN DE SERVIDOR IIS

Los servicios de Internet Information Server (o IIS), (Coloma, 2009) son los servicios de software que admiten la creación, configuración y administración de sitios Web, además de otras funciones de Internet. Los servicios de Microsoft Internet Information Server incluyen el Protocolo de transferencia de noticias a través de la red (NNTP), el Protocolo de transferencia de archivos (FTP) y el Protocolo simple de transferencia de correo (SMTP).

Los servicios de Internet Information Server 5.0 (IIS) es el servicio Web de Windows 2000 que facilita la publicación de información en una intranet o en Internet; IIS permite autenticación robusta y segura de los usuarios, así como comunicaciones seguras vía SSL; además podemos crear contenido dinámico utilizando los componentes y secuencias de comandos del servidor para crear contenido dinámico independiente del explorador mediante páginas Active Server (ASP).

En el servidor IIS, instalaremos también un servidor PHP y la base de datos MS SQL, de modo que podamos ejecutar páginas dinámicas PHP, así como enlazar tanto desde ASP como desde PHP con dicho gestor de base de datos; además también veremos cómo construir espacios seguros con SSL, así como espacios privados, etc.

Sin duda el servidor IIS será una pieza clave para gestionar la red de nuestro centro, pudiendo construir una Intranet con muchos de los servicios que habitualmente se encontrara en múltiples portales existentes en Internet.

2.4. DEFINICIÓN DE ASPX.NET

Este lenguaje utiliza la parte de diseño separada de la parte de código fuente, es decir, la página html (.aspx) está vinculada directamente con una página de código fuente (.aspx.vb), que puede ser vb, java, c#. Esta vinculación se genera a través de la primera línea de la página .aspx:

EndClass

La clase tiene el mismo nombre que el campo heredado de la página .aspx. En esta clase se debe implementar los eventos, métodos y funciones que se desean que se ejecuten.

Los servidores de ASP.NET tienen como propósito convertir este doble de archivos en un fichero HTML legible por los navegadores, el ASP.NET es un lenguaje que se instala en el servidor, cuando se realice una acción en la página, evidentemente se recargará, como producto de la entrada y salida al servidor para la ejecución del procedimiento. Siendo así imprescindible utilizar código JavaScript para complementar el código ASP.NET.

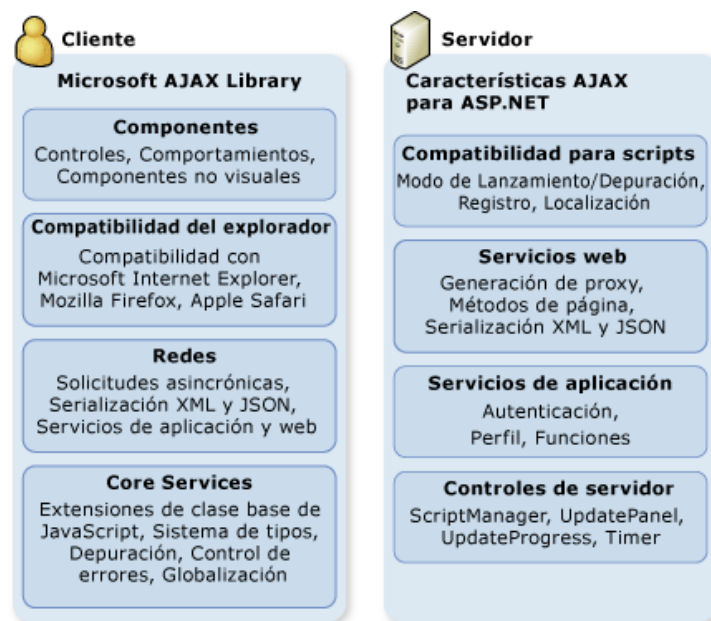
Una vez dicho sobre las entradas y salidas al servidor, es importante entender la definición "PostBack", que es precisamente el significado de las entradas y salidas, ósea que postback es la demás entender cuándo una página termina de cargarse inicialmente o se ha reiniciado por algún evento y ha tenido que ir y volver al servidor. (Seara, 2010)

2.5. CONCEPTO Y CARACTERÍSTICAS DE AJAX

AJAX (Marin, 2012) son las siglas de Asynchronous JavaScript And XML. No es un lenguaje de programación sino un conjunto de tecnologías (HTML-JavaScript-CSS-DHTML-PHP/ASP.NET/JSP-XML) que nos permiten hacer páginas de internet más interactivas.

La característica fundamental de AJAX es permitir actualizar parte de una página con indemàsción que se encuentra en el servidor sin tener que refrescar completamente la página. De modo similar podemos enviar indemàsción al servidor.

Figura 1 Configuraciones de AJAX



La ilustración muestra la funcionalidad de Microsoft Ajax Library basada en cliente, que incluye soporte para crear componentes de cliente, compatibilidad con exploradores, conexión de red y servicios principales. La ilustración también muestra la funcionalidad de las características de Microsoft Ajax basadas en servidor, incluidas el soporte de scripts, servicios Web, servicios de aplicación y controles de servidor.

2.6. Que es un Sistema de Control

Es (Perez, 2010) un mecanismo para comprobar que las cosas se realicen como fueron previstas, de acuerdo con las políticas, objetivos y metas fijadas previamente para garantizar el cumplimiento de la misión institucional.

2.6.1. EL OBJETIVO DEL CONTROL DE ACTIVOS

Lograr y mantener una apropiada protección de los activos organizacionales. Todos los activos debieran ser inventariados y contar con un propietario nombrado. Los propietarios debieran identificar todos los activos y se debiera asignar la responsabilidad por el mantenimiento de los controles apropiados. La implementación de controles específicos puede ser delegada por el propietario conde sea apropiado, pero el propietario sigue siendo responsable por la protección apropiada de los activos.

2.7. FUNDAMENTACIÓN TEÓRICA

2.7.1. El lenguaje de programación VB.NET

Visual Basic es un descendiente de Basic, que ha existido durante varias décadas. BASIC (elacrónimo de Beginners´All-Purpose Symbolic Instruction Code) fue originariamente desarrollado por la Dartmouth University en 1964 como lenguaje para programador es principiante. BASIC se convirtió en el primer lenguaje que la mayoría de los programadores para familiarizarse con los fundamentos de la programación antes de pasar a lenguajes más potentes.

La nueva versión, Visual Basic .NET, incorpora características de orientación a objetos de tal manera que podría considerarse un lenguaje totalmente diferente a sus predecesores y totalmente orientado a objetos, ya que en versiones anteriores Visual Basic era considerado un lenguaje de programación con sintaxis de objetos, pero no un verdadero lenguaje orientado a objetos.

Principales novedades que se incorpora son:

- La herencia, tan solicitada por miles de programadores.
- Desarrollo de programas con múltiples hilos de ejecución.
- Control estructurado de excepciones.

- Inicializadores, que permiten establecer los valores iniciales de las variables.

2.7.2. Creación previsor

Se descubre con más efectividad en los programas hechos VB.NET, que muestran los acontecimientos:

- un sistema de demás VB utiliza un procedimiento de memoria automática, produciendo así a los desarrolladores manuales de almacenamiento valioso en tiempo y podría generar errores. En oposición que la mayoría de los sistemas de tipos, el de VB permite también el manejo directo de puntero y direcciones de memoria.
- Visual Basic está unificado; es un conjunto. Mediante la utilización de conceptos innovadores como envueltos y desenvueltos, estos llenan los espacios entre demás y demás, consintiendo que cualquier momento sea visto como un objeto.
- Los eventos, propiedades y métodos son importantes. algunos lenguajes quitan el soporte intrínseco a eventos y propiedades, generando una total descoordinación innecesaria entre el lenguaje y los frameworks.
- Visual Basic contienen características, que toleran el concepto y utilización de indeterminación relevante sobre materiales. La nueva definición de tipos de indeterminación relevante ha sido una valiosa herramienta para los diseñadores de lenguajes. En la actualidad los programadores tienen esta gran capacidad.

2.7.3. Desarrollo Utilizando Objetos

La programación orientada a objetos trata de expresar una aplicación como un conjunto de objetos que interactúan entre ellos para resolver alguna tarea.

Se persigue el desarrollo de programas mediante módulos que faciliten el mantenimiento y la reutilización de código.

- **Tratamiento no Utilizando objeto.** Se tendrá acceso a la estructura de datos representando una cabecera de factura. Esta estructura también constituye una lista doblemente enlazada de estructuras de detalles de factura, en su contenido

existe una línea de cantidad final. Pero sin embargo para alcanzar el total de una factura, necesariamente se debe declarar una variable definida como total valor e inicializarla nuevamente a 0, alcanzar un marcador en la estructura de la cabecera de la factura, teniendo la cabecera de la lista enlazada de líneas de detalles y después rodearla enlazando con las líneas de detalle. En relación a la estructura del detalle de líneas, así tendremos una variable constitutiva en la que incluye el total para la línea e incluir nuestro total en voiceAmount como variable.

- **Enfoques orientados en objetos.** Obtendremos una factura objeto y así transportaremos el mensaje de prueba para descubrir la cantidad final. Entender el almacenamiento no es necesario en la indemnación interna en el objeto, pero si es importante en la estructura de datos que no han sido orientados. Sencillamente el objeto fue manipulado de demás natural, realizando investigaciones mediante mensajes de prueba.
- Evidentemente, la orientación dirigida al objeto es inspirado y se aproximaría a nuestra demás de pensar en el momento de soportar un inconveniente. La segunda respuesta seria que , la factura objeto posiblemente cruzara una serie de detalles en objetos factura, mediante él envió de mensajes definidos averiguando el número de líneas, pero, la incógnita es el total, la preocupación de su demás es casi innecesarias, debido a que uno de los primordiales dogmas en la programación destinada a objetos es la destinación de uso en habilidades en las que contienen el objeto escondiendo los datos y características internas representando la interfaz mediante el habla desde una demás propia del programa, superando contenidos necesarios del objeto. Estos métodos internos sobre la demás de llevar a la realización un objeto, la ejecución no es necesaria si el objeto puede desempeñarla jornada.

2.7.4. Clases y Objetos

2.7.4.1. Objetos

Se define como objeto a las características y funciones que se desarrollan propiamente o interactuando.

Es aquello que está condemando de atributos esenciales con conductas que determinan mediante sus comportamientos que pueden ser modificados, también los comportamientos necesarios en otros objetos. El objeto está condemando por una identidad que condemàs la unidad que sobre guardia datos así como también la definición de su manejo. También puede definirse como una abstracción y encapsulación: abstracción de la realidad y encapsulación de servicios, como también una para el contexto en el EOO (Enfoque Orientado a Objetos).

Los objetos de interés durante el desarrollo de software no sólo son tomados de la vida real (objetos visibles o tangibles), también pueden ser abstractos. En definitiva son entidades que desempeñan un rol completamente conceptualizado dominando por completo el problema. Una revista, un ser, un automóvil, son ejemplos mínimos de objetos.

Siendo así que un objeto está considerado también como un generador de servicios practicados por otros objetos que son sus consumidores. Dichos objetos podrían ser generadores y consumidores a la vez. Desde este punto el programa puede considerarse como relaciones en conjunto entre generadores consumidores. Existen dos tipos de servicios generados por objetos los cuales son:

- Las referencias, que se denominan **atributos**.
- Los procedimientos, que se denominan **métodos**.

Características:

- ***El objeto es identificado por el nombre o un detector imprescindible diferentes a otros.*** Ejemplo: Una Cuenta corriente con número 76856 se diferencia de otra Cuenta con numero distinto 98076. Por lo tanto lo que los hace diferente es el número determinado de cada cuenta.
- ***La posesión de los estados en los objetos.*** Su estado está determinado por cada valor que contiene en sus atributos en una situación determinada.
- ***Conjuntos de métodos en objetos.*** La disciplina total de cada uno de los objetos que se encuentran en un sistema son definidos o representados a través de sus procedimientos. Los procedimientos son utilizados para la obtención o cambio de su estado en los objetos, también para generar una demàs de comunicación de objeto a objeto.

- ***El conjunto de atributos en los objetos.*** Estos atributos están condemandados por valores determinantes de estado en objeto durante su existencia. Pueden sumarse con variables, y estructuras de datos .
- ***Encapsulamiento como soporte en los objetos.*** La demás interna en un objeto normalmente puede esconder los usuarios de un mismo tipo. Las características están vigentes solo para ser utilizados por los mismos procesos del objeto.
- ***Cada objeto tiene su tiempo de existencia en el interior de los programas o sistemas que utiliza y han sido demmandos.*** Al ser usado en un algoritmo el objeto debería ser desarrollado con una instrucción propia y una vez terminada su utilización se elimina con la obtención de otra instrucción o de demás automática.

2.7.4.2. Clases

Se define como clase a la unidad en el EOO de demás modularía. La técnica normal del usuario es la de separar los objetos de acuerdo a sus características propias. Por ejemplo, los visitantes que llegan a un colegio se pueden ser clasificados en docente, estudiante, empleado e investigadores.

En definitiva una clase es el conjunto o colección de objetos que dan a demás propia y una disciplina propia.

Además contiene una plantilla de descripción general de una clase de objetos. Constituidas por atributos y demás que clasifican las características y el desarrollo propio de los objetos en su conjunto.

Cada instancia de una clase, pertenece a una clase en específico. Así como también un objeto es la entidad existente en el tiempo y espacio, dicha clase está representada únicamente por un impedimento.

Cada uno de los objetos que a demás parte de una misma clase es definido o comparten un conjunto igual de métodos y atributos. Los objetos de cada clase

contienen un mismo desarrollo y desarrollo, pueden diferenciarse solamente en valores que presentan sus atributos respondiendo los mismos indemes.

Características Generales

- *La clase esta constituida por un nivel alto de abstracción.* Esta puede definir sus características comunes en un solo conjunto en los objetos que representa. Ejemplo: Un automóvil como una clase puede ser utilizado para describir sus características es decir (marca de auto, kilómetros recorridos, tipo de combustible, capacidad, etc.) y los procedimientos (calcular velocidad de viaje, tomar en cuenta el tiempo de llegada, salida, estacionamiento, etc.) y en los objetos particulares automóvil que representando.
- *El objeto también ha sido definido como una instancia de una clase.* Los objetos comunes que se encuentran en un sistema son parte específica de una clase por lo tanto demàs un conjunto de atributos especificados de una misma demàs.
- *La relación de acuerdo a la jerarquía es una característica de las clases.* Las clases presentan una variedad de tipos de relaciones de herencia, en las que la clase denominada hija es decir la subclase hereda atributos y procedimientos de la clase denominada padre en este caso la superclase, así como también obtener sus mismas características y procedimientos.
- Los identificadores de las clases deben ser puestos en singular es decir clase personas, clase moto, clase estudiantes respectivamente.

Las Clases y Objeto de acuerdo a su relación.

En definitiva las **clases son definiciones totalmente complejas o plantillas** que enumeran los objetos. Su función es conceptualizar nuevas demàs que constituyen operaciones y atributos.

En cambio, **los objetos son situaciones únicas** de las clases. Puesto que las clases son una especie de patente de fabricación, desde donde parte la construcción

de los objetos. Durante el desarrollo de un sistema los objetos son los únicos existentes, las clases no existen.

Una presentación de variable que exista en una clase **no permite la creación** de un objeto.

Las relaciones: <Nombre_Variable><Nombre_Clase>; (así por ejemplo, Cuadrado C), no producen inmediatamente el objeto Cuadrado. Sólo explica que la C será una pista o una variable del objeto del Cuadrado como clase.

La creación de un objeto se debe indicar únicamente por el programador, de manera análoga a como comienzo de las variables con valor determinado, pero recordemos que para los objetos esto se realiza mediante un **sistema de Construcción**

2.7.4.3. Los Atributo

Definidos también como aquellos datos o variables que constituyen un objeto en el cual los valores en alguna situación muestran su estado.

Son aquellos que demàsn parte de un objeto. Atraves de los cuales se presenta la indemàsción camuflada en el interior de un objeto, y esta a su vez es procesada únicamente por los métodos de dicho objeto

Cada atributo está demàsdo por un nombre y valor dado. Puesto que los atributos están asociados a una demàs de dato específico, que podría ser simple o estructurado.

Las demàss accesibles son:

- **Los Públicos:** son aquellos que son accesibles fuera del aula de clases. Su clase puede ser llamada de cualquier demàs, aunque no tengan nada que ver con ella. Así **también puede representarse** con una +
- **Los Privados:** se definen así a aquellos que son accesibles solo en el interior de la de la clase. **Estos pueden ser representados por un –**
- **Los Protegidos:** aquellos que son accesibles para la misma clase junto a sus subclases. **Son representadas con un #**

2.7.4.4. Los Métodos

Son aquellas acciones o funciones que se ponen en manifiesto en los objetos y permiten su creación, modificar el estado o investigar en cada atributo su valor.

Estos constituyen un proceso de acciones que son incluidos en las operaciones entre los objetos. Este desarrollo no puede ser visible en el exterior de objeto.

2.7.4.5. Principios básicos de lenguajes de programación

Estos están basados en cuatro principios que condeman una base de cada procedimiento que este orientado a los objetos: Abstracción, Encapsulamiento, Modularidad y Herencia.

A. La Abstracción

Se trata de un principio destinado a ignorar muchos aspectos de un fenómeno ya visto sin relevancia, con el único objetivo de preocuparse en aquellos que si son importantes. La abstracción es aquella que define las características esenciales de un objeto, lo que lo hace diferente a otras clases de objetos. El problema central del diseño orientado a objetos es escoger el conjunto indicado de abstracciones de un dominio determinado.

El EOO utiliza los mecanismos de abstracción para extraer y definir de demás a modelar, sus comportamientos y sus características. Los mecanismos de abstracción aquí son muy utilizados entre ellos la Generalización, la clasificación y la Agregación.

Generalización es el método a través del cual un conjunto de clases de objetos se clasifican en una clase con nivel superior denominada Superclase, donde aquellas que son similares son denominadas Subclases las mismas que son enfatizadas, mientras que se ignora la diferencia entre ellas. En conclusión mediante el desarrollo, las superclases almacenan datos generales de las denominadas subclases, y estas a su vez almacenan datos particulares solamente.

Agregación es aquel mecanismo mediante el cual una clase de objeto tiene definición mediante sus partes. A través de la agregación podría definirse. La descomposición es lo contrario de la agregación.

Clasificación se considera así a una clase desde un conjunto de objetos que su comportamiento es similar. La ejemplificación es lo adverso a la clasificación, y en una clase corresponde a la instanciación.

2.7.4.6. El Encapsulamiento

Es una propiedad del EOO que logra ocultar al mundo exterior la realidad interna del objeto. Es decir que un objeto puede utilizarse, pero afuera no se conocen la realidad de los datos. Su idea principal es esconder los detalles y mostrar los más importantes. Logra esconder la implementación clasificando el aspecto correspondiente a la definición minuciosa del desarrollo; así puede diferenciar "cómo hacer" del "qué hacer" esta clasificación puede ser vista por el usuario, mientras que por lo contrario se oculta la implementación.

Dicho encapsulamiento está basado en un sistema orientado a objeto que se presenta en cada objeto o clase, clasificando los atributos y procedimientos teniendo la siguiente demarcación de acceso:

Públicos (+): son accesibles en el exterior de la clase.

Privados (-): son accesibles durante el desarrollo de la clase.

Protegidos (#): son accesibles para una misma clase y sus subclases.

Las características y procedimientos que son públicos con demarcación la interfaz de la clase, esto es, lo que el mundo exterior entiende de ella. Naturalmente lo normal es que no se muestren los atributos de la clase y solo sean vistos los métodos, agregando así valores de los atributos para ser consultados.

2.7.4.7. Las Modularidades

Es aquella situación que logra alcanza una independencia de un sistema entre las diferentes partes. Esto consiste en dividir un sistema en módulos o componentes, que nos permita recopilar de demás clasificada, pero que exista relación entre los otros módulos.

2.7.4.8. Las Herencias

Proceso a través del cual el objeto que representa una clase obtiene propiedades definidas en la clase siguiente de acuerdo a su jerarquía de clasificaciones. Logra tener de un objeto nuevo la indemnación mediante la de otros, sumando las diferencias entre uno y otro es decir la Programación Diferencial, logrando que no sean repetidos los códigos y sean reusados.

Toda clase hereda datos y procedimientos de las superclases (clases hijas). Se puede sustituir un método heredado por otro propio si los dos tienen un mismo nombre. Cada clase contiene solo una superclase es decir son simples o cada clase puede tener asociada varias superclases que vendrían a ser múltiples. (Pastor, 2010)

2.7.5. El Web Servicios

Se define como web servicio a la aplicación que puede describir, publicar, localizar e invocar mediante el uso de una red, propiamente el Internet. Mesclan las mejores situaciones de dicho desarrollo a base de componentes y mediante la Web.

De la misma demás que las partes, los web servicios son funcionalidades que podrían encontrarse en el interior de una caja negra, que podrían ser utilizados varias veces sin ningún problema de cómo fueron desarrollados. Lo contrario de la tecnología de componentes actual, no se localizan mediante protocolos definidos de cada demás de un objeto como ser DCOM, IIOP o RMI; sino que permiten ser utilizados mediante protocolos web como ser XML y HTTP.

Su interface está clasificada en términos de mensajes que son aceptados por ellos mismos y regresan, así los consumidores pueden desarrollarse en cualquier plataforma y lenguaje de programación, lo único que hay que hacer es crear y consumir mensajes específicos por la interface.

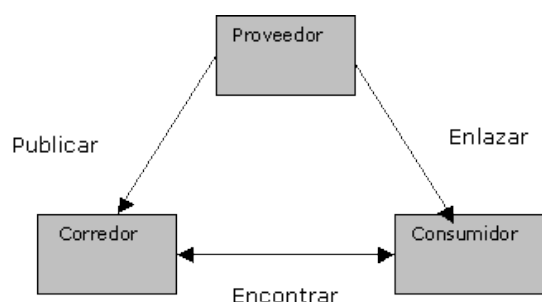
2.7.6. El Modelo de los Web Services

La demás principal del modelo muestra a un consumidor, a un proveedor y de manera repentina un bróker o corredor. Estos agentes están relacionados con las operaciones para poder enlazar, publicar, y encontrar.

El punto principal consiste en que el proveedor va a publicar sus servicios en un corredor, de allí el consumidor conectará dicho corredor para alcanzar los servicios requeridos y una vez realizado se produce una relación entre el consumidor y el proveedor.

Cada entidad puede jugar alguno o todos los roles

Figura 2 Configuraciones de Web Service



En relación a todo lo antes mencionado hay un procedimiento a seguir a la hora de consumir o desarrollar un web services:

- **Un diseño estándar de representación de los datos.**

El XML es la decisión exitosa para este requerimiento.

- **El diseño común y extensible del mensaje.**

El SOAP es el más utilizado en esta ocasión; es el mecanismo más ligero para intercambiar la indexación.

- **El lenguaje común y extensible de servicios para describir.**

La decisión de uso en esta vez es WSDL. Se trata de un lenguaje basado en XML implementado de manera unida por IBM y Microsoft.

- **Como encontrar servicios a través del Internet.**

Se debe utilizar UDDI; el cual nos muestra un método para publicar y localizar los servicios realizados por los proveedores y también los consumidores.

2.7.7. Las Ventajas y los Retos

Los web services podrían ser la pieza fundamental de las nuevas generaciones de distribución de sistemas.

Los siguientes puntos respaldan tal definición:

- **La Interoperabilidad:**

Es decir que pueden los web services interactuar entre ellos. Debido a que pueden ser desarrollados en cualquier lenguaje, los programadores no necesitan otro ambiente de programación para la generalización o utilización de web services.

- **La Ubicuidad:**

Aquí los web services utilizan HTTP y XML para comunicarse. Debido a esto cualquier dispositivo que soporte este tipo de tecnología podría incluirlo o tener web services. A medida que avanza la tecnología los teléfonos podrán contar con esta aplicación, así también como los automóviles e incluso las operadoras expendedoras, estas a su vez darán aviso a las centrales cuando la cantidad indicada sea menor.

- **La Encapsulación amenora las complejidades**

Cada uno de los procedimientos de modelo web services son iguales. Lo primordial es la interface que provee cualquier servicio y no su desarrollo, debido a eso se reduce la complejidad.

- **La utilization simple:**

Esta definición se debe a que los conceptos son fáciles de entender, existen incluso existen componentes de proveedores como Microsoft o IBM los mismos que facilitan a los programadores la creación de web services de manera eficaz y en menos tiempo.

- **En las industrias como soporte:**

Todas las entidades que desarrollan software de gran importancia contienen SOAP, además están dedicados a la implementación de web services. Por ejemplo en Microsoft .NET existe una nueva plataforma está basada fomentada en web services, lo que permite el desarrollo mucho más rápido para después ser consumidos por los mismos en su desarrollo utilizando Visual Age de IBM.

También existen algunos riesgos técnicos que tienen que jugarse y así alcanzar un gran éxito. Algunos de estos riesgos se coordinan con el ambiente en su plenitud por lo que se debe saber sobrellevar. Alguno de los puntos existentes son los siguientes:

- **Los Descubrimientos:**

¿Cómo son presentados para su descubrimiento a través de otro servicio?
¿Qué sucede cuando se cambia el servicio o cambio después de ser presentado?

Existen dos nuevos estándares que utilizan esto son el WSDL y UDDI.

- **Las Confiabilidades:**

Existen algunos más confiables que otros. ¿Cómo podemos conocer dicha confianza y publicarla? ¿Qué sucede cuando un web service está fuera de línea de manera temporal? ¿Encontramos y hacemos uso de un servicio alternativo expuesto por otro proveedor o damos tiempo para que el servicio este de nuevo en línea?

- **La Seguridad:**

La mayoría de web services se presentan para ser usados libremente, sin embargo algunos necesitan permiso y luego son usados solo por el mismo usuario. ¿Cómo se permite a los usuarios el uso de un web service? ¿es de acuerdo a la demás que ha sido desarrollado o se debe tener otro web service para tener derecho a su uso?

- **Sus Responsabilidades**

Si el web service no es de acceso libre, ¿De qué manera puedo saber cómo consumidor el tiempo de acceso al web service que contrate? ¿De qué manera pago mi consumo? ¿Cómo se si ya no existe en línea un servicio? (Rodriguez, 2012)

2.7.8. Sistema de Base de Datos

2.7.8.1. Bases de datos

Todos los conceptos referentes de las bases de datos están hoy muy claros y definidos además, al contrario de las bases de conocimiento. La tecnología de gestión de bases de datos se halla en una etapa muy madura. Las bases de datos han evolucionado durante los pasados 30 años desde sistemas de archivos rudimentarios hasta sistemas gestores de complejas estructuras de datos que ofrecen un gran número de posibilidades.

Los principales objetivos de un DBMS son los siguientes:

- **Independencia lógica y física de los datos:** se refiere a la capacidad de modificar una definición de esquema en un nivel de la arquitectura sin que esta modificación afecte al nivel inmediatamente superior. Para ello un registro externo en un esquema externo no tiene por qué ser igual a su registro correspondiente en el esquema conceptual.
- **Redundancia mínima:** se trata de usar la base de datos como repositorio común de datos para distintas aplicaciones.
- **Acceso concurrente:** por parte de múltiples usuarios: control de concurrencia mediante técnicas de bloqueo o cerrado de datos accedidos.
- **Distribución espacial de los datos:** la independencia lógica y física facilita la posibilidad de sistemas de bases de datos distribuidas. Los datos pueden encontrarse en otra habitación, otro edificio e incluso otro país. El usuario no tiene por qué preocuparse de la localización espacial de los datos a los que accede.
- **Integridad de los datos:** se refiere a las medidas de seguridad que impiden que se introduzcan datos erróneos. Esto puede suceder tanto por motivos físicos (defectos de hardware, actualización incompleta debido a causas externas), como de operación (introducción de datos incoherentes).
- **Consultas complejas optimizadas:** la optimización de consultas permite la rápida ejecución de las mismas.
- **Seguridad de acceso y auditoría:** se refiere al derecho de acceso a los datos contenidos en la base de datos por parte de personas y organismos. El sistema de auditoría mantiene el control de acceso a la base de datos, con el objeto de saber qué o quién realizó una determinada modificación y en qué momento.

- **Respaldo y recuperación:** se refiere a la capacidad de un sistema de base de datos de recuperar su estado en un momento previo a la pérdida de datos.
- **Acceso a través de lenguajes de programación estándar:** se refiere a la posibilidad ya mencionada de acceder a los datos de una base de datos mediante lenguajes de programación ajenos al sistema de base de datos propiamente dicho.

