



INSTITUTO TECNOLÓGICO SUPERIOR "CORDILLERA"

INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA

ESCUELA DE SISTEMAS

Proyecto de Grado, previa obtención del título de:
Tecnólogo Analista de Sistemas

TEMA

DESARROLLO E IMPLEMENTACION DE UN SISTEMA DE
MANTENIMIENTO DE EQUIPOS HOSPITALARIOS PARA IX
BIOTRON

AUTOR:

Jefferson Eduardo Castro Orosco

TUTOR:

Ing. Carlos Romero

2011

QUITO-ECUADOR



CAPITULO I

EL PROBLEMA

1.1 Planteamiento del Problema

La empresa auspiciante para el proyecto de grado es IX Biotron del Ecuador, y entre sus actividades están:

- Reparación y mantenimiento de equipo médico.
- Contratos de mantenimiento preventivo/ correctivo planificado.
- Asistencia técnica por llamada.
- Programas de mantenimiento especiales 24 horas 7 días a la semana.
- Calibración y certificación de operatividad y QA de equipo médico.
- Calibración y certificación de operatividad y QA de equipos de rayos x.
- Capacitación Técnica, cursos y seminarios de acuerdo a necesidades.

Actualmente la empresa IX Biotron está encargada del Mantenimiento de Equipos Hospitalarios dentro del Hospital Carlos Andrade Marín, y uno de los problemas que se le ha presentado es llevar los procesos de una forma manual o por órdenes de trabajo los cuales hace que se delimite el tiempo de procesos de información, a causa de eso no se puede llevar de manera adecuada los siguientes parámetros que a continuación se detalla:

- El ingreso de datos cuando un equipo se encuentra en mal estado, no se lo controla, ya que muchas veces la información es repetitiva. Este problema se puede visualizar al momento que la empresa direcciona al Técnico a solucionar el problema que ya fue resuelto con anterioridad.



- Al momento las órdenes de trabajo no se encuentran digitalizadas, dificultando una búsqueda rápida en el historial de vida útil de un equipo. Esto por las veces que fueron reparadas dichos equipos.
- Para la realización del mantenimiento sea preventivo o correctivo de los equipos, no existe una planificación mensual, bimensual o semestral. Dificultando el trabajo óptimo de los Técnicos, ya que sin contar con esta, se realizan los procesos de una manera lenta y desordenada.
- Desperdicio de recurso humano, como se menciona; se direccionan los procesos manualmente por parte de secretaria de mantenimiento. En este punto se hace énfasis ya que los propios Técnicos podrían guiarse y realizar cronogramas, según el criterio técnico profesional de cada uno y no en base a la secretaría.
- La problemática más importante es que la empresa IX Biotron no cuenta con un Sistema que controle los Procesos de Mantenimiento de Equipos Médicos. El sistema es uno de los requisitos que impuso el Hospital "Carlos Andrade Marín" sobre la empresa al momento que gano el contrato durante el periodo 2011-2012.

Anexo: 1

Título: Carta de Auspicio

Fuente: IX BIOTRON

1.2 Formulación del Problema

¿Con el Desarrollo e Implementacion de un Sistema de Mantenimiento de Equipos Hospitalarios se logrará solucionar los problemas de cotrol de ordenes de servicio y tiempos muertos de los mantenimientos preventivos y correctivos realizados por la empresa IX Biotron?



1.3 Delimitación del Problema

El presente proyecto iniciará su realización en el mes de Noviembre del 2011 y culminará en el mes de Abril del 2012 con la sustentación y su posterior entrega. Este proyecto se lo realizará para la empresa IX Biotron, pero para su uso, se lo implementará en las instalaciones del Hospital Carlos Andrade Marín, esto ya que le empresa tiene contrato por un año con el Hospital y cuenta con una sucursal en el mismo.



Ubicación Geográfica del Hospital

Gráfico N°: 01

Fuente: Internet



1.4 Objetivos

Con el sistema lo que se desea obtener es dar solución a la problemática que lleva la empresa IX Biotron en cuanto a los procesos de mantenimiento.

1.4.1 Objetivo General

Diseñar, desarrollar e implementar un Sistema de Mantenimiento de Equipos Hospitalarios, de calidad dando todas las facilidades a la empresa IX Biotron, a fin de mejorar los procesos de mantenimiento que se lleva dentro de la empresa.

1.4.2 Objetivos Específicos

- Analizar cada una de los procesos de mantenimientos a fin satisfacer las necesidades del usuario.
- Diseñar procesos para reducir el tiempo que en la actualidad se llevan las órdenes de trabajo.
- Desarrollar un sistema de confiabilidad en seguridades y claves para la privacidad de información.
- Implementar un sistema que controle el Mantenimiento de los Equipos Hospitalarios, dentro la institución auspiciante.
- Capacitar al usuario final para obtener mejores beneficios de la aplicación.

1.5 Justificación e Importancia

1.5.1 Justificación

El sistema permitirá controlar de una mejor manera el estado en que se encuentran los equipos, sea estos que se encuentren en estado operativo o en proceso de



reparación, a su vez se podrá emplear de mejor manera el recurso humano que se cuenta dentro de la empresa, ya que se minimizará el tiempo de espera en búsquedas de historiales de los equipos, realización de cronogramas, utilización de inventarios, etc.

El sistema se justifica de una manera adecuada, ya que por parte de la empresa auspiciante es primordial contar con un Sistema de Mantenimiento de Equipos

Hospitalarios, ya que fué uno de los requisitos cuando ganó el contrato con el Hospital Carlos Andrade Marín.

1.5.2 Importancia

Es pertinente realizar este sistema en la empresa **IX Biotron**, ya que la empresa no cuenta con un sistema que controle los procesos de mantenimiento, a su vez la realización del proyecto es factible ya que la Empresa cuenta con todos los recursos necesarios para la realización del Proyecto.

El sistema es muy relevante dentro de la Empresa, ya que uno de los requisitos al momento de ganar el contrato con el Hospital "Carlos Andrade Marín", es presentar y utilizar un sistema que controle los procesos internos de mantenimiento.

1.6 Alcance

Desarrollar un sistema basado en arquitectura tres capas las cuales se detallan en la parte inferior.

- Capa de presentación: esta capa presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso esta va será la interfaz gráfica la cual será muy amigable.



- Capa de negocio: esta capa permite que residan los programas que se ejecutan, adicional recibe las peticiones del usuario y se envían las respuestas tras el proceso, es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.
- Capa de datos: es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

El sistema se convertirá en una herramienta fundamental para el departamento de mantenimiento de equipo biomédico, ya que permitirá el seguimiento y control de cada equipo dentro del Hospital.



CAPITULO II

MARCO TEÓRICO

2.1 Antecedentes

Con la Implementación de un Sistema de Mantenimiento de Equipos Hospitalarios para la empresa IX Biotron, esperamos llenar todos los requerimientos necesarios para un buen manejo de los mantenimientos preventivos y correctivos, además que mucho de los procesos conlleva realizar un buen análisis.

2.2 Reseña Histórica

IX BIOTRON DEL ECUADOR CIA. LTDA. Está constituida legalmente desde 1991 precedida por el Ing. Mauricio Suasnavas Mera. Desde su fundación puso al servicio toda su experiencia en la venta y mantenimiento de Equipo Médico, así como también el servicio personalizado en diseño y construcción de bioinstrumentos.

2.2.1 Misión

Llegar a establecer un campo de acción tecnológico que satisfaga al usuario de vanguardia dentro y fuera de los límites nacionales, integrando la experiencia y el conocimiento en el campo de la bioingeniería con innovadores diseños biomédicos tanto en la comercialización de equipos médicos como en el desarrollo de productos propios.

2.2.2 Visión

Brindar un servicio de calidad y tecnología de vanguardia con seriedad, puntualidad y honestidad. Donde cuenta el valor humano con el menor riesgo posible,



implementando estrategias de alto nivel en la comercialización, mantenimiento, capacitación y diseño en el área de la ingeniería biomédica.

2.3 Marco Referencial

Ya identificado el problema según las necesidades de la empresa se realiza los respectivos estudios e investigaciones para poder validar la viabilidad del proyecto y verificar bajo que plataforma se va a trabajar y en cual trabaja la empresa, con que lenguaje se va a realizar la programación, que software se va a utilizar para diseñar la base de datos y el motor de base de datos que se va a ocupar.

2.3.1 Hardware

Se realiza un análisis de la estructura informática que la empresa necesita en lo que refiere a hardware para lo cual se toma en cuenta computadores, servidores.

Servidor es una computadora que forma parte de una red, provee servicios a otras computadoras denominadas clientes. También se suele denominar con la palabra servidor a:

- Una aplicación informática o programa que realiza algunas tareas en beneficio de otras aplicaciones llamadas clientes. Algunos servicios habituales son los servicios de archivos, que permiten a los usuarios almacenar y acceder a los archivos de una computadora y los servicios de aplicaciones, que realizan tareas en beneficio directo del usuario final. Este es el significado original del término. Es posible que un ordenador cumpla simultáneamente las funciones de cliente y de servidor.
- Una computadora en la que se ejecuta un programa que realiza alguna tarea en beneficio de otras aplicaciones llamadas clientes, tanto si se trata de



un ordenador central (*mainframe*), un miniordenador, un ordenador personal, una PDA o un sistema embebido; sin embargo, hay computadoras destinadas únicamente a proveer los servicios de estos programas: estos son los servidores por antonomasia.

- Un servidor no es necesariamente una máquina de última generación de grandes proporciones, no es necesariamente un superordenador; un servidor puede ser desde una computadora vieja, hasta una máquina sumamente potente (ej.: servidores web, bases de datos grandes, etc. Procesadores especiales y hasta varios terabytes de memoria). Todo esto depende del uso que se le dé al servidor. Si usted lo desea, puede convertir al equipo desde el cual usted está leyendo esto en un servidor instalando un programa que trabaje por la red y a la que los usuarios de su red ingresen a través de un programa de servidor web como Apache.

Por lo cual podemos llegar a la conclusión de que un servidor también puede ser un proceso que entrega información o sirve a otro proceso. El modelo Cliente-servidor no necesariamente implica tener dos ordenadores, ya que un proceso cliente puede solicitar algo como una impresión a un proceso servidor en un mismo ordenador.

2.3.2 Software

Es importante analizar y conocer la estructuración de software que se implantará en la empresa para el desarrollo del proyecto, así como también los términos utilizados en el mismo.

- **Rational Rose:** define un conjunto de propiedades para los elementos de UML que describen sus atributos. Estas propiedades se definen en los archivos .pty. De manera similar, los estereotipos para los elementos de UML se definen en los archivos .ini. Rational Software Architect define las propiedades y los estereotipos



de los elementos de UML usando perfiles de UML. Un perfil de UML se define como un paquete que se almacena como un archivo con una extensión .epx. Incluye un conjunto estándar de perfiles de UML, pero estos perfiles no incluyen muchos de los estereotipos y de las propiedades de Rational Rose. En consecuencia, cuando se importa un modelo de Rational Rose a Rational Software Architect, todas las propiedades y los estereotipos que se encuentran en el modelo de Rational Rose pero que no están disponibles en Rational Software Architect son importados como perfiles de UML.

Subunidades y fragmentos: Rational Rose puede usar uno o más archivos para dividir y almacenar los modelos. Si un modelo ha sido dividido en distintos archivos, los archivos que no sean el principal (.mdl) se denominan subunidades.

De manera similar, Rational Software Architect también puede usar uno o más archivos para dividir y almacenar modelos, pero si un modelo ha sido dividido en distintos archivos, los archivos que no sean el principal (.emx) se denominan fragmentos.

Las subunidades de un modelo de Rational Rose pueden importarse a Rational Software Architect como fragmentos. La división de un modelo en subunidades o fragmentos permite que el equipo de un proyecto trabaje y desarrolle un modelo en paralelo.

- **UML "Lenguaje Unificado de Modelado"**: es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG(Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones



de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Es importante resaltar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional o RUP), pero no especifica en sí mismo qué metodología o proceso usar.

UML no puede compararse con la programación estructurada, pues UML significa Lenguaje Unificado de Modelado, no es programación, solo se diagrama la realidad de una utilización en un requerimiento. Mientras que, programación estructurada, es una forma de programar como lo es la orientación a objetos, sin embargo, la programación orientada a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos.

- **SharePoint 2010:** es la plataforma de colaboración empresarial que le permite incrementar la productividad y administrar los contenidos a través de la conocida interfaz de Office. Tanto si implementa SharePoint 2010 de forma local como si lo hace en servicios hospedados, los recursos integrados, mejorados por las tecnologías de búsqueda, le permitirán responder con rapidez frente a las cambiantes necesidades empresariales. Tome decisiones bien informadas e implemente soluciones personalizadas de forma rápida y segura para mejorar la colaboración en toda la empresa y con terceros. La consolidación de soluciones mediante SharePoint 2010 contribuye a ahorrar mediante la reducción de los gastos de mantenimiento y formación y a través del aumento de la productividad del



departamento de TI. Todo gracias a una plataforma que ofrece un control total y además cumple con las normativas.

- **Microsoft SharePoint Designer:** es un editor web tipo WYSIWYG para el diseño de sitios web desarrollado por Microsoft exclusivos para sitios SharePoint y parte de los productos de la familia SharePoint.

Esta aplicación junto con Expression Web, son los reemplazantes de FrontPage. Comparte el motor de renderizado con Expression Web, que es diferente al Trident de Internet Explorer. La diferencia entre SharePoint Designer y Expression Web, radica en que la segunda es una aplicación de diseño web general, mientras que la primera es exclusiva para sitios SharePoint.

Desde el sitio web de Microsoft se indica que: "Office SharePoint Designer ofrece herramientas para automatizar los procesos empresariales, crear aplicaciones eficaces sobre la plataforma Microsoft SharePoint y ajustar el sitio SharePoint a necesidades particulares".

- **Microsoft SQL Server:** es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional. Sus lenguajes para consultas son T-SQL y ANSI SQL. Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle, PostgreSQL o MySQL.

2.3.3 Know how

Para el análisis, diseño, desarrollo del proyecto es bueno mencionar que se tiene un gran conocimiento con experiencia en estudios y trabajos realizados con anterioridad lo que facilitara el trabajo en este proyecto convirtiéndose más sencillo el entender las necesidades del cliente y plasmarlas en un sistema.



2.4 Marco Legal

2.4.1 Ley de la Propiedad Intelectual

Como parte legal de la creación del proyecto de grado, así como también la propiedad del mismo por parte del estudiante, se han contemplado los siguientes artículos para salvaguardar los derechos del autor.

Sección II.-

Objeto del Derecho de Autor.-

Art. 8. La protección del derecho de autor recae sobre todas las obras del ingenio, en el ámbito literario o artístico, cualquiera que sea su género, forma de expresión, mérito o finalidad. Los derechos reconocidos por el presente Título son independientes de la propiedad del objeto material en el cual está incorporada la obra y su goce o ejercicio no están supeditados al requisito del registro o al cumplimiento de cualquier otra formalidad.

