



UNIVERSIDAD CATÓLICA DE TEMUCO
FACULTAD DE CIENCIAS.

SISTEMA DE INVENTARIO
HOSPITAL DR. HERNÁN HENRÍQUEZ ARAVENA

por

CHRISTIAN ROBERTO PARRA VILLENA

Trabajo de Título presentado a la
Facultad de Ciencias de la Universidad Católica de Temuco
Para Optar al Título de Ingeniero de Ejecución en Informática
- Temuco, 2005 -

COMISIÓN EXAMEN DE TÍTULO

Este Examen de Título ha sido realizado en la Escuela de Informática.

Presidente Comisión:

Sr. Oriel Herrera Gamboa
Ingeniero Civil Industrial M/Informática
Doctor (c) en Ciencias de la Computación

Profesor Guía:

Sr. Roberto Conejeros Gómez
Ingeniero Civil Industrial M/Informática
Jefe Depto. Informática Hospital Dr. Hernán
Henríquez A.

Profesor Informante:

Sr. Alberto Caro Saldivia
Ingeniero Civil en Informática

Secretario Académico:

Sr. Alberto Caro Saldivia
Ingeniero Civil en Informática

Temuco, Enero de 2006




INFORME TRABAJO DE TÍTULO

TÍTULO : "Sistema de Inventario Hospital Dr. Hernan Henríquez Aravena"
ALUMNO : Christian Roberto Parra Villena

En mi condición de profesor guía de este trabajo puedo efectuar las siguientes observaciones:

- El Informe de tesis presentado se ve bien estructurado y ordenado, según una metodología previamente seleccionada.
- Se observa una presentación muy interesante y útil para el Hospital Dr. Hernán Henríquez Aravena, dado que hasta la fecha al estar en etapa de explotación ha permitido el aporte valioso de un Sistema de Información más para el Hospital Temuco.
- Se sugiere para el informe de trabajo de título corregir observaciones menores y crear para posterior un manual de usuario.

De acuerdo a estas consideraciones califico este trabajo con nota 6,8 (Seis coma ocho)


Roberto Iván Conejeros Gómez
Profesor Guía

Temuco, 4 de Enero de 2006



UNIVERSIDAD CATÓLICA DE TEMUCO
FACULTAD DE CIENCIAS

DEDICATORIO

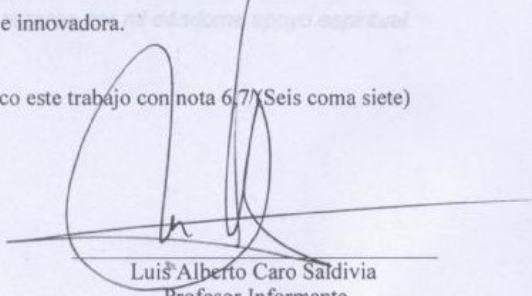
INFORME TRABAJO DE TÍTULO

TÍTULO : "Sistema de Inventario Hospital Dr. Hernan Henriquez Aravena"
ALUMNO : Christian Roberto Parra Villena

En mi condición de profesor informante de este trabajo puedo efectuar las siguientes observaciones:

- El trabajo esta ordenado y bien estructurado y se han usado herramientas metodológicas adecuadas.
- Se aprecia un buen nivel de dedicación y claridad en los objetivos planteados.
- Se sugiere una mejora en los gráficos y esquemas utilizados.
- Se sugiere que utilice un esquema grafico (enfoque sistémico) que muestre los distintos agentes que interactúan con el nuevo sistema de inventario y sus macro flujos de información.
- La propuesta es atractiva, muy útil e innovadora.

De acuerdo a estas consideraciones califico este trabajo con nota 6,7 (Seis coma siete)



Luis Alberto Caro Saldivia
Profesor Informante

Temuco, 5 de Diciembre de 2005

DEDICATORIO

*Este trabajo esta dedicado en primer lugar a Dios,
por permitirme llegar a esta instancia,
a mi familia por todo su apoyo,
a mi madre por su constante preocupación,
a mi padre por sus palabras de aliento,
a mi polola Paola por su cariño y su ánimo,
y a quienes rezaron por mi dándome apoyo espiritual.*

AGRADECIMIENTOS

Al departamento de Informática y sus funcionarios quienes durante todo el desarrollo del Trabajo de Título se volvieron una gran ayuda entregándome sus consejos y experiencia.

Al Sr. Neldo Jofre por su apoyo en la parte de flujo de trabajo del inventario.

A mi polola Paola por su ayuda en la parte de formato y estructura del informe.

Y a todos los que aportaron su granito de arena para que este trabajo haya sido posible.

ÍNDICE

Contenido	Página
ÍNDICE DE CONTENIDOS	i
ÍNDICE DE FIGURAS	v
ÍNDICE DE TABLAS	vii
RESUMEN	viii
INTRODUCCIÓN	1
1 OBJETIVOS	3
1.1.1 OBJETIVO GENERAL	3
1.1.2 OBJETIVOS ESPECÍFICOS	4
2 METODOLOGÍA	5
3 DESARROLLO	6
3.1 TOMA DE REQUERIMIENTOS	6
3.1.1 CICLO DE VIDA	8
3.1.2 SISTEMA ANTIGUO	12
3.1.3 OTROS PROCESOS	18
3.2 ANÁLISIS	20
3.2.1 REQUERIMIENTOS SISTEMA ANTERIOR	20
3.2.2 REQUERIMIENTOS NUEVO SISTEMA	21
3.2.2.1 Ingreso de Bienes	23
3.2.2.2 Corrección de Datos	23
3.2.2.3 Consulta de Bienes	24
3.2.2.4 Hojas Murales	24
3.2.2.5 Movimiento de Bienes	25
3.2.2.6 Baja de Bienes	25
3.2.2.7 Ver Movimientos de Bienes	25
3.2.2.8 Ver Baja de Bienes	26
3.2.2.9 Búsqueda de Servicios	26
3.2.2.10 Búsqueda de Dependencias	26
3.2.2.11 Servicios y dependencias	26

3.2.2.12 Administración	27
3.2.2.13 Agregar Dependencias	27
3.2.2.14 Agregar Servicios	27
3.2.2.15 Agregar Usuarios	27
3.2.2.16 Modificar Dependencias	28
3.2.2.17 Modificar Servicios	28
3.2.2.18 Modificar Usuarios	28
3.2.2.19 Eliminar Dependencias	28
3.2.2.20 Eliminar Servicios	28
3.2.2.21 Eliminar Usuarios	29
3.2.2.22 Limpiar Archivos	29
3.3 DISEÑO	29
3.3.1 DESCRIPCIÓN DE CAMPOS	33
3.3.1.1 Código de Barra	33
3.3.1.2 Código Antiguo	33
3.3.1.3 Nombre	34
3.3.1.4 Descripción	34
3.3.1.5 Marca	34
3.3.1.6 Modelo	35
3.3.1.7 Serie	35
3.3.1.8 Estado	35
3.3.1.9 Propiedad	35
3.3.1.10 Servicio	36
3.3.1.11 Dependencia	36
3.3.1.12 Grupo- Cuenta- SubCuenta	37
3.3.2 DISEÑO DE PANTALLAS	37
3.3.2.1 Acceso al Sistema	37
3.3.2.2 Ingreso dependencias	38
3.3.2.3 Ingreso servicios	38
3.3.2.4 Ingreso usuarios	39
3.3.2.5 Ingreso bienes	40
3.3.2.6 Vista de Ingreso	41

3.3.2.7 Consultas	42
3.3.2.8 Hojas Murales	44
3.3.2.9 Traspaso bienes, Baja Bienes	45
3.3.2.10 Ver Movimiento, Bajas	46
3.3.2.11 Búsqueda Servicios, Dependencias	47
3.3.2.12 Servicios y Dependencias	48
3.3.2.13 Modifica Dependencia, Servicio	48
3.3.2.14 Modifica Usuario	49
3.3.2.15 Eliminación de Dependencia, Servicio	50
3.3.2.16 Eliminar Usuario	51
3.3.3 DISEÑO DE LA BASE DE DATOS	51
3.3.3.1 Tablas	54
3.3.3.1.1 Bienes	54
3.3.3.1.2 Usuarios	56
3.3.3.1.3 Establecimientos	56
3.3.3.1.4 Servicio	57
3.3.3.1.5 Dependencia	57
3.3.3.1.6 Encargados	58
3.3.3.1.7 Movimientos	58
3.3.3.1.8 Clasificación	59
3.3.4 MODELO RELACIONAL INVENTARIO	60
3.4 IMPLEMENTACIÓN	61
3.4.1 LENGUAJE PHP	61
3.4.2 PHP Y EL INVENTARIO	65
3.4.3 APACHE	65
3.4.3.1 Módulos	67
3.4.4 MYSQL	68
3.4.5 JAVASCRIPT	69
3.4.6 CSS	71
3.4.7 MODELO CLIENTE SERVIDOR	73
3.4.7.1 Ventajas de la arquitectura cliente-servidor	75
3.4.8 APLICACIONES WEB	76

3.4.9 SOFTWARE NECESARIO	77
3.4.9.1 HARDWARE	77
3.4.9.2 SOFTWARE	77
3.4.10 INSTALACIÓN DEL SOFTWARE	78
3.4.10.1 Instalando apache	78
3.4.10.2 Instalando MySQL	79
3.4.10.3 Instalando PHP	83
3.4.11 EDITOR	86
3.4.12 FUNCIONES IMPORTANTES	87
3.4.13 IMPLEMENTACIÓN DEL SISTEMA	88
3.4.13.1 Modulo ingreso	88
3.4.13.2 Modulo consulta de bienes	92
3.4.13.3 Modulo hojas murales	94
3.4.13.4 Modulo Traspaso de bienes	96
3.4.13.5 Módulo Baja Bienes	99
3.4.13.6 Modulo Ver Movimientos de Bienes	101
3.4.13.7 Módulo Ver Baja de Bienes	102
3.4.13.8 Modulo Búsqueda de Servicio	102
3.4.13.9 Módulo Búsqueda de Dependencias	103
3.4.13.10 Módulo Servicios y Dependencias	104
3.4.13.11 Módulo Agregar Dependencias	104
3.4.13.12 Módulo Agregar Servicios	105
3.4.13.13 Módulo Crear Usuarios	106
3.4.13.14 Módulo Modificar Dependencias	107
3.4.13.15 Módulo Modificar Servicio	107
3.4.13.16 Módulo Modificar Usuario	108
3.4.13.17 Módulo Eliminación de Dependencias	109
3.4.13.18 Módulo Eliminación de Servicio	109
3.4.13.19 Módulo eliminar usuario	110
3.4.13.20 Módulo Limpiar archivos	111
3.5 PRUEBAS	112
3.5.1 LOGIN USUARIO	113

3.5.2 CAMPOS OBLIGATORIOS	113
3.5.3 BÚSQUEDA PARA PROCESO	114
3.5.4 CONSULTA DE BIEN	115
3.5.5 INGRESO DEPENDENCIA	116
4 CONCLUSIONES	117
5 BIBLIOGRAFIA	119

ÍNDICE DE FIGURAS

Figura	Página
1. Flujo de Vida de un Bien	10
2. Flujo de trabajo del encargado de inventario	11
3. Menú principal Contabilidad	12
4. Menú principal de Inventario	13
5. Mantención de Archivos	14
6. Entrada de Datos	15
7. Consultas por pantallas	16
8. Emisión de Informes	18
9. Código de Barra	21
10. Diagrama de Flujo Inventario	22
11. Diseño Menú Principal	30
12. Diseño Ingreso al Sistema	37
13. Diseño Ingreso Dependencia	38
14. Diseño Ingreso Servicio	38
15. Diseño Ingreso Usuario	39
16. Diseño Ingreso bienes	41
17. Diseño Vista de Ingreso	42
18. Diseño Consulta Bienes	43
19. Diseño Hojas Murales	45
20. Diseño Traspaso Bienes	46
21. Diseño Ver Movimientos Bienes	47
22. Diseño Búsqueda Servicios	47
23. Diseño Servicios y Dependencias	48
24. Diseño Modifica Dependencia	49
25. Diseño Modifica Usuario	50
26. Diseño Elimina dependencia	50
27. Diseño Elimina Usuario	51
28. Modelo Relacional	60

29. Ejemplo de script PHP	65
30. Modelo Cliente-Servidor	75
31. Modelo Aplicación Web	76
32. Verificación de la instalación de PHP	85
33. Editor PHP	87
34. Formulario de Ingreso	91
35. Vista de Ingreso	91
36. Búsqueda de Bienes	92
37. Hoja Mural	96
38. Traspaso de Bienes	97
39. Baja de Bienes	100
40. Movimiento de Bienes	101
41. Baja de Bienes	102
42. Búsqueda de Servicios	103
43. Búsqueda de Dependencias	103
44. Servicios y sus dependencias	104
45. Ingreso de dependencia	105
46. Ingreso de Servicio	106
47. Ingreso de Usuario	106
48. Modificación de dependencia	107
49. Modifica Servicio	108
50. Modifica Usuario	108
51. Elimina dependencia	109
52. Elimina Servicio	110
53. Elimina Usuario	111
54. Elimina archivos temporales	112
55. Error login usuario	113
56. Error campo obligatorio	114
57. Error en proceso	115
58. Error Búsqueda registro	115
59. Error ingreso dependencia	116

ÍNDICE DE TABLAS

Tabla	Página
1. Clasificación de bienes (fragmento)	17
2. Esquema de registro de muebles	52
3. Esquema de registro vehículos	53
4. Estructura Tabla Bienes	55
5. Estructura Tabla usuarios	56
6. Estructura Tabla Establecimientos	56
7. Estructura Tabla Servicio	57
8. Estructura Tabla Dependencia	57
9. Estructura Tabla Encargados	58
10. Estructura Tabla Movimientos	59
11. Estructura Tabla Clasificación	59
12. Algunas bases de datos soportadas por PHP	64

RESUMEN

En la actualidad, para cubrir las necesidades de una organización, existe una gran variedad de tecnologías de información con la que se podría construir un Sistema y migrar así desde uno más antiguo. En el caso del Sistema de Inventario del Hospital Hernán Henríquez Aravena, se aprovechó la introducción de una nueva forma de marcar los bienes denominado Código de Barra que, junto con la realización de un catastro completo de los bienes existentes, se creó un Sistema que procesará y almacenará los nuevos datos los cuales se podrán ver como una foto en un momento determinado de todos los bienes y sus características. Con esto se introdujo un avance tecnológico respecto del Sistema antiguo.

Este nuevo sistema utiliza la Intranet del establecimiento y la tecnología Web. Para esto se utilizó un servidor Web, base de datos y lenguajes de desarrollo. Esta plataforma permite una mayor flexibilidad ya que se pueden incorporar diversas tecnologías como audio, imágenes, películas, texto, animaciones, etc.; lo que permite una gran gama de posibilidades a la hora de construir software, además de la ventaja que presentan los sistemas cliente-servidor.

Viendo estas ventajas se diseñó y se construyó el sistema, generando así el modelo relacional de la base de datos y los módulos de la capa de aplicación. El diseño correspondiente a colores, imágenes de fondo, banners, etc., serán encargados a un diseñador externo por lo que no se consideró en esta

oportunidad para el desarrollo del trabajo.

El sistema se descompone en módulos que realizan acciones concretas como ingresos de bienes, modificaciones, eliminaciones, etc.,. Dándole un enfoque modular, será posible crear en el futuro nuevos módulos con nuevas funcionalidades o mejorar las ya existentes todo esto para tener finalmente un informe tangible llamado *hoja mural*, la cual es distribuida a cada oficina del Hospital.

Con esto se logrará un trabajo más limpio y flexible en lo que se refiere al inventario, tratando de hacer esta función más fácil al encargado. Esto también constituye una gran ventaja en el área de informática ya que la base de datos puede ser manipulada por una gran cantidad de lenguajes (no sólo cobol como en el sistema antiguo) teniendo la posibilidad más adelante de poder migrarla a una más potente. Todo esto finalmente repercute en los encargados de departamentos a los cuales les será más fácil hacer una retroalimentación con el sistema y sincronizar la información con la realidad.

Palabras claves: *sistema, módulo, inventario.*

INTRODUCCIÓN

En toda organización que se desee establecer un orden y una buena administración de las especies que la conforman, debiera llevar un control de todo lo que ingresa y lo que sirva como herramienta laboral así como también lo que ya ha cumplido una vida útil y por lo tanto debe retirarse de circulación. Todo este movimiento de bienes que pueden incluir: equipos, mobiliario, ropa, utensilios, etc., son fundamentales para el buen desempeño de sus funcionarios, lo que a su vez, repercute directamente en los usuarios finales.

Toda esta complejidad justifica un buen sistema capaz de manejar y dar soluciones a esta problemática de manera eficiente y cubriendo así todos los requerimientos percibidos, haciendo uso además de la tecnología al alcance de los desarrolladores como son las aplicaciones Web, tan útiles en entornos distribuidos como lo es el Hospital Dr. Hernán Henríquez A. Esto significa una gran ventaja si se desarrolla sobre una intranet la que está disponible desde cualquier punto dentro del Hospital, también usando Software Open Source, lo que libera el pago de licencias y por ende reduce los costos prácticamente a cero. Con todas estas herramientas se pretende construir un sistema mejorable y perfeccionable a través del tiempo que sea además de fácil uso para el usuario.

En este trabajo, se muestran los pasos necesarios para la construcción del software empezando por la toma de requerimientos, en donde se verá como se

realiza el trabajo de inventario en la parte administrativa como en la práctica incluyendo el uso del sistema anterior.

En el análisis se estudiará esta estructura y como representarla de la mejor forma posible en un nuevo esquema informático, para esto se deben definir acciones, y los datos necesarios para la entrada y salida de la información que será útil en la tarea de inventario.

Luego se diseñará un modelo de la aplicación que represente exactamente lo que hará el sistema. Esto incluye, los módulos, funciones, validaciones, y el modelo de la base de datos que se deberá crear para este fin. También se diseñaran las ventanas que serán donde se presente la información al usuario, tanto la de entrada como la de salida.

Una vez que se tenga listo el diseño se pasará a la etapa de implementación, en donde se describe todo lo necesario para desarrollar el sistema y que pueda funcionar según lo esperado, esto incluye máquina, sistema operativo, servidores Web, compiladores, bases de datos.

Solo faltaría evaluar el sistema ingresando datos reales y ver como responde, esta etapa se denomina prueba ya que aquí se detectaran los problemas que pueda presentar y así corregirlos antes de que pueda ser puesto en marcha. Esto produce una retroalimentación la que servirá para depurar los errores y hacerlo un sistema robusto y confiable.

1 OBJETIVOS

1.1 OBJETIVO GENERAL

El objetivo general de este trabajo de Título es desarrollar e implementar un nuevo sistema de inventario acorde con las necesidades descritas por el Hospital Dr. Hernán Henríquez Aravena utilizando la plataforma Web.

1.2 OBJETIVOS ESPECÍFICOS

- Obtener los requerimientos necesarios para construir el sistema de acuerdo a las necesidades actuales. Esto implica nuevos tipos de bienes, nueva forma de hacer las altas, etc.

- Analizar los requerimientos obtenidos para optimizarlos y poder traducirlos a un diseño el cual posteriormente será implementado en lenguaje computacional.

- Implementar el diseño propuesto sobre la plataforma elegida.

- Entregar el producto a la comunidad del Hospital Dr. Hernán Henríquez Aravena, para que contribuya al buen manejo de sus bienes.

2 METODOLOGIA

La metodología usada para desarrollar este trabajo, esta basado en el “Proceso Unificado de Desarrollo”, tomando en cuenta los pasos fundamentales para la construcción de software los cuales son: Captura de requerimientos, Análisis, Diseño, Implementación, Pruebas.