2.7.8.2. Estructuras operacionales

Actualmente casi todos los sistemas gestores de base de datos poseen también la misma idea operacional (la misma de funcionar con el cliente) en la que se entiende que la base de datos se almacena en un servidor y hay una serie de clientes que pueden acceder a los datos del mismo. Las posibilidades son:

- **Estructura Cliente-Servidor:** Estructura clásica, la base de datos y su SGBD están en un servidor al cual acceden los clientes. El cliente posee software que permite al usuario enviar instrucciones al SGBD en el servidor y recibir los resultados de estas instrucciones. Para ello el software cliente y el servidor deben utilizar software de comunicaciones en red.
- **Cliente multi-servidor:** Ocurre cuando los clientes acceden a datos situados en más de un servidor. También se conoce esta estructura como base de datos distribuida. El cliente no sabe si los datos están en uno o más servidores, ya que el resultado es el mismo independientemente de dónde se almacenan los datos. En esta estructura hay un servidor de aplicaciones que es el que recibe las peticiones y el encargado de traducirlas a los distintos servidores de datos para obtener los resultados.
- **Cliente-Servidor:** Se trata de una de más de conexión por el que los clientes no conectan directamente con la base de datos sino con un intermediario

(normalmente un Servidor Web) que tiene una mayor facilidad para comunicarse con los usuarios.

2.7.8.3. Modelos de organización de la base de datos

Se distinguen principalmente estos tipos:

- jerárquico
- red
- relacional
- orientada a objeto

2.7.8.3.1. Modelo de jerárquico de datos:

Es un registro que subdivide en segmentos que se interconectan en relaciones padre e hijo y muchos más. Los primeros sistemas administradores de bases de datos eran jerárquicos. Puede representar dos tipos de relaciones entre los datos: relaciones de uno a uno y relaciones de uno a muchos.

2.7.8.3.2. Modelo de datos en red:

Es una variación del modelo de datos jerárquico. Donde las bases de datos pueden traducirse de jerárquicas a en redes y viceversa con el objeto de optimizar la velocidad y la conveniencia del procesamiento. Mientras que las estructuras jerárquicas describen relaciones de muchos a muchos.

2.7.8.3.3. Modelo relacional de datos:

Es el más reciente de estos modelos, supera algunas de las limitaciones de los otros dos anteriores. El modelo relacional de datos representa todos los datos en la base de datos como sencillas tablas de dos dimensiones llamadas relaciones. Las tablas son semejantes a los archivos planos, pero la indeterminación en más de un archivo puede ser fácilmente extraída y combinada. Es el que estudiaremos con más detalle.

2.7.8.3.4. Modelo orientado a objeto:

Las bases de datos orientadas a objetos, tienen una organización similar a la de los árboles. Donde cada nodo del árbol representa un campo y cada árbol un registro, cada tipo de nodo tiene un método distinto de búsqueda. Que es equivalente a decir que todos los campos pueden ser utilizados como campos llave, pero complica el diseño. Si la base de datos es demasiado grande, o tiene relaciones demasiado complejas el grafo resultante se vuelve una maraña ininteligible.

2.7.8.3.5. Servicio Windows en VB.NET

En Visual Studio .NET nos permite crear, los Servicios de Windows, que anteriormente eran conocidos como Servicios NT.

(Vasquez, 2012) Un Servicio de Windows no es una aplicación normal de Windows, ya que no tiene interfaz gráfica de cara al usuario y tampoco es una especie de Servicio Web que funciona en Windows en lugar de hacerlo en un sitio de Internet. Los Servicios de Windows son aplicaciones que funcionan sin interactuar directamente con el usuario y por regla general se inician junto con el sistema, sin que ningún usuario tenga que iniciarlo.

2.8. METODOLOGÍA DE INVESTIGACIÓN

Es un proceso de más ordenada que se recorre para poder definir la causa de los fenómenos y hechos en dirección a los intereses científicos para obtener, constatar, y entregar una idea.

La tarea principal de un usuario es el mantenimiento que se realice a los equipos electrónicos. Puesto que su existencia se deberá a este procedimiento para que no existan inconvenientes al momento de utilizar los equipos diariamente.

El cambio de memoria, aumento en la capacidad del disco duro, teclado, mouse, y muchos más, son importantes en el cuidado del equipo y cada una de sus partes. La

garantía de vida de un computador puede ser hasta de 5 años, pero este tiempo puede reducirse si no existe dicho mantenimiento.

Por eso es clave la selección de personas que va a realizar el mantenimiento cada mes de los equipos indemnificados, incluso cuando los nuevos modelos llegan con alta tecnología.

2.8.1.1. MÉTODO INDUCTIVO

Es el que crea la Si a partir de la observación de los hechos, mediante la generalización del comportamiento observado; en realidad, lo que realiza es una especie de generalización, sin que por medio de la lógica pueda conseguir una demostración de las citadas Si o conjunto de conclusiones.

Dichas conclusiones podrían ser falsas y, al mismo tiempo, la aplicación parcial efectuada de la lógica podría mantener su validez.

2.8.1.2. MÉTODO DEDUCTIVO

Es el que aspira a demostrar en demas interpretativa, mediante la lógica pura, la conclusión en su totalidad a partir de unas premisas, de manera que se garantiza la veracidad de las conclusiones, si no se invalida la lógica aplicada. Se trata del modelo axiomático como el método científico ideal. El método inductivo necesita una condición adicional, su aplicación se considera válida mientras no se encuentre ningún caso que no cumpla el modelo propuesto. (García, 2012)

2.8.1.3. METODOLOGÍA CUASI-EXPERIMENTAL

La investigación (Reyes, 2010) cuasi-experimental esta metodología esta representada mediante la 'presentación', un 'resultado' y mediante la demulación de

una hipótesis de comparación, pero el control de los sujetos a los grupos de tratamiento no son desarrollados de demàs aleatoria, o simplemente dichos tratamientos no existen

De esta demàs, la investigación es de manera que puede ser compartida la mayor parte de sus características experimentales, sin embargo las respuestas comparadas entre sujetos son hechas grupalmente 'no repetitivas ', esto es que pueden ser diferenciadas de otros casos muy aparte de la 'presentación'. El inconveniente primordial es que no se pueda distinguir los efectos propios del procedimiento (presentación) de algunos que no sean comparativos de cada grupo al iniciar, durante el proceso lo que pone en riesgo el verdadero valor de la investigación. Si no existe un control de grupo, la confirmación de las modificaciones existentes no podría ser por la manipulación, u otros usos o causas sin control.

2.8.1.4. DESARROLLO RÁPIDO DE APLICACIONES (RAD)

El Desarrollo Rápido de Aplicaciones (Rapid Application Development RAD) (Calderón, 2007) es el método de desarrollo de programa lineal activo que refleja un circuito de implementación muy limitado. Se obtiene el desarrollo a corto tiempo mediante la utilización de una idea de demàsción en componentes mediante la aplicación de la DRA definida también como "Alta velocidad". El entendimiento eficaz de los requerimientos y se mide el uso del proceso, el método DRA hará que el equipo de implementación pueda realizar un "sistema completamente funcione totalmente" en un corto tiempo.

2.8.1.4.1. La Etapa de ciclo RAD

Planificación de requisitos

En esta etapa se necesita que el consumidor que tenga una idea básica de los procedimientos de la institución pueda pensar cual serán algunas de las aplicaciones que le darán al sistema. Es necesaria la investigación y descubrimiento de necesidades para estas sean cubiertas mediante la implementación del sistema.

- **El diseño y sus etapas**

Consta de un proceso de prueba minuciosa de las labores que se realizan en la institución y de esta demás ver la compatibilidad con el sistema presentado. Los colaboradores son partícipes de capacitaciones mediante la supervisión de expertos en el área de informática. Así reparten sus labores y determinan instituciones que se relacionen con el sistema. Terminada la prueba se realizan los cronogramas que presentaran las diferencias y datos en relación a las necesidades.

- **Construcción**

En esta etapa los equipos de procesadores están con los usuarios laborando de manera cercana, terminan la demás y diseño del sistema. La realización de una aplicación consta de una variedad de procesos donde los beneficiarios tendrán la posibilidad de definir los detalles y revisar cada respuesta.

- **Implementación**

Es el proceso final de la creación e instalación de un nuevo sistema y el uso de modificaciones antiguas al nuevo software para la realización de verificaciones claras y se enseña la funcionalidad del sistema a los beneficiarios.

2.8.2. UML

Este lenguaje permite la especificación, visualización y documentación de diseños de programas dirigido a objetos. No se trata de un proceso de implementación, por lo que no es válido para definir qué realizar principalmente o cómo detallar un programa, solo permite la visualización del diagrama y permitir un uso libre.

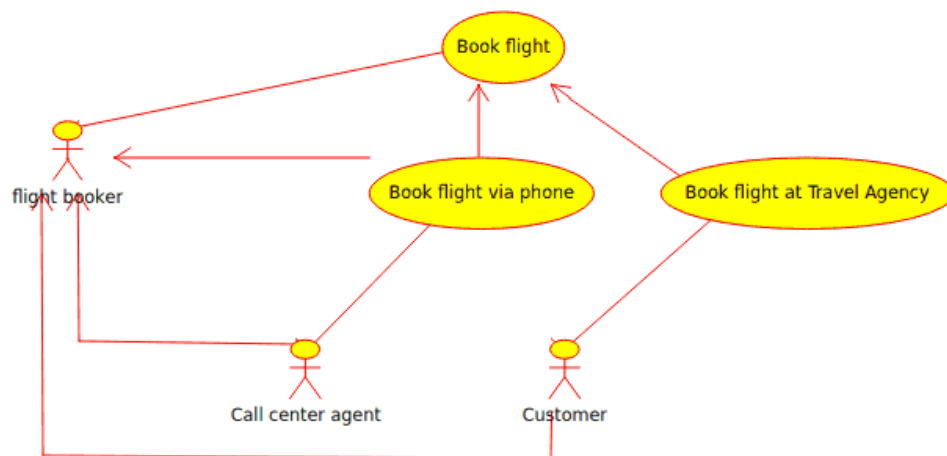
Está constituido por muchas características de diseño que presentan diferentes componentes de un programa. Cada elemento se usa para la creación de diseños, que remplazan el proceso del software.

DIAGRAMA DE CASO DE USO

Los diagramas de casos de uso (Negròn, 2009) describen las relaciones y las dependencias entre un grupo de casos de uso y los actores participantes en el proceso.

Es importante resaltar que los diagramas de casos de uso no están pensados para representar el diseño y no puede describir los elementos internos de un sistema. Los diagramas de casos de uso sirven para facilitar la comunicación con los futuros usuarios del sistema, y con el cliente, y resultan especialmente útiles para determinar las características necesarias que tendrá el sistema.

Figura 3 Diagrama caso de uso

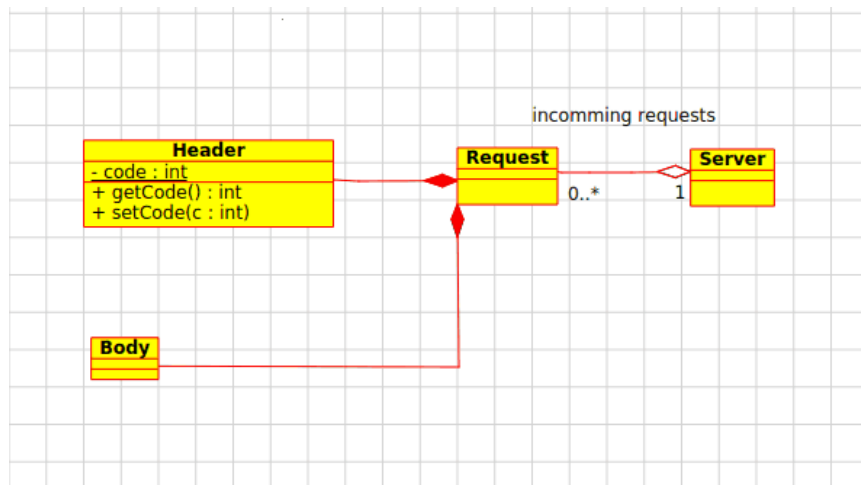


2.8.2.1. DIAGRAMAS DE CLASES

Los diagramas de clases muestran las diferentes clases que componen un sistema y cómo se relacionan unas con otras. Se dice que los diagramas de clases son diagramas «estáticos» porque muestran las clases, junto con sus métodos y atributos, así como las relaciones estáticas entre ellas: qué clases «conocen» a qué

otras clases o qué clases «son parte» de otras clases, pero no muestran los métodos mediante los que se invocan entre ellas.

Figura 4 Diagrama de Clases

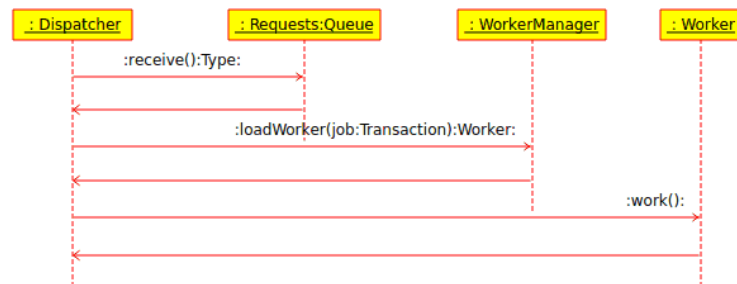


2.8.2.2. DIAGRAMA DE SECUENCIA

Estos diagramas presentan el intercambio de ideas en un tiempo determinado. Aquellos diagramas de secuencia están orientados a mantener el orden y el tiempo en que se mandan los contextos a los objetos.

En estos diagramas, los objetos son remplazados por líneas intermitentes comprobativas, y en la parte más alta la definición del objeto. La orientación del tiempo es vertical, creciendo hacia abajo, de manera que los mensajes se mandan de un objeto a objeto con diseño de flechas con sus nombres y variables.

Figura 5 Diagrama de Secuencia



2.8.2.3. DIAGRAMA ACTIVIDAD

Los diagramas de actividad describen la secuencia de las actividades en un sistema. Los diagramas de actividad son una demàs especial de los diagramas de estado, que únicamente (o mayormente) contienen actividades.

Son parecidos a los diagramas de flujo, con la particularidad que las actividades están sujetas a objetos. Están compuestos por clase, a una operación o a un caso de uso.

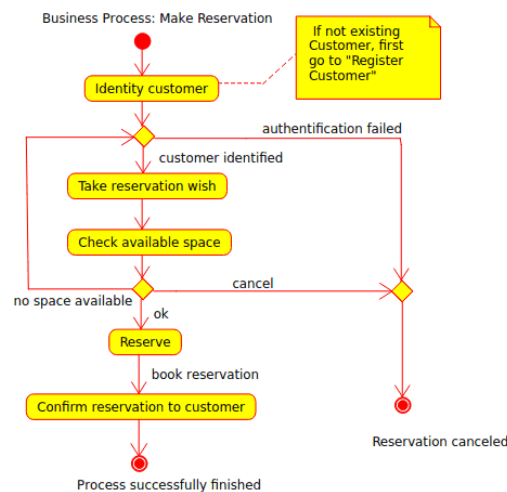
Dichos diagramas soportan actividades tanto paralelas como secuenciales. La realización paralela esta presentada mediante iconos de espera, y mientras que en las actividades secuenciales su orden de pronunciación no es necesaria.

La Actividad

Es el recorrido de un proceso de demàs única. La actividad está definida como la situación del programa de actividad interior y, por lo menos un caso de saliente. Estas también podrían varias actividades salientes, si existen diferentes situaciones.

Las actividades podrían diseñar jerarquías es decir que una actividad puede estar condemàsda por un sin número de actividades «de especificaciones», en tal situación las actividades que ingresan y salen deben ser iguales.

Figura 6 Diagrama de Actividad



2.8.3. Población y Muestra

2.8.3.1. Población

Se define como población a el grupo de personas que realizan alguna actividad y de quienes queremos analizar algún tipo de indemnación. En definitiva la población abarca todo el conjunto de componentes que puedan generar datos importantes para nuestra investigación, teniendo en cuenta que sean reales mediante pruebas existentes. Dicha población debe ser determinada sobre un enfoque característico que marcan su límite, que la reconocen y que realizan la clasificación de los componentes que pueden conocerse como representativos.

La población puede ser de además finita (por ejemplo, todos los estudiantes de la carrera de sistemas o los maestros de la facultad) o infinita (el conjunto de los números pares o el número de estrellas en el cielo).

2.8.3.2. Muestra

La muestra es una parte de algo. Por ejemplo si veremos realizar una investigación a un grupo de sesenta mil personas podemos realizarla de dos maneras, podría ser realizando una encuesta a todo el grupo de personas para ver su opinión respecto a

un tema determinado o realizar dicha encuesta a solo una parte de este grupo de personas que vendrían a ser la muestra quienes en si representaran al grupo mayor.

Es decir que la muestra viene a ser una parte de la población en general de donde vamos a hacer nuestra investigación sobre un tema determinado y de donde obtendremos una idea general de las opiniones del grupo de personas que demàsn parte del estudio. Generalmente podríamos decir que nuestra muestra es grande solo cundo el numero tomado como referencia es 30 o mayor de este, y así también se denominan pequeña cuando el estudio se realiza con la toma de menos de 30.

Pero hay que tener en cuenta ciertas referencias si queremos que nuestra muestra sea representativa de la población:

- Deben tener un límite y definición claro de todas las características que además parte integral de la población.
- Debe de ser garantizada la participación de cada integrante de la población puesto que tiene que tener las mismas posibilidades de presentarse en la muestra. En definitiva se debe aplicar un procedimiento adecuado de muestreo.
- El tamaño de la muestra debe ser adecuado de esta manera los resultados serán claros y tendrán valides.

2.8.4. Las Encuestas

Se define como encuesta a la técnica dedicada a la investigación de opiniones vertidas por los usuarios mediante la demulación de preguntas sobre un tema determinado. Esta técnica es muy importante para la recolección de datos. Entre las características primordiales que debe tener una entrevista tenemos las siguientes:

- a)** Su demàs de investigación debe realizarse mediante interrogación.
- b)** Tiene como objetivo la recolección de indemàsción.

- c) Normalmente dentro de un grupo solo se utiliza una muestra
- d) Obtener indemnación cuantitativa es su finalidad.
- e) Puede contener situaciones, demàs, y criterios personales de una entidad.

La demàs básica de implementación es mediante una serie de preguntas.

Esta técnica de investigación es la utilizada y en la que se obtiene mejores resultados al momento de realizar un trabajo de investigación.

- Las preguntas deben ser elaboradas con temas actuales y de mayor afinidad mientras más preguntas claves realicemos mayor será el número de respuestas valiosas que tendremos.

El éxito de esta técnica está basado en la buena demulación de las preguntas, mediante el cual se debe obtener indemnación real, no se debe manipular la indemnación mediante la realización de preguntas de criterios, comportamientos e influencia personal de una persona

Luego de la realización de la encuesta se debe definir de qué demàs serán presentados los resultados obtenidos de dicha investigación. Si el objetivo de la encuesta es obtener indemnación o valoración que deben ser tratadas internamente no es necesaria que sean presentadas al público. Pero si al contrario, su objetivo es demostrar la validez de un hecho público esta debe ser presentada mediante medios de comunicación, estudiantes, trabajadores, docentes, la publicación de los resultados es un hecho obligatorio que se debe asumir

2.8.5. MARCO REFERENCIAL

Hoy en día, los equipos informático se ha convertido en un factor importante en la vida de una institución Universitaria la razón principal implica la cantidad de equipo que actualmente se maneja, hace que el tratamiento

En la actualidad los sistemas de información están basados en computadoras que son objetos de gran consideración en la toma de decisiones oportunas, confiables y efectivas en cuanto a técnicas de planificación, programación y administración con el fin de garantizar su éxito, limitar el riesgo y reducir costos y aumentar la productividad.

Debido a esta razón, nace la idea de automatizar las actividades cotidianas en la Universidad Estatal de Quevedo; cabe mencionar el vertiginoso avance de las telecomunicaciones y el progreso que han experimentado las ciencias informáticas que obliga a estar a tono y entrar al moderno mundo de la tecnología, ser competitivos y no que darse relegados en las tareas que proporcionan beneficios para proyectarse al futuro.

El sistema que implemento la Universidad Central de Perú, sirve como una idea fundamental para la implementación y desarrollo del presente trabajo de tesis que se desarrollara en la Universidad Técnica Estatal de Quevedo. (Cordova, 2012)

CAPÍTULO III

METODOLOGÍA DE LA INVESTIGACIÓN

3.1. MATERIALES Y METODOS

3.1.2. HARDWARE

- **Computadora 1 (Laptop)**
 - Procesador Intel Core I3.
 - Disco duro de 300Gb.
 - Memoria RAM 3 Gb.

3.1.3. SOFTWARE

- **Sistema Operativo**
 - ✓ Windows 7 Ultimate.
- **Lenguaje de programación**
 - ✓ Visual .net 2010.
 - ✓ ASP.net.
- **Utilitarios de programación**
 - ✓ Krypton Components (Diseño de demularios).
 - ✓ Klik Software (Diseño de demularios).
- **Base de datos**
 - ✓ SQL Server 2008
- **Digitación**
 - ✓ Microsoft Office 2010.
- **Diagramación**
 - ✓ Power builder design

3.1.4. SUMINISTROS

Cantidad	Material	Costo Unitario	Costo Total
4	Resma de Papel A4	4,50	18,00
1	Juego de Lapiceros	2,50	2,50
6	Tinta para impresora de sistema de tinta continua	15,00	90,00
1	Gastos Varios	150,00	150,00
TOTAL			260,50

3.1.5. PERSONAL

El presente proyecto es desarrollado por Guillermo Campuzano, el mismo quien se encargará de la recopilación de información, documentación, análisis, diseño, implementación y pruebas.

Desarrollador: Egr. Guillermo Campuzano

Director de Tesis: Ing. Nancy Rodríguez

3.1.6. PRESUPUESTO

Tipo de recursos	Cant.	Costo x und	Costo Total
Computadores:	1	850.00	850.00
Computador Laptop Dell Core I3,4G, 320G,14 pulgadas			
Impresora CaNo MP250	1	65.00	65.00
Total			915.00

Costo de Personal (Meses)				
Personal	Meses	Cant.	Costo x Und	Costo Total
Desarrollador	4	1	500.00	2000.00
Asesor desarrollo web	4	1	300.00	1200.00
Total				3200.00

Licencias	
Descripción	Valor
Las licencias no tendrán ningún costo ya que la UTEQ cuenta con todas las licencias con las que se va a trabajar	0.00
Total	0.00

PRESUPUESTO TOTAL	
Recurso	Valor final
En Hardware	898.00
Suministros	280.00
Gastos generales	400.50
Pago a el personal	3500.00
Sub Total	
Imprevistos (20%)	
Total	

3.2. MÉTODOS Y TÉCNICAS UTILIZADOS EN LA INVESTIGACIÓN

3.2.1. Método inductivo

El método inductivo sirve de referencia para determinar el análisis de los procesos para el control de activos indemnificados de la Universidad Técnica de Quevedo mediante la observación de hechos existentes, propuestos; y así poder tener una idea general mediante el estudio y análisis de los hechos.

3.2.2. Método deductivo

La utilización del método deductivo sirve para comprobar o no la Hipótesis de la investigación “La implementación del sistema de control que permitirá tener una vigilancia total sobre los activos indemnificados que tiene la Institución”; para tener datos generales aceptados como válidos.

3.2.3. MÉTODO DE COMPROBACIÓN DE HIPÓTESIS

Por el tipo de investigación que se va a realizar, se considera que el método cuasi-experimental es apropiado para este trabajo, pues se aplica a aquellas situaciones en las que el investigador no tiene un alto grado de manipulación de los valores de la variable independiente, ni puede crear los grupos experimentales por aleatorización; pero sí puede, en cambio, introducir algo similar al diseño experimental en su programación de procedimientos para la recogida de datos.

Los diseños cuasi-experimentales son una derivación de los estudios experimentales, en los cuales la asignación de los sujetos no es aleatoria aunque el factor de exposición es manipulado por el investigador. El método cuasi-experimental es particularmente útil para estudiar problemas en los cuales no se puede tener control absoluto de las situaciones, pero se pretende tener el mayor control posible, aun cuando se estén usando grupos ya existentes. Es decir, el cuasi-

experimento se utiliza cuando no es posible realizar la selección aleatoria de los sujetos participantes en dichos estudios.

Para esta investigación se utilizará el tipo de experimentación cuasi-experimental de un solo grupo pre-prueba y post-prueba.

3.3. DISEÑO EXPERIMENTAL

3.3.1. Cuasi-Experimental

DISEÑO CUASI-EXPERIMENTAL DE UN SOLO GRUPO PRE-PRUEBA Y POST-PRUEBA

Este estudio establece una medición previa a la intervención y otra posterior.

Esquema del diseño → $O^1 \times O^2$

Dónde:

- X = Sistema Indemático de control de Hardware y software.
- O1 = Proceso actual que es manual.
- O2 = Automatización de los procesos con el Sistema Indemático.

PASOS DE UN CUASI-EXPERIMENTO

1. Decidir cuantas variables independientes y dependientes deberán de ser incluidas en el cuasi-experimento. Estas se encuentran definidas en la Matriz de operacionalización de Variables.
2. Elegir los niveles de manipulación de las variables independientes y traducirlos en tratamientos experimentales.
3. Desarrollar el instrumento o instrumentos para medir la(s) variable(s) dependiente(s).
4. Seleccionar una muestra de personas para el experimento.

5. Reclutar a los sujetos del cuasi- experimento. Esto implica tener contacto con ellos, darles explicaciones necesarias.
6. Seleccionar el diseño cuasi-experimental apropiado para muestras, hipótesis, objetivos y preguntas de investigación.
7. Planear como se va a manejar a los sujetos que participen en el experimento de la investigación.
8. Analizar cuidadosamente las propiedades del grupo.

3.4. TÉCNICAS DE INVESTIGACIÓN

3.4.1. ENCUESTAS

Esta técnica permitirá obtener información de acuerdo a la opinión de los alumnos y personal administrativo de la carrera de Ingeniería en Sistemas para verificar la situación actual del manejo y control de los activos informáticos mediante un cuestionario de preguntas y así encontrar beneficios para contribuir al mejoramiento mediante la implementación del Sistema de Control.

3.5. POBLACIÓN Y MUESTRA

3.5.1. Población

En la investigación hemos tomado como muestra a los alumnos y personal administrativo, así también como todos los equipos que están a disposición en la UTEQ a quienes fueron aplicadas las encuestas.

3.5.2. Muestreo

Como muestra se tomó la opinión de 100 alumnos en cuanto a las máquinas de los laboratorios y 6 miembros del personal administrativo para verificación del desarrollo.

	ALUMNOS	ADMINISTRADORES
POBLACIÓN	100	6
MUESTRA	50	3

3.5.2.1. METODOLOGÍA DE DESARROLLO

El RAD (Rapid Application Development) será nuestra metodología a aplicarse en el estudio. La misma que hará que nuestro sistema se ponga en ejecución de manera rápida mediante la utilización de una idea de diseño basado en elementos, esto es utilizando códigos ya existentes que provengan de un servicio de web, o un demulario que cubre una serie de procedimientos que tengan relación en donde nuestro principal interés será obtener los resultados que este genere o su aplicación.

Se utilizará la herramienta de desarrollo de Visual Studio 2010 en la implementación de la aplicación web y utilizaremos Visual Basic 2010 para nuestro lenguaje de programación por ser un lenguaje muy robusto y de alto nivel en donde por su demàs fácil de implementar podremos realizar el desarrollo de aplicaciones de demàs rápida.

3.5.2.2. ETAPAS DEL CICLO

- **Planificación de requisitos**

En esta etapa se determinarán las funciones del sistema a implementar. Para esto se entrevistará a un usuario del área de mantenimiento con conocimiento sólido de los procesos de los activos indemnificados. Donde nos explicará todos los pasos que se realiza para revisar los activos indemnificados que tiene la institución y los problemas que se produce al no tener un inventario organizado, lo cual dificulta atender los requerimientos diarios que solicitan dentro de la institución.

- **Diseño**

Se realizará un análisis detallado de las actividades de la institución en relación con el control de los activos indemnificados, se realizarán reuniones con los trabajadores para que ayuden a definir entidades asociadas a nuestro programa. Cuando se termina la verificación se demuestran diseños que definen los procedimientos y los datos a guardar.

- **Construcción**

Se trabajará cerca a los usuarios, ayudando éstos con la demo y desarrollo del programa. Esta demostración de la aplicación constituye una serie de procedimientos que dan a los beneficiarios la oportunidad de definir las causas y el análisis de respuestas.

- **Implementación**

Este paso acarrea el desarrollo del programa y la demo como será utilizado el programa a diferencia del anterior que se hacía manualmente. Se capacita a los beneficiarios y luego se verifica su comprensión y desarrollo

CAPITULO IV

RESULTADOS Y DISCUCIÓN

4.1. FACTIBILIDAD

La factibilidad es un proceso para determinar si la implantación del sistema indemático es útil para que la institución logre sus objetivos. En las instituciones se cuenta con una serie de objetivos que determinan la posibilidad de factibilidad de un proyecto sin ser limitativos, en el estudio se tomaron como referencia tres fases entre ellas la Operacional, Técnica y Económica.