Las obras protegidas comprenden, entre otras, las siguientes:

Libros, folletos, impresos, epistolarios, artículos, novelas, cuentos, poemas, crónicas, críticas, ensayos, misivas, guiones para teatro, cinematografía, televisión, conferencias, discursos, lecciones, sermones, alegatos en derecho, memorias y otras obras de similar naturaleza, expresadas en cualquier forma;

Colecciones de obras, tales como antologías o compilaciones y bases de datos de toda clase, que por la selección o disposición de las materias constituyan creaciones intelectuales, sin perjuicio de los derechos de autor que subsistan sobre los materiales o datos;

Proyectos, planos, maquetas y diseños de obras arquitectónicas, de ingeniería, ilustraciones, gráficos, mapas y diseños relativos a la geografía, la topografía, y en general a la ciencia; Programas de ordenador; y Adaptaciones, traducciones,



arreglos, revisiones, actualizaciones y anotaciones; compendios, resúmenes y extractos; y, otras transformaciones de una obra, realizadas con expresa autorización de los autores de las obras originales, y sin perjuicio de sus derechos.

Sección V.-

Disposiciones Especiales sobre ciertas Obras

Parágrafo Primero

De los Programas de Ordenador

Art. 28. Los programas de ordenador se consideran obras literarias y se protegen como tales. Dicha protección se otorga independientemente de que hayan sido incorporados en un ordenador y cualquiera sea la forma en que estén expresados, ya sea en forma legible por el hombre (código fuente) o en forma legible por máquina (código objeto), ya sean programas operativos y programas aplicativos, incluyendo diagramas de flujo, planos, manuales de uso, y en general, aquellos elementos que conformen la estructura, secuencia y organización del programa.

Art. 29. Es titular de un programa de ordenador, el productor, esto es la persona natural o jurídica que toma la iniciativa y responsabilidad de la realización de la obra. Se considerará titular, salvo prueba en contrario, a la persona cuyo nombre conste en la obra o sus copias de la forma usual.

Dicho titular está además legitimado para ejercer en nombre propio los derechos morales sobre la obra, incluyendo la facultad para decidir sobre su divulgación.

El productor tendrá el derecho exclusivo de realizar, autorizar o prohibir la realización de modificaciones o versiones sucesivas del programa, y de programas derivados del mismo.

Las disposiciones del presente artículo podrán ser modificadas mediante acuerdo entre los autores y el productor.



Art. 30. La adquisición de un ejemplar de un programa de ordenador que haya circulado lícitamente, autoriza a su propietario a realizar exclusivamente:

Una copia de la versión del programa legible por máquina (código objeto) con fines de seguridad o resguardo;

Fijar el programa en la memoria interna del aparato, ya sea que dicha fijación desaparezca o no al apagarlo, con el único fin y en la medida necesaria para utilizar el programa; y, salvo prohibición expresa, adaptar el programa para su exclusivo uso personal, siempre que se limite al uso normal previsto en la licencia. El adquirente no podrá transferir a ningún título el soporte que contenga el programa así adaptado, ni podrá utilizarlo de ninguna otra forma sin autorización expresa, según las reglas generales.

Se requerirá de autorización del titular de los derechos para cualquier otra utilización, inclusive la reproducción para fines de uso personal o el aprovechamiento del programa por varias personas, a través de redes u otros sistemas análogos, conocidos o por conocerse.

Art. 31. No se considerará que exista arrendamiento de un programa de ordenador cuando éste no sea el objeto esencial de dicho contrato. Se considerará que el programa es el objeto esencial cuando la funcionalidad del objeto materia del contrato, dependa directamente del programa de ordenador suministrado con dicho objeto; como cuando se arrienda un ordenador con programas de ordenador instalados previamente.

Art. 32. Las excepciones al derecho de autor establecidas en los artículos 30 y 31 son las únicas aplicables respecto a los programas de ordenador.

Las normas contenidas en el presente Parágrafo se interpretarán de manera que su aplicación no perjudique la normal explotación de la obra o los intereses legítimos del titular de los derechos.



CAPITULO III

INVESTIGACIÓN CIENTÍFICA

3.1 Tipos de Investigación

3.1.1 Investigación Cuantitativa

Bibliográfica Documental

Para el estudio del Sistema de Mantenimiento de Equipos Hospitalarios, se utilizará las distintas normativas hospitalarias necesarias para la realización del periodismo de los distintos tipos de mantenimiento a través de ecuaciones enfocadas al tema y sugerencias de los distintos tipos de manuales técnicos de los equipos.

3.1.2 Investigación de Campo

Investigación Descriptiva

A través de la investigación se encontró el problema que tiene al momento la empresa, que es de llevar de manera manual los cronogramas de mantenimiento preventivo y correctivo. Para solucionar este problema y guiarnos correctamente haremos un análisis del tiempo periódico para cada mantenimiento a través de normativas hospitalarias relacionadas al caso, ayudándonos posteriormente a medir la calidad y cantidad de trabajo en periodos concernientes a la empresa.

Investigación Explicativa

El problema que tiene la empresa al momento de realizar cronogramas de mantenimiento. Se da mucho antes que esta gane el contrato actual con el Hospital Carlos Andrade Marín, en cual se realizaban los cronogramas ha criterio personal de la persona encargada del mantenimiento preventivo y correctivo, causando a largo



tiempo daños por la excesiva manipulación interna del equipo, así como también el mal uso del tiempo, desplazando muchos equipos para el mantenimiento al final del periodo. A lo largo de la investigación lograremos concretar los puntos necesarios para solucionar las falencias heredadas que tiene actualmente la empresa.

3.2 Métodos de Investigación

3.2.1 Histórico - Lógico

El problema actual que tiene la empresa auspiciante fue heredado de las anteriores empresas que trabajaban prestando servicios profesionales hacia el Hospital Carlos Andrade Marín, ya que estas no tenían un software para llevar de una manera adecuada los procedimientos de mantenimiento preventivo y correctivo.

3.2.2 Analítico – Sintético

Mantenimiento Correctivo	Llamada hacia mantenimiento por problemas en el equipo.	Detección de problemas en el equipo por parte de enfermera encargada del mismo.
	Generación de tarjetas amarillas.	Una vez detectado el problema por parte de la enfermera encargada del equipo se dispone a realizar una tarjeta amarilla para mantenimiento, ya que esta es una constancia para el trabajo a realizarse.
	Visita técnica al servicio por parte de mantenimiento.	El momento que ingresa una tarjeta amarilla al departamento m de mantenimiento, este dirige al técnico más



		idóneo para solucionar el problema en dicho servicio.
	Determinación de problemas en el equipo.	Al momento de revisar el equipo, el técnico encargado determina el problema en el mismo de una manera más técnica y objetiva.
	Corrección de averías en el equipo.	Una vez determinado el problema, el técnico determina si es necesario dar un mantenimiento correctivo en el mismo lugar o si es necesario dirigirlo al equipo hacia el servicio de mantenimiento.
	Generación de orden de servicio.	Solucionados los problemas en el equipo, se procede a realizar una orden de servicio. En el cual se desglosa el trabajo realizado en el mismo.

3.2.3 Inductivo

Al visualizar el resultado de la información que necesitamos podemos plantearnos el proceso y los procedimientos que vamos a manejar para poder realizar el análisis, el diseño, hasta llegar al desarrollo concluyendo con la implementación y sus pruebas respectivas.

3.2.4 Deductivo



Este método fue utilizado para estudiar las actividades que realiza el personal día a día, adicional permitió conocer el funcionamiento de los departamentos los procesos que maneja la empresa, la estructura que tiene establecida y las funciones de cada persona a cargo.

3.3 Herramientas de recolección de información

3.3.1 Observación

Al momento la empresa no cuenta con un departamento de sistemas, pero tiene una pequeña red interna. En la cual se encuentra conectado; dos computadoras de escritorio, una laptop, y una impresora.

La empresa no cuenta con un software que permita controlar el trabajo realizado, por lo que se lo lleva a cabo en Excel.

Esto a su vez dificulta, el control de órdenes de trabajo preventivo y correctivo, tiempos muertos de trabajo, asignación de tarjetas amarillas, control de materiales, etc.

3.3.2 Entrevista

- Pregunta: ¿Cómo se generan las tarjetas amarillas?

Respuesta: Las tarjetas amarillas se generan de acuerdo a las necesidades de cada equipo, por parte de secretaria de las distintas áreas.

Análisis: La generación de las tarjetas amarillas, se lo realiza de acuerdo al daño o falla del equipo médico en las distintas áreas del hospital. Este es un formato interno que lleva al Hospital Carlos Andrade Marín para solucionar sus problemas.

- Pregunta: ¿Cómo se generan las órdenes de trabajo?



Respuesta: Las órdenes de trabajo se generan una vez realizado el mantenimiento preventivo o correctivo.

Análisis: Para la creación de las órdenes de trabajo, los Técnicos encargados de los mantenimientos preventivos y correctivos deben haber cumplido dichos trabajos en el tiempo programado.

- Pregunta: ¿La empresa cuenta con una intranet necesaria para el proyecto?

Respuesta: Si. Si cuenta con una intranet, que la ocupan solo para compartir pequeños documentos.

Análisis: Se encontró que la intranet actual de la empresa es una base fundamental para la implementación del proyecto de grado ya que la proyección es entregar una laptop a cada técnico distribuido en el hospital, los cuales deberán ingresar los datos de cada mantenimiento preventivos y correctivos al finalizar el mismo.

- Pregunta: ¿La empresa cuenta con un departamento de informática para el mantenimiento del sistema?

Respuesta: No. No cuenta con un departamento de sistemas que respalde la buena utilización del sistema.

Análisis: Se encontró que la empresa no cuenta con un departamento de sistemas el cual dificulta un poco el soporte técnico. La solución más adecuada por parte del estudiante es dar el soporte técnico del mismo en el periodo de contrato con el Hospital Carlos Andrade Marín.

CAPITULO IV



DESARROLLO DE LA PROPUESTA

4.1 Diagnostico Situacional

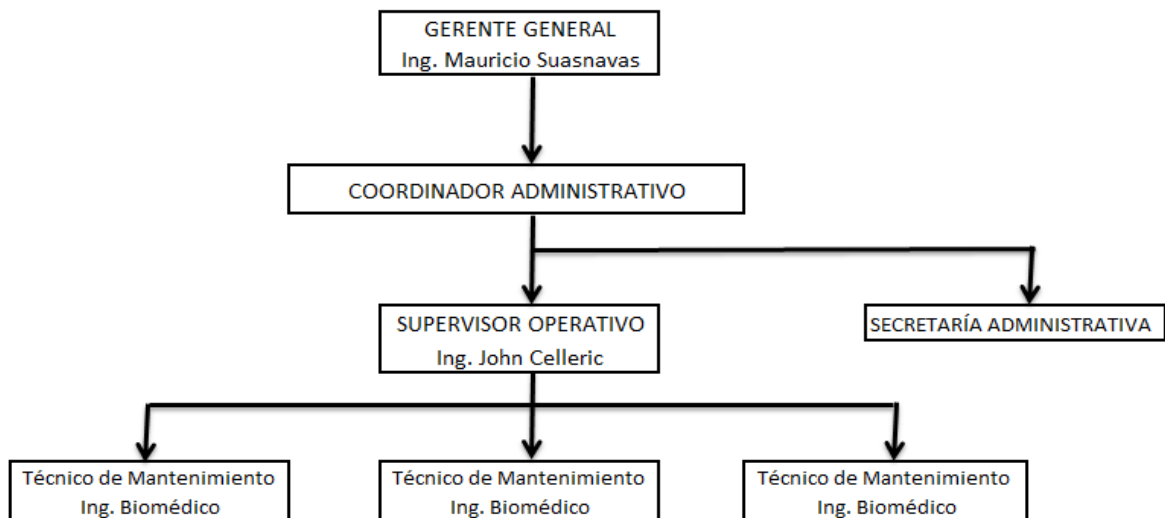
Al momento la empresa auspiciante, no cuenta con un departamento de sistemas, pero tiene una pequeña red interna. En la cual se encuentra conectado; dos computadoras de escritorio, una laptop, y una impresora.

La empresa no cuenta con un software que permita controlar el trabajo realizado, por lo que se lo lleva a cabo en Excel.

Esto a su vez dificulta, el control de órdenes de trabajo preventivo y correctivo, tiempos muertos de trabajo, asignación de tarjetas amarillas, control de materiales, etc.

4.2 Estructura Organizacional

- Estructura Gerencial





La estructura organizacional de la empresa auspiciante se encuentra dividida en dos partes. La matriz y sucursal HCAM, con su cabeza principal el Gerente General Ing. Mauricio Suasnavas.

Estructura Gerencial

Gráfico N°: 02

Fuente: IX Biotron

- Estructura organizacional de mantenimiento en HCAM por IX Biotron

Anexo: 2

Título: Estructura Organizacional

Fuente: IX BIOTRON

4.3 Infraestructura Informática

4.3.1 Hardware

HARDWARE			
CARACTERÍSTICAS	S.O.	TIPO	UBICACIÓN
Intel CoreI3-2100	Windows 7	Estación (Escritorio)	Secretaría
Intel CoreI7-2600	Windows 7	Estación (Escritorio)	Secretaría
Intel CoreI3-2100	Windows 7	Estación (Laptop)	Jefatura

Hardware



Tabla N°: 01

Fuente: IX Biotron

4.3.2 Software

SOFTWARE	
NOMBRE	DETALLE
Windows 7 Profesional	Licencia general
Microsoft Office 2007 Vs 12.0.4518	Licencia general
AutoCad Vs 2010	Licencia general
Adobe Reader Vs 9.3.3	Licencia general
Antivirus EsetNod32 Vs 4.0.314	Licencia general

Software

Tabla N°: 02

Fuente: IX Biotron

4.3.3 Comunicaciones

Router			
Cantidad	Marca	Modelo	Serie



1	D-link	DIR-615	IIR615ANA
---	--------	---------	-----------

Comunicaciones

Tabla Nº: 03

Fuente: IX Biotron

4.3.4 Recurso Humano

NOMBRE	DESCRIPCIÓN
Ing. Rómulo Arguello	Supervisor de mantenimiento IX Biotron
Ing. Paola Gómez	Coordinadora operativa
Ing. Margarita Barrionuevo	Técnico mantenimiento
Ing. Juan Fernando Vela	Técnico mantenimiento
Ing. Diego Guerrero	Técnico mantenimiento
Ing. Andrés Yangári	Técnico mantenimiento
Tlgo. Carlos Plua	Técnico mantenimiento

Recurso Humano

Tabla Nº: 04

Fuente: IX Biotron



4.4 Descripción de Alternativas

Con la finalidad de encontrar la mejor solución para el problema actual que tiene la empresa se ha realizado la descripción de tres alternativas de solución.