Esta aplicación se implementará utilizando sistemas operativos y software Open Source. En el caso del sistema operativo, se usará Linux, un servidor Web, lenguajes del lado del servidor y lenguajes del lado del cliente. Se implementará soluciones basados en aplicaciones Web de estructura cliente-servidor utilizando los servidores y redes de intranet existentes. Por la parte de programación, ésta se hará de modo estructurado con indentación bien marcadas para facilitar su lectura y depuración.

Debido a la nueva manera utilizada de marcar los bienes, o sea el código de barra, en el futuro este sistema se complementará con otra unidad de consulta on-line utilizando dispositivos móviles tipo palm, para hacer uso de consultas online usando un lector de código de barra. Creando un pequeño sistema en el dispositivo, este podrá consultar la base de datos de inventario obteniendo la información actualizada que permita tomar decisiones acertadas de acuerdo a la situación.

3 DESARROLLO

3.1 TOMA DE REQUERIMIENTOS

En esta etapa tan importante para el desarrollo del sistema se han tomado en cuenta los dos procesos importantes para realizar la toma de requerimientos: el flujo de trabajo realizado por el encargado de inventario, y el sistema de inventario antiguo.

Primero se describirá el entorno en el cual se desarrolla el flujo de trabajo que se lleva a cabo en la sección de contabilidad, que es a donde está enfocado este sistema, acá lo que se debe conocer es la forma en que el encargado de inventario realiza su trabajo.

Todo este proceso comienza cuando después de buscar y elegir los bienes necesarios, se procede a hacer la compra a la entidad correspondiente, cuando se realiza esta compra los bienes llegan al hospital a la bodega de abastecimiento en donde permanecen hasta que se realiza el proceso de alta del bien del cual es encargado la persona de inventario de contabilidad. Esta persona debe concurrir a la bodega con la factura u orden de compra para certificar que todo lo que se estipula ahí efectivamente allá llegado y en buen estado, luego se procede a inventariar el bien como se ha pensado para este nuevo sistema ya no con un número escrito con lápiz plumón sino con una etiqueta plástica que contiene un código de barra que identifica como único a

cada bien que existe en el recinto hospitalario.

Una vez que ya ha sido marcado se debe decidir su destino dentro del hospital esto quiere decir tiene que enviarse a un servicio y dentro de este a una dependencia los cuales están ya definidos dentro de la estructura administrativa, una vez establecido estos parámetros el encargado debe ingresar al sistema de inventario (antiguo) todos estos datos obtenidos durante la compra y la definición de su destino final dentro de la institución. Con esto el bien ya podrá ser llevado a su destino físico para empezar a hacer uso de él.

Cabe destacar que en cada dependencia existe una “hoja mural” en la cual se especifican los bienes que pertenecen a ese lugar, en esta hoja se muestran algunos detalles asociados a cada bien, los cuales son extraídos del sistema de inventario por lo que el ingreso correcto de estos datos es muy importante a la hora de publicar estas hojas murales, ya que el jefe de servicio es responsable de lo que allí aparece una vez que se a corroborado que todo lo que indica la hoja existe efectivamente en el lugar.

Se debe actualizar esta hoja cuando ha sido agregado un nuevo bien, esto lo debe hacer el encargado de inventario para mantener la coherencia entre la parte física y la parte administrativa lo cual puede evitar problemas y malos entendidos.

3.1.1 CICLO DE VIDA

La vida de un determinado bien ya sea tangible o intangible, comienza por una necesidad de este de parte del servicio o dependencia en donde haga falta, por esto para que un bien llegue a su destino y por lo tanto pase a formar parte de los bienes del hospital siendo registrado por inventario este debe pasar por algunos pasos que a continuación se describirán las vías por las cuales un bien puede llegar a un determinado lugar.

El primer paso para obtener un bien es pedirlo. Para hacer esto hay varias vías:

- **Entorno laboral:** se concursa un proyecto y la persona se encarga por sus propios medios de cotizar, hablar con el proveedor, presentar 3 cotizaciones y la comisión de entorno laboral aprueba o rechaza la compra. Una vez que se compra esto llega a la bodega de abastecimiento.

- **Necesidad del servicio:** el jefe de servicio presenta una solicitud de compra de bien muebles y la subdirección administrativa aprueba o rechaza esta petición siendo ésta acompañada de sus respectivas cotizaciones, una vez que se aprueba esto llega a la Bodega de Abastecimiento.

- **Compras por necesidad de mantención y reposición:** esto se refiere a materiales eléctricos, carpintería, gasfitería, petróleo para la caldera, esto se hace por intermedio de bodega solicitándolo a abastecimiento, esto forma parte de un programa mensual, una vez comprado llega a bodega de abastecimiento.

- **Compras de entrega inmediata:** son necesidades urgentes de servicios como alimentación, artículos de escritorio y varios, lo que se compra con un fondo de caja chica. El servicio que lo requiere lo solicita por intermedio de abastecimiento.

- **Compras programadas para reponer artículos de insumos (medicamentos, pabellón, ropa):** esto es lo que mas abarca dentro de las compras, esto se hace a través de **Chile Compra**. Esto lo hace directamente abastecimiento sección de bodega de insumos.

- **Fabricación interna:** se manda por intermedio de mantenimiento la fabricación y confección de muebles o mesas de trabajo para oficinas recién entregadas.

- **Compra equipos médicos:** se compra por medio del jefe de abastecimiento, del jefe de servicio o jefe de pabellón, del proveedor, del servicio técnico de la empresa proveedora de este instrumento. Se forma una comisión que hace el estudio donde se analizan garantías,

capacitación, etc.

Todas estas vías pueden seguir los bienes dependiendo su clasificación como puede ser ropa, alimentos, material de escritorio, equipo médico, muebles, etc. Luego de hacer el pedido todos estos bienes llegan al mismo lugar llamado bodega Anexo Abastecimiento donde es recibido por el encargado de la bodega el cual solicita la ayuda del técnico del servicio de mantenimiento y se solicita la presencia de la persona solicitante del producto para aprobarlo. Luego se llama a la persona de inventario. El encargado de anexo abastecimiento designa o el propio servicio se encarga de ir a retirarlo. Luego la persona de inventario lo ingresa al sistema en donde queda formalmente registrado en alta.

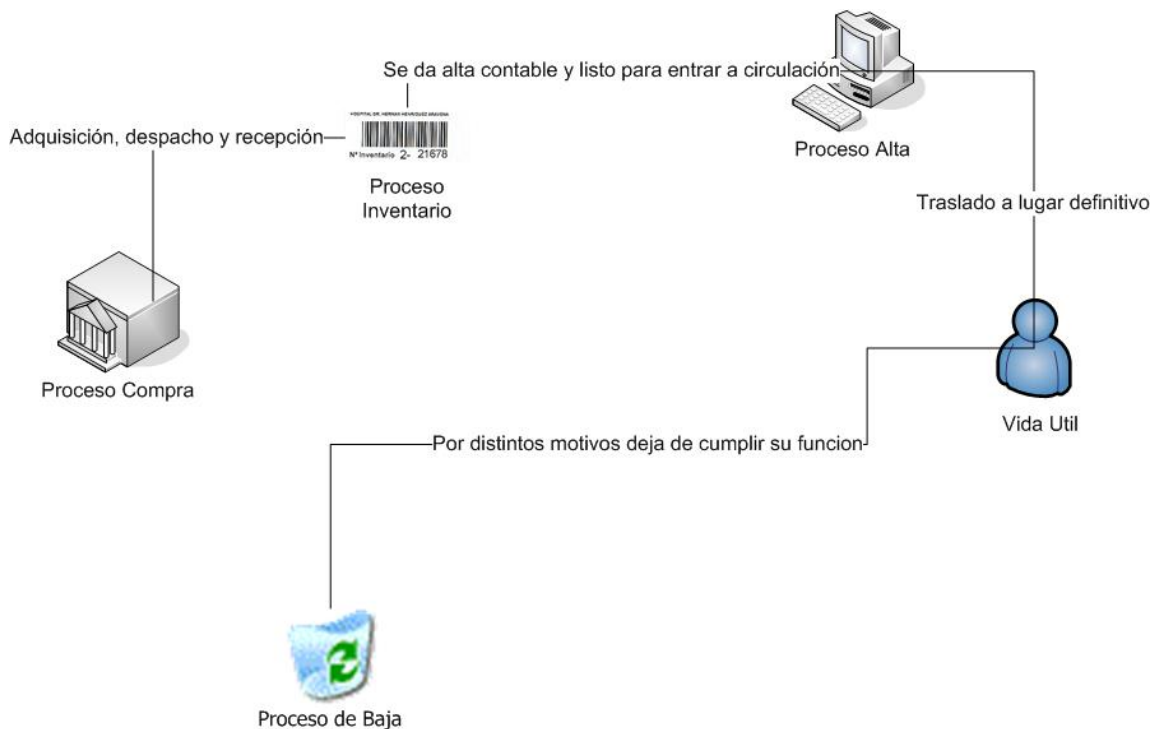


Figura 1. Flujo de Vida de un Bien.

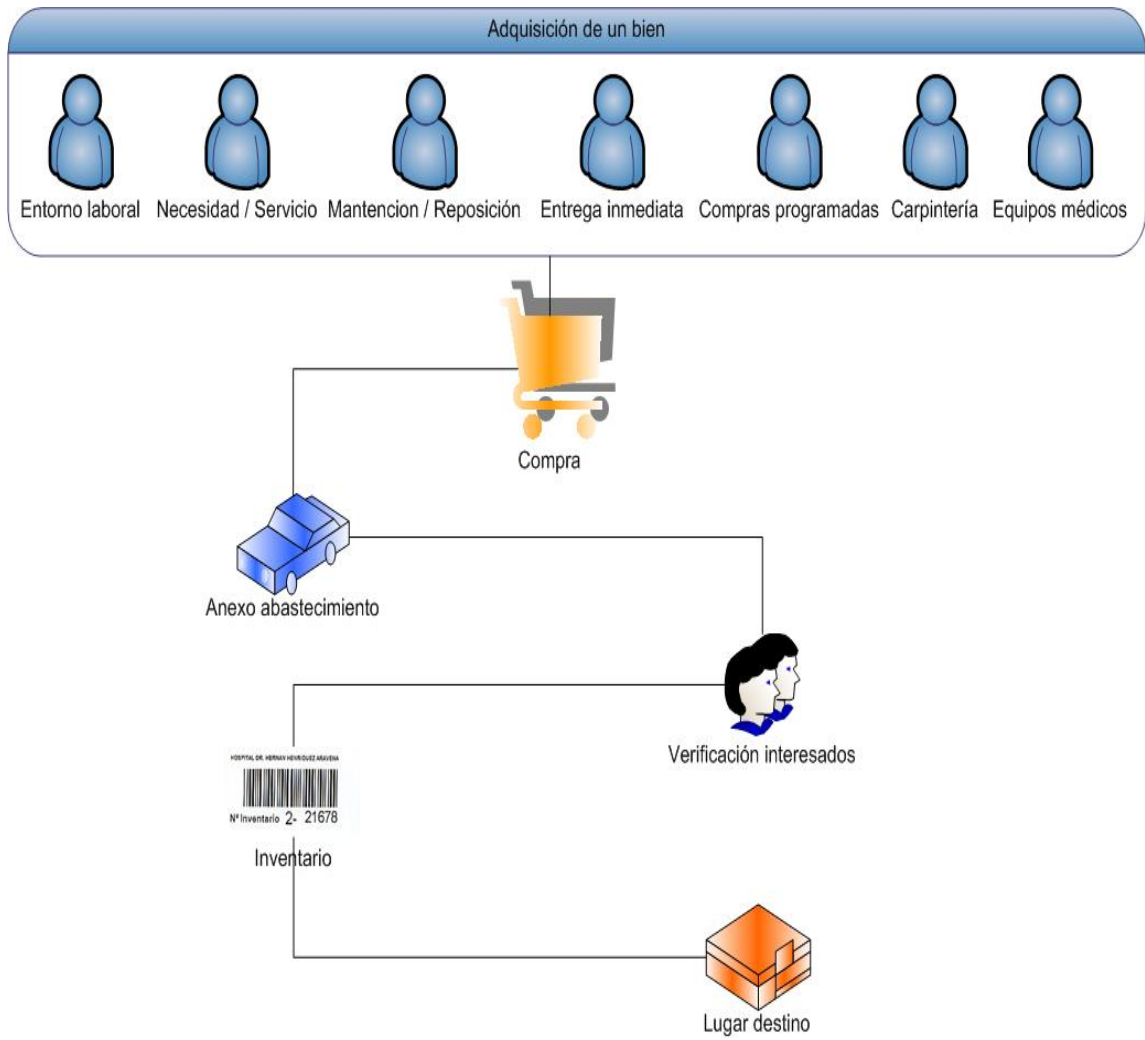


Figura 2. Flujo de trabajo del encargado de inventario.

3.1.2 SISTEMA ANTIGUO

Está desarrollado en cobol. Almacena la información en una base de datos Cobol. Este programa se ejecuta en un Terminal que ofrece el servidor AIX (aquí es donde está instalado el programa y la base de datos) en donde se accede con una cuenta especialmente creada para dicho uso, una vez que se accede al sistema se muestra un menú que corresponde al menú usado por la sección de contabilidad dentro del cual aparece el ítem "Sistema de Inventario".



Figura 3. Menú principal Contabilidad.

Una vez que se entra al ítem “Sistema de Inventario” se desplegarán por pantalla 4 menús principales:

- Mantención de archivos
- Entrada de datos
- Consultas por pantalla
- Emisión de informes



Figura 4. Menú principal de Inventario.

En el ítem de **Mantención de archivos** se muestran las 4 subdivisiones a las que se puede acceder: Mantención de Establecimientos, Mantención de cuentas y sub.-cuentas, Mantención de archivos de Recintos, Mantención de archivos de Unidades.



Figura 5. Mantención de Archivos.

En el ítem de Mantención de Archivo de Establecimientos se tiene 4 acciones que se pueden realizar: Ingreso, Modificación, Eliminación.

Esta mismas cuatro acciones descritas anteriormente se pueden realizar con los otros ítems: Mantención de Cuentas y Subcuentas, Mantención de archivos de Recintos, Mantención de archivos de Unidades.

En **Entrada de datos** hay 3 submenús que son:

Altas de inventario de bienes, Altas de inventarios de vehículos y Traspaso de bienes entre establecimientos.

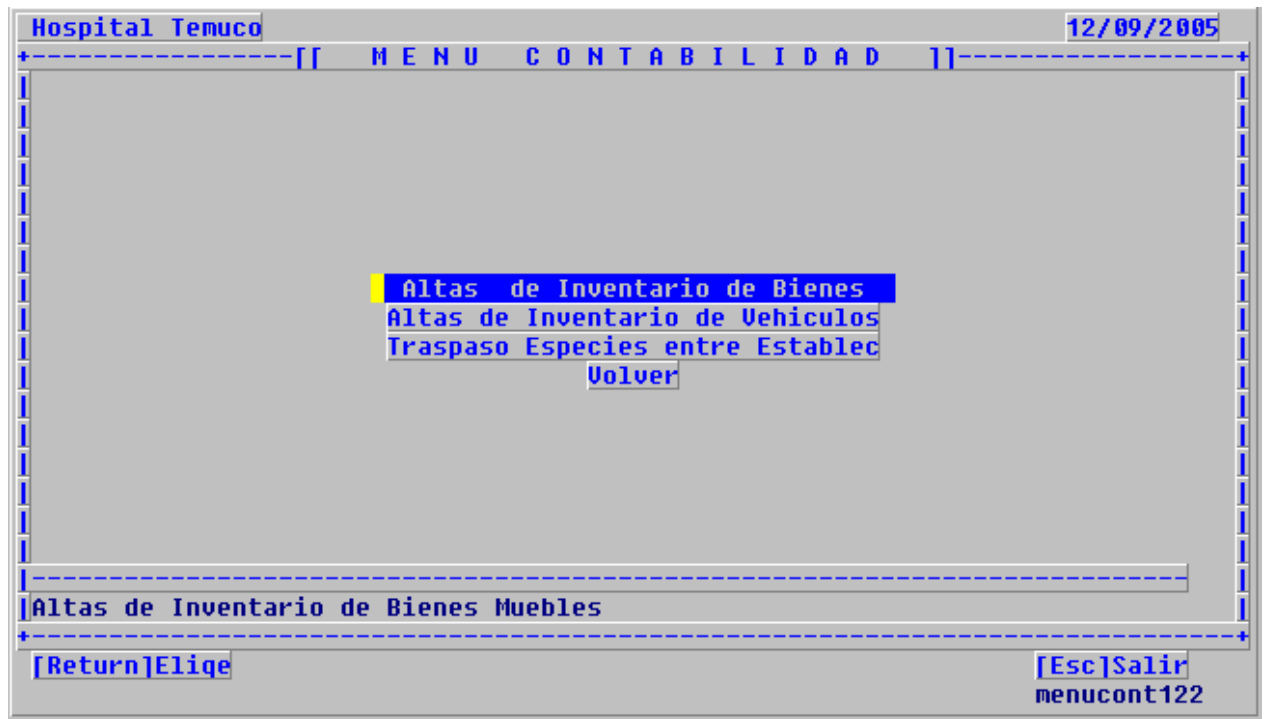


Figura 6. Entrada de Datos.

También en cada una de las opciones mencionadas se puede hacer 3 opciones:

- Ingresar
- Modificar.
- Eliminar

En **Consultas por pantalla** se puede hacer las consultas correspondientes a las distintas funciones que presta el inventario. Este menú presenta los siguientes ítemes:

- Consultas establecimientos.
- Consultas cuentas, subcuentas etc.
- Consulta recintos por nombre.
- Consulta hoja mural por recinto.
- Consulta de tabla unidades.
- Listado de inventario por cuenta.

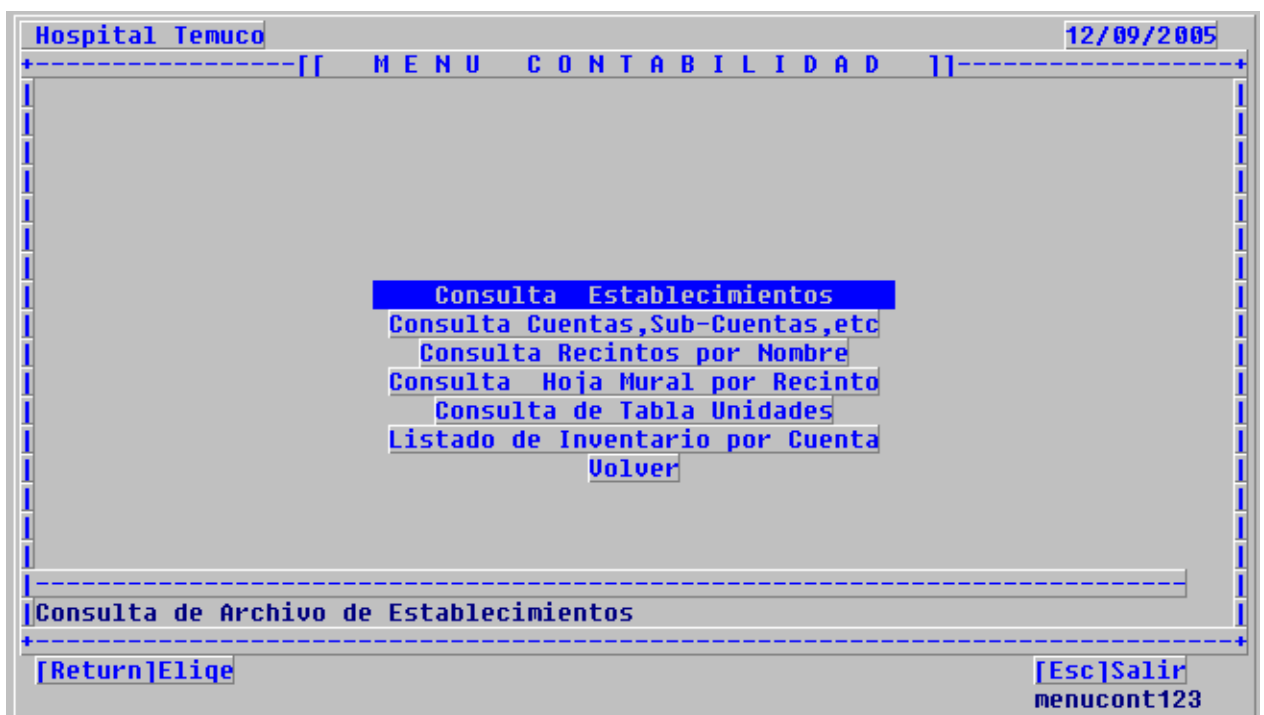


Figura 7. Consultas por pantallas.

En el ítem *Consulta Establecimientos*, se puede hacer una búsqueda por código y saber a que establecimiento corresponde, lo mismo sucede con las cuentas y subcuentas en donde se puede ver la clasificación de los bienes. En cuentas y subcuentas se usa una codificación especial para identificar los distintos tipos

de bienes y sus subcategorías por ejemplo:

Tabla N° 1
Clasificación de Bienes (fragmento)

Grupo	Cuenta	Subcuenta	Denominación
44101	0	0	EDIFICACIONES
44101	1	0	EDIFICACIONES PARA OFICINAS
44101	2	0	EDIFICACIONES PARA VIVIENDAS
44101	3	0	EDIF. ESTAB. MILITARES
44101	4	0	EDIF. ESTAB. EDUCACIONALES
44101	5	0	EDIF. ESTAB. ASISTENCIALES
44101	6	0	OTRAS EDIFICACIONES
44106	0	0	MAQUINARIAS
44106	6	0	MAQUINARIAS INDUSTRIALES
44106	6	1	PRENSAS
44106	6	2	FRESADORAS
44106	6	3	TELARES
44106	6	4	TORNOS
44106	6	5	TALADROS
44106	6	6	SOLDADORAS
44106	6	7	COMPRESOR (DENTAL)
44106	6	8	SIERRAS CIRCULARES

Fuente: Oficina Contabilidad.

Estas consultas sirven solo para verlas *on-line* pues no se imprime ni se emite ningún informe de estas.

En **Emisión de Informes** se pueden generar todo tipo de informes relacionados con las entidades del inventario, como establecimientos, unidades, recintos, bienes, etc.

Los movimientos o traslados de especies de un servicio a otro se hacen por intermedio de un formulario de traspaso el cual es autorizado por la oficina de inventario. Este proceso es solicitado por el jefe de servicio y una vez autorizado es firmado por inventario, el que entrega el bien y el que lo recibe.

Un bien puede tener distintos motivos de bajas los cuales son:

- No hay repuesto.
- Terminó de vida útil
- Modernización por equipo nuevo.

La baja se hace por intermedio del técnico quien emite un informe del estado del equipo el cual hace llegar una copia al jefe de servicio, a mantenimiento y a la oficina de inventario para su baja, al tramitar todos estos pasos la oficina de mantenimiento hace llegar a la bodega de rezago estos equipos. Los bienes de baja se entregan en donación para remate a distintos establecimientos hospitalarios.

Los bienes mobiliarios se hacen entrega en donación a la institución Unacos Emaus, ubicada en la población Las Quilas, autorizado por inventario, previa visita del encargado de esta corporación para que apruebe la entrega de los artículos en mal estado.

3.2 ANÁLISIS

Se hará una descripción más detallada del flujo de trabajo de inventario. Con esto se podrán avistar varias soluciones que apunten a construir un sistema que maneje todos estos datos de forma optima.

Lo primero en tener en cuenta es que el sistema de inventario nuevo debe cumplir como mínimo las mismas funciones que el sistema antiguo para poder cubrir las necesidades que se desprenden del trabajo diario de la persona de inventario. Con esto se asegura una continuidad un poco más suave al mantener la compatibilidad con el sistema anterior puesto que el diseño anterior permitía un manejo ordenado y estructurado de los bienes con funciones bien definidas en cuanto al trabajo y los pasos del flujo de trabajo de inventario.

Como se planteó en los requerimientos, se tienen 2 referencias importantes para el diseño del Sistema de Inventario, uno basado en la descripción del flujo de trabajo descrito por el propio encargado de esta tarea y la otra basada en el sistema antiguo de inventario.

3.2.1 REQUERIMIENTOS SISTEMA ANTERIOR

Cuando se describió el sistema antiguo de inventario se mostró todo el alcance de este en cuanto a las acciones que se podían realizar con los bienes, por esto es muy importante cumplir con las mismas acciones que se realizaban y los

mismos campos de la base de datos para conservar una compatibilidad que también será útil para el encargado de inventario.

3.2.2 REQUERIMIENTOS NUEVO SISTEMA

Esto tiene como idea principal la de incorporar una nueva forma de etiquetar los bienes basado en el código de barra, esto permitirá mas adelante incorporar otras tecnologías como una palm con lector de código de barra lo que permitirá hacer consultas a la base de datos estando en terreno.



Figura 9. Código de barra.

Se deberá separar las tareas usuales que se realizan en un sistema de inventario con las tareas que se usen para administración del sistema. A continuación se describe El Diagrama de flujo del Inventario.

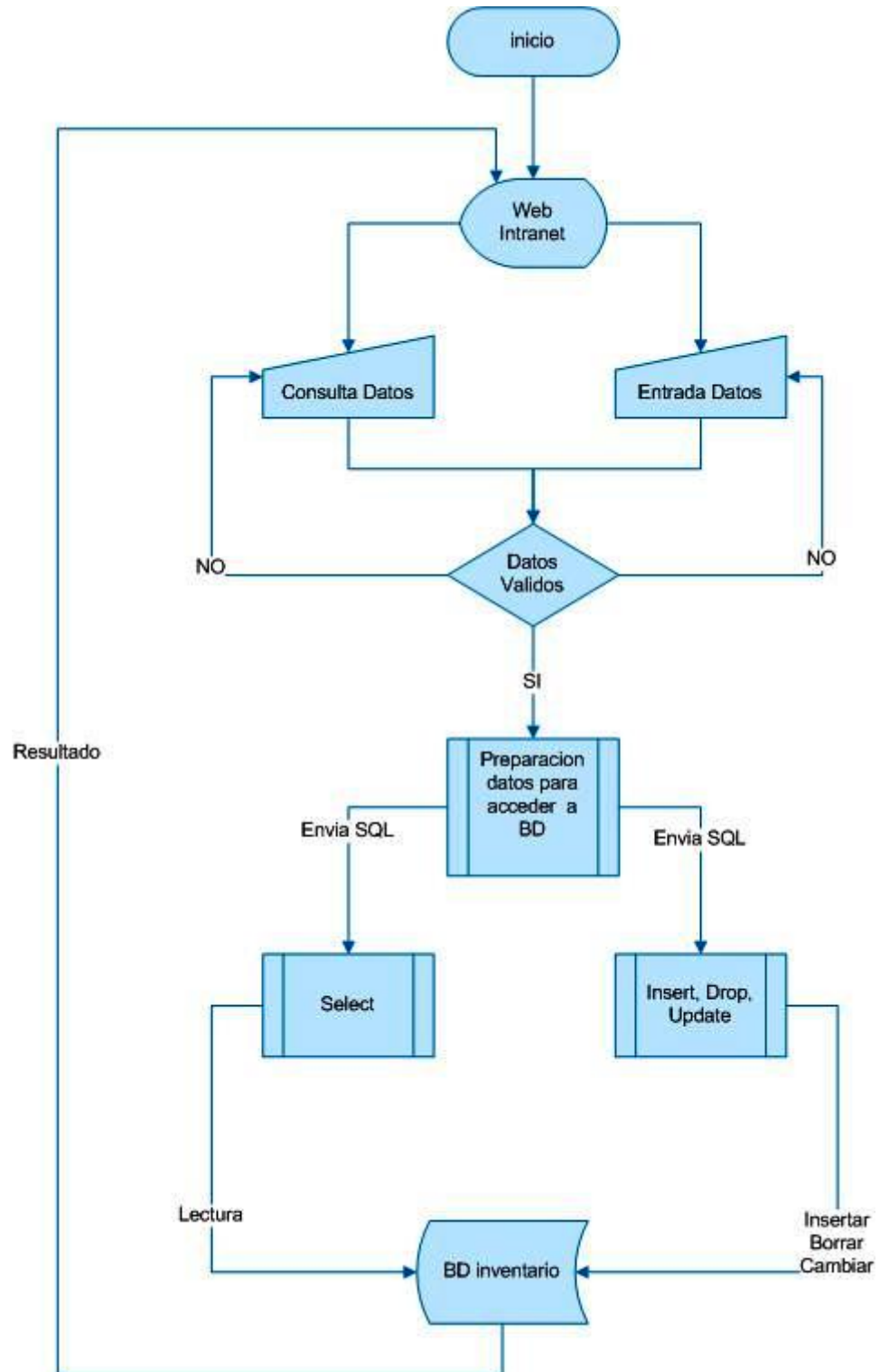


Figura 10. Diagrama de Flujo Inventario.

3.2.2.1 Ingreso de Bienes

Se necesitará un módulo, el cual corresponderá al ingreso de datos, que servirá como entrada al sistema a través de un formulario que deberá contener los datos más relevantes de los bienes como nombre, descripción, marca, medidas, etc. Para hacer este ingreso se debe validar algunos campos importantes los cuales pueden producir problemas en la base de datos o en el módulo que procese la entrada. Algunos datos a validar serían: la fecha en formato (dd-mm-aaaa), números (de inventario), campos de texto que no se ingresen números y en campos de números no ingresen textos o símbolos, etc.

Este módulo deberá contar además con otros aspectos como la presentación, disposición de los controles del usuario en la pantalla, colores, conexión a la base de datos, etc., lo cual se definirá en el apartado de diseño.

3.2.2.2 Corrección de datos

Otro módulo importante es el poder corregir los datos ingresados, esto se refiere a modificar algunos datos y en otros casos eliminar el registro completo si es un error muy grave, esto se deberá indicar explícitamente con algún lugar en el registro por pantalla lo cual da mayor opción de hacer un manejo de datos más rápido.

3.2.2.3 Consulta de bienes

Se trata de un módulo en el que se pueden filtrar información haciendo distintas consultas a la base de datos ya sea de una tabla o varias tablas. Con esto se podrán mostrar los datos requeridos por la persona que hace la consulta. Para que sea lo más completo posible se deberá incluir los campos más importantes por los que se desea agrupar los bienes, también se debe poder elegir por parte del usuario que columnas desea que se muestren el resultado de la consulta.

3.2.2.4 Hojas murales

Un módulo sumamente importante dentro del sistema es la de poder sacar reportes de los datos almacenados, lo que en este caso correspondería a las hojas murales, las cuales tienen la función de mostrar los bienes agrupados dentro de la estructura interna del hospital, o sea en servicios y dependencias, gracias a esta hoja el encargado de inventario podrá distribuir las a través de todas la oficinas mostrando a los encargados de cada lugar cuales son los bienes que corresponden a ese lugar , además de mostrar algunas características principales del bien (nombre, descripción, numero de inventario, etc.). Estas hojas se deben ir actualizando a medida que cambien los datos en el sistema, esto es que se produzcan nuevas altas o nuevas bajas, o movimientos de bienes.

3.2.2.5 Movimiento de bienes

Este módulo se usará en el caso de querer hacer algún traspaso o movimiento de uno o más bienes. Esto se puede producir por muchas razones pero principalmente es por la necesidad de contar con algún bien que es más necesario en un lugar que otro, o por cambio de una oficina de lugar físico, etc.

3.2.2.6 Baja de bienes

Como se explicaba en el ítem anterior, las bajas se producen por 3 causas:

- No hay repuesto.
- Terminó de vida útil
- Modernización por equipo nuevo.

Entonces lo que se debería hacer según el flujo de trabajo de inventario sería mover el bien de dependencia a una llamada bodega de rezago, lo cual deberá hacer este módulo.

3.2.2.7 Ver movimiento de bienes

Se necesita un módulo que permita visualizar todos los movimientos realizados por las personas que manejan el sistema, también es conveniente que se puedan filtrar estos movimientos ya sea por fecha, por servicio, por dependencia, etc.

3.2.2.8 Ver bajas de bienes

Así como es necesario ver el movimiento de los bienes también se necesita ver las bajas realizadas por el encargado de inventario, así se pueden sacar estadísticas, o la información que se estime necesaria que pueda indicar que bienes son mas predispuestos a tener bajas, de que servicios, etc.

3.2.2.9 Búsqueda de Servicios

Se necesita también un módulo donde el usuario pueda encontrar un servicio ya sea buscándolo por el nombre de este o si tiene el código también es posible que encuentre el nombre si se necesita para alguna operación.

3.2.2.10 Búsqueda de Dependencias

Lo mismo que sucede con los servicios es igualmente valido para las dependencias.

3.2.2.11 Servicios y Dependencias

Este módulo servirá para encontrar todos los servicios y sus dependencias asociadas. Por ejemplo, si se selecciona el Departamento de Informática, deberá mostrar sus 4 dependencias asociadas, en este caso corresponden a "Sala Computación", "Bodega", "Sala Servidores" y "Jefe Informática"

3.2.2.12 Administración

Este enlace lleva a una sección de labores Administrativos, la cual contiene varios submódulos que se describen a continuación:

3.2.2.13 Agregar dependencias

En este módulo se crearán las nuevas dependencias que se generen para un servicio determinado. Para esto se deberá asignar un código de dependencia, una descripción, RUT del encargado, código de servicio a cual pertenecerá.

3.2.2.14 Agregar servicios

En este ítem se agregarán aquellos servicios que se creen dentro de la estructura administrativa del hospital. Estos es menos frecuente que en el caso de las dependencias. En este caso también se deberá ingresar un código de servicio, una descripción del servicio y el RUT del encargado.

3.2.2.15 Agregar Usuarios

En éste módulo se crean a los usuarios que podrán acceder al sistema de inventario, guardando de ellos su nombre, apellidos, RUT, cargo y password de acceso.

3.2.2.16 Modificar Dependencias

Este módulo será usado para editar y modificar los valores de las dependencias en el caso que sea necesario.

3.2.2.17 Modificar Servicios

En éste módulo se pueden editar y modificar sus valores en el caso que sea necesario.

3.2.2.18 Modificar Usuarios

En éste módulo se pueden editar y modificar los datos de los usuarios en caso que hallan sido ingresados en forma incorrecta o que después de un tiempo hallan cambiado.

3.2.2.19 Eliminar Dependencias

Este módulo se usa para eliminar una dependencia que ya no existe o que halla sido ingresada en forma incorrecta.

3.2.2.20 Eliminar Servicios

Este módulo elimina los servicios que ya no existan y/o que hallan sido

ingresados en forma incorrecta.

3.2.2.21 *Eliminar Usuarios*

En este módulo se podrán eliminar los usuarios que ya no tengan acceso al sistema, por lo que serán borrados de la base de datos de usuario.

3.2.2.22 *Limpiar archivos*

En este módulo se podrán visualizar los archivos generados en el momento de generar las hojas murales, éstas quedarán en el servidor y una copia es enviada al cliente que la solicite. Luego de visualizarlo se podrán eliminar todos los archivos que tengan extensión RTF.

3.3 DISEÑO

Luego de haber visto todos los detalles del sistema anterior en cuanto a los procesos y las etapas que se iban cumpliendo de acuerdo a las necesidades de los bienes como mover o dar de baja, también se ha visto el flujo de trabajo llevado a cabo en contabilidad por el encargado de inventario al hacer todo este proceso incluyendo correcciones de errores que van surgiendo en el camino tanto administrativos como técnicos, se llega al siguiente diseño del nuevo sistema de inventario.

La parte diseño que no esta involucrada con la funcionalidad del sistema será encargada a una persona externa, la cual será contratada para diseñar la imagen corporativa, los banners, etc., de este y otros proyectos que se desarrollan en paralelo al sistema de inventario, enfocadas a distintas áreas del hospital, por lo que hasta el momento el trabajo esta centrado en la parte funcional y los procesos informáticos que se desarrollan con cada parte del sistema.

Se deberá crear una interfaz principal la cual invoque a todas las opciones necesarias para el correcto funcionamiento del sistema. Este llevara un menú al que se deberá acceder previo a obtener un nombre de usuario y una contraseña, la cual deberá ser proporcionada por el administrador del sistema que en este caso puede ser una persona de la sección de computación o bien el mismo encargado de inventario una vez adquiridos los conocimientos necesarios para el cumplimiento de esta tarea.

Ingresar	Consultas	Reportes	Movimientos	Bajas	Mantenición
bienes	Bienes	H. murales	Bienes	Bienes	Servicios
Sevicios	Servicios		Ver	Ver	Dependencias
Dependencias	Dependencias				Usuarios
Uuarios	Serv. y Dep.				Archivos temporales

Figura 11. Diseño Menú Principal

Existen dos grupos de módulos: los de uso cotidiano y los que cumplen tareas

administrativas. El uso cotidiano es donde puede consultar el sistema, puede sacar hojas murales, hacer movimientos de bienes, etc., las tareas propias de una persona que trabaja con los bienes, en el caso que puedan haber mas de una persona haciendo este trabajo, también existen las tareas administrativas del sistema que es quien esta encargado de gestionar los recursos del sistema como por ejemplo crear usuarios, crear nuevas dependencias, nuevos servicios, modificar datos, etc.

Por lo tanto el usuario verá el primer menú al cual tendrá acceso a todas sus opciones y abra un menú especial para administrar el cual maneja otra clave para este acceso. Las opciones disponibles para el trabajo diario son las siguientes:

- ***Ingreso de Bienes***
- ***Consultas Bienes***
- ***Hojas Murales***
- ***Movimiento Bienes***
- ***Baja Bienes***
- ***Ver Movimiento Bienes***
- ***Ver Baja Bienes***
- ***Búsqueda de Servicios***
- ***Búsqueda de Dependencias***

➤ ***Servicios y Dependencias***

➤ ***Administración***

Las opciones disponibles para administrar son las siguientes:

➤ ***Agregar Dependencias***

➤ ***Agregar Servicios***

➤ ***Agregar usuarios***

➤ ***Modificar Dependencias***

➤ ***Modificar Servicios***

➤ ***Modificar usuarios***

➤ ***Eliminar Dependencias***

➤ ***Eliminar Servicios***

➤ ***Eliminar usuarios***

➤ ***Limpiar archivos***

3.3.1 DESCRIPCIÓN DE CAMPOS

También se establecerán que campos son los que se usaran como entradas, las cuales estarán directamente relacionadas con los campos de la base de datos, estos campos serán usados en todas las interfaces que vera el usuario para hacer ingreso o consulta de la información del inventario, obviamente en algunas habrán más campos presentes que en otras interfaces, pero cuando se refiera a un campo en un módulo y luego en otro módulo se entenderá que se refiere al mismo objeto.

3.3.1.1 Código Barra

En esta opción se buscará el bien por su nuevo código computacional de inventario, el cual cuenta de un número 2, un guión y un número de hasta 5 dígitos. Si no se especifica otra opción buscará dentro de toda la base de datos.

3.3.1.2 Código Antiguo

Con esta opción se buscan los bienes por su código antiguo, el cual no todos los bienes lo poseen en forma visible. Si no se especifica otras opciones buscará en toda la base de datos.

3.3.1.3 Nombre

En esta opción se buscarán todos los bienes por su nombre especificado o que esté contenido dentro de algún otro nombre. Si no se especifica otras opciones buscará en toda la base de datos.

3.3.1.4 Descripción

En esta opción busca bienes en los cuales se tenga una vaga referencia por lo que entregará muchos datos inexactos. Luego se podrá ir filtrando a medida que te tengan más referencias. Si no se especifica otras opciones buscará en toda la base de datos.

3.3.1.5 Marca

En esta opción se podrá buscar un bien por la marca comercial específica, por ejemplo si se quiere saber cuales son los computadores marca IBM de todo el hospital, se debe especificar en el campo “nombre” la palabra computador, y en el campo “marca” la palabra IBM. Si no se especifica otras opciones buscará en toda la base de datos.

3.3.1.6 Modelo

Se buscan bienes por un modelo específico, lo cual también puede ir acompañado de otras opciones si se estima necesario. Por ejemplo todas las impresoras modelo IP1000 de Cannon. Si no se especifica otras opciones buscará en toda la base de datos.

3.3.1.7 Serie

Este campo se usa para buscar equipos en forma muy específica o sea usando su número de serie.

3.3.1.8 Estado

En este ítem se pueden especificar consultas basadas en el estado físico del bien, que en este caso serían: bueno-reparable- malo. Por ejemplo: todas las sillas que estén en buen estado. Si no se especifica otras opciones buscará en toda la base de datos.

3.3.1.9 Propiedad

En esta consulta se puede especificar la propiedad del bien o sea a quien pertenece. En este ítem se reconocen 4 opciones hospital-comodato-personal-otro. El término *hospital* se refiere a aquellos bienes que han sido comprados

por el mismo hospital para su uso, ya sea mobiliario, equipos, insumos etc. El término *comodato* se refiere a aquellos bienes que están en calidad de préstamo por otra institución. El término *personal* se refiere a aquellos bienes que son propios de personas que son funcionarios del hospital. El término *otro* se atañe a aquellos bienes que no encasillen en algunas de las clasificaciones anteriores. Por ejemplo todos los hervidores que corresponden a los funcionarios.

3.3.1.10 Servicio

La búsqueda se acota al servicio especificado lo cual no excluye otro tipo de filtro. Para esto se debe utilizar el código de servicio, por ejemplo todos los PC del servicio 3520 (CDT). Si no se especifica otras opciones buscará en toda la base de datos.

3.3.1.11 Dependencia

La búsqueda se acota aún más utilizando este campo (se refiere a una oficina en particular). En esta búsqueda se deben utilizar los códigos de la dependencia. Por ejemplo, las mesas de la dependencia 1941 (Sala de Computación). Si no se especifica otras opciones buscará en toda la base de datos.

3.3.1.12 Grupo- Cuenta- Subcuenta

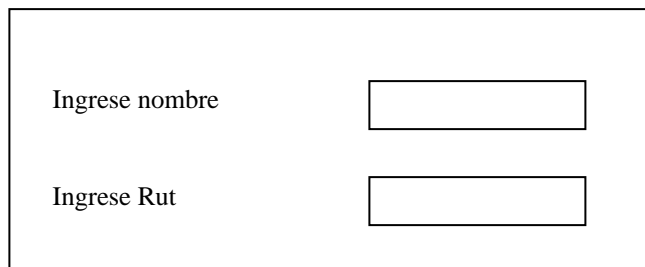
Estas opciones se están trabajando para poder estandarizar los códigos que se usarán, uniendo los códigos de contabilidad con los códigos del MINSAL por los cual esto todavía no se ha habilitado.

3.3.2 DISEÑO DE PANTALLAS

Las pantallas que se usaran en el sistema son importantes ya que es la interfaz que comunicará al usuario, por esto debe ser intuitiva y amigable al usuario. En el diseño se contemplan todas las ventanas que dispondrá el sistema para el trabajo con datos.

3.3.2.1 Acceso al sistema

El ingreso al sistema a cualquiera de sus módulos estará restringido a usuario con claves de ingreso.



Ingrese nombre	<input type="text"/>
Ingrese Rut	<input type="text"/>

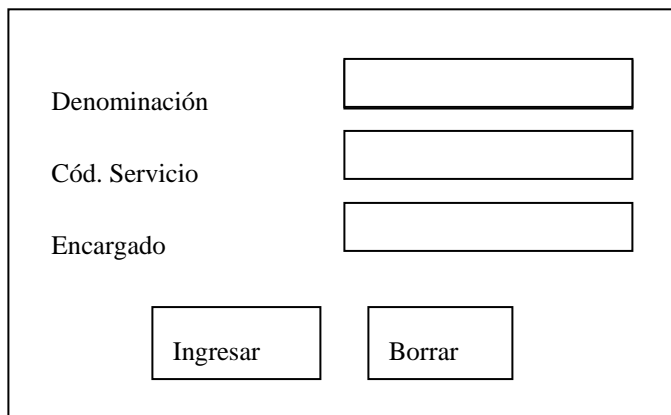
Figura 12. Diseño Ingreso al Sistema.

Para ingresar al sistema se necesita estar registrado en él para que los campos

sean válidos.

3.3.2.2 Ingreso dependencias

En el caso del ingreso de nuevas dependencias se tiene el siguiente diseño:



Formulario de ingreso de dependencias con los siguientes campos:

- Denominación
- Cód. Servicio
- Encargado

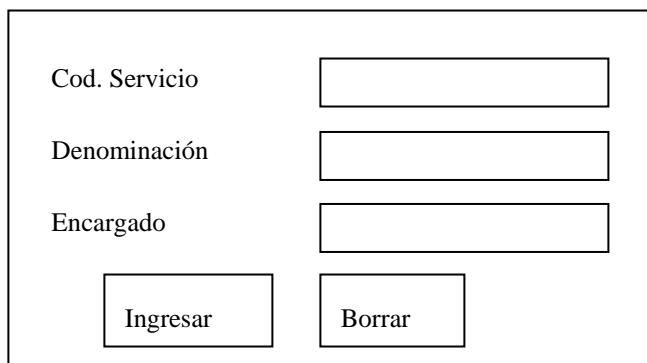
Botones de acción:

- Ingresar
- Borrar

Figura 13. Diseño Ingreso Dependencia.

3.3.2.3 Ingreso servicios

Para el ingreso de servicios se presentara una entrada análoga con los datos correspondientes al nuevo servicio.



Formulario de ingreso de servicios con los siguientes campos:

- Cod. Servicio
- Denominación
- Encargado

Botones de acción:

- Ingresar
- Borrar

Figura 14. Diseño Ingreso Servicio.

3.3.2.4 Ingreso usuarios

En el caso de ingresar nuevos usuarios al sistema se presenta el siguiente diseño siguiendo el esquema general para el ingreso de datos:

Rut	<input type="text"/>
Nombres	<input type="text"/>
Apellidos	<input type="text"/>
Cargo	<input type="text"/>
Password	<input type="text"/>
<input type="button" value="Ingresar"/>	<input type="button" value="Borrar"/>

Figura 15. Diseño Ingreso Usuario

Estos diseños son simples y constituyen una buena manera de que el usuario no se distraiga en otros temas que le podrían ocasionar distorsión de su quehacer, presentando mucha información que para el usuario no es necesaria.

También se ha diseñado una interfaz similar para la tarea de ingresar, modificar, eliminar datos, excepto en el caso de ingreso y consulta de bienes.

3.3.2.5 Ingreso bienes

Una vez que se accede al sistema se procederá a ingresar los datos de un nuevo bien que fue inventariado para ese efecto, se debe tener un formulario que cuente con todos los detalles que se quieren registrar del bien en la base de datos. Se aprecia el siguiente diseño.

Estos campos registrarán los datos más relevantes del bien, los que serán almacenados en la base de datos.

Para ingresar al sistema, el usuario deberá escoger de un Listbox el servicio (se mostrará el nombre y su código) y una vez elegido el servicio se mostrarán sus dependencias disponibles donde también se mostrarán su nombre y su código. Con esta información podrá acceder al formulario donde se ingrese los datos.

<i>Número Antiquo Inventario</i>	<input type="text"/>	<i>Marca</i>	<input type="text"/>	<i>Grupo</i>	<input type="text"/>
<i>Nombre Especie</i>	<input type="text"/>	<i>Modelo</i>	<input type="text"/>	<i>Cuenta</i>	<input type="text"/>
<i>Descripción</i>	<input type="text"/>	<i>Serie</i>	<input type="text"/>	<i>Sub Cuenta</i>	<input type="text"/>
<i>Estado</i>	<input type="text"/>	<i>Medida</i>	<input type="text"/>	<i>Proveedor</i>	<input type="text"/>
<i>Propiedad</i>	<input type="text"/>	<i>Vida Útil</i>	<input type="text"/>	<i>Servicio Técnico</i>	<input type="text"/>
<i>Cantidad</i>	<input type="text"/>	<i>Vida Esperada</i>	<input type="text"/>	<i>Año Instalación</i>	<input type="text"/>
<i>Número Inventario</i>	<input type="text"/>	<i>Año Fabricación</i>	<input type="text"/>	<i>Tipo Vehículo</i>	<input type="text"/>
<i>Chasis</i>	<input type="text"/>	<i>Patente</i>	<input type="text"/>	<i>Tipo Carrocería</i>	<input type="text"/>
<i>Valor \$</i>	<input type="text"/>	<i>Número Alta</i>	<input type="text"/>		

Figura 16. Diseño Ingreso bienes.

3.3.2.6 Vista de Ingreso

A medida que el usuario vaya ingresando bienes sobre el formulario de ingreso se irá desplegando la información de los bienes recientemente ingresado en el caso que el ingreso sea exitoso. En el caso que ocurra algún error, por ejemplo un código de barra se encuentre ingresado, el bien no aparecerá en la lista de bienes recientemente ingresado, a cambio de esto se desplegará en la pantalla del PC un mensaje de error.

<i>Acciones</i>	<i>Número Inventario Antigo</i>	<i>Nombre Especie</i>	<i>Descripción</i>	<i>Característica</i>				<i>Estado</i>	<i>Propiedad</i>	<i>Cantidad</i>	<i>Número Inventario</i>	<i>Grupo</i>	<i>Cuenta</i>	<i>Sub Cuenta</i>	<i>Chasis</i>
				<i>Marca</i>	<i>Modelo</i>	<i>Serie</i>	<i>Medida</i>								

Figura 17. Diseño Vista de Ingreso

También en la tabla desplegada con los bienes ingresados se mostrarán 2 opciones, una para borrar el registro creado, esto en caso que halla sido un error en el ingreso del código de barra y la otra opción es la de modificación con lo que se pueden modificar todos los valores del bien excepto el código de barra el cual si se quiere modificar se deberá eliminar el registro.

3.3.2.7 Consultas

Para las consultas de los bienes se debe organizar las opciones de búsqueda así como las acciones de columnas que se quiera mostrar. En este caso las opciones de búsqueda son análogas a la de vistas de columnas.

Codigo Barra		Estado	
Codigo Antiguo		Propiedad	
Marca (empieza con)		Nombre (empieza con)	
Modelo (empieza con)		Descripcion (empieza con)	
Serie (empieza con)		Grupo (codigo)	
Servicio (codigo)		Cuenta (codigo)	
Dependencia (codigo)		Subcuenta (codigo)	

Figura 18. Diseño Consulta Bienes

En esta tabla, el usuario especifica cuales son los datos que desea ver, los cuales son optativos. El sistema mostrará por defecto 3 opciones marcadas las cuales el usuario podrá modificar según estime conveniente. Las opciones disponibles son:

- Código Barra
- Código Antiguo
- Nombre
- Descripción
- Marca
- Modelo
- Serie
- Medida
- Estado
- Propiedad

- Cantidad
- Servicio
- Dependencia
- Nominal
- Código grupo
- Código cuenta
- Código Subcuenta

También se incluirán 2 botones para facilitar el trabajo al usuario: uno con el que podrá seleccionar todas las opciones y el otro donde deselecciona las opciones elegidas por el usuario y deja las opciones por defecto del sistema.

3.3.2.8 Hojas murales

Se creara un módulo en el cual se podrán emitir reportes por dependencia o por servicios de los bienes estos serán generados en formato RTF

Servicio Salud Araucanía
Departamento de Finanzas

Fecha

INVENTARIO

Establecimiento:

Servicio:

Dependencia:

Código:

Código:

Encargado Inventario

Jefe Servicio

N° Inv.	Cant.	Nombre	Descripción	Estado	Propiedad

Figura 19. Diseño Hojas Murales.

3.3.2.9 Traspaso bienes, Baja Bienes

Se creara un módulo en donde se podrán efectuar los traspasos de bienes entre servicios y dependencias donde solo se deberá ingresar el código del bien a mover y sus códigos de destino. Este diseño también se aplica a la Baja de Bienes el cual usa los mismos parámetros sólo con la diferencia que el destino es siempre la Bodega de Rezago.

Cod. barra	<input type="text"/>	<input type="button" value="Buscar"/>	<input type="button" value="Borrar"/>
Traspaso N°	<input type="text"/>		
Fecha Actual	<input type="text"/>		
ACTUALMENTE			
Codigo barra	<input type="text"/>	<input type="text"/>	
Estado	<input type="text"/>		
Establecimiento	<input type="text"/>		
Servicio	<input type="text"/>	<input type="text"/>	
Dependencia	<input type="text"/>	<input type="text"/>	
DESTINO			
Establecimiento	<input type="text"/>		
Servicio	<input type="text"/>		
Dependencia	<input type="text"/>		
	<input type="button" value="Traspasar"/>		

Figura 20. Diseño Traspaso Bienes.

3.3.2.10 Ver Movimiento, Bajas

Esta interfaz, muestra los movimientos realizados por los bienes entre dependencias y servicios. Este diseño, es homólogo para ver bajas.

Busqueda por		Ordenado por
Num Traspaso	<input type="text"/>	<input type="radio"/>
Fecha Movimiento	Año <input type="text"/> Mes <input type="text"/> Dia <input type="text"/>	<input type="radio"/>
Codigo barra	<input type="text"/>	<input type="radio"/>
Estado	<input type="text"/>	<input type="radio"/>
Establecimiento (codigo)	<input type="text"/>	<input type="radio"/>
Servicio (codigo)	<input type="text"/>	<input type="radio"/>
Dependencia (codigo)	<input type="text"/>	<input type="radio"/>

Figura 21. Diseño Ver Movimientos Bienes

3.3.2.11 Búsqueda Servicios, Dependencias.

Se creará un módulo en donde el usuario pueda efectuar búsquedas de servicios ya sea por su código, por su denominación o por el Rut del encargado. Este diseño también se aplica a la búsqueda de dependencias.

Busqueda por		Ordenado por
Código de Servicio	<input type="text"/>	<input type="radio"/>
Nombre del Servicio	<input type="text"/>	<input type="radio"/>
RUT del Encargado	<input type="text"/>	<input type="radio"/>

Figura 22. Diseño Búsqueda Servicios

3.3.2.12 Servicios y Dependencias

En éste módulo, se muestra la interfaz donde se podrán encontrar las listas de los servicios y sus respectivas dependencias todas con sus códigos. Para esto se usarán listas desplegables.

Servicio	<input type="text"/>	<input type="text"/>
	<input type="text"/>	<input type="text"/>
Dependencia	<input type="text"/>	<input type="text"/>
	<input type="text"/>	<input type="text"/>

Figura 23. Diseño Servicios y Dependencias

3.3.2.13 Modifica Dependencia, Servicio

En esta interfaz, se provee los elementos necesarios para poder modificar una dependencia en cualquiera de sus campos. Este mismo diseño se aplica también para modificar un servicio.

Modificar dependencia

Característica de la Dependencia

Código

Descripción

Servicio al que pertenece

RUT de encargado

Fecha de Creación

Figura 24. Diseño Modifica Dependencia

3.3.2.14 Modifica Usuario

Para el módulo que modifica los datos del usuario se creó la interfaz donde ingresando el Rut se obtienen todos los datos del usuario, aquí se pueden cambiar sus nombres, Rut, claves de acceso y cargo.

Ingreso Rut

Características del Usuario

RUT

Nombre

Apellido

Cambio de password

Cargo

Fecha de creación

Figura 25. Diseño Modifica Usuario.

3.3.2.15 Eliminación de Dependencia, Servicio.

En ese diseño, ingresando el código de la dependencia se muestran sus datos si es que existe, luego dando paso a su posible eliminación. Esto es análogo para eliminar servicio. Hay que destacar que si los servicios o dependencias contienen bienes asociados no se permitirá su eliminación.

Dependencia

Característica de la Dependencia

Código

Descripción

Servicio al que pertenece

RUT de encargado

Fecha de Creación

Figura 26. Diseño Elimina dependencia.

3.3.2.16 Eliminar Usuario

Este diseño es análogo al Modificar usuario con la diferencia que en vez de ingresarlo, el usuario se escoge desde una lista despegable mostrada por pantalla, una vez visualizado los datos de éste, se puede eliminar del sistema.

Usuario a eliminar

Mostrar

Características del Usuario

RUT

Nombre

Apellido

Cambio de password

Cargo

Fecha de creación

Eliminar

Figura 27. Diseño Elimina Usuario

3.3.3 DISEÑO DE LA BASE DE DATOS

Toda la manipulación y el ingreso de datos deben tener un repositorio donde se almacene de forma clara y sin distorsiones que causen inconsistencia de datos por lo que un buen diseño de la base de datos es fundamental. En este caso se usara la base de datos Open Source MySQL, por motivos de estabilidad y costo.

También se tomo en cuenta las tablas de la base de datos Cobol en la que se guardaban los datos de la base de datos antigua.

TABLA N° 2

Esquema de registro de muebles

mueble	
establecimiento:	INTEGER
numero inventario:	INTEGER
dependencia:	INTEGER
cuenta:	INTEGER
numero alta:	INTEGER
fecha alta:	DATE
unidad:	INTEGER
nombre:	INTEGER
marca:	INTEGER
caracteristica:	INTEGER
estado:	INTEGER
vida util:	INTEGER
vida esperada:	INTEGER
valor:	INTEGER
observacion:	INTEGER
n-equipos dental:	INTEGER
cantidad:	INTEGER
numero dae:	INTEGER
fecha traspaso:	INTEGER
numero traspaso:	INTEGER
unidad traspaso:	INTEGER
dependencia traspaso:	INTEGER

Fuente: Sección Computación

TABLA N° 3

Esquema de registro vehículos

vehículo
◆ fecha alta: DATE
◆ unidad: INTEGER
◆ tipo vehiculo: INTEGER
◆ marca vehiculo: INTEGER
◆ modelo vehiculo: INTEGER
◆ motor vehiculo: INTEGER
◆ chasis vehiculo: INTEGER
◆ año vehiculo: INTEGER
◆ tipo carroceria vehiculo: INTEGER
◆ patente vehiculo: INTEGER
◆ patente 1: INTEGER
◆ patente 2: INTEGER
◆ patente 3: INTEGER
◆ estado: INTEGER
◆ vida util: INTEGER
◆ vida esperada: INTEGER
◆ valor: INTEGER
◆ observacion: INTEGER
◆ n-equipo dental: INTEGER
◆ cantidad: INTEGER
◆ numero dae: INTEGER
◆ fecha traspaso: INTEGER
◆ numero traspaso: INTEGER
◆ unidad traspaso: INTEGER
◆ dependencia traspaso: INTEGER

Fuente: Sección Computación

Se incluirá al sistema actual los siguientes campos del sistema antiguo por compatibilidad y requerimientos del sistema anterior:

- Vida útil.
- Vida esperada.
- Fecha traspaso.
- Numero traspaso.
- Chasis vehiculo

- Tipo vehiculo
- Tipo carrocería
- Patente.

Los campos que se descartaron por su falta de uso son:

- N-Equipo Dental.
- Numero DAE.

3.3.3.1 Tablas

Se definirán las tablas que se utilizaran en la creación de la base de datos:

- *Bienes*
- *Usuarios*
- *Establecimientos*
- *Servicio*
- *Dependencia*
- *Encargados*
- *Movimientos*

3.3.3.1.1 Bienes

Esta es la tabla principal donde se almacena la información más importante sobre una instancia de un bien, guarda los siguientes valores:

TABLA N° 4

Estructura Tabla Bienes

bienes	
	codigo_barra: VARCHAR(20)
	codigo: VARCHAR(30)
	nombre: VARCHAR(45)
	descripcion: VARCHAR(45)
	marca: VARCHAR(45)
	modelo: VARCHAR(45)
	serie: VARCHAR(30)
	medida: VARCHAR(30)
	estado: VARCHAR(12)
	propiedad: VARCHAR(12)
	cantidad: INTEGER
	codigo_servicio: VARCHAR(6)
	codigo_dependencia: VARCHAR(6)
	clasificacion: VARCHAR(10)
	rut: VARCHAR(10)
	nominal: CHAR
	num_alta: VARCHAR(10)
	fecha_alta: DATE
	observacion: TEXT
	valor: INTEGER
	fecha_inventario: DATE
	ult_fecha_traspaso: DATE
	chasis: VARCHAR(25)
	tipo_vehiculo: INTEGER
	tipo_carroceria: VARCHAR(30)
	patente: VARCHAR(6)
	vida_util: INTEGER
	vida_esperada: INTEGER
	servicio_tecnico: VARCHAR(40)
	proveedor: VARCHAR(30)
	ano_fabricacion: INTEGER
	ano_instalacion: INTEGER
	establecimiento: VARCHAR(4)
	fecha_modificacion: DATE
	num_factura: VARCHAR(11)

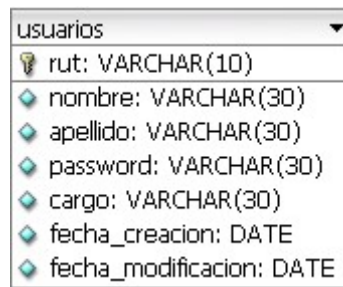
Fuente: Sistema Inventario.

3.3.3.1.2 Usuarios

La tabla usuarios contendrá los usuarios que pueden acceder a la aplicación sus nombres y password.

TABLA N° 5

Estructura Tabla usuarios



usuarios	
rut	VARCHAR(10)
nombre	VARCHAR(30)
apellido	VARCHAR(30)
password	VARCHAR(30)
cargo	VARCHAR(30)
fecha_creacion	DATE
fecha_modificacion	DATE

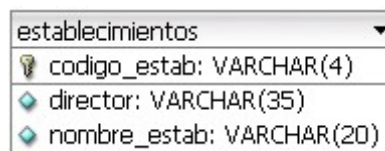
Fuente: Sistema Inventario.

3.3.3.1.3 Establecimientos

Se usara en el caso de traspasar bienes entre establecimientos como en el caso de las bajas:

TABLA N° 6

Estructura Tabla Establecimientos



establecimientos	
codigo_estab	VARCHAR(4)
director	VARCHAR(35)
nombre_estab	VARCHAR(20)

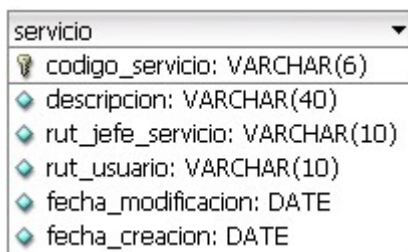
Fuente: Sistema Inventario.

3.3.3.1.4 Servicio

Es el campo con las características del servicio al que pertenecerá un determinado bien:

TABLA N° 7

Estructura Tabla Servicio



servicio	
codigo_servicio	VARCHAR(6)
descripcion	VARCHAR(40)
rut_jefe_servicio	VARCHAR(10)
rut_usuario	VARCHAR(10)
fecha_modificacion	DATE
fecha_creacion	DATE

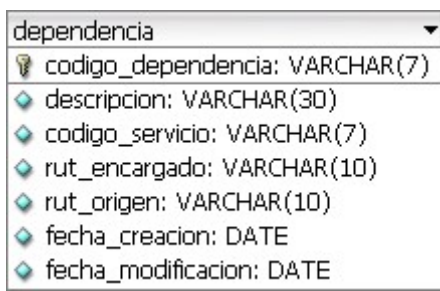
Fuente: Sistema Inventario.

3.3.3.1.5 Dependencia

En este campo se almacenan los datos más relevantes respecto a la dependencia a la cual pertenece el bien:

TABLA N° 8

Estructura Tabla Dependencia



dependencia	
codigo_dependencia	VARCHAR(7)
descripcion	VARCHAR(30)
codigo_servicio	VARCHAR(7)
rut_encargado	VARCHAR(10)
rut_origen	VARCHAR(10)
fecha_creacion	DATE
fecha_modificacion	DATE

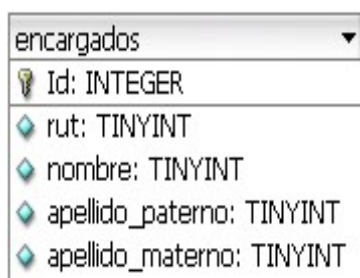
Fuente: Sistema Inventario.

3.3.3.1.6 Encargados

Mantiene la información relativa a los encargados ya sea dependencias o servicios con algunos datos relevantes para el sistema:

TABLA N° 9

Estructura Tabla Encargados



The image shows a screenshot of a database table structure for the table 'encargados'. The table has the following fields:

Field Name	Data Type
Id	INTEGER
rut	TINYINT
nombre	TINYINT
apellido_paterno	TINYINT
apellido_materno	TINYINT

Fuente: Sistema Inventario.

3.3.3.1.7 Movimientos

Esta tabla es importante por que tiene los datos de los movimientos y bajas de los bienes, aquí se guarda un historial del bien, en donde estaba, los códigos, la fecha en que se movió:

TABLA N° 10

Estructura Tabla Movimientos



The screenshot shows the structure of the 'movimiento' table with the following fields:

Field Name	Data Type
Id	INTEGER
codigo_barra	VARCHAR(20)
fecha_traspaso	DATE
estado_anterior	VARCHAR(12)
servicio_anterior	VARCHAR(6)
dependencia_anterior	VARCHAR(7)
establecimiento_anterior	VARCHAR(4)
fecha_remate	DATE
factura_remate	VARCHAR(10)
num_traspaso	INTEGER
rut_encargado	VARCHAR(10)
nombre	VARCHAR(45)
num_solicitud_baja	INTEGER

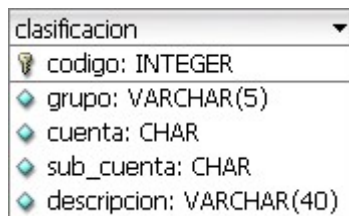
Fuente: Sistema Inventario.

3.3.3.1.8 Clasificación

Existe otra tabla llamada clasificación la cual se usará para guardar la clasificación de los bienes de acuerdo a una nomenclatura usada en contabilidad de grupo, cuentas, subcuenta. En este momento se esta trabajando para unificar las cuentas con la nomenclatura del Minsal, ya que se ocupara para el proyecto de normalización.

TABLA N° 11

Estructura Tabla Clasificación



The screenshot shows the structure of the 'clasificacion' table with the following fields:

Field Name	Data Type
codigo	INTEGER
grupo	VARCHAR(5)
cuenta	CHAR
sub_cuenta	CHAR
descripcion	VARCHAR(40)

Fuente: Sistema Inventario.

3.3.4 MODELO RELACIONAL INVENTARIO

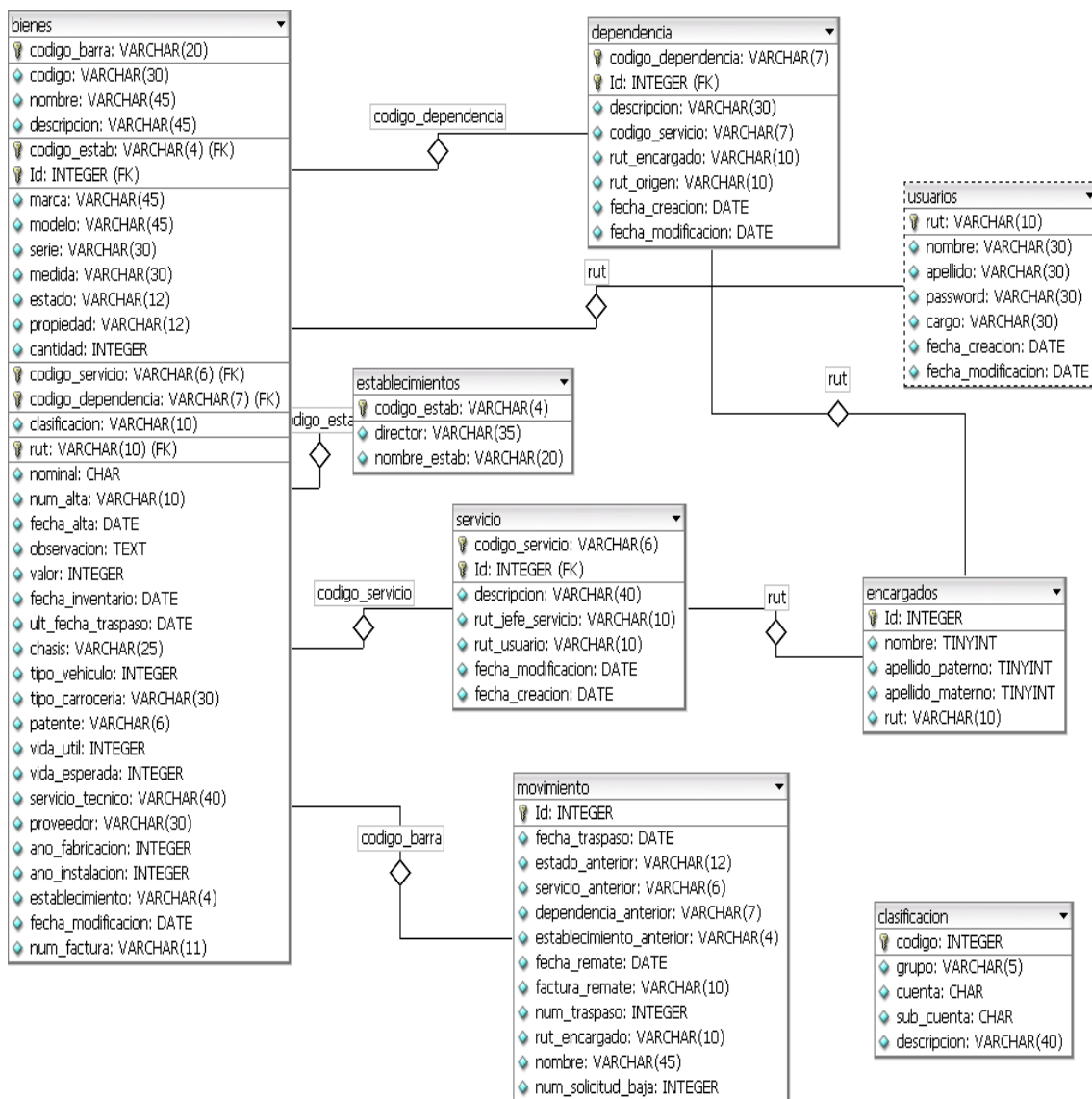


Figura 28. Modelo Relacional.

3.4 IMPLEMENTACIÓN

3.4.1 LENGUAJE PHP

PHP, acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje "Open Source" interpretado de alto nivel, especialmente pensado para desarrollos Web y el cual puede ser embebido en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl y es fácil de aprender. La meta de este lenguaje es permitir escribir a los creadores de páginas Web, páginas dinámicas de una manera rápida y fácil, aunque se pueda hacer mucho más con PHP.

Existen tres campos en los que scripts escritos en PHP son usados.

- Scripts en la parte del servidor. Este es el campo más tradicional y el principal campo de trabajo. Se necesitan tres cosas para que esto funcione. El analizador PHP (CGI ó módulo), un servidor Web y un navegador. Se necesita correr el servidor Web con PHP instalado. El resultado del programa PHP se puede obtener a través del navegador, conectando con el servidor Web.

- Scripts en línea de comandos. Puedes crear un script PHP y correrlo sin ningún servidor Web ó navegador. Solamente se necesita el parseador PHP para usarlo de esta manera. Este tipo de uso es ideal para scripts ejecutados regularmente desde cron (en *nix ó Linux) ó el

Planificador de tareas (en Windows). Estos scripts también pueden ser usados para tareas simples de procesado de texto

- Escribir aplicaciones gráficas clientes. PHP no es probablemente el mejor lenguaje para escribir aplicaciones gráficas, pero si se sabe bien PHP, y se quiere utilizar algunas características avanzadas en programas clientes, se puede utilizar PHP-GTK para escribir dichos programas. Es también posible escribir aplicaciones independientes de una plataforma. PHP-GTK es una extensión de PHP, no disponible en la distribución principal.

- PHP puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo Linux, muchas variantes Unix (incluido HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS y probablemente alguno más. PHP soporta la mayoría de servidores Web de hoy en día, incluyendo Apache, Microsoft Internet Information Server, Personal Web Server, Netscape y iPlanet, O'Reilly Website Pro server, Caudium, Xitami, OmniHTTPd y muchos otros. PHP tiene módulos disponibles para la mayoría de los servidores, para aquellos otros que soporten el estándar CGI, PHP puede usarse como procesador CGI.

Con PHP se tiene la libertad de escoger el sistema operativo y el servidor a gusto. También se tiene la posibilidad de usar programación de procedimientos ó programación orientada a objetos. Aunque no todas las características estándares de la programación orientada a objetos están implementadas en la versión actual de PHP, muchas librerías y aplicaciones grandes (incluyendo la librería PEAR) están escritas íntegramente usando programación orientada a objetos.

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

Quizás la característica más potente y destacable de PHP es su soporte para una gran cantidad de bases de datos. Escribir un interfaz vía Web para una base de datos es una tarea simple con PHP. Las siguientes bases de datos están soportadas actualmente:

TABLA N° 12

Algunas bases de datos soportadas por PHP

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

Fuente: <http://www.wikipedia.org>

```
<html>
<head>
  <title> Ejemplo de PHP</title>
</head>
<body>
  <?php echo "<p>Hola Mundo</p>";
?>
</body>
</html>
```

Figura 29. Ejemplo de Script PHP

3.4.2 PHP Y EL INVENTARIO

Dadas estas características del lenguaje es ideal para trabajar en un sistema como el de inventario, tanto para acceder a una base de datos como para procesar los datos que se introducirán a la base de datos, como los que serán devueltos por la base de datos.

3.4.3 APACHE

El **servidor HTTP Apache** es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que

implementa el protocolo HTTP/1.1 (RFC 2616) y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que originalmente Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, *a patchy server* (un servidor *parcheado*)

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

En la actualidad (2005), Apache es el servidor HTTP más usado, siendo el servidor HTTP del 68% de los sitios Web en el mundo y creciendo aún su cuota de mercado (estadísticas históricas y de uso diario proporcionadas por Netcraft).

El núcleo 2.x de Apache tiene varias mejoras clave sobre el núcleo de Apache 1.x. Estas mejoras incluyen threads de UNIX, mejor soporte para plataformas no Unix (como Windows), un nuevo API, y soporte de IPv6

3.4.3.1 Módulos

La arquitectura del servidor Apache es muy modular. El servidor consta de un sección *core* y mucha de la funcionalidad que podría considerarse básica para un servidor Web es provista por módulos. Algunos de estos son:

- `mod_ssl` - Comunicaciones Seguras vía TLS.
- `mod_rewrite` - reescritura de direcciones servidas.
- `mod_dav` - Soporte del protocolo WebDAV (RFC 2518).
- `mod_deflate` - Compresión transparente con el algoritmo deflate del contenido enviado al cliente.
- `mod_auth_ldap` - Permite autenticar usuarios contra un servidor LDAP.

El servidor de base puede ser extendido con la inclusión de módulos externos entre los cuales se encuentran:

- `mod_perl` - Páginas dinámicas en Perl.
- `mod_php` - Páginas dinámicas en PHP.
- `mod_python` - Páginas dinámicas en Python.
- `mod_jk` - Conector para enlazar con el servidor Jakarta Tomcat de páginas dinámicas en Java (servlets y JSP). En la próxima versión del servidor este módulo se incluirá en la distribución base.

3.4.4 MYSQL

Esta es una base de datos desarrolladas bajo la filosofía de código abierto. La desarrolla y mantiene la empresa MySQL AB pero puede utilizarse gratuitamente y su código fuente está disponible.

Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello, atrajo a los desarrolladores de páginas Web con contenido dinámico, justamente por su simplicidad; aquellos elementos faltantes fueron llenados por la vía de las aplicaciones que la utilizan.

Entre las características disponibles en las últimas versiones se puede destacar:

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones o el mayor número de operaciones disponibles.
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

La licencia GPL de MySQL obliga a distribuir cualquier producto derivado

(aplicación) bajo esa misma licencia. Si un desarrollador desea incorporar MySQL en su producto pero no desea distribuirlo bajo licencia GPL, puede adquirir la licencia comercial de MySQL que le permite hacer justamente eso.

3.4.5 JAVASCRIPT

JavaScript es un lenguaje interpretado orientado a las páginas Web, con una sintaxis semejante a la del lenguaje Java.

El lenguaje fue inventado por Brendan Eich en la empresa Netscape Communications, que es la que fabricó los primeros navegadores de Internet comerciales.

Apareció por primera vez en el producto de Netscape llamado Netscape Navigator 2.0.

Se utiliza en páginas Web HTML, para realizar tareas y operaciones en el marco de la aplicación cliente.

Los autores inicialmente lo llamaron Mocha y más tarde LiveScript pero fue rebautizado como JavaScript en un anuncio conjunto entre Sun Microsystems y Netscape, el 4 de diciembre de 1995.

En 1997 los autores propusieron JavaScript para que fuera adoptado como estándar de la the European Computer Manufacturers' Association ECMA, que a pesar de su nombre no es europeo sino internacional, con sede en Ginebra. En junio de 1997 fue adoptado como un estándar ECMA, con el nombre de

ECMAScript. Poco después también lo fue como un estándar ISO.

JScript es la implementación de ECMAScript de Microsoft, muy similar al JavaScript de Netscape, pero con ciertas diferencias en el modelo de objetos del navegador que hacen a ambas versiones con frecuencia incompatibles.

Para evitar estas incompatibilidades, el World Wide Web Consortium diseñó el estándar Document Object Model (DOM, ó Modelo de Objetos del Documento en castellano), que incorporan las versiones 6 de Internet Explorer y Netscape Navigator, Opera versión 7, y Mozilla desde su primera versión.

Existen objetos predefinidos, cada uno con distintas propiedades, métodos y manejadores de eventos. A diferencia de JAVA aquí no se puede derivar clases, por los que se habla de una jerarquía de instancia.

Los objetos en JavaScript los podemos clasificar en 4 niveles:

1. **window**, navigator, Math, String y Date
 2. **document**, history, location, Frame y frames
 3. **Form**, Anchor y link
 4. Button, Checkbox, Hidden, Password, Radio, elements, Select, Text, Textarea, Submit y Reset.
- Todo el segundo nivel pertenece a Windows.
 - Todo el tercer nivel pertenece a document.
 - Todo el cuarto nivel pertenece a form.

3.4.6 CSS

Las **hojas de estilo en cascada** (*Cascading Style Sheets*, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirá de estándar para los agentes de usuario o navegadores.

La idea que se encuentra detrás del desarrollo de CSS es separar la *estructura* de un documento de su *presentación*.

Por ejemplo, el elemento de HTML H1 indica que un bloque de texto es un encabezamiento y que es más importante que un bloque etiquetado como <H2>. Versiones más antiguas de HTML permitían atributos extra dentro de la etiqueta abierta para darle formato (como el color o el tamaño de fuente). No obstante, cada etiqueta <H1> debía disponer de esa información si se deseaba un diseño consistente para una página, y además, una persona que lea esa página con un navegador pierde totalmente el control sobre la visualización del texto.

Cuando se utiliza CSS, la etiqueta <H1> no debería proporcionar información sobre como va a ser visualizado, solamente marca la estructura del documento. La información de estilo separada en una hoja de estilo, especifica como se ha de mostrar <H1>: color, fuente, alineación del texto, tamaño, y otras características no visuales como definir el volumen de un sintetizador de voz,

por ejemplo.

La información de estilo puede ser adjuntada tanto como un documento separado o en el mismo documento HTML. En este último podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo "style".

Las ventajas de utilizar CSS (u otro lenguaje de estilo) son:

- Control centralizado de la presentación de un sitio Web completo con lo que se agiliza de forma considerable la actualización del mismo.
- Los Navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio Web remoto, con lo que aumenta considerablemente la accesibilidad. Por ejemplo, personas con deficiencias visuales pueden configurar su propia hoja de estilo para aumentar el tamaño del texto o remarcar más los enlaces.
- Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario. Por ejemplo, para ser impresa, mostrada en un dispositivo móvil, o ser "leída" por un sintetizador de voz.
- El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño.

Hay varias versiones: CSS1 y CSS2, con CSS3 en desarrollo por el World Wide Web Consortium (W3C). Los navegadores modernos implementan CSS1 bastante bien, aunque existen pequeñas diferencias de implementación según marcas y versiones de los navegadores. CSS2, sin embargo, está solo parcialmente implantado en los más recientes.

3.4.7 MODELO CLIENTE SERVIDOR

La arquitectura cliente-servidor llamado modelo cliente-servidor o servidor-cliente es una forma de dividir y especializar programas y equipos de cómputo a fin de que la tarea que cada uno de ellos realiza se efectúe con la mayor eficiencia, y permita simplificar las actualizaciones y mantenimiento del sistema.

En esta arquitectura la capacidad de proceso está repartida entre el servidor y los clientes.

En la funcionalidad de un programa distribuido se pueden distinguir 3 capas o niveles:

1. Manejador de Base de Datos (Nivel de almacenamiento),
2. Procesador de aplicaciones o reglas del negocio (Nivel lógico) y
3. Interfase del usuario (Nivel de presentación)

En una arquitectura **monolítica** no hay distribución; los tres niveles tienen lugar en el mismo equipo.

En un comienzo, los mainframes concentraban la funcionalidad de almacenamiento (#1) y lógica (#2) y a ellos se conectaban terminales tontas, posiblemente ubicadas en sitios remotos.

En el modelo cliente-servidor, en cambio, el trabajo se reparte entre dos ordenadores. De acuerdo con la distribución de la lógica de la aplicación hay dos posibilidades:

1. Cliente delgado: si el cliente solo se hace cargo de la presentación.
2. Cliente pesado: si el cliente asume también la lógica del negocio.

En la actualidad se suele hablar de arquitectura de tres niveles, donde la capa de almacenamiento y la de aplicación se ubican en (al menos) dos servidores diferentes, conocidos como servidores de datos y servidores de aplicaciones.

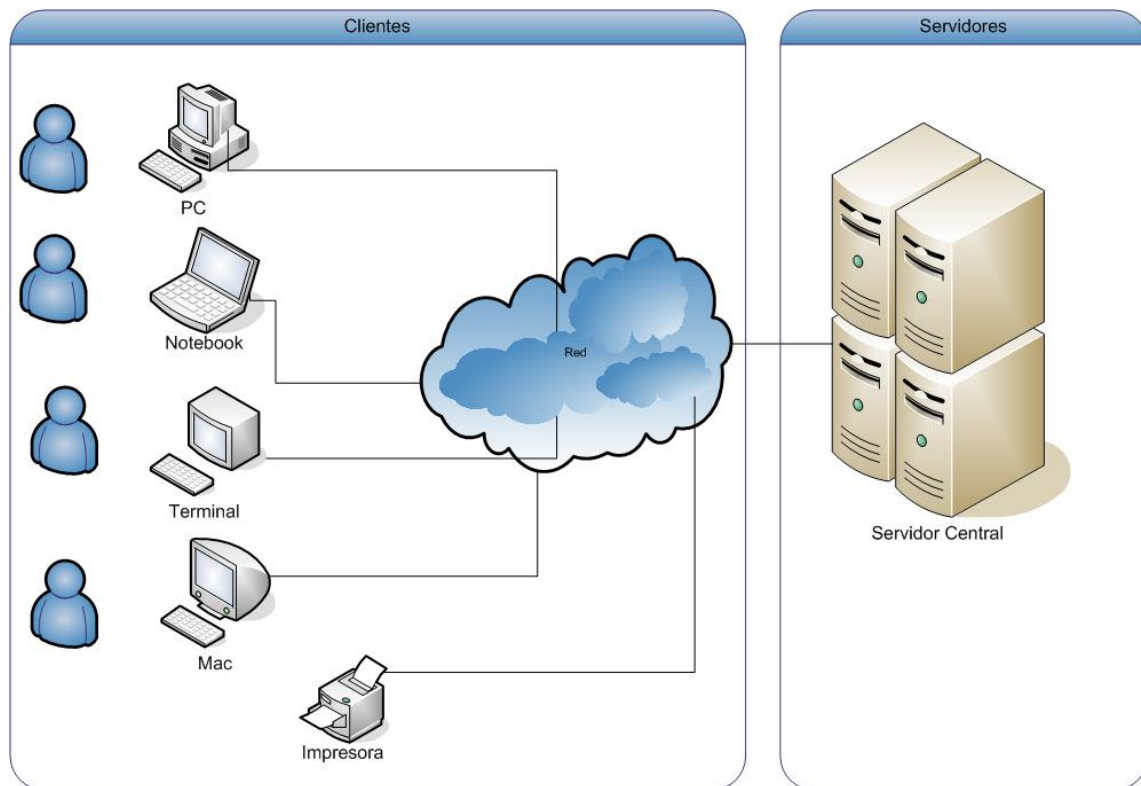


Figura 30. Modelo Cliente-Servidor.

3.4.7.1 Ventajas de la arquitectura cliente-servidor

- El servidor no necesita tanta potencia de procesamiento, parte del proceso se reparte con los clientes.
- Se reduce el tráfico de red considerablemente. Idealmente, el cliente se conecta al servidor cuando es estrictamente necesario, obtiene los datos que necesita y cierra la conexión dejando la red libre.

3.4.8 APLICACIONES WEB

En ingeniería del software una aplicación Web es aquella que los usuarios usan desde un servidor Web a través de Internet o de una intranet. Las aplicaciones Web son populares debido a la ubicuidad del navegador como un cliente, a veces llamado cliente ligero. La habilidad para actualizar y mantener aplicaciones Web sin distribuir e instalar software en miles de potenciales clientes es otra razón de su popularidad. Aplicaciones como los Webmail, amazon.com y eBay son ejemplos bien conocidos de aplicaciones Web, pero éstas tienen usos en otras áreas como los negocios y la ciencia, debido a las limitaciones inherentes de los actuales navegadores Web.

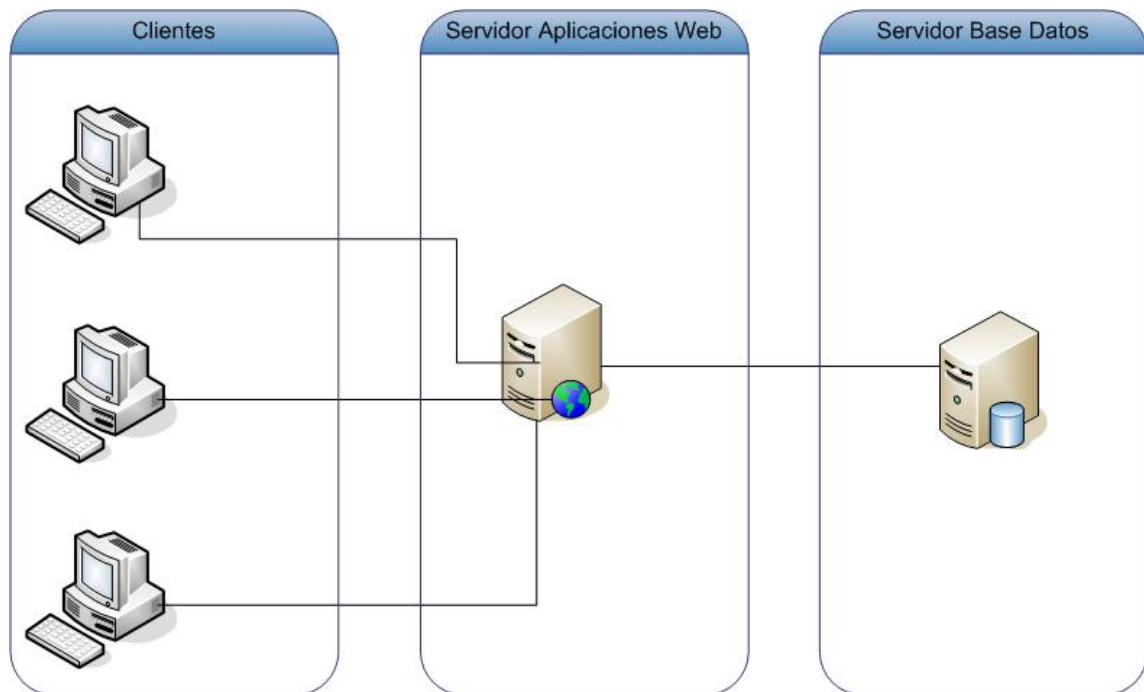


Figura 31. Modelo Aplicación Web.

3.4.9 SOFTWARE NECESARIO

Para implementar el sistema de inventario se debe tener los siguientes componentes:

3.4.9.1 Hardware:

- Disco duro de 40 Gb o más.
- Procesador de 1800 Mhz o mas.
- 512 Mb RAM o más.
- Lector y/o grabador CD-ROM.
- Unidad disquete 3 ¼.
- Red intranet/Internet.

3.4.9.2 Software:

- Sistema operativo Linux.
- Servidor Web apache.
- Compilador PHP.
- Base de datos MySQL.

3.4.10 INSTALACIÓN DEL SOFTWARE

Estos son los pasos necesarios para instalar exitosamente los programas necesarios para que funcione el sistema de inventario sobre una máquina con sistema operativo Linux.

3.4.10.1 Instalando apache

Se descarga la versión de apache elegida en este caso se puede descargar desde:

```
wget http://apache.247available.com/httpd/httpd-2.0.49.tar.gz
```

Se extrae el archivo:

```
tar xvfz httpd-2.0.49.tar.gz
```

Se cambia de directorio a l recién creado:

```
cd httpd-2.0.49
```

Ahora se configuran las opciones deseadas:

--enable-so: para poder compilar PHP como módulo dinámico.

--enable-rewrite: algunas reglas para corregir URL.

--enable-deflate: para comprimir las páginas al ser enviadas.

Además hay un gran número de otras opciones que se pueden agregar a gusto

y según lo que se desee hacer.

Ahora se invoca el comando para aplicar la configuración:

```
./configure --enable-so --enable-rewrite --enable-deflate=yes
```

Luego:

```
Make
```

Luego:

```
Make install
```

Para simplificar la administración de apache puede ser conveniente agregar en el PATH la ruta del binario de apache:

```
export PATH=$PATH:/usr/local/apache2/bin
```

Con esto ya esta listo el servidor Web apache.

3.4.10.2 Instalando MySQL

Se descarga el software de <http://www.mysql.com>

Luego se descomprime:

```
Tar xvfz mysql-4.1.13.tar.gz
```

Se entra al directorio recién creado:

cd mysql-4.1.13

Antes de seguir es necesario por seguridad crear un usuario y un grupo de nombre mySQL con el fin de que sea este usuario bajo el que se ejecute MySQL, y sea también el propietario de las bases de datos.

groupadd mysql**useradd mysql -q mysql mysql**

Mediante el programa configure se realice una configuración automática de los fuentes comprobando que todas las librerías necesarias están instaladas. Es posible utilizar diferentes parámetros para variar el comportamiento por defecto de la instalación. Así por ejemplo el parámetro importante es --prefix, mediante el cual se indica el path en el que se desea instalar la aplicación, los otros parámetros se pueden ver haciendo:

./configure --help.

./configure --prefix=/usr/local/mysql

Luego se compilan las fuentes:

make

Luego se instalan los archivos:

make install

Una vez instalado hay varias cosas que configurar antes de poder empezar a trabajar, como crear un archivo de configuración, crear las tablas de permisos (grant tables), configurar los permisos de los programas y la carpeta de datos y, por ultimo, configurar MySQL para que se inicie automáticamente al arrancar la maquina.

Crear archivo de configuración: en la carpeta support-files hay varios ejemplos que se pueden tomar como base.

```
cp support-files/my-medium.cnf /etc/my.cnf
```

```
chown mysql:mysql /etc/my.cnf
```

```
chmod 600 /etc/my.cnf
```

Se le pone permisos de lectura y escritura únicamente al usuario root, de forma de que nadie mas o pueda acceder al archivo.

Tablas de permisos: la primera vez que se instala MySQL es necesario crear estas tablas, para ello disponemos de un programa que permite hacerlo automáticamente.

```
/usr/local/mysql/bin/mysql_install_db --user=mysql
```

El valor mySQL que se indica en el parámetro –user debe coincidir con el usuario que se ha creado al comienzo de la instalación.

Permisos: se debe configurar los permisos de los programas generados así como de la carpeta que contiene las bases de datos. Se colocara como propietario de todos los archivos al usuario root, y como grupo mySQL, salvo en la carpeta de datos var y su contenido, que le pondremos como usuario propietario mySQL.

```
chown -R root /usr/local/mysql
```

```
chown -R mysql /usr/local/mysql/var
```

```
chgrp -R mysql /usr/local/mysql
```

Arranque de MySQL: se lanzara el servidor MySQL para comprobar si la instalación fue exitosa.

```
/usr/local/mysql/bin/mysqld_safe -user=mysql &
```

Se puede entrar en la consola de mySQL utilizando el usuario root, el cual por defecto no tiene password, lo cual se cambiara por seguridad.

```
/usr/local/mysql/bin/mysql -uroot
```

Una vez que todo a funcionado bien se configure para que se active automáticamente al iniciar la máquina, para esto se copia el archive support-files/mysql.server en la ubicación de scripts de arranque en la maquina.

```
cp support-files/mysql.server /etc/rc.d/init.d/mysqld
```

Se puede probar el script llamándole directamente e indicándole la acción que queremos: start, stop o restart.

Una vez copiado el script es necesario activarlo, por ejemplo usando chkconfig:

```
chkconfig --add mysqld.
```

Para cambiar la contraseña a root se usa la siguiente instrucción.

```
./mysqladmin -u root password "nuevacontraseña"
```

Para crear la base de datos:

```
./mysqladmin create hospital -uroot -p
```

Si se quiere administrar vía remota con algún cliente mySQL se debe crear un usuario con estas características con los privilegios necesarios.

```
./mysql -uroot -p
```

```
mysql> GRANT ALL PRIVILEGES ON hospital.* TO 'root@%' IDENTIFIED  
BY 'contraseña'
```

3.4.10.3 Instalando PHP

Se obtiene el paquete desde algún sitio Web

```
wget http://us4.php.net/get/php-4.3.6.tar.gz/from/this/mirror
```

Se extrae el archive y se accede a la carpeta recién creada:

```
Tar xvfz php-4.3.6.tar.gz
```

```
Cd php-4.3.6
```

Los parámetros de configuración que se usaran son:

--with-apxs2: para que se cargue como módulo dinámico de apache.

--with-mySQL: para usarlo con la base de datos MySQL.

Por supuesto se pueden agregar o borrar opciones según las necesidades.

```
./configure --with-apxs2=/usr/local/apache2/bin/apxs --with-  
mySQL=/usr/local/mySQL
```

Luego se compila y se instala:

Make

Make install

Después de esto hay que modificar los archivos de configuración de apache para que reconozca las extensiones .php como código PHP. Se puede declarar cualquier extensión para que se procese como PHP como puede ser .html o .asp, en el archivo /usr/local/apache2/conf/httpd.conf hay que agregar estas líneas a la altura donde están definidos los otros AddType.

```
AddType application/x-httpd-php .php
```

```
AddType application/x-httpd-php .wml
```

```
AddType application/x-httpd-php-source .phps
```

También hace falta el archivo de configuración de PHP, el que se puede copiar con la configuración recomendada para luego modificarlo según las necesidades:

```
cp /php.ini-recomended /usr/local/lib/php.ini
```

En este caso se editara y se cambiara el valor register-globals=On para que se pueda hacer uso de las variables ambiente de PHP.

Ahora si se puede arrancar apache con los cambios:

```
apachectl stop
```

```
apachectl start
```

Para verificar la instalación lo mejor es ejecutar un pequeño programa PHP que ejecute la función phpinfo(), ahí verifica la lista de módulos instalados y su configuración:

```
<html>
<head>
  <title> Configuración de PHP</title>
</head>
<body>
  <?php phpinfo(); ?>
</body>
</html>
```

Figura 32. Verificación de la instalación de PHP.

3.4.11 EDITOR

Para este trabajo se eligió un programa llamado “PHP Designer 2005” como editor de código, esto debido a sus buenas características técnicas y una completa línea de herramientas necesarias para la codificación de PHP. El ambiente del desarrollo en él combina muchas características como esquemas personalizables de sintaxis para PHP, HTML, XHTML, CSS, Perl, C #, Javascript, VB, Java y SQL (Ingres, Interbase, MSSQL, MySQL, Oracle, Sybase y SQL estándar), un browser de clases, prueba y elimina errores, también integra el manual de PHP, tiene acceso a las bibliotecas comunes de código. El diseñador 2005 de PHP también incluye características de ahorro de tiempo como un encargado del FTP, encargado de proyecto, encargado de archivos, de las plantillas del código, posibilidad de la exportación HTML, rtf, etc.

Este programa es freeware, pero la principal razón por la que se eligió es la demarcación de llaves, corchetes y paréntesis lo que facilita en gran medida el control del código escrito especialmente cuando este empieza a crecer demasiado. Como desventaja se puede decir que no provee vista de diseño como Dreamweaver.

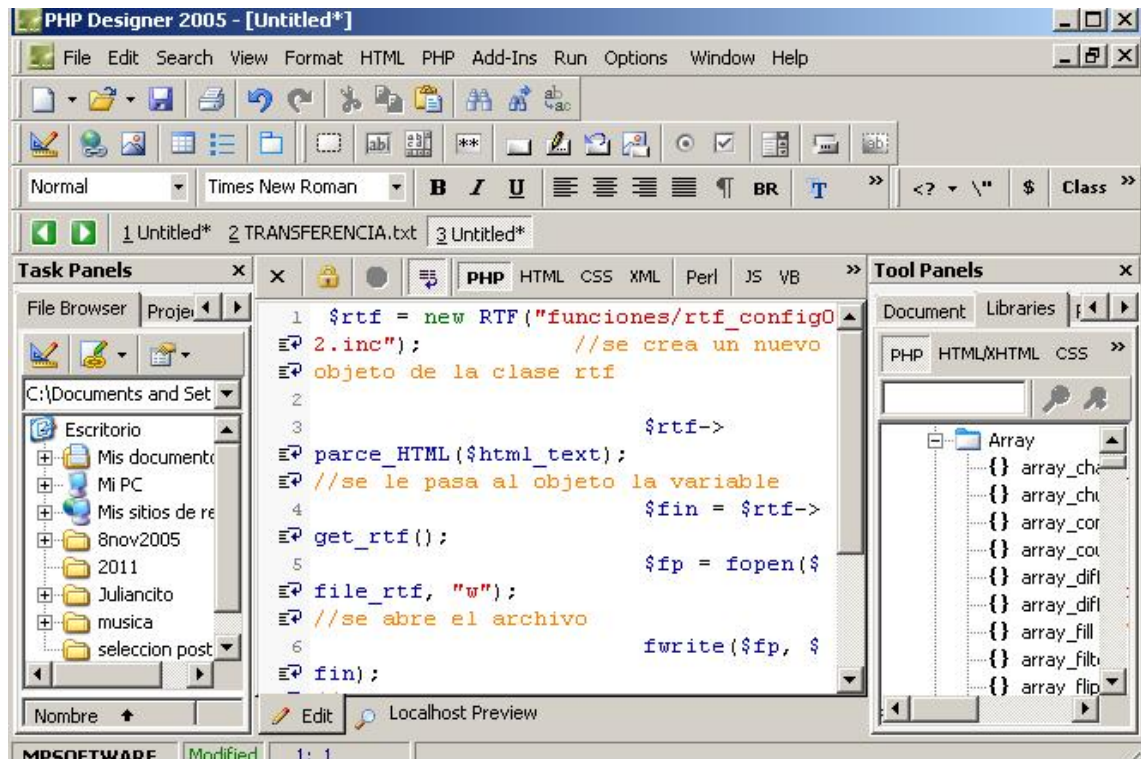


Figura 33. Editor PHP.

3.4.12 FUNCIONES IMPORTANTES

mysql_close -- cierra el enlace con MySQL

mysql_connect -- Abre una conexión a un servidor MySQL

mysql_error -- Devuelve el texto del mensaje de error de la última operación MySQL

mysql_fetch_array -- Extrae la fila de resultado como una matriz asociativa, una matriz numérica o ambas

mysql_query -- Envía una consulta de MySQL

3.4.13 IMPLEMENTACIÓN DEL SISTEMA



Figura 34. Inicio Sistema.

Cabe aclarar que por seguridad en todos los módulos se pedirá un usuario y una contraseña, talo que también servirá para identificar que acciones esta ejecutando cada usuario, guardando en la base de datos esta información como el Rut, la fecha, etc.

3.4.13.1 Módulo ingreso

En este módulo se hará el ingreso de los datos para esto se necesita un listbox que muestre primero a que servicio se desea acceder y luego en otro dentro del

servicio en que dependencia se quiere insertar los datos.

Para esto se crea un algoritmo que lee de la base de datos, los servicios y dependencias disponibles en forma dinámica, para esto se usara php y javascript:

```
$SQLserv="select * from servicio order by codigo_servicio";  
  
$res=mysql_query($SQLserv);  
  
while($fila=mysql_fetch_array($res)){  
  
print("<option value=\""$fila[codigo_servicio]\"");  
  
print(">$fila[codigo_servicio] $fila[descripcion]</option>\n ");  
  
print("");  
  
}
```

Con estas líneas lee la tabla donde se encuentran los servicios y los lista en un listbox.

De forma análoga se llama a la tabla dependencias para listarlas mostrando su código y descripción.

Luego ingresa al formulario de toma de inventario en formulario.php, aquí se presenta un formulario para ingresar los datos del inventario. Una vez que se ingresa un registro, este chequea en la base de datos si existe el código de barra que es la clave primaria de la tabla bienes. También cheque usando JavaScript que algunos campos obligatorios no estén en blanco al momento de

ingresar los datos a la base de datos.

```
$sentencia = mysql_query("INSERT INTO bienes
    (codigo,nombre,descripcion,marca,modelo,serie,medida,estado,propiedad,codigo
_servicio,codigo_dependencia,codigo_barra,rut,clasificacion,fecha_inventario,chasis,tipo
_vehiculo,tipo_carroceria,patente,vida_util,vida_esperada,servicio_tecnico,proveedor,a
no_fabricacion,ano_instalacion,valor,num_alta)
VALUES
('$nInventario','$nomEspecie','$descripcion','$marca','$modelo','$serie','$medida','$estad
o','$propiedad','$servicio','$dependencia','$usoInt','$rut','$clasificacion','$fecha_reg','$cha
sis','$tipo_vehi','$tipo_carr','$patente','$vida_util','$vida_espe','$serv_tec','$proveedor','$a
no_fab','$ano_inst','$valor','$n_alta')",$link);
if(!$sentencia)
{
    echo "<br><b>Error al Agregar Registro</b>";
}
```

INGRESO DE NUEVOS BIENES

Número Antiguo Inventario	2- <input type="text"/>	Marca	<input type="text"/>	Grupo	<input type="text" value="00000"/>
Nombre Especie	<input type="text"/>	Modelo	<input type="text"/>	Cuenta	<input type="text" value="00"/>
Descripción	<input type="text"/>	Serie	<input type="text"/>	Sub Cuenta	<input type="text" value="000"/>
Estado	Bueno(B) <input type="button" value="v"/>	Medida	<input type="text"/>	Proveedor	<input type="text"/>
Propiedad	Hospital(H) <input type="button" value="v"/>	Vida Útil	<input type="text"/>	Servicio Técnico	<input type="text"/>
Cantidad	<input type="text" value="1"/>	Vida Esperada	<input type="text"/>	Año Instalación	<input type="text"/>
Número Inventario	2- <input type="text"/>	Año Fabricación	<input type="text"/>	Tipo Vehículo	<input type="text"/>
Chasis	<input type="text"/>	Patente	<input type="text"/>	Tipo Carrocería	<input type="text"/>
Valor \$	<input type="text"/>	Número Alta	<input type="text"/>		

Figura 34. Formulario de Ingreso

Una vez aceptada la información de parte de la base de datos se irán listando los artículos ingresados por el usuario, cabe señalar que solo se mostrara los bienes ingresados por el usuario.

		Usuario		Fecha											
		13964203-1		26-12-2005											
		Servicio	Dependencia	Jefe Servicio	Encargado	Reg. Ingresados por usuario									
		1941 SECCION COMPUTACION		02018 SALA COMPUTACION		4									
Acciones	Numero Inventario Antiguo	Nombre Especie	Descripción	Característica				Estado	Propiedad	Cantidad	Numero Inventario	Grupo	Cuenta	Sub Cuenta	Chasis
				Marca	Modelo	Serie	Medida								
	2-	PC	CPU DISKETERA LECTOR CD	AOPEN	COMPONENT PRO			BUENO	HOSPITAL	1	21742	00000	00	000	
	2-	PC	EQUIPO NUEVO	OLIDATA	CELERON	004472-000004		BUENO	HOSPITAL	1	21483	00000	00	000	
	2-	PC	EQUIPO NUEVO	OLIDATA	CELERON	0044F2-000001		BUENO	HOSPITAL	1	21486	00000	00	000	
	2-	TET	ERTE					BUENO	HOSPITAL		20101	00000	00	000	

Figura 35. Vista de Ingreso

3.4.13.2 Módulo consulta de bienes

En este módulo el usuario puede consultar todo lo referente a algún bien de acuerdo a cualquiera de las características ingresadas. Se tendrán 2 columnas una con los criterios por los cuales se va a realizar la búsqueda, y otra en donde el usuario puede elegir las columnas que le interesan ver. Esto se encuentra en `lispers.php`.

Ingrese en el cuadro correspondiente la información que desea buscar

Código barra	<input type="text"/>	Estado	<input type="text" value="—"/>
Código antiguo	<input type="text"/>	Propiedad	<input type="text" value="—"/>
Marca (empieza con)	<input type="text"/>	Nombre (empieza con)	<input type="text"/>
Modelo (empieza con)	<input type="text"/>	Descripción (empieza con)	<input type="text"/>
Serie (empieza con)	<input type="text"/>	Grupo (Código)	<input type="text"/>
Servicio (Código) ?	<input type="text"/>	Cuenta (Código)	<input type="text"/>
Dependencia (Código) ?	<input type="text"/>	Subcuenta (Código)	<input type="text"/>

Figura 36. Búsqueda de Bienes.

Se implementa una estructura de decisiones donde se decidirá la sentencia SQL que se ejecutará sobre la base de datos, en base a las opciones que vaya escogiendo el usuario tanto en la parte de la consulta como en la parte de la muestra de datos.

```
if($codigo_barra != "") { $SQL .= "bienes.codigo_barra = '$codigo_barra' AND ";
```

```
$titulo2 .= "<tr><td>Código Barra
```

```
<i>$codigo_barra<i></td></tr>\n";
```

```
$titulo2txt .= "Código Barra: $codigo_barra\n";}
```

...

Luego se juntan las opciones elegidas por el usuario en una select que será la consulta final que se enviara a la base de datos, se usara la tabla bienes donde esta la mayoría de la información y la tabla servicios y dependencias de donde se sacara los nombres de los códigos asociados a estos servicios y dependencias.

```
$$SQL .= " 1 "; //opciones elegidas por el usuario concatenadas aquí
```

```
$con = $$SQL;
```

```
$$SQL = "SELECT codigo_barra, código, nombre, bienes.descripcion as bienes_desc,
```

```
marca, modelo, serie, medida, estado, propiedad, cantidad,
```

```
servicio.codigo_servicio,servicio.descripcion as serv_desc,
```

```
dependencia.codigo_dependencia,dependencia.descripcion as dep_desc,
```

```
clasificación, Rut, nominal, num_alta, fecha _ alta, observación, valor,
```

```
fecha _ inventario FROM servicio INNER JOIN(dependencia INNER JOIN bienes
```

```
ON bienes.codigo_dependencia = dependencia.codigo_dependencia) ON
```

```
bienes.codigo_servicio = servicio.codigo_servicio WHERE ".$$SQL."order by
```

```
".$ordenar;
```

Esta consulta genera de vuelta un array con los resultados el que se muestra en una nueva página Web, creada dinámicamente este array se muestra dentro de un ciclo While el cual crea una tabla html donde se Irán listando todos los bienes que corresponden con la búsqueda. Para esto se crearon 2 funciones

una que esta pensada para escribir los títulos de las columnas seleccionadas, y otra función para ir escribiendo los datos.

```
mostrarA($fp, $fptxt, $genera, $codB, $codA, $nom, $des, $mar, $mod, $ser, $med,
$est, $pro, $can, $codS, $codD, $rut, $nomi, $codG, $codC, $codSU, $alt, $fec, $obs,
$val); //muestra cabeceras
do {
    mostrarB($fp, $fptxt, $genera, $row, $codB, $codA, $nom, $des, $mar,
$mod, $ser, $med, $est, $pro, $can, $codS, $codD, $rut, $nomi, $codG, $codC, $codSU,
$alt, $fec, $obs, $val); //muestra detalle
} while($row = mySQL_fetch_array($result));
```

3.4.13.3 Módulo hojas murales

Este es uno de los más importantes ya que aquí se podrán entregar los informes que serán luego adheridos a las respectivas dependencias como reemplazo de las hojas murales antiguas obtenidas del sistema antiguo.

Aquí se eligió los campos más representativos para el encargado de inventario al momento de identificarse con una dependencia. En este caso se mostrara el número de inventario, la cantidad, nombre, descripción, estado, propiedad. Estos reportes son exportados a un archivo RTF el cual es guardado en el servidor para luego ser descargado por el usuario.

El usuario tiene dos opciones para sacar estos reportes. En el primero se saca por servicio, aquí se hace un select a la tabla de servicio que entregará toda la información referente a él, esta se almacena en un arreglo y se muestra en un listbox, desde donde se puede elegir a que servicio se le generará la hoja mural.

Una vez que se elige el servicio, este código se pasa a la página generaServ.php, aquí se genera el archivo RTF que contendrán las hojas murales de todas las dependencias de este servicio. Primero se crea una variable con formato htm que tiene la forma del archivo RTF en donde se almacenan los valores que se quieren mostrar. Luego para crear el archivo RTF se crea un objeto de tipo RTF. Se le pasa la variable que contiene el documento en forma de htm, luego se escribe el objeto resultante en el archivo predefinido para esta función.

```
$rtf = new RTF("funciones/rtf_config02.inc");           //se crea un nuevo objeto de la
clase rtf

                $rtf->parce_HTML($html_text);           //se le pasa al
objeto la variable

                $fin = $rtf->get_rtf();

                $fp = fopen($file_rtf, "w");           //se abre el archivo

                fwrite($fp, $fin);                     //se guarda

                echo "Archivo generado, Nombre: <b>".$file_rtf."</b> guardado en disco.";

//mensaje
```

@fclose(\$fp);

//se cierra el

archivo

Este procedimiento es análogo en el caso que se elija la opción generar las hojas murales por dependencia en donde se traspasa el código de dependencia y se le entrega a generaDep.php.

Servicio Salud Araucanía
Departamento de Finanzas

Fecha: 13/11/2005

INVENTARIO

Establecimiento: Hospital Doctor Hernán Henríquez Aravena
Servicio: SERVICIO CLINICO PEDIATRIA Código: 3340
Dependencia: SALA MEDICO Código: 01002

Encargado de Inventario-----
Jefe Servicio

N Inv.	Cant.	Nombre	Descripción	Estado	Propiedad
13005	1	BOTIQUIN	MELAMINA PUERTA VIDRIO	BUENO	OTROS
12988	1	COMPUTADOR		BUENO	HOSPITAL
12980	1	CUADRO	MISION SERVICIO	BUENO	OTROS
13004	1	CUADRO	DECORATIVO FLOR	BUENO	OTROS
12990	1	ESCRITORIO	MADERAY TEVINIL	BUENO	PERSONAL
13000	1	ESCRITORIO	MADERAY VIDRIO	BUENO	PERSONAL
13002	1	ESCRITORIO COMPUTADOR	2 CAJONES ENCHAPADO	BUENO	OTROS
13003	1	LAMPARA PIE	METALICA	BUENO	PERSONAL
12987	1	MUEBLE COMPUTADOR	METALICO	BUENO	HOSPITAL
13001	1	MUEBLE LAVAMANOS	MELAMINA 2 PUERTAS	BUENO	PERSONAL
13006	1	MUEBLE MODULAR	MASISA Y VIDRIO 2 CUERPOS	REPARABLE	OTROS
12986	1	NEGATOSCOPIO		BUENO	HOSPITAL
12979	1	PIZARRON	MELAMINA	BUENO	OTROS
12981	1	RELOJ	PARED	BUENO	OTROS
12982	1	ROPERILLO	6 PUERTAS ENCHAPADO	BUENO	HOSPITAL
12983	1	ROPERILLO	METALICO 2 CUERPOS	BUENO	HOSPITAL

Figura 37. Hoja Mural

3.4.13.4 Módulo Traspaso de bienes

Este módulo fue construido con el fin de modificar el campo código de barra de los a bienes algo que no se puede hacer cuando se esta ingresando un bien en

el formulario de ingreso, allí se pueden modificar todos los campos menos el código de barra, el servicio y la dependencia, la modificación de los códigos de servicio y/o dependencia significa que se esta ingresando en otro servicio lo cual debe ser tratado con mucho cuidado ya que el error de un numero significa que se esta ingresando en un lugar físico totalmente distinto.

Codigo Barra		2	-	<input type="text"/>	<input type="button" value="Buscar"/>	<input type="button" value="Borrar"/>
Traspaso N°		<input type="text" value="buscar"/>				
Fecha actual		<input type="text" value="2005-12-26"/>				
ACTUALMENTE						
Codigo Barra	2 -	<input type="button" value="Buscar"/>	<input type="text" value="Buscar"/>			
Estado	<input type="button" value="Buscar"/>					
Establecimiento	<input type="button" value="Buscar"/>					
Servicio	<input type="button" value="Buscar"/>	<input type="text" value="Buscar"/>				
Dependencia	<input type="button" value="Buscar"/>	<input type="text" value="Buscar"/>				
DESTINO						
Establecimiento	<input type="text" value="0953 Hosp. Temuco"/> <input type="button" value="v"/>					
Servicio	<input type="text" value="Seleccione el servicio"/> <input type="button" value="v"/>					
<input type="button" value="Traspasar"/>						

Figura 38. Traspaso de Bienes

Se busca el bien que se quiere moer o traspasar de lugar para esto se usa su código de barra en el textfield buscar y si existe mostrara algunas de sus características, como el nombre, servicio actual, dependencia actual, descripciones. Al tener listo el artículo a procesar se elige en un listbox leído desde la base de datos el nuevo destino, el que se puede determinar usando estos controles. Una vez que todos los datos están completos se llama al

módulo interno que realiza este cambio en este caso hacetraspaso.php.

```
$cambio=" UPDATE bienes
        SET
            codigo_servicio= '$nuevoSer',
            codigo_dependencia= '$nuevaDep',
            ult_fecha_traspaso= '$fecha',
            establecimiento= '$estDestino'

        where codigo_barra = $codigo ";
```

A este módulo se le envía el código de barra, código de dependencia y el código de servicio. Con esta información se puede ejecutar el cambio de dependencia. Con esta operación también se ingresan algunos valores a la tabla de movimientos donde queda registrado el traspaso guardando un histórico con el código de servicio, de dependencia anterior, el usuario que lo realizó y la fecha en que se realizó; esto será útil en el momento en que se quiera hacer el seguimiento de un bien.

```
$movimiento=" INSERT INTO
    movimiento
        (codigo_barra,
         fecha_traspaso,
         estado_anterior,
```

```
servicio_anterior,  
dependencia_anterior,  
establecimiento_anterior,  
num_traspaso,  
rut_encargado,  
nombre)  
VALUES  
('$codigo','$fecha','$est','$serv','$dep','$estDestino','$numtraspaso','$rut','$nom') ";
```

3.4.13.5 Módulo Baja Bienes

Este módulo se construyó para representar el momento en que un bien ha cumplido su vida útil o se ha dañado en forma irreparable. La idea es similar al módulo anterior, se elige el bien que se quiere dar de baja utilizando la búsqueda que se ofrece. Luego se selecciona la bodega de rezago donde quedará físicamente y lógicamente, con este procedimiento más la supervisión del encargado del inventario, el artículo queda dado de baja formalmente.

Codigo Barra		2	-	<input type="text"/>	<input type="button" value="Buscar"/>	<input type="button" value="Borrar"/>
Baja N°		<input type="text" value="buscar"/>				
Fecha actual		<input type="text" value="2005-12-26"/>				
ACTUALMENTE						
Codigo Barra	2 -	<input type="button" value="Buscar"/>	<input type="text" value="Buscar"/>			
Estado	<input type="button" value="Buscar"/>					
Establecimiento	<input type="button" value="Buscar"/>					
Servicio	<input type="button" value="Buscar"/>	<input type="text" value="Buscar"/>				
Dependencia	<input type="button" value="Buscar"/>	<input type="text" value="Buscar"/>				
DESTINO						
Establecimiento	<input type="text" value="0953 Hosp. Temuco"/> <input type="button" value="v"/>					
Servicio	<input type="text" value="Seleccione el servicio"/> <input type="button" value="v"/>					
<input type="button" value="Dar Baja"/>						

Figura 39. Baja de Bienes

El módulo interno que realiza la baja es `haceBaja.php`

```
$cambio=" UPDATE bienes
```

```
SET
```

```
    codigo_servicio= '$nuevoSer',
```

```
    codigo_dependencia= '$nuevaDep',
```

```
    ult_fecha_traspaso= '$fecha'
```

```
where codigo_barra = '$codB' ";
```

Al igual que el módulo anterior, esto se registra en la tabla de movimientos

donde se almacenan todos los datos anteriores a la baja.

3.4.13.6 Módulo Ver Movimientos de Bienes

Este consiste en consultar la tabla de movimientos para buscar alguno en particular o verlos todos: Para esto se utiliza el formulario que se presenta, el cual permite búsquedas, por número de traspaso, fecha movimiento, código de barra, estado, establecimiento, servicio y dependencia. Sino se especifica nada, mostrará todos los movimientos almacenados.

MOVIMIENTOS DE BIENES

Busqueda por		Ordenado por
Num Traspaso	<input type="text"/>	<input checked="" type="radio"/>
Fecha Movimiento	Año <input type="text"/> - Mes <input type="text"/> - Día <input type="text"/>	<input type="radio"/>
Codigo barra	<input type="text"/>	<input type="radio"/>
Estado	— <input type="button" value="v"/>	<input type="radio"/>
Establecimiento (Código)	0953 Hosp. Temuco <input type="button" value="v"/>	<input type="radio"/>
Servicio (Código)	— <input type="button" value="v"/>	<input type="radio"/>
Dependencia (Código)	— <input type="button" value="v"/>	<input type="radio"/>

Figura 40. Movimiento de Bienes

Este formulario envía cualquiera de estos códigos que pueda ser ingresado a un módulo que creará la sentencia para consultar en la base de datos, luego se almacenarán en un arreglo el cual se mostrará a través de un tabla de html dentro de una página Web creada dinámicamente que será la que finalmente

verá el usuario como resultado de su consulta.

3.4.13.7 Módulo Ver Baja de Bienes

Esto es análogo al módulo anterior con la diferencia que se usará el Número Solicitud Baja como clave primaria.

BAJAS DE BIENES

Busqueda por		Ordenado por
Num Solicitud Baja	<input type="text"/>	<input checked="" type="radio"/>
Fecha Baja	Año <input type="text"/> - Mes <input type="text"/> - Día <input type="text"/>	<input type="radio"/>
Codigo barra	<input type="text"/>	<input type="radio"/>
Estado	— <input type="button" value="v"/>	<input type="radio"/>
Establecimiento (Código)	0953 Hosp. Temuco <input type="button" value="v"/>	<input type="radio"/>
Servicio (Código)	— <input type="button" value="v"/>	<input type="radio"/>
Dependencia (Código)	— <input type="button" value="v"/>	<input type="radio"/>

Figura 41. Baja de Bienes

3.4.13.8 Módulo Búsqueda de Servicio

Esto se creó con la finalidad de encontrar un servicio ya sea teniendo el nombre del servicio o algún nombre que se refiera a él o por el código para saber de que servicio se trata. En el formulario se ingresa uno de los dos datos el cual se pasa al módulo que realiza la búsqueda en si. Éste hace la consulta la base de datos utilizando los parámetros que le entrega el formulario. El resultado lo guarda lo guarda en un array y lo luego lo muestra en una página Web creada

en forma dinámica.

Consulta por		Ordenado por
Codigo de servicio	<input type="text"/>	<input checked="" type="radio"/>
Nombre del servicio	<input type="text"/>	<input type="radio"/>
Rut del encargado	<input type="text"/>	<input type="radio"/>

Figura 42. Búsqueda de Servicios

3.4.13.9 Módulo Búsqueda de Dependencias

Este módulo es análogo al anterior, o sea ayudará a buscar una dependencia por su nombre o por su código utilizando una consulta a la base de datos en la tabla de dependencias. Esto lo hace en BusqDep.php.

Consulta por		Ordenado por
Codigo de dependencia	<input type="text"/>	<input type="radio"/>
Codigo de servicio	<input type="text"/>	<input checked="" type="radio"/>
Nombre de dependencia	<input type="text"/>	<input type="radio"/>
Rut del encargado	<input type="text"/>	<input type="radio"/>

Figura 43. Búsqueda de Dependencias

3.4.13.10 Módulo Servicios y Dependencias

Este módulo muestra los servicios con su respectivas dependencias leyéndolas directamente de las tablas servicios y dependencias de la base de datos, para esto se utiliza una consulta select que se encarga de recoger toda esta información.

Servicio	3430 SERVICIO CLIN. TRAUM. Y ORTOPEDIA ADULTO
Dependencia	05009 SALA HOSPITALIZACION

Figura 44. Servicios y sus dependencias

3.4.13.11 Módulo Agregar Dependencias

Este módulo es capaz de agregar una nueva dependencia a la base de datos la cual está vinculada a un servicio específico, por lo tanto, una dependencia no puede pertenecer a dos servicios, por el contrario un servicio puede tener “n” dependencias. Para crearla se debe ingresar un código el cual debe ser creado e ingresado por el encargado de inventario procurando llevar un orden correlativo respecto a las demás, además deberá ingresar una denominación que identificará la dependencia. También deberá seleccionar de la lista un servicio al cual pertenecerá dicha dependencia. Por último y en forma optativa, se puede ingresar un nombre o RUT de la persona encargada a modo de tener más información acerca de esa dependencia.

Codigo Dependencia	<input type="text"/>
Denominacion	<input type="text"/>
Codigo Servicio	1941 SECCION COMPUTACION <input type="button" value="v"/>
Encargado	<input type="text"/>

Figura 45. Ingreso de dependencia

Para poder ingresar a una dependencia primero se debe comprobar que ese código de dependencia no exista ya en la base de datos, esto se realiza con un select, una vez verificado esto se puede agregar una entrada más a la tabla de dependencias.

```
$$SQL1 = (" INSERT INTO  
dependencia(codigo_dependencia,descripcion,codigo_servicio,rut_encargado,rut_origen  
,fecha_creacion,fecha_modificacion)  
  
VALUES('$codigo','$descripcion','$codigo2','$encargado','$rut','$fecha','$fecha') ");
```

3.4.13.12 Módulo Agregar Servicios

Se debe ingresar un código que será entregado por el Hospital en el caso que se cree un nuevo servicio. También se ingresa una denominación para referenciar el servicio. En forma optativa se ingresa el nombre o RUT de una

persona que será el encargado. Estos datos son enviados a otro módulo que recibe estas variables la cual comprueba que no exista otro código igual en la tabla de servicios. Si se verifica esto se podrá ingresar el nuevo servicio.

Codigo Servicio	<input type="text"/>
Denominacion	<input type="text"/>
Encargado	<input type="text"/>

Figura 46. Ingreso de Servicio

3.4.13.13 Módulo Crear Usuarios

Para crear usuarios se deberá ingresar RUT, nombres, apellidos, cargo y password de acceso. Los datos de la persona son corroborados contra la base de datos, sino existe se agregará y si ya existe se enviará un mensaje de error. Para esta tabla se usará el RUT como clave primaria.

Rut	<input type="text"/>
Nombres	<input type="text"/>
Apellidos	<input type="text"/>
Cargo	<input type="text"/>
Password de acceso	<input type="text"/>

Figura 47. Ingreso de Usuario

3.4.13.14 Módulo Modificar Dependencias

Este módulo es análogo al módulo crear dependencias. Aquí se ingresa el código de dependencia que se desea editar, este código se lo entrega a una función del módulo en forma interna, el cual procesa esta información y si existe devuelve las características de la dependencia para ser modificadas.

Modificar Dependencia

Características de la Dependencia	
Codigo	<input type="text" value="Buscar"/>
Descripcion	<input type="text" value="Buscar"/>
Servicio al que pertenece	<input type="text" value="Buscar"/>
Rut de Encargado	<input type="text" value="Buscar"/>
Fecha de creacion	<input type="text" value="Buscar"/>

Figura 48. Modificación de dependencia

3.4.13.15 Módulo Modificar Servicio

Este en forma análoga a modificar dependencias, se le entrega el código del servicio a una función que la procesa y entrega los datos concernientes al servicio para modificarlos.

Modificar Servicio

Buscar

Características del Servicio	
Codigo	Buscar <input type="text"/>
Descripcion	Buscar <input type="text"/>
Rut de Encargado	Buscar <input type="text"/>
Fecha de creacion	Buscar <input type="text"/>

Figura 49. Modifica Servicio

3.4.13.16 Módulo Modificar Usuario

Aquí se ingresa el RUT del usuario que se desea modificar el cual es pasado a una función que va buscar la información el la base de datos devolviendo todos los datos de este.

Ingrese rut:

Buscar

Características del Usuario	
Rut	<input type="text"/>
Nombre	<input type="text"/>
Apellido	<input type="text"/>
Cambio de password	<input type="text"/>
Cargo	<input type="text"/>
Fecha de creacion	<input type="text"/>

Figura 50. Modifica Usuario

3.4.13.17 Módulo Eliminación de Dependencias

Se elige de una lista desplegable el código de la dependencia que se desea eliminar por alguna razón la cual es pasada a una función interna que chequea si existe y trae sus datos al formulario para presentarlos al usuario. Ahora se podrá eliminar sólo si está vacía, el sistema no permitirá eliminar dependencias que contengan bienes. Para esto primero se debe ir al módulo eliminar bienes y una vez que se hayan eliminado todo de la dependencia, se podrá eliminar esta.

Dependencia	
	Buscar
Características de la Dependencia	
Codigo	Buscar
Descripcion	Buscar
Servicio al que pertenece	Buscar
Rut de Encargado	Buscar
Fecha de creacion	Buscar
Eliminar	

Figura 51. Elimina dependencia

3.4.13.18 Módulo Eliminación de Servicio

En forma análoga a eliminar dependencias, se elige de una lista desplegable el servicio que se va a eliminar. Una vez realizado esto, una función devuelve al

formulario la información obtenida desde la base de datos, en este caso se produce la misma situación anterior: el servicio no debe contener dependencias las que a su vez no deben tener bienes. Entonces para eliminar un servicio primero se deben eliminar todos los bienes de cada una de las dependencias del servicio y entonces se estará en condiciones de eliminar un servicio.

Servicio a eliminar 1941 SECCION COMPUTACION

Características del Servicio	
Codigo	<input type="button" value="Buscar"/>
Descripcion	<input type="text" value="Buscar"/>
Rut jefe Servicio	<input type="button" value="Buscar"/>
Fecha de creacion	<input type="button" value="Buscar"/>

Figura 52. Elimina Servicio

3.4.13.19 Módulo eliminar usuario

Para eliminar un usuario se escoge de una lista desplegable uno de los usuarios existentes en el sistema. Esta lista se lee directamente desde la tabla de usuarios la cual contiene toda la información acerca de ellos. Una vez elegido se muestra por pantalla información anexa al usuario para luego proceder a eliminarlo.

Usuario a eliminar 13964203-1 Christian Parra

Características del usuario	
Rut	<input type="text" value="buscar"/>
Nombre	<input type="text"/>
Apellido	<input type="text"/>
Cargo	<input type="text" value="Buscar"/>
Fecha de creacion	<input type="text" value="Buscar"/>

Figura 53. Elimina usuario

3.4.13.20 Módulo Limpiar archivos

Cada vez que se generan hojas murales una copia de este queda en el servidor, lo cual si se suma en el tiempo da una gran cantidad de archivos que no tienen ninguna utilidad y consumen espacio del disco, por ende se crea este módulo que elimina los archivos residuales ejecutando un CGI, este CGI esta escrito en lenguaje Perl y se invocará cada vez que se quiera hacer un limpieza de este tipo de archivos.

Limpiar

Nombre Archivo	Fecha creacion	Espacio bytes	Descargar
hoja_mural_Serv_1941.rtf	26-Dec	267886	hoja_mural_Serv_1941.rtf
listado_02018_13964203-1.rtf	26-Dec	4026	listado_02018_13964203-1.rtf
listado_10008_13964203-1.rtf	26-Dec	1975	listado_10008_13964203-1.rtf
listado_10029_13964203-1.rtf	26-Dec	1965	listado_10029_13964203-1.rtf

Figura 54. Elimina archivos temporales

3.5 PRUEBAS

Cuando ya se ha terminado de programar, el sistema se debe verificar su correcto funcionamiento y depuración de errores que no fueron detectados en la etapa de diseño, para esta etapa se introducirán algunos valores determinados y ver si produce la salida esperada.

En este nivel e puede depurar en dos áreas:

Aplicación: esta área es donde podría producirse la mayoría del los problemas ya que es la parte que tiene mas entrada de datos, respecto a estos últimos existen 3 grupos: alfabéticos, numéricos, alfanuméricos. Estas validaciones se hacen tanto en el cliente como en el servidor, por ejemplo a nivel de cliente se puede validar que un campo obligatorio no este vacío, y a nivel de servidor, podría ser que alguna variable PHP no este vacía.

Base de datos: aquí se pueden producir problemas en la sintaxis SQL,

haciendo, por ejemplo, referencias a tablas inexistentes, campos con distinto formato al ingresado, etc. También puede producir problemas el paso de variables de PHP a MySQL que no fueron validadas correctamente antes de ser usada como parámetro de una consulta SQL.

3.5.1 LOGIN USUARIO

Aquí se valida con una consulta SQL al servidor para comparar que coincidan los datos y así permitir la entrada al sistema. Usando un Rut válido permite que se ingrese a la siguiente interfaz, sino envía el siguiente mensaje de error:

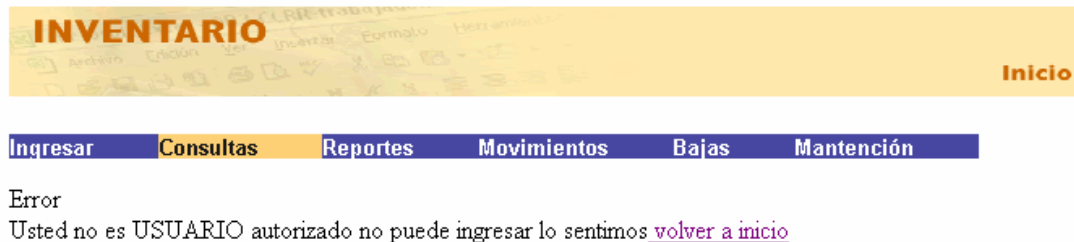


Figura 55. Error login usuario

3.5.2 CAMPOS OBLIGATORIOS

También se validan algunos campos que en la base de datos eran obligatorios por ende no pueden ir en blanco, se ha probado tratar de ingresar un bien sin cumplir estos requisitos pero exitosamente el sistema lo rechaza como se ve en la figura.

INGRESO DE NUEVOS BIENES

Número Antiguo Inventario	2-	<input type="text"/>	Ma	<input type="text"/>	0000	
Nombre Especie		<input type="text"/>	Ma	<input type="text"/>		
Descripción		<input type="text"/>	Se	<input type="text"/>	00	
Estado	Bueno(B)	<input type="text"/>	Medida	<input type="text"/>	Proveedor	<input type="text"/>
Propiedad	Hospital(H)	<input type="text"/>	Vida Útil	<input type="text"/>	Servicio Técnico	<input type="text"/>

Microsoft Internet Explorer ✖

 Escriba el campo Nombre Especie

Figura 56. Error campo obligatorio

3.5.3 BÚSQUEDA PARA PROCESO

Cuando se quiere hacer algún proceso ya sea sobre un bien, una dependencia, o un servicio, se llama a una función la que consulta a la base de datos si esta entidad existe en ella, antes de aceptar los parámetros a procesar, con esto a la vez de validar la consulta también, en caso de encontrarse el valor, se mostrara la información referente a ella como una descripción algún código asociado, fechas, etc.

Movimientos Bajas Mantención

Movimiento de Bienes



Figura 57. Error En proceso

3.5.4 CONSULTA DE BIEN

En el ámbito de las consultas también se puede encontrar una consulta SQL hacia la base de datos, si no encuentra el valor buscado mostrara un mensaje.



Figura 58. Error Búsqueda registro

3.5.5 INGRESO DEPENDENCIA

Cuando se quiere hacer un nuevo ingreso de dependencias también se debe validar que ese código de dependencia no se encuentre repetido en la tabla de dependencias para evitar inconsistencia de datos, si ya existe deberá avisar con un mensaje y no debe permitir que grabe el información en la base de datos.

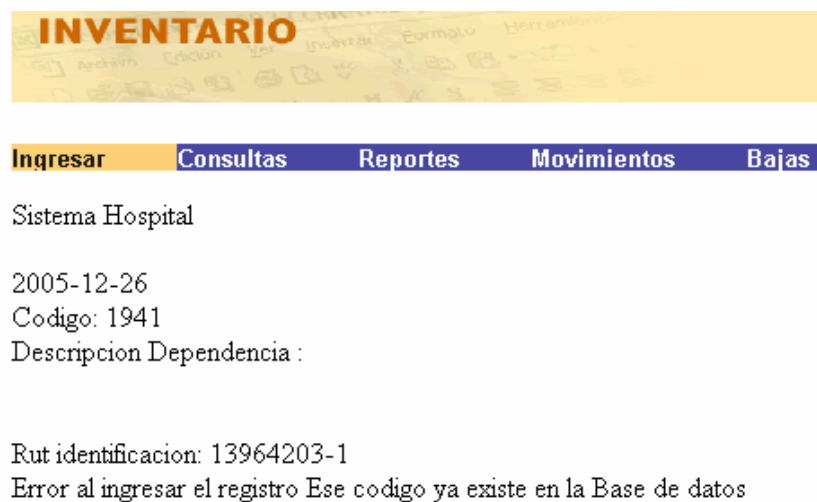


Figura 59. Error ingreso dependencia

4 CONCLUSIONES

- El resultado final entrega un Software con las siguientes características: Escalable, Mejorable, Multiusuario, Tolerante a fallos, Robusto, Rápido, Eficiente. Su desarrollo en forma modular hace más fácil su depuración, modificación, lectura de código, etc.
- Se ha aprovechado las ventajas de la programación Web como traspaso de variables por URL, links, etc.
- El uso de lenguajes interpretados como PHP, HTML, JavaScript además de la gran información que existe en Internet acerca de ellos, facilitan su depuración y visión global.
- Debido a la gran cantidad de información que puede almacenar el Sistema, se está estudiando la posibilidad de enlazar la Base de Datos de inventario con la de mantenimiento ya que ellos llevan su propio catastro de equipos el cual puede completarse entregando más información.
- Una desafío pendiente desde el punto de vista técnico es la de migrar el Sistema a un nuevo servidor con Oracle para entregarle una mejor performance al procesamiento de datos.

- Se tiene considerado complementar este sistema con un dispositivo PDA como cliente, el cual podrá capturar el Código de Barra de un bien y hacer las consultas necesarias a la base de datos estando en terreno.

5 BIBLIOGRAFÍA

Ivar Jacobson, Grady Booch, James Rumbaugh. RATIONAL SOFTWARE CORPORATION.

AÑO 2000

Proceso Unificado de Desarrollo

PRIMERA EDICION.

“Los flujos de trabajo fundamentales” Pág. 105-299.

Madrid, España.

PHP Group.

Thies C. Arntzen, Stig Bakken, Shane Caraveo, Andi Gutmans, Rasmus Lerdorf, Sam Ruby,

Sascha Schumann, Zeev Suraski, Jim Winstead, Andrei Zmievski

2005

“VI. Referencia de funciones”

<http://www.php.net>

MySQL AB

2005

“Download and Documentation”

<http://www.mysql.com>

Jimmy Wales, Larry Sanger

2005

“PHP”, “Apache”, “MySQL”, “Modelo Cliente-Servidor”, “Aplicaciones Web”.

<http://www.wikipedia.org>

Michael Pham

2005

“PHP Designer 2005”

<http://www.mpsoftware.dk>

Fernando Ortiz

2005

“Configuración de Apache/PHP/Informix's Setup”

<http://www.lacorona.com.mx/fortiz/informix/>