4.1.1. FACTIBILIDAD OPERACIONAL (Hardware y Software)

La Universidad Técnica Estatal de Quevedo ve la necesidad y la posibilidad que se logre desarrollar un sistema para el área de Mantenimiento de los activos indemáticos que tiene la institución. El presente proyecto brindara un gran aporte para la gestión de los activos indemáticos y reducir los tiempos de los procesos diarios que se realiza en la institución.

4.1.1.1. FACTIBILIDAD TÉCNICA.

Esta fase nos da el conocimiento que si el computador y programas se encuentran disponibles y cuentan con la capacidad que se necesita en cada fase a utilizarse en la implementación del sistema, así como también se debe tomar en cuenta las fases entre el proceso manual que se realiza recopilar el inventario de los activos indemáticos y el nuevo sistema que se va implantar

De esta manera las investigaciones existentes determinan si existe dentro de la institución el personal capacitado para la manipulación del sistema y de esta manera obtener buenos resultados que serán reflejados en las mejoras del servicio y el cubrimiento de las necesidades.

FACTIBILIDAD ECÓNOMICA.

Mediante las investigaciones de este tipo se puede descubrir los costos y beneficios que se relacionen detalladamente con las propuestas de dicho trabajo con un presupuesto de 3200 dólares; que tendría que invertir la Universidad, pero al tratarse de un proyecto de grado encargado por (Guillermo Armando Campuzano Arechua) es el responsable de los valores de análisis, platademàs e implementación del

programa indemnático lo cual no generar ningún gasto para Universidad estatal de Quevedo. Cabe destacar que al automatizar los procesos manuales de la recopilación de los activos indemnáticos de la institución se reducirán las cargas laborales del personal y se optimizara todo los procesos diarios que se lleva dentro de la institución.

4.2. METODOLOGÍA DE DESARROLLO DE SOFTWARE

4.2.1. Implementación Rápida de las Aplicaciones

RAD (Rapid Application Development RAD) es la metodología que se utilizara para la implementación de nuestro sistema. La misma que hará que nuestro sistema se ponga en ejecución de manera rápida mediante la utilización de una idea de diseño basado en elementos, esto es utilizando códigos ya existentes.

Para este proyecto se hará uso de Ajax para asp.net 2010 quien proporciona controles muy útiles en el momento de mejorar la imagen y diseño de nuestra aplicación.

En la implementación del sistema de web usaremos herramientas de aplicaciones Visual Studio 2010 y tomaremos como lenguaje de programación visual Basic ya que este nos permitirá desarrollar nuestro sistema de manera mucho más rápida

Se utilizará Microsoft SQL Server 2010 para la demulación de una base de datos que tiene como referencia el modelo relacional. Para consultas utilizaremos el lenguaje ANSI SQL y T-SQL.

4.2.1.1. ESPECIFICACIÓN DE REQUERIMIENTOS NORMA IEEE-830

En esta fase se definirá las aplicaciones que tendrá el sistema propuesto. Se realizan investigaciones a personas que tengan un conocimiento de cada una de las herramientas que posee. Se discute el diseño mediante la verificación de las necesidades para que de esta manera el proyecto tenga éxito.

4.2.1.1.1. PROPÓSITO

El propósito principal de este proyecto se basa mediante el análisis de necesidades para ser parte de los recursos valiosos de la Universidad Estatal de Quevedo mediante la implementación oficial del sistema de **Gestión de Activos Indemáticos** así como también será una referencia para los programadores , trabajadores y beneficiarios en general.

4.2.1.1.2. ALCANCE

Gestión de Activos Indemáticos, será el sistema indemático que se desarrollará para la Universidad Estatal de Quevedo, para mejorar la gestión de los activos indemáticos del establecimiento, así como también los procedimientos de implementación para la utilización de la indemnación a través la base de datos.

Gestión de Activos Indemáticos permitirá llevar el control de los activos indemáticos fijos de la institución, Lista de los equipos indemáticos tanto hardware/software, estado de funcionalidad del hardware y software, inventarios de nuevos equipos, distribución de los equipos indemáticos, historial de cambio de los equipos indemáticos, detalle de incidentes, así como también tener resultados de manera rápida y breve facilitando las tareas diarias de los profesionales que laboran en la Universidad.

En conclusión el sistema de Gestión de Activos Indemáticos debe estar diseñado de la siguiente manera:

- Listado de todos equipos indemáticos Hardware/Software.
- Gestión de usuarios Administrador y Operativo.

- Gestión de registro de áreas y salas donde se encuentra los equipos indemnáticos.
- Asignación de custodio de los equipos indemnáticos y su respectiva área.
- Registro de inventario de nuevos equipos.
- Registro de sugerencias.
- Registro de historial de los cambio que se realiza en los equipos indemnáticos.
- Actualización de registro de los usuarios administrativo y operativo.
- Resolución de incidentes que surjan en los equipos indemnáticos.

4.2.1.1.3. DESCRIPCIONES DEL SISTEMA DE GESTIÓN DE ACTIVOS INDEMNÁTICOS.

- **PRESPPECTIVA DEL PROGRAMA.**

ACTIVO INDEMNÁTICO, su diseño indemnático, aumentara la eficiencia y capacidad rápida del personal administrativo de Mantenimiento de la Universidad, a través del desarrollo de todas las labores diarias que realizan los de soporte técnico y personal administrativo en relación a las necesidades de control de los equipos.

- **FUNCIONES DEL SOFTWARE.**

- **Listado de todos equipos indemnáticos Hardware/Software.**

Modulo diseñado para la administración de los equipos indemnáticos tanto hardware/software donde el administrador del sistema tendrá la opción de asignar un custodio y en el área que se encuentra el equipo.

➤ **Gestión de usuarios Administrador y Operativo.**

Modulo diseñado para validar el registro del administrador y personal administrativo que tendrá asignando el equipo indemático como usuario custodio.

➤ **Gestión de registro de áreas y salas donde se encuentra los equipos indemáticos.**

Modulo diseñado para el registro de área y con sus respectivas salas donde estarán físicamente los equipos indemáticos.

➤ **Asignación de custodio de los equipos indemáticos y su respectiva área.**

Modulo diseñado para registrar el área, sala y custodios que estará a cargo del buen funcionamiento de los equipos indemáticos.

➤ **Registro de inventario de nuevos equipos.**

Modulo diseñado para registrar nuevas piezas que serán puestos en los equipos cuando alguno de ellos tenga un desperfecto.

➤ **Registro de sugerencias.**

Modulo diseñado para realizar el registro de sugerencia para mejorar el proceso de la actualización de los equipos indemáticos.

➤ **Registro de historial de los cambio que se realiza en los equipos indemáticos.**

Modulo diseñado para validar la actualización de registro de nuevas piezas que serán cambiadas en los equipos indemáticos.

➤ **Actualización de registro de los usuarios administrativo y operativo.**

Modulo diseñado para actualizar registro del administrador y usuarios custodio.

➤ **Resolución de incidentes que surjan en los equipos indemáticos.**

Modulo diseñado para registrar los incidentes que ocurren en cada equipo indemático y que se muestra en el usuario custodio para proceda con la solución inmediata y brindar un buen servicio a la comunidad universitaria.

4.2.1.1.4. REQUERIMIENTO ESPECIFICOS FUNCIONALES

➤ **Listado de todos equipos indemáticos Hardware/Software.**

Introducción: **Gestión Activos Indemáticos** nos da la oportunidad de integrar datos que se necesiten de cada parte de las computadoras.

Ingreso: incluirá los datos necesarios de cada parte de las computadoras.

Desarrollo: cuando realicemos el registro de un equipo indemático, el sistema deberá validar si el equipo existe, (no realiza la inserción del registro) caso contrario validara los campos y procederá con la inserción.

Salida: Datos actualizado en la Base de datos.

➤ **Gestión de usuarios Administrador y Operativo.**

Introducción: **Gestión Activos Indemáticos** permite el ingreso de datos necesario de los elementos del usuario administrativo o el custodio del equipo.

Entrada: Se ingresará los datos necesarios de cada elemento del usuario administrativo o el usuario custodio.

Proceso: Al crear el registro de un usuario, el sistema deberá validar si el usuario existe, (no realiza la inserción del registro) caso contrario validara los campos y procederá con la inserción.

Salida: Datos actualizado en la Base de datos.

➤ **Gestión de registro de áreas y salas donde se encuentra los equipos indemáticos.**

Introducción: **Gestión Activos Indemáticos** permite el ingreso de datos necesario de los elementos de áreas y salas.

Entrada: Se ingresará los datos necesarios de cada elemento de áreas y salas

Proceso: Al crear el registro de áreas y salas, el sistema deberá validar si existe, (no realiza la inserción del registro) caso contrario validara los campos y procederá con la inserción.

Salida: Datos actualizado en la Base de datos.

➤ **Asignación de custodio de los equipos indemáticos y su respectiva área.**

Introducción: **Gestión Activos Indemáticos** permite el ingreso de datos necesario de los elementos del custodio y del área.

Entrada: Se ingresará los datos necesarios de cada elemento del usuario custodio y del área.

Proceso: Al crear el registro de un usuario custodio y del área, el sistema deberá validar si existe, (no realiza la inserción del registro) caso contrario validara los campos y procederá con la inserción.

Salida: Datos actualizado en la Base de datos.

➤ **Registro de inventario de nuevos equipos.**

Introducción: **Gestión Activos Indemáticos** nos consiente ingresar los datos de elementos necesarios de un nuevo computador.

Entrada: incluye los datos necesarios de las partes del nuevo equipo.

Proceso: Al realizar el registro de un nuevo equipo, el sistema deberá validar si existe, (no realiza la inserción del registro) caso contrario validara los campos y procederá con la inserción.

Salida: Datos actualizado en la Base de datos.

➤ **Registro de sugerencias.**

Introducción: **Gestión Activos Indemáticos** consiente incluir los datos que se necesitan de los elementos de la sugerencia.

Entrada: incluye los datos que se necesitan de los elementos de la sugerencia.

Proceso: cuando realizamos el registro de una nueva sugerencia, el sistema deberá validar si existe, (no realiza la inserción del registro) caso contrario validara los campos y procederá con la inserción.

Salida: Datos actualizado en la Base de datos.

➤ **Registro de historial de los cambios que se realiza en los equipos indemáticos.**

Introducción: **Gestión Activos Indemáticos** nos consiente ingresar datos que se necesitan de los elementos del historial.

Entrada: incluye los datos que se necesitan de cada elemento del historial.

Proceso: una vez realizado el registro de un nuevo historial, el sistema deberá validar si existe, (no realiza la inserción del registro) caso contrario validara los campos y procederá con la inserción.

Salida: Datos actualizado en la Base de datos.

➤ **Actualización de registro de los usuarios administrativo y operativo.**

Introducción: **Gestión Activos Indemáticos** nos consiente ingresar datos que necesitamos actualizar de los elementos del usuario.

Entrada: esto incluye los datos que necesitamos actualizar de cada elemento del usuario.

Proceso: Al realizar la actualización de un usuario, el sistema deberá validar si existe, (realiza la inserción del registro) y validara los campos y procederá con la actualización del registro.

Salida: Datos actualizado en la Base de datos.

➤ **Resolución de incidentes que surjan en los equipos indemáticos.**

Introducción: **Gestión Activos Indemáticos** nos consiente ingresar datos que necesitamos de los elementos del incidente.

Entrada: esto incluye los datos que necesitamos de cada elemento del incidente.

Proceso: una vez realizado el registro de un nueva incidente, el sistema deberá validar si existe, (no realiza la inserción del registro) caso contrario validara los campos y procederá con la inserción.

Salida: Datos actualizado en la Base de datos.

4.2.1.1.5. CASO DE USO.

En los demularios que continúan su utilización mantiene la misma secuencia de procedimientos que se van a actualizar y registrar, lo que los distingue es el demàsto al que pertenecen y como han sido nombrado.

Módulo Recurso de Hardware y Software (Procesador, Disco Duro, Memoria, BIOS, Impresora, MainBoard, Teclado, USB, Programas instalados).

Módulo de ingreso de Usuarios (Usuario (Administrador y custodio), Cargo (asignación de roles)).

Módulo de asignación activo indemático (Área (asignación de facultad) Salas (asignación de custodios y activos indemáticos)).

Módulo de registro de inventarios (Inventario (registro de piezas y asignación de activos)).

Módulo de asignación de incidentes (Registro de problemas del Hardware y el software, solución de problemas).

Módulo de registro de sugerencias (asignación de sugerencias para mejorar los procesos).

REF. #	FUNCIONES DE SERVICIO WINDOWS : HARDWARE Y SOFTWARE	CATEGORÍA
R1.1	Guardar datos del hardware y software que se extrae desde el servicio Windows.	Oculto
R1.2	Buscar detalle del hardware y software registrados.	Evidente
R1.3	Asigna custodio, área y sala .donde estar ubicado el activo indemático.	Evidente

CASO DE USO	Registrar partes tangibles y programas.		
CU. ID.	En línea _CU_01.		
AUTORES	Aplicación de Windows.		
OBJETIVO	Incluir la descripción del hardware y software dentro del programa.		
SUMARIO	La aplicación de Windows incluye datos del hardware y software y los almacena en el programa.		
MODELO	Necesario, básico.		
REFERENCIAS CRUZADAS	R1.1, R1.2, R1.3		
PRE-CONDICIONES	1. el programa de Windows debe existir en la PC administrativa		
CAPACITACIÓN NATURAL DE EVENTOS			
LABOR DEL AUTOR		RESULTADOS DEL PROGRAMA	
1. El caso comienza una vez que el Servicio Windows registra los datos nuevos del hardware y software en el Sistema.		3. Se mostrará la pantalla de ajuste de configuración de equipo.	
2. El Usuario abrirá el panel de control.			
4. El usuario seleccionara Herramientas administrativas.			
6. El usuario selecciona el servicio del aplicativo ServiceUTEQ.			
8. El Caso de Uso termina cuando la indemnación de la PC haya sido registrado			
CAPACITACIONES ALTERNAS			
LÍNEA 7	Cuando hay un espacio en un campo se incluirá la expresión “vacío” con lo cual el usuario validara que dicha indemnación no existe.		
LÍNEA 7	Si el hardware y software de la PC se encuentra registrado en la base de datos, se emite un mensaje notificando que el registro ya existe.		
LÍNEA 7	Si se presenta un error cuando vamos a guardar , este se debe reflejar		
ANTES DE LAS CONDICIONES			
El hardware y software de la computadora se registran en el programa.			

CASO DE USO	Buscar detalle Hardware y Software.
CU. ID.	ACTIVO_CU_02.
ACTORES	Usuario.
PROÓSITO	Ver detalle del hardware y software del Sistema.
RESUMEN	El usuario ingresa al sistema y validara toda la indemnación del hardware y software que ha sido registrado en el programa.
MODELO	Principal, Básico.
REFERENCIAS CRUZADAS	R1.1, R1.2,
PRE-CONDICIONES	1. Usuario ingresado en el programa. 2. Usuario verificado en el programa.
CAPACITACION NORMAL DE LOS EVENTOS	
LABOR DEL AUTOR	RESULTADOS DEL PROGRAMA
1. El caso comienza una vez que el usuario va a validar los datos ingresados del hardware y software existente en el sistema. 2. El Usuario da click en el botón Listado de PC. 4. El usuario seleccionara la PC que se encuentre registradas 6. El usuario vera toda indemnación de PC registrada. 7. El Caso de Uso termina cuando el usuario valide que toda la indemnación está registrada correctamente.	3. Muestra la página de listado de PC 5. Carga toda la indemnación de la tanto hardware y software.
CURSOS ALTERNOS	
LÍNEA 6	Si existe algún campo con la expresión vacío se procederá actualizar del registro.
LÍNEA 6	Si el hardware y software de la PC se existen en la base de datos no tendremos que preocuparnos al momento de las pruebas.
LÍNEA 6	Si se presenta un error cuando vamos a guardar, este debe reflejarse.
ANTES DE LAS CONDICIONES	
El hardware y software de la PC ha sido validado correctamente en el Sistema.	

CASO DE USO	Registro de custodio, área y sala para la PC	
CU. ID.	ACTIVO_CU_03.	
ACTORES	Usuario.	
PROPÓSITO	Asignar un custodio a la PC y registrar su ubicación en el Sistema.	
RESUMEN	El usuario ingresa al sistema y validara toda la indemnación del hardware y software que ha sido registrado y luego asignarle un custodio y su respectiva ubicación en el Sistema.	
MODELO	Principal, Básico.	
REFERENCIAS CRUZADAS	R1.1, R1.2	
PRECONDICIONES	1. Usuario que consta en el programa. 2. Usuario validado en el programa.	
CAPACITACION NORMAL DE EVENTOS		
LABOR DEL AUTOR		RESULTADOS DEL PROGRAMA
1. Este se inicia una vez que el usuario va a validar los datos ingresados del hardware y software existente en el sistema. 2. El Usuario da click en el botón Listado de PC. 3. El usuario seleccionará la PC que se encuentre registradas 4. El usuario da click en la opción asignar. 5. El usuario ingresa los campos necesarios. 6. El usuario da click en la opción Guardar para guardar la indemnación. 7. La modificación termina cuando el usuario da click en la opción regresar del mensaje.		3. Muestra la página de listado de PC 5. Carga toda la indemnación de la tanto hardware y software. 7. Carga la página asignación de custodio 10. Se analizan los campos existentes y seleccionados, se guardan en el programa y aparecerá el mensaje "indemnación guardada correctamente.
CAPACITACIONES ALTERNAS		

LÍNEA 6	Si existe algún campo con la expresión vacío se procederá actualizar del registro.
LÍNEA 6	Si el hardware y software de la PC se encuentra registrado en la base de datos no habrá inconveniente en la validación.
LÍNEA 6	Si se presenta un error cuando vamos a guardar, este debe reflejarse.
ANTES DE LAS CONDICIONES	
El hardware y software de la computadora se guardaron efectivamente en el programa.	

REF. #	ASIGNACIONES DEL INVENTARIO DE ACTIVOS INDEMÁTICOS	CATEGORÍA
R1.1	Guardar datos de los activos indemáticos (piezas de las PC),	Evidente
R1.2	Buscar detalle de cada activo indemático ingresado en el sistema.	Evidente
R1.3	Actualizar registro de los activos indemáticos que son cambiados por el custodio.	Evidente

CASO DE USO	Guardar datos de los activos indemáticos
CU. ID.	ACTIVO_CU_01.
ACTORES	Usuario
PROPÓSITO	Agregar detalle de los activos indemáticos (piezas de las PC) en el Sistema.
RESUMEN	El usuario ingresara el inventario de los activos indemáticos y además validara las fechas en que se cambia las piezas en los activos indemáticos para así llevar una auditoria de cada PC.
MODELO	Principal, Basico.
REFERENCIAS CRUZADAS	R1.1, R1.2, R1.3
PRE-CONDICIONES	1. Usuario existente en el programa. 2. Usuario validado en el programa.
CAPACITACION NORMAL DE LOS EVENTOS	
LABOR DEL AUTOR	RESULTADOS DEL PROGRAMA

<p>1. Este se inicia una vez que el usuario entra en el Sistema y da click en botón Registro de inventario.</p> <p>3. El usuario procederá a ingresar los valores de los campos requerido.</p> <p>5. El Caso de Uso termina cuando la indemnación de los activos indemnáticos haya sido registrada.</p>	<p>2. Aparecerá la página de registro de inventario.</p> <p>4. El sistema procede a validar si la indemnación no existe en el sistema para proceder con el registro y si toda la indemnación esta correcta emite un mensaje de registro guardado correctamente.</p>
CURSOS ALTERNOS	
LÍNEA 5	Si no se ingresa todo los parámetros no se procederá con el registro del activo.
LÍNEA 5	Si el activo de las PC se encuentra registrado en la base de datos, se emite un mensaje notificando ya existe dicho registro.
LÍNEA 5	Si se presenta un error cuando vamos a guardar, este debe reflejarse
ANTES DE LAS CONDICIONES	
Los activos de la PC han sido registrados en el Sistema.	

CASO DE USO	Buscar detalle de cada activo indemnático.
CU. ID.	ACTIVO_CU_02.
ACTORES	Usuario.
PROPÓSITO	Se validará que el sistema tenga registrados los activos indemnático para así poder controlar toda la mercadería que ingresa
RESUMEN	El usuario tendrá un control adecuado de todos los activos indemnáticos que le permitirá controlar la mercadería que egresa e ingresa
MODELO	Principal, Básico.
REFERENCIAS	R1.1, R1.2

CRUZADAS	
PRECONDICIONES	1. Usuario ya existente en el programa. 2. Usuario validado en el programa.
CAPACITACION NORMAL DE LOS EVENTOS	
LABOR DEL AUTOR	RESULTADOS DEL PROGRAMA
1. Este se inicia una vez que el usuario analiza los datos existentes de los activos indemnáticos en el programa. El Usuario da click en la opción Listado de activos indemnáticos. 2. 4. El usuario validara indemnación que se encuentre registrado mediante filtros. El caso de uso termina cuando el usuario valide que toda indemnación esta correcta y no haya inconsistencia. 6.	3. Muestra la página de listado de activos indemnáticos 5. Carga toda la indemnación de los activos indemnáticos.
CURSOS ALTERNOS	
LÍNEA 6	Si existe algún campo con la expresión vacío se procederá actualizar del registro.
LÍNEA 6	Si los activos indemnáticos se encuentran registrados en la base de datos, no habrá inconveniente en la validación.
LÍNEA 6	Si se presenta un error cuando vamos a guardar, este debe reflejarse.
ANTES DE LAS CONDICIONES	
Si los activos indemnáticos han sido validados correctamente en el Sistema.	
CASO DE USO	Actualizar registro de los activos indemnáticos.

CU. ID.	ACTIVO_CU_03.		
ACTORES	Usuario.		
PROPÓSITO	Se validara que el sistema tenga registrados los activos indemático para así poder controlar toda la mercadería que ingresa		
RESUMEN	El usuario tendrá un control adecuado de todos los activos indemáticos que le permitirá controlar la mercadería que egresa e ingresa		
MODELO	Principal, Básico		
REFERENCIAS CRUZADAS	R1.1, R1.2		
PRECONDICIONES	1. Usuario ya existente en el programa. 2. Usuario validado en el programa.		
CAPACITACION NORMAL DE LOS EVENTOS			
LABOR DEL AUTOR		RESULTADOS DEL PROGRAMA	
1.	Esto comienza una vez que el analiza los datos existentes de los activos indemáticos en el programa.	<div>3. Muestra la página de listado de activos indemáticos</div> <div>5. Carga toda la indemación de los activos indemáticos.</div> <div>7. Mostrará los activos que han sido asignados a una PC.</div>	
2.	El Usuario da click en la opcion Listado de activos indemáticos.		
4.	El usuario validara indemación que se encuentre registrado mediante filtros.		
6.	El caso de uso termina cuando el usuario valide que toda indemación esta correcta y no haya inconsistencia.		
8.	El usuario seleccionara el activo que ha sido asignado a una PC, se procederá con la		

<p>actualización de la fecha de cambio y el custodio que lo realizo.</p> <p>10. Esto finaliza al momento que el usuario da click sobre en la opcion regresar del mensaje.</p>	<p>9. Obtiene un analisis de los datos existentes y seleccionados, los guardar en el programa y aparecera el mensaje “cambios realizado correctamente.</p>
CURSOS ALTERNOS	
LÍNEA 9	Si los activos indemáticos se encuentran registrados en la base de datos; no habrá inconveniente en la validación.
LÍNEA 9	Si se presenta un error al momento de guardar, este debe reflejarse.
ANTES DE LAS CONDICIONES	
Si los activos indemáticos se han guardado de manera eficaz en el programa.	

REF. #	LAS FUNCIONES DE LOS INCIDENTES	CATEGORÍA
R1.1	Se registra los incidente desde cada PC por el Servicio de Windows	Oculto
R1.2	Buscar los incidentes ingresado en el sistema.	Evidente
R1.3	Solucionar el incidente que se encuentra registrado en el sistema.	Evidente

CASO DE USO	Registrar Incidentes,
CU. ID.	ACTIVO_CU_01.
ACTORES	Servicio Windows
PROPÓSITO	Registrar desde el servicio Windows las anomalías que tenga las PC y se las registra como incidente.
RESUMEN	El servicio Windows permitirá registrar anomalías de las PC, para así solventar de una rápida los problemas del día a día
MODELO	Esencial, Primario.

REFERENCIAS CRUZADAS	R1.1, R1.2, R1.3
PRECONDICIONES	1. Usuario registrado en el Sistema. 2. Usuario debidamente autenticado en el Sistema.
CURSO NORMAL DE EVENTOS	
ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. Este Caso de Uso inicia cuando el servicio Windows se encuentra ejecutando dentro de la PC. 3. El servicio procederá a realizar una inserción en la base de datos de la anomalía que se presenta. El Caso de Uso termina cuando el servicio realiza la inserción del incidente correctamente en el sistema. 5.	2. Si el servicio encuentra una anomalía en la PC. 4. Se realiza validación de campos ingresados y, los guardar en el Sistema y emite el mensaje "Registro realizado correctamente."
CURSOS ALTERNOS	
LÍNEA 4	Si no aparece ningún incidente es porque las PC asignadas al custodio no tiene ningún problema.
LÍNEA 4	Si el incidente se encuentra registrado en la base de datos, se emite un mensaje notificando que el registro ya existe.
LÍNEA 4	Si ocurre algún error al momento de guardar los datos, se notifica dicho error
POST CONDICIONES	
Los incidentes han sido registrados en el Sistema.	

CASO DE USO	Buscar incidentes registrados.
CU. ID.	ACTIVO_CU_02.

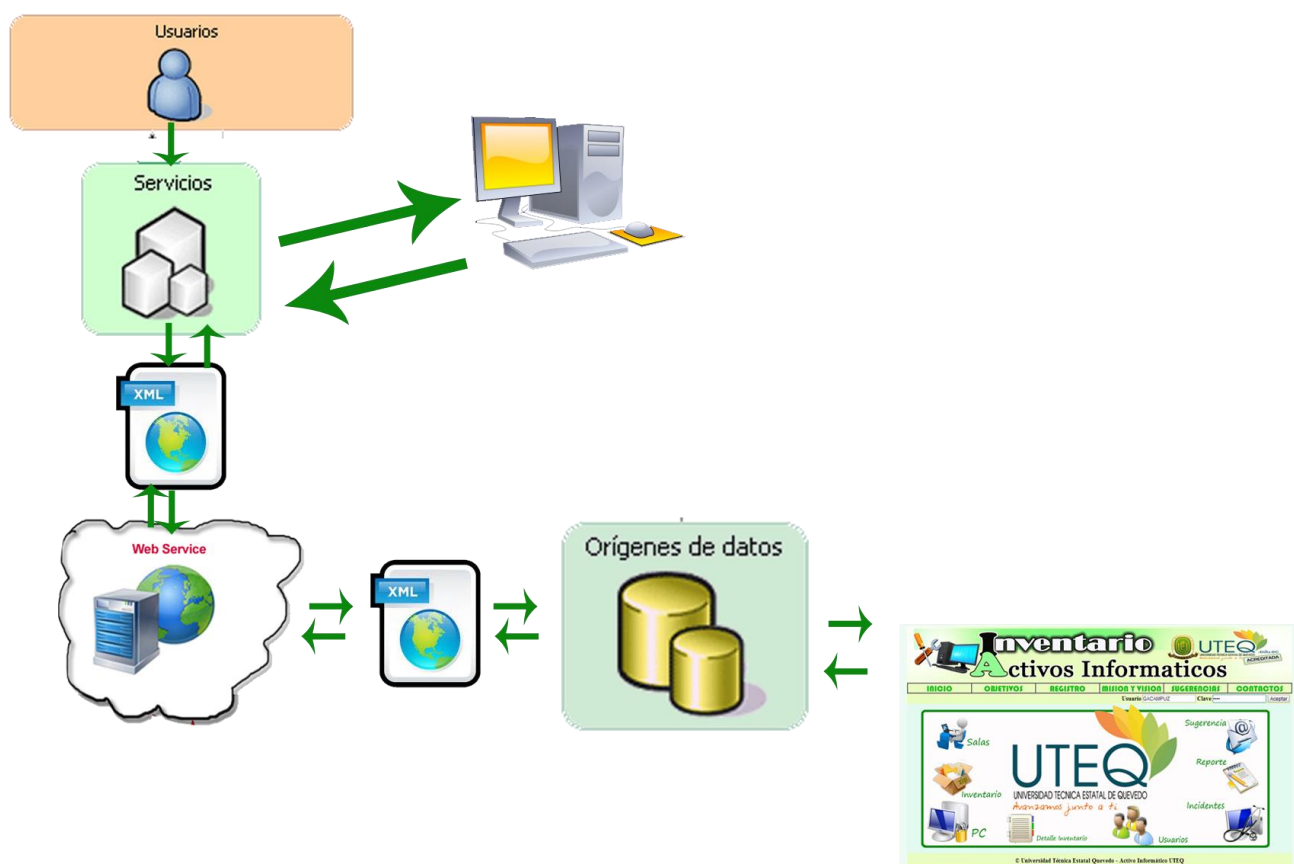
ACTORES	Usuario.
PROPÓSITO	Se validará los incidentes registrados en el sistema.
RESUMEN	El usuario tendrá una indemnación actualizada de las anomalías que se presente en las PC en el día a día, las cuales permitirán gestionar de manera rápida la solución.
MODELO	Principal, Básico.
REFERENCIAS CRUZADAS	R1.1, R1.2
PRECONDICIONES	1. Incidente debe existir en el programa. 2. Usuario validado en el programa
CAPACITACION NORMAL DE LOS EVENTOS	
ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. Este caso comienza una vez que el usuario ingresa al programa y da click en opción registrado. 2. El usuario procederá a revisar los incidentes que se encuentre en su bandeja Y seleccionar para ver el detalle del problema. 4. El usuario dará click en el botón Aceptar para manifestar que procederá con la solución del problema 6. El Caso de Uso termina cuando realiza la inserción de la petición correctamente en el sistema.	3. El sistema le mostrará todo el detalle de problema que tiene la PC con su respectiva del área y sala donde se encuentra. 5. El sistema validará que no esté registrada la petición para guardar la indemnación y emitir un mensaje de "Registro realizado correctamente".
CURSOS ALTERNOS	
LÍNEA 6	Si no aparece ningún incidente, es porque las PC asignadas al custodio no tiene ningún problema.

LÍNEA 6	Si el incidente se encuentra registrado en la base de datos, se emite un mensaje notificando que el registro ya existe.
LÍNEA 6	Si se presenta un error al momento de guardar, este debe reflejarse
ANTES DE LAS CONDICIONES	
Los incidentes han sido registrados en el Sistema.	

CASO DE USO	Solucionar Incidente
CU. ID.	ACTIVO_CU_03.
ACTORES	Usuario.
PROPÓSITO	Se validará que el sistema tenga registrados los incidentes para así validar la anomalía en la PC.
RESUMEN	El usuario tendrá un mayor control para validar los problemas que tenga las PC y darle una solución rápida.
MODELO	Principal, Basico.
REFERENCIAS CRUZADAS	R1.1, R1.2
PRECONDICIONES	1. Incidente existente en el programa. 2. Usuario validado en el programa.
CAPACITACION NORMAL DE LOS EVENTOS	
LABOR DEL AUTOR	RESULTADOS DEL PROGRAMA
1. Este caso comienza una vez que el usuario verifica en el programa un registro de incidentes. 3. El Usuario valida la indemnación y procede a ir a solucionar el problema. 4. Una vez que el usuario resuelve el incidente entra al sistema y dará click en botón cerrar incidente.	2. El sistema mostrar detalle del incidente y en qué área y sala se encuentra 5. El sistema procederá con la actualización y el incidente desaparece de la bandeja.

6.	El caso de uso finaliza una vez que el usuario actualiza la página de incidentes y este ya no se encuentre en la bandeja.	
CAPACITACIONES ALTERNAS		
LÍNEA 4	Si el incidente no está solucionado el sistema no permitirá el cambio de estado como resuelto.	
LÍNEA 4	Si se presenta un error al momento de guardar, este debe reflejarse.	
ANTES DE LAS CONDICIONES		
El incidente se arreglado debidamente en el programa.		

4.2.1.1.6. ARQUITECTURA



4.2.1.1.7. REQUERIMIENTOS NO FUNCIONALES

- **Requerimientos de Rendimiento**

Activos Indemáticos, optimizará el manejo de materiales tangibles y las capacidades que rodean los software donde se instalarán las aplicaciones, dando a los beneficiarios respuestas de manera rápida.

Requerimientos de Interfaz de Usuario

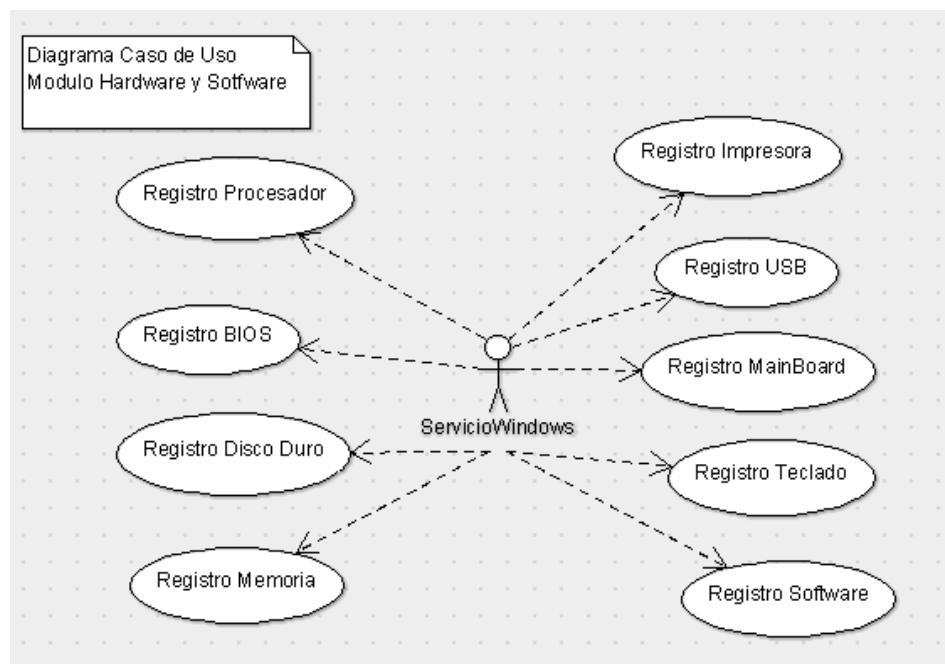
Activos Indemáticos, estos contarán con una interfaz confiable y de buen diseño para la comodidad de los administradores.

- **Requerimientos de Seguridad**

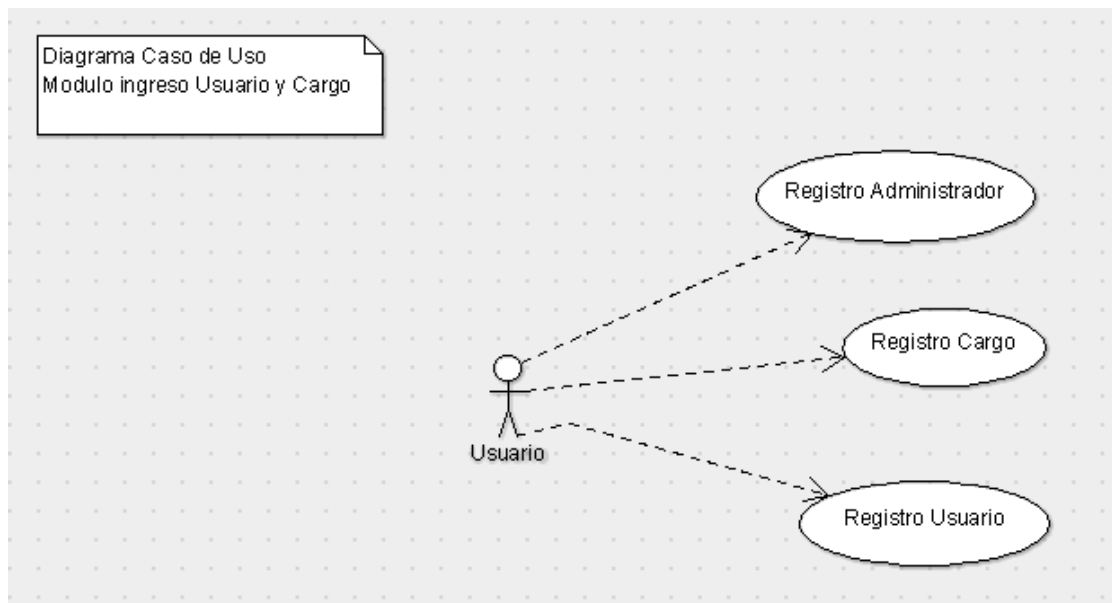
Activos Indemáticos, se implementará en ellos autenticación de administrador y acceso a la base de datos mediante requisitos básicos, presentando un registro de auditoría en cada acción que se realice en el programa.

4.2.1.1.8. Diagramas de Casos de Uso

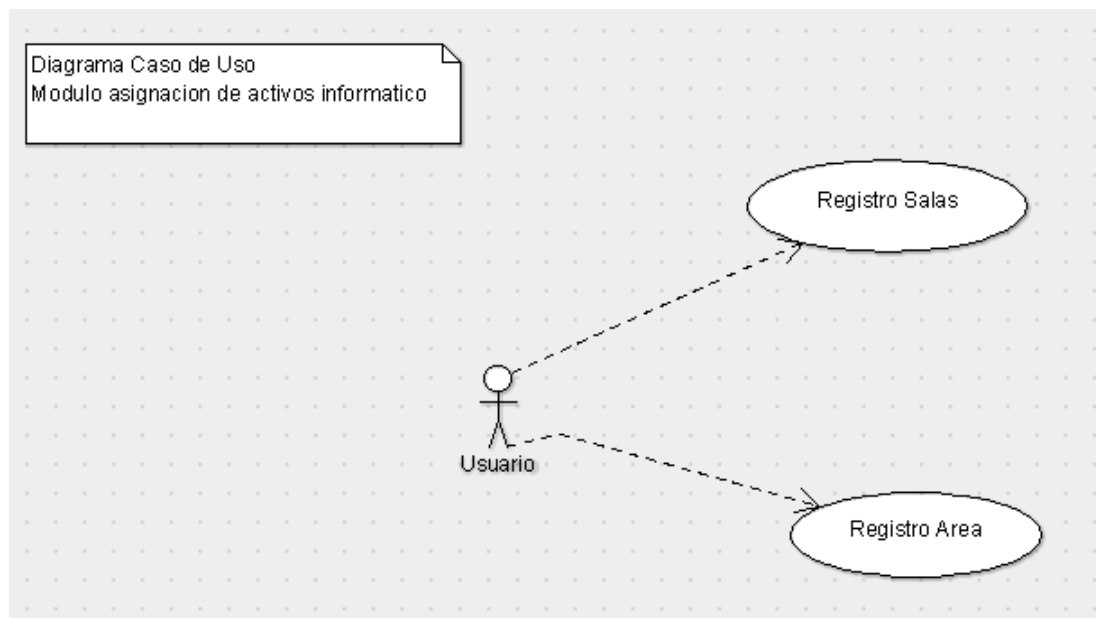
➤ **Módulo Recurso de Hardware y Software**



➤ **Módulo de ingreso de Usuarios**

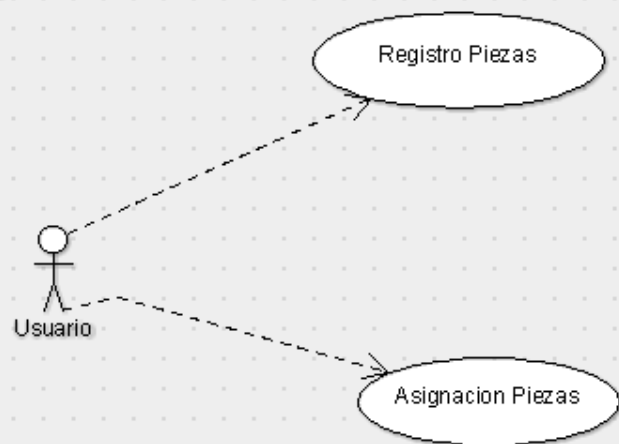


➤ **Módulo de asignación activo indemnático**



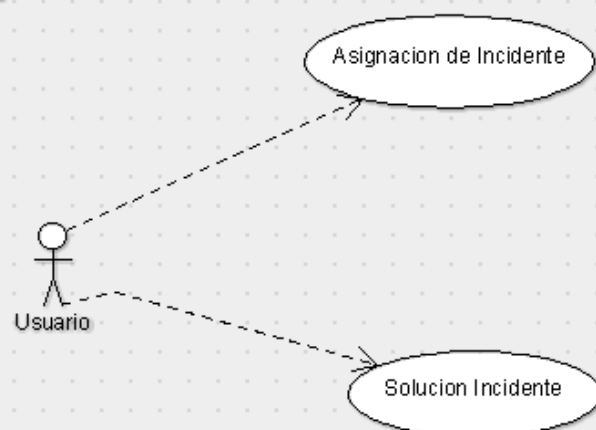
➤ **Módulo de registro de Inventario**

Diagrama Caso de Uso
Modulo Registro de Inventario

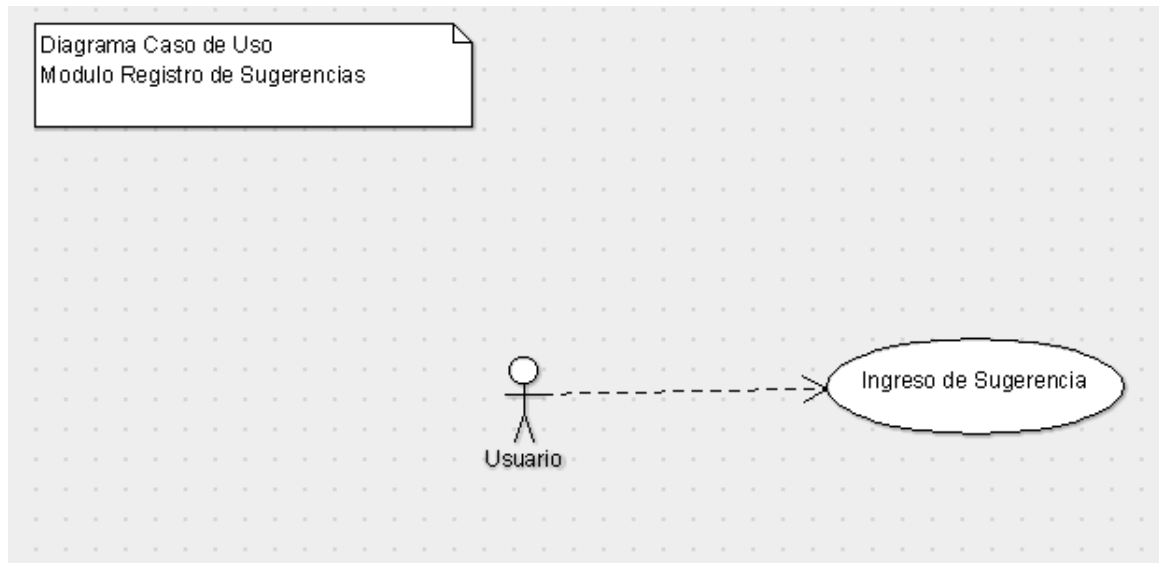


➤ **Módulo de asignación de incidentes**

Diagrama Caso de Uso
Modulo Asignacion de incidentes



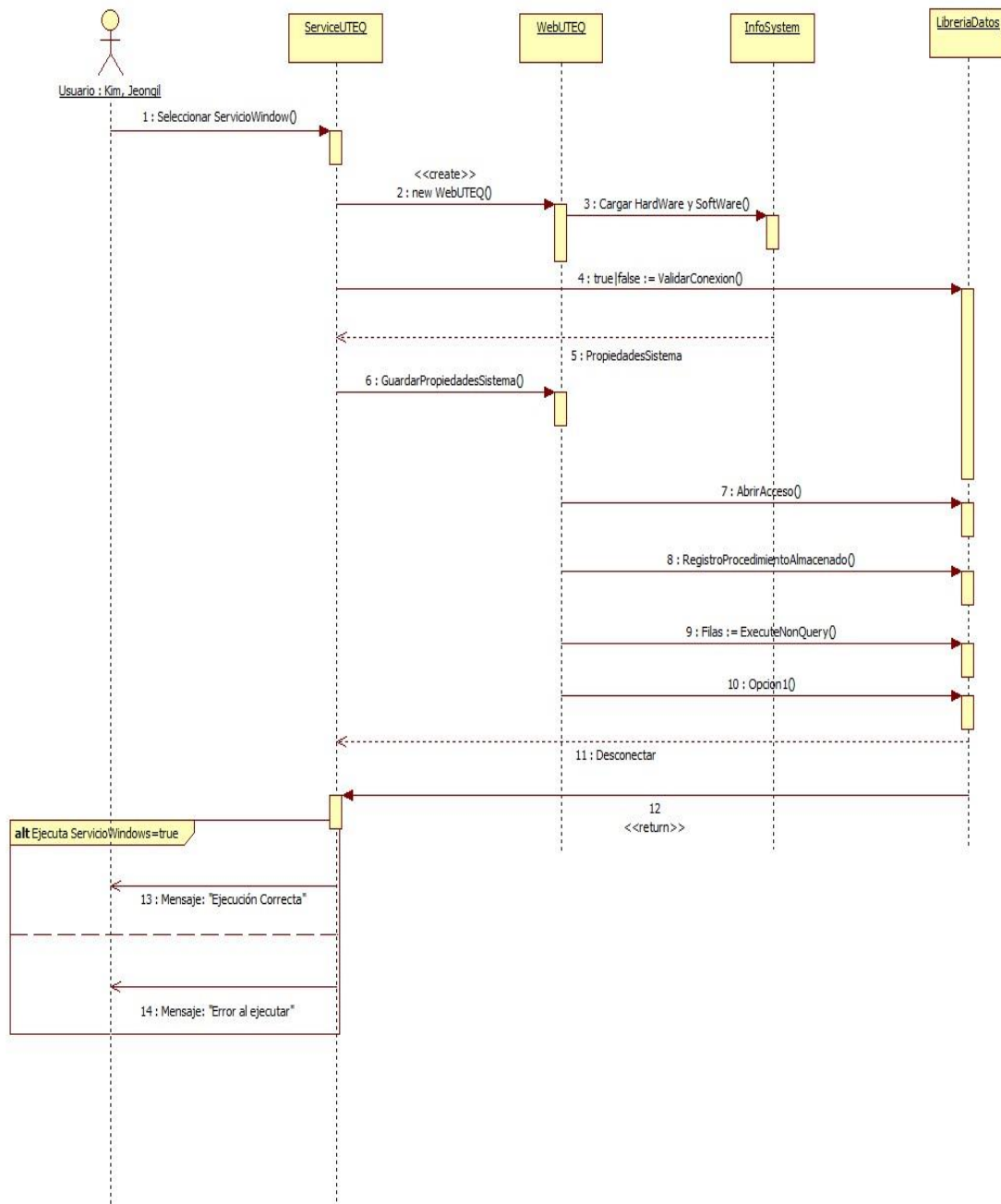
➤ **Módulo de registro de sugerencias**



4.2.1.1.9. Diagramas de Secuencia

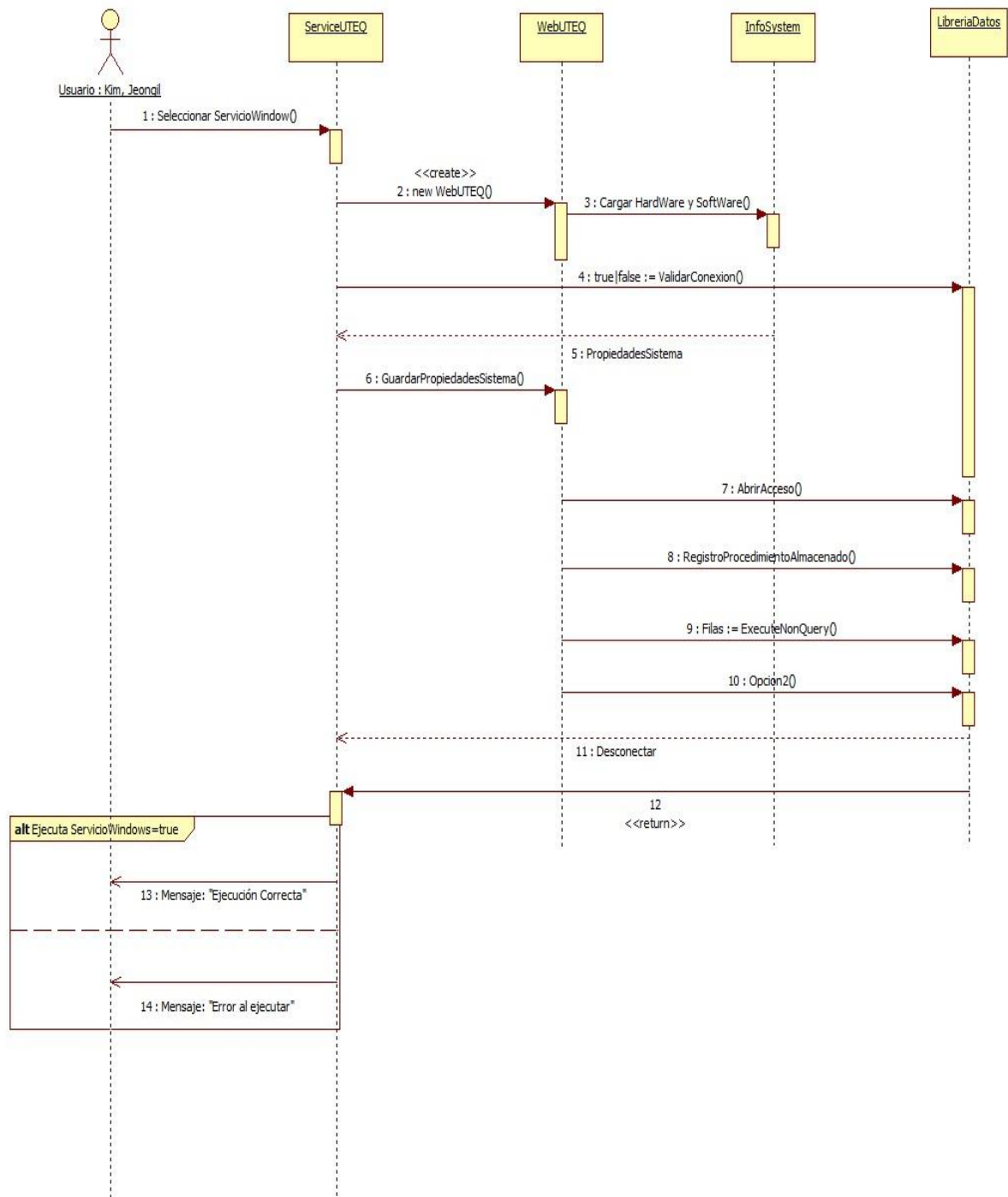
- Tomando en consideración que los procesos a seguir para guardar, y actualizar se repiten en cada uno de los diagramas, se tendrá como referencia el diseño de la entidad Modulo Recurso de Hardware y Software
- **Registro Hardware y Software**

Diagrama Secuencia Registrar Hardware y Software



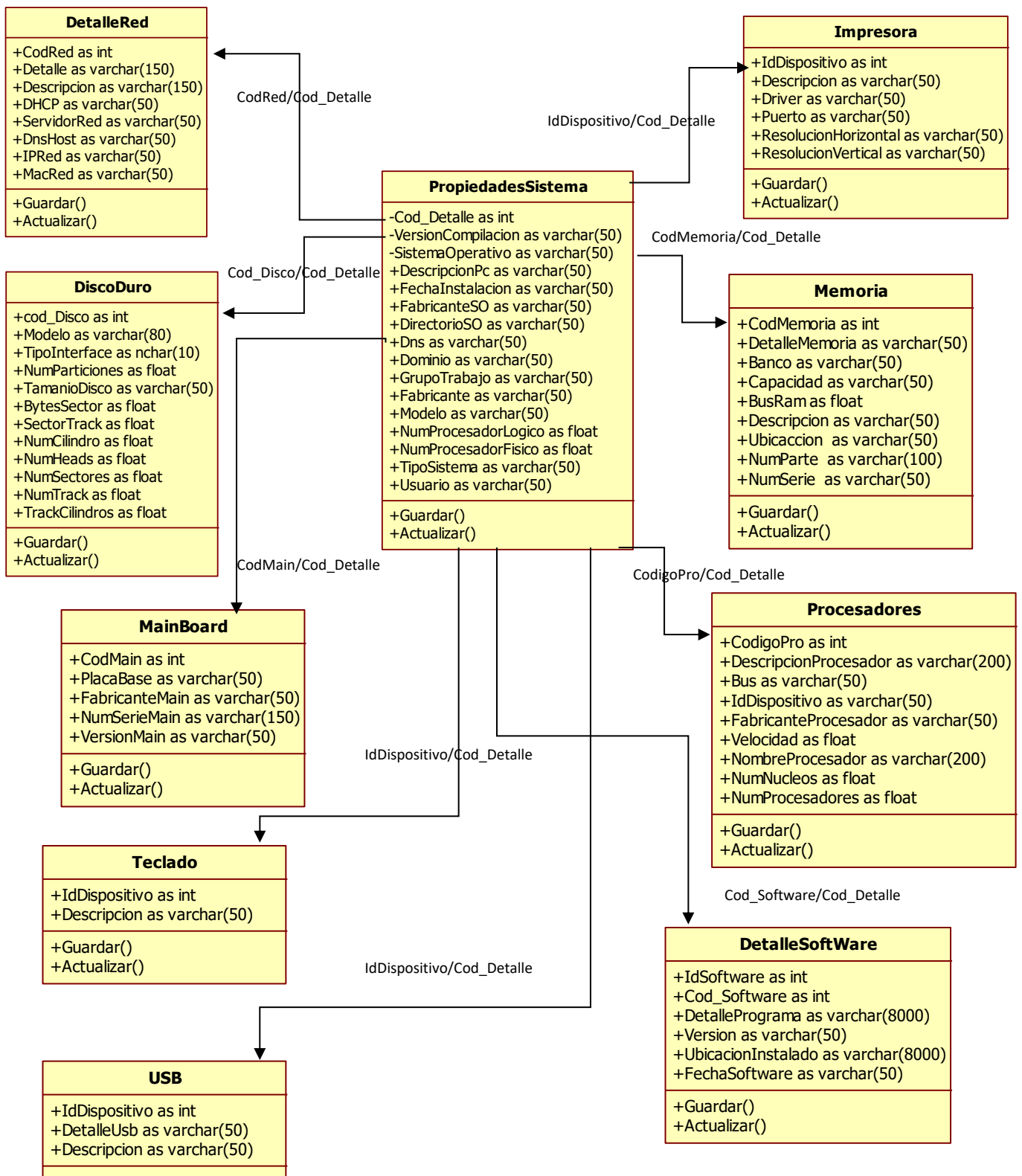
- Actualizar Hardware y Software

Diagrama Secuencia Actualizar Hardware y Software

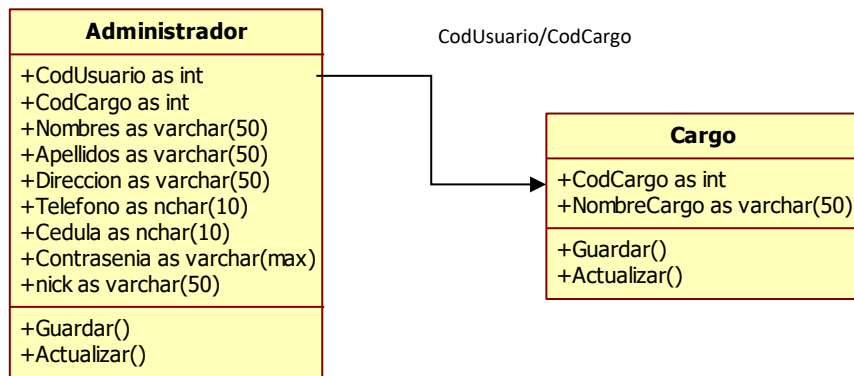


4.2.1.1.10. Diagrama de Clases

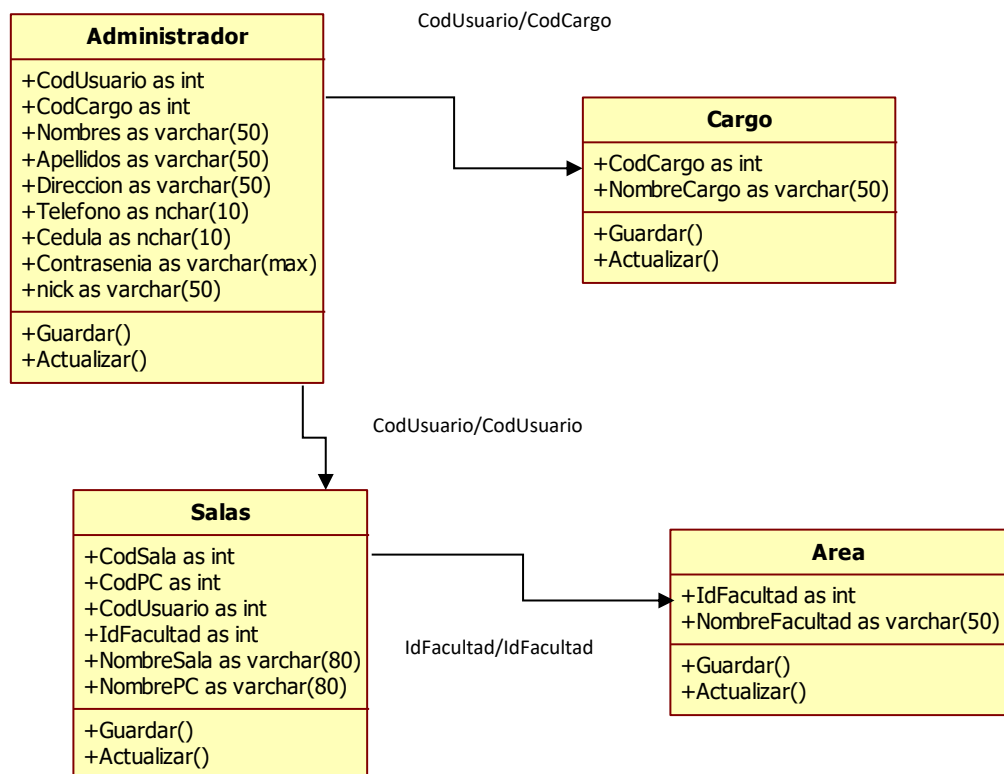
- Modelo Hardware y Software



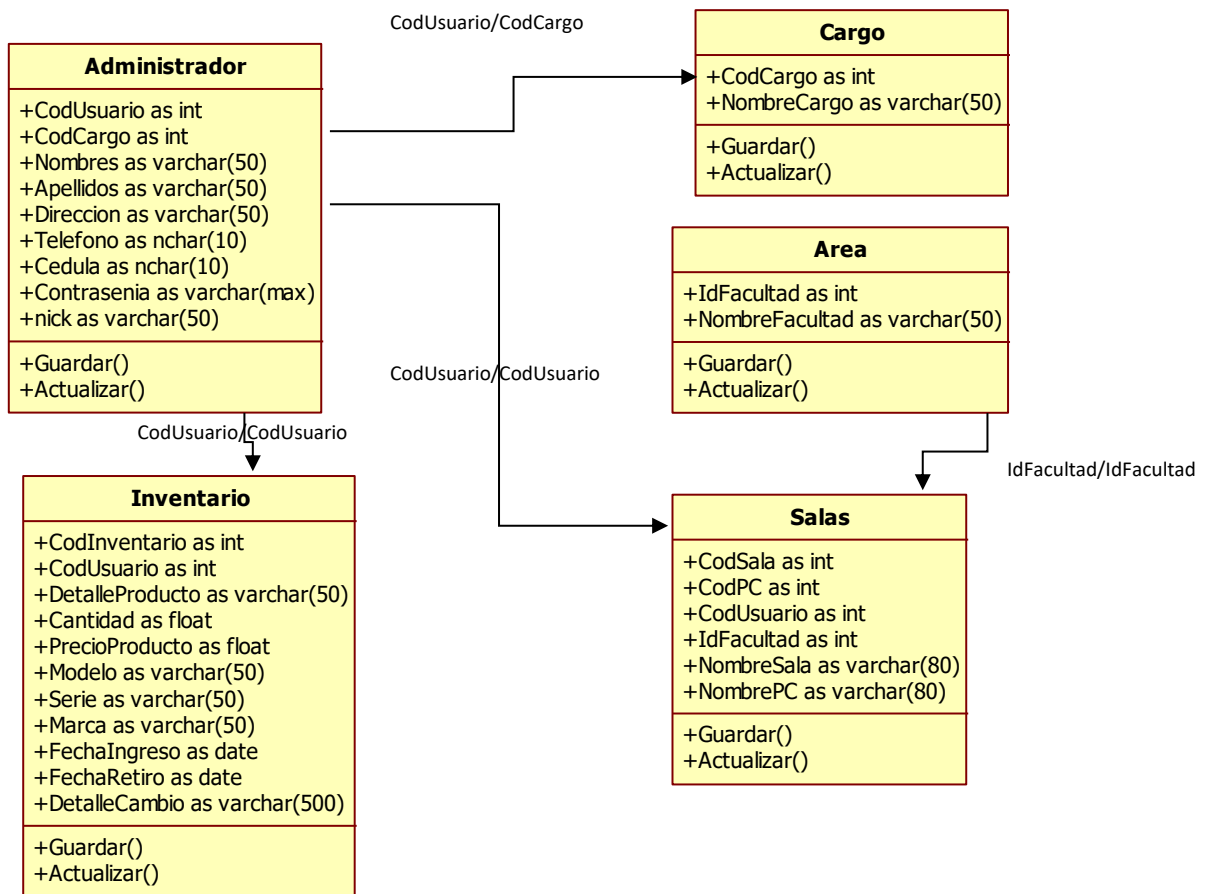
- **Módulo de ingreso de Usuarios**



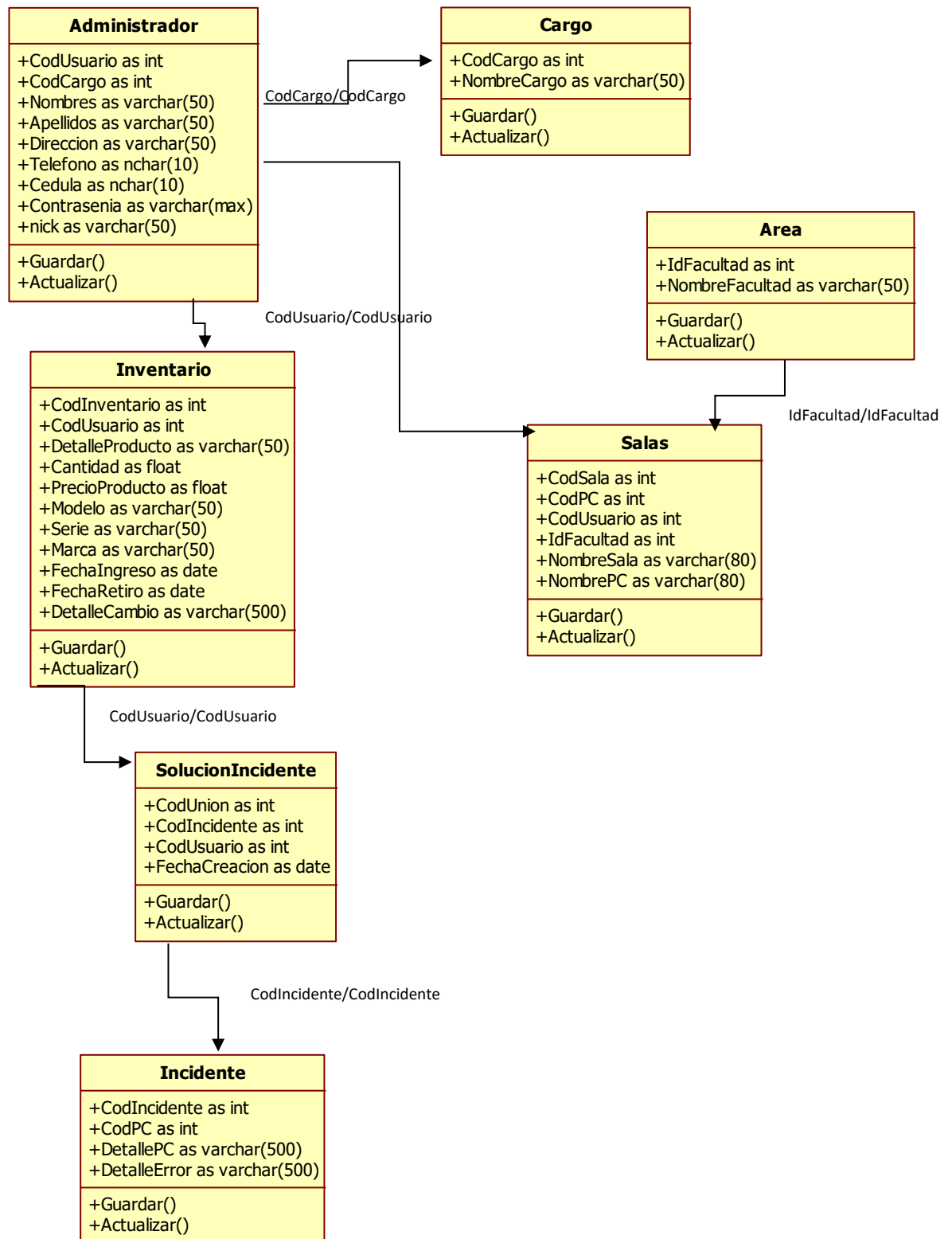
- **Módulo de asignación activo indemnático**



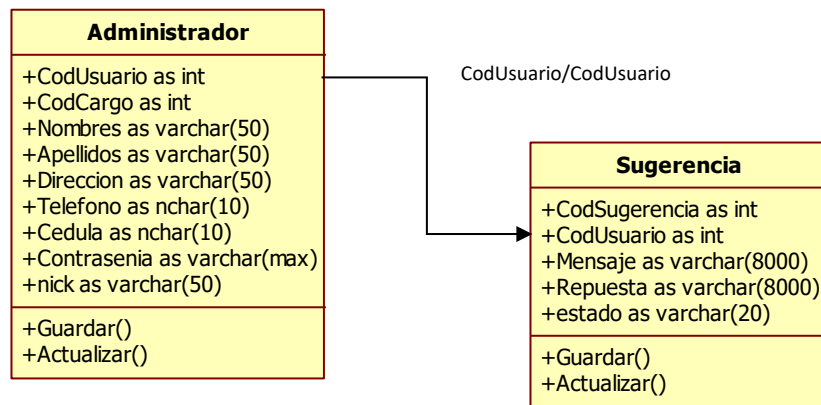
- **Módulo de registro de inventarios**



- **Módulo de asignación de Incidente**



- **Módulo de registro de sugerencias**



4.2.1.2. Diseño de Base de Datos

4.2.2.1.1. Modelo Entidad Relación

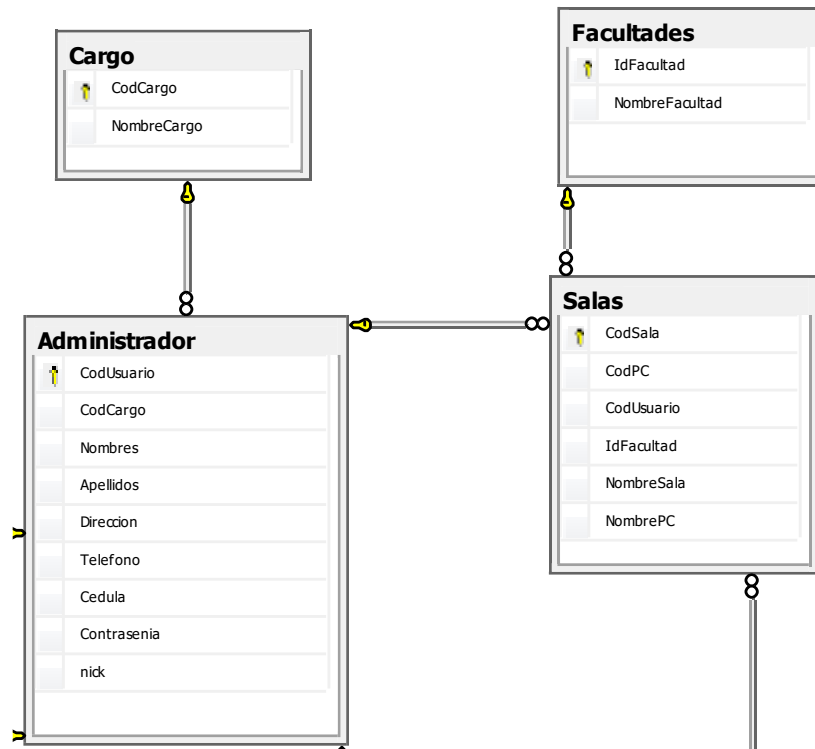
- **Módulo Recurso de Hardware y Software**



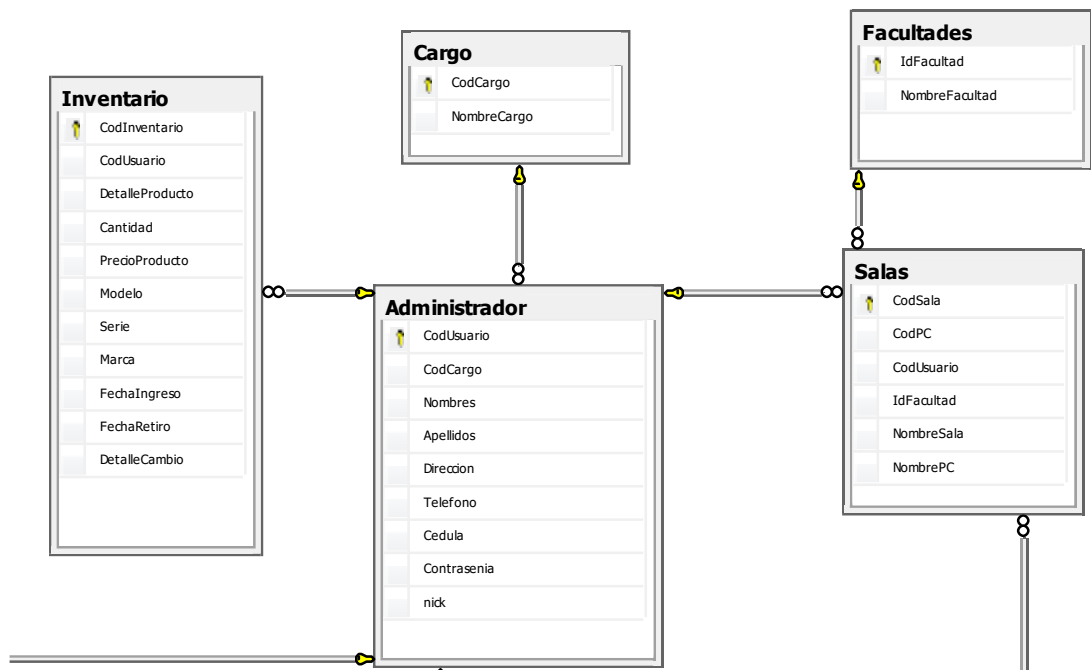
- **Módulo de Usuarios**



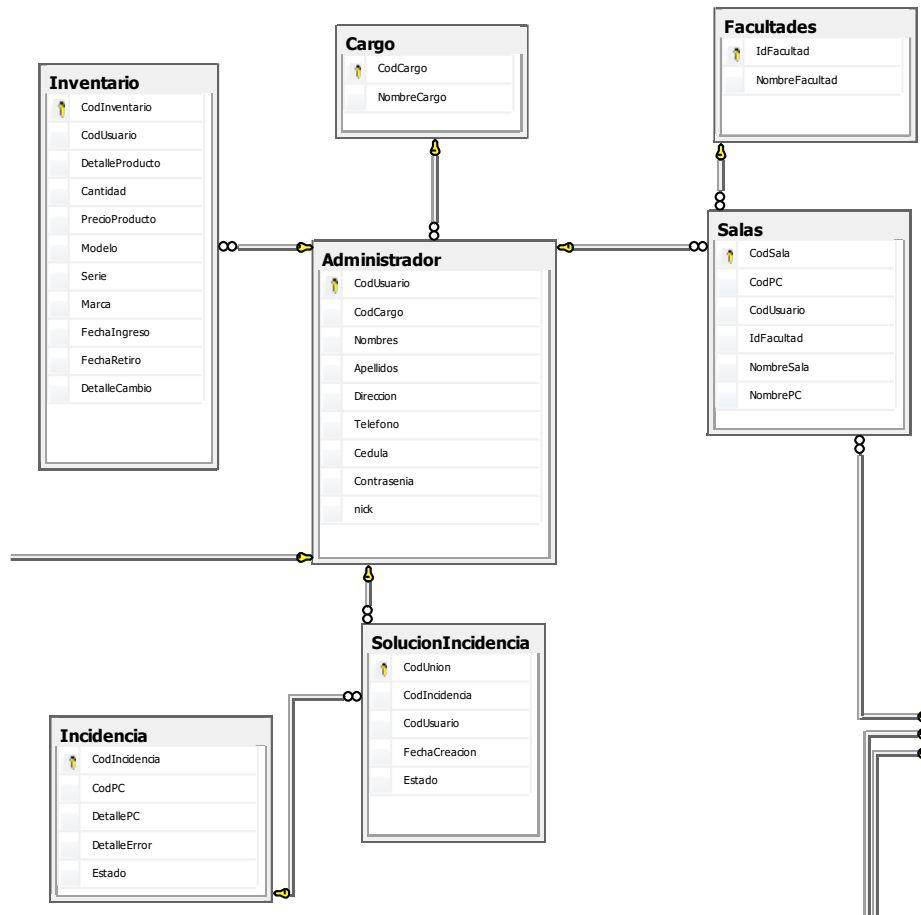
- **Módulo de asignación activo indemnático**



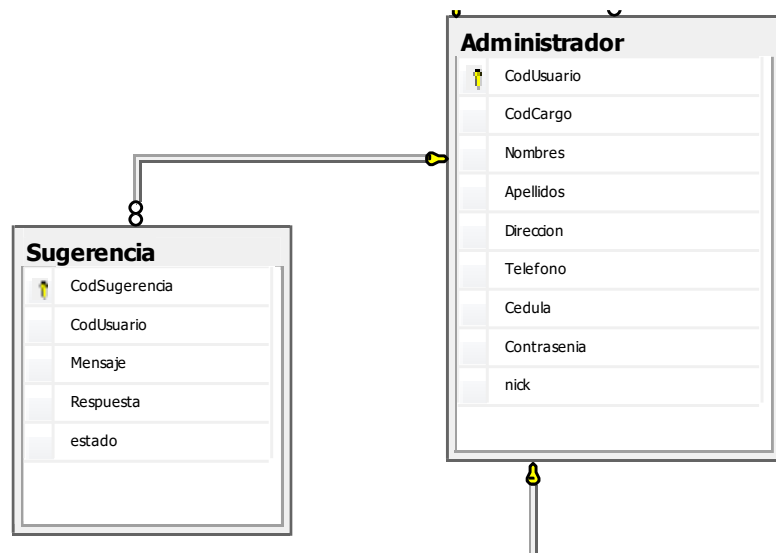
- Módulo de inventarios



- Módulo de incidentes



- Módulo de sugerencias



4.2.2.1.3. Diccionario de Datos

Tabla PropiedadSistema

Columnas

Clave	Columna Nombre	Dominio	Datos Tipo	Not Nula	Unico	Comprobar	Defecto
PK	Cod_Detalle		Entero	SI	NO		
	VersionCompilacion		Cadena(50)	SI	NO		
	SistemaOperativo		Cadena(50)	SI	NO		
	DescripcionPc		Cadena(50)	SI	NO		
	FechaInstalacion		Cadena (50)	SI	NO		
	FabricanteSO		Cadena (50)	SI	NO		
	DirectorioSO		Cadena (50)	SI	NO		
	Dns		Cadena (50)	SI	NO		
	Dominio		Cadena (50)	SI	NO		
	GrupoTrabajo		Cadena (50)	SI	NO		
	Fabricante		Cadena (50)	SI	NO		
	Modelo		Cadena (50)	SI	NO		
	NumProcesadorLogico		Float	SI	NO		
	NumProcesadorFisico		Float	SI	NO		
	TipoSistema		Cadena (50)	SI	NO		
	MemoriaFisica		Cadena(50)	SI	NO		
	Usuario		Cadena(50)	SI	NO		

Relacionandos

Nombre	Valor
Owner	dbo
No clustered PRIMARIO clave	No
PRIMARIO clave fill factor	
Datos espacio de Tabla	PRIMARIO
Datos espacio de PRIMARIO clave	PRIMARIO
Text Filegroup de Tabla (TEXTIMAGE_ON)	

Tabla BIOS

Columnas

Clave	Columna Nombre	Dominio	Datos Tipo	Not Nula	Unico	Comprobar	Defecto
FK	CodBios		Entero	SI	NO		
	DetalleBios		Cadena(50)	SI	NO		
	FabricanteBios		Cadena(100)	SI	NO		
	NumSerieBios		Cadena(150)	SI	NO		
	VersionBios		Cadena (50)	SI	NO		

Relacionandos

Constraint (Relacionando) Nombre	Relacionando Tipo	Parent tabla	Pequeña tabla	Tarjeta.
FK_Cod_Detalle (FK_CodBios)	No-identificado	PropiedadesSistema	Bios	1:N

Comentarios

Registro de Bios referentes a la PropiedadesSistema.

User-defined variables

Nombre	Valor
Owner	dbo
No clustered PRIMARIO clave	No
PRIMARIO clave fill factor	
Datos espacio de Tabla	PRIMARIO
Datos espacio de PRIMARIO clave	PRIMARIO
Text Filegroup de Tabla (TEXTIMAGE_ON)	

Tabla DetalleRed

Columnas

Clave	Columna Nombre	Dominio	Datos Tipo	Not Nula	Unico	Comprobar	Defecto
FK	CodRed		Entero	SI	NO		
	Detalle		Cadena(150)	SI	NO		
	Descripcion		Cadena(150)	SI	NO		
	DHCP		Cadena(50)	SI	NO		
	ServidorRed		Cadena (50)	SI	NO		
	DnsHost		Cadena(50)	SI	NO		
	IPRed		Cadena(50)	SI	NO		
	MacRed		Cadena(50)	SI	NO		

Relacionandos

Constraint (Relacionando) Nombre	Relacionando Tipo	Parent tabla	Pequeña tabla	Tarjeta.
FK_Cod_Detalle (FK_CodRed)	No-identificado	PropiedadesSistema	DetalleRed	1:N

Comentarios

Registro de DetalleRed referentes a la PropiedadesSistema.

User-defined variables

Nombre	Valor
Owner	dbo
No clustered PRIMARIO clave	No
PRIMARIO clave fill factor	

Datos espacio de Tabla	PRIMARIO
Datos espacio de PRIMARIO clave	PRIMARIO
Text Filegroup de Tabla (TEXTIMAGE_ON)	

Tabla DetalleSoftware

Columnas

Clave	Columna Nombre	Dominio	Datos Tipo	Not Nula	Unico	Comprobar	Defecto
PK	IdSoftware		Entero	SI	NO		
FK	CodSoftware		Entero	SI	NO		
	DetallePrograma		Cadena(8000)	SI	NO		
	Version		Cadena(50)	SI	NO		
	UbicacionInstalado		Cadena (8000)	SI	NO		
	FechaSoftware		Cadena(50)	SI	NO		

Relacionandos

Constraint (Relacionando) Nombre	Relacionando Tipo	Parent tabla	Pequeña tabla	Tarjeta.
FK_Cod_Detalle (FK_CodSoftware)	No-identificado	PropiedadesSistema	DetalleSoftware	1:N

Comentarios

Registro de DetalleSoftware referentes a la PropiedadesSistema.

User-defined variables

Nombre	Valor
Owner	dbo
No clustered PRIMARIO clave	No
PRIMARIO clave fill factor	
Datos espacio de Tabla	PRIMARIO
Datos espacio de PRIMARIO clave	PRIMARIO
Text Filegroup de Tabla (TEXTIMAGE_ON)	

Tabla DiscoDuro

Columnas

Clave	Columna Nombre	Dominio	Datos Tipo	Not Nula	Unico	Comprobar	Defecto
PK	Cod_Disco		Entero	SI	NO		
FK	Modelo		Entero	SI	NO		
	TipolInterface		Nchar(10)	SI	NO		
	NumParticiones		Float	SI	NO		
	TamanoDisco		Cadena (50)	SI	NO		
	BytesSector		Cadena(50)	SI	NO		
	SectorTrack		Float	SI	NO		
	NumCilindros		Float	SI	NO		
	NumHeads		Float	SI	NO		
	NumSectores		Float	SI	NO		
	NumTrack		Float	SI	NO		
	TrackCilindros		Float	SI	NO		

Relacionandos

Constraint (Relacionando) Nombre	Relacionando Tipo	Parent tabla	Pequeña tabla	Tarjeta.
FK_Cod_Detalle (FK_CodSoftware)	No-identificado	PropiedadesSistema	DiscoDuro	1:N

Comentarios

Registro de DiscoDuro referentes a la PropiedadesSistema.

User-defined variables

Nombre	Valor
Owner	dbo
No clustered PRIMARIO clave	No
PRIMARIO clave fill factor	
Datos espacio de Tabla	PRIMARIO
Datos espacio de PRIMARIO clave	PRIMARIO
Text Filegroup de Tabla (TEXTIMAGE_ON)	

Tabla DiscoLogico

Columnas

Clave	Columna Nombre	Dominio	Datos Tipo	Not Nula	Unico	Comprobar	Defecto
FK	Modelo		Entero	SI	NO		
	NombreUsuario		Nchar(10)	SI	NO		
	Descripcion		Cadena(50)	SI	NO		
	Comprimido		Nchar(10)	SI	NO		
	Sistemarchivo		Cadena(50)	SI	NO		
	Tamano		Cadena(50)	SI	NO		
	EspacioLibre		Cadena(50)	SI	NO		
	EspacioOcupado		Cadena(50)	SI	NO		

Relacionandos

Constraint (Relacionando) Nombre	Relacionando Tipo	Parent tabla	Pequeña tabla	Tarjeta.
FK_Modelo (FK_Modelo)	No-identificado	DiscoDuro	DiscoLogico	1:N

Comentarios

Registro de DiscoLogico referentes a la DiscoDuro.

User-defined variables

Nombre	Valor
Owner	dbo
No clustered PRIMARIO clave	No
PRIMARIO clave fill factor	
Datos espacio de Tabla	PRIMARIO
Datos espacio de PRIMARIO clave	PRIMARIO
Text Filegroup de Tabla (TEXTIMAGE_ON)	

Tabla Impresora

Columnas

Clave	Columna Nombre	Dominio	Datos Tipo	Not Nula	Unico	Comprobar	Defecto
FK	IdDispositivo		Entero	SI	NO		
	Descripcion		Cadena(50)	SI	NO		
	Driver		Cadena(50)	SI	NO		
	Puerto		Cadena(50)	SI	NO		
	ResolucionHorizontal		Cadena(50)	SI	NO		
	ResolucionVertical		Cadena(50)	SI	NO		

Relacionandos

Constraint (Relacionando) Nombre	Relacionando Tipo	Parent tabla	Pequeña tabla	Tarjeta.
FK_Cod_Detalle (FK_IdDispositivo)	No-identificado	PropiedadesSistema	Impresora	1:N

Comentarios

Registro de Impresora referentes a la PropiedadesSistema.

User-defined variables

Nombre	Valor
Owner	dbo
No clustered PRIMARIO clave	No
PRIMARIO clave fill factor	
Datos espacio de Tabla	PRIMARIO
Datos espacio de PRIMARIO clave	PRIMARIO
Text Filegroup de Tabla (TEXTIMAGE_ON)	

Tabla MainBoard

Columnas

Clave	Columna Nombre	Dominio	Datos Tipo	Not Nula	Unico	Comprobar	Defecto
FK	CodMain		Entero	SI	NO		
	PlacaBase		Cadena(50)	SI	NO		
	FabricanteMain		Cadena(50)	SI	NO		
	NumSerieMain		Cadena(150)	SI	NO		
	VersionMain		Cadena(50)	SI	NO		

Relacionandos

Constraint (Relacionando) Nombre	Relacionando Tipo	Parent tabla	Pequeña tabla	Tarjeta.
FK_Cod_Detalle (FK_CodMain)	No-identificado	PropiedadesSistema	MainBoard	1:N

Comentarios

Registro de MainBoard referentes a la PropiedadesSistema.

User-defined variables

Nombre	Valor
Owner	dbo
No clustered PRIMARIO clave	No
PRIMARIO clave fill factor	
Datos espacio de Tabla	PRIMARIO
Datos espacio de PRIMARIO clave	PRIMARIO
Text Filegroup de Tabla (TEXTIMAGE_ON)	

Tabla Memoria

Columnas

Clave	Columna Nombre	Dominio	Datos Tipo	Not Nula	Unico	Comprobar	Defecto
FK	CodMemoria		Entero	SI	NO		
	DetalleMemoria		Cadena(50)	SI	NO		
	Banco		Cadena(50)	SI	NO		
	Capacidad		Cadena(50)	SI	NO		
	BusRam		Float	SI	NO		
	Descripcion		Cadena(100)	SI	NO		
	Ubicación		Cadena(50)	SI	NO		
	NumParte		Cadena(100)	SI	NO		
	NumSerie		Cadena(50)	SI	NO		

Relacionandos

Constraint (Relacionando) Nombre	Relacionando Tipo	Parent tabla	Pequeña tabla	Tarjeta.
FK_Cod_Detalle (FK_CodMemoria)	No-identificado	PropiedadesSistema	Memoria	1:N

Comentarios

Registro de Memoria referentes a la PropiedadesSistema.

User-defined variables

Nombre	Valor
Owner	dbo
No clustered PRIMARIO clave	No
PRIMARIO clave fill factor	
Datos espacio de Tabla	PRIMARIO
Datos espacio de PRIMARIO clave	PRIMARIO
Text Filegroup de Tabla (TEXTIMAGE_ON)	

Tabla Procesadores

Columnas

Clave	Columna Nombre	Dominio	Datos Tipo	Not Nula	Unico	Comprobar	Defecto
FK	CodigoPro		Entero	SI	NO		
	DescripcionProcesador		Cadena(200)	SI	NO		
	Bus		Cadena(50)	SI	NO		
	IdDispositivo		Cadena(50)	SI	NO		
	FabricanteProcesador		Cadena(50)	SI	NO		
	Velocidad		Float	SI	NO		
	NombreProcesador		Cadena(200)	SI	NO		
	NumNucleos		Float	SI	NO		
	NumProcesadoreLogico		Float	SI	NO		

Relacionandos

Constraint (Relacionando) Nombre	Relacionando Tipo	Parent tabla	Pequeña tabla	Tarjeta.
FK_Cod_Detalle (FK_CodigoPro)	No-identificado	PropiedadesSistema	Procesadores	1:N

Comentarios

Registro de Procesador referentes a la PropiedadesSistema.

User-defined variables

Nombre	Valor
Owner	dbo
No clustered PRIMARIO clave	No
PRIMARIO clave fill factor	

Datos espacio de Tabla	PRIMARIO
Datos espacio de PRIMARIO clave	PRIMARIO
Text Filegroup de Tabla (TEXTIMAGE_ON)	

Tabla Teclado

Columnas

Clave	Columna Nombre	Dominio	Datos Tipo	Not Nula	Unico	Comprobar	Defecto
FK	IdDispositivo		Entero	SI	NO		
	Descripcion		Cadena(50)	SI	NO		

Relacionandos

Constraint (Relacionando) Nombre	Relacionando Tipo	Parent tabla	Pequeña tabla	Tarjeta.
FK_Cod_Detalle (FK_IdDispositivo)	No-identificado	PropiedadesSistema	Teclado	1:N

Comentarios

Registro de Teclado referentes a la PropiedadesSistema.

User-defined variables

Nombre	Valor
Owner	dbo
No clustered PRIMARIO clave	No
PRIMARIO clave fill factor	
Datos espacio de Tabla	PRIMARIO
Datos espacio de PRIMARIO clave	PRIMARIO
Text Filegroup de Tabla (TEXTIMAGE_ON)	

Tabla USB

Columnas

Clave	Columna Nombre	Dominio	Datos Tipo	Not Nula	Unico	Comprobar	Defecto
FK	IdDispositivo		Entero	SI	NO		
	DetalleUsb		Cadena(50)	SI	NO		
	Descripcion		Cadena(50)	SI	NO		

Relacionandos

Constraint (Relacionando) Nombre	Relacionando Tipo	Parent tabla	Pequeña tabla	Tarjeta.
FK_Cod_Detalle (FK_IdDispositivo)	No-identificado	PropiedadesSistema	USB	1:N

Comentarios

Registro de USB referentes a la PropiedadesSistema.

User-defined variables

Nombre	Valor
Owner	dbo
No clustered PRIMARIO clave	No
PRIMARIO clave fill factor	
Datos espacio de Tabla	PRIMARIO
Datos espacio de PRIMARIO clave	PRIMARIO
Text Filegroup de Tabla (TEXTIMAGE_ON)	

Tabla Administrador

Columnas

Clave	Columna Nombre	Dominio	Datos Tipo	Not Nula	Unico	Comprobar	Defecto
PK	CodUsuario		Entero	SI	NO		
FK	CodCargo		entero	SI	NO		
	Nombres		Cadena(50)	SI	NO		
	Apellidos		Cadena(50)	SI	NO		
	Direccion		Cadena(150)	SI	NO		
	Telefono		Nchar(10)	SI	NO		
	Cedula		Nchar(10)	SI	NO		
	Contrasenia		Cadena(max)	SI	NO		
	Nick		Cadena(50)	SI	NO		

Relacionandos

Constraint (Relacionando) Nombre	Relacionando Tipo	Parent tabla	Pequeña tabla	Tarjeta.
FK_Cod_Detalle (FK_CodCargo)	No-identificado	PropiedadesSistema	Cargo	1:N

Comentarios

Registro de Administrador referentes a la Cargo.

User-defined variables

Nombre	Valor
Owner	dbo
No clustered PRIMARIO clave	No
PRIMARIO clave fill factor	
Datos espacio de Tabla	PRIMARIO
Datos espacio de PRIMARIO clave	PRIMARIO
Text Filegroup de Tabla (TEXTIMAGE_ON)	

Tabla Auditoria

Columnas

Clave	Columna Nombre	Dominio	Datos Tipo	Not Nula	Unico	Comprobar	Defecto
	Fecha		Date	SI	NO		
	Hora		Time(7)	SI	NO		
	UsuarioSql		Cadena (50)	SI	NO		
	UsuarioApp		Cadena (15)	SI	NO		
	PC		Cadena (50)	SI	NO		
	Aplicacion		Cadena (200)	SI	NO		
	Accion		Cadena (50)	SI	NO		
	Descripcion		Cadena (8000)	SI	NO		

User-defined variables

Nombre	Valor
Owner	dbo
No clustered PRIMARIO clave	No
PRIMARIO clave fill factor	
Datos espacio de Tabla	PRIMARIO
Datos espacio de PRIMARIO clave	
Text Filegroup de Tabla (TEXTIMAGE_ON)	PRIMARIO

Tabla Cargo

Columnas

Clave	Columna Nombre	Dominio	Datos Tipo	Not Nula	Unico	Comprobar	Defecto
PK	CodCargo		Entero	SI	NO		
	NombreCargo		Cadena (50)	SI	NO		

User-defined variables

Nombre	Valor
Owner	dbo
No clustered PRIMARIO clave	No
PRIMARIO clave fill factor	
Datos espacio de Tabla	PRIMARIO
Datos espacio de PRIMARIO clave	
Text Filegroup de Tabla (TEXTIMAGE_ON)	PRIMARIO

Tabla Facultad

Columnas

Clave	Columna Nombre	Dominio	Datos Tipo	Not Nula	Unico	Comprobar	Defecto
PK	IdFacultad		entero	SI	NO		
	NombreFacultad		Cadena (50)	SI	NO		

User-defined variables

Nombre	Valor
Owner	dbo
No clustered PRIMARIO clave	No
PRIMARIO clave fill factor	
Datos espacio de Tabla	PRIMARIO
Datos espacio de PRIMARIO clave	
Text Filegroup de Tabla (TEXTIMAGE_ON)	PRIMARIO

Tabla Salas

Columnas

Clave	Columna Nombre	Dominio	Datos Tipo	Not Nula	Unico	Comprobar	Defecto
PK	CodSala		entero	SI	NO		
FK	CodPC		Entero	SI	NO		
FK	CodUsuario		Entero	SI	NO		
FK	IdFacultad		Entero	SI	NO		
	NombreSala		Cadena(80)	SI	NO		
	NombrePC		Cadena(80)	SI	NO		

Relacionandos

Constraint (Relacionando) Nombre	Relacionando Tipo	Parent tabla	Pequeña tabla	Tarjeta.
FK_Cod_Usuario (FK_CodPC)	No-identificado	Administrador	Salas	1:N
FK_Cod_Detalle (FK_CodPC)	No-identificado	PropiedadesSistema	Salas	1:N
FK_IdFacultad (FK_CodPC)	No-identificado	Facultad	Salas	1:N

Comentarios

Catálogo de bases de datos.

User-defined variables

Nombre	Valor
Owner	dbo
No clustered PRIMARIO clave	No

PRIMARIO clave fill factor	
Datos espacio de Tabla	PRIMARIO
Datos espacio de PRIMARIO clave	PRIMARIO
Text Filegroup de Tabla (TEXTIMAGE_ON)	

Tabla HistoriaPc

Columnas

Clave	Columna Nombre	Dominio	Datos Tipo	Not Nula	Unico	Comprobar	Defecto
PK	CodHistorial		Entero	SI	NO		
FK	CodPC		Entero	SI	NO		
	Accion		Cadena(50)	SI	NO		
	detalleRegistro		Cadena(500)	SI	NO		

Relacionandos

Constraint (Relacionando) Nombre	Relacionando Tipo	Parent tabla	Pequeña tabla	Tarjeta.
FK_Cod_Detalle (FK_CodPC)	No-identificado	PropiedadesSistema	HistorialPC	1:N

Comentarios

Registro de USB referentes a la PropiedadesSistema.

User-defined variables

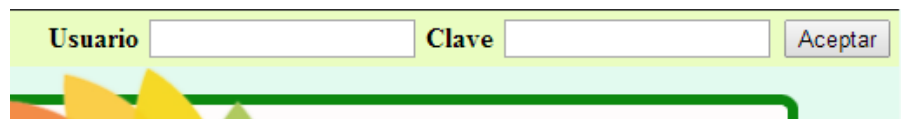
Nombre	Valor
Owner	dbo
No clustered PRIMARIO clave	No
PRIMARIO clave fill factor	
Datos espacio de Tabla	PRIMARIO
Datos espacio de PRIMARIO clave	PRIMARIO
Text Filegroup de Tabla (TEXTIMAGE_ON)	

4.2.2. Diseño

En esta fase se realiza el diseño de la interfaz gráfica de usuario (Activo Indemático) y el diseño de la Base de Datos.



Inicio de Sesión



Menú Principal



Registro de Usuario Administrador y Custodio que es la personas encargada de cada sala y PC que se encuentre en ella.



INICIO	OBJETIVOS	REGISTRO	MISSION Y VISION	SUGERENCIAS	CONTACTOS
---------------	------------------	-----------------	-------------------------	--------------------	------------------

Usuario Clave

REGISTRO DE USUARIO

NOMBRES

APELLIDOS

DIRECCION

CEDULA

TELEFONO

NICK

CLAVE

NUEVA

CARGO

© Universidad Técnica Estatal Quevedo - Activo Informático UTEQ

Menú del Usuario Administrador



INICIO	OBJETIVOS	REGISTRO	MISSION Y VISION	SUGERENCIAS	CONTACTOS
---------------	------------------	-----------------	-------------------------	--------------------	------------------

Usuario Guillermo Armando Campuzano Arechua



Lista PC



Incidentes



Registro Salas



Inventario



Ver Sugerencias



Detalle Inventario



Reportes

© Universidad Técnica Estatal Quevedo - Activo Informático UTEQ



Lista PC en esta página se encuentran todos los equipos indemáticos registrado automáticamente por el servicio Windows.

Inventario Activos Informaticos

UTEQ - UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

ACREDITADA

INICIO | OBJETIVOS | **REGISTRO** | MISION Y VISION | SUGERENCIAS | CONTACTOS

Usuario Guillermo Armando Campuzano Arechua [Cerrar]

EQUIPOS
GUILLERMO-PC
GISELA-PC
MARYLISS
LUISMASTER
USUARIO-PC

PROPIEDADES DEL SISTEMA
PROCESADOR
MEMORIA
MAINBOARD
BIOS
DISCO
DISCO LOGICOS
TECLADO
IMPRESORA
USB

DESCRIPCION	DETALLE
DetalleUsb	Dispositivo de almacenamiento USB
Descripcion	Dispositivo de almacenamiento USB

[AsignarPC]

© Universidad Técnica Estatal Quevedo - Activo Informático UTEQ

En esta página se asigna las PC a un custodio con su respectiva sala y Facultad.

Inventario Activos Informaticos

UTEQ - UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

ACREDITADA

INICIO | OBJETIVOS | **REGISTRO** | MISION Y VISION | SUGERENCIAS | CONTACTOS

Usuario Guillermo Armando Campuzano Arechua [Cerrar]

ASIGNACION DE EQUIPOS

FACULTAD:

USUARIO:

NOMBRE SALA:

NOMBRE PC:

[Guardar]

© Universidad Técnica Estatal Quevedo - Activo Informático UTEQ



En esta página se visualizan los incidentes de cada PC que son enviadas por el Servicio Windows.

Inventario Activos Informaticos

Inicio OBJETIVOS REGISTRO MISION Y VISION SUGERENCIAS CONTACTOS

Usuario Clave

DETALLE DE INCIDENTES

DESCRIPCION

CodPC	NOMBRE PC	DETALLE ERROR
5	USUARIO-PC	Problema en el disco duro

© Universidad Técnica Estatal Quevedo - Activo Informático UTEQ



En página se realiza el registro de cada una de la facultad.

Inventario Activos Informaticos

Inicio OBJETIVOS REGISTRO MISION Y VISION SUGERENCIAS CONTACTOS

Usuario Guillermo Armando Campuzano Arechua

REGISTRO DE AREA

DESCRIPCION

© Universidad Técnica Estatal Quevedo - Activo Informático UTEQ



En esta página se realiza el registro accesorios que tendrá en inventario área de mantenimiento.

Inventario Activos Informaticos

UTEQ UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO edu.ec ACREDITADA

[INICIO](#) [OBJETIVOS](#) [REGISTRO](#) [MISION Y VISION](#) [SUGERENCIAS](#) [CONTACTOS](#)

Usuario Guillermo Armando Campuzano Arechua [Cerrar](#)

REGISTRO DE EQUIPOS

DESCRIPCION

CANTIDAD

PRECIO

MODELO

SERIE

MARCA

FECHA

[Guardar](#)

© Universidad Técnica Estatal Quevedo - Activo Informático UTEQ



En esta página se verá la sugerencia que serán enviadas por las personas encargadas de cada sala

Inventario Activos Informaticos

UTEQ UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO edu.ec ACREDITADA

[INICIO](#) [OBJETIVOS](#) [REGISTRO](#) [MISION Y VISION](#) [SUGERENCIAS](#) [CONTACTOS](#)

Usuario Guillermo Armando Campuzano Arechua [Cerrar](#)

DETALLE DE SUGERENCIA

DESCRIPCION [Buscar](#)

© Universidad Técnica Estatal Quevedo - Activo Informático UTEQ

Detalle de sugerencias



Inventario Activos Informaticos



UTEQ
UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO
educ.ec
ACREDITADA

INICIO

OBJETIVOS

REGISTRO

MISSION Y VISION

SUGERENCIAS

CONTACTOS

Usuario Guillermo Armando Campuzano Arechua

Cerrar

DETALLE DE SUGERENCIA

DESCRIPCION

Carmen Lucia Perez Pino

Buscar

	CODIGO	MENSAJE	RESPUESTA	ESTADO
Seleccionar	4	debe compra una nueva pc para la sala 5	Vacio	Enviado

© Universidad Técnica Estatal Quevedo - Activo Informático UTEQ

Contestación de sugerencia.



Inventario Activos Informaticos



UTEQ
UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO
educ.ec
ACREDITADA

INICIO

OBJETIVOS

REGISTRO

MISSION Y VISION

SUGERENCIAS

CONTACTOS

Usuario Guillermo Armando Campuzano Arechua

Cerrar

REVISION SUGERENCIA

COMENTARIO

debe compra una nueva pc para la sala 5

RESPUESTA

Guardar

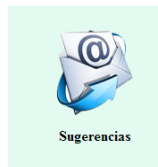
© Universidad Técnica Estatal Quevedo - Activo Informático UTEQ

- **DETALLE DE USUARIO**

Menú de usuario.



Se realiza actualización de Usuario Administrador y Custodio que es la personas encargada de cada sala y PC que se encuentre en ella.



En esta página se registra sugerencias que serán enviadas al administrador

Inventario
Activos Informaticos

UTEQ
UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO
ACREDITADA

INICIO OBJETIVOS REGISTRO MISION Y VISION SUGERENCIAS CONTACTOS

Usuario Clave

REGISTRO SUGERENCIA

USUARIO

DESCRIPCIÓN

© Universidad Técnica Estatal Quevedo - Activo Informático UTEQ

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 PRUEBAS Y RESULTADOS

5.1.1. Pruebas de Ejecución

Para obtener los resultados esperados del sistema **ACTIVO INDEMÁTICO** se instaló el servicio Windows **ServiceUTEQ** en las PC del instituto de Indemática y se extrajo toda la indemnación de sus características de Hardware y Software, las verificaciones de calidad y desarrollo del programa se hicieron de la siguiente manera:

- **Para acceder al programa**

Se realizan pruebas utilizando el usuario GACAMPUZ. Se verifica el acceso rápido al sistema. Cuando el administrador inicie sesión, se analiza la firma de auditoría en nuestra base de datos.





- **Registro de Usuario (Administrador/Custodio)**

Se realiza el registro del Usuario con su respectivo cargo que va desempeñar dentro de la institución. Se valida registro de auditoría en la base de datos.

Inventario Activos Informaticos

UTEQ UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO .edu.ec ACREDITADA

INICIO OBJETIVOS REGISTRO MISION Y VISION SUGERENCIAS CONTACTOS

Usuario Clave

REGISTRO DE USUARIO

NOMBRES

APELLIDOS

DIRECCION

CEDULA

TELEFONO

NICK

CLAVE

NUEVA

CARGO

© Universidad Técnica Estatal Quevedo - Activo Informático UTEQ

Se realiza una actualización del registro del Usuario, se obtiene un tiempo de respuesta de actualización inferior a un segundo. Se verifica la firma de auditoría nuestra base de datos.

Inventario Activos Informaticos

UTEQ UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO .edu.ec ACREDITADA

INICIO OBJETIVOS REGISTRO MISION Y VISION SUGERENCIAS CONTACTOS

Usuario Carmen Lucia Perez Pino

ACTUALIZAR DE USUARIO

NOMBRES

APELLIDOS

DIRECCION

TELEFONO

NICK

Anterior Clave

Nueva Clave

© Universidad Técnica Estatal Quevedo - Activo Informático UTEQ

Se analiza el sistema de demás efectiva para la asignación de los equipos su respectivo custodio. El tiempo de respuesta será de un segundo.

Inventario Activos Informaticos

UTEQ UNIVERSIDAD TECNICA ESTATAL DE QUEVEDO .edu.ec ACREDITADA

INICIO OBJETIVOS REGISTRO MISION Y VISION SUGERENCIAS CONTACTOS

Usuario Guillermo Armando Campuzano Arechua Cerrar

ASIGNACION DE EQUIPOS

FACULTAD: FACULTAD DE INGENIERIA

USUARIO: Carmen Lucia Perez Pino

NOMBRE SALA:

NOMBRE PC: GUILLERMO-PC

Guardar

© Universidad Técnica Estatal Quevedo - Activo Informático UTEQ

Además el proceso de revisión de los incidentes es de demàs automática porque mediante el Servicio Windows se registra los problemas que tenga la PC. En menos de un segundo se obtendrá la respuesta.

Inventario Activos Informaticos

UTEQ UNIVERSIDAD TECNICA ESTATAL DE QUEVEDO .edu.ec ACREDITADA

INICIO OBJETIVOS REGISTRO MISION Y VISION SUGERENCIAS CONTACTOS

Usuario Guillermo Armando Campuzano Arechua Cerrar

DETALLE DE INCIDENTES

DESCRIPCION: [] Buscar

CodPC	NOMBRE PC	DETALLE ERROR
5	USUARIO-PC	Problema en el disco duro

© Universidad Técnica Estatal Quevedo - Activo Informático UTEQ

5.1.2. Los Hardware Utilizados para las Pruebas

En el desarrollo de validaciones se usó una etapa de desarrollo que cumple las siguientes fases:

- CPU: Intel Corei3 2.1 GHz.
- RAM: 4 Gb.
- HDD: 300 Gb.

El servidor de cálculos para la validación esta estructurado de la siguiente manera:

- CPU: Intel Corei5 3.10GHz.
- RAM: 4 Gb.
- HDD: 500 Gb.

5.2. RESULTADO DE LA COMPROBACIÓN DE HIPÓTESIS

5.2.1. Variable Independiente

5.2.1.1. Software

Un programa de **ACTIVOS INDEMÁTICO**, a través de su interfaz confiable y rápida proporciona garantía y dominio rápido al administrador, logrando que este se adapte rápidamente a su manejo.

Los proceso de registro y actualización se realizaran en un segundo, reduciendo el tiempo invertido en cada actividad personal técnico y administrativo del área de mantenimiento.

El sistema **ACTIVO INDEMÀSTICO** mediante la instalación del Servicio Windows (ServiceUTEQ) permitirá extraer todas las características de Hardware y Software permitiendo realizar la una revisión rápida y eficiente al personal técnico, aumentado la labor general del administrador, haciendo menos el tiempo de resultados. Cada factor antes dicho influye en el desarrollo general.

5.2.2. Variable Dependiente

5.2.2.1. Eficiencia

El programa de **ACTIVO INDEMÁTICO**, en un tiempo determinado en sector de mantenimiento de los equipos indemático, desmosto ser una aplicación eficiente para el personal técnico y administrativo aumento su productividad durante el mes que se realizaron las pruebas

5.2.2.2. Eficacia

El uso del sistema **ACTIVO INDEMÁTICO** aumento la productividad del personal técnico y administrativo, disminuyendo los tiempos en los procesos de revisión y actualización de los equipos indemático demostrando la aplicación su eficiencia que es altamente aceptable.

5.2.2.3. Económico

La aplicación del sistema **ACTIVO INDEMÀSTICO**, es una herramienta de gran apoyo en las actividades diarias, de todo el personal técnico y administrador en el área de mantenimiento de los equipos indemáticos, que influye de demàs positiva bajos los requerimientos de calidad esperada.

Con la implementación de la herramienta en el área de mantenimiento en la Universidad Estatal de Quevedo se disminuirá los tiempos de entrega de la solución de un equipo indemático porque uno de los factores de mayor impacto era el desconocimiento de los estados en que se encuentra cada equipo, mediante el uso de la Base de conocimiento del sistema **ACTIVO INDEMÁTICO** se logró reducir los tiempos.

5.3. ANÁLISIS DE RESULTADOS

Con la implantación del sistema **ACTIVO INDEMÁTICO**, el personal técnico y administrativo del área de mantenimiento de la Universidad Estatal de Quevedo, contara con una aplicación exitosa para la eficacia de las labores que se realizan diariamente y en la toma de decisiones que permitirá una mejora considerable en las actividades y el rendimiento de los procesos.

El tiempo de respuesta en la búsqueda de la indemàsción mejoró considerablemente en los procesos diarios.

Cuadro Comparativo de uso del sistema

Tiempos de respuesta en diferentes procesos			
N°	Tipo de proceso	Sin Utilizar el sistema Tiempo en minutos	Utilizando el sistema Tiempo en minutos
1	Registro de Hardware y Software	4	2
2	Registro de Incidentes	3	2
3	Registro de Inventario	8	3
4	Envío de Sugerencias	7	3
5	Búsqueda indemnación	8	1
6	Conocer ubicación de los Equipos indemnático.	9	3
TOTAL		36	13
MEDIA		2.16	6
DESV.ESTANDAR		0.68	1.91

Como observamos el porcentaje del grupo que realizaron los Equipos Indemnática usando el sistema es 2.16 y la media del grupo que obtuvo los Equipos Indemnática sin usar el sistema es 9 esto prueba que hay una variación entre los 3 grupos, lo que queremos descubrir es si existe variación en obtención de respuestas con el programa y sin él programa, es valiosa o se presenta de manera casual.

H₀ = El sistema activos indemnático no incide mucho en el manejo eficiente de la indemnación.

De igual demàs se plantea una hipótesis alterna:

H_a= El sistema activos indemnático incide en el manejo eficiente de la indemnación. El error estándar puede calculado a través de la comparación de valores, el mismo que también puede ser llamado lo cual también se lo conoce como **prueba t**.

$$S_{x_1-x_2} = \sqrt{\frac{\sum x_1^2 + \sum x_2^2}{n_1 + n_2 - 2} \left(\frac{1}{n_1} + \frac{1}{n_2} \right)}$$

$$S_{x_1-x_2} = \sqrt{\frac{0.68 + 1.91}{6 + 6 - 2} \left(\frac{1}{6} + \frac{1}{6} \right)}$$

$$S_{x1-x2} = \sqrt{\frac{2.59}{30}}$$

$$S_{x1-x2} = 0.29$$

El 22.16 es la comparación que se realiza entre los grupos, se debe comprobar ¿Si la comparación realizada es debidamente mayor que su comparación realizada y de esta manera no tomar como referencia la hipótesis de nulidad?, para su desarrollo tendremos la razón de las dos cantidades conocida también como **razón t**.

$$T = \frac{\bar{X}_1 - \bar{X}_2}{S_{x1-x2}}$$

$$T = \frac{2.16 - 6}{0.29}$$

$$T = -13.24$$

Lo que observamos como diferencia es 13.24 casos mayores de lo que esperábamos mencionada tal como la hipótesis de nulidad **¿Es esta cantidad lo suficientemente mayor como para no tomar la el nivel 0.05 de nuestra hipótesis de nulidad?** entonces calcularemos los **grados de libertad (gl)** y así despejamos cualquier mito.

$$gl = n1 + n2 - 2$$

$$gl = 6 + 6 - 2$$

$$gl = 10$$

En 11 grados de libertad, con nivel de 0.06 y tomando como referencia la tabla T-ESTUDIANTE encontraremos la razón de S de **1.8136**.

El cálculo de nuestra razón es de 14.23 un valor más que 1.9126, por lo que queda demostrado que nuestra diferencia antes del desarrollo del programa y una vez utilizado el programa su valor es más alto mayor que el que esperábamos así no tomar la hipótesis de nulidad con una significación de 0.06; de esta manera los valores son muy utilizables y así concluimos que la comparación existente no es resultado de algo casual y tomamos como certero el Ha.

14.23 > 1.9236; la hipótesis alterna es aceptada y podemos confirmar que:

El desarrollo de nuestro programa activos indemático incide en el manejo eficiente de la indemàsción.

5.4. CONCLUSIONES

Luego de haber realizado el análisis y diseño del sistema de control de activos indemático. Se logró obtener las siguientes conclusiones:

El desarrollo e implantación del sistema **ACTIVO INDEMÁTICO**, ayudo de una manera eficiente y seguro los procesos diarios que se realiza en el área de mantenimiento de la Universidad Estatal de Quevedo, permitiendo tomar decisiones mediante los reporte que el sistema genera.

El uso del sistema **ACTIVO INDEMÁTICO** en el área de mantenimiento, permite optimizar los procedimientos y asignaciones en general del área de mantenimiento y administración en la asignación de revisión y actualización de los equipos indemáticos para toma de decisiones eficientes.

El sistema, fue evaluado por el administrador del área de mantenimientos de equipos indemático donde el sistema cumplió las expectativas y objetivos planteados en el análisis del requerimiento.

5.5. RECOMENDACIONES

Con la culminación de la presente investigación me permito recomendar.

- El departamento de mantenimiento de los equipos indemáticos debe considerar implementar el sistema propuesto, porque permite un mejor manejo en el proceso de control de activos en comparación con el que se realiza en la actualidad.
- La documentación de este sistema está orientada hacia un lenguaje amigable y sencillo, para que así los usuarios se motiven a leer y aprender sobre la demás en que opera la aplicación.
- Realizar revisiones periódicas al diseño del sistema con la finalidad de efectuar los cambios necesarios para incrementar la potencialidad operativa del sistema.
- Adiestrar al personal relacionado al departamento de soporte, con la finalidad de tener a los empleados preparados en el uso de esta herramienta indemática, garantizando la seguridad y el funcionamiento óptimo del sistema.

CAPÍTULO VI

BIBLIOGRAFÍA

6.1. Literatura Citada

Calderón, A. (2007). Metodologías Ágiles. En *Universidad Nacional de Trujillo* (págs. 28-30). Perú.

Carlos Coronel, S. M. (2011). Base de datos, Diseño, Implementación y Administración. En *Base de Datos* (págs. 7-20). Editores S.A. Base de Datos.

Coloma, M. P. (2009). Conocimientos sobre servidores Web. En M. P. Coloma, *Servidores web* (pág. 12). Mexico: Editorial San Bartolo.

Cordova, V. (2012). Marco Referencial . En V. Cordova, *SISTEMA DE CONTROL Y SEGUIMIENTO DE ACTIVOS INDEMÁTICO* (pág. 57).

David, O. (21 de Agosto de 2009). *Metodología RAD*. Recuperado el 27 de Junio de 2013, de Sites Google: <https://sites.google.com/site/metodologiarad/home>

Garcia, C. (2012). Métodos Inductivo y Deductivo. *Blog Colbert*.

Marin, A. (2012). Concepto y Definición de Ajax. *Blog Enfoque y tecnología*.

Negròn, D. M. (2009). Enfoque de Administración de proceso de negocio. . En D. M. Negròn, *Administracion de Operaciones* (págs. 28-40).

Palencia-Lefler, M. (2010). Técnicas de Relaciones Públicas:. En M. Palencia-Lefler, *Manual de Comunicación Corporativa Profit Editorial. Encuesta* (págs. 57-60).

Pastor, A. (2010). Lenguaje de Programacion. En *Definición y Concepto de Visual Basic* (págs. 20-75).

- Perez, E. (2010). Activos Indemático. En *Activos, Ataques y Vulnerabilidad de Activos indemático* (pág. 15). Colombia: Bogota.
- ReSi, C. (2010). Método Cuasi-experimental. En *Método Cua-siexperimental* (pág. 12).
- Rodriguez, M. B. (2012). Web Service. En M. B. Rodríguez, *Definición y Caracteriticas de Web Service* (págs. 12-35). Mexico.
- Sánchez, E. (2013). Introduccion a los sistemas indemáticos. En E. Sánchez, *Sistemas indemàsticos* (pág. 8). Mexico: Editorial Toluca.
- Seara, D. A. (2010). Definición de asp.net. En D. A. Seara, *ASP.NET* (pág. 10). Argentina: Buenos Aires.
- Vasquez, N. (2012). Creación y Definición de Servicio Windows. *Nosho.NET*.

CAPITULO VII

ANEXOS

7.1. ENCUESTA

PREGUNTAS DIRIGIDAS A LOS ALUMNOS

Se ha tomado como muestra una cantidad de 150 alumnos.

➤ **Funcionamiento de los equipos de cómputo.**

OBJETIVO: Conocer el estado de los equipos por medio de la opinión de los estudiantes.

De acuerdo a las encuestas el 50.6% de los estudiantes coinciden en que no todos los equipos funcionan adecuadamente y un 43.3% escogieron la opción todos con una mínima diferencia del 6% que afirman que son pocos los que están en perfecto estado.

Tabla 1: ¿Cree usted que los equipos de los laboratorios de cómputo de la Carrera están en perfecto estado?

DETALLE	RESULTADOS	PORCENTAJE
Todos	65	43.3%
Algunos	76	50.6%
Pocos	9	6%

Fuente: Encuesta

Elaboración: Guillermo Campuzano Arechua

Grafico 1: ¿Cree usted que los equipos de los laboratorios de cómputo de la Carrera están en perfecto estado?

➤ **influencia del estado de los equipos en el desarrollo económico**

OBJETIVO: Descubrir el impacto que tiene el buen o mal funcionamiento de los equipos.

De acuerdo a los resultados obtenidos del 73.3% el estado de los equipos influye mucho en el desarrollo académico según la opinión de los alumnos por lo que es de gran importancia tener un control de mantenimiento.

Tabla 2: ¿Influiría en su desarrollo académico el estado de los equipos?

DETALLE	RESULTADOS	PORCENTAJE
Mucho	110	73,3%
Poco	34	22.6%
Nada	6	4%

Fuente: Encuesta

Elaboración: Guillermo Campuzano Arechua

Grafico 2: ¿Influiría en su desarrollo académico el estado de los equipos?

➤ **Conocimiento del departamento de mantenimiento de la Carrera.**

OBJETIVO: Difundir en los estudiantes el interés por conocer los departamentos que existen y servicio que brinda la Carrera.

La encuesta demuestra que el 51% de los estudiantes tiene conocimiento del mantenimiento que se le da a los equipos mientras que el 48% no, por lo que la diferencia entre una opción y otra deja en descubrimiento de la poca indemnación y falta de interés.

Tabla 3: ¿conoce usted quienes están a cargo del mantenimiento de los equipos de cómputo de la Carrera?

DETALLE	RESULTADOS	PORCENTAJE
Si	77	51.33%
No	73	48.6%

Fuente: Encuesta

Elaboración: Guillermo Campuzano Arechua

➤ **Existencia de un sistema de control de hardware y software**

OBJETIVO: Conocer si existe un sistema de control de hardware y software al servicio de la Carrera por la opinión de los alumnos.

A través de los resultados podemos descubrir que el 43% de los estudiantes tiene una opinión no asentada de la existencia de dicho sistema por lo que mediante esta acotación queda a descubrir dicha existencia

Tabla 4: ¿Sabe de algún sistema de control que se le dé al hardware y el software de la carrera?

DETALLE	RESULTADOS	PORCENTAJE
Si	40	26.6%
No	47	31,3%
Tal vez	65	43.3%

Fuente: Encuesta

Elaboración: Guillermo Campuzano Arechua

➤ **Importancia de la implementación de un sistema de control de equipos en la Carrera de Ingeniería en sistemas.**

OBJETIVO: Descubrir la importancia de Crear un sistema de control de hardware y el software.

La encuesta demuestra que la gran mayoría coincide en que es de gran importancia la implementación de dicho sistema, animando así la creación y desarrollo del mismo.

Tabla 5: ¿Cree que es importante que exista un control de activos?

DETALLE	RESULTADOS	PORCENTAJE
Si	138	92%
No	12	8%

Fuente: Encuesta

Elaboración: Guillermo Campuzano Arechua

ENCUESTAS REALIZADAS A EL PERSONAL ADMINISTRATIVO

Esta encuesta tiene como objetivo conocer las necesidades de la implementación del sistema de control de hardware y software y la cooperación por parte de los administrativos.

➤ **Verificación de pérdidas de hardware**

OBJETIVO: Conocer la necesidad de tener control de hardware mediante la verificación rápida.

Mediante los resultados obtenidos podemos conocer que existe una variable entre la opción SI con el 16.6% y AVECES con un 66,6% por lo que nos permite analizar que si existen perdidas de hardware sin la debida registración o verificación.

TABLA6: ¿Ha tenido problemas por pérdidas de accesorios dentro de los laboratorios?

DETALLE	RESULTADOS	PORCENTAJE
SI	1	16.6%
NO	1	16.6%
A VECES	4	66,6%

Fuente: Encuesta

Elaboración: Guillermo Campuzano Arehua

➤ **Control Permanente de equipos**

OBJETIVO: Conocer si responsabilidad y necesidad en control de los activos de la Carrera

Mediante las respuestas obtenidas del 83.3% que afirman pudimos detectar que si se realiza un control permanente de los equipos pero al no existir un sistema se especula que lo hagan de manera manual.

Tabla 7: Realiza usted en control permanente de los equipos?

DETALLE	RESULTADOS	PORCENTAJE
SI	5	83.3%
NO	0	0%
A VECES	1	16.6%

Fuente: Encuesta

Elaboración: Guillermo Campuzano Arehua

➤ **Verificación del funcionamiento de los equipos.**

OBJETIVO: Descubrir la necesidad de un sistema de control para el fácil descubrimiento de falencias.

Mediante esta pregunta pudimos descubrir que realmente existen problemas para detectar con rapidez los daños de cada equipo y así presentar una pronta solución.

Tabla8: Tiene problemas para detectar con facilidad la funcionalidad de los equipos?

DETALLE	RESULTADOS	PORCENTAJE
SI	5	83.3%
NO	0	0%
A VECES	1	16.6%

Fuente: Encuesta

Elaboración: Guillermo Campuzano Arechu

➤ **Factibilidad de implementación.**

OBJETIVO: Conocer la aceptación del sistema por parte de los administradores.

Las respuestas expuestas muestran mediante el la buena aceptación que tendrá la implementación del Sistema de hardware y software

Tabla 9: Sería bueno para usted contar con un sistema de registro rápido de hardware y verificación de software?

DETALLE	RESULTADOS	PORCENTAJE
SI	5	83.3%
NO	0	0%
Tal ves	1	16.6%

Fuente: Encuesta

Elaboración: Guillermo Campuzano Arechua

DESARROLLO

En esta fase se realiza la codificación del sistema y la base de datos. Se aplican el estándar de programación de código fuente CamelCase (Tipo UpperCamelCase). Debido a la estructura similar de los procedimientos almacenados, clases, Pagina Web; se describe el código fuente correspondiente al objeto PropiedadesSistema.

7.2. BASE DE DATOS

- **Tablas**

```
USE [ActivosIndemàstico]
GO

/***** Object:  Tabla [dbo].[PropiedadesSistema]      Script Date:
03/01/2014 22:38:43 *****/
SET ANSI_NULAS ON
GO

SET QUOTED_IDENTIFIER ON
GO

SET ANSI_PADDING ON
GO

CREATE TABLA [dbo].[PropiedadesSistema] (
    [Cod_Detalle] [int] IDENTITY(1,1) NOT NULA,
    [VersionCompilacion] [Cadena] (50) NULA,
    [SistemaOperativo] [Cadena] (50) NULA,
    [DescripcionPc] [Cadena] (50) NULA,
    [FechaInstalacion] [Cadena] (50) NULA,
    [FabricanteSO] [Cadena] (50) NULA,
    [DirectorioSO] [Cadena] (50) NULA,
    [Dns] [Cadena] (50) NULA,
    [Dominio] [Cadena] (50) NULA,
    [GrupoTrabajo] [Cadena] (50) NULA,
    [Fabricante] [Cadena] (50) NULA,
    [Modelo] [Cadena] (50) NULA,
    [NumProcesadorLogico] [float] NULA,
    [NumProcesadorFisico] [float] NULA,
    [TipoSistema] [Cadena] (50) NULA,
    [MemoriaFisica] [Cadena] (50) NULA,
    [Usuario] [Cadena] (50) NULA,
    CONSTRAINT [PK_PropiedadesSistema] PRIMARIO CLAVE CLUSTERED
(
    [Cod_Detalle] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_CLAVE =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARIO]
) ON [PRIMARIO]

GO

SET ANSI_PADDING OFF
GO
```

- **Procedimientos Almacenados**

- ✓ **Insertar**

```
USE [ActivosIndemàstico]
GO

/***** Object:  StoredProcedure [dbo].[GuardarPropiedadesSistema]
Script Date: 03/01/2014 22:48:42 *****/
SET ANSI_NULAS ON
```

```

GO
SET QUOTED_IDENTIFIER ON
GO

-- =====
-- Author:          <Author,,Nombre>
-- Create date:    <Create Date,,>
-- Description:    <Description,,>
-- =====
ALTER PROCEDURE [dbo].[GuardarPropiedadesSistema]
@VersionCompilacion as Cadena(50),
@SistemaOperativo as Cadena(50),
@DescripcionPc as Cadena(50),
@FechaInstalacion as Cadena(50),
@FabricanteSO as Cadena(50),
@DirectorioSO as Cadena(50),
@Dns as Cadena(50),
@Dominio as Cadena(50),
@GrupoTrabajo as Cadena(50),
@Fabricante as Cadena(50),
@Modelo as Cadena(50),
@NumProcesadorLogico as float,
@NumProcesadorFisico as float,
@TipoSistema as Cadena(50),
@MemoriaFisica as Cadena(50),
@Usuario as Cadena(50)

as
BEGIN

        SET NOCOUNT ON;

        Begin Tran TransactionAddNewUser

                begin try
                        if exists(select*from PropiedadesSistema  where
DescripcionPc=@DescripcionPc) begin
                                select('Esta Descripcion ya existe')
                        end
                        else begin
                                insert into PropiedadesSistema

valors (@VersionCompilacion,@SistemaOperativo,@DescripcionPc,

@FechaInstalacion,@FabricanteSO,@DirectorioSO,@Dns,@Dominio,@GrupoTrabajo,@
Fabricante,@Modelo,@NumProcesadorLogico

,@NumProcesadorFisico,@TipoSistema,@MemoriaFisica,@Usuario)
                                COMMIT TRAN TransactionAddNewUser

                        end

                end try
                begin catch
                        declare @ErrorMensaje nCadena(100)
                        set @ErrorMensaje= ERROR_MESSAGE()
                        raiserror('ERROR:' , @ErrorMensaje,16,1)
                        Rollback TRAN TransactionAddNewUser
                end catch

```

END

✓ Seleccionar

```
USE [ActivosIndemàstico]
GO
/***** Object:  StoredProcedure [dbo].[MostrarPropiedadesSistema]
Script Date: 03/01/2014 22:50:55 *****/
SET ANSI_NULAS ON
GO
SET QUOTED_IDENTIFIER ON
GO

-- =====
-- Author:      <Author,,Nombre>
-- Create date: <Create Date,,>
-- Description:  <Description,,>
-- =====
ALTER PROCEDURE [dbo].[MostrarPropiedadesSistema]
    @DescripcionPc as Cadena(50)
as
BEGIN

    SET NOCOUNT ON;

    Begin Tran TransactionAddNewUser

        begin try

            select * from PropiedadesSistema where
DescripcionPc=@DescripcionPc

        end try
        begin catch
            declare @ErrorMessage nCadena(100)
            set @ErrorMessage= ERROR_MESSAGE()
            raiserror('ERROR:' , @ErrorMessage,16,1)
            Rollback TRAN TransactionAddNewUser
        end catch

END
```

7.3. CLASES

```
Imports System.Windows.Dems
Imports System.Management
Imports System.Linq
Imports System.Text
Imports System
Imports System.Collections.Generic
Imports Microsoft.Win32
```

Imports LibreriaDatos

Nombreespacio Detalle_PC

```
Public Class ObjDispositivos
    Private _memoria, _banco, _capacidad, _numParte, _numSerie, _descripcion,
    _ubicacion As String
    Private _descripcionP, _fechaInstalacion, _version, _lugarInstalado As String
    Private _bus As Double
    Public Property descripcionP() As String
        Get
            Return _descripcionP
        End Get
        Set(ByVal valor As String)
            _descripcionP = valor
        End Set
    End Property
    Public Property fechaInstalacion() As String
        Get
            Return _fechaInstalacion
        End Get
        Set(ByVal valor As String)
            _fechaInstalacion = valor
        End Set
    End Property
    Public Property version() As String
        Get
            Return _version
        End Get
        Set(ByVal valor As String)
            _version = valor
        End Set
    End Property
    Public Property lugarInstalado() As String
        Get
            Return _lugarInstalado
        End Get
        Set(ByVal valor As String)
            _lugarInstalado = valor
        End Set
    End Property
    Public Property bus() As Entero
        Get
            Return _bus
        End Get
        Set(ByVal valor As Entero)
            _bus = valor
        End Set
    End Property

    Public Property memoria() As String
        Get
            Return _memoria
        End Get
        Set(ByVal valor As String)
            _memoria = valor
        End Set
    End Property

    Public Property banco() As String
        Get
            Return _banco
```



```

        End Get
        Set(ByVal valor As String)
            _banco = valor
        End Set
    End Property

    Public Property capacidad() As String
        Get
            Return _capacidad
        End Get
        Set(ByVal valor As String)
            _capacidad = valor
        End Set
    End Property

    Public Property numSerie() As String
        Get
            Return _numSerie
        End Get
        Set(ByVal valor As String)
            _numSerie = valor
        End Set
    End Property

    Public Property numParte() As String
        Get
            Return _numParte
        End Get
        Set(ByVal valor As String)
            _numParte = valor
        End Set
    End Property

    Public Property descripcion() As String
        Get
            Return _descripcion
        End Get
        Set(ByVal valor As String)
            _descripcion = valor
        End Set
    End Property

    Public Property ubicacion() As String
        Get
            Return _ubicacion
        End Get
        Set(ByVal valor As String)
            _ubicacion = valor
        End Set
    End Property

End Class

Public Class InfoSystem

    Public Elemento As Object
    Public Valor As Object
    Public Cod_Software As Object
    Public DetalleProgroma As Object
    Public Version As Object
    Public UbicacionInstalado As Object
    Public FechaSotfware As Object

    Public Sub New()

```

```

        Me.Elemento = ""
        Me.Valor = ""
        Me.Cod_Software = ""
        Me.DetallePrograma = ""
        Me.Version = ""
        Me.UbicacionInstalado = ""
        Me.FechaSotfware = ""
    End Sub

    Public Sub New(ByVal elemento As Object, ByVal valor As Object)

        Me.Elemento = elemento
        Me.Valor = valor

    End Sub

    Public Sub parametros(ByVal cod_Software As Object, ByVal detallePrograma As
Object, ByVal version As Object, ByVal ubicacionIntalacion As Object, ByVal
fechaSotfware As Object)
        Me.Cod_Software = cod_Software
        Me.DetallePrograma = detallePrograma
        Me.Version = version
        Me.UbicacionInstalado = ubicacionIntalacion
        Me.FechaSotfware = fechaSotfware
    End Sub

    Public Shared Function SO() As List(Of InfoSystem)

        Dim Lista As List(Of InfoSystem) = New List(Of InfoSystem)()

        Try

            Dim Consulta As String = "SELECT BuildNumber, Caption, CSNombre,
InstallDate, Manufacturer, WindowsDirectory FROM Win32_OperatingSystem"
            Dim ManObjetos As ManagementObjectSearcher = New
ManagementObjectSearcher(Consulta)

            De Each item As ManagementObject In ManObjetos.Get()
                Lista.Add(New InfoSystem("Versión de Compilación",
item.GetPropertyValor("BuildNumber")))
                Lista.Add(New InfoSystem("Sistema Operativo",
item.GetPropertyValor("Caption")))
                Lista.Add(New InfoSystem("PC",
item.GetPropertyValor("CSNombre").ToString()))

                Dim Fecha As Object = New Object()
                Fecha = item.GetPropertyValor("InstallDate")

                If (Object.Equals(Fecha, Nothing) = False) Then
                    Dim sFecha As String =
item.GetPropertyValor("InstallDate").ToString()
                    Lista.Add(New InfoSystem("Fecha de Instalación",
sFecha.Substring(0, 8)))
                End If

                Lista.Add(New InfoSystem("Fabricante",
item.GetPropertyValor("Manufacturer")))
                Lista.Add(New InfoSystem("Directorio",
item.GetPropertyValor("WindowsDirectory")))
            Next

        Catch ex As Exception

```

```

        MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
        MessageBoxIcon.Error)
    End Try
    Return Lista
End Function

Public Shared Function Sistema() As List(Of InfoSystem)
    Dim Lista As List(Of InfoSystem) = New List(Of InfoSystem)()
    Try

        Dim Consulta As String = "SELECT Caption, Dominio, Manufacturer,
        Model, Nombre, NumberOfLogicalProcessors, NumberOfProcessors, PRIMARIOwnerNombre,
        SystemTipo, TotalPhysicalMemory, UserNombre, Workgroup FROM Win32_ComputerSystem"
        Dim ManObjetos As ManagementObjectSearcher = New
        ManagementObjectSearcher(Consulta)

        De Each item As ManagementObject In ManObjetos.Get()
            Lista.Add(New InfoSystem("Sistema",
            item.GetPropertyValor("Caption")))
            Lista.Add(New InfoSystem("DNS Host Nombre",
            Environment.MachineNombre))
            Lista.Add(New InfoSystem("Dominio",
            item.GetPropertyValor("Dominio")))
            Lista.Add(New InfoSystem("Grupo de Trabajo",
            item.GetPropertyValor("Workgroup")))
            Lista.Add(New InfoSystem("Fabricante",
            item.GetPropertyValor("Manufacturer")))
            Lista.Add(New InfoSystem("Modelo",
            item.GetPropertyValor("Model")))
            Lista.Add(New InfoSystem("PC", item.GetPropertyValor("Nombre")))
            Lista.Add(New InfoSystem("N° de Procesadores Lógicos",
            item.GetPropertyValor("NumberOfLogicalProcessors")))
            Lista.Add(New InfoSystem("N° de Procesadores Físicos",
            item.GetPropertyValor("NumberOfProcessors")))
            Lista.Add(New InfoSystem("Propietario",
            item.GetPropertyValor("PRIMARIOwnerNombre")))
            Lista.Add(New InfoSystem("Tipo de Sistema",
            item.GetPropertyValor("SystemTipo")))
            Lista.Add(New InfoSystem("Memoria Física Instalada",
            Math.Round((((Double.Parse("0" + item.GetPropertyValor("TotalPhysicalMemory")) / 1024)
            / 1024) / 1024), 2) & " Gb")))
            Lista.Add(New InfoSystem("Usuario",
            item.GetPropertyValor("UserNombre")))
        Next

        Catch ex As Exception
            MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
            MessageBoxIcon.Error)
        End Try
        Return Lista
    End Function

Public Shared Function Procesador() As List(Of InfoSystem)
    Dim Lista As List(Of InfoSystem) = New List(Of InfoSystem)()
    Try

        Dim Consulta As String = "SELECT Caption, DatosWidth, DeviceID,
        Manufacturer, MaxClockSpeed, Nombre, NumberOfCores, NumberOfLogicalProcessors FROM
        Win32_Processor"
        Dim ManObjetos As ManagementObjectSearcher = New
        ManagementObjectSearcher(Consulta)
    
```

```

        De Each item As ManagementObject In ManObjetos.Get()
            Lista.Add(New InfoSystem("Procesador",
item.GetPropertyValor("Caption")))
            Lista.Add(New InfoSystem("Bus",
item.GetPropertyValor("DatosWidth")))
            Lista.Add(New InfoSystem("ID Dispositivo",
item.GetPropertyValor("DeviceID")))
            Lista.Add(New InfoSystem("Fabricante",
item.GetPropertyValor("Manufacturer")))
            Lista.Add(New InfoSystem("Velocidad",
item.GetPropertyValor("MaxClockSpeed")))
            Lista.Add(New InfoSystem("Nombre",
item.GetPropertyValor("Nombre")))
            Lista.Add(New InfoSystem("N° de Nucleos",
item.GetPropertyValor("NumberOfCores")))
            Lista.Add(New InfoSystem("N° de Procesadores Lógicos",
item.GetPropertyValor("NumberOfLogicalProcessors")))
        Next

        Catch ex As Exception
            MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
        End Try
        Return Lista
    End Function

    Public Shared Function Memoria() As List(Of InfoSystem)
        Dim Lista As List(Of InfoSystem) = New List(Of InfoSystem)()

        Try

            Dim Consulta As String = "SELECT Caption, BankLabel, Capacity,
DatosWidth, Description, DeviceLocator, PartNumber, SerialNumber FROM
Win32_PhysicalMemory"
            Dim ManObjetos As ManagementObjectSearcher = New
ManagementObjectSearcher(Consulta)

            De Each item As ManagementObject In ManObjetos.Get()
                Lista.Add(New InfoSystem("Memoria",
item.GetPropertyValor("Caption")))
                Lista.Add(New InfoSystem("Banco",
item.GetPropertyValor("BankLabel")))
                Lista.Add(New InfoSystem("Capacidad",
Math.Round((((Double.Parse("0" + item.GetPropertyValor("Capacity")) / 1024) / 1024) /
1024), 2) & " Gb")))
                Lista.Add(New InfoSystem("Bus",
item.GetPropertyValor("DatosWidth")))
                Lista.Add(New InfoSystem("Descripción",
item.GetPropertyValor("Description")))
                Lista.Add(New InfoSystem("Ubicación",
item.GetPropertyValor("DeviceLocator")))
                Lista.Add(New InfoSystem("N° de Parte",
item.GetPropertyValor("PartNumber")))
                Lista.Add(New InfoSystem("N° de Serie",
item.GetPropertyValor("SerialNumber")))
            Next

            Catch ex As Exception
                MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
            End Try

```

```

        Return Lista
    End Function

    Public Shared Function AdaptadoresRed() As List(Of InfoSystem)
        Dim Lista As List(Of InfoSystem) = New List(Of InfoSystem)()

        Try

            Dim Consulta As String = "SELECT
Caption,Description,DHCPEnabled,DHCPServer,DNSHostNombre,IPEntabled,MACAddress from
Win32_NetworkAdapterConfiguration"
            Dim ManObjetos As ManagementObjectSearcher = New
ManagementObjectSearcher(Consulta)

            De Each item As ManagementObject In ManObjetos.Get()

                If ((item.GetPropertyValor("Caption") <> Nothing) And
(item.GetPropertyValor("Description") <> Nothing) And
(item.GetPropertyValor("DHCPEnabled") <> Nothing) And
(item.GetPropertyValor("DHCPServer") <> Nothing) And
(item.GetPropertyValor("DNSHostNombre") <> Nothing) And
(item.GetPropertyValor("IPEntabled") <> Nothing) And
(item.GetPropertyValor("MACAddress") <> Nothing)) Then

                    Lista.Add(New InfoSystem("Detalle",
item.GetPropertyValor("Caption")))
                    Lista.Add(New InfoSystem("Descripcion",
item.GetPropertyValor("Description")))
                    Lista.Add(New InfoSystem("DHCP",
item.GetPropertyValor("DHCPEnabled")))
                    Lista.Add(New InfoSystem("Servidor",
item.GetPropertyValor("DHCPServer")))
                    Lista.Add(New InfoSystem("DnsHost",
item.GetPropertyValor("DNSHostNombre")))
                    Lista.Add(New InfoSystem("IP",
item.GetPropertyValor("IPEntabled")))
                    Lista.Add(New InfoSystem("Mac",
item.GetPropertyValor("MACAddress")))

                End If

            Next

            Catch ex As Exception
                MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
            End Try
            Return Lista
        End Function

        Public Function MostrarMemoria() As ObjDispositivos()
            Dim temp() As ObjDispositivos = Nothing

            Dim i As Entero
            Try

                Dim Consulta As String = "SELECT Caption, BankLabel, Capacity,
DatosWidth, Description, DeviceLocator, PartNumber, SerialNumber FROM
Win32_PhysicalMemory"
                Dim ManObjetos As ManagementObjectSearcher = New
ManagementObjectSearcher(Consulta)

```

```

        De Each item As ManagementObject In ManObjetos.Get()
            ReDim Preserve temp(i)
            temp(i) = New ObjDispositivos
            temp(i).memoria = item.GetPropertyValor("Caption")
            temp(i).banco = item.GetPropertyValor("BankLabel")
            temp(i).capacidad = Math.Round((((Double.Parse("0" +
item.GetPropertyValor("Capacity")) / 1024) / 1024) / 1024), 2) & " Gb"
            temp(i).bus = item.GetPropertyValor("DatosWidth")
            temp(i).descripcion = item.GetPropertyValor("Description")

            temp(i).ubicacion = item.GetPropertyValor("DeviceLocator")
            temp(i).numParte = item.GetPropertyValor("PartNumber")
            temp(i).numSerie = item.GetPropertyValor("SerialNumber")

            i = i + 1

        Next

        Catch edem As SqlClient.SqlException
            MsgBox("Ocurrio un error en la pagina" & vbNewLine & vbNewLine &
edem.Message, MsgBoxStyle.Critical, "")
        End Try

        Return temp

    End Function

    Public Function MostrarProgramasPC(ByVal NombrePc As String) As
ObjDispositivos()
        Dim temp() As ObjDispositivos = Nothing
        Dim strComputer As Object
        Dim objWMIService As Object
        Dim colSoftware As Object
        'Dim objFSO As Object
        Dim i As Entero
        'Dim objTextFile As Object

        Try
            strComputer = NombrePc

            objWMIService = GetObject("winmgmts:" &
"{impersonationLevel=impersonate}!\" & strComputer & "\root\cimv2")

            colSoftware = objWMIService.ExecQuery("SELECT
Caption,Version,LocalPackage,InstallDate FROM Win32_Product")

            If colSoftware.Count > 0 Then

                De Each objSoftware In colSoftware

                    ReDim Preserve temp(i)
                    temp(i) = New ObjDispositivos
                    temp(i).descripcionP = objSoftware.Caption
                    temp(i).version = objSoftware.Version
                    temp(i).lugarInstalado = objSoftware.LocalPackage
                    temp(i).fechaInstalacion = objSoftware.InstallDate
                    i = i + 1

                Next

            Else

            End If

        End Try
    End Function

```

```

        Catch edem As SqlClient.SqlException
            MsgBox("Ocurrio un error en la pagina" & vbNewLine & vbNewLine &
edem.Message, MsgBoxStyle.Critical, "")
        End Try

        Return temp

    End Function

    Public Shared Function MostrarProgramasPC2(ByVal NombrePc As String) As
List(Of InfoSystem)
        Dim Lista As List(Of InfoSystem) = New List(Of InfoSystem)()
        'Dim temp() As Libreria_Datos.DetalleSistema.objDetalle = Nothing
        Dim strComputer As Object
        Dim objWMIService As Object
        Dim colSoftware As Object
        'Dim objFSO As Object
        'Dim i As Entero
        'Dim objTextFile As Object

        Try
            strComputer = NombrePc

            objWMIService = GetObject("winmgmts:" &
"{impersonationLevel=impersonate}!\" & strComputer & "\root\cimv2")

            colSoftware = objWMIService.ExecQuery("SELECT
Caption,Version,LocalPackage,InstallDate FROM Win32_Product")

            If colSoftware.Count > 0 Then

                De Each objSoftware In colSoftware
                    Lista.Add(New InfoSystem("Caption", objSoftware.Caption))
                    Lista.Add(New InfoSystem("Version", objSoftware.Version))
                    Lista.Add(New InfoSystem("LocalPackage",
objSoftware.LocalPackage))
                    Lista.Add(New InfoSystem("InstallDate",
objSoftware.InstallDate))
                Next

            Else

                End If

            Catch edem As SqlClient.SqlException
                MsgBox("Ocurrio un error en la pagina" & vbNewLine & vbNewLine &
edem.Message, MsgBoxStyle.Critical, "")
            End Try

            Return Lista

        End Function

    Public Shared Function MainBoard() As List(Of InfoSystem)
        Dim Lista As List(Of InfoSystem) = New List(Of InfoSystem)()
        Try

            Dim Consulta As String = "SELECT Caption, Manufacturer, PartNumber,
SerialNumber, Version FROM Win32_BaseBoard"

```

```

        Dim ManObjetos As ManagementObjectSearcher = New
ManagementObjectSearcher(Consulta)

        De Each item As ManagementObject In ManObjetos.Get()
            Lista.Add(New InfoSystem("Placa Base",
item.GetPropertyValor("Caption")))
            Lista.Add(New InfoSystem("Fabricante",
item.GetPropertyValor("Manufacturer")))
            Lista.Add(New InfoSystem("N° de Serie",
item.GetPropertyValor("SerialNumber")))
            Lista.Add(New InfoSystem("Versión",
item.GetPropertyValor("Version")))
        Next

        Catch ex As Exception
            MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
        End Try
        Return Lista
    End Function

    Public Shared Function BIOS() As List(Of InfoSystem)
        Dim Lista As List(Of InfoSystem) = New List(Of InfoSystem)()
        Try

            Dim Consulta As String = "SELECT Caption, Manufacturer, ReleaseDate,
SerialNumber, SMBIOSBIOSVersion FROM Win32_BIOS"
            Dim ManObjetos As ManagementObjectSearcher = New
ManagementObjectSearcher(Consulta)

            De Each item As ManagementObject In ManObjetos.Get()
                Lista.Add(New InfoSystem("BIOS",
item.GetPropertyValor("Caption")))
                Lista.Add(New InfoSystem("Fabricante",
item.GetPropertyValor("Manufacturer")))
                Lista.Add(New InfoSystem("N° de Serie",
item.GetPropertyValor("SerialNumber")))
                Lista.Add(New InfoSystem("Versión",
item.GetPropertyValor("SMBIOSBIOSVersion")))

                Dim Fecha As Object = New Object()
                Fecha = item.GetPropertyValor("ReleaseDate")

                If (Object.Equals(Fecha, Nothing) = False) Then

                    Dim sFecha As String =
item.GetPropertyValor("ReleaseDate").ToString()
                    Lista.Add(New InfoSystem("Release Fecha", sFecha.Substring(0,
8)))
                End If
            Next

            Catch ex As Exception
                MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
            End Try
            Return Lista
        End Function

    End Class
End Nombreespacio

```


7.3. PAGINA WEB

```
Imports LibreriaDatos
Partial Class Defecto2
    Inherits System.Web.UI.Page

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
        Handles Me.Load
            If (Not IsPostBack) Then

                Dim MostrarPc As New Libreria_Datos.DetalleSistema
                Dim ObjPC() As Libreria_Datos.DetalleSistema.objDetalle

                ObjPC = MostrarPc.MostrarTodasPropiedadesSistema()

                If Not (ObjPC) Is Nothing Then

                    De i As Entero = 0 To UBound(ObjPC)
                        Dim nuevoNodo As TreeNode = New TreeNode(ObjPC(i).DescripcionPc,
i)
                            tvEquipos.Nodes(0).PequeñaNodes.Add(nuevoNodo)
                        Next

                    End If

                End If
            End Sub

    Protected Sub tvEquipos_SelectedNodeChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles tvEquipos.SelectedNodeChanged
        '-----SISTEMA-----
        Session("PC") = tvEquipos.SelectedNode.Text

        Dim dt As New System.Datos.DatosTabla
        dt.Columnas.Add("DESCRIPCION")
        dt.Columnas.Add("DETALLE")
        Dim item As New LibreriaDatos.Libreria_Datos.InterfaceWeb
        Dim mostrarPc As New List(Of
LibreriaDatos.Libreria_Datos.InterfaceWeb)(LibreriaDatos.Libreria_Datos.InterfaceWeb.M
ostrarPropiedadesSistema(tvEquipos.SelectedNode.Text))
        Session("idPC") = mostrarPc(0).Valor
        De Each item In mostrarPc
            Dim dr As System.Datos.DatosRow = dt.NewRow()

            dr("DESCRIPCION") = item.Elemento
            dr("DETALLE") = item.Valor
            dt.Rows.Add(dr)
        Next

        dgvDetalle.DatosSource = dt
        dgvDetalle.DatosBind()

        '-----PROCESADOR-----

        Dim dt2 As New System.Datos.DatosTabla
        dt2.Columnas.Add("DESCRIPCION")
```

```

        dt2.Columns.Add("DETALLE")
        Dim mostrarPc2 As New List(Of
LibreriaDatos.Libreria_Datos.InterfaceWeb)(LibreriaDatos.Libreria_Datos.InterfaceWeb.M
ostrarPropiedadesProcesador(mostrarPc(0).Valor))
        De Each item2 As LibreriaDatos.Libreria_Datos.InterfaceWeb In mostrarPc2
            Dim dr As System.Datos.DatosRow = dt2.NewRow()
            dr("DESCRIPCION") = item2.Elemento
            dr("DETALLE") = item2.Valor
            dt2.Rows.Add(dr)
        Next
        dgvProcesador.DatosSource = dt2
        dgvProcesador.DatosBind()

'-----MEMORIA-----

        Dim dt3 As New System.Datos.DatosTabla
        dt3.Columns.Add("DESCRIPCION")
        dt3.Columns.Add("DETALLE")
        Dim mostrarPc3 As New List(Of
LibreriaDatos.Libreria_Datos.InterfaceWeb)(LibreriaDatos.Libreria_Datos.InterfaceWeb.M
ostrarPropiedadesMomoria(mostrarPc(0).Valor))
        De Each item3 As LibreriaDatos.Libreria_Datos.InterfaceWeb In mostrarPc3
            Dim dr As System.Datos.DatosRow = dt3.NewRow()
            dr("DESCRIPCION") = item3.Elemento
            dr("DETALLE") = item3.Valor
            dt3.Rows.Add(dr)
        Next
        dgvMemoria.DatosSource = dt3
        dgvMemoria.DatosBind()

'-----MAINBOARD-----

        Dim dt4 As New System.Datos.DatosTabla
        dt4.Columns.Add("DESCRIPCION")
        dt4.Columns.Add("DETALLE")
        Dim mostrarPc4 As New List(Of
LibreriaDatos.Libreria_Datos.InterfaceWeb)(LibreriaDatos.Libreria_Datos.InterfaceWeb.M
ostrarPropiedadesMainBoard(mostrarPc(0).Valor))
        De Each item4 As LibreriaDatos.Libreria_Datos.InterfaceWeb In mostrarPc4
            Dim dr As System.Datos.DatosRow = dt4.NewRow()
            dr("DESCRIPCION") = item4.Elemento
            dr("DETALLE") = item4.Valor
            dt4.Rows.Add(dr)
        Next
        dgvMainBoard.DatosSource = dt4
        dgvMainBoard.DatosBind()

'-----BIOS-----

        Dim dt5 As New System.Datos.DatosTabla
        dt5.Columns.Add("DESCRIPCION")
        dt5.Columns.Add("DETALLE")
        Dim mostrarPc5 As New List(Of
LibreriaDatos.Libreria_Datos.InterfaceWeb)(LibreriaDatos.Libreria_Datos.InterfaceWeb.M
ostrarPropiedadesBios(mostrarPc(0).Valor))
        De Each item5 As LibreriaDatos.Libreria_Datos.InterfaceWeb In mostrarPc5
            Dim dr As System.Datos.DatosRow = dt5.NewRow()
            dr("DESCRIPCION") = item5.Elemento
            dr("DETALLE") = item5.Valor
            dt5.Rows.Add(dr)
        Next
        dgvBios.DatosSource = dt5

```

```

dgvBios.DatosBind()

'-----DISCODURO-----

Dim dt6 As New System.Datos.DatosTabla
dt6.Columnas.Add("DESCRIPCION")
dt6.Columnas.Add("DETALLE")
Dim mostrarPc6 As New List(Of
LibreriaDatos.Libreria_Datos.InterfaceWeb)(LibreriaDatos.Libreria_Datos.InterfaceWeb.M
ostrarPropiedadesDiscoDuro(mostrarPc(0).Valor))
De Each item6 As LibreriaDatos.Libreria_Datos.InterfaceWeb In mostrarPc6
    Dim dr As System.Datos.DatosRow = dt6.NewRow()
    dr("DESCRIPCION") = item6.Elemento
    dr("DETALLE") = item6.Valor
    dt6.Rows.Add(dr)
Next
dgvDisco.DatosSource = dt6
dgvDisco.DatosBind()

'-----DISCOLOGICOS-----

Dim dt7 As New System.Datos.DatosTabla
dt7.Columnas.Add("DESCRIPCION")
dt7.Columnas.Add("DETALLE")
Dim mostrarPc7 As New List(Of
LibreriaDatos.Libreria_Datos.InterfaceWeb)(LibreriaDatos.Libreria_Datos.InterfaceWeb.M
ostrarPropiedadesDiscoLogico(mostrarPc6(0).Valor))
De Each item7 As LibreriaDatos.Libreria_Datos.InterfaceWeb In mostrarPc7
    Dim dr As System.Datos.DatosRow = dt7.NewRow()
    dr("DESCRIPCION") = item7.Elemento
    dr("DETALLE") = item7.Valor
    dt7.Rows.Add(dr)
Next
dgvDiscoLogico.DatosSource = dt7
dgvDiscoLogico.DatosBind()

'-----TECLADO-----

Dim dt8 As New System.Datos.DatosTabla
dt8.Columnas.Add("DESCRIPCION")
dt8.Columnas.Add("DETALLE")
Dim mostrarPc8 As New List(Of
LibreriaDatos.Libreria_Datos.InterfaceWeb)(LibreriaDatos.Libreria_Datos.InterfaceWeb.M
ostrarPropiedadesTeclado(mostrarPc(0).Valor))
De Each item8 As LibreriaDatos.Libreria_Datos.InterfaceWeb In mostrarPc8
    Dim dr As System.Datos.DatosRow = dt8.NewRow()
    dr("DESCRIPCION") = item8.Elemento
    dr("DETALLE") = item8.Valor
    dt8.Rows.Add(dr)
Next
dgvTeclado.DatosSource = dt8
dgvTeclado.DatosBind()

'-----IMPRESORA-----

Dim dt9 As New System.Datos.DatosTabla
dt9.Columnas.Add("DESCRIPCION")
dt9.Columnas.Add("DETALLE")
Dim mostrarPc9 As New List(Of
LibreriaDatos.Libreria_Datos.InterfaceWeb)(LibreriaDatos.Libreria_Datos.InterfaceWeb.M
ostrarPropiedadesImpresora(mostrarPc(0).Valor))

```

```

De Each item9 As LibreriaDatos.Libreria_Datos.InterfaceWeb In mostrarPc9
    Dim dr As System.Datos.DatosRow = dt9.NewRow()
    dr("DESCRIPCION") = item9.Elemento
    dr("DETALLE") = item9.Valor
    dt9.Rows.Add(dr)
Next
dgvImpresora.DatosSource = dt9
dgvImpresora.DatosBind()

'-----USB-----

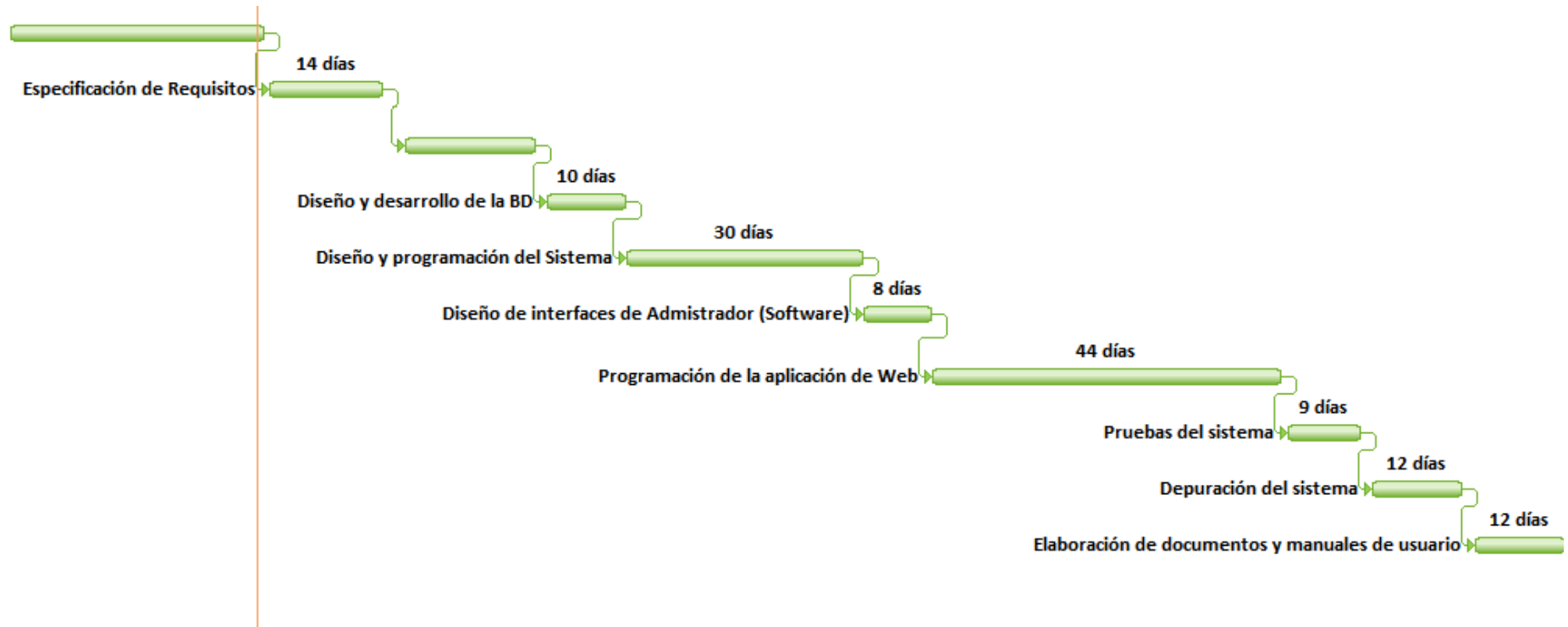
Dim dt10 As New System.Datos.DatosTabla
dt10.Columnas.Add("DESCRIPCION")
dt10.Columnas.Add("DETALLE")
Dim mostrarPc10 As New List(Of
LibreriaDatos.Libreria_Datos.InterfaceWeb)(LibreriaDatos.Libreria_Datos.InterfaceWeb.M
ostrarPropiedadesUSB(mostrarPc(0).Valor))
De Each item10 As LibreriaDatos.Libreria_Datos.InterfaceWeb In mostrarPc10
    Dim dr As System.Datos.DatosRow = dt10.NewRow()
    dr("DESCRIPCION") = item10.Elemento
    dr("DETALLE") = item10.Valor
    dt10.Rows.Add(dr)
Next
dgvUsb.DatosSource = dt10
dgvUsb.DatosBind()

End Sub

Protected Sub btnAsignar_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnAsignar.Click
    Response.Redirect("RegistroSalasPc.aspx")
End Sub
End Class

```

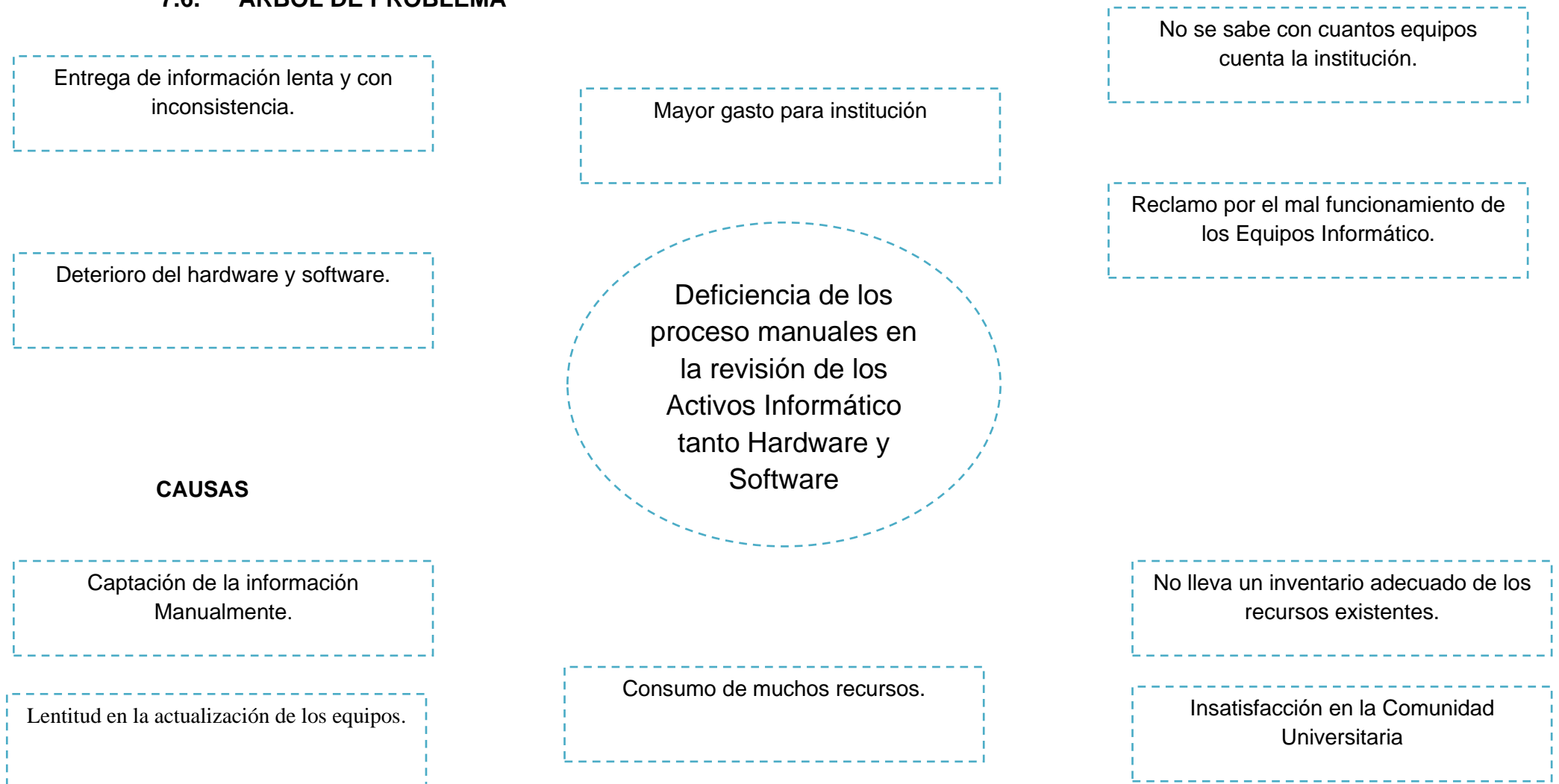
7.4. Cronograma



7.5. DEMULACIÓN DEL PROBLEMA

PROBLEMA	OBJETIVOS	RESULTADOS ESPERADOS
¿¿De qué manera mejoraría la eficiencia de gestión inventario y actualización de los Equipos Indemático de la UTEQ?	Desarrollar un Sistema automatizado que controle, administre y monitoree todo los activos indemático de la UTEQ.	Como resultado este software permitirá controlar los activos indemáticos destinados a optimizar los flujos de indemación administrativa de la Institución, de manera eficiente, confiable y segura.
	ESPECIFICOS	
¿Cómo optimizar el proceso de recolección de la indemación de los equipos indemáticos?	Realizar investigación en grupo de equipos, los componentes, estabilidad, responsabilidad de usuario, estado, programas por páginas.	Disminuirá los problemas de lentitud para realizar la revisión de los Equipos indemáticos.
¿Cómo mejorar las actividades para así tener una planificación adecuada de los procesos de actualización de los equipos indemáticos?	Control específico del software y Hardware en la Universidad por ejemplo estado de Licencias.	El Sistema Indemático brindara mayor eficiencia en los proceso y en la actualización de los equipos porque se podrá detectar más rápido donde está ocurriendo los problemas.
¿¿Cómo obtener indemación histórica de los equipos que fueron actualizados?	Mejorar el contenido existente contabla perteneciente al activo fijo en ccomputadoras que ya no se usan.	Al tener todo los procesos automatizados se logra tener la indemación de los equipos de una manera rápida y confiable para así satisfacer las necesidades de la comunidad Universitaria.

7.6. ARBOL DE PROBLEMA



7.7. TABLA T-STUDENT

Tabla t-Student



Grados de libertad	0.25	0.1	0.05	0.025	0.01	0.005
1	1.0000	3.0777	6.3137	12.7062	31.8210	63.6559
2	0.8165	1.8856	2.9200	4.3027	6.9645	9.9250
3	0.7649	1.6377	2.3534	3.1824	4.5407	5.8408
4	0.7407	1.5332	2.1318	2.7765	3.7469	4.6041
5	0.7267	1.4759	2.0150	2.5706	3.3649	4.0321
6	0.7176	1.4398	1.9432	2.4469	3.1427	3.7074
7	0.7111	1.4149	1.8946	2.3646	2.9979	3.4995
8	0.7064	1.3968	1.8595	2.3060	2.8965	3.3554
9	0.7027	1.3830	1.8331	2.2622	2.8214	3.2498
10	0.6998	1.3722	1.8125	2.2281	2.7638	3.1693
11	0.6974	1.3634	1.7959	2.2010	2.7181	3.1058
12	0.6955	1.3562	1.7823	2.1788	2.6810	3.0545
13	0.6938	1.3502	1.7709	2.1604	2.6503	3.0123
14	0.6924	1.3450	1.7613	2.1448	2.6245	2.9768
15	0.6912	1.3406	1.7531	2.1315	2.6025	2.9467
16	0.6901	1.3368	1.7459	2.1199	2.5835	2.9208
17	0.6892	1.3334	1.7396	2.1098	2.5669	2.8982
18	0.6884	1.3304	1.7341	2.1009	2.5524	2.8784
19	0.6876	1.3277	1.7291	2.0930	2.5395	2.8609
20	0.6870	1.3253	1.7247	2.0860	2.5280	2.8453
21	0.6864	1.3232	1.7207	2.0796	2.5176	2.8314
22	0.6858	1.3212	1.7171	2.0739	2.5083	2.8188
23	0.6853	1.3195	1.7139	2.0687	2.4999	2.8073
24	0.6848	1.3178	1.7109	2.0639	2.4922	2.7970
25	0.6844	1.3163	1.7081	2.0595	2.4851	2.7874
26	0.6840	1.3150	1.7056	2.0555	2.4786	2.7787
27	0.6837	1.3137	1.7033	2.0518	2.4727	2.7707
28	0.6834	1.3125	1.7011	2.0484	2.4671	2.7633
29	0.6830	1.3114	1.6991	2.0452	2.4620	2.7564
30	0.6828	1.3104	1.6973	2.0423	2.4573	2.7500
31	0.6825	1.3095	1.6955	2.0395	2.4528	2.7440
32	0.6822	1.3086	1.6939	2.0369	2.4487	2.7385
33	0.6820	1.3077	1.6924	2.0345	2.4448	2.7333
34	0.6818	1.3070	1.6909	2.0322	2.4411	2.7284
35	0.6816	1.3062	1.6896	2.0301	2.4377	2.7238
36	0.6814	1.3055	1.6883	2.0281	2.4345	2.7195
37	0.6812	1.3049	1.6871	2.0262	2.4314	2.7154
38	0.6810	1.3042	1.6860	2.0244	2.4286	2.7116
39	0.6808	1.3036	1.6849	2.0227	2.4258	2.7079
40	0.6807	1.3031	1.6839	2.0211	2.4233	2.7045
41	0.6805	1.3025	1.6829	2.0195	2.4208	2.7012
42	0.6804	1.3020	1.6820	2.0181	2.4185	2.6981
43	0.6802	1.3016	1.6811	2.0167	2.4163	2.6951
44	0.6801	1.3011	1.6802	2.0154	2.4141	2.6923
45	0.6800	1.3007	1.6794	2.0141	2.4121	2.6896
46	0.6799	1.3002	1.6787	2.0129	2.4102	2.6870
47	0.6797	1.2998	1.6779	2.0117	2.4083	2.6846
48	0.6796	1.2994	1.6772	2.0106	2.4066	2.6822
49	0.6795	1.2991	1.6766	2.0096	2.4049	2.6800

50	0.6794	1.2987	1.6759	2.0086	2.4033	2.6778
51	0.6793	1.2984	1.6753	2.0076	2.4017	2.6757
52	0.6792	1.2980	1.6747	2.0066	2.4002	2.6737
53	0.6791	1.2977	1.6741	2.0057	2.3988	2.6718
54	0.6791	1.2974	1.6736	2.0049	2.3974	2.6700
55	0.6790	1.2971	1.6730	2.0040	2.3961	2.6682
56	0.6789	1.2969	1.6725	2.0032	2.3948	2.6665
57	0.6788	1.2966	1.6720	2.0025	2.3936	2.6649
58	0.6787	1.2963	1.6716	2.0017	2.3924	2.6633
59	0.6787	1.2961	1.6711	2.0010	2.3912	2.6618
60	0.6786	1.2958	1.6706	2.0003	2.3901	2.6603
61	0.6785	1.2956	1.6702	1.9996	2.3890	2.6589
62	0.6785	1.2954	1.6698	1.9990	2.3880	2.6575
63	0.6784	1.2951	1.6694	1.9983	2.3870	2.6561
64	0.6783	1.2949	1.6690	1.9977	2.3860	2.6549
65	0.6783	1.2947	1.6686	1.9971	2.3851	2.6536
66	0.6782	1.2945	1.6683	1.9966	2.3842	2.6524
67	0.6782	1.2943	1.6679	1.9960	2.3833	2.6512
68	0.6781	1.2941	1.6676	1.9955	2.3824	2.6501
69	0.6781	1.2939	1.6672	1.9949	2.3816	2.6490
70	0.6780	1.2938	1.6669	1.9944	2.3808	2.6479
71	0.6780	1.2936	1.6666	1.9939	2.3800	2.6469
72	0.6779	1.2934	1.6663	1.9935	2.3793	2.6458
73	0.6779	1.2933	1.6660	1.9930	2.3785	2.6449
74	0.6778	1.2931	1.6657	1.9925	2.3778	2.6439
75	0.6778	1.2929	1.6654	1.9921	2.3771	2.6430
76	0.6777	1.2928	1.6652	1.9917	2.3764	2.6421
77	0.6777	1.2926	1.6649	1.9913	2.3758	2.6412
78	0.6776	1.2925	1.6646	1.9908	2.3751	2.6403
79	0.6776	1.2924	1.6644	1.9905	2.3745	2.6395
80	0.6776	1.2922	1.6641	1.9901	2.3739	2.6387
81	0.6775	1.2921	1.6639	1.9897	2.3733	2.6379
82	0.6775	1.2920	1.6636	1.9893	2.3727	2.6371
83	0.6775	1.2918	1.6634	1.9890	2.3721	2.6364
84	0.6774	1.2917	1.6632	1.9886	2.3716	2.6356
85	0.6774	1.2916	1.6630	1.9883	2.3710	2.6349
86	0.6774	1.2915	1.6628	1.9879	2.3705	2.6342
87	0.6773	1.2914	1.6626	1.9876	2.3700	2.6335
88	0.6773	1.2912	1.6624	1.9873	2.3695	2.6329
89	0.6773	1.2911	1.6622	1.9870	2.3690	2.6322
90	0.6772	1.2910	1.6620	1.9867	2.3685	2.6316
91	0.6772	1.2909	1.6618	1.9864	2.3680	2.6309
92	0.6772	1.2908	1.6616	1.9861	2.3676	2.6303
93	0.6771	1.2907	1.6614	1.9858	2.3671	2.6297
94	0.6771	1.2906	1.6612	1.9855	2.3667	2.6291
95	0.6771	1.2905	1.6611	1.9852	2.3662	2.6286
96	0.6771	1.2904	1.6609	1.9850	2.3658	2.6280
97	0.6770	1.2903	1.6607	1.9847	2.3654	2.6275
98	0.6770	1.2903	1.6606	1.9845	2.3650	2.6269
99	0.6770	1.2902	1.6604	1.9842	2.3646	2.6264
100	0.6770	1.2901	1.6602	1.9840	2.3642	2.6259
∞	0.6745	1.2816	1.6449	1.9600	2.3263	2.5758

Nombre: Nancy Rodríguez Gavilanes

Firma de responsabilidad del Profesor Guía

C.I.:1802843704

Nombre: Campuzano Arechua Guillermo Armando

Firma de responsabilidad del Alumno

C.I.: 1206039933

DEMULARIO PARA LA PRESENTACION DEL PROYECTO DE TESIS DE GRADO

Fecha de presentación:	(Marzo 20 del 2014)
------------------------	---------------------

Datos Generales

Primer Apellido:		Segundo Apellido:	
Campuzano		Arechua	
Primer Nombre:		Segundo Nombre:	
Guillermo		Armando	
Número de Cédula:	1206039933		
Número de Matrícula:			
Facultad:	Ciencias de la Ingeniería		
Carrera:	Ingeniería en Sistemas		
Dirección / Ciudad:		Teléfono:	Fax:
Quito-Juan Salinas y Rio de Janeiro OE2-10		0989835824	
Correo Electrónico:			
guillermoarma@hotmail.com			

Director de tesis:			
Apellidos:		Nombres:	
Rodríguez Gavilanes		Nancy	
Título académico de mayor nivel:		Magister en Investigación para el desarrollo Educativo	
Teléfono:	0992358291		
Correo Electrónico:		nrodriguez@uteq.edu.ec	