- Alternativa 1: Soluciones Integrales OmnisoftCia. Ltda

CUADRO DE RECURSOS HUMANOS		
REQUERIMIENTO	CUMPLE	NO CUMPLE
ARQUITECTURA 3 CAPAS		X
METODOLOGÍA RUP		X
MODELO DE DESARROLLO	X	
CASOS DE USO		X
DIAGRAMA DE INTERACCIÓN	X	
DIAGRAMA DE CLASES		X
DIAGRAMA DE COLABORACIÓN	X	
DISEÑO LÓGICO	X	
DISEÑO FÍSICO	X	
DESARROLLO .NET(C#, ASP)	X	
PRUEBAS		X
INTEGRACIÓN	X	
VALIDACIÓN		X
SISTEMA	X	
SEGURIDAD		X
INTERFACE	X	
IMPLEMENTACIÓN	X	
CAPACITACIÓN TÉCNICA	X	
CAPACITACIÓN USUARIOS		X
ENTREGA MANUALES TÉCNICO USUARIO		X
CÓDIGO FUENTE	X	
TIEMPO DESARROLLO	X	
COSTO		X
GARANTÍA TÉCNICA		X
SOPORTE TÉCNICO		X

Alternativa 1



Tabla Nº: 05

Fuente: Investigación

- Alternativa 2: QuitoNet Cía. Ltda

CUADRO DE RECURSOS HUMANOS		
REQUERIMIENTO	CUMPLE	NO CUMPLE
ARQUITECTURA 3 CAPAS	X	
METODOLOGÍA RUP	X	
MODELO DE DESARROLLO		X
CASOS DE USO		X
DIAGRAMA DE INTERACCIÓN	X	
DIAGRAMA DE CLASES	X	
DIAGRAMA DE COLABORACIÓN	X	
DISEÑO LÓGICO	X	
DISEÑO FÍSICO		X
DESARROLLO .NET(C#, ASP)		X
PRUEBAS		X
INTEGRACIÓN	X	
VALIDACIÓN	X	
SISTEMA	X	
SEGURIDAD	X	
INTERFACE	X	
IMPLEMENTACIÓN		X
CAPACITACIÓN TÉCNICA		X
CAPACITACIÓN USUARIOS		X
ENTREGA MANUALES TÉCNICO USUARIO		X
CÓDIGO FUENTE		X
TIEMPO DESARROLLO		X
COSTO		X
GARANTÍA TÉCNICA		X
SOPORTE TÉCNICO		X

Alternativa 2



Tabla Nº: 06

Fuente: Investigación

- Alternativa 3: Proyecto de grado

CUADRO DE RECURSOS HUMANOS		
REQUERIMIENTO	CUMPLE	NO CUMPLE
ARQUITECTURA 3 CAPAS	X	
METODOLOGÍA RUP	X	
MODELO DE DESARROLLO	X	
CASOS DE USO	X	
DIAGRAMA DE INTERACCIÓN		X
DIAGRAMA DE CLASES	X	
DIAGRAMA DE COLABORACIÓN		X
DISEÑO LÓGICO	X	
DISEÑO FÍSICO	X	
DESARROLLO .NET(C#, ASP)	X	
PRUEBAS	X	
INTEGRACIÓN	X	
VALIDACIÓN	X	
SISTEMA	X	
SEGURIDAD	X	
INTERFACE	X	
IMPLEMENTACIÓN	X	
CAPACITACIÓN TÉCNICA	X	
CAPACITACIÓN USUARIOS	X	
ENTREGA MANUALES TÉCNICO USUARIO	X	
CÓDIGO FUENTE	X	
TIEMPO DESARROLLO	X	
COSTO	X	
GARANTÍA TÉCNICA	X	
SOPORTE TÉCNICO	X	

Alternativa 3



Tabla Nº: 07

Fuente: Investigación

4.5 Evaluación y Selección de Alternativas

Con la finalidad de evaluar correctamente las tres alternativas presentadas por los casos oferentes se han determinado parámetros a fin de que la evaluación se lo realice considerando aspectos, técnicos, económicos y operativos.

La primera acción que se ha considerado es la creación pesos (valores) únicos a cada uno de los ítems, dependiendo de la importancia de que tiene cada uno de ellos en el manejo del desarrollo de la ingeniería del software.

Igualmente se establecieron los siguientes parámetros y porcentajes como se detalla a continuación.

Técnico 70%

Económico 20%

Garantía Técnica 5%

Soporte Técnico 5%

Con estos porcentajes se alcanza una calificación técnica y operativamente razonable a fin de poder cumplir con las expectativas en la ejecución del proyecto "Sistema de Mantenimiento de Equipos Hospitalarios".

EVALUACIÓN TÉCNICA

REQUERIMIENTOS	PESOS	Alternativa 1	Alternativa 2	Alternativa 3
----------------	-------	---------------	---------------	---------------



ARQUITECTURA 3 CAPAS	5%	5%	5%	5%
METODOLOGÍA RUP	5%	5%	5%	5%
CASOS DE USO	5%	3%	5%	5%
DISEÑO LÓGICO	5%	3%	5%	5%
DISEÑO FÍSICO	5%	3%	2%	5%
DESARROLLO .NET(C#, ASP)	5%	3%	2%	3%
PRUEBAS	5%	5%	3%	5%
INTEGRACIÓN	5%	5%	4%	5%
VALIDACIÓN	5%	5%	5%	5%
SISTEMA	5%	5%	3%	5%
SEGURIDAD	5%	5%	2%	5%
INTERFACE	5%	5%	5%	5%
IMPLEMENTACIÓN	5%	5%	5%	5%
CAPACITACIÓN TÉCNICA	5%	5%	5%	5%
CAPACITACIÓN USUARIOS	5%	5%	3%	5%
CÓDIGO FUENTE	5%	5%	5%	5%
TIEMPO DESARROLLO	5%	5%	5%	5%
COSTO	5%	5%	5%	5%
GARANTÍA TÉCNICA	5%	5%	5%	5%
SOPORTE TÉCNICO	5%	5%	5%	5%
	100%	92%	85%	98%



Evaluación técnica

Tabla N°: 08

Fuente: Investigación

EVALUACIÓN ECONÓMICA

EMPRESA	PESOS	COSTO
Soluciones Integrales OmnisoftCia. Ltda	16	\$ 3000
QuitoNet Cía. Ltda	13	\$ 2000
Proyecto de Grado	20	\$ 0

Evaluación económica

Tabla N°: 09

Fuente: Investigación

4.6 Factibilidad Técnica

De acuerdo a las evaluaciones realizadas de las empresas oferentes se puede determinar claramente que el proyecto es factible realizarlo desde el punto de vista técnico, económico y tiempo de ejecución. Por lo tanto el orden de prelación final de las empresas oferentes en el proyecto "Sistema de Mantenimiento de Equipos Hospitalarios", quedo el orden de la siguiente manera.

- Como mejor alternativa con un 98 % de satisfacción es la alternativa N° 3 que pertenece al Proyecto de Grado.



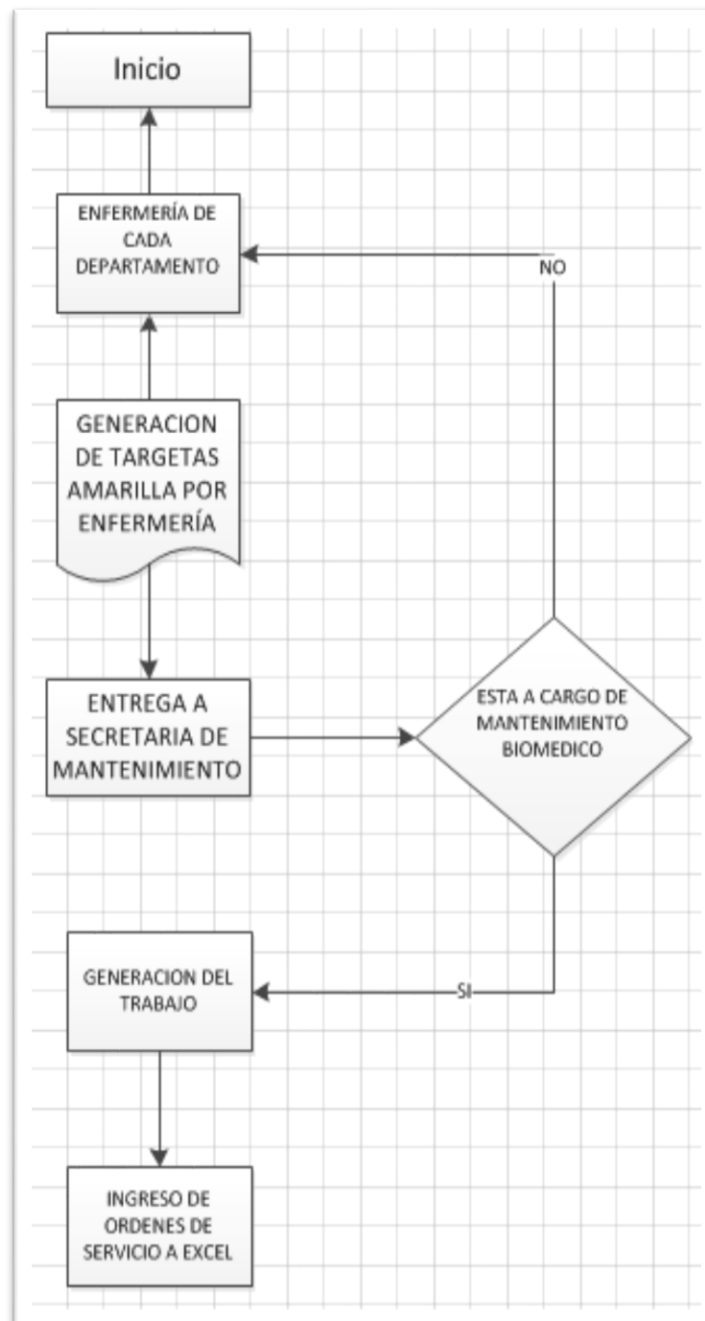
- La segunda mejor opción para la realización del proyecto con un 92 % de satisfacción es la alternativa Nº 1 que pertenece a la empresa Soluciones Integrales OmnisoftCia. Ltda.
- Y por último, la alternativa Nº 2 es la que produce menos aceptación con un 85 % perteneciente a la empresa QuitoNet Cía. Ltda.

En conclusión se determina que el proyecto se lo realizará con la más alta puntuación obtenida que corresponde al proyecto de grado.

4.7 Descripción de Procesos

- Generación de tarjetas amarillas

La generación de tarjetas amarillas se lo realiza por parte del Hospital Carlos Andrade Marín para poder solucionar los imperfectos de los equipos médicos, esto es muy esencial mencionar ya que esta es la orden para generar el trabajo de mantenimiento preventivo o correctivo dependiendo del caso que se amerite.



Generación tarjetas amarillas

Gráfico N°: 03

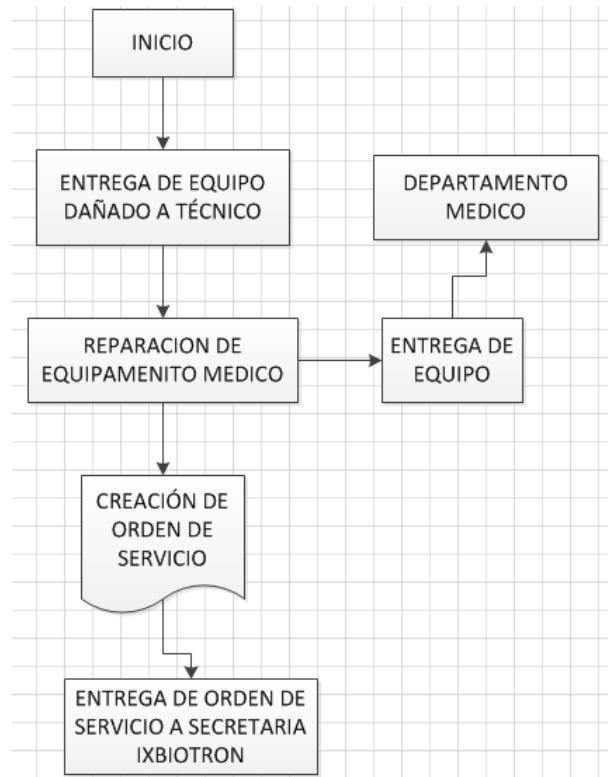
Fuente: IX Biotron



- Reparación de equipos

La reparación de los equipos se lo realiza cuando el equipo con fallas es llevado al área de mantenimiento biomédico. En esta área se lo realiza un previo chequeo y posteriormente su reparación.

Cuando el equipo se encuentra reparado, se crea una orden de servicio y se entrega el mismo al área correspondiente.



Reparación de equipos

Gráfico N°: 04

Fuente: IX Biotron



4.8 Descripción de la Metodología de Desarrollo

En el proceso unificado de desarrollo RUP, es la metodología que se utiliza para el desarrollo de software de tres capas lo que inducirá en analizar todo el conjunto de actividades necesarias para convertir los requerimientos necesarios de un usuario en un sistema.

Esta metodología por su gran flexibilidad se adapta a todo tipo de sistemas (multicapa), áreas, tipos de organización, y tamaños de proyectos. Se basa en componentes interconectados a través de interfaces y utiliza el lenguaje Unificado de Modelado (UML) para preparar todos los procesos de diagramas necesarios en el desarrollo del proyecto "Sistema de Mantenimiento de Equipos Hospitalarios".

Por consiguiente el objetivo general se estructurará de la siguiente manera.

Objetivo General:

Aplicar el proceso unificado de desarrollo en la construcción del "Sistema de Mantenimiento de Equipos Hospitalarios".

Objetivos Específicos:

- Identificar y establecer interacciones para casos de usos.
- Analizar el dominio del problema.
- Establecer una base arquitectónica sólida del sistema.
- Eliminar los elementos de alto riesgo del proyecto.
- Elaborar los casos de uso de acuerdo al plan de interacciones.
- Desarrollar y probar el software.



Con estas premisas iniciaremos el análisis y síntesis de las fases que constituyen la metodología RUP versus los flujos de trabajos estandarizados en el desarrollo del software de tres capas.

FASE DE INICIO

En la fase de inicio se realizó una gran actividad en el levantamiento de requerimientos y validación de los mismos manteniendo una organización partitiva de todos los actores involucrados en el "Sistema de Mantenimiento de Equipos Hospitalarios", el proceso consistió en diseñar ciertas estrategias de recopilación de información través de formularios, entrevistas y observaciones directas de todos los procesos y procedimientos que se realizan en cada uno de los sitios de trabajo, es lógico suponer que para realizar esta actividad se tuvo con anterioridad que inteligenciarnos de todos los procesos que se cumplen en la empresa; por tal motivo con conocimiento de causa se validaron requerimientos de usuario y validación de actividades; como conclusión de esta actividad se puede señalar que los requerimientos se lo establecieron en el manejo de procesos, equipamiento de hardware y cultura informática de cada uno de los usuarios.

En lo referente a la fase de inicio y flujo de trabajo de análisis se determinaron inicialmente las áreas críticas sobre las que tenemos que trabajar, a fin de mantener la continuidad del negocio; también se establecieron las tareas y actividades que son susceptibles de automatización. En este análisis se determinó que algunas de las tareas pueden funcionar dentro del sistema de una forma semiautomática lo que implica que este tipo de tareas solo ingresaran a formar parte directa del diseño del sistema.

También se determinó el alcance del sistema y los requerimientos de hardware necesarios para que soporte el "Sistema de Mantenimiento de Equipos Hospitalarios" con una visión del servidor de aplicaciones y de datos del funcionamiento y valides de por lo menos cinco años.



En la parte de inicio del diseño se determinó un esbozo inicial de cómo quedaría la estructura del diseño del sistema realizando un breve análisis de los casos de uso, de los diagramas de iteraciones, de los diagramas de colaboración y del diagrama de clases. Lo que determinaron que se identifiquen los subsistemas más importantes a desarrollarse. También cabe indicar que en esta parte se identificaron los riesgos más importantes del proyecto; se planificó en detalle las fases de elaboración y se estimó el proyecto de una manera aproximada.

FASE DE ELABORACIÓN

En la fase de elaboración aún siguen manteniendo vigencia ya que es el insumo que constantemente se encuentra retroalimentando la información al desarrollo del sistema a medida que se va avanzando el proyecto, los requerimientos van cumpliendo con la actividad de verificadores de necesidades reales de líneas de códigos en determinados procesos. Pero al mismo tiempo han alimentado de suficiente información para que el análisis se lo realice de una manera completa con un enfoque global del "Sistema de Mantenimiento de Equipos Hospitalarios", inclusive analizando las necesidades externas ya que las mismas requieren de flujos de información que tendrán que ser diseñadas en sus respectivos procesos.

También se estructura y dimensiona el control de información para poder determinar la capacidad de los discos duros y dispositivos de almacenamiento, también se analizó el número de transacciones y el tipo de ejecución de cada una de ellas a fin de establecer el hardware, velocidad de procesamiento y cantidad de memoria RAM y CACHE para cumplir estas actividades.

En esta fase tiene mucha importancia el diseño del sistema ya que es la misma o vamos a iniciar con la diagramación de los insumos necesarios para poder obtener la lógica de negocio de una manera coherente.



Iniciamos este proceso con el diseño y estructuración de los casos de uso que permitirá determinar todas las tareas involucradas y los actores responsables de cada una de ellos. Posteriormente se realizara el diseño de diagrama de clases en la que se colocará la estructura estática del negocio esto tomando en consideración que esta clase en la generación del script de la base de datos, se comunicara en las tablas relacionadas en la que tendrían que agregarle el código de programación.

En el diagrama de colaboración analizamos las relaciones que se manejan entre las distintas clases ya que las mismas nos permitirán manejar las interacciones dinámicas y que ellos manipulan así ellos indican como son revisados los comportamientos en el diagrama de actividad.

En el diagrama de actividad describimos el comportamiento del negocio o flujo de trabajo del negocio.

Con todos estos insumos se procede a la estructuración y diagramación de la lógica del negocio indicando como referencia que en las clases no tenemos que colocar las operaciones que se ejecutan en cada una de ellas este mismo diagrama se convertirá en diagrama físico y posteriormente generamos el script de la base de datos.

En lo referente a la implementación de esta fase ya comenzamos con la ejecución de misma ya que el generar la base de datos con la finalidad de probar la consistencia de la misma realizaremos el módulo de seguridad y módulo de mantenimiento.

Las pruebas también comienzan a tener su auge ya que se determinaran prueba de interacción, validación, en los módulos anteriormente elaborados.



FASE DE CONSTRUCCIÓN

En la fase de construcción los requerimientos ya han sido considerados en su totalidad por lo tanto su aporte ya no tendrá la misma inconsistencia que otras fases; en lo referente al análisis este flujo de trabajo viene a constituir la materia prima que se necesita en la reestructuración del sistema especialmente para poder determinar el diseño final del sistema.

La parte de construcción que es la estructuración de la línea base de la arquitectura crece hasta convertirse en el sistema completo. La descripción que evoluciona hasta convertirse en producto preparado para ser entregado a la comodidad del usuario en esta fase con la generación del modelo físico, lógico y script de base de datos y al final de esta parte el producto contiene todos los casos de uso que la dirección y el cliente han encontrado para el desarrollo del "Sistema de Mantenimiento de Equipos Hospitalarios".

Sin embargo, puede que no esté completamente libre de defectos muchos de estos defectos se trataran de solucionar durante la fase de transición.

En la fase de construcción la implementación tiene su mejor auge ya que se inicia en gran escala el desarrollo de todo el sistema es decir el módulo de seguridad, mantenimiento, reglas de negocio, que estarán reflejadas a través de 3 capas: presentación, lógica y reglas de negocio.

Cabe señalar que la construcción se lo realizara con el manejo de clases y métodos los mismos que permitirán estructurar de mejor manera el código necesarios para que cumplan el objetivo de casa una de las capas antes mencionadas.

Las pruebas que se tienen que realizar en esta fase son de mayor intensidad ya que debemos realizar pruebas de unidad, validación, integración, seguridad, sistema, interfaz



FASE DE TRANSICIÓN

En la fase de transición se cubre todo el periodo mediante el cual el sistema se convertirá en un número reducido de usuarios que se utilizó para verificar la consistencia, las pruebas que se realizaron para el manejo de la información y flujo de datos tiene origen y se incorporan algunas mejoras sugeridas en una lección general dirigidas a la totalidad de la comunidad de usuarios.

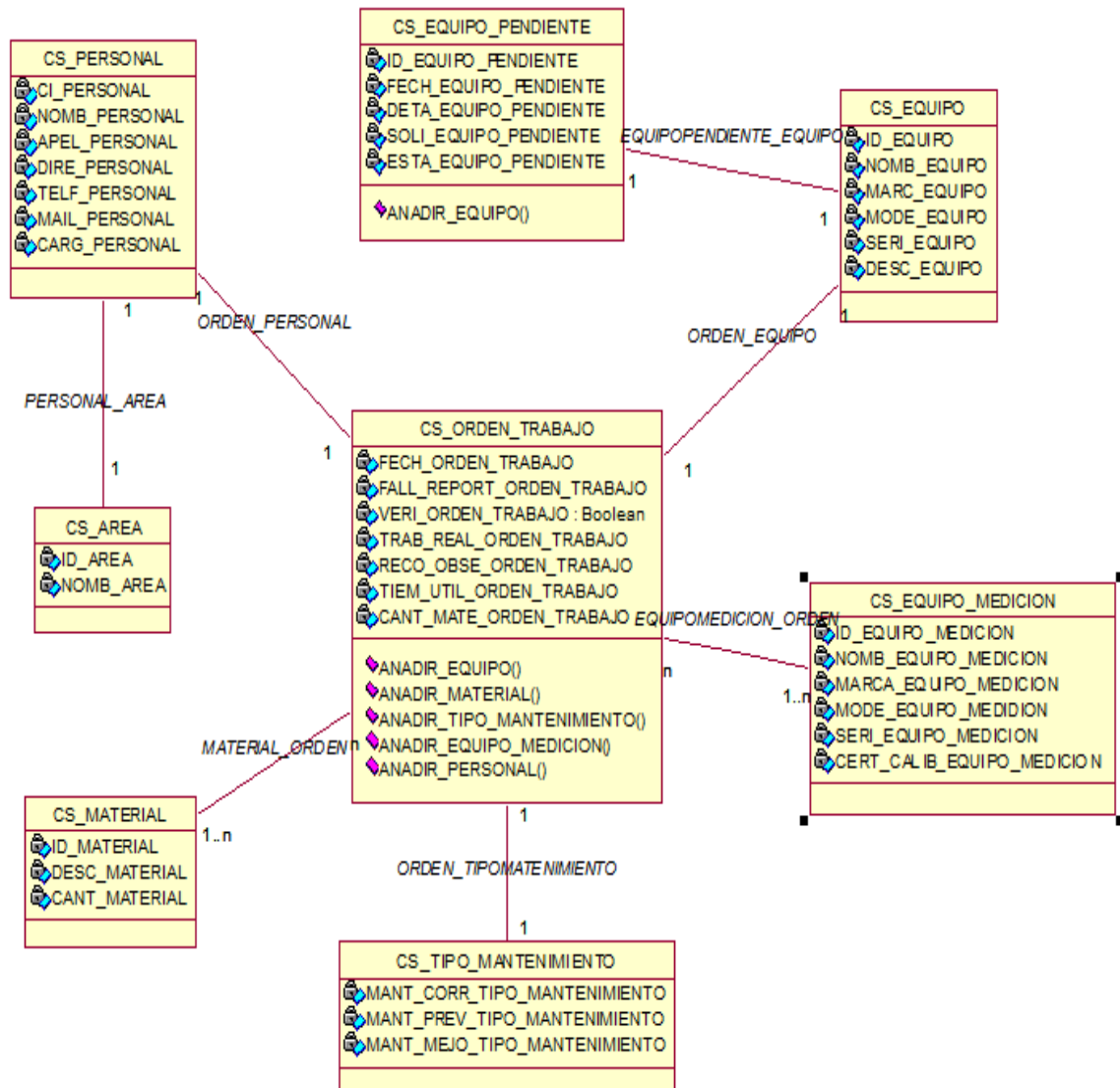
En esta fase igualmente las pruebas se las realiza a fin de pulir y poner a punto el sistema en las mejores condiciones y luego de haber incorporado estándares reducción de código de programación, el manejo de sus características como herencia, polimorfismo, encapsulamiento, determinando la calidad y eficiencia del desenvolvimiento al ejecutar.

4.9 Modelo Conceptual

DG_LOGICO_MANTENIMINETO

Para la creación de las tablas necesarias para el sistema, se realizó un previo análisis dentro de la empresa en el cual se llegó a necesitar distintas tablas como Personal, Equipos, Equipo Pendientes, Áreas, Órdenes de trabajo, Tipos de Mantenimientos, Materiales.

Estas tablas nos ayudarán a crear correctamente el sistema.



DG_LOGICO_MANTENIMIENTO

Nº: 05

Fuente: IX Biotron



DG_FISICO_MANTENIMINETO

Gráfico N°: 06

Fuente: Rational Rose

4.11 Diccionario de Datos

<<Schema>> S_0

Parent Package	Schemas	Assigned Component Package	--Not Defined--
Global	No		

Classes	T_1	T_CS_EQUIPO_MEDICION	T_0
T_CS_PERSONAL	T_CS_AREA	T_CS_EQUIPO_PENDIENTE	T_CS_EQUIPO
T_CS_ORDEN_TRABAJO	T_CS_TIPO_MANTENIMIENTO	T_CS_MATERIAL	

Property Settings

Data Modeler

dmlItem	True	IsSchema	True
dmSchema	Data Modeler	ShortenNameEngine	True
DMName	S_0	CodeName	
DatabaseID	4F69A02F00FE	SourceId	4F69534803D4













SourceType	omPackage	dmDomainPackage	
IsDomainPackage	False	IsRootSchema	False
IsRootDomainPackage	False	IsSchemaPackage	False
DBMS			

<<Table>> T_0 {Data Modeler}

Parent Package	S_0	Abstract	No
Export Control	PublicAccess	Link Class for	None
Class Kind	NormalClass	Cardinality	n
Space		Concurrency	Sequential
Persistence	No		

Assigned Components	DB_MANTENIMIENTO
----------------------------	-------------------------

Operations

Name	Signature	Class
 PK_T_08	 <<PK>> PK_T_08 ()	T_0
 FK_T_06	 <<FK>> FK_T_06 ()	T_0
 FK_T_07	 <<FK>> FK_T_07 ()	T_0
 TC_T_09	 <<Index>> TC_T_09 ()	T_0
 TC_T_010	 <<Index>> TC_T_010 ()	T_0



Attributes

Name	Class	Type	Initial Value
PK T_CS_ORDEN_TRABAJO_ID	T_0	INT	
PK T_CS_MATERIAL_ID	T_0	INT	

Associations

Name	My Role	My Class	Other Role	Other Element
--Not Named--	--Not Named--	T_0	--Not Named--	T_CS_ORDEN_TRABAJO
--Not Named--	--Not Named--	T_0	--Not Named--	T_CS_MATERIAL

Property Settings

Data Modeler

dmlItem	True	IsTable	True
DMName	T_0	CodeName	
TableSpaceID		SourceId	666
SourceType	n:n Association	HasLikeStatement	False
LikeTableName		LikeIncludeIdentity	False
LikeColumnAttr	False	IsView	False
IsDomain	False	IsSPPackage	False
Synonymns		CorrelationName	





SelectClause		IsUpdateable	True
CheckOption	None	IsSnapShot	False
IsDistinct	False	PersistToServer	
IsPackage	False	IsPackageWrapped	False
IsPackageBodyWrapped	False		

<<Table>> T_1 {Data Modeler}



Parent Package	S_0	Abstract	No
Export Control	PublicAccess	Link Class for	None
Class Kind	NormalClass	Cardinality	n
Space		Concurrency	Sequential
Persistence	No		

Assigned Components	DB_MANTENIMIENTO
----------------------------	-------------------------



Operations

Name	Signature	Class
 PK_T_19	 <<PK>> PK_T_19 ()	T_1
 FK_T_19	 <<FK>> FK_T_19 ()	T_1
 FK_T_110	 <<FK>> FK_T_110 ()	T_1
 TC_T_112	 <<Index>> TC_T_112 ()	T_1



 TC_T_113	 <<Index>> TC_T_113 ()	T_1
--	---	-----

Attributes

Name	Class	Type	Initial Value
 T_CS_EQUIPO_MEDICION_ID	T_1	INT	
 T_CS_ORDEN_TRABAJO_ID	T_1	INT	

Associations

Name	My Role	My Class	Other Role	Other Element
--Not Named--	--Not Named--	T_1	--Not Named--	T_CS_EQUIPO_MEDICION
--Not Named--	--Not Named--	T_1	--Not Named--	T_CS_ORDEN_TRABAJO

Property Settings

Data Modeler

dmlItem	True	IsTable	True
DMName	T_1	CodeName	
TableSpaceID		SourceId	666
SourceType	n:n Association	HasLikeStatement	False
LikeTableName		LikeIncludeIdentity	False
LikeColumnAttr	False	IsView	False
IsDomain	False	IsSPPackage	False









Synonymns		CorrelationName	
SelectClause		IsUpdateable	True
CheckOption	None	IsSnapShot	False
IsDistinct	False	PersistToServer	
IsPackage	False	IsPackageWrapped	False
IsPackageBodyWrapped	False		

<<Table>> T_CS_AREA {Data Modeler}

Parent Package	S_0	Abstract	No
Export Control	PublicAccess	Link Class for	None
Class Kind	NormalClass	Cardinality	n
Space		Concurrency	Sequential
Persistence	No		

Assigned Components	DB_MANTENIMIENTO
----------------------------	-------------------------

Operations

Name	Signature	Class
 PK_T_CS_AREA2	 <<PK>> PK_T_CS_AREA2 ()	T_CS_AREA
 FK_T_CS_AREA2	 <<FK>> FK_T_CS_AREA2 ()	T_CS_AREA
 TC_T_CS_AREA22	 <<Index>> TC_T_CS_AREA22 ()	T_CS_AREA



Attributes

Name	Class	Type	Initial Value
NN ID_AREA	T_CS_AREA	SMALLINT	
NN NOMB_AREA	T_CS_AREA	SMALLINT	
PK T_CS_AREA_ID	T_CS_AREA	INT	
FK T_CS_PERSONAL_ID	T_CS_AREA	INT	

Associations

Name	My Role	My Class	Other Role	Other Element
--Not Named--	--Not Named--	T_CS_AREA	--Not Named--	T_CS_PERSONAL
--Not Named--	--Not Named--	T_CS_AREA	--Not Named--	T_CS_PERSONAL
--Not Named--	--Not Named--	T_CS_AREA	--Not Named--	T_CS_PERSONAL

Property Settings

Data Modeler

dmlItem	True	IsTable	True
DMName	T_CS_AREA	CodeName	
TableSpaceID		Sourceld	4F69588E02F3
SourceType	Class	HasLikeStatemenet	False
LikeTableName		LikeIncludelidentity	False





LikeColumnAttr	False	IsView	False
IsDomain	False	IsSPPackage	False
Synonymns		CorrelationName	
SelectClause		IsUpdateable	True
CheckOption	None	IsSnapShot	False
IsDistinct	False	PersistToServer	
IsPackage	False	IsPackageWrapped	False
IsPackageBodyWrapped	False		

<<Table>> T_CS_EQUIPO {Data Modeler}

Parent Package	S_0	Abstract	No
Export Control	PublicAccess	Link Class for	None
Class Kind	NormalClass	Cardinality	n
Space		Concurrency	Sequential
Persistence	No		

Assigned Components	DB_MANTENIMIENTO
----------------------------	------------------

Operations

Name	Signature	Class
 TC_T_CS_EQUIPO18	 <<Unique>> TC_T_CS_EQUIPO18 ()	T_CS_EQUIPO



PK_T_CS_EQUIPO0	<<PK>> PK_T_CS_EQUIPO0 ()	T_CS_EQUIPO
FK_T_CS_EQUIPO13	<<FK>> FK_T_CS_EQUIPO13 ()	T_CS_EQUIPO
TC_T_CS_EQUIPO19	<<Index>> TC_T_CS_EQUIPO19 ()	T_CS_EQUIPO

Attributes

Name	Class	Type	Initial Value
ID_EQUIPO	T_CS_EQUIPO	SMALLINT	
NOMB_EQUIPO	T_CS_EQUIPO	SMALLINT	
MARC_EQUIPO	T_CS_EQUIPO	SMALLINT	
MODE_EQUIPO	T_CS_EQUIPO	SMALLINT	
SERI_EQUIPO	T_CS_EQUIPO	SMALLINT	
DESC_EQUIPO	T_CS_EQUIPO	SMALLINT	
T_CS_EQUIPO_ID	T_CS_EQUIPO	INT	
T_CS_EQUIPO_PENDIENTE_ID	T_CS_EQUIPO	INT	

Associations

Name	My Role	My Class	Other Role	Other Element
--Not Named--	--Not Named-- -	T_CS_EQUIPO	--Not Named-- -	T_CS_EQUIPO_PENDIENTE
--Not Named--	--Not Named-- -	T_CS_EQUIPO	--Not Named-- -	T_CS_ORDEN_TRABAJO



--Not Named--	--Not Named--	T_CS_EQUIPO	--Not Named--	T_CS_ORDEN_TRABAJO
---------------	---------------	-------------	---------------	--------------------

Property Settings

Data Modeler

dmlItem	True	IsTable	True
DMName	T_CS_EQUIPO	CodeName	
TableSpaceID		Sourceld	4F69552900F9
SourceType	Class	HasLikeStatemenet	False
LikeTableName		LikeIncludelidentity	False
LikeColumnAttr	False	IsView	False
IsDomain	False	IsSPPackage	False
Synonymns		CorrelationName	
SelectClause		IsUpdateable	True
CheckOption	None	IsSnapShot	False
IsDistinct	False	PersistToServer	
IsPackage	False	IsPackageWrapped	False
IsPackageBodyWrapped	False		

<<Table>> T_CS_EQUIPO_MEDICION {Data Modeler}



Parent Package	S_0	Abstract	No
Export Control	PublicAccess	Link Class for	None










Class Kind	NormalClass	Cardinality	n
Space		Concurrency	Sequential
Persistence	No		

Assigned Components	DB_MANTENIMIENTO
----------------------------	------------------

Operations

Name	Signature	Class
 PK_T_CS_EQUIPO_MEDICION4	 <<PK>> PK_T_CS_EQUIPO_MEDICION4 ()	T_CS_EQUIPO_MEDICION

Attributes

Name	Class	Type	Initial Value
 ID_EQUIPO_MEDICION	T_CS_EQUIPO_MEDICION	SMALLINT	
 NOMB_EQUIPO_MEDICION	T_CS_EQUIPO_MEDICION	SMALLINT	
 MARCA_EQUIPO_MEDICION	T_CS_EQUIPO_MEDICION	SMALLINT	
 MODE_EQUIPO_MEDIDION	T_CS_EQUIPO_MEDICION	SMALLINT	
 SERI_EQUIPO_MEDICION	T_CS_EQUIPO_MEDICION	SMALLINT	
 CERT_CALIB_EQUIPO_MEDICION	T_CS_EQUIPO_MEDICION	SMALLINT	
 T_CS_EQUIPO_MEDICION_ID	T_CS_EQUIPO_MEDICION	INT	



Associations

Name	My Role	My Class	Other Role	Other Element
--Not Named--	-- Not Named--	T_CS_EQUIPO_MEDICION	-- Not Named--	T_1
--Not Named--	-- Not Named--	T_CS_EQUIPO_MEDICION	-- Not Named--	T_CS_ORDEN_TRABAJO

Property Settings

Data Modeler

dmlItem	True	IsTable	True
DMName	T_CS_EQUIPO_MEDICION	CodeName	
TableSpaceID		SourceId	4F695F6F0075
SourceType	Class	HasLikeStatement	False
LikeTableName		LikeIncludeIdentity	False
LikeColumnAttr	False	IsView	False
IsDomain	False	IsSPPackage	False
Synonymns		CorrelationName	
SelectClause		IsUpdateable	True
CheckOption	None	IsSnapShot	False
IsDistinct	False	PersistToServer	
IsPackage	False	IsPackageWrapped	False





IsPackageBodyWrapped	False		
-----------------------------	-------	--	--

<<Table>> T_CS_EQUIPO_PENDIENTE {Data Modeler}




Parent Package	S_0	Abstract	No
Export Control	PublicAccess	Link Class for	None
Class Kind	NormalClass	Cardinality	n
Space		Concurrency	Sequential
Persistence	No		

Assigned Components	DB_MANTENIMIENTO
----------------------------	------------------

Operations

Name	Signature	Class
 PK_T_CS_EQUIPO_PENDIENTE7	 <<PK>> PK_T_CS_EQUIPO_PENDIENTE7 ()	T_CS_EQUIPO_PENDIENTE

Attributes

Name	Class	Type	Initial Value
 ID_EQUIPO_PENDIENTE	T_CS_EQUIPO_PENDIENTE	SMALLINT	
 FECH_EQUIPO_PENDIENTE	T_CS_EQUIPO_PENDIENTE	SMALLINT	
 DETA_EQUIPO_PENDIENTE	T_CS_EQUIPO_PENDIENTE	SMALLINT	



NN SOLI_EQUIPO_PENDIENTE	T_CS_EQUIPO_PENDIENTE	SMALLINT	
NN ESTA_EQUIPO_PENDIENTE	T_CS_EQUIPO_PENDIENTE	SMALLINT	
PK T_CS_EQUIPO_PENDIENTE_ID	T_CS_EQUIPO_PENDIENTE	INT	

Associations

Name	My Role	My Class	Other Role	Other Element
--Not Named--	--Not Named--	T_CS_EQUIPO_PENDIENTE	--Not Named--	T_CS_EQUIPO

Property Settings

Data Modeler

dmlItem	True	IsTable	True
DMName	T_CS_EQUIPO_PENDIENTE	CodeName	
TableSpaceID		SourceId	4F699D6202AA
SourceType	Class	HasLikeStatement	False
LikeTableName		LikeIncludeIdentity	False
LikeColumnAttr	False	IsView	False
IsDomain	False	IsSPPackage	False
Synonymns		CorrelationName	
SelectClause		IsUpdateable	True
CheckOption	None	IsSnapShot	False





IsDistinct	False	PersistToServer	
IsPackage	False	IsPackageWrapped	False
IsPackageBodyWrapped	False		

<<Table>> T_CS_MATERIAL {Data Modeler}




Parent Package	S_0	Abstract	No
Export Control	PublicAccess	Link Class for	None
Class Kind	NormalClass	Cardinality	n
Space		Concurrency	Sequential
Persistence	No		

Assigned Components	DB_MANTENIMIENTO
----------------------------	------------------

Operations

Name	Signature	Class
 PK_T_CS_MATERIAL5	 <<PK>> PK_T_CS_MATERIAL5 ()	T_CS_MATERIAL

Attributes

Name	Class	Type	Initial Value
 ID_MATERIAL	T_CS_MATERIAL	SMALLINT	
 DESC_MATERIAL	T_CS_MATERIAL	SMALLINT	
 CANT_MATERIAL	T_CS_MATERIAL	SMALLINT	



PK T_CS_MATERIAL_ID	T_CS_MATERIAL	INT	
----------------------------	---------------	-----	--

Associations

Name	My Role	My Class	Other Role	Other Element
--Not Named- -	--Not Named- -	T_CS_MATERIAL	--Not Named- -	T_0
--Not Named- -	--Not Named- -	T_CS_MATERIAL	--Not Named- -	T_CS_ORDEN_TRABAJO

Property Settings

Data Modeler

dmlItem	True	IsTable	True
DMName	T_CS_MATERIAL	CodeName	
TableSpaceID		SourceId	4F69602A02B4
SourceType	Class	HasLikeStatement	False
LikeTableName		LikeIncludeIdentity	False
LikeColumnAttr	False	IsView	False
IsDomain	False	IsSPPackage	False
Synonymns		CorrelationName	
SelectClause		IsUpdateable	True
CheckOption	None	IsSnapShot	False
IsDistinct	False	PersistToServer	









IsPackage	False	IsPackageWrapped	False
IsPackageBodyWrapped	False		

<<Table>> T_CS_ORDEN_TRABAJO {Data Modeler}





















Parent Package	S_0	Abstract	No
Export Control	PublicAccess	Link Class for	None
Class Kind	NormalClass	Cardinality	n
Space		Concurrency	Sequential
Persistence	No		

Assigned Components	DB_MANTENIMIENTO
----------------------------	-------------------------

Operations

Name	Signature	Class
 TC_T_CS_ORDEN_TRABAJO2	 <<Unique>> TC_T_CS_ORDEN_TRABAJO2 ()	T_CS_ORDEN_TRABAJO
 PK_T_CS_ORDEN_TRABAJO3	 <<PK>> PK_T_CS_ORDEN_TRABAJO3 ()	T_CS_ORDEN_TRABAJO
 TC_T_CS_ORDEN_TRABAJO5	 <<Unique>> TC_T_CS_ORDEN_TRABAJO5 ()	T_CS_ORDEN_TRABAJO



 TC_T_CS_ORDEN_TRABAJ O16	 <<Unique>> TC_T_CS_ORDEN_TRABAJ O16 ()	T_CS_ORDEN_TRABA JO
 FK_T_CS_ORDEN_TRABAJ O0	 <<FK>> FK_T_CS_ORDEN_TRABAJO0 ()	T_CS_ORDEN_TRABA JO
 FK_T_CS_ORDEN_TRABAJ O8	 <<FK>> FK_T_CS_ORDEN_TRABAJO8 ()	T_CS_ORDEN_TRABA JO
 FK_T_CS_ORDEN_TRABAJ O3	 <<FK>> FK_T_CS_ORDEN_TRABAJO3 ()	T_CS_ORDEN_TRABA JO
 FK_T_CS_ORDEN_TRABAJ O1	 <<FK>> FK_T_CS_ORDEN_TRABAJO1 ()	T_CS_ORDEN_TRABA JO
 FK_T_CS_ORDEN_TRABAJ O11	 <<FK>> FK_T_CS_ORDEN_TRABAJO11 ()	T_CS_ORDEN_TRABA JO
 FK_T_CS_ORDEN_TRABAJ O12	 <<FK>> FK_T_CS_ORDEN_TRABAJO12 ()	T_CS_ORDEN_TRABA JO
 TC_T_CS_ORDEN_TRABAJ O24	 <<Index>> TC_T_CS_ORDEN_TRABAJO 24 ()	T_CS_ORDEN_TRABA JO
 TC_T_CS_ORDEN_TRABAJ O28	 <<Index>> TC_T_CS_ORDEN_TRABAJO 28 ()	T_CS_ORDEN_TRABA JO
 TC_T_CS_ORDEN_TRABAJ O23	 <<Index>> TC_T_CS_ORDEN_TRABAJO 23 ()	T_CS_ORDEN_TRABA JO



TC_T_CS_ORDEN_TRABAJ O27	<<Index>> TC_T_CS_ORDEN_TRABAJO 27 ()	T_CS_ORDEN_TRABA JO
TC_T_CS_ORDEN_TRABAJ O26	<<Index>> TC_T_CS_ORDEN_TRABAJO 26 ()	T_CS_ORDEN_TRABA JO
TC_T_CS_ORDEN_TRABAJ O25	<<Index>> TC_T_CS_ORDEN_TRABAJO 25 ()	T_CS_ORDEN_TRABA JO

Attributes

Name	Class	Type	Initial Value
NN FECH_ORDEN_TRABAJO	T_CS_ORDEN_TRABAJO	SMALLINT	
NN FALL_REPORT_ORDEN_TRABAJO	T_CS_ORDEN_TRABAJO	SMALLINT	
NN VERI_ORDEN_TRABAJO	T_CS_ORDEN_TRABAJO	BIT	
NN TRAB_REAL_ORDEN_TRABAJO	T_CS_ORDEN_TRABAJO	SMALLINT	
NN RECO_OBSE_ORDEN_TRABAJO	T_CS_ORDEN_TRABAJO	SMALLINT	
NN TIEM_UTIL_ORDEN_TRABAJO	T_CS_ORDEN_TRABAJO	SMALLINT	
NN CANT_MATE_ORDEN_TRABAJO	T_CS_ORDEN_TRABAJO	SMALLINT	
PK T_CS_ORDEN_TRABAJO_ID	T_CS_ORDEN_TRABAJO	INT	
FK T_CS_EQUIPO_ID	T_CS_ORDEN_TRABAJO	INT	
FK NN T_CS_EQUIPO_T_CS_EQUIPO_ID	T_CS_ORDEN_TRABAJO	INT	
FK NN T_CS_PERSONAL_ID	T_CS_ORDEN_TRABAJO	INT	
FK T_CS_EQUIPO_MEDICION_ID	T_CS_ORDEN_TRABAJO	INT	



FK T_CS_MATERIAL_ID	T_CS_ORDEN_TRABAJO	INT	
FK T_CS_TIPO_MANTENIMIENTO_ID	T_CS_ORDEN_TRABAJO	INT	

Associations

Name	My Role	My Class	Other Role	Other Element
--Not Named--	-- Not Named-- -	T_CS_ORDEN_TRABAJO	-- Not Named-- -	T_1
--Not Named--	-- Not Named-- -	T_CS_ORDEN_TRABAJO	-- Not Named-- -	T_0
--Not Named--	-- Not Named-- -	T_CS_ORDEN_TRABAJO	-- Not Named-- -	T_CS_EQUIPO
--Not Named--	-- Not Named-- -	T_CS_ORDEN_TRABAJO	-- Not Named-- -	T_CS_EQUIPO_MEDICION
--Not Named--	-- Not Named-- -	T_CS_ORDEN_TRABAJO	-- Not Named-- -	T_CS_PERSONAL
--Not Named--	-- Not Named-- -	T_CS_ORDEN_TRABAJO	-- Not Named-- -	T_CS_EQUIPO
--Not Named--	-- Not Named-- -	T_CS_ORDEN_TRABAJO	-- Not Named-- -	T_CS_MATERIAL
--Not Named--	-- Not Named-- -	T_CS_ORDEN_TRABAJO	-- Not Named-- -	T_CS_TIPO_MANTENIMIENTO



Property Settings

Data Modeler

dmlItem	True	IsTable	True
DMName	T_CS_ORDEN_TRABAJO	CodeName	
TableSpaceID		SourceId	4F6958D0021C
SourceType	Class	HasLikeStatement	False
LikeTableName		LikeIncludeIdentity	False
LikeColumnAttr	False	IsView	False
IsDomain	False	IsSPPackage	False
Synonymns		CorrelationName	
SelectClause		IsUpdateable	True
CheckOption	None	IsSnapShot	False
IsDistinct	False	PersistToServer	
IsPackage	False	IsPackageWrapped	False
IsPackageBodyWrapped	False		

<<Table>> T_CS_PERSONAL {Data Modeler}













Parent Package	S_0	Abstract	No
-----------------------	-----	-----------------	----





Export Control	PublicAccess	Link Class for	None
Class Kind	NormalClass	Cardinality	n
Space		Concurrency	Sequential
Persistence	No		

Assigned Components	DB_MANTENIMIENTO
----------------------------	-------------------------

Operations

Name	Signature	Class
 TC_T_CS_PERSONAL8	 <<Unique>> TC_T_CS_PERSONAL8 ()	T_CS_PERSONAL
 PK_T_CS_PERSONAL1	 <<PK>> PK_T_CS_PERSONAL1 ()	T_CS_PERSONAL
 FK_T_CS_PERSONAL5	 <<FK>> FK_T_CS_PERSONAL5 ()	T_CS_PERSONAL
 FK_T_CS_PERSONAL4	 <<FK>> FK_T_CS_PERSONAL4 ()	T_CS_PERSONAL
 TC_T_CS_PERSONAL20	 <<Index>> TC_T_CS_PERSONAL20 ()	T_CS_PERSONAL
 TC_T_CS_PERSONAL21	 <<Index>> TC_T_CS_PERSONAL21 ()	T_CS_PERSONAL

Attributes

Name	Class	Type	Initial Value
 CI_PERSONAL	T_CS_PERSONAL	SMALLINT	
 NOMB_PERSONAL	T_CS_PERSONAL	SMALLINT	



NN APEL_PERSONAL	T_CS_PERSONAL	SMALLINT	
NN DIRE_PERSONAL	T_CS_PERSONAL	SMALLINT	
NN TELF_PERSONAL	T_CS_PERSONAL	SMALLINT	
NN MAIL_PERSONAL	T_CS_PERSONAL	SMALLINT	
NN CARG_PERSONAL	T_CS_PERSONAL	SMALLINT	
PK T_CS_PERSONAL_ID	T_CS_PERSONAL	INT	
FK T_CS_AREA_ID	T_CS_PERSONAL	INT	
FK NN T_CS_AREA_T_CS_AREA_ID	T_CS_PERSONAL	INT	

Associations

Name	My Role	My Class	Other Role	Other Element
--Not Named- -	--Not Named- -	T_CS_PERSONAL	--Not Named- -	T_CS_AREA
--Not Named- -	--Not Named- -	T_CS_PERSONAL	--Not Named- -	T_CS_AREA
--Not Named- -	--Not Named- -	T_CS_PERSONAL	--Not Named- -	T_CS_AREA
--Not Named- -	--Not Named- -	T_CS_PERSONAL	--Not Named- -	T_CS_ORDEN_TRABAJO

Property Settings

Data Modeler



dmlItem	True	IsTable	True
DMName	T_CS_PERSONAL	CodeName	
TableSpaceID		SourceId	4F6956D201C2
SourceType	Class	HasLikeStatement	False
LikeTableName		LikeIncludeIdentity	False
LikeColumnAttr	False	IsView	False
IsDomain	False	IsSPPackage	False
Synonyms		CorrelationName	
SelectClause		IsUpdateable	True
CheckOption	None	IsSnapshot	False
IsDistinct	False	PersistToServer	
IsPackage	False	IsPackageWrapped	False
IsPackageBodyWrapped	False		



<<Table>> T_CS_TIPO_MANTENIMIENTO {Data Modeler}

Parent Package	S_0	Abstract	No
Export Control	PublicAccess	Link Class for	None
Class Kind	NormalClass	Cardinality	n
Space		Concurrency	Sequential
Persistence	No		







Assigned Components	DB_MANTENIMIENTO
---------------------	------------------

Operations

Name	Signature	Class
 PK_T_CS_TIPO_MANTENIMIENTO6	 <<PK>> PK_T_CS_TIPO_MANTENIMIENTO6 ()	T_CS_TIPO_MANTENIMIENTO

Attributes

Name	Class	Type	Initial Value
 MANT_CORR_TIPO_MANTENIMIENTO	T_CS_TIPO_MANTENIMIENTO	SMALLINT	
 MANT_PREV_TIPO_MANTENIMIENTO	T_CS_TIPO_MANTENIMIENTO	SMALLINT	
 MANT_MEJO_TIPO_MANTENIMIENTO	T_CS_TIPO_MANTENIMIENTO	SMALLINT	
 T_CS_TIPO_MANTENIMIENTO_ID	T_CS_TIPO_MANTENIMIENTO	INT	

Associations

Name	My Role	My Class	Other Role	Other Element
--Not Named--	-- Not Named--	T_CS_TIPO_MANTENIMIENTO	-- Not Named--	T_CS_ORDEN_TRABAJO



Property Settings

Data Modeler

dmlItem	True	IsTable	True
DMName	T_CS_TIPO_MANTENIMIENTO	CodeName	
TableSpaceID		SourceId	4F6960920365
SourceType	Class	HasLikeStatement	False
LikeTableName		LikeIncludeIdentity	False
LikeColumnAttr	False	IsView	False
IsDomain	False	IsSPPackage	False
Synonymns		CorrelationName	
SelectClause		IsUpdateable	True
CheckOption	None	IsSnapShot	False
IsDistinct	False	PersistToServer	
IsPackage	False	IsPackageWrapped	False
IsPackageBodyWrapped	False		

<<Identifying>> (theT_1:T_1) (theT_CS_EQUIPO_MEDICION:T_CS_EQUIPO_MEDICION)

Derived	No	Link Class	--Not Defined--
Element 1	T_1	Element 2	T_CS_EQUIPO_MEDICION



Role 1	--Not Named--	Role 2	--Not Named--
--------	---------------	--------	---------------

Property Settings

Data Modeler

dmlItem	True	IsRelationship	True
DMName		CodeName	
RIMethod	DRI	ParentUpdateRule	No Action
ParentDeleteRule	No Action	ChildMultiplicity	False
ParentUpdateRuleName		ParentDeleteRuleName	
ChildInsertRestrict	False	ChildInsertRestrictName	
ChildMultiplicityName		SourceId	4F69978E00B3
SourceType	Source Role		

<<Identifying>> (theT_1:T_1) (theT_CS_ORDEN_TRABAJO:T_CS_ORDEN_TRABAJO)

Derived	No	Link Class	--Not Defined--
Element 1	<u>T_1</u>	Element 2	<u>T_CS_ORDEN_TRABAJO</u>
Role 1	--Not Named--	Role 2	--Not Named--

Property Settings

Data Modeler



dmlItem	True	IsRelationship	True
DMName		CodeName	
RIMethod	DRI	ParentUpdateRule	No Action
ParentDeleteRule	No Action	ChildMultiplicity	False
ParentUpdateRuleName		ParentDeleteRuleName	
ChildInsertRestrict	False	ChildInsertRestrictName	
ChildMultiplicityName		Sourceld	4F69978E00B3
SourceType	Source Role		

<<Identifying>> (theT_0:T_0) (theT_CS_ORDEN_TRABAJO:T_CS_ORDEN_TRABAJO)

Derived	No	Link Class	--Not Defined--
Element 1	<u>T_0</u>	Element 2	<u>T_CS_ORDEN_TRABAJO</u>
Role 1	<u>--Not Named--</u>	Role 2	<u>--Not Named--</u>

Property Settings

Data Modeler

dmlItem	True	IsRelationship	True
DMName		CodeName	
RIMethod	DRI	ParentUpdateRule	No Action
ParentDeleteRule	No Action	ChildMultiplicity	False



ParentUpdateRuleName		ParentDeleteRuleName	
ChildInsertRestrict	False	ChildInsertRestrictName	
ChildMultiplicityName		SourceId	4F6998EC0064
SourceType	Source Role		

<<Identifying>> (theT_0:T_0) (theT_CS_MATERIAL:T_CS_MATERIAL)

Derived	No	Link Class	--Not Defined--
Element 1	<u>T_0</u>	Element 2	<u>T_CS_MATERIAL</u>
Role 1	--Not Named--	Role 2	--Not Named--

Property Settings

Data Modeler

dmlItem	True	IsRelationship	True
DMName		CodeName	
RIMethod	DRI	ParentUpdateRule	No Action
ParentDeleteRule	No Action	ChildMultiplicity	False
ParentUpdateRuleName		ParentDeleteRuleName	
ChildInsertRestrict	False	ChildInsertRestrictName	
ChildMultiplicityName		SourceId	4F6998EC0064
SourceType	Source Role		



<<Non-Identifying>> (theT_CS_PERSONAL:T_CS_PERSONAL) (theT_CS_AREA:T_CS_AREA)

Derived	No	Link Class	--Not Defined--
Element 1	<u>T_CS_PERSONAL</u>	Element 2	<u>T_CS_AREA</u>
Role 1	--Not Named--	Role 2	--Not Named--

Property Settings

Data Modeler

dmlItem	True	IsRelationship	True
DMName		CodeName	
RIMethod	DRI	ParentUpdateRule	No Action
ParentDeleteRule	No Action	ChildMultiplicity	False
ParentUpdateRuleName		ParentDeleteRuleName	
ChildInsertRestrict	False	ChildInsertRestrictName	
ChildMultiplicityName		SourceId	4F69951C024B
SourceType	Association		

<<Non-Identifying>> (theT_CS_PERSONAL:T_CS_PERSONAL) (theT_CS_AREA:T_CS_AREA)

Derived	No	Link Class	--Not Defined--
----------------	----	-------------------	-----------------



Element 1	<u>T_CS_PERSONAL</u>	Element 2	<u>T_CS_AREA</u>
Role 1	<u>--Not Named--</u>	Role 2	<u>--Not Named--</u>

Property Settings

Data Modeler

dmlItem	True	IsRelationship	True
DMName		CodeName	
RIMethod	DRI	ParentUpdateRule	No Action
ParentDeleteRule	No Action	ChildMultiplicity	False
ParentUpdateRuleName		ParentDeleteRuleName	
ChildInsertRestrict	False	ChildInsertRestrictName	
ChildMultiplicityName		SourceId	4F69950B0034
SourceType	Association		

<<Non-Identifying>> (theT_CS_AREA:T_CS_AREA) (theT_CS_PERSONAL:T_CS_PERSONAL)

Derived	No	Link Class	--Not Defined--
Element 1	<u>T_CS_AREA</u>	Element 2	<u>T_CS_PERSONAL</u>
Role 1	<u>--Not Named--</u>	Role 2	<u>--Not Named--</u>



Property Settings

Data Modeler

dmlItem	True	IsRelationship	True
DMName		CodeName	
RIMethod	DRI	ParentUpdateRule	No Action
ParentDeleteRule	No Action	ChildMultiplicity	False
ParentUpdateRuleName		ParentDeleteRuleName	
ChildInsertRestrict	False	ChildInsertRestrictName	
ChildMultiplicityName		Sourceld	4F6993C30369
SourceType	Association		

<<Non-Identifying>> (theT_CS_EQUIPO:T_CS_EQUIPO) (theT_CS_EQUIPO_PENDIENTE:T_CS_EQUIPO_PENDIENTE)

Derived	No	Link Class	--Not Defined--
Element 1	<u>T_CS_EQUIPO</u>	Element 2	<u>T_CS_EQUIPO_PENDIENTE</u>
Role 1	<u>--Not Named--</u>	Role 2	<u>--Not Named--</u>

Property Settings

Data Modeler



dmlItem	True	IsRelationship	True
DMName		CodeName	
RIMethod	DRI	ParentUpdateRule	No Action
ParentDeleteRule	No Action	ChildMultiplicity	False
ParentUpdateRuleName		ParentDeleteRuleName	
ChildInsertRestrict	False	ChildInsertRestrictName	
ChildMultiplicityName		SourceId	4F699F1000A6
SourceType	Association		

<<Non-Identifying>> (theT_CS_ORDEN_TRABAJO:T_CS_ORDEN_TRABAJO) (theT_CS_EQUIPO:T_CS_EQUIPO)

Derived	No	Link Class	--Not Defined--
Element 1	<u>T_CS_ORDEN_TRABAJO</u>	Element 2	<u>T_CS_EQUIPO</u>
Role 1	--Not Named--	Role 2	--Not Named--

Property Settings

Data Modeler

dmlItem	True	IsRelationship	True
DMName		CodeName	
RIMethod	DRI	ParentUpdateRule	No Action



ParentDeleteRule	No Action	ChildMultiplicity	False
ParentUpdateRuleName		ParentDeleteRuleName	
ChildInsertRestrict	False	ChildInsertRestrictName	
ChildMultiplicityName		Sourceld	4F6961C50226
SourceType	Association		

<<Non-Identifying>> (theT_CS_ORDEN_TRABAJO:T_CS_ORDEN_TRABAJO) (theT_CS_EQUIPO_MEDICION:T_CS_EQUIPO_MEDICION)

Derived	No	Link Class	--Not Defined--
Element 1	<u>T_CS_ORDEN_TRABAJO</u>	Element 2	<u>T_CS_EQUIPO_MEDICION</u>
Role 1	<u>--Not Named--</u>	Role 2	<u>--Not Named--</u>

Property Settings

Data Modeler

dmlItem	True	IsRelationship	True
DMName		CodeName	
RIMethod	DRI	ParentUpdateRule	No Action
ParentDeleteRule	No Action	ChildMultiplicity	False
ParentUpdateRuleName		ParentDeleteRuleName	
ChildInsertRestrict	False	ChildInsertRestrictName	
ChildMultiplicityName		Sourceld	4F6964CD0302



SourceType	Association		
------------	-------------	--	--

<<Non-Identifying>> (theT_CS_ORDEN_TRABAJO:T_CS_ORDEN_TRABAJO) (theT_CS_PERSONAL:T_CS_PERSONAL)

Derived	No	Link Class	--Not Defined--
Element 1	<u>T_CS_ORDEN_TRABAJO</u>	Element 2	<u>T_CS_PERSONAL</u>
Role 1	--Not Named--	Role 2	--Not Named--

Property Settings

Data Modeler

dmlItem	True	IsRelationship	True
DMName		CodeName	
RIMethod	DRI	ParentUpdateRule	No Action
ParentDeleteRule	No Action	ChildMultiplicity	False
ParentUpdateRuleName		ParentDeleteRuleName	
ChildInsertRestrict	False	ChildInsertRestrictName	
ChildMultiplicityName		SourceId	4F699C290308
SourceType	Association		



<<Non-Identifying>> (theT_CS_ORDEN_TRABAJO:T_CS_ORDEN_TRABAJO) (theT_CS_EQUIPO:T_CS_EQUIPO)

Derived	No	Link Class	--Not Defined--
Element 1	<u>T_CS_ORDEN_TRABAJO</u>	Element 2	<u>T_CS_EQUIPO</u>
Role 1	--Not Named--	Role 2	--Not Named--

Property Settings

Data Modeler

dmlItem	True	IsRelationship	True
DMName		CodeName	
RIMethod	DRI	ParentUpdateRule	No Action
ParentDeleteRule	No Action	ChildMultiplicity	False
ParentUpdateRuleName		ParentDeleteRuleName	
ChildInsertRestrict	False	ChildInsertRestrictName	
ChildMultiplicityName		SourceId	4F699BFD010D
SourceType	Association		

<<Non-Identifying>> (theT_CS_ORDEN_TRABAJO:T_CS_ORDEN_TRABAJO) (theT_CS_MATERIAL:T_CS_MATERIAL)

Derived	No	Link Class	--Not Defined--
----------------	----	-------------------	-----------------



Element 1	<u>T_CS_ORDEN_TRABAJO</u>	Element 2	<u>T_CS_MATERIAL</u>
Role 1	<u>--Not Named--</u>	Role 2	<u>--Not Named--</u>

Property Settings

Data Modeler

dmlItem	True	IsRelationship	True
DMName		CodeName	
RIMethod	DRI	ParentUpdateRule	No Action
ParentDeleteRule	No Action	ChildMultiplicity	False
ParentUpdateRuleName		ParentDeleteRuleName	
ChildInsertRestrict	False	ChildInsertRestrictName	
ChildMultiplicityName		SourceId	4F6998E6008D
SourceType	Association		

<<Non-Identifying>> (theT_CS_ORDEN_TRABAJO:T_CS_ORDEN_TRABAJO) (theT_CS_TIPO_MANTENIMIENTO:T_CS_TIPO_MANTENIMIENTO)

Derived	No	Link Class	--Not Defined--
Element 1	<u>T_CS_ORDEN_TRABAJO</u>	Element 2	<u>T_CS_TIPO_MANTENIMIENTO</u>
Role 1	<u>--Not Named--</u>	Role 2	<u>--Not Named--</u>



Property Settings

Data Modeler

dmlItem	True	IsRelationship	True
DMName		CodeName	
RIMethod	DRI	ParentUpdateRule	No Action
ParentDeleteRule	No Action	ChildMultiplicity	False
ParentUpdateRuleName		ParentDeleteRuleName	
ChildInsertRestrict	False	ChildInsertRestrictName	
ChildMultiplicityName		SourceId	4F699BC402A6
SourceType	Association		

4.12 Estándares

Consideraciones Generales

Nomenclatura

El nombre de los objetos de base de datos no deberá exceder los 30 caracteres y no deberá terminar en “_”.

En el caso que el nombre esté compuesto por más de una palabra, las mismas tendrán que separarse por el signo “_” underscore.

Los nombres de objetos de base de datos deberán estar escritos en español, ser auto-descriptivos (nombres completos) y estarán sujetos a verificación por parte del área de Diseño y Aseguramiento de Calidad, quienes solicitarán el cambio en el caso de no ser auto-descriptivos y/o presentar ambigüedades.



El Área de Diseño asignará el nombre del proyecto, nombre de esquema y el prefijo de la aplicación a utilizarse en la nomenclatura de los objetos de base de datos.

Tipos de Datos

Tipo de Dato	Cuando se debe utilizar
VARCHAR2	Para campos de texto de tamaño variable de hasta 4000 caracteres. Siempre se utilizará VACHAR2 en lugar de VARCHAR.
CHAR	Para campos de texto de tamaño fijo, por ejemplo para el uso de estados (SI/NO), (ACT/INA).
NUMBER	Para campos numéricos. Siempre se especificará la precisión, por ejemplo para un número de 5 cifras enteras y 2 decimales será: NUMBER (7,2).
DATE	Para campos de fecha y de fecha y hora.
CLOB	Para campos de texto de más de 4000 caracteres.
BLOB	Para almacenamiento de archivos binarios, por ejemplo imágenes, archivos pdf, Word, etc.
XMLTYPE	Para campos en formato XML sobre los cuales se ejecutarán consultas con XPATH desde la Base de Datos. Si no se requerirá utilizar el motor de XML de la Base de Datos, es preferible utilizar un campo CLOB

Interacción entre instancias de Base de Datos y Replicas

En el caso de requerir interacción entre las instancias de base de datos antes mencionadas, se deberán crear replicas que permitan manejar independencia entre dichas instancias de base de datos. La periodicidad y el tipo de refrescamiento se analizarán con el área de Diseño.



Si se considera que ciertos procesos no requieren mantener dicha independencia, se revisará con el área de Diseño para determinar la mejor implementación.

De manera general no se debe crear sinónimos públicos o vistas que apunten directamente a la fuente de información.

Tablas

Nombre:	[Prefijo de la aplicación] "_" [Nombre auto-descriptivo en "singular"] De manera general el nombre auto-descriptivo de una tabla de rompimiento se conformará con la unión del nombre de las dos tablas padres.
Alias:	Se utilizará el alias generado por Designer. Si ya se encuentra en uso el prefijo asignado, se añadirá al final un número secuencial de 1 a 9.
Consideraciones:	Toda tabla tendrá una "clave primaria numérica de un solo campo" salvo mejor criterio de Diseño.

Clave Primaria (Primary Key)

Nombre Campo:	"CÓDIGO_" [Nombre de la tabla en singular , sin prefijo de la aplicación, hasta donde alcance]
Nombre PK:	Utilizar el nombre asignado por Designer ([alias de la tabla] "_PK")



Consideraciones:	<p>La clave primaria debe ser numérica y de un solo campo, salvo mejor criterio de Diseño.</p> <p>Como caso de excepción se permitirá la creación de claves primarias compuestas, en el caso en que la mejor representación a nivel de entidades en un diagrama de clases sea una relación de muchos a muchos y:</p> <ul style="list-style-type: none">➤ No se requiere registrar información adicional en la tabla de rompimiento➤ No se requiere relacionar la tabla de rompimiento con ninguna otra tabla.
Documentación:	<p>Se requiere únicamente la documentación del campo en la tabla (no se requiere documentar el primarykey). En el caso de utilizar una secuencia, se colocará la referencia a la misma utilizando la sección SECUECE de Designer.</p>

Campos

Nombre:	<p>El nombre de un campo deberá ser auto-descriptivo y estar en singular.</p> <p>Únicamente la clave primaria y los campos que provienen de un foreignkey pueden comenzar con la palabra código.</p> <p>El nombre no debe incluir valores permitidos, por ejemplo: ESTADO_ACTIVO_INACTIVO, lo correcto sería: ESTADO.</p>
Tipo de Dato:	<p>Revisar las consideraciones del punto "0Tipos de Datos"</p>



Consideraciones:	<p>Todo campo con un conjunto de valores limitados que:</p> <p>a) Varían en el tiempo, harán referencia a la tabla o réplica de la tabla correspondiente. Por ejemplo NUMERO_RUC debe tener un foreignkey hacia RUC_CONTRIBUYENTES, CODIGO_PAIS a ADM_PAISES, etc.</p> <p>Diseño analizará los casos de excepción en los que no es posible crear foreignkeys, por ejemplo:</p> <p>Tablas no replicadas.</p> <p>Tablas replicadas con tipo COMPLETE</p> <p>En tablas de log.</p> <p>b) Son fijos, tendrán obligatoriamente allowablevalues registrados en Designer de la siguiente manera:</p> <p>Value: tres letras representativas de cada palabra, por ejemplo:</p> <p>Para ACTIVO, INACTIVO, serían ACT e INA.</p> <p>Para ACEPTADO_MANUALMENTE y ACEPTADO_AUTOMATICAMENTE serían ACE_MAN y ACE_AUT.</p> <p>Para valores con menos de tres letras se colocará el nombre completo, por ejemplo: SI, NO, 0+.</p> <p>Meaning: nombre completo del valor permitido, por ejemplo ACTIVO, INACTIVO, etc.</p> <p>Caso de Excepción para MDA: en el caso de aplicaciones que cuenten con un modelo UML los allowablevalues deberán ser diseñados en UML a través de Enumeraciones y se cargarán en Designer</p>
-------------------------	---



	<p>como checkconstraints en lugar de allowablevalues.</p> <p>La documentación del meaning deberá ser incluida en el comment del checkconstraint, ejemplo: "ACT = Activo, INA = Inactivo"</p>
Documentación:	<p>Comments: se debe incluir la utilidad del campo y los casos de uso, por ejemplo:</p> <p><i>Código de la Agencia principal del SRI, en la cual se reciben los trámites que llegan de otras oficinas</i></p> <p>Description: este campo es opcional y debe incluir aclaraciones y ejemplos respecto al contenido del campo.</p> <p>En el caso de utilizar una secuencia, se colocará la referencia a la misma utilizando la sección SEQUENCE de Designer.</p>

ForeignKeys

Nombre Campo:	<p>Deberá cumplir la nomenclatura de claves primarias de la tabla referida.</p> <p>Se podrán añadir al final palabras que permitan una mejor descripción, por ejemplo: CODIGO_USUARIO_CREA.</p> <p>En casos de excepción como NUMERO_RUC, se consultará el nombre del campo a Diseño.</p>
Nombre FK:	Nombre generado por Designer.



	[Alias de la tabla hija] "_" [Alias de la tabla padre] "_FK"
Documentación:	Se requiere únicamente la documentación del campo en la tabla (no en el foreignkey).

Índices

Nombre:	<p>[Prefijo de la aplicación] "_" [Alias tabla] "_" [Campos] "_" [Tipo de índice]</p> <p>Campos: se selecciona las tres primeras letras de cada palabra de los campos que conforman el índice hasta donde alcance.</p> <p>Tipo de índice:</p> <p>Índice único: "_UK_I"</p> <p>Índice no único: "_NU_I"</p> <p>Índice tipo Bitmap: "_BM_I"</p> <p>Ejemplo:</p> <p>ADM_USO_APE_NOM_NU_I</p> <p>Si el índice es generado automáticamente por Designer (primarykeys, foreignkeys) se debe mantener el nombre asignado por la herramienta.</p>
Tablespace:	Para aplicaciones antiguas que mantienen los tablespace64K, 2M y 64M, se asignará el índice al tablespace de nombre IDX del mismo tamaño al que fue asignada la tabla.



Consideraciones:	<p>Los índices únicos se utilizarán cuando a más de crear el índice se desee crear un constraint de unicidad de los campos. Es importante considerar que estos índices son más rápidos que los no únicos.</p> <p>Los índices bitmap se utilizarán únicamente en tablas de gran cantidad de registros que no sufren inserciones o actualizaciones transaccionales y el campo indexado tiene pocos valores diferentes.</p>
Documentación:	Los índices no requerirán documentación

Scripts

[Nombre de BDD]	INTER, SRI, SRIDW.
Datos:	<p>Contiene: archivos que involucren sentencias DML: insert, update, delete, etc.</p> <p>Nomenclatura:</p> <p>[yyyy-mm-dd]" _ "[Nombre esquema]" _ "[Descripción]" .sql"</p> <p>El nombre del esquema se refiere al esquema de base de datos en el que se va a ejecutar el script. El nombre del esquema debe estar en mayúsculas y la descripción en minúsculas.</p> <p>Por ejemplo: 2008-03-28_ADM_inserta_roles.sql</p>



Objetos:	
Permisos:	<p>Contiene: archivos de grants. Los grants deberán estar creados también sobre los objetos en Designer.</p> <p>Nomenclatura:</p> <p>[yyyy-mm-dd]"_[Nombre esquema]"_[Descripción]" .sql"</p> <p>El nombre del esquema se refiere al esquema de base de datos en el que se va a ejecutar el script. La sección [Descripción] es opcional.</p> <p>Por ejemplo: 2008-03-28_ADM.sql</p> <p>2008-03-28_ADM_para_roles.sql</p>
Tablas:	<p>Contiene: las características que no se pudo registrar en el objeto, tal como cambios de tablespace para columnas LOB, tipos de dato no soportados por designer como XMLTYPE, etc.</p> <p>En esta sección también se incluirá los scripts de creación de tablas del tipo CDC (Change Data Capture) y tablas TEMPORALES. Estas tablas deben cumplir con el estándar correspondiente.</p> <p>Documentación:</p> <p>Los scripts contarán con la documentación de:</p> <p>COMMENT ON TABLE</p>



	<p style="text-align: center;">COMMENT ON COLUMN</p> <p>Adicionalmente se deberá llenar el campo Description del script con la descripción de la tabla.</p> <p>Nomenclatura:</p> <p>[Nombre Tabla]"_[Descripción]" .sql"</p> <p>La sección [Descripción] es opcional.</p> <p>Por ejemplo:</p> <p>CCL_ARCHIVOS_CONCILIACION.sql</p>
<p>Vistas</p> <p>Materializadas:</p>	<p>Contiene: vistas materializadas que no pudieron ser creadas como objetos en Designer o que a pesar de estar creadas como objetos presentan errores en la generación.</p> <p>Documentación:</p> <p>Los scripts contarán con la documentación de:</p> <p style="text-align: center;">COMMENT ON MATERIALIZED VIEW</p> <p style="text-align: center;">COMMENT ON COLUMN</p> <p>Adicionalmente se deberá llenar el campo Description del script con la descripción de la vista materializada.</p> <p>Nomenclatura:</p> <p>[Nombre Vista Materializada]" .sql"</p> <p>Revisar la nomenclatura y consideraciones del punto "referencia. ¡Error"</p>



	<p>Por ejemplo:</p> <p>ADM_PAISES.sql</p> <p>MAV_MVW_CONTRATOS.sql</p>
Paquetes:	<p>Contiene: réplicas de paquetes de servicios, con menos funciones que el paquete original y que conservan la misma lógica del negocio.</p> <p>Nomenclatura Especificación:</p> <p>[Nombre de Paquete]".pks"</p> <p>Por ejemplo:</p> <p>ADM_PCK_SERVICIOS.pks</p> <p>Nomenclatura Body:</p> <p>[Nombre de Paquete]".pkb"</p> <p>Por ejemplo:</p> <p>ADM_PCK_SERVICIOS.pkb</p>
DDLs:	<p>Contiene: scripts de creación de objetos de base de datos.</p> <p>Consideraciones: únicamente se creará esta carpeta para aplicaciones que se encuentren en designer dentro de "APLICACIONES_PROPIETARIAS", por ejemplo: INTALIO.</p> <p>Nomenclatura:</p>



	[yyyy-mm-dd]" _"[Aplicación]" _"[Versión]" _"[Descripción]" .sql"
--	---

4.13 Pantallas y reportes

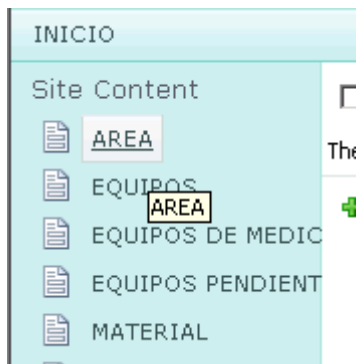
1. Ingresamos a la dirección donde se encuentra alojado el proyecto.



2. Ingresa al Home del Proyecto, donde podemos visualizar el nombre del Sitio, tablas, calendarios, etc.



3. En la barra izquierda podemos visualizar las distintas tablas que podemos dar mantenimiento. Elegimos la tabla Área.



4. Una vez seleccionada la tabla a trabajar se nos despliega una pantalla en el cual elegimos la opción Add document (agregar documento)



☐ Type Name

There are no items to show in this view

[+ Add document](#)

5. Posteriormente se abre la ventana en donde podemos visualizar, agregar, modificar, y guardar registros.



TABLA DE MANTENIMIENTO: ÁREA

Ver Registros

Código

Nombre del Área

*

Guardar Cambios

6. Clic en ver registros.




Ver Registros

7. Se despliega la informacion de la tabla

Código	Nombre del Área
1	Quirófanos
2	Sala de partos
3	Recuperación
4	Neurología
5	Cardiororácica
6	Urgencias
7	Dermatología

8. Para insertar un nuevo elemento damos clic en la opcion Insertar elemento

9. Si se el numero de ID coincide con un registro previamente guardado nos da una alerta.


 No se puede enviar el formulario debido a un error.

7	Dermatología
7	Ginecología

Guardar Cambios

10. Una vez creado correctamente el ID aparece una alerta de que fueron guardados los datos correctamente.



4	<div> Los Datos fueron guardados correctamente.</div> <div>Aceptar</div>	
5		
6		
7		
8	Dermatología	
	Ginecología	

☐ Insertar elemento

11. Por ultimo regresamos a la ventana principal en la cual podemos ver los registros que se encuentran dentro de la tabla.



TABLA DE MANTENIMIENTO: ÁREA

Ver Registros

Código	Nombre del Área
1	Quirófanos
2	Sala de partos
3	Recuperación
4	Neurología
5	Cardiorrácica
6	Urgencias
7	Dermatología
8	Ginecología

☐ Insertar elemento

Guardar Cambios

4.14 Pruebas y depuración



4.14.1 Pruebas de Unidad

Se concentra en la verificación de la unidad más pequeña del diseño del software: el componente o módulo del software.

- La verificación en el diseño se realizó determinando que exista las relaciones entre cada una de las tablas, poniendo especial énfasis en las tablas que contienen la información que almacena las Órdenes de Trabajo, así como también sus Cronogramas.
- Las pruebas de unidad se concentran en la lógica del negocio, en los procesos internos que se relacionan con la finalidad que la relación entre las tablas y almacenamientos de información pueda mantener una estructura adecuada y lógica.

4.14.2 Pruebas de Integración

- Esta prueba se realizó individualmente verificando que todo funcione correctamente, por lo tanto las relaciones existentes en el modelo establecido se verificaron, indicando las claves primarias y foráneas establecidas, a fin de poder verificar que los datos ingresen correctamente.
- La prueba de integración es una técnica sistemática para construir la arquitectura del software, mientras, al mismo tiempo, se aplican las pruebas para descubrir errores asociados con la interfaz. La integración de los datos en la tabla de los clientes, determina la robustez del modelo colocando especial énfasis en las asociaciones de las tablas que tienen relación con la información de las Órdenes de Trabajo.

4.14.3 Pruebas de Validación



- Las pruebas de validación empiezan tras la culminación de la prueba de integración, cuando se han ejercitado los componentes individuales. Se ha terminado de ensamblar el software como paquete y se han descubierto y corregido los errores de interfaz.

Este tipo de pruebas se la realiza en el módulo de seguridad establecido para poder controlar el acceso de los usuarios al sistema. También se estableció para poder verificar la integración que existe de los datos transaccionales con los datos espaciales.

4.14.4 Pruebas del Sistema

- Al final del desarrollo el software se incorpora a otros elementos del sistema (hardware, personas, información) y se realiza una serie de pruebas de integración del sistema y de validación.

La carga de datos en el sistema constituye un factor determinante especialmente al realizar la migración de los datos de las órdenes de trabajo al software.

La integración de los datos y las relaciones entre cada uno de las tablas se verifico, por medio de la consistencia de los datos y la integridad de los mismos.

4.14.5 Prueba de Interfaces Gráficas de Usuario (GUI , GraphicalUser Interface)

- Uso de una lista de chequeo preestablecida:
Se pruebas con la ejecución del manejo de los iconos de acceso al sistema y la integración con la base de datos, a la lista de equipos biomédicos en los reportes respectivos.

4.15 Instalación del sistema



Como parte del Proyecto de grado el estudiante dejará instalando en la empresa un Servidor Active Directory, y un Servidor de Share Point. Desde el cual, las demás estaciones de trabajo perteneciendo al mismo dominio podrán tener acceso al sistema.

Contando con seguridades pertinentes para el acceso de usuarios y datos dentro del sistema.

El requerimiento de las estaciones de trabajo para la utilización del sistema son mínimas, lo único importante es que cuente con un navegador de internet, ya que el entorno gráfico del proyecto es web.

Como requerimiento mínimo para los servidores es que cuenten con un Procesador doble núcleo y cuatro gigas en RAM, cada uno.

4.16 Recopilación y carga de Datos

La empresa auspiciante no cuenta con una base de datos por lo cual se lo debe ingresar manualmente los inventarios de los equipos biomédicos. Este trabajo se realizará por medio de la Secretaría, en este caso la persona encargada es la Srta. Mónica Rodríguez. Para el ingreso de los datos se realizará un cronograma interno pertinente.

4.17 Prueba y Depuración Final en Funcionamiento

Con el ingreso de datos y de usuarios se pudo denotar algunos errores menores, como pequeñas validaciones que se pudieron corregir fácilmente.

Las Pruebas de Funcionamiento, por petición de la empresa auspiciante se seguirán realizando por un lapso más de tiempo, ya que se quiere implementar el mismo sistema en otra institución, de la misma manera que se está realizando en el HCAM.

4.18 Puesta en Marcha del Sistema



Para la puesta en marcha del sistema en su totalidad, se requiere un periodo de tiempo alargado, ya que la cantidad de equipos biomédicos dentro de la institución es bastante alta.

El sistema se lo seguirá utilizando diariamente, hasta que llegue a su totalidad del ingreso de los inventarios.

Por el momento el usuario que está habilitado para el uso de sistema es la Srta. Mónica Rodríguez ya que es la persona responsable del ingreso de estos inventarios.

4.19 Capacitación al Usuario Final

La metodología que se utilizará será de Capacitación Personal, ya que las personas que serán capacitadas es un grupo pequeño.

Esta capacitación se la realizara con prácticas y uso correcto del nuevo sistema que se está implementando en la empresa. Todas las personas capacitadas deben ser sometidas a evaluaciones antes de que utilicen solos el sistema dentro de la empresa.

El uso correcto de esta herramienta será en beneficio de la empresa, por ello el manejo de este software por parte de los usuarios debe ser profesional.

4.20 Capacitación el Personal Técnico

En la empresa auspiciante no se cuenta con un Departamento de Sistemas, por ello la capacitación al personal Técnico no se lo podrá realizar.

Por parte del estudiante, se compromete a dar un soporte Técnico durante seis meses posteriores a la graduación, para la correcta utilización del sistema.

CAPITULO V



PRINCIPALES IMPACTOS

5.1 Impacto Científico

Con el desarrollo del proyecto se tendrá un gran impacto sobre los usuarios del software, ya que estos son Doctores jefes de Áreas, Ingenieros y personas relacionadas a los procesos de mantenimiento de equipos biomédicos. Esto se logrará a través de un chequeo continuo de los procesos cumplidos por medio de cronogramas automáticos del sistema.

5.2 Impacto Educativo

A través del desarrollo del proyecto de grado se ve la necesidad de la utilización de distintas herramientas y plataformas como Windows Server 2008, Share Point, Active Directory. Viéndose como punto de partida para las próximas generaciones su aprendizaje e implementación.

5.3 Impacto Técnico

Para la creación del proyecto se llegó a utilizar distintos software para desarrollar el mismo, tales como Rational Rose, Windows Server 2008 R2, SQL server, Share Point.

Esto a su vez en cierto modo obliga a la empresa auspiciante a crear un departamento de sistemas para poder controlar y mantener activo el proyecto.

5.4 Impacto Tecnológico



El desarrollo del proyecto tuvo un gran impacto sobre el estudiante ya que lo obligo a tener conocimiento y experticia en herramientas que desconocía. Esto a su vez, combinado con los conocimientos aprendidos a lo largo de la carrera estudiantil, es lo que logro para realizar de manera óptima el desarrollo del proyecto de grado.

5.5 Impacto Empresarial

Siendo la visión emprendedora en relación que en un futuro, contar con una empresa, se ha cristalizado el primer paso que consiste en tener un aplicativo que maneje las reglas del negocio de manera adecuada adaptándose así a las estrategias del negocio.

5.6 Impacto Social

El desarrollo del proyecto tiene un gran impacto sobre los usuarios de este tipo de software ya que el mismo está enfocado a la medicina. En el mercado nacional e internacional no se cuenta con una gran variedad de software que controlen los procesos de mantenimientos de equipos biomédicos en centros hospitalarios. Esto hace que el software tenga un gran impacto sobre las personas que lo utilizarán. Ya que este ha sido creado por un desarrollador nacional con todos los conocimientos de los procesos internos.

5.7 Impacto Económico

El mayor de todos los impactos que se puede tener es el económico, ya que uno de los requerimientos de la empresa auspiciante al momento de ganar el contrato en el Hospital Carlos Andrade Marín es contar con un software de control de mantenimientos de equipos biomédicos. En el mercado no se encuentra un tipo de software así por lo que los pocos



software semejantes tienen un elevado costo por lo que con el desarrollo de este tipo de proyecto de grado la empresa se beneficiará gratuitamente.

5.8 Conclusiones

- La aplicabilidad de los pasos que se siguen en ingeniería de software fueron determinantes ya que mediante el análisis me permitieron discernir todo el flujo de información que se ejecuta en el HCAM y así mismo poder diseñar adecuadamente los procesos, sus relaciones, las bases de datos y sus objetos a fin de poder obtener una organización metódica y bien estructurada del sistema realizado.
- El conocer las reglas del negocio que se ejecutan diariamente en la Institución me permitieron definir métodos y procedimientos acordes con la realidad de los procesos enseñanza aprendizaje, lo que se complementó con la validación de los mismos en el campo, por tanto el Sistema de Mantenimiento de Equipos Hospitalarios, abarco sin duda todas las tareas que ayudan en cada uno de los procesos y sus relaciones entre tutor, alumno, empresa que son actores de todo el entorno del sistema para la empresa.
- La asignación de trabajo a los técnicos no se lo realizaba de manera adecuada, ya que muchas veces se asignaba mucho trabajo y otras veces poco trabajo a los técnicos. Con el sistema se logró la asignación uniforme de trabajo para cada técnico.
- En conclusión la empresa presentaba una falta de agilidad con los procesos de mantenimiento, debido a que todo se lo llevaba en papeles, esto producía mal estar en la coordinación directa del Hospital Carlos Andrade Marín.



5.9 Recomendaciones

- Automatizar todos los procesos dentro de la empresa con el fin de optimizar el tiempo y los niveles de seguridad adecuados para la protección de toda la información que se maneja.
- Todos los procesos que se toman en cuenta para la realización del sistema deben estar bien definidos y adecuadamente bien documentados, las cuales mejorará el desempeño del sistema y su adecuada adaptación al medio.
- Realizar un seguimiento en todas las etapas de elaboración del sistema para cumplir con las metas planteadas y definir estándares en la parte de desarrollo para obtener un sistema eficiente.
- Para la correcta utilización del sistema se deberá dar una nueva charla después de su instalación. Esto ayudará a despejar todas las dudas que pueden quedar con el uso del mismo.



CAPITULO VI

ASPECTOS ADMINISTRATIVOS

6.1 Presupuesto

Ingreso dinero \$		Egresos dinero \$	
Detalle	Valor	Detalle	Valor
Recurso Propio	\$ 3600	Estación de trabajo	\$900
Aporte institución de apoyo	\$ 0	Impresora	\$50
		Internet	\$150
		Varios (Luz, agua, teléfono)	\$80
		Transporte	\$50
		Material de escritorio	\$20
		Impresiones	\$50
		Empastado	\$60
Total ingresos:	\$ 3600	Total egresos:	\$ 1360

Presupuesto

Tabla Nº: 10

Fuente: Estudiante



6.2 Cronograma

El siguiente Cronograma se lo podrá visualizar en el siguiente anexo.

Anexo: 3

Título: Cronograma de Actividades

Fuente: Estudiante

6.3 Bibliografía

- Burbano Homero, Altamirano Edwin, (2005), Manual de Métodos y Técnicas de Investigación, Imp.Eurocolor, Ecuador – Quito.
- Villalba C. (2004), Guía para la elaboración de anteproyectos y proyectos, Edición II, Sur editores Ecuador – Quito.
- NilssonNils, (1987), Principios de Inteligencia Artificial, Editorial Díaz de Santos, Edición II, España – Madrid,
- Siemens Aktiengesellschaft, (1984), Sistemas Expertos: Introducción a la Técnica y Aplicación, Editorial Marcombo, Edición I, España – España.
- Kendall F. (2001), Músculos, pruebas, funciones y dolor postural, España – Marban.

6.4 Web grafía

- **Metodología RUP**
<http://www.scribd.com/doc/297224/RUP>.
- **Metodología RUP**
<http://www.slideshare.net/msch/utilizando-metodologia-rup-parte1>
- **Sistema Expertos**
http://www.geocities.com/shop/sistemas_expertos.hmt



- **Sistemas Expertos**

<http://www.cinefantastico.com/nexus7/ia/expertos.htm>

- **Manuales de principios – Programación php**

www.programacionphp.net/

- **Principios de diseño de base de datos**

<http://www.programacion.com/articulo/principiosdedisenodbasesdedatos264.html>

- **Servidores Web**

http://www.yoreparo.com/foros/disenos_web/que-es-apache-t38371.html

6.5 Glosario de Términos

- UML

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables

- SCRIPT

Es un programa usualmente simple, que por lo regular se almacena en un archivo de texto plano.

- DICCIONARIO DE DATOS

Es un conjunto de metadatos que contiene las características lógicas y puntuales de los datos que se van a utilizar en el sistema que se programa, incluyendo nombre, descripción, alias, contenido y organización.

6.5 Anexos