



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

SISTEMA DE CONTROL DE INVENTARIOS Y CENSO DE EQUIPOS DE CÓMPUTO DE LA FACULTAD DE INGENIERÍA (SICICE)

T E S I S

QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN

P R E S E N T A N

OSCAR JOSAFAT GASCÓN BUSIO
NÉLIDA VIVIANA VELÁZQUEZ SEDÁN

DIRECTORA DE TESIS

ING. MARÍA DEL ROSARIO BARRAGÁN PAZ

Ciudad Universitaria, México D.F. 2012





Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

Oscar:

Gracias, de corazón. Nunca es fácil escribir esta parte y más cuando tanta gente ha contribuido de diferentes maneras a realizar este trabajo.

Gracias a mi madre y a mi hermana, porque su presencia ha sido y será siempre el motivo que me ha impulsado, han sido mi motor y mi red de seguridad.

Agradezco todos los esfuerzos en los momentos más difíciles y más felices, el objetivo logrado también es tuyo y la fuerza que me ayudo a conseguirlo fue tu apoyo mama.

Gracias Mili por tu comprensión, tu apoyo y por compartir todo este tiempo lleno amor y de cosas que jamás imaginamos.

Gracias por mostrarme que para el verdadero amor nunca hay obstáculos. Y recuerda que: "Si no me hubieras encontrado tú a mí, te habría encontrado yo a ti..."

Nelida:

Gracias a mi familia por su apoyo incondicional, por ser mi principal fuente de inspiración, gracias mamá, papá, Claudia, Miriam, Brenda y Paulina.

Gracias a mi querida Universidad porque haber estado en sus aulas y recorrido sus pasillos y jardines ha sido lo mejor que me ha sucedido en la vida, en ningún lugar nunca me he sentido tan a salvo, tan feliz.

Gracias a la Unidad de Servicios de Cómputo Académico por ser una pieza clave en mi formación como profesional.

Gracias a cada uno de mis amigos y compañeros de universidad porque no solo con ellos aprendí sino porque además de ellos aprendí.

Gracias con mucho cariño.

INTRODUCCIÓN	1
CAPÍTULO 1. DEFINICIÓN DEL PROYECTO	
1.1. Antecedentes	4
1.2. Análisis y problemática actual	6
1.3. Objetivos	11
1.3.1. <i>Objetivo general</i>	11
1.3.2. <i>Objetivos específicos</i>	11
1.4. Alcance	11
1.5. Requerimientos por parte de las divisiones	13
1.6. Infraestructura	17
1.6.1. <i>Departamento de Investigación y Desarrollo</i>	17
1.6.2. <i>Infraestructura de Hardware</i>	17
1.6.3. <i>Infraestructura de Software</i>	18
1.7. Beneficios	18
CAPÍTULO 2. MARCO TEÓRICO	
2.1. Ingeniería de Software	20
2.1.1. <i>Definición</i>	21
2.1.2. <i>El Proceso</i>	22
2.1.3. <i>Modelos</i>	24
2.1.3.1. <i>Modelo en Cascada</i>	25
2.1.3.2. <i>Modelo de Desarrollo Evolutivo</i>	27
2.1.3.3. <i>Ingeniería de Software Basada en Componentes</i>	28
2.1.4. <i>Requerimientos</i>	31
2.2. Software Libre	33
2.2.1. <i>¿Qué es software libre?</i>	34
2.2.2. <i>La filosofía de este proyecto</i>	34
2.3. Herramientas de Desarrollo	35
2.3.1. <i>Java</i>	35
2.3.2. <i>PostgreSQL</i>	37
2.3.3. <i>Apache Tomcat</i>	38
CAPÍTULO 3. DESARROLLO	
3.1. Necesidades del sistema	43
3.2. Requerimientos del sistema	44
3.2.1. <i>Requerimientos funcionales del sistema</i>	44
3.2.2. <i>Requerimientos no funcionales del sistema</i>	47
3.3. Análisis del sistema	49
3.3.1. <i>Definición del alcance del sistema</i>	49
3.3.2. <i>Operaciones del sistema</i>	50
3.3.2.1. <i>Altas</i>	50

3.3.2.2. <i>Bajas</i>	51
3.3.2.3. <i>Modificaciones</i>	52
3.3.2.4. <i>Movimientos</i>	53
3.3.2.5. <i>Reportes</i>	54
3.3.3. <i>Casos de uso</i>	56
3.3.4. <i>Diagrama de casos de uso</i>	65
3.4. <i>Diseño basado en componentes</i>	66
3.4.1. <i>Componentes del sistema</i>	66
3.4.2. <i>Diseño de los componentes</i>	70
3.4.3. <i>Flujo de pantallas</i>	71
3.4.4. <i>Diagramas de secuencias</i>	72
3.4.5. <i>Diccionario de datos</i>	73
3.4.6. <i>Tecnologías de programación a implementar</i>	76
3.4.6.1. <i>Java EE</i>	76
3.4.6.2. <i>Javascript</i>	78
3.4.6.3. <i>XML</i>	78
CAPÍTULO 4. PRUEBAS	
4.1. <i>Objetivos de las pruebas</i>	80
4.2. <i>Diseño de casos de prueba</i>	81
4.2.1. <i>Pruebas funcionales</i>	84
4.2.2. <i>Pruebas estructurales</i>	84
4.2.3. <i>Formato de casos de prueba</i>	84
4.3. <i>Técnicas de pruebas de caja blanca</i>	86
4.3.1. <i>Prueba del camino básico</i>	86
4.3.2. <i>Prueba de condición</i>	87
4.3.3. <i>Prueba de bucles</i>	88
4.4. <i>Técnicas de pruebas de caja negra</i>	89
4.4.1. <i>Partición equivalente</i>	90
4.5. <i>Casos de Prueba</i>	90
CAPÍTULO 5. IMPLEMENTACIÓN	
5.1. <i>Manual de Usuario</i>	120
5.2. <i>Manual de Instalación</i>	154
CONCLUSIONES	158
ANEXOS	160
BIBLIOGRAFÍA Y REFERENCIAS	178

INTRODUCCIÓN

A través de los años la necesidad de contabilizar y mantener el control de los bienes de las Organizaciones ya sean públicas o privadas ha incitado no sólo a la creación de métodos para realizar esta tarea, sino de toda una teoría que debido a la naturaleza de esta tesis sólo se dará un breve panorama de su origen y definición; la teoría a la que nos referimos son *Los Inventarios*.

“Un inventario consiste en la existencia de productos físicos que se conservan en un lugar y momento determinado.”

La eficiente gestión de los inventarios no sólo implica la implantación de las medidas que son necesarias para mantener su seguridad y control administrativo-contable con el propósito de preservar su integridad física ante los riesgos propios de su operación (por ejemplo, robos, deterioro, etcétera), sino que además debe cumplir tres condiciones básicas:

- Garantizar a los clientes la calidad del servicio deseado.
- Mantener en los niveles más bajos posibles el capital inmovilizado en inventarios.
- Gestionar la función con los menores costos para la empresa.

Anteriormente la información que se podía obtener de los inventarios era pobre y poco precisa, sin embargo los avances en las tecnologías de la información han cambiado drásticamente las posibilidades de aplicar eficientes técnicas para el control de inventarios lo que permite la optimización de recursos y la correcta toma de decisiones.

La implementación de un sistema de inventarios en la Facultad de Ingeniería facilitará la toma de decisiones en cuanto al control, compra, asignación y distribución de los equipos de cómputo en las distintas divisiones que conforman a la Facultad y a su vez detectar puntos de oportunidad.

En este trabajo se expone el proceso que se llevó a cabo para el desarrollo del Sistema de Inventarios de Equipos de Cómputo y Censo de la Facultad de Ingeniería (SICICE) que basado en las premisas expuestas anteriormente pretende mejorar y facilitar el inventario de cómputo de la Facultad.

El contenido se divide en los cinco capítulos siguientes:

Capítulo I Definición del proyecto. Este capítulo plantea los antecedentes del proyecto, es decir, el contexto de la Institución para la cual fue diseñado este sistema, tomando en cuenta lo anterior se analiza la problemática y se presentan los primeros requerimientos para establecer un objetivo y alcances del proyecto. Asimismo, se describen los beneficios que representa este sistema.

Capítulo II Marco teórico. Presenta las bases teóricas en las cuales se fundamenta el desarrollo del sistema, empezando por la definición de Ingeniería de Software y algunos conceptos

Introducción

relacionados como modelos, validación y verificación de software y finalmente se exponen temas de herramientas de desarrollo para aplicar estas teorías.

Capítulo III Desarrollo. Describe el proceso seguido para el diseño y desarrollo del sistema, requerimientos funcionales y no funcionales del mismo con base a los cuales se realiza un profundo análisis de las necesidades y se presenta el modelo del sistema desde varias perspectivas utilizando el Lenguaje de modelado UML. Finalmente, se explican las técnicas de programación que se consideraron más adecuadas para la implementación del modelo presentado.

Capítulo IV Pruebas de Software. Detalla diferentes tipos de validaciones de software sobre los cuales se diseñaron los casos de prueba, se presenta el proceso y formato para aplicación de éstas.

Capítulo V Implementación. Finalmente este capítulo trata temas de suma importancia ya que contiene los manuales de instalación y de usuario que serán útiles para el soporte y mantenimiento del sistema.

1

DEFINICIÓN DEL PROYECTO

Este primer capítulo presenta un panorama general de las Instituciones involucradas en el proyecto, su descripción y principales funciones; se pone especial énfasis en su estructura organizacional ya que este será un punto fundamental para el planteamiento de la solución.

Se estudia la problemática a la que se enfrenta actualmente la Facultad de Ingeniería, en cuanto al proceso de realizar el censo e inventario de sus bienes de cómputo.

En base a estos dos puntos primordiales se establece el objetivo y los alcances del proyecto, así mismo se definen los requerimientos de las dependencias de la Facultad y la infraestructura con la que se cuenta para llevarlo a cabo. Finalmente se describen los beneficios que se esperan alcanzar con este desarrollo.

Contenido:

- 1.1 Antecedentes
- 1.2 Análisis y problemática actual
- 1.3 Objetivos
- 1.4 Alcance
- 1.5 Requerimientos por parte de las divisiones
- 1.6 Infraestructura
- 1.7 Beneficios

Capítulo 1

Definición del Proyecto

1.1 Antecedentes

La Facultad de Ingeniería es una Institución pública de educación superior que se encarga de formar de manera integral recursos humanos en Ingeniería, realizar investigación acorde con las necesidades de la sociedad, y difundir ampliamente la cultura nacional y universal.

Actualmente está organizada de la siguiente manera (Figura 1.1):

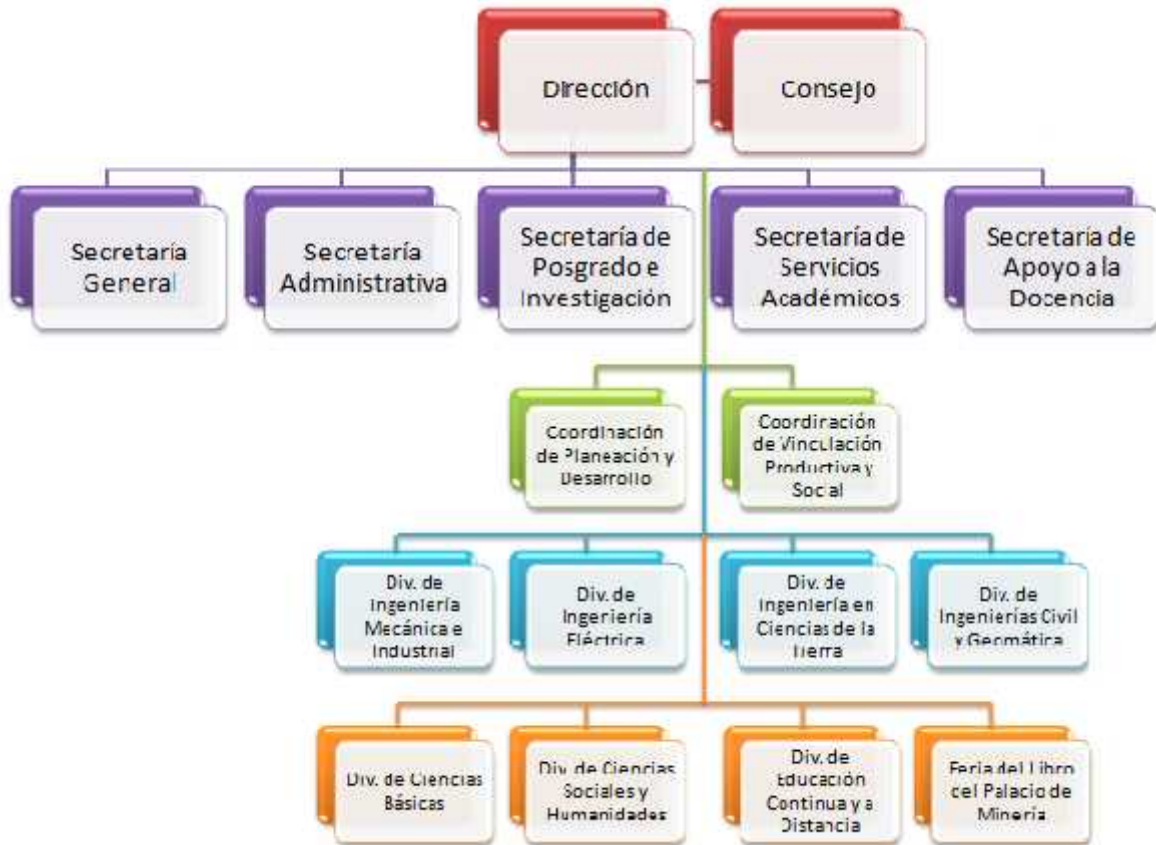


Figura 1.1 Estructura organizacional de la Facultad de Ingeniería

Además de las Secretarías, Coordinaciones y distintas Divisiones con las que cuenta la Facultad, existen otras dependencias que son un importante apoyo para las actividades administrativas, académicas y de soporte en tópicos de cómputo, tal es el caso de la Unidad de Servicios de Cómputo Académico (UNICA).

En 1994 surgen, como resultado de la división del Centro de Cálculo, la Unidad de Servicios de Cómputo Académico y la Unidad de Servicios de Cálculo Académico con el objetivo de llevar a cabo las tareas académicas y administrativas de la Facultad de Ingeniería.

Capítulo 1

Definición del Proyecto

Los objetivos de la Unidad de Servicios de Cómputo Académico son proporcionar a nivel institucional, los servicios de apoyo en cómputo que los alumnos de la Facultad requieren para la realización y cumplimiento eficaz de sus tareas sustantivas, formar recursos humanos de calidad, tanto en el área de cómputo como el desempeño de la vida profesional. Ofrecer a la comunidad de la Facultad capacitación en lo relativo a tópicos de cómputo. Esta capacitación se divide en cursos de capacitación para el personal académico y cursos complementarios para alumnos.

Mantener a la vanguardia en todo lo relativo al cómputo a través de la investigación e incorporar a la actividad de la Facultad nuevas tendencias de cómputo.

Apoyar a la Secretaría General en las actividades que involucren institucionalmente a la Facultad de Ingeniería.

La Unidad de Servicios de Cómputo Académico está organizada por departamentos y cuenta con la siguiente estructura (Figura 1.2):



Figura 1.2 Estructura organizacional de la Unidad de Servicios de Cómputo Académico

Cada una de estas dependencias desde la Dirección pasando por las Secretarías, Coordinaciones y Divisiones hasta llegar a cada uno de los departamentos y salas de cómputo de UNICA tienen asignados para la realización de sus respectivas funciones determinado número de bienes de cómputo.

Actualmente la Facultad cuenta con alrededor 3560 de equipos de cómputo distribuidos en cada una de sus divisiones, secretarías, coordinaciones y otras dependencias.

Cada año se realiza en la Facultad de Ingeniería el censo, proceso a través del cual se hace una revisión y actualización de los datos de los bienes de cómputo con los que cuentan cada una de

Capítulo 1

Definición del Proyecto

estas dependencias. Los tipos de bienes que se toman en cuenta para dicho censo son los siguientes:

- Access Point
- Cámara Digital
- Concentrador
- Modem de Fibra Óptica
- Plotter
- Router
- Monitor
- Mouse
- Teclado
- Unidad de CD externa
- Impresora
- Escáner
- Disco Duro Externo
- Regulador
- Hub
- Switch
- No-Break
- Proyector
- Computadoras (PC, Portátiles y MAC)

El número de inventario de la UNAM con el que cuentan cada uno de estos bienes es la base para el control del registro. El número de inventario consta de 7 dígitos, sin embargo a partir del año 2009 los bienes cuentan con inventarios alfanuméricos.

1.2 Análisis y problemática actual

En un principio el proceso de censo era atendido de forma manual; situación que obligaba a las dependencias a invertir tiempo en esta actividad, lo cual tenía como consecuencia una mayor probabilidad de retrasos, omisiones, duplicidad de trabajo y datos.

No existía una convención para un formato estándar por lo que cada dependencia realizaba los censos con los datos que creía más convenientes, lo que ocasionaba dificultades al momento de integrar la información de las distintas dependencias de la Facultad.

A partir del año 2002 se liberó la primera versión del Sistema de Control de Inventarios (SICI), una aplicación WEB que permitía a los usuarios acceder a través de un nombre de usuario y contraseña. A cada dependencia le era asignado el login y contraseña que le permitía manipular los datos que correspondían únicamente a esa dependencia.

En este sistema se podían realizar los siguientes procesos:

- *Ver los bienes existentes.* Cada división tenía la posibilidad de ver listados los bienes que tenía bajo su custodia.
- *Dar de alta un bien.* A través de este sistema se podía agregar los datos de un bien a la base de datos. Algunos de estos datos eran:
 - Número de inventario

Capítulo 1

Definición del Proyecto

- Número de serie
 - Número de resguardo
 - Responsable del equipo
 - Marca
 - Modelo
- *Dar de baja un bien.* Cuando un bien quedaba inutilizable era posible dar de baja sus datos. Esto quería decir que el bien quedaba sin responsable y no aparecía en la lista de inventario activo de ninguna de las divisiones, sin embargo los demás datos eran conservados en la base de datos para su registro en casos de auditorías.
 - *Modificar los datos de un bien.* Esta sección permitía cambiar un bien, ya sea de responsable dentro de la misma división que tenía la custodia o de otras divisiones.

Como se ha comentado este sistema se apoyaba de una base de datos relacional para el almacenamiento de los datos tanto de los bienes como de los usuarios del sistema. (Figura 1.3)

Esta base de datos contaba con 19 tablas que permitían almacenar y gestionar toda la información del SICI. El diseño de la base estaba enfocado en la tabla de *Equipo* con la cual todas las demás tablas tenían relación excepto la tabla de *Equipo_b* que era donde se almacenaban los datos de los equipos que eran dados de baja.

Las tablas: monitor, mouse, scanner, dd_externo, hub, nobreak, teclado, impresora, plotter, cdwr_externo, switch y regulador, son similares, en la tabla 1.1 se describen cada uno de sus campos.

Campo	Descripción
Id.	Identificador único del bien en la base de datos
Inv_unam	Número de siete dígitos utilizado por la UNAM
no_serie	Número de serie del bien
movimiento	Campo que indica si el bien estaba bajo custodia de alguna división para poder ser asignado a otra
fecha_ing	Fecha en que el bien fue dado de alta al sistema
fecha_mov	Si el bien ha sido cambiado de división se indica la fecha de este cambio
marca	Marca del bien
modelo	Modelo del bien
garantía	Campo que indica si el bien contaba con garantía
estado	Campo que indica si el bien está dado de alta o es baja
no_resguardo	Número especial asignado para cada división
bien	Campo que identifica el tipo de bien

Tabla 1.1 Descripción de los campos de tablas similares

Capítulo 1

Definición del Proyecto

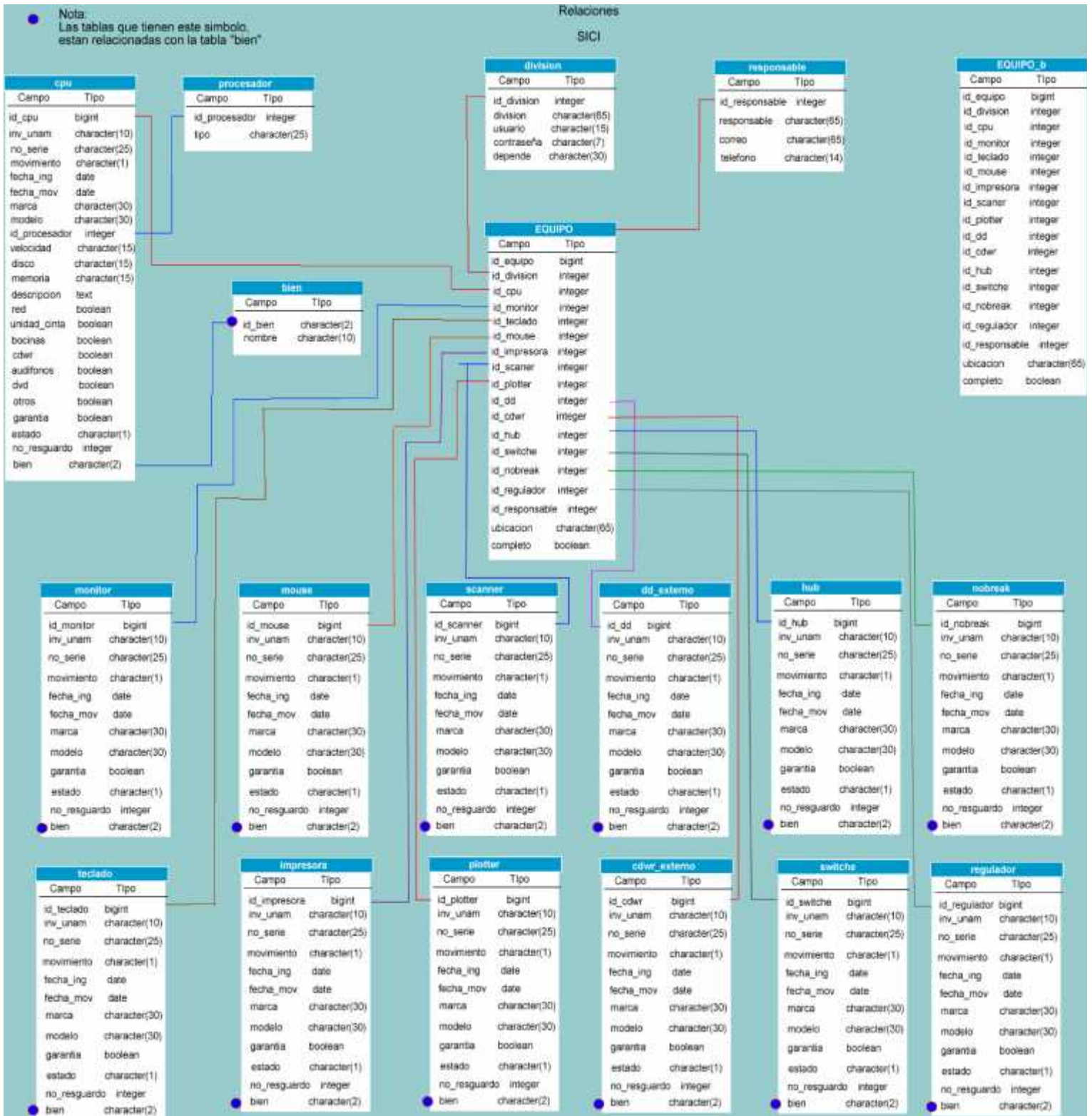


Figura 1.3 Modelo relacional de la base de datos de la primera versión del SICI

Capítulo 1

Definición del Proyecto

La tabla cpu además de contar con estos datos, por su naturaleza deben ser agregados otros campos específicos como los mostrados en la Tabla 1.2

Campo	Descripción
id_procesador	Este es el campo que guarda la relación con la tabla procesador que contiene la lista de procesadores disponibles.
velocidad	Velocidad del procesador en GHz
disco	Capacidad del disco duro
memoria	Capacidad de Memoria RAM
descripción	En este campo se podía indicar se trataba de una PC, Workstation o Portátil
red	Valor booleano que indica si la máquina está en red
unidad_cinta	Valor booleano que indica si cuenta con unidad de cinta
bocinas	Valor booleano que indica si cuenta con bocinas
cdwr	Valor booleano que indica si cuenta con unidad de CD
audifonos	Valor booleano que indica si cuenta con audífonos
dvd	Valor booleano que indica si cuenta con unidad de DVD
otros	Valor booleano que indica si cuenta con algún otro elemento.

Tabla 1.2 Descripción de campos específicos de la tabla cpu

Los campos de las tablas procesador y bien se describen a continuación en las Tablas 1.3 y 1.4 respectivamente. Estas tablas fungían como una especie de catálogo.

Campo	Descripción
id_procesador	Número único que identifica al tipo de procesador
Tipo	Breve descripción del tipo de procesador

Tabla 1.3 Descripción de campos de la tabla procesador

Campo	Descripción
id_bien	Número único que identificaba a un tipo de bien
Nombre	Nombre del tipo de bien.

Tabla 1.4 Descripción de campos de la tabla bien

Otras dos tablas importantes son la de responsable y la de división, la primera almacena los datos de las personas que son designados como los responsables de los bienes y la segunda almacena los datos de los usuarios del sistema. En las tablas 1.5 y 1.6 se describen respectivamente sus campos.

Campo	Descripción
id_responsable	Identificador único de responsable
responsable	Nombre del responsable
Correo	Dirección electrónica del responsable
telefono	Teléfono del responsable

Tabla 1.5 Descripción de campos de la tabla responsable

Capítulo 1

Definición del Proyecto

Campo	Descripción
id_division	Identificador único del usuario
División	Nombre completo de la división
Usuario	Nombre corto que se usa como nombre de usuario
contraseña	Contraseña de usuario
depende	Este campo indica si la división depende de la Secretaría General.

Tabla 1.6 Descripción de los campos de la tabla división

Finalmente el objetivo de la tabla equipo es guardar relación con las demás tablas, la idea es que un equipo puede tener al menos un bien ya sea de la tabla cpu, monitor, mouse, scanner, dd_externo, hub, nobreak, teclado, impresora, plotter, cdwr_externo, switch o regulador. También debe tener un responsable y pertenecer a una división. Los bienes mencionados anteriormente por si solos no tienen un responsable y una división, primero estos deben estar relacionados con la tabla equipo para que tengan un id de equipo. En la Tabla 1.7 se describen los campos de la tabla equipo.

Campo	Descripción
id_equipo	Identificador único del equipo. Varios bienes pueden tener el mismo id de equipo si pertenecen a un mismo registro de ésta tabla
Id_division	Id de la división a la que pertenece este equipo, es la relación guardada con la tabla división
Id_cpu	Identificador de CPU del equipo
Id_monitor	Identificador de Monitor
Id_mouse	Identificador de Mouse
Id_impresora	Identificador de Impresora
Id_scanner	Identificador de Scanner
Id_plotter	Identificador de Plotter
Id_dd	Identificador de Disco duro externo
Id_cdwr	Identificador de Unidad de CD
Id_hub	Identificador de Hub
Id_switche	Identificador de switch
Id_nobreak	Identificador de Nobreak
Id_regulador	Identificador de Regulador
Id_responsable	Identificador del responsable a cargo
ubicacion	Ubicación física del equipo
completo	Valor booleano para indicar si el equipo tiene asignados todos los campos.

Tabla 1.7 Descripción de los campos de la tabla equipo

Este sistema fue el primer paso para dar solución al problema. Uno de sus principales valores fue el hecho de lograr, a través de la base de datos descrita anteriormente, la centralización de la información.

Capítulo 1

Definición del Proyecto

A pesar de que con este paso se solucionaron varios problemas, con el paso del tiempo fueron surgiendo otras necesidades que el sistema ya no fue capaz de satisfacer. A continuación se describen los principales problemas que se lograron detectar en la primera versión de sistema.

- *Nuevos tipos de bienes.* En los últimos años se han adquirido nuevos tipos de bienes que es necesario incluir en el control de inventario, algunos de estos equipos son access point, cámaras digitales y routers.
- *Función para la creación de reportes no integrada al sistema.* Aunque si existe una utilidad para la creación de reportes esta no está integrada al sistema y su instalación y uso son complejos.
- *Creación de reportes poco flexible.* Los reportes que se pueden crear solo son una lista independiente de bajas y altas por división
- *Falta de una interfaz más intuitiva y fácil de manejar.* El actual proceso para la búsqueda y modificación de datos es complicada y confusa. Por otra parte cuando un usuario desea ver la lista de equipos bajo su custodia la información es desplegada de forma poco conveniente para su lectura
- *Repetición de información.* Debido a la falta de correctas validaciones es fácil repetir información como nombres de responsables

1.3 Objetivos

1.3.1 Objetivo general

El objetivo general del proyecto es proporcionar a la Facultad de Ingeniería un sistema que permita automatizar, facilitar, mejorar y mantener actualizado el inventario del equipo de cómputo en tiempo real.

1.3.2 Objetivos específicos

- 1) Controlar las altas, bajas y modificaciones del inventario de equipo de cómputo
- 2) Generar reportes detallados por cada dependencia
- 3) Centralizar la información de los bienes de cómputo
- 4) Realizar respaldos de la información
- 5) Desarrollar la documentación que permita dar soporte y mantenimiento al sistema

1.4 Alcance

Con este proyecto se propone tener un mejor control de la información en cuanto a los equipos de cómputo dentro de la Facultad de Ingeniería.

Capítulo 1

Definición del Proyecto

Para poder realizar lo anterior, se necesita un sistema confiable, fácil de utilizar, que permita ingresar la información necesaria obedeciendo a la dependencia que lo esté usando, para así generar reportes detallados y personalizados.

En las siguientes líneas se encuentra descrito el alcance de este proyecto el cual se desarrolló basándonos en una técnica de gestión de proyectos llamada Estructura de Descomposición de Trabajo, WBS por siglas en inglés (Work Breakdown Structure) que es un modelo jerárquico que proporciona una visión de arriba hacia abajo de las actividades, donde el nivel superior viene dado por el propio proyecto y las hojas corresponden a las actividades o tareas que parten de la división que se realiza en los procesos a alto nivel. Esta estructuración proporciona un punto de vista general del proyecto y de las tareas que lo componen.

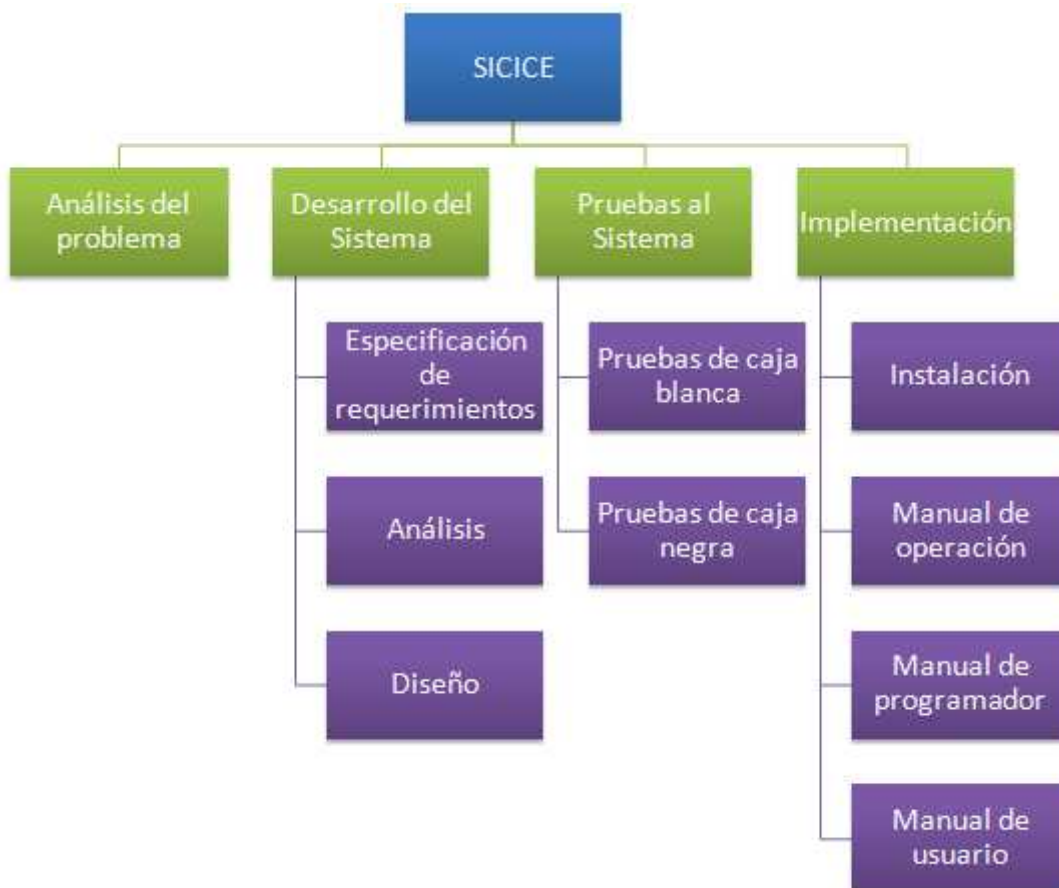


Figura 1.4 Alcances del proyecto

En el diagrama de la Figura 1.4 se observan las actividades que se llevaron a cabo a lo largo del proyecto, este es un primer acercamiento general a la solución del problema por lo que sólo pretende describir los alcances del proyecto más no los del sistema, es decir se describe lo que se va a obtener y lo que no.

Capítulo 1

Definición del Proyecto

En las siguientes líneas se describe con más detalle cada actividad del diagrama:

Análisis del problema. En el primer capítulo de este trabajo se presenta el panorama general de la situación para poder establecer el rumbo del proyecto, obtener un primer acercamiento de lo que se desea alcanzar y los recursos con que se cuentan para llevarlo a cabo.

Desarrollo del Sistema. Es en esta etapa se desarrolla y obtiene lo que en sí será el producto con el cual se le dará solución al problema, es decir, el software, para lo cual se deben seguir una serie de procesos para que este sea de la mejor calidad posible

Pruebas de sistema. Una vez finalizado el producto de software es necesario hacer una revisión y realizar un periodo de pruebas para validar y verificar que cumpla con los requerimientos y expectativas establecidos.

Implementación. Esta es la etapa final del proyecto en la cual se pone en funcionamiento el sistema y se elaboran una serie de manuales para el futuro soporte y mantenimiento del sistema, así como manuales de usuario para la capacitación de personal que opere dicho producto de software.

Para finalizar es importante hacer explícitos algunos puntos que el proyecto no cubre

- El proyecto no contempla la inclusión de otro tipo de equipos o bienes que no sean de cómputo
- El proyecto solo toma en cuenta el inventario a nivel de hardware, es decir no se toman elementos cuantitativos o cualitativos de software de los equipos como direcciones IP, tipos de programas instalados en los equipos etc.

1.5 Requerimientos por parte de las divisiones

En las siguientes tarjetas se muestran de manera estructurada los requerimientos de usuario, éstos son abstractos, de alto nivel por lo que sólo describen el comportamiento externo que se desea que el sistema tenga.

El formato de estas tarjetas consta de los siguientes campos:

- a) El número de requerimiento.
- b) La descripción del requerimiento, marcado en negritas.
- c) El fundamento del requerimiento.
- d) La relación que tiene con otros requerimientos.
- e) La fuente de donde se obtuvo el requerimiento.

Capítulo 1

Definición del Proyecto

No. 1 Acceso vía WEB

Los usuarios deberán poder acceder al sistema a través de internet, esto es a través de una página WEB en un navegador.

Fundamento: Es necesario contar con un sistema al que se pueda acceder en cualquier lugar desde cualquier máquina con el mínimo de requerimientos pues esto ayuda a evitar dependencias de determinadas plataformas o software.

Relación: Ninguna

Fuente: María del Rosario Barragán, Jefa del departamento de Investigación y Desarrollo, UNICA

No. 2 Opción de Altas

El sistema deberá tener una opción en la que se pueda dar de alta un bien, esto es una opción en la que se puedan capturar y almacenar los datos del bien. Los datos que se deben poder guardar son: número de inventario, número de serie, número de resguardo, marca, modelo, fecha de ingreso, si el equipo cuenta con garantía o no, su ubicación física y adicionalmente el nombre, correo y teléfono del responsable a cargo del bien.

Fundamento: Esto ayuda a estandarizar el número y tipo de datos que se guardan acerca de un bien y facilitan la recolección de los datos.

Relaciones: Requerimiento No. 3 y No. 4

Fuente: María del Rosario Barragán, Jefa del departamento de Investigación y Desarrollo, UNICA

No. 3 Opción de bajas

El sistema debe tener una opción en la que se pueda dar de baja un bien, pero siempre conservando los datos del equipo dado de baja

Fundamento: Cuando el equipo se vuelve obsoleto y/o inutilizable debe ser dado de baja, por lo cual es necesario que en el sistema se pueda indicar este hecho, de manera que el bien no aparezca en el censo, y se quite la responsabilidad a la dependencia a cargo, sin embargo es preciso conservar los datos de inventario, número de serie, marca, modelo, número de resguardo y fecha de ingreso para los casos en que haya auditorías.

Relación: Requerimiento No. 2

Fuente: María del Rosario Barragán, Jefa del departamento de Investigación y Desarrollo, UNICA

Capítulo 1

Definición del Proyecto

No. 4 Opción de modificaciones

El sistema debe tener una opción en la que se puedan modificar los datos de los equipos dados de alta.

Fundamento: En algunas ocasiones la ubicación física del equipo o el responsable a cargo cambian por lo que es necesario contar con una opción en el sistema que permita hacer estas modificaciones.

Relación: Requerimiento No. 2

Fuente: María del Rosario Barragán, Jefa del departamento de Investigación y Desarrollo, UNICA

No. 5 Opción de traslados de equipos

El sistema debe permitir que un equipo pueda ser trasladado a otra dependencia, lo que significa que una dependencia puede liberarse de la responsabilidad de un equipo para que otra lo tome a su cargo. Por ejemplo si un equipo ha dejado de ser útil para la División de Ciencias Básicas este podrá ser puesto a disposición para que alguien más lo pueda aprovechar; dicho equipo ya no aparecerá más como responsabilidad de la dependencia que lo dejó de usar.

Fundamento: Es de utilidad para la administración de los equipos que cuando éstos vayan a ser trasladados y ser asignados a otra dependencia quede establecido a través del sistema en donde se encuentra cada bien y quien es el que está a cargo de él para mantener un control exacto de lo que utiliza cada división.

Relación: Requerimiento No. 7

Fuente: María del Rosario Barragán, Jefa del Departamento de Investigación y Desarrollo, UNICA

No. 6 Creación de reportes

El sistema deberá tener una opción con la cual se puedan crear reportes en Excel con los datos de los equipos que tiene asignada cada dependencia.

Fundamento: Es una herramienta de ayuda para que la información pueda ser usada para su presentación o intercambio en los diversos procesos como son los censos o auditorías.

Relaciones: Requerimientos No. 2 y No. 3

Fuente: María del Rosario Barragán, Jefa del Departamento de Investigación y Desarrollo, UNICA

Capítulo 1

Definición del Proyecto

No. 7 Cuentas de acceso

Cada dependencia de la Facultad involucrada en el proyecto deberá contar con un nombre de usuario y contraseña para acceder al sistema.

Fundamento: Cada dependencia de la Facultad debe manipular los datos de sus bienes de manera aislada a las demás, ya que esto facilita el control y evita mezcla de datos. Además agiliza el censo y la clasificación de datos y es más sencillo determinar quién es responsable de cada bien.

Relaciones: Requerimientos No. 2 al No. 6

Fuente: Persiste desde la primera versión del SICI

No. 8 Limitación de privilegios

Dependiendo del tipo de usuario que ingrese al sistema, habilitar las diferentes funciones que se podrán realizar dentro de este.

Fundamento: Es importante que ciertas funciones del sistema, estén bloqueadas u ocultas para algunos usuarios, como es la creación de reportes más detallados y extensos, ya que esta información debe permanecer confidencial para ciertos usuarios y sólo debe ser manejada por personal con autorización.

Relación: Requerimiento No. 7

Fuente: María del Rosario Barragán, Jefa del Departamento de Investigación y Desarrollo, UNICA

No. 9 Documentación

El proyecto deberá contemplar el desarrollo de documentación como son manual de operación, manual de usuario y manual del programador.

Fundamento: Los manuales de operación y de programador ayudarán al mantenimiento y soporte del sistema. El manual de usuario permitirá conocer el correcto manejo del sistema lo que evitará errores y prácticas inadecuadas.

Relación: Todos los requerimientos

Fuente: María del Rosario Barragán, Jefa del Departamento de Investigación y Desarrollo, UNICA

Capítulo 1

Definición del Proyecto

No. 10 Facilidad de uso.

El sistema deberá ser intuitivo y fácil de utilizar.

Fundamento: Un sistema con una interfaz sencilla y que permita identificar fácilmente los elementos del sistema y lo que cada uno de éstos permite hacer ayuda a los usuarios a reducir su curva de aprendizaje y por lo tanto agiliza el proceso de implementación del sistema.

Relación: Requerimientos del No. 1 al No. 8

Fuente: Noé Cruz Marín, Jefe del Departamento de Operación de Servidores, UNICA

1.6 Infraestructura

Para el desarrollo es necesario contar elementos de hardware y software específicos así como personal capacitado que realizará cada una de las actividades en el proceso, la Facultad de Ingeniería a través de la Unidad de Servicios de Cómputo Académico proporcionará la infraestructura necesaria para llevar a cabo dicho proyecto.

1.6.1 Departamento de Investigación y Desarrollo

Específicamente el Departamento de investigación y Desarrollo (DID) es el encargado de desarrollar sistemas de información, administración y control de apoyo a las actividades académico-administrativas, internas de la Unidad, de la Facultad y eventualmente para la solución de problemas específicos de clientes externos.

Las funciones esenciales del DID:

- a) Investigación, administración y control de hardware y software de vanguardia.
- b) Desarrollo de sistemas requeridos por la Unidad y por la Facultad.
- c) Proyectos específicos a solicitud del cliente.
- d) Prestación de servicios de asesoría sobre tópicos especializados.
- e) Estudio y aplicación de software para desarrollar interfaces gráficas para diferentes manejadores de bases de datos.
- f) Coordinación del área de multimedia.
- g) Coordinar los aspectos técnicos del desarrollo web de la Facultad de Ingeniería.
- h) Administración de la página web de UNICA.

1.6.2 Infraestructura de Hardware

Para la realización de este proyecto en cuanto a hardware el DID proporcionará el siguiente equipo:

Capítulo 1

Definición del Proyecto

- Un servidor de aplicaciones que dará almacenamiento a la aplicación.
- Un servidor de bases de datos donde se centralizará la información que manejará el sistema.
- Máquinas de desarrollo para programar el sistema y hacer pruebas.

1.6.2 Infraestructura de Software

En cuanto a software, se utilizará básicamente software libre:

- Manejador de bases de datos: PostgreSQL
- Servidor WEB: Apache Tomcat
- Lenguaje de desarrollo: Java
- Plataforma de desarrollo: Linux y Windows
- Plataforma de implementación: Linux

1.7 Beneficios

Con la puesta en marcha de este proyecto se esperan los siguientes resultados

- Se modernizarán, los procedimientos actuales.
- Hacer más rápido y eficiente el proceso de actualización del inventario del equipo de cómputo de la Facultad.
- Realizar reportes del equipo con que cuenta cada dependencia de manera rápida y sencilla.
- Facilitar la realización del censo del equipo de cómputo que se lleva a cabo cada año en la Facultad.
- Mejorar el control y registro de equipos de cómputo dentro de la Facultad.
- Reducción del tiempo de atención del servicio.
- Acceso al servicio de forma fácil, amigable y segura.
- Generar reportes con la información de los equipos.
- Integración y centralización de la información en una base de datos.
- Contar con documentación del sistema:
 - A. Manual de instalación.
 - B. Manual de usuario.

2

MARCO TEÓRICO

Es fundamental para el desarrollo de este sistema contar con un marco teórico de referencia que permita la correcta toma de decisiones para dar la mejor solución al problema. A lo largo de este capítulo se presentarán los conceptos básicos de la Ingeniería de Software que representan la guía para el análisis y diseño del sistema.

Así mismo se describen brevemente las ventajas de las herramientas de software utilizadas en las etapas de desarrollo e implementación.

Contenido:

- 2.1 Ingeniería de Software
- 2.2 Software Libre
- 2.3 Herramientas de desarrollo

Capítulo 2

Marco teórico

2.1 Ingeniería de Software

La “crisis del software”, término utilizado por primera vez en 1968 en la conferencia de la OTAN sobre Ingeniería de Software (término también de reciente uso en esa época, atribuido al profesor Friedrich L. Bauer y aceptado en esta misma conferencia) sentó las bases para la evolución del desarrollo de software. Esta conferencia tenía la intención de hacer reflexionar sobre los problemas que se percibían en ese entonces pues no solo la sobreestimación de tiempo y costos de los proyectos sino, sobre todo, lo críticos que éstos llegaban a ser constituían una crisis.

Edsger W. Dijkstra en su obra “The Humble Programmer” (1972) habla de la crisis de software como un problema que se derivó de las diferencias entre la evolución de software y hardware, explica que en un principio los programadores afrontaban grandes retos ya que contaban con recursos limitados y el hecho de que las máquinas fueran cada vez más sofisticadas en lugar de resolver éstos problemas los enfrentó a una crisis del software, Dijkstra se pregunta por qué y nos describe dos causas:

- La primera se refiere a la facilidad de manejar las máquinas antiguas que eran secuenciales y totalmente deterministas, en suma el cambio en la estructura de las máquinas representó un conflicto para los programadores.
- La segunda y principal causa fue el inesperado crecimiento del número de máquinas y el paralelo crecimiento de las expectativas de la sociedad en aplicar estas máquinas. El aumento del poder del hardware aunado al aún más dramático incremento en su confiabilidad hizo factibles soluciones que los programadores no habían siquiera imaginado por lo que no estaban preparados para hacer frente la situación.

Hoy en día la crisis del software es vista más allá de un conflicto de diferencias entre hardware y software pero es cierto también que todos los problemas relacionados al desarrollo de software fueron causados por el lento cambio evolutivo, determinado por los cambios explosivos en las disciplinas relacionadas con el software y la poca e inmadura cultura de la programación.

El autor Roger. S. Pressman, autor de Ingeniería de Software Un enfoque práctico, se cuestiona si el término “crisis” es adecuado ya que ésta palabra es definida por el diccionario como “*un punto decisivo en el curso de algo*”; y sostiene que nunca ha habido un punto decisivo en la evolución del software sino solamente una “*aflicción crónica*”. Pressman afirma que esta aflicción crónica no se limita al software que no funciona bien sino que abarca todos los problemas asociados a cómo desarrollar software, cómo mantener el volumen cada vez mayor existente y cómo mantenerse al corriente de la demanda.

Adecuado o no, el término de *Crisis del software* fue el punto de partida para lograr el avance que ha habido en el campo de la Ingeniería de Software, la demanda no sólo de software a secas sino software de calidad que se ajuste a las verdaderas necesidades de los clientes y sea útil durante mucho tiempo ha ido en aumento por lo que en nuestros días es prácticamente imposible concebir un proyecto en el que la Ingeniería de Software no juegue un rol esencial.

Capítulo 2

Marco teórico

A lo largo de los años el proceso de crear software ha ido madurando y aunque las prácticas han evolucionado y en la actualidad todavía es imposible determinar con exactitud los costos de recursos y tiempo de un proyecto de software, gracias a la Ingeniería de software es posible lograr una aproximación y generar software de mejor calidad.

2.1.1 Definición

Es común confundir el término ingeniería de software con el de ingeniería de sistemas, si bien ambos están relacionados es necesario diferenciarlos, por lo que a continuación se presentan las definiciones de ambas.

La ingeniería de sistemas es la actividad de especificar, diseñar, implementar, validar, utilizar y mantener sistemas socio-técnicos. Esta actividad no solo está relacionada con el software sino con el hardware y las interacciones del sistema con los usuarios y su entorno, trata de los servicios que el sistema proporciona, las restricciones sobre las que se debe construir y funcionar así como las formas en las que el sistema es usado para cumplir su propósito.¹

Por otro lado, la ingeniería de software es una disciplina que ofrece métodos y técnicas para desarrollar y mantener software de calidad.²

La ingeniería de software comprende todos los aspectos de la producción de software, desde las etapas iniciales de la especificación del sistema hasta el mantenimiento de éste después de que se utiliza.³

Otras definiciones de Ingeniería de software que nos ofrecen otros autores son las siguientes:

*Ingeniería del Software es la aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadora y la documentación asociada requerida para desarrollar, operar (funcionar) y mantenerlos. Se conoce también como desarrollo de software o producción de software.*⁴

*Ingeniería del Software trata del establecimiento de los principios y métodos de la ingeniería a fin de obtener software de modo rentable que fiable y trabaje en máquinas reales*⁵

*La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación (funcionamiento) y mantenimiento del software; es decir, la aplicación de ingeniería al software.*⁶

¹ SOMERVILLE Ian, *Ingeniería de Software*, Séptima edición, Editorial Addison Wesley, España 2005.

² PRESSMAN Roger S, *Ingeniería del Software, Un enfoque práctico*, Quinta edición, Editorial Mc Graw Hill, España 2002

³ SOMERVILLE Ian, *Ingeniería de Software*, Séptima edición, Editorial Addison Wesley, España 2005.

⁴ BOEHM, B. W, *Software Engineering.*, IEEE Transactions on *Cornputers*, C-25, núm. 12.

⁵ BAUER, F. L, *Software Engineering, Information Processing*, 71, North Holland Publishing Co., Amsterdarn, 1972

⁶ IEEE: *Standards Collection: Software Engineering*, IEEE Standard 610.12-1990, IEEE.

Capítulo 2

Marco teórico

Estudiando y comparando todas estas definiciones podemos decir que el común denominador es ver a la ingeniería de software como una serie de herramientas que permiten desarrollar, operar y mantener software.

La ingeniería de software aparece como consecuencia de la ingeniería de sistemas al ser ésta más antigua que la primera debido que a las personas han especificado, diseñado y construido durante más tiempo todo tipo de sistemas como aviones y plantas químicas. La ingeniería de sistemas en lugar de centrarse únicamente en el software se centra en diversos elementos, analizando, diseñando y organizando esos elementos en un sistema que pueden ser un producto, un servicio o una tecnología para la transformación o control de información.

2.1.2 El proceso

El proceso del software tiene lógicamente, influencia del proceso de la ingeniería de sistemas, sin embargo existen dos diferencias básicas:

1. En la ingeniería de sistemas una vez que el diseño está hecho y se ha puesto en marcha el desarrollo es complicado hacer cambios, por el contrario la ingeniería de software permite responder a nuevos requerimientos durante el desarrollo del sistema con relativa facilidad.
2. La ingeniería de sistemas es una actividad en la que las personas involucradas tienen bases de conocimiento diversas y a menudo se presentan discrepancias para el uso de terminología y convenciones. Si bien la ingeniería de software no está exenta de esta situación, las posibilidades de reducir al tratarse de un producto específico.

La mayoría de los autores definen un proceso del software como un conjunto de actividades o tareas que se requieren para la creación de un producto de software, sin embargo Roger Pressman añade algo muy importante; el proceso de software no solo debe conducir a un producto de software si no que este producto debe ser de alta calidad.

No existe un proceso de software ideal, las organizaciones emplean el enfoque que consideran que mejor aprovecha las capacidades de sus recursos humanos y que mejor se adapta al sistema que desean obtener. A pesar de las diferencias que tengan los procesos todos tienen algunas actividades fundamentales, comunes entre ellos como son:

1. *Especificación del software.* Se debe definir la funcionalidad del software y las restricciones en su operación.
2. *Diseño e implementación del software.* Se debe producir software que cumpla su especificación.
3. *Validación del software.* Se debe validar el software para asegurar que hace lo que el cliente desea
4. *Evolución del software.* El software debe evolucionar para cubrir las necesidades cambiantes del cliente.

Capítulo 2

Marco teórico

El proceso de software elegido por algunas organizaciones no siempre suele ser el mejor para sus necesidades o pueden incluir técnicas anticuadas o no aprovechar las mejores prácticas de ingeniería de software por lo que es común que las organizaciones busquen estandarizar sus procesos.

Los procesos de software se pueden mejorar por la estandarización del proceso donde la diversidad de procesos del software en una organización sea reducida. Esto conduce a mejorar la comunicación y a una reducción de tiempo de formación. La estandarización también es un primer paso importante para introducir nuevos métodos, técnicas y buenas prácticas de la ingeniería del software.⁷

Actualmente existen varios estándares y modelos de calidad que las organizaciones utilizan no sólo con el objetivo de mejorar sus procesos sino además como una forma de buscar la confianza de sus clientes.

Uno de los modelos de calidad más importantes y quizá el más significativo en la industria del software a nivel internacional es el modelo desarrollado por el Instituto de Ingeniería de Software de la Universidad de Carnegie Mellon CMMI (Capability Maturity Model Integration) que se basa en un conjunto de funciones de ingeniería del software que deberían estar presentes conforme las organizaciones alcanzan diferentes niveles de madurez del proceso, estos niveles se describen a continuación:

Nivel 1: Inicial. El proceso del software se caracteriza según el caso, y ocasionalmente incluso de forma caótica. Se definen pocos procesos, y el éxito depende del esfuerzo individual.

Nivel 2: Repetible. Se establecen los procesos de gestión del proyecto para hacer seguimiento del coste, de la planificación y de la funcionalidad. Para repetir éxitos anteriores en proyectos con aplicaciones similares se aplica la disciplina necesaria para el proceso.

Nivel 3: Definido. El proceso del software de las actividades de gestión y de ingeniería se documenta, se estandariza y se integra dentro de un proceso de software de toda una organización. Todos los proyectos utilizan una versión documentada y aprobada del proceso de la organización para el desarrollo y mantenimiento del software. En este nivel se incluyen todas las características definidas para el nivel 2.

Nivel 4: Gestionado. Se recopilan medidas detalladas del proceso de software y de la calidad del producto. Mediante la utilización de medidas detalladas, se comprenden y se controlan cuantitativamente tanto los productos como el proceso del software. En este nivel se incluyen todas las características definidas para el nivel 3.

⁷ SOMERVILLE Ian, *Ingeniería de Software*, Séptima edición, Editorial Addison Wesley, España 2005.

Capítulo 2

Marco teórico

Nivel 5: Optimización. Mediante una retroalimentación cuantitativamente del proceso, ideas y tecnologías innovadoras se posibilita una mejora del proceso. En este nivel se incluyen todas las características definidas para el nivel 4.

Otros estándares de relevancia son los creados por la IEEE e ISO.

El Instituto de Ingenieros Electricistas y Electrónicos (IEEE por sus siglas en inglés) es una asociación de profesionales internacional dedicada a promover la innovación tecnología a través de publicaciones, conferencias, estándares tecnológicos y actividades profesionales y educativas. Cuenta con más de 55 estándares relacionados con ingeniería de software, algunos de los cuales se presentan a continuación:

- IEEE Std.830 Prácticas recomendadas para las especificaciones de software.
- IEEE Std. 1362 Guía para la especificación del documento de requisitos “ConOps”.
- IEEE Std. 1063 Estándar para la documentación de usuario de software.
- IEEE Std. 1012 Estándar para la verificación y validación de software.
- IEEE Std. 1219 Estándar para el mantenimiento del software.
- IEEE Std. 1061 Estándar para la metodología de métricas de la calidad de software.

La Organización Internacional para la Estandarización (ISO), es una organización no gubernamental que se encarga de desarrollar y publicar estándares internacionales. Es una red de institutos de estandarización en 163 países, un miembro por país, con una Secretaría Central en Ginebra Suiza que coordina el sistema.

Algunos estándares ISO relacionados con ingeniería de software son los siguientes:

- ISO/IEC 25000 Ingeniería de Software – Requerimientos y evaluación de calidad del producto de software (SQuaRE).
- ISO/IEC 9126 Ingeniería de software – Calidad del producto.
- ISO/IEC 24773 Certificación de Profesionales de ingeniería de software.
- ISO/IEC 14598 Ingeniería de Software – Evaluación del producto.
- ISO/IEC 15910 Tecnologías de la información – Proceso de documentación de usuarios de software.
- ISO/IEC 14471 Tecnologías de la Información – Ingeniería de Software – Guía para la adopción de herramientas CASE.

2.1.3 Modelos

En el inciso anterior se habló de modelos de calidad como CMMI, es importante no confundirlos con los modelos de proceso del software, ya que los primeros se interpretan como sinónimo de estándar debido a que están diseñados para evaluar la calidad del proceso y producto de software

Capítulo 2

Marco teórico

mientras que los modelos de proceso del software son sólo una representación abstracta de un proceso de software, que no siempre puede resultar el más adecuado para determinado proyecto.

Los proyectos de software necesitan una estrategia de desarrollo que conduzca al proceso, los métodos y herramientas que utilizan, a esta estrategia se le llama *modelo de proceso o paradigma de proceso*.

Cada proyecto es distinto por lo que es vital elegir un modelo que vaya de acuerdo a la naturaleza del proyecto, de la aplicación, métodos, herramientas y tiempo de entrega.

Los modelos simplemente representan enfoques particulares y no existe uno mejor que otro pero pueden ser adaptados o extendidos para crear procesos más específicos y de hecho es común encontrar variantes de un mismo modelo.

En este capítulo se describirán tres modelos. Es fundamental tener en cuenta que éstos y ningún otro son excluyentes y por lo cual pueden ser combinados en un mismo proyecto.

2.1.3.1 Modelo en cascada

Este modelo (Figura 2.1) también conocido como ciclo de vida del software considera las actividades fundamentales del proceso de especificación, desarrollo, validación y evolución y los representa como fases separadas del proceso, tales como la especificación de requerimientos, el diseño del software, implementación, las pruebas etcétera.

Las principales etapas de este modelo se transforman en actividades fundamentales de desarrollo.

1. *Análisis y definición de requerimientos*. Se definen con detalle los servicios, restricciones y metas del sistema a partir de las consultas con los usuarios. Estos datos sirven como especificación del sistema.
2. *Diseño del sistema y del software*. El proceso de diseño establece una arquitectura completa y divide los requerimientos en sistemas de hardware y software.
3. *Implementación y prueba de unidades*. El diseño del software se lleva a cabo como un conjunto o unidades de programas. La prueba de unidades implica verificar que cada una cumpla su especificación.
4. *Integración y prueba del sistema*. Los programas o las unidades individuales de programas se integran y se prueban como un sistema completo para asegurar que se cumplan los requerimientos de software. Después de las pruebas, el sistema se entrega al cliente.
5. *Funcionamiento y mantenimiento*. El sistema se instala y se pone en funcionamiento práctico. El mantenimiento implica corregir errores no descubiertos en las etapas anteriores, mejorar la implementación de las unidades del sistema y resaltar los servicios del sistema una vez que se descubren nuevos requerimientos.

Capítulo 2

Marco teórico

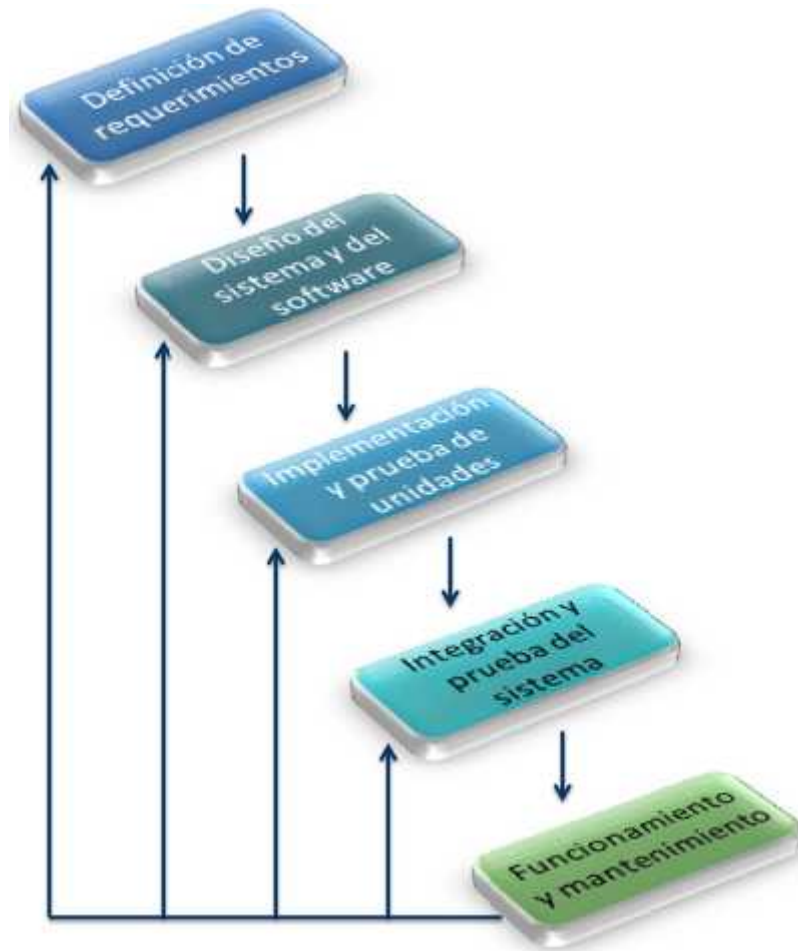


Figura 2.1 Modelo en cascada

El resultado de cada fase es uno o más documentos aprobados y la siguiente fase no debe comenzar hasta que la fase previa no haya finalizado sin embargo el proceso del software no es un modelo lineal simple, sino que implica una serie de iteraciones de las actividades de desarrollo.

Una de las principales desventajas de este modelo es su poca flexibilidad ya que es complicado y costoso responder a los cambios de requerimientos del cliente debido a que las iteraciones en las etapas implican rehacer el trabajo y por lo tanto después de determinado número de iteraciones se deben congelan las partes del desarrollo como la especificación para poder continuar con las siguientes etapas lo que podría ocasionar que el sistema no haga lo que los usuarios desean. Una vez que el sistema se ha puesto en funcionamiento y se descubren errores u omisiones en los requerimientos originales puede implicar repetir varias etapas del proceso.

Las ventajas del modelo son que la documentación se produce en cada fase y que es compatible con otros modelos del proceso de ingeniería.

Capítulo 2

Marco teórico

2.1.3.2 Modelo de desarrollo evolutivo

Se basa en la idea de desarrollar una implementación inicial, exponiéndola a los comentarios del usuario y refinándola a través de las diferentes versiones hasta que se desarrolla un sistema adecuado. Las actividades de desarrollo y validación se entrelazan, con una rápida retroalimentación entre éstas.

Un enfoque evolutivo para el desarrollo de software suele ser más efectivo que el enfoque de cascada, ya que satisface las necesidades inmediatas de los clientes.



Figura 2.2 Modelo de Desarrollo Evolutivo

La principal ventaja de este modelo (Figura 2.2) es que la especificación se puede desarrollar en forma creciente y tan pronto como los usuarios van desarrollando un mejor entendimiento del problema esto se ve reflejado en el sistema, no obstante este tipo de modelo es recomendado para sistemas pequeños y de tamaño medio.

Desde la perspectiva de ingeniería y de gestión, el enfoque evolutivo puede tener dos problemas:

El proceso no es visible. Los administradores tienen que hacer entregas regulares para medir el progreso. Si los sistemas se desarrollan rápidamente, no es rentable producir documentos que reflejen cada versión del sistema.

La estructura del sistema puede llegar a ser deficiente. Los cambios continuos tienden a corromper la estructura del software. Incorporar cambios en él se convierten cada vez más en una tarea difícil y costosa.

Capítulo 2

Marco teórico

2.1.3.3 Ingeniería del software basada en componentes

La ingeniería del software basada en componentes (ISBC) tiene fundamento en la reutilización, es decir, cuando las personas que trabajan en un proyecto conocen diseños o código similares al requerido. Los buscan, los modifican según lo creen necesario y los incorporan en el sistema. Éstos diseños o códigos se convierten entonces en componentes del sistema.

En ocasiones los componentes son sistemas por sí mismos (sistemas comerciales) que se pueden utilizar para proporcionar una funcionalidad específica. Existen tres tipos de componentes⁸:

1. *De distribución*, que conforman el fundamento de los sistemas ejecutables.
2. *Para trabajar en el producto*, a partir de los cuales se han creado los componentes de distribución.
3. *De ejecución*, creados como resultado de un sistema en ejecución.

Roger S. Pressman define un componente como una parte reemplazable, casi independiente y no trivial de un sistema que cumple una función clara en el contexto de una arquitectura bien definida.

Ahora bien cuando se desea integrar determinado componente a dicha arquitectura es deseable que tenga ciertas características como:

- Que se ajuste a un modelo *estandarizado*. Este modelo puede definir interfaces de componentes, metadatos de componentes, documentación, composición y despliegue.
- Que sea *independiente*, que pueda ser integrado y desplegado sin tener que utilizar otros componentes específicos.
- Que sea *componible*, es decir, aquel en el que todas las interacciones externas tienen lugar a través de interfaces definidas públicamente. Debe proporcionar, también, acceso externo a la información sobre sí mismo, como sus métodos y atributos.
- Que sea *desplegable* es decir, que el componente sea capaz de funcionar como una entidad autónoma o sobre una plataforma de componentes. Esto significa que el componente no tiene que compilarse antes de ser desplegado.
- Que esté *documentado* para que los usuarios potenciales puedan decidir si realmente satisfacen sus necesidades.

Aunque la base de la ISBC es la reutilización cuando un componente no satisface completamente las necesidades o no es posible adaptarlo a la arquitectura seleccionada, es necesario crear nuevos componentes que posteriormente también se deben integrar y ser comprobados a conciencia.

⁸ Extraído de Aprendiendo UML en 24 horas.

Capítulo 2

Marco teórico

El proceso ISBC no sólo debe identificar los componentes candidatos sino que también debe evaluar la interfaz de cada componente, debe adaptar los componentes para extraer las faltas de coincidencias arquitectónicas, ensamblar los componentes en un estilo arquitectónico seleccionado y actualizar los componentes a medida que cambian los requisitos del sistema. En la figura 2.3 se muestra el modelo básico de la ISBC:

1. *Análisis de componentes.* Dada la especificación de requerimientos, se buscan los componentes para su implementación. Por lo general los componentes sólo proporcionan parte de la funcionalidad.
2. *Modificación de requerimientos.* Los requerimientos se analizan utilizando información acerca de los componentes que se han descubierto. Entonces éstos se modifican para reflejar los cuales están disponibles. Si las modificaciones no son posibles, la actividad de análisis de componentes se puede llevar a cabo nuevamente para buscar soluciones alternativas.
3. *Diseño del sistema con reutilización.* Se diseña o se reutiliza un marco de trabajo para el sistema. Los diseñadores tienen en cuenta los componentes que se reutilizan y organizan el marco de trabajo para que los satisfaga. Si los componentes reutilizables no están disponibles, se puede tener en cuenta diseñar nuevo software.
4. *Desarrollo e integración.* Para crear el sistema, el software que no se puede adquirir externamente se desarrolla y los componentes se integran.



Figura 2.3 Ingeniería basada en componentes

Los fundamentos de la ingeniería del software basada en componentes son:

1. *Componentes independientes.* Debe haber una clara separación entre la interfaz de los componentes y su implementación para que una implementación de un componente pueda reemplazarse por otro sin cambiar el sistema.
2. *Estándares de componentes,* que facilitan la integración de los componentes. Si los componentes cumplen con los estándares, entonces su funcionamiento es independiente

Capítulo 2

Marco teórico

del lenguaje de programación y por lo tanto los componentes escritos en diferentes lenguajes pueden integrarse en un mismo sistema.

3. *El middleware*⁹. Para conseguir que componentes independientes y distribuidos trabajen juntos, se necesita soporte middleware que integre los componentes, maneje las comunicaciones entre ellos. Un producto middleware también puede proporcionar soporte para asignación de recursos, gestión de transacciones, seguridad y concurrencia.
4. *Un proceso de desarrollo*. Se necesita un proceso que se adapte a la ingeniería del software basada en componentes.

La ingeniería del software basada en componentes tiene la ventaja de reducir la cantidad de software a desarrollarse reduciendo, tiempo, costos y hasta cierto punto riesgos ya que puede presentar algunos inconvenientes como:

1. *Confiabilidad de los componentes*. El código de los componentes puede no estar disponible o tener modos de fallo no documentados, su comportamiento puede no ser el esperado o podría ocultar código dañino, lo que podría comprometer el sistema.
2. *Equilibrio de requerimientos*. Se tiene que encontrar un equilibrio entre los requerimientos ideales y los componentes disponibles en el proceso de especificación y diseño del sistema. Se necesita un método de análisis de equilibrio sistemático y más estructurado para ayudar a los diseñadores a seleccionar y configurar componentes.

⁹ Es un nombre genérico para designar un tipo de software cuyo propósito es conectar componentes de software o aplicaciones para que puedan intercambiar datos entre éstas.

Capítulo 2

Marco teórico

2.1.4 Requerimientos

Los requerimientos para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas.

Existen diversas clasificaciones, la más general es aquella que los divide en dos tipos (Figura 2.4):

- a) Requerimientos del usuario
- b) Requerimientos del sistema.



Figura 2.4 Requerimientos de usuario y de sistema

- a) Requerimientos del usuario

Pueden llegar ser ambiguos debido a la falta de claridad, confusión o combinación de requerimientos por lo que es conveniente contar con algún formato que permita estandarizarlos utilizando el lenguaje de forma consistente.

- b) Requerimientos del sistema

Los requerimientos del sistema se utilizan para comunicar, de forma precisa, las funciones que debe proporcionar el sistema. Para reducir la ambigüedad, se pueden redactar en un formulario estructurado del lenguaje natural complementando con tablas y modelos del sistema.

Otra forma de clasificar los requerimientos es dividirlos en funcionales y no funcionales. En las figuras 2.5 y 2.6 se resaltan las definiciones de la división.

Capítulo 2

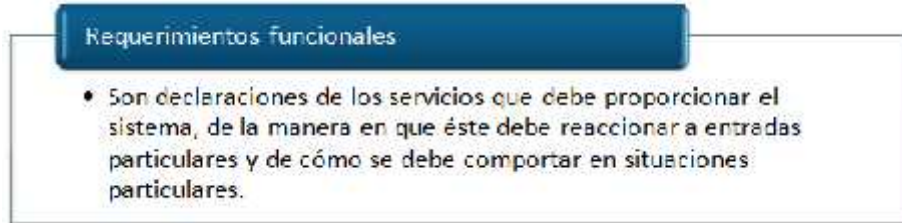


Figura 2.5 Definición de requerimientos funcionales

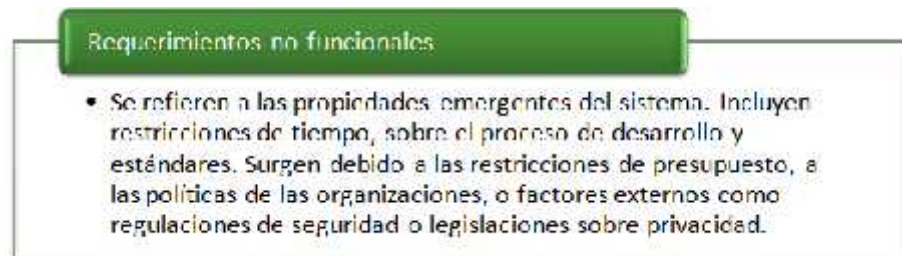


Figura 2.6 Definición de requerimientos no funcionales

En la figura 2.7 se muestra como los requerimientos no funcionales se dividen a su vez en requerimientos del producto, organizacionales y externos.



Figura 2.7 Tipos de requerimientos no funcionales.

Requerimientos del producto. Especifican el comportamiento del producto. Algunos ejemplos son los requerimientos de rendimiento, rapidez, fiabilidad, portabilidad y usabilidad.

Capítulo 2

Marco teórico

Requerimientos organizacionales. Se derivan de políticas y procedimientos existentes en la organización del cliente y en la del desarrollador. Algunos ejemplos son los estándares en los procesos, lenguajes de programación que deben usarse, tiempos de entrega del producto y su documentación

Requerimientos externos. Son requerimientos que se derivan de los factores externos al sistema y de su proceso de desarrollo como requerimientos legislativos para asegurar que el sistema funcione dentro de la ley o requerimientos éticos para asegurar que el sistema será aceptado por sus usuarios y público en general

Clasificar los requerimientos permite tener una mejor visión de lo que el cliente desea y por lo tanto satisfacer sus necesidades, así mismo permitirá redactar documentos donde éstos, sean menos ambiguos y con más detalle.

2.2 Software libre

Richard Stallman inició en 1984 el proyecto GNU con el objetivo de crear un sistema operativo completamente libre, la creación de este proyecto fue un hecho histórico decisivo que dio origen a lo que hoy conocemos como Software Libre.

La comunidad donde se desarrolló Richard Stallman en los años 70 cuando trabajaba en el laboratorio de Inteligencia Artificial del MIT solía compartir software sin ninguna restricción, cuando las empresas o universidades deseaban usar, portar o modificar algún programa este era compartido sin reserva, sin embargo en los años 80 la situación cambió radicalmente pues las computadoras modernas de la época contaban con sus propios sistemas operativos y ninguno de ellos era software libre, por lo que tener al menos una copia ejecutable se debía firmar un acuerdo de no divulgación.

Stallman creía firmemente que una forma de ayudar al prójimo era compartiendo software y la nueva dinámica representaba dividir, restringir y traicionar a los usuarios de computadoras. Ante la situación tuvo tres opciones, unirse al mundo del software privativo como lo hicieron muchos de sus compañeros, abandonar el campo de la computación para evitar renunciar a sus principios o comenzar de nuevo; eligió esta última opción y decidió que debía iniciar escribiendo un sistema operativo que fuera libre ya que era la base para ejecutar cualquier otro programa, así mismo invitó a participar a cualquiera que estuviera dispuesto a trabajar bajo la misma filosofía. Fue de esta manera que comenzó a crecer una nueva comunidad de hackers que ayudó a desarrollar, impulsar y mejorar el proyecto.

La Free Software Foundation (FSF) fue fundada en 1985 como consecuencia del creciente número de colaboradores del proyecto GNU. La FSF es una organización sin fines de lucro con una misión global para promover la libertad y defender los derechos de los usuarios de computadoras impulsando el desarrollo y uso de software libre y documentación, particularmente del sistema

Capítulo 2

Marco teórico

operativo GNU y haciendo campañas en contra de amenazas a la libertad de los usuarios como las tecnologías DRM (Digital Restrictions Management).

La FSF se encarga de mantener el histórico de los artículos que cubren la filosofía del software libre y mantiene la definición de Software Libre para mostrar claramente lo que deber ser verdad acerca de un programa en particular para que sea considerado software libre.

2.2.1 ¿Qué es software libre?

Según la definición del proyecto GNU el software libre es una cuestión de libertad de los usuarios de ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. Esto significa que los usuarios de los programas tienen las cuatro libertades esenciales.

- Libertad 0: Ejecutar el programa, para cualquier propósito.
- Libertad 1: Estudiar cómo trabaja el programa, y cambiarlo para que haga lo que usted quiera. El acceso al código fuente es una condición necesaria para ello.
- Libertad 2: Redistribuir copias para que pueda ayudar al prójimo.
- Libertad 3: Distribuir copias de sus versiones modificadas a terceros. Si lo hace, puede dar a toda la comunidad una oportunidad de beneficiarse de sus cambios.

Un programa es software libre si los usuarios tienen todas esas libertades. Entonces debería ser libre de redistribuir copias, tanto con o sin modificaciones, ya sea gratis o cobrando una tarifa por distribución, a cualquiera en cualquier parte. El ser libre de hacer esto, significa entre otras cosas, que no tiene que pedir o pagar el permiso.

Software libre es tener el control sobre la tecnología que usamos en nuestros hogares, escuelas y negocios, donde las computadoras trabajan para nuestro beneficio individual o comunal y no para el de compañías propietarias de software o gobiernos que podrían buscar restringirnos o monitorearnos.¹⁰

2.2.2 La filosofía de este proyecto

El objetivo de este proyecto es brindar apoyo a la Comunidad Universitaria y retribuir a la Facultad de Ingeniería lo que nos ha proporcionado.

Debido a que este proyecto fue desarrollado e implantado con herramientas de software libre es natural que el resultado sea también software libre.

¹⁰ Free Software Foundation

Capítulo 2

Marco teórico

En conclusión se autoriza la copia, redistribución y/o modificación del producto de software resultante de este proyecto (SICICE) bajo los términos de la Licencia Pública General GNU publicada por la Fundación de Software Libre, versión 1.3 y por lo tanto las personas son libres:

- Para compartir. Para copiar, distribuir y transmitir el trabajo
- A modificar. Para adaptar el trabajo bajo las siguientes condiciones:
 - ✓ Atribución. Atribuir el trabajo de la manera especificada por el autor o persona que lo haya licenciado (pero no de manera que sugiera que estas personas respaldan el uso que se haga del trabajo)
 - ✓ Compartir similar. En caso de alterar, transformar o ampliar este trabajo, deberá distribuir el trabajo resultante sólo bajo la misma licencia o una muy similar

2.3 Herramientas de Desarrollo

Para el desarrollo del sistema se decidió emplear preferentemente herramientas de software libre, a continuación se presentan las tres principales herramientas de trabajo de este proyecto que además de cumplir con esta condición tienen otras ventajas que se detallan en las siguientes líneas.

2.3.1 Java

Java es un lenguaje de programación cuyas principales características son que es orientado a objetos e independiente de la plataforma.

La Programación Orientada a Objetos es una metodología de desarrollo de software en la que un programa es conceptualizado como un grupo de objetos que trabajan juntos. Los objetos son creados utilizando definiciones concretas de elementos llamadas clases que contienen datos y sentencias requeridos para usar esos datos.

Esta metodología permite fabricar programas de forma más parecida al pensamiento humano pues simplifica el problema dividiéndolo en objetos. Usando este tipo de desarrollo se pueden hacer programas reusables, confiables y entendibles.

La independencia de plataforma significa que un programa tiene la habilidad de correr sin modificación alguna en diferentes ambientes de computadora ya sea hardware o software. Los programas hechos en Java son compilados en un formato llamado *bytecode* que puede ser ejecutado por un sistema operativo, software o dispositivo con un intérprete de Java. Esto significa que se puede crear un programa sobre Windows que corra de igual manera en Linux, OS X o una Palm. Siempre y cuando estas plataformas tengan el intérprete de java, podrán correr el *bytecode*.

Capítulo 2

Marco teórico

Los programas Java no son ejecutables, en su lugar son interpretados por una aplicación conocida como la máquina virtual de Java (JVM). En la Figura 2.8 se observa como el código fuente en Java se precompila generando un código (que no es directamente ejecutable) conocido como bytecode, este código generado normalmente en archivos con extensión .class es leído por la máquina virtual de Java que interpreta las instrucciones de los bytecodes, ejecutando el código de la aplicación. A este método de ejecución de programas en tiempo real se le llama Just in Time (JIT).

El bytecode se puede ejecutar en cualquier plataforma, lo único que se requiere es que esa plataforma posea la máquina virtual de Java que además es un programa pequeño y se distribuye gratuitamente para prácticamente todos los sistemas operativos.

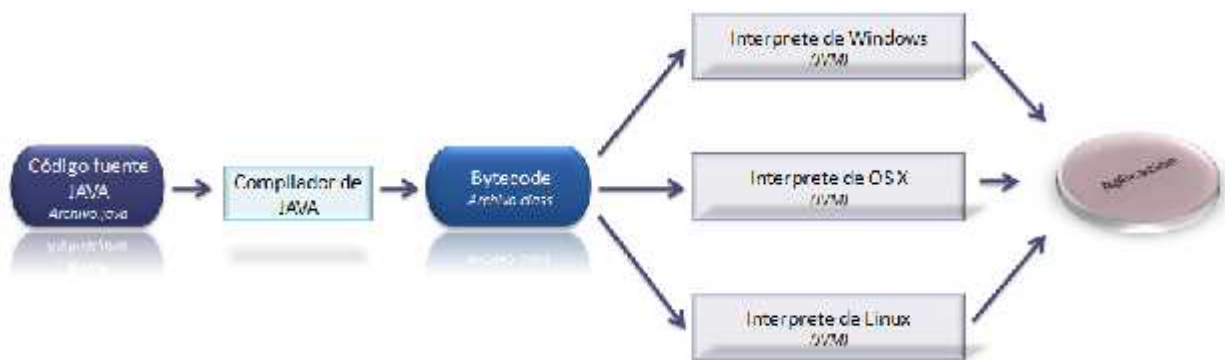


Figura 2.8 Proceso de compilación para un programa hecho en Java

Otra característica importante del lenguaje es que realiza verificaciones en busca de errores tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos de errores, lo antes posible, en el ciclo de desarrollo. Maneja la memoria a través de su recolector para eliminar las preocupaciones por parte del programador para la liberación o corrupción de memoria.

Elimina instrucciones dependientes de la máquina como los apuntadores que generaban posibilidades de atacar sistemas, tampoco permite el acceso directo a la memoria y el verificador de bytecode permite comprobar que el comportamiento del código es correcto y que sigue las reglas de Java. El segundo paso para verificar la seguridad es el verificador de clase que se asegura que las clases que se cargan son realmente las del sistema original de Java y no clases creadas y reemplazadas artificialmente. Finalmente hay un administrador de seguridad que es un programa configurable que permite al usuario indicar niveles de seguridad a su sistema para todos los programas de Java.

Capítulo 2

Marco teórico

2.3.2 PostgreSQL

PostgreSQL es el sistema administrador de bases de datos de código abierto más avanzado del mundo. Provee varias características que usualmente sólo son encontradas en bases de datos comerciales como DB2 u Oracle. A continuación se presenta un breve resumen de características fundamentales.

Enfoque *orientado a objetos y relacional* permite el manejo de rutinas y reglas complejas. Ejemplos de su funcionalidad avanzada son queries SQL declarativas, control de concurrencia multi-versión, soporte multi-usuarios, transacciones, optimización de queries, herencia y arreglos.

Extensible. PostgreSQL soporta operadores definidos por los usuarios, funciones, métodos de acceso y tipos de datos.

Integridad Referencial. PostgreSQL soporta integridad referencial, que es usada para asegurar la validez de los datos.

API Flexible. La flexibilidad de la API de PostgreSQL ha permitido proveer soporte fácilmente para algunas interfaces como Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++ y Pike.

Lenguajes de procedimientos. PostgreSQL tiene soporte para lenguajes de procedimientos internos, incluyendo su lenguaje nativo llamado PL/pgSQL. Este lenguaje es comparable al lenguaje de procedimientos PL/SQL de Oracle. Otra ventaja es la habilidad de usar Perl, Python o TCL como lenguajes de procedimientos embebidos.

MVCC. El Control de Concurrencia Multi-Versión es la tecnología que permite evitar bloqueos innecesarios. En otros manejadores de bases de datos suele ocurrir que cuando un usuario desea leer de la bases de datos tiene que esperar para acceder a la información. La espera es causada por otros usuarios que están escribiendo en la base de datos. En resumen el usuario que lee información es bloqueado por el que está escribiendo o actualizando registros.

Usando MVCC el usuario que lee nunca es bloqueado por el que escribe, en su lugar, PostgreSQL guarda un historial de todas las transacciones realizadas por los usuarios y por lo tanto es posible manejar los registros sin necesidad de esperar a que éstos estén disponibles.

Arquitectura Cliente/Servidor. PostgreSQL utiliza un proceso por usuario. Existe un proceso maestro que se fragmenta para proveer conexiones adicionales para cada cliente que se desea conectar al servidor.

Write Ahead Logging (WAL). Esta característica que se puede traducir como *escritura adelantada de Logs*; permite incrementar la disponibilidad guardando logs de cualquier cambio antes de ser escrito en la base de datos, esto asegura que si ocurre algún fallo habrá un registro de las transacciones para restaurarla. Esto puede ser benéfico en un escenario de falla en el que

Capítulo 2

Marco teórico

cualquier cambio no escrito en la base de datos pueda ser recuperado usando los datos que previamente fueron guardados. Una vez que el sistema es restaurado el usuario puede continuar trabajando en el punto exacto antes de ocurrir el fallo.

2.3.3 Apache Tomcat

Durante los últimos años Internet se ha convertido en una de las plataformas más importantes de los negocios y en el principal lenguaje de las tecnologías de la información. En la actualidad la World Wide Web es la fuente primaria de información y un importante medio de comunicación para muchas personas, debido a estas necesidades las páginas estáticas que dominaron en los primeros años han sido reemplazadas hace mucho tiempo con contenido dinámico.

El usuario moderno de la Web no sólo espera ver páginas que contengan, texto, información gráfica y contenido dinámico sino también desea usar la Web como front end de cualquier servicio. Aplicaciones Web para realizar compras, transacciones bancarias y financieras y otros servicios en línea son la evolución del concepto original de Internet.

Una de las primeras soluciones para cubrir todas estas expectativas fue la tecnología Common Gateway Interface (CGI). Los scripts CGI pronto se convirtieron en el mecanismo más popular para implementar aplicaciones que corrieran en un servidor web, sin embargo, no eran muy eficientes debido a que cada vez que el servidor recibía una petición éste debía crear un nuevo proceso y para aplicaciones con cientos o miles de usuarios esto se convertía en un serio problema de desempeño.

Otra desventaja de los CGI es que fue y sigue siendo un lenguaje script no estándar, mientras que la mayoría de los desarrolladores usan Perl, otros lenguajes como PHP también son populares.

Muchas alternativas fueron desarrolladas desde entonces como:

- Fast-CGI fue desarrollado para eliminar algunos de los problemas de performance, sin embargo, siguen sin ser del todo eficientes debido al lenguaje script usado.
- La comunidad Perl ha desarrollado un amplio número de librerías estándar que pueden mejorar la portabilidad de los scripts Perl.
- En las plataformas de Microsoft las Active Server Pages (ASP) implementadas con su servidor web IIS han reemplazado la tecnología CGI.
- Sun Microsystems creó la Tecnología de los Servlets y los JSP (Java Server Pages) como una alternativa a los CGI para plataformas que soportaran Java.

De todas estas alternativas la última se ha convertido en una de las más populares debido a que es libre, escalable y portable. Los servlets requieren un contenedor, éste recibe las peticiones http de los navegadores y las envía a los servlets que generan la respuesta. Un contenedor también puede ser integrado con servidores web para el manejo más eficiente de contenido estático.

Capítulo 2

Marco teórico

Aunque los servlets representaban una gran mejora sobre CGI, tenían un inconveniente, el código html embebido hace poco flexible el mantenimiento ya que cada vez que es necesario modificar el código html, el servlet debe ser recompilado.

Una segunda consecuencia de este esquema es que el diseñador html debía tener el conocimiento suficiente de Java para evitar dañar los servlets. Para resolver este problema, Sun Microsystems desarrolló la tecnología Java Server Pages (JSP).

Los JSP's en realidad siguen siendo servlets, pero escritos como páginas html con código Java embebido. Las páginas JSP son traducidas a servlets y después estos se compilan de manera normal para ser conservados en memoria. Cada vez que el servidor recibe una petición el servlet responde dicha petición haciendo el proceso mucho más eficiente que las ASP pues éstas requieren que el servidor analice y compile el documento cada vez que el usuario realiza una petición.

JSP se enfoca en la presentación de la información mientras que los servlets se enfocan en la funcionalidad. Arquitecturas como el Modelo Vista Controlador (MVC) se encargan de profundizar en el diseño de las aplicaciones Web para tomar lo mejor de cada tecnología.

Jakarta Tomcat es un servidor Web de código abierto de la Apache Software Foundation que proporciona soporte para la creación de contenido dinámico usando el lenguaje de programación Java.

Tomcat tiene sus orígenes en los primeros días de la tecnología servlet. Sun Microsystems creó el primer contenedor de servlets, el Java Web Server para demostrar esta tecnología, pero no era muy robusto. Al mismo tiempo la Apache Software Foundation (ASF) creaba JServ, un motor de servlets que integraba el servidor web Apache.

En 1999, Sun Microsystems dono el código del Java Web Server a la ASF y de estos dos proyectos surgió Tomcat.

La figura 2.9 muestra la Arquitectura de Tomcat que consiste en una serie de componentes anidados jerárquicamente.

Capítulo 2

Marco teórico

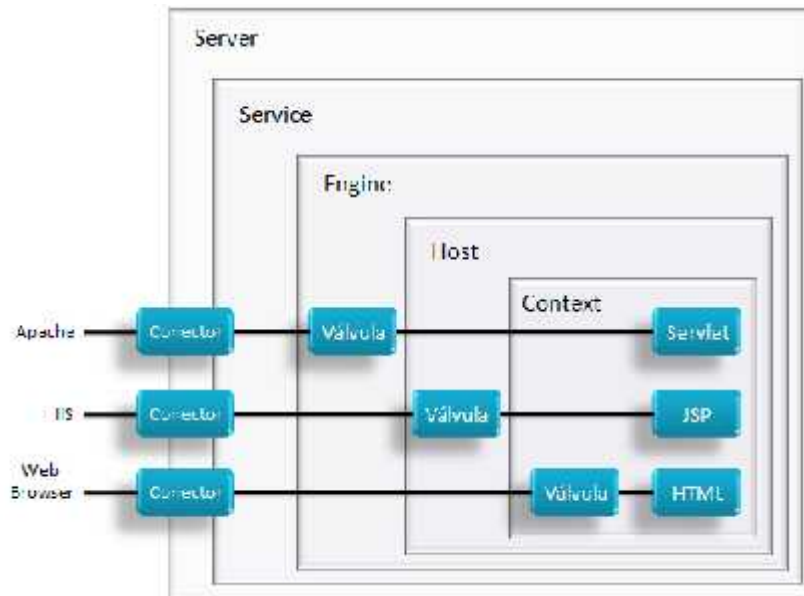


Figura 2.9 Ejemplo de una configuración de Tomcat.

Tomcat está configurado con el Lenguaje de Marcas Extensible (XML) que refleja la jerarquía de componentes. El archivo que guarda esta configuración es el *server.xml*

Componentes de alto nivel.

- **Servidor.** Este componente es una instancia del servidor Tomcat. Sólo se puede crear una instancia dentro de una JVM (java Virtual Machine). Es posible instalar servidores separados configurados en diferentes puertos para separar las aplicaciones y se puedan reiniciar independientemente. De esta manera si una JVM deja de funcionar las demás aplicaciones estarán a salvo en otra instancia del servidor.
- **Servicio.** Examina las cabeceras HTTP para determinar a que host o contexto debe pasar la petición. Este componente acepta las peticiones, las enruta a la aplicación Web adecuada y regresa el resultado de la petición procesada.

Conectores

Los conectores conectan las aplicaciones web con los clientes. Son el punto donde las peticiones de los clientes son recibidas y cada una tiene un puesto único en el servidor. El puerto HTTP de Tomcat por default es el 8080 para evitar cualquier interferencia con cualquier servidor web corriendo en el puerto 80, el puerto estándar de HTTP.

Componentes contenedores

Capítulo 2

Marco teórico

Los componentes contenedores reciben las peticiones de los componentes de alto nivel y las procesan.

- Engine (Motor). Este componente es el contenedor de alto nivel y no puede estar contenido dentro de otro. Solo puede existir un de este tipo por cada servicio. Representa el Motor de servlets Catalina. Examina las cabeceras HTTP para determinar a que host virtual o contexto debe transmitir la petición. De esta manera se puede observar como la progresión de la petición a través de los diferentes niveles de la jerarquía. El nombre de host del servidor es asignado en el engine si es necesario. Un engine puede contener hosts representando a un grupo de aplicaciones web y contextos, cada contexto representando a una sola aplicación
- Host. Permite configurar múltiples servidores una misma máquina y ser identificados por distintas IP's o nombres de host, por ejemplo pueden existir `www.unam.mx` y `www.ingenieria.unam.mx` en el mismo servidor. El engine examina la cabecera HTTP para determinar a que host debe ser enviada la petición.
- Contexto. Es el componente de más bajo nivel también conocido como aplicación web. Cuando se configura un contexto se informa al contenedor de servlets la localización de la carpeta raíz de la aplicación y de esta manera el contenedor puede enrutar la petición efectivamente. Un contexto también puede incluir "error pages", que permiten configurar los mensajes de error de acuerdo a la aplicación. Finalmente también es posible configurar un contexto con parámetros de inicialización para la aplicación que representa y para el control de acceso (restricciones de autenticación y autorización)

Componentes anidados

Estos componentes están anidados dentro de un componente contenedor y proveen servicios de administración. Su principal propósito es habilitar los distintos contenedores de Tomcat.

- Válvula. Es un elemento de procesamiento. Se usan para interceptar peticiones y procesos antes de que alcance su destino. Una válvula es parecida a un filtro. Las válvulas son generalmente usadas para llevar una bitácora de peticiones, direcciones IP de los clientes y uso del servidor. Esta técnica es conocida como "request dumping". Se guardan los registros de las cabeceras HTTP y cualquier cookie enviada en la petición.

3

DESARROLLO

El objetivo de este capítulo es describir a nivel conceptual la arquitectura del sistema, primero tomando como base el contexto presentado en el Capítulo 1 y en segundo lugar fundamentado en la revisión de conceptos de ingeniería del software del Capítulo 2.

Para comenzar se hace un reconocimiento de las necesidades de la primera versión del Sistema de Inventarios (SICI) con la finalidad de detectar puntos de oportunidad y para determinar los requerimientos para la nueva versión (SICICE).

Después se realiza un análisis de los requerimientos que permite hacer el primer esbozo del nuevo sistema, se definen sus operaciones básicas y secundarias para finalmente proceder al diseño.

Contenido:

- 3.1 Necesidades del sistema
- 3.2 Requerimientos del sistema
- 3.3 Análisis del sistema
- 3.4 Diseño basado en componentes

Capítulo 3

Desarrollo

3.1 Necesidades del sistema

En el capítulo 1 se realizó una exploración de la estructura y funciones básicas de la primera versión del SICI y se expusieron los requerimientos de usuario, los cuales permitieron detectar las siguientes necesidades:

- a. *Rediseñar la base de datos.* Debido a la introducción de nuevos tipos de equipos de cómputo es necesario crear nuevas tablas y eliminar los campos de algunas de ellas ya sea porque se ha detectado que no se usan o redundan información.
- b. *Flexibilidad del sistema.* Es necesario que el sistema sea capaz de adaptarse fácil y rápidamente cuando se desean agregar nuevas tablas a la base de datos. Esto es básicamente pensado para los casos en los que se deban agregar nuevos tipos de bienes, sin embargo este pensamiento debe acompañar el entero diseño del sistema.
- c. *Diseñar una interfaz más intuitiva.* Una interfaz sencilla e intuitiva permitirá al usuario reducir la curva de aprendizaje del manejo del sistema.
- d. *Creación de reportes.* Para las divisiones (Figura 1.1) es de gran importancia poder contar con una herramienta que les permita filtrar la información que necesitan y en un formato que sea fácil de compartir. Tal es el caso de las auditorías o censos donde se debe obtener información específica de manera rápida.
- e. *Validaciones.* Es necesario que el sistema pueda realizar el mayor número de validaciones posibles para evitar que los usuarios ingresen información incorrecta, repetida o inválida en la base de datos.
- f. *Creación de un usuario administrador.* El contar con un usuario administrador permitirá dar fácil mantenimiento al sistema.
- g. *Documentación.* Es importante que los usuarios cuenten con un completo manual de referencia que les permita resolver el mayor número de dudas posibles con respecto al funcionamiento del sistema. Así mismo es fundamental contar con la documentación del diseño, manuales de instalación y programación para facilitar el mantenimiento del mismo.

Capítulo 3

Desarrollo

3.2 Requerimientos del sistema

En el capítulo 1 se describieron los requerimientos del usuario, en las siguientes páginas se presentan formalmente, se clasifican en funcionales y no funcionales lo que nos permitirá visualizar de manera general las funciones y restricciones operativas del sistema.

3.2.1 Requerimientos funcionales del sistema

La Tabla 3 presenta organizados los requerimientos del sistema. Al final de la tabla de requerimientos se presentan otras cinco relacionadas con algunos de los requerimientos y a las que se hace referencia a lo largo del documento.

Cada dependencia de la Facultad cuenta con un usuario para el sistema por lo que en este caso usuario y dependencia son sinónimos.

No.	Requerimiento
R01	El sistema deberá ser accedido vía web a través de un usuario y contraseña.
R02	Los usuarios sólo podrán ver y gestionar los bienes que tengan asignados. Por ejemplo, si el usuario de la Secretaría Administrativa entra al sistema sólo podrá ver y manipular los datos de los bienes que tenga bajo su cargo y no podrá ver los de otros usuarios del sistema.
R03	El sistema contará con un módulo de Altas en el cual se podrán de registrar los datos de la Tabla 3.3 de cada uno de los bienes especificados en la Tabla 3.2.
R04	Para el caso de las Computadoras se agregarán adicionalmente los datos especificados en la tabla 3.4. Este requerimiento es derivado del requerimiento 3.
R05	Para el caso de las Impresoras se deberá indicar adicionalmente el tipo de impresora
R06	Cuando un usuario desee dar de alta un bien, deberá indicar obligatoriamente el número de inventario y para el caso de Computadoras adicionalmente los datos especificados en la tabla 3.5. Este requerimiento se deriva del requerimiento 3.
R07	Para asignar un responsable a un bien se podrán elegir dos opciones: a. Elegir de una lista un responsable existente. b. Asignar un nuevo responsable para lo cual se deberán especificar los datos de la tabla 3.6 para cada responsable nuevo. Este requerimiento es derivado de los requerimientos 3 y 10.
R08	Cuando se asigne un responsable nuevo se deberá indicar obligatoriamente el nombre, la ubicación y el teléfono. Este requerimiento es derivado de requerimiento 7.
R09	El sistema contará con un módulo de Bajas en el que se podrá dar de baja del sistema cualquiera de los bienes de la tabla 3.2.
R10	El sistema tendrá un módulo de Modificaciones que ofrecerá dos opciones: a. Editar la información de cualquiera de los responsables de los equipos. b. Editar los datos de los bienes de la tabla 3.2, permitirá cambiar los datos especificados en las tablas 3.3 y 3.4. Cuando se desee modificar el responsable del equipo se deberá tomar en cuenta el requerimiento 7.

Capítulo 3

Desarrollo

R11	<p>En el sistema existirá un módulo de Movimientos que permita a un usuario liberar de su custodia cualquier bien, para que este pueda ser asignado a otro usuario. El módulo ofrecerá tres opciones:</p> <ol style="list-style-type: none"><i>Poner un bien en movimiento.</i> En esta sección el usuario puede elegir uno varios de los bienes que tenga bajo su cargo para que sean liberados, de manera que una vez realizada esta operación, el bien liberado aparecerá como que no pertenece a ningún usuario (dependencia) y puede ser asignado a otro usuario.<i>Asignar un bien en movimiento.</i> Esta sección presentará un menú de todos los bienes que fueron liberados y por lo tanto no pertenecen a ningún usuario (dependencia). El usuario podrá elegir uno de estos bienes y ponerlo a su cargo.<i>Crear un equipo nuevo.</i> En esta sección el usuario podrá formar un equipo de varios bienes en movimiento.
R12	<p>En el sistema deberá existir una sección donde el usuario pueda ver los detalles de los bienes que tiene bajo su custodia. Esta sección podrá consistir simplemente en una página donde se muestra en un listado los datos de cada bien.</p>
R13	<p>El sistema deberá contar con una opción que permita al usuario generar reportes de los bienes que tiene bajo su responsabilidad y deberán cumplir con las siguientes especificaciones:</p> <ol style="list-style-type: none">Deberán generarse en formato Excel.Se deberá generar una tabla con los datos de todos los bienes que el usuario tiene bajo su custodia, esta tabla podrá ser dinámica por lo que el sistema permitirá al usuario filtrar los datos que necesite, es decir podrá elegir que datos de las tablas 3.3 y 3.4 deberán aparecer en el reporte.Deberá presentarse una segunda tabla donde se muestre un resumen con el número de cada bien con que cuenta esa dependencia.El usuario podrá generar reportes de los bienes que tiene bajo su responsabilidad.En el Anexo 1 de este trabajo se puede observar el formato en que se deben presentar los reportes.
R14	<p>Existirá una sección especial para visualizar y generar en formato Excel el censo de los bienes que tiene cada usuario. En esta sección se deberán generar siete tablas (Ver formatos en el Anexo 2):</p> <ol style="list-style-type: none">Una tabla que presente los equipos de computo por estado, es decir si están en uso o desuso y clasificados bajo los siguientes criterios:<ol style="list-style-type: none">Computadoras de escritorio.Macintosh.Portátiles.Alto rendimiento.Una tabla que presente los equipos de cómputo por usuario final y clasificados bajo los siguientes criterios.<ol style="list-style-type: none">Computadoras de escritorio.Macintosh.Portátiles.Alto rendimiento.Una tabla donde se presenten los equipos de cómputo por sistema operativo y clasificados bajo los siguientes criterios:

Capítulo 3

Desarrollo

	<ol style="list-style-type: none">a. Computadoras de escritorio.b. Macintosh.c. Portátiles.d. Alto rendimiento. <ol style="list-style-type: none">4. Una tabla que presente los equipos de cómputo por el tipo de conexión que tienen y organizados bajo los siguientes criterios:<ol style="list-style-type: none">a. Computadoras de escritorio.b. Macintosh.c. Portátiles.d. Alto rendimiento.5. Una tabla que presente los equipos de impresión por estado, es decir, si están en uso o desuso.6. Una tabla que presente los equipos (cámaras digitales, proyectores y scanners) por estado, es decir, si están en uso o desuso.7. Una tabla que presente los equipos de telecomunicaciones (Access point, concentradores, módems de fibra óptica, ruteadores y switches) por estado, es decir, si se encuentran en uso o desuso.
--	--

Tabla 3.1 Requerimientos funcionales del sistema

Tipo de bien	Datos de los bienes
Access Point	Número de inventario
Cámara Digital	Número de serie
Concentrador	Número de resguardo
Modem de Fibra Óptica	Responsable
Plotter	Marca
Router	Modelo
Monitor	Garantía
Mouse	
Teclado	
Unidad de CD externa	
Impresora	
Escáner	
Disco duro externo	
Regulador	
Hub	
Switch	
Nobreak	
Proyector	
Computadora (CPU y portátiles)	

Tabla 3.2 Tipo de bienes

Tabla 3.3 Datos de los bienes

Capítulo 3

Desarrollo

Datos específicos para Computadoras	
Tipo:	PC MAC Portátil Servidor
Tipo de usuario:	Personal Académico Personal Administrativo Alumnos Investigadores Usuarios de bibliotecas Administrativos de bibliotecas Servidor general Servidor WEB
Conexión:	Cableada Inalámbrica Vía módem Ninguna
Elementos adicionales:	Bocinas, Tarjeta de red, CD Writer, Audifonos, DVD, Unidad de Cinta Otros
Velocidad del procesador	
Capacidad del disco duro	
Memoria RAM	
Sistema Operativo	

Tabla 3.4 Datos específicos para computadoras

Campos obligatorios para Computadoras	
Tipo de equipo	
Número de serie	
Sistema operativo	
Tipo de usuario	
Conexión	
Procesador	

Tabla 3.5 Campos obligatorios para computadoras

Datos para dar de alto a un responsable de un bien	
Nombre completo	
Ubicación	
Correo electrónico	
Teléfono	
Fax	

Tabla 3.6 Datos responsable

3.2.2 Requerimientos no funcionales del sistema

Los siguientes requerimientos se fundamentan en las políticas de la Facultad de Ingeniería para la cual será desarrollado el sistema.

No.	Requerimiento
R15	A cada división o dependencia de la Facultad de Ingeniería le corresponderá un usuario. Las dependencias que deberán contar con usuario se describen en la tabla 3.7
R16	A cada división o entidad (dependencia) le corresponderá un usuario.
R17	La interfaz del sistema deberá de ir de acuerdo a la imagen de la Institución por lo cual deberá emplear tonalidades rojas.
R18	En la interfaz del sistema se deberá mostrar la identidad de la institución educativa para la cual fue elaborado por lo cual deberá tener los escudos de la UNAM y de la Facultad de Ingeniería. Adicionalmente deberá mostrar la identidad de la Institución que lo elaboró, por lo que deberá mostrar el escudo de UNICA.
R19	El desarrollo del sistema se deberá apegar al Estándar de Herramientas Tecnológicas Para el Desarrollo de Sistemas de UNICA.
R20	El sistema deberá contar con documentación, por lo que se deberán elaborar:

Capítulo 3

Desarrollo

	<ol style="list-style-type: none">Manual de instalación.Manual de usuario.Documentación del Análisis y Desarrollo del sistema
--	---

Dependencias a las que se les debe asignar un usuario en el sistema	
División de Ingeniería Civil, Topográfica y Geodésica	Sala UNICA 1
División de Ingeniería en Ciencias de la Tierra	Sala UNICA 2
División de Ciencias Básicas	Sala UNICA 3
División de Ingeniería Mecánica e Industrial	Sala UNICA 4
División de Ciencias Sociales y Humanidades	Información y Estadística
División de Ingeniería Eléctrica	Apoyo Editorial
División de Estudios de Posgrado	Apoyo a Consejo
División de Educación Continua	Comunicación
Secretaría Administrativa	Personal Académico
Secretaría de Servicios Académicos	Superación Académica
Dirección de la Facultad de Ingeniería	Atención Diferenciada
Oficina del Secretario	Revista de Ingeniería
Jefatura de UNICA	Centro de Docencia
Departamento de Investigación y Desarrollo, UNICA	Coordinación de Evaluación Educativa
Departamento de Servicios Académicos, UNICA	Coordinación de Planeación y Desarrollo
Departamento de Redesy Operación de Servidores UNICA	Coordinación de bibliotecas
Departamento de Seguridad, UNICA	

Tabla 3.7 Dependencias a las que se les debe asignar un usuario en el sistema.

Capítulo 3

Desarrollo

3.3 Análisis del sistema

Los requerimientos son la guía que permiten definir detalladamente cada una de las funciones y restricciones operativas del sistema, para hacer el análisis se utilizará el Lenguaje Unificado de Modelado, UML.

UML (Unified Modeling Language) es un lenguaje que permite modelar, construir, visualizar, especificar y documentar los elementos que forman un sistema orientado a objetos.

UML no es una metodología. Aunque los iniciadores y autores de UML reconocen la significancia de las metodologías, las consideran distintas de los lenguajes y sistemas de notaciones. Una metodología considera un marco y condiciones específicas para el dominio de una aplicación, el ambiente organizacional, entre otras cosas. UML puede ser usado con varias metodologías y puede formar las bases para distintos enfoques, ya que provee un conjunto definido de modelos dentro de una notación y semántica uniforme. Éste es uno de los principales motivos para emplear UML como herramienta y porque se ha convertido en el estándar de la industria, debido a que ha sido concebido por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh.

Otro motivo importante es apearse a los requerimientos del cliente que especificó emplear el Estándar de Herramientas Tecnológicas Para el Desarrollo de Sistemas de UNICA el cual identifica este lenguaje como una de las principales herramientas para el desarrollo de sistemas.

3.3.1 Definición del alcance del sistema

El sistema tiene por objetivo cubrir las necesidades de la Facultad de Ingeniería, éstas consisten básicamente en contar con un método eficiente que le permita llevar el control de sus equipos de cómputo.

De acuerdo a las necesidades detectadas y a los requerimientos definidos, el sistema contará con las siguientes cinco operaciones básicas:

- *Altas*. Esta operación permitirá ingresar los datos de un equipo al sistema, una vez dado de alta el equipo podrá ser administrado y se podrán realizar otras operaciones sobre él. Cuando un bien es dado de alta, queda asignado automáticamente a la división (usuario) que realizó la operación.
- *Bajas*. Esta operación permite eliminar los datos de un equipo del sistema.
- *Modificaciones*. Esta operación permitirá editar los datos tanto de los equipos como de los responsables de dichos equipos.
- *Movimientos*. Esta operación está pensada para que las divisiones puedan eliminar de su lista de bienes bajo su cargo, equipo que ha dejado de ser útil para ellos. A diferencia de una baja, cuando un bien está en movimiento a pesar de no pertenecer a ninguna división, aún está registrado en el sistema, esto quiere decir, que aunque el equipo ha dejado de ser útil para la división a la que pertenecía, aún se encuentra en buen estado y puede ser

Capítulo 3

Desarrollo

utilizado por otra división, de esta manera esta sección también permitirá asignar bienes en movimiento.

- *Reportes.* Esta operación permitirá visualizar y generar reportes completos o personalizados de los bienes que tiene bajo su cargo una división, así mismo permitirá generar censos que contabilicen y resuman de acuerdo a ciertos criterios los bienes de las divisiones.

El sistema contará con dos tipos de usuarios:

- *Divisiones.* Este tipo de usuario podrá realizar todas las operaciones descritas, pero sólo bajo los bienes que tenga asignados.
- *Administrador.* Este tipo de usuario a diferencia del usuario de tipo división podrá generar reportes de cualquiera de las divisiones del sistema. Podrá realizar cualquiera de las otras operaciones (altas, bajas, modificaciones y movimientos) pero los bienes pertenecerán al administrador.

El sistema se limitará a la administración del inventario a nivel de hardware, es decir, no se toman en cuenta elementos cuantitativos o cualitativos del software de los equipos como número o tipo de programas que tengan instalados.

El sistema no está pensado para la inclusión de bienes que no estén relacionados con equipo de cómputo, sin embargo debe ser lo suficientemente flexible para permitir agregar nuevos tipos de bienes.

3.3.2 Operaciones del sistema

En esta sección se describe de manera breve las operaciones básicas del sistema, a través de diagramas de actividades.

Los diagramas de actividades están diseñados para mostrar una visión simplificada de lo que ocurre durante una operación o proceso, nos permiten tener una perspectiva de cómo será el flujo de dicha operación, son similares a los diagramas de flujo, pero a diferencia de éstos, los diagramas de actividades permiten representar flujos en paralelo.

En el SICICE se distinguen cinco operaciones: Altas, Bajas, Modificaciones, Movimientos y Reportes; cada una de estas operaciones corresponden a los cinco módulos en que está dividido el sistema. En las siguientes imágenes se muestra el proceso de cada una de ellas.

3.3.2.1 Altas

El módulo de altas es uno de los más importantes del sistema ya que permite ingresar los datos de uno o más bienes dentro del sistema, en la Figura 3.1 se describe el proceso para realizar esta operación.

Capítulo 3

Desarrollo

Debido a la simplicidad del proceso es posible describir de manera paralela las acciones que van realizando alternativamente el usuario y el sistema. El diagrama nos permite visualizar los puntos fundamentales a implementar tales como las validaciones que si bien no se detallan, es posible identificar en qué casos se deben aplicar.

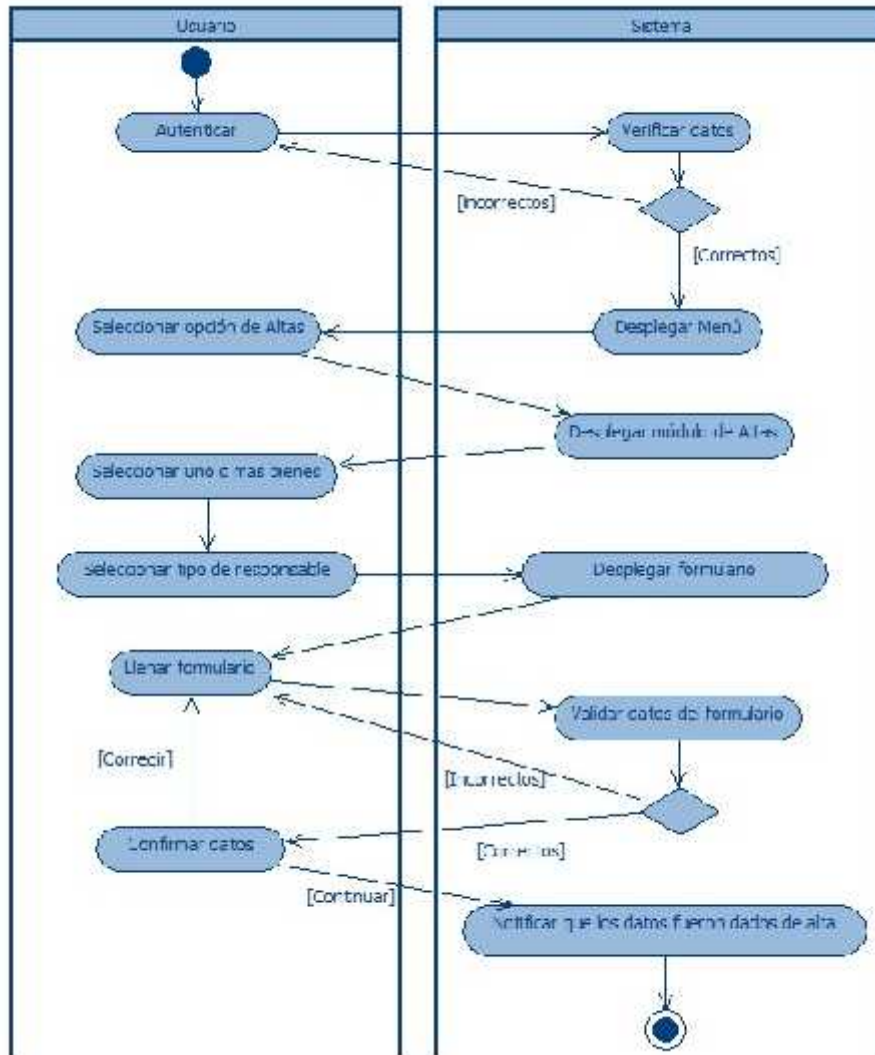


Figura 3.1 Diagrama para registrar nuevos bienes (altas)

3.3.2.2 Bajas

El módulo de bajas implica una búsqueda ya que es necesario identificar el bien que se desea eliminar del sistema. Este proceso al igual que el de altas sigue siendo simple a pesar de la búsqueda por lo que también es posible representar que acciones realiza el usuario y cuáles el sistema.

Capítulo 3

Desarrollo

El nivel de detalle de este diagrama es bajo por lo que refleja de manera breve el objetivo del módulo. Figura 3.2.

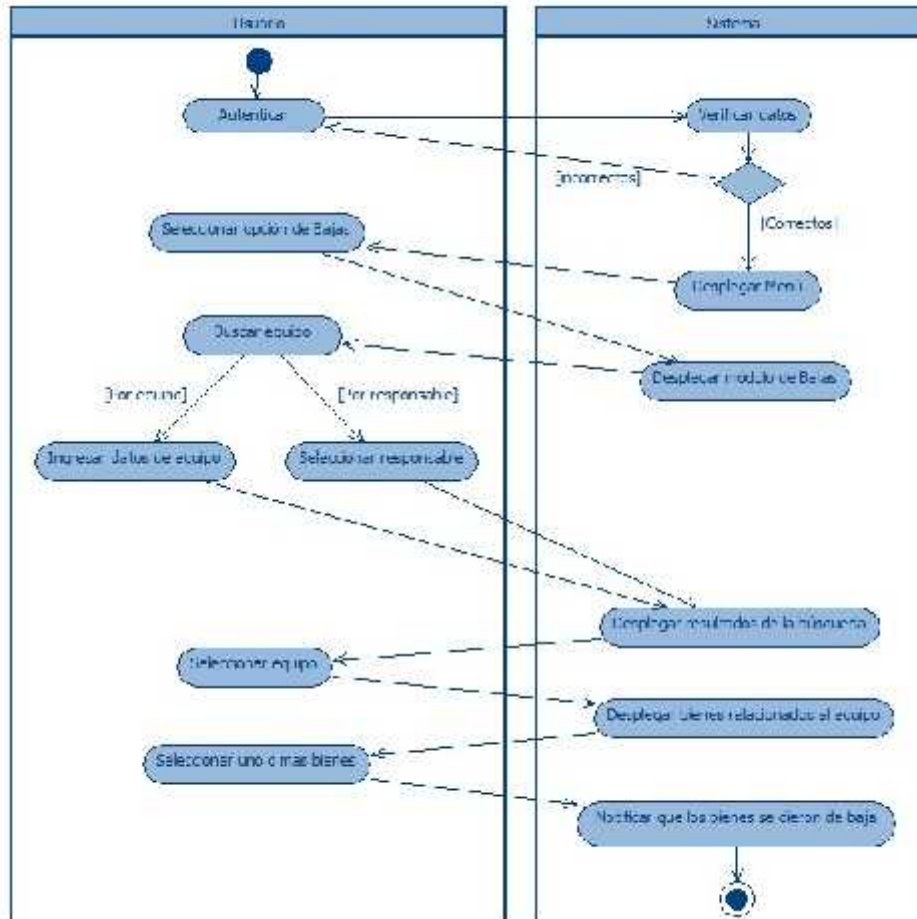


Figura 3.2 Diagrama para eliminar bienes (bajas)

3.3.2.3 Modificaciones

El proceso para representar este módulo es mucho más complicado que los anteriores por lo que en estos diagramas ya no se separan las acciones del usuario y el sistema, en cambio se muestra con más nivel de detalle en cada uno de los pasos del proceso. La figura 3.3 permite representar un aspecto fundamental de este módulo, el proceso se separa en dos ramas, la primera describe el proceso para modificar los datos de un responsable y la segunda para modificar los datos de un bien.

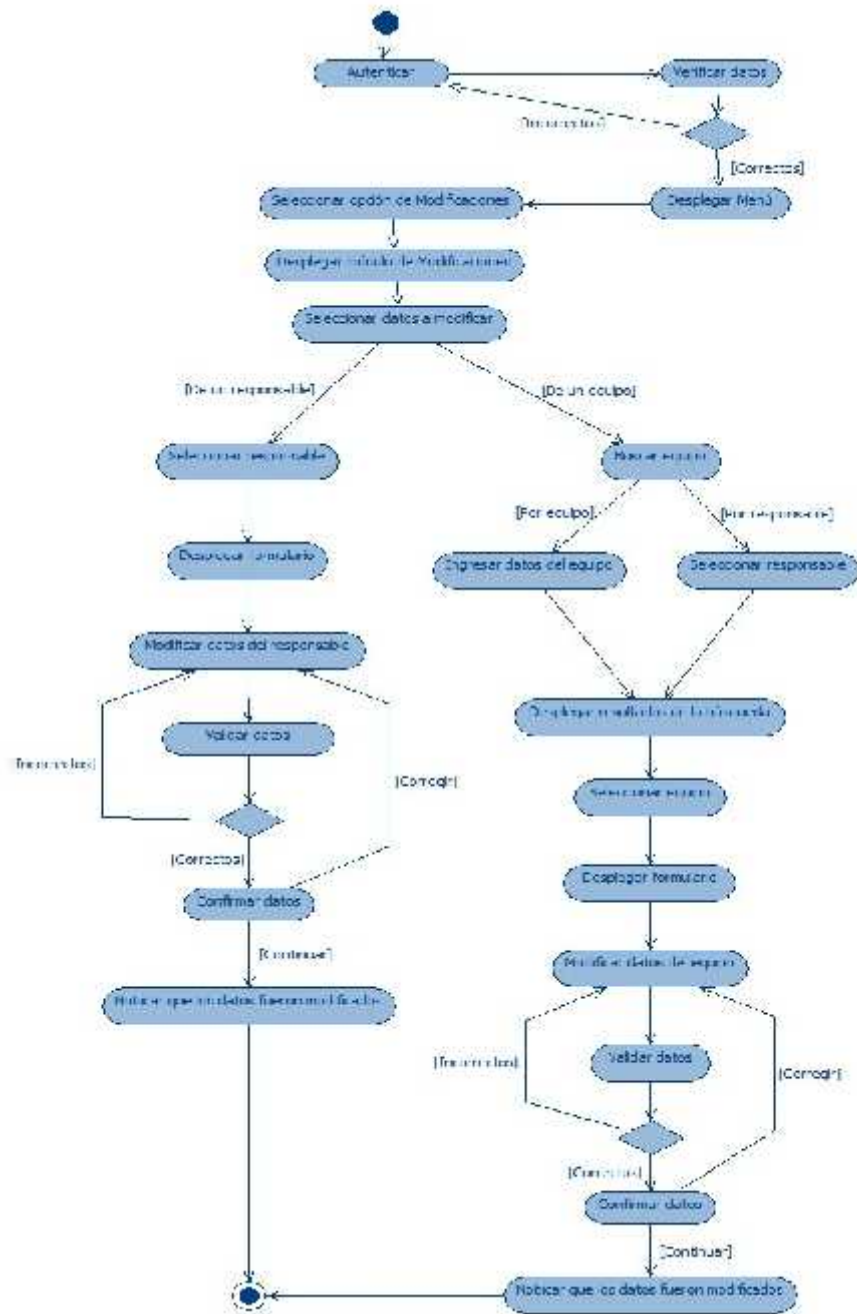


Figura 3.3 Diagrama para editar información de los bienes (modificaciones)

3.3.2.4 Movimientos

El diagrama de la figura 3.4 representa el proceso para el módulo de movimientos, nuevamente en este diagrama también se pueden observar que el proceso se divide en dos ramas la primera

Capítulo 3

Desarrollo

representa el proceso para poner bienes en movimiento y la segunda para armar equipos completos de bienes en movimiento.

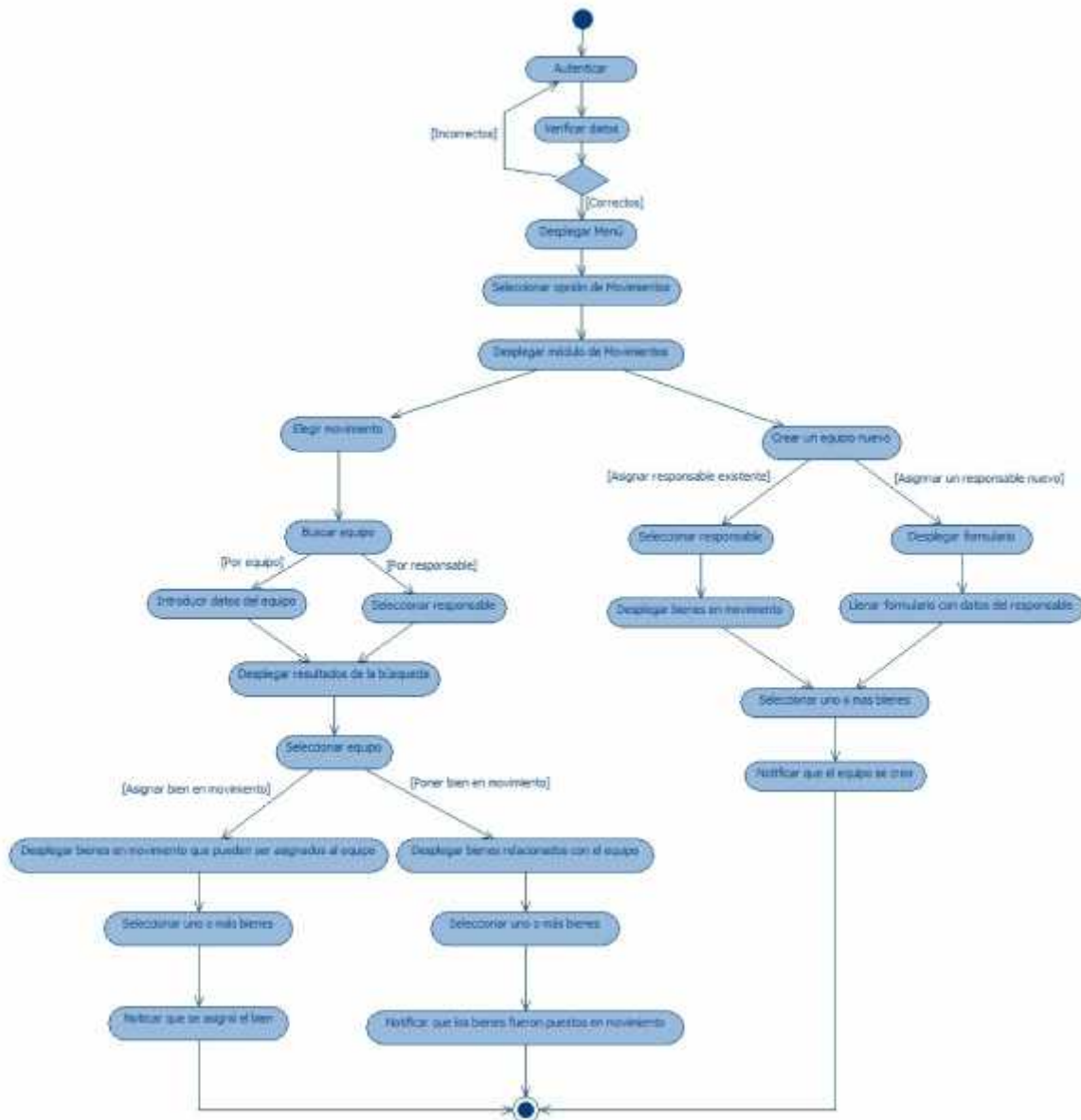


Figura 3.4 Diagrama de movimientos

3.3.2.5 Reportes

El diagrama de la figura 3.5 muestra el proceso para generar reportes varía dependiendo del usuario e implica representar los varios tipos de reportes que se pueden generar.

Capítulo 3

Desarrollo

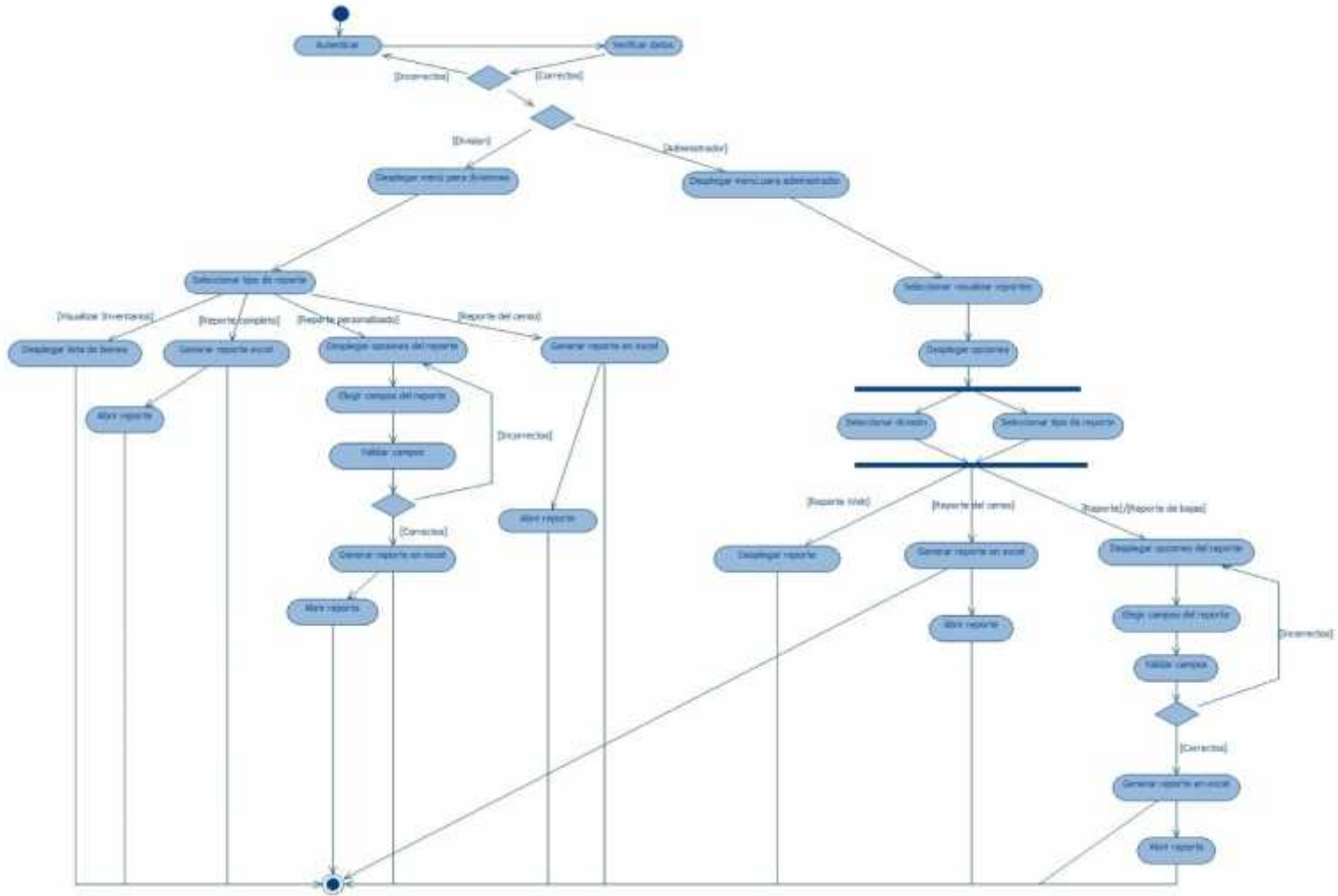


Figura 3.5 Diagrama de reportes

Capítulo 3

Desarrollo

3.3.3 Casos de uso

Un caso de uso especifica un conjunto de acciones que son ejecutadas en un sistema y que llevan a un resultado. Este resultado es normalmente importante para un actor, dicho *actor* representa a un usuario que tiene un rol particular y que interactúa dentro del sistema.

Un caso de uso es una típica interacción entre el usuario y el sistema que se está desarrollando y nos ayudará a capturar la funcionalidad que tiene que proveer el sistema de software.

Antes de comenzar con el análisis de los casos de uso, es fundamental identificar a los actores del sistema. Para este sistema se identifican dos actores:

- *División*. Es cada una de las dependencias especificadas en la tabla 3.6 que tienen asignado un nombre de usuario en el sistema, este tipo de usuario, podrá realizar altas, bajas, modificaciones, movimientos y reportes solo de los bienes que tiene bajo su cargo.
- *Administrador*. Es la persona que podrá obtener reportes de cualquiera de las divisiones dadas de alta en el sistema.

Una vez identificados los actores a continuación se presentan los 13 casos de uso que se obtuvieron del análisis.

0. Autenticación	
Breve descripción:	Verificar que sea un usuario válido y permite acceder al sistema.
Actor:	División
Pre-condiciones:	El usuario debe haber sido de alta en el sistema.
Flujo Principal:	<ol style="list-style-type: none">1. El usuario ingresa su clave.2. El usuario ingresa su contraseña.3. El sistema verifica que el usuario y contraseña sean válidos (Excepción 1).4. El sistema presenta las opciones a realizar de acuerdo con el tipo de usuario.
Flujos de Excepción:	Excepción 1 . En caso de que el usuario y/o la contraseña sean incorrectos, el sistema alerta al usuario y le permite ingresar los datos nuevamente.
Post-condiciones:	La sesión de un usuario deberá de expirar en caso de más de 10 minutos de inactividad.

1. Dar de alta un equipo	
Breve descripción:	Permite dar de alta en el sistema los datos de un equipo.
Actor:	División
Pre-condiciones:	El usuario debe estar autenticado en el sistema.
Flujo Principal:	<ol style="list-style-type: none">1. El usuario elige la opción de Altas en el sistema.2. El usuario elige los bienes que compondrán un equipo, elige uno o más de la tabla 3.1.3. El usuario elige la forma de asignar un responsable al equipo.<ol style="list-style-type: none">a. Asignar un responsable existente. (Alternativo 1)b. Asignar un responsable nuevo (Alternativo 2)4. El sistema valida los datos ingresados (Excepción 2).5. El sistema muestra la página de confirmación de datos y permite:

Capítulo 3

Desarrollo

	<ul style="list-style-type: none"> a. Continuar (Alternativo 3). b. Modificar (Alternativo 4). c. Cancelar (Alternativo 5).
Flujos Alternos:	<p>Alternativo 1</p> <ol style="list-style-type: none"> 1. El sistema despliega los campos correspondientes para dar de alta los datos de cada bien (Excepción 1). 2. El sistema despliega un combo con la lista de responsables dados de alta en el sistema. 3. El usuario ingresa los datos de cada bien que componen el equipo. 4. El usuario elige de la lista de usuarios existentes un responsable para el equipo. 5. El usuario ingresa la ubicación física del equipo. <p>Alternativo 2</p> <ol style="list-style-type: none"> 1. El sistema despliega los campos correspondientes para dar de alta los datos de cada bien (Excepción 1) 2. El sistema despliega los campos para dar de alta los datos del responsable nuevo. 6. El usuario ingresa los datos de cada bien que componen el equipo. 3. El usuario da ingresa el nombre, correo, teléfono y fax del responsable del equipo. 4. El usuario ingresa la ubicación física del equipo. <p>Alternativo 3</p> <ol style="list-style-type: none"> 1. El sistema almacena los datos en la base de datos. 2. El sistema notifica al usuario que los datos han sido dados de alta exitosamente. <p>Alternativo 4</p> <ol style="list-style-type: none"> 1. El sistema despliega los campos de cada bien con los datos que el usuario ingresó. 2. El usuario modifica los campos necesarios. 3. El sistema valida los datos ingresados (Excepción 2): 4. El sistema muestra la página de confirmación de datos y permite: <ul style="list-style-type: none"> a. Continuar (Alternativo 3). b. Modificar (Alternativo 4). c. Cancelar (Alternativo 5). <p>Alternativo 5</p> <ol style="list-style-type: none"> 1. El sistema cancela el proceso, no guarda ningún dato y redirecciona al menú principal
Flujos de Excepción:	<p>Excepción 1. En caso de que el usuario no haya elegido ningún bien, el sistema lo alertará de que debe elegir al menos un bien.</p> <p>Excepción 2. Si alguno de los datos obligatorios no fue ingresado o alguno de los campos no cumple las especificaciones de validación, el sistema envía una alerta al usuario y no permitirá continuar con el proceso.</p>

2. Dar de baja un equipo	
Breve descripción:	Permite eliminar del sistema los datos de un equipo.
Actor:	División
Pre-condiciones:	<p>El equipo que se va a eliminar debe haber sido de alta previamente en el sistema.</p> <p>Para que un usuario (División) pueda dar de baja un bien, dicho bien debe pertenecer a esa División.</p> <p>El usuario debe estar autenticado en el sistema.</p>
Flujo Principal:	<ol style="list-style-type: none"> 1. El usuario elige la opción de Bajas en el sistema. 2. El usuario elige la forma de buscar el bien que desea eliminar. <ul style="list-style-type: none"> a. Buscar por Responsable del equipo (Alternativo 1). b. Buscar por Equipo (Alternativo 2). 3. El sistema elimina el bienes o bienes. 4. El sistema envía un mensaje para confirmar que se han dado de baja el o los bienes elegidos. (Excepción 2).
Flujos Alternos	<p>Alternativo 1</p> <ol style="list-style-type: none"> 1. El sistema despliega una lista de los responsables de la división. 2. El usuario elige un responsable de una lista.

Capítulo 3

Desarrollo

	<ol style="list-style-type: none"> 3. El sistema despliega una lista con los bienes que tiene asignado el responsable elegido. 4. El usuario elige uno de los equipos. 5. El Sistema despliega una lista con todos los bienes que componen el equipo. 6. El usuario elige de la lista uno o más bienes (Excepción 1). <p>Alternativo 2</p> <ol style="list-style-type: none"> 1. El usuario elige de una lista si desea buscar el bien por inventario, número de serie, número de resguardo o modelo. 2. El usuario escribe en una caja de texto el dato que haya elegido. 3. El sistema despliega la lista de los bienes que coinciden con la búsqueda y las opciones para reasignar un responsable si así se deseara. 4. El usuario elige uno de los bienes de la lista. 5. El sistema despliega una lista con todos los bienes que componen el equipo. 6. El usuario elige uno o más bienes (Excepción 1).
Flujos de Excepción:	<p>Excepción 1. Si el usuario no elige ningún bien, el sistema envía un mensaje e impide continuar el proceso.</p> <p>Excepción 2. Si ocurre un problema al dar de baja el o los bienes el sistema manda un mensaje de error.</p>

3. Modificar datos de un bien	
Breve descripción:	Caso de uso que permite modificar los datos de un bien.
Actor:	División
Pre-condiciones:	<p>El bien a modificar debe estar dado de alta en el sistema.</p> <p>Para que un usuario (División) pueda modificar un bien, dicho bien debe pertenecer a esa división.</p> <p>El usuario debe estar autenticado en el sistema.</p>
Flujo Principal:	<ol style="list-style-type: none"> 1. El usuario elige la opción de modificaciones del sistema. 2. El usuario elige la opción para modificar los datos de un equipo. 3. El sistema despliega las opciones para buscar el equipo a modificar. 4. El usuario elige una opción para buscar el equipo. <ol style="list-style-type: none"> a. Por responsable (Alternativo 1). b. Por bien (Alternativo 2).
Flujos Alternos	<p>Alternativo 1</p> <ol style="list-style-type: none"> 1. El sistema despliega la lista de los responsables de la división. 2. El usuario elige uno de los responsables. 3. El sistema despliega una lista de los equipos asociados al responsable elegido y las opciones para poder reasignar un responsable si así se deseara. 4. El usuario elige alguno de los bienes de la lista. 5. El usuario elige la forma de asignar un responsable: <ol style="list-style-type: none"> a. Responsable existente (Alternativo 3). b. Responsable nuevo (Alternativo 4). <p>Alternativo 2</p> <ol style="list-style-type: none"> 7. El usuario elige de una lista si desea buscar el bien por inventario, número de serie, número de resguardo o modelo. 8. El usuario escribe en una caja de texto el dato que haya elegido. 9. El sistema despliega la lista de los bienes que coinciden con la búsqueda y las opciones para reasignar un responsable si así se deseara. 10. El usuario elige uno de los bienes de la lista. 11. El usuario elige la forma de asignar un responsable: <ol style="list-style-type: none"> a. Responsable existente (Alternativo 3). b. Responsable nuevo (Alternativo 3). <p>Alternativo 3</p> <ol style="list-style-type: none"> 1. El sistema despliega los campos con la información actual del bien. 2. El sistema despliega la lista de los responsables de la división con el responsable actual

Capítulo 3

Desarrollo

	<p>seleccionado.</p> <ol style="list-style-type: none"> El usuario cambia los campos que desee. El sistema valida la información (Excepción 1). El sistema muestra la página de confirmación de datos y permite: <ol style="list-style-type: none"> Continuar (Alternativo 5). Modificar (Alternativo 3). Cancelar (Alternativo 6). <p>Alternativo 4</p> <ol style="list-style-type: none"> El sistema despliega los campos con la información actual del bien. El sistema despliega los campos para dar de alta el nuevo responsable. El usuario modifica los campos que desee . El usuario ingresa el nombre, correo, teléfono y fax del responsable nuevo del bien El sistema valida la información (Excepción 1). El sistema muestra la pagina de confirmación de datos y permite: <ol style="list-style-type: none"> Continuar (Alternativo 5). Modificar (Alternativo 4). Cancelar (Alternativo 6). <p>Alternativo 5</p> <ol style="list-style-type: none"> El sistema almacena los datos en la base de datos. El sistema notifica al usuario que los datos han sido modificados exitosamente. <p>Alternativo 6</p> <ol style="list-style-type: none"> El sistema cancela el proceso, no guarda ninguna modificación y redirecciona a la página de modificaciones.
Flujos de Excepción:	Excepción 1. Si alguno de los datos obligatorios no fue ingresado o alguno de los campos no cumple las especificaciones de validación, el sistema envía una alerta al usuario y no permitirá continuar con el proceso.
Post-condiciones:	La sesión de un usuario deberá de expirar en caso de más de 10 minutos de inactividad.

4. Modificar datos de un responsable	
Breve descripción:	Este caso de uso permite modificar los datos de un responsable existente.
Actor:	División
Pre-condiciones:	El responsable debe estar dado de alta en el sistema. El usuario debe estar autenticado en el sistema.
Flujo Principal:	<ol style="list-style-type: none"> El usuario elige la opción de modificaciones del sistema. El usuario elige la opción para modificar los datos de un responsable. El sistema despliega la lista con los responsables de la división. El usuario elige uno de los responsables. El sistema despliega los campos con los datos actuales del responsable. El usuario modifica los campos que desee. El sistema valida la información (Excepción 1). El sistema muestra la pagina de confirmación de datos y permite: <ol style="list-style-type: none"> Continuar (Alternativo 1). Modificar (Alternativo 2). Cancelar (Alternativo 3).
Flujos Alternos	<p>Alternativo 1</p> <ol style="list-style-type: none"> El sistema almacena los datos en la base de datos. El sistema notifica al usuario que los datos han sido modificados exitosamente. <p>Alternativo 2</p> <ol style="list-style-type: none"> El sistema regresa a la página que despliega los campos con los datos que el usuario acaba de ingresar. El usuario modifica los campos que desee. El sistema valida la información (Excepción 1). El sistema muestra la pagina de confirmación de datos y permite:

Capítulo 3

Desarrollo

	<ol style="list-style-type: none"> a. Continuar (Alternativo 1). b. Modificar (Alternativo 2). c. Cancelar (Alternativo 3). <p>Alternativo 3</p> <ol style="list-style-type: none"> 1. El sistema cancela el proceso, no guarda ninguna modificación y redirecciona a la página de modificaciones.
Flujos de Excepción:	Excepción 1. Si alguno de los datos obligatorios no fue ingresado o alguno de los campos no cumple las especificaciones de validación, el sistema envía una alerta al usuario y no permitirá continuar con el proceso.

5. Poner un bien en movimiento	
Breve descripción:	Permite liberar la custodia de un bien a un usuario del sistema.
Actor:	División
Pre-condiciones:	El bien a liberar debe estar dado de alta en el sistema. La custodia del bien a liberar debe pertenecer al usuario que se autentico en el sistema. El usuario debe estar autenticado en el sistema.
Flujo Principal:	<ol style="list-style-type: none"> 1. El usuario elige la opción de movimientos. 2. El sistema despliega una lista con las opciones de movimientos y el tipo de búsqueda del bien 3. El usuario elige la opción de “Poner un bien en movimiento”. 4. El usuario elige el tipo de búsqueda. <ol style="list-style-type: none"> a. Por responsable (Alternativo 1). b. Por equipo (Alternativo 2). 5. El sistema pone en movimiento el bien. 6. El sistema envía un mensaje al usuario para informar que los bienes han sido puestos en movimiento (Excepción 2).
Flujos Alternos	<p>Alternativo 1</p> <ol style="list-style-type: none"> 1. El sistema despliega una lista con los responsables de la división. 2. El usuario elige uno de los responsables. 3. El sistema despliega una lista con los equipos asociados al responsable seleccionado. 4. El usuario elige uno de los equipos. 5. El sistema despliega una lista con los bienes asociados al equipo elegido. 6. El usuario elige uno o más bienes de la lista (Excepción 1). <p>Alternativo 2</p> <ol style="list-style-type: none"> 1. El usuario elige el tipo de bien que desea poner en movimiento. 2. El usuario elige si va a buscar por número de inventario o por número de serie. 3. El usuario escribe el número de inventario o de serie del bien. 4. El sistema despliega la lista con las coincidencias de la búsqueda. 5. El usuario elige una opción de la lista. 6. El sistema despliega una lista con todos los bienes que componen el equipo. 7. El usuario elige uno o más bienes (Excepción 1).
Flujos de Excepción:	Excepción 1. Si el usuario no elige ningún bien, el sistema envía un mensaje e impide continuar el proceso. Excepción 2. Si ocurre un problema al dar de baja el o los bienes el sistema manda un mensaje de error.
Post-condiciones:	El bien sigue existiendo en la base de datos pero no pertenece a ninguna división por lo tanto no puede ser dado de baja ni hacer modificaciones en sus datos hasta que sea asignado nuevamente a una división.

6. Asignar un bien en movimiento	
Breve descripción:	Este caso de uso permite asignar bienes en movimiento a un responsable o a un equipo ya armado, esto significa que el bien asignado pertenecerá a la división (usuario) que se autenticó en

Capítulo 3

Desarrollo

	el sistema.
Actor:	División
Pre-condiciones:	El usuario debe estar autenticado en el sistema. El bien a asignar no debe pertenecer a ninguna división, es decir, debe estar en movimiento. El responsable o equipo al que se vaya a asignar debe existir en el sistema.
Flujo Principal:	<ol style="list-style-type: none"> 1. El usuario elige la opción de movimientos. 2. El sistema despliega una lista con las opciones de movimientos y las opciones para asignar los bienes. 3. El usuario elige la opción de “Asignar un bien en movimiento”. 4. El usuario elige como asignará el bien: <ol style="list-style-type: none"> a. A un responsable (Alternativo 1). b. A un equipo (Alternativo 2). 5. El usuario elige uno o más bienes que serán asignados al equipo (Excepción 1). 6. El sistema envía un mensaje para informar al usuario que los bienes se han asignado correctamente.
Flujos Alternos	<p>Alternativo 1</p> <ol style="list-style-type: none"> 7. El sistema despliega una lista con todos los responsables de la división. 8. El usuario elige uno de los responsables de la lista. 9. El sistema despliega la lista de equipos que el responsable tiene bajo su cargo. 10. El usuario elige uno de los equipos. 11. El sistema despliega todos los bienes relacionados con el equipo y una lista con los bienes en movimiento que se le pueden asignar a dicho equipo. <p>Alternativo 2</p> <ol style="list-style-type: none"> 1. El usuario elige si va a buscar el equipo por inventario, número de serie, número de resguardo o modelo. 2. El usuario escribe en la caja de texto el número de la opción que eligió. 3. El sistema despliega la lista de equipos que coincidan con la búsqueda. 4. El usuario elige uno de los equipos. 5. El sistema despliega los bienes que componen el equipo y la lista de bienes en movimiento que pueden ser asignados a dicho equipo.
Flujos de Excepción:	Excepción 1. Si el usuario no elige ningún bien el sistema envía un mensaje de error e impide continuar el proceso.

7. Crear un nuevo equipo	
Breve descripción:	Este caso de uso permite armar un equipo nuevo con bienes que están en movimiento, es decir, bienes que no pertenecen a ninguna división. Se puede asignar un responsable existente o uno nuevo.
Actor:	División
Pre-condiciones:	El usuario debe estar autenticado en el sistema. Los bienes con lo que se creará el equipo nuevo deben estar en movimiento.
Flujo Principal:	<ol style="list-style-type: none"> 1. El usuario elige la opción de movimientos. 2. El usuario va a la sección de Crear un Equipo Nuevo y selecciona Armar equipo. 3. El sistema despliega las opciones para asignar un responsable al nuevo equipo. 4. El usuario elige una de las opciones: <ol style="list-style-type: none"> a. Responsable existente (Alternativo 1). b. Responsable nuevo (Alternativo 2). 5. El sistema despliega la lista con todos los bienes que se encuentran en movimiento. 6. El usuario elige al menos un bien de cada tipo (Excepción 1). 7. El sistema almacena los datos en la base de datos e informa al usuario que el equipo se creó satisfactoriamente.
Flujos Alternos:	<p>Alternativo 1</p> <ol style="list-style-type: none"> 1. El sistema despliega la lista de responsables existentes. 2. El usuario elige el responsable e indica la ubicación del nuevo equipo.

Capítulo 3

Desarrollo

	Alterno 2 <ol style="list-style-type: none">1. El sistema despliega los campos para ingresar los datos del nuevo responsable.2. El usuario ingresa el nombre, teléfono, fax y correo electrónico del responsable.3. El usuario ingresa la ubicación del nuevo equipo.4. El sistema valida los datos (Excepción 2).
Flujos de excepción:	Excepción 1. Si el usuario no elige ningún bien, el sistema alertará al usuario y no permitirá continuar el proceso. Excepción 2. Si alguno de los datos obligatorios no fue ingresado o alguno de los campos no cumple las especificaciones de validación, el sistema envía una alerta al usuario y no permitirá continuar con el proceso.

8. Visualizar inventarios	
Breve descripción:	Permite visualizar la información completa de todos los bienes que pertenecen a la división.
Actor:	División
Pre-condiciones:	El usuario debe estar autenticado en el sistema. El usuario (división) debe tener bajo su custodia algún bien.
Flujo Principal:	<ol style="list-style-type: none">1. El usuario elige la opción de visualizar Inventarios2. El sistema despliega una lista con los datos de los bienes bajo la custodia del usuario.3. Al final de la lista el sistema presenta la opción de generar un reporte en Excel con la información que acaba de ser presentada. (Alterno 1).
Flujos Alternos:	Alterno 1 <ol style="list-style-type: none">1. El usuario elige la opción para generar el reporte en Excel.2. El sistema genera el reporte y presenta la opción para descargarlo.3. El usuario elige la opción de descargar.4. El sistema abre el reporte.
Flujos de Excepción:	Ninguno

9. Reporte Completo	
Breve descripción:	Permite generar en un reporte de Excel la información completa de todos los bienes que pertenecen a esa división.
Actor:	División
Pre-condiciones:	El usuario debe estar autenticado en el sistema. El usuario debe tener bienes bajo su cargo . Se debe tener instalado algún software que permita visualizar hojas de cálculo.
Flujo Principal:	<ol style="list-style-type: none">1. El usuario elige la opción de para generar el reporte.2. El sistema genera el reporte y presenta la opción para descárgalo.3. El usuario elige la opción de descargar.4. El sistema abre el reporte.
Flujos de Excepción:	Ninguno
Post-condiciones	El reporte queda almacenado por unos días en el servidor. Para conservar el reporte el usuario debe guardar el documento en su máquina local.

10. Reporte Personalizado	
Breve descripción:	Este caso de uso permite especificar que datos de los bienes se necesitan generar en el reporte. Funciona como filtro al permitir al usuario, definir que campos deberán aparecer en el reporte.
Actor:	División
Pre-condiciones:	El usuario debe estar autenticado en el sistema. El usuario debe tener bienes bajo su cargo.

Capítulo 3

Desarrollo

	Se debe tener instalado algún software que permita visualizar hojas de cálculo.
Flujo Principal:	<ol style="list-style-type: none">1. El usuario elige la opción para generar un reporte personalizado.2. El sistema despliega la lista de los campos que puede contener el reporte.3. El usuario elige uno o más campos (Excepción 1).4. El sistema genera el reporte y presenta la opción para descargar el reporte.5. El usuario elige descargar el reporte.6. El sistema abre el reporte.
Flujos de Excepción:	Excepción 1. Si el usuario no elige ningún campo, el sistema enviará una alerta para informar al usuario que debe elegir al menos un campo e impedirá continuar el proceso.
Post-condiciones	El reporte queda almacenado por unos días en el servidor. Para conservar el reporte el usuario debe guardar el documento en su máquina local.

11. Reporte del Censo	
Breve descripción:	Permite generar un reporte que contiene información resumida y contabilizada de acuerdo a los siguientes criterios: Por tipo de máquinas. Por tipo de usuario. Por tipo de sistema operativo. Por tipo de conexión. Por equipos de impresión. Por equipos digitales. Por equipos de telecomunicaciones.
Actor:	División
Pre-condiciones:	El usuario debe estar autenticado en el sistema. El usuario debe tener bienes bajo su cargo. Se debe tener instalado algún software que permita visualizar hojas de cálculo.
Flujo Principal:	<ol style="list-style-type: none">1. El usuario elige la opción de Reporte del censo.2. El sistema genera el reporte con el censo y presenta la opción de descargarlo.3. El usuario elige la opción de descargar.4. El sistema abre el reporte.
Flujos de Excepción:	Ninguno
Post-condiciones:	El reporte queda almacenado por unos días en el servidor Para conservar el reporte el usuario debe guardar el documento en su máquina local

12. Obtener reportes	
Breve descripción:	Permite visualizar y generar diversos tipos de reportes de los bienes de cualquier división. Estos reportes pueden ser completos o personalizados de los bienes que están dados de alta o que han sido dados de baja, así mismo permite crear reportes de tipo censo que entregan información resumida y contabilizada de los bienes de las divisiones.
Actor:	Administrador
Pre-condiciones:	El usuario debe estar autenticado en el sistema. Se debe tener instalado algún software que permita visualizar hojas de cálculo.
Flujo Principal:	<ol style="list-style-type: none">1. El usuario elige la opción de Visualizar Inventarios.2. El sistema despliega distintas agrupaciones de divisiones para facilitar la búsqueda al usuario.3. El usuario elige la agrupación en la que le conviene buscar para generar el reporte que desea.<ol style="list-style-type: none">a. Reportes por cada división (Alternativo 1).b. Secretaría General (Alternativo 1).c. Unidad de Servicios de Cómputo Académico (Alternativo 1).d. Facultad de Ingeniería (Alternativo 2).

Capítulo 3

Desarrollo

	<ul style="list-style-type: none">e. Secretaría de Apoyo a la Docencia (Alternativo 1).f. Coordinación de Vinculación Productiva y Social (Alternativo 1).
Flujos Alternos:	<p>Alternativo 1</p> <ul style="list-style-type: none">1. El usuario elige de una lista la división de la que desea ver o crear el reporte.2. El usuario elige el tipo de reporte que desea generar:<ul style="list-style-type: none">a. Reporte de bajas (Alternativo 3).b. Reporte Excel (Alternativo 3).c. Reporte Web (Alternativo 4).d. Reporte Censo (Alternativo 5). <p>Alternativo 2</p> <ul style="list-style-type: none">1. El usuario elige el tipo de reporte que desea generar:<ul style="list-style-type: none">a. Reporte Bajas.b. Reporte Altas.2. El sistema genera un reporte que contiene el número total de bienes de cada división y presenta la opción de descargar.3. El usuario elige descargar el reporte.4. El sistema abre el reporte. <p>Alternativo 3</p> <ul style="list-style-type: none">1. El sistema presenta la lista de campos que puede contener el reporte.2. El usuario elige uno o más campos (Excepción 1).3. El sistema genera el reporte y presenta la opción para descargar el reporte.4. El usuario elige descargar el reporte.5. El sistema abre el reporte. <p>Alternativo 4</p> <ul style="list-style-type: none">1. El sistema despliega en el explorador una lista con los datos de todos los bienes de la división elegida. <p>Alternativo 5</p> <ul style="list-style-type: none">1. El sistema genera el reporte que censa el número de bienes de la división y presenta la opción para descargar el reporte.2. El usuario elige descargar el reporte.3. El sistema abre el reporte.
Flujos de Excepción:	<p>Excepción 1. Si el usuario no elige ningún campo, el sistema enviará una alerta para informar al usuario que debe elegir al menos un campo e impedirá continuar el proceso.</p>
Post-condiciones:	<p>Los reportes quedan almacenados por unos días en el servidor. Para conservar el reporte el usuario debe guardar el documento en su máquina local.</p>

Capítulo 3

Desarrollo

3.3.4 Diagrama de casos de uso

El diagrama de la Figura 3.6 permite describir las relaciones entre el conjunto de casos de uso y los actores que participan en estos casos de uso. Se muestra como quedaron integrados los actores con sus respectivos casos de uso dentro del SICICE.

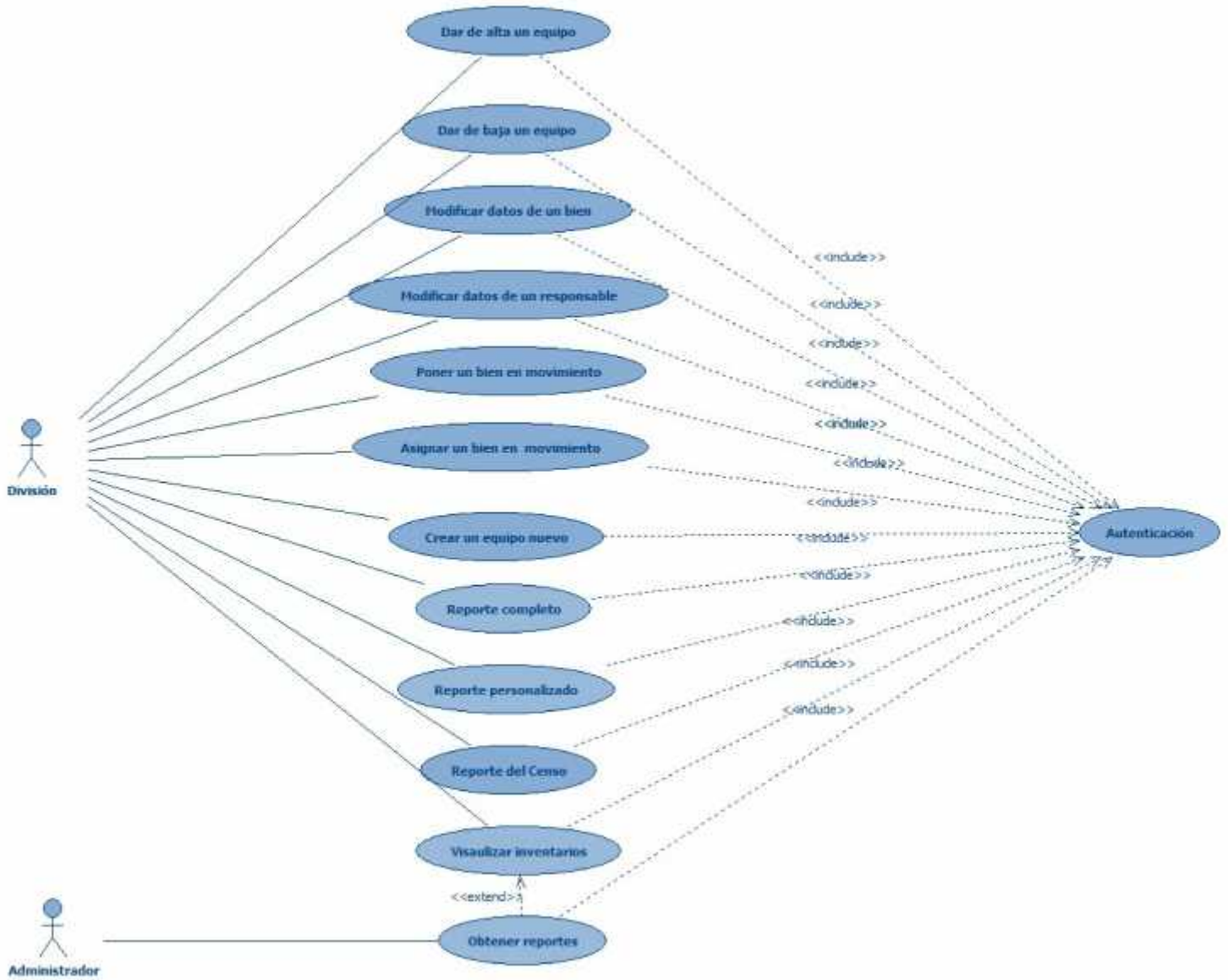


Figura 3.6 Relación de casos de uso y los actores

Capítulo 3

Desarrollo

3.4 Diseño basado en componentes

En los últimos años la ingeniería basada en componentes se ha adoptado como una solución para reducir el ciclo de vida del software, reducir costos, incrementar la productividad, facilitar la integración y permitir mayor flexibilidad. De hecho muchas aplicaciones web en la actualidad están desarrolladas con software libre y/o con tecnologías propietarias permitiendo mejoras en el proceso de desarrollo, sin embargo la ingeniería basada en componentes puede, de hecho, implicar gran complejidad y por lo tanto dificultar el proceso de implementación y liberación de los sistemas.

Para evitar riesgos y la complejidad que puede llegar a representar la implementación de un diseño basado en componentes es importante elegir y definir cuidadosamente los componentes. En esta sección más que definir los componentes de la aplicación la cual en si misma se puede representar como un conjunto de componentes (por estar basada en un paradigma basado en componentes), se expone la arquitectura del sistema completo a través de la representación de sus componentes y la interacción de estos, al mismo tiempo esta representación permite tener una completa perspectiva de la implementación del sistema.

3.4.1 Componentes del sistema

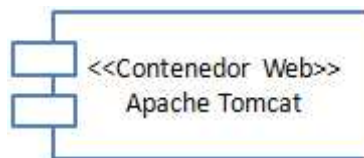
- Apache Tomcat

De acuerdo a la descripción de tipos de componentes del capítulo dos Apache Tomcat es un componente de distribución; es al mismo tiempo un contenedor de servlets y es un servidor Web, por lo que permite traducir los JSP a servlets para que después puedan ser compilados y gestiona las peticiones http de los clientes.

Ventajas

- Cumple con las características de estandarización.
- Al estar escrito en Java lo hace compatible para su uso.
- Cuenta con una amplia documentación.
- Es un servidor de aplicaciones experimentado.

Este componente puede ser sustituido por componentes similares como Glashfish, JBoss, WebLogic.¹¹



- PostgreSQL

¹¹ Estos programas son también conocidos como servidores de aplicaciones y son más robustos que Apache Tomcat pues además de ser contenedores de servlets son contenedores EJB.

Capítulo 3

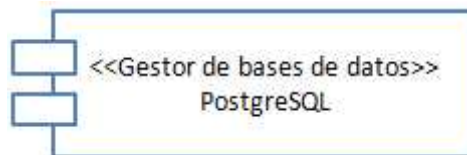
Desarrollo

Este es otro componente de distribución, un manejador de bases de datos del cual, en el capítulo dos, se describieron las características por las que fue elegido.

Ventajas

- Este componente está estandarizado.
- Independiente ya que no necesita ningún componente adicional para funcionar.
- La documentación es extensa y pública.
- Multiplataforma.
- Se puede extender su funcionalidad.

Es un componente que puede ser sustituido por componentes similares como MySQL, Oracle o Microsoft SQL Server.



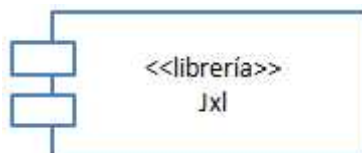
- Jxl

Jxl es del tipo de componentes para trabajar en el producto; es un API de código libre escrita en Java que permite escribir, leer y modificar hojas de cálculo, es un componente independiente ya que no requiere librerías adicionales, al estar escrito en Java se apega a los estándares del lenguaje incluyendo el soporte de documentación.

Ventajas

- Soporta la generación de graficas.
- Compatibilidad con Excel 95, 97, 98, 2000, XP, 2003 y 2007.
- Soporta formato para números y fechas.

Existen componentes similares como Jasper, sin embargo este componente fue elegido porque cumple con las con las características deseables de un componente de software y por sus facilidad de uso.



Capítulo 3

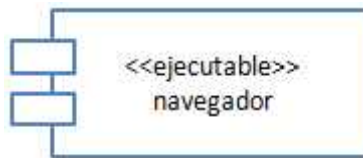
Desarrollo

- Explorador

Este es un componente de ejecución, un explorador es el componente básico que permite que el sistema se ejecute, permite enviar y recibir las peticiones y respuestas del servidor.

Debido a que el código html del sistema está desarrollado de acuerdo a los estándares de la W3C¹², este debe poder ser ejecutado en cualquier navegador, de esta manera se cumple con el requisito de estandarización.

Un navegador es un componente independiente, componible y desplegable.



- Conector JDBC

Un conector JDBC es el componente de software que permite a una aplicación Java interactuar con una base de datos. Para este sistema se utilizó específicamente el conector JDBC para PostgreSQL: postgresql-jdbc.

Este componente de distribución a pesar de depender de otros componentes (el manejador de la base de datos) cumple con ser, estandarizado y documentado.



- Hoja de Cálculo

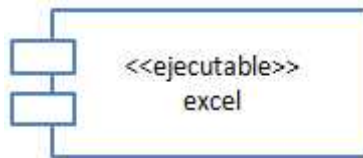
Este es otro componente de ejecución del sistema ya que permite visualizar los reportes que genera el sistema. Por lo general se utiliza Excel, sin embargo éste puede ser sustituido por cualquier componente similar.

¹² El Consorcio World Wide Web (W3C) es una comunidad internacional donde las organizaciones, personal a tiempo completo y el público en general trabajan conjuntamente para desarrollar estándares Web. Liderado por el inventor de la Web Tim Berners-Lee y el Director Ejecutivo (CEO) Jeffrey Jaffe, la misión del W3C es guiar la Web hacia su máximo potencial.

Capítulo 3

Desarrollo

Excel cumple con las características deseables de un componente ya que está estandarizado, independiente, componible y desplegable.



- Páginas JSP

Este es un componente para trabajar en el producto, consiste en varias páginas JSP que contienen código que fue específicamente desarrollado para realizar las principales funciones del sistema (altas, bajas, modificaciones y movimientos).

Una vez que el sistema ha sido desplegado, este componente sólo requiere para su ejecución un navegador web, no requiere ningún componente adicional para realizar sus funciones básicas, es necesario sin embargo, contar con un componente de hoja de cálculo si se desea obtener los reportes en dicho formato.

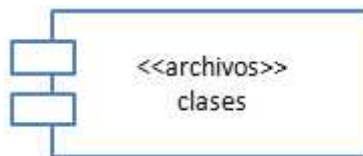
El componente es componible ya que cada una de sus interfaces están definidas públicamente, existen métodos y atributos que permiten acceder a la información contenida por el componente. Es desplegable ya que se puede ejecutar en cualquier máquina con conexión a internet y un navegador web, elementos prácticamente inherentes en cualquier computadora. No es necesario compilarlo o llevar a cabo un proceso adicional para que funcione.

Finalmente se puede asegurar que es un componente documentado, pues se cuenta con los manuales de usuario y de programador.



- Clases

Este es otro componente importante, todas estas clases se refieren a componentes de distribución, pues consisten en Java Beans y código de librerías de apoyo para realizar las funciones de las páginas JSP.



Capítulo 3

Desarrollo

3.4.2 Diseño de los componentes

El siguiente diagrama (Figura 3.7) muestra las relaciones entre cada uno de los componentes, se agrupan en paquetes de acuerdo a su ubicación física, es decir, los componentes del sistema se distribuyen en tres diferentes máquinas: un servidor de bases de datos, un servidor de aplicaciones y la computadora del cliente.

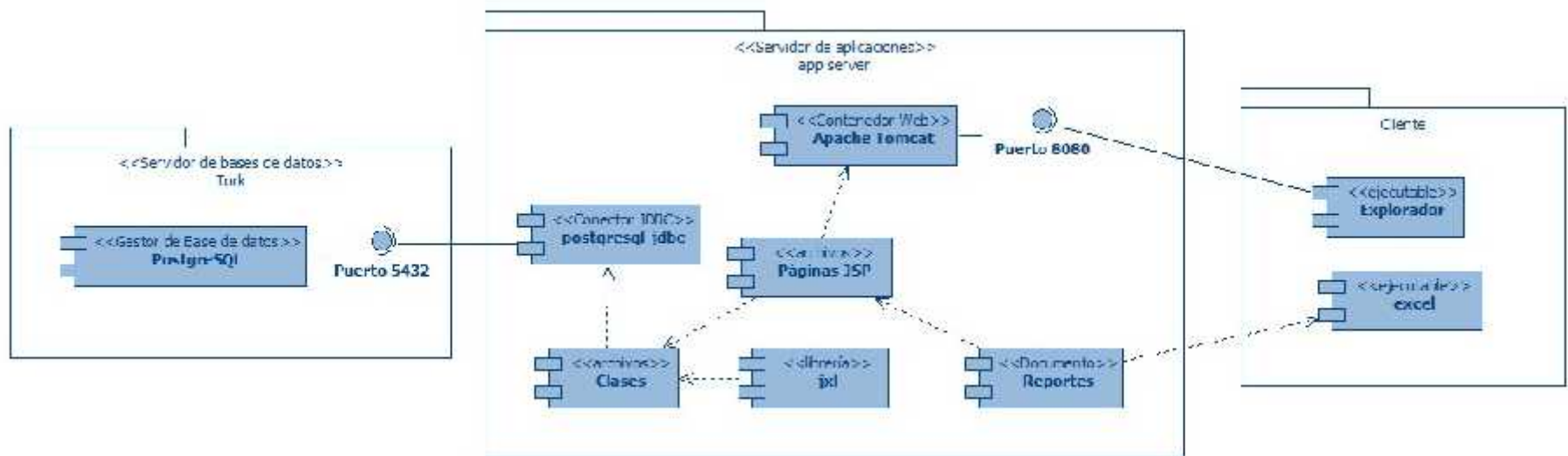


Figura 3.7 Diseño de componentes

Capítulo 3

Desarrollo

3.4.3 Flujo de pantallas

Esta sección tiene como propósito representar los cambios en el sistema como respuesta a los sucesos y al tiempo, esto permite observar cómo reacciona el sistema a determinadas acciones del usuario.

A diferencia de lo que se observa en los diagramas presentados anteriormente que representan relaciones y asociaciones entre los objetos o, en el caso de los diagramas de actividades de manera abstracta, el comportamiento del sistema, el flujo de pantallas muestra explícitamente lo que el usuario verá como resultado de sus acciones. Estos diagramas junto con los casos de uso tienen como fin lograr una mejor comunicación con el cliente y por lo tanto un mejor acercamiento a los requerimientos.

Para representar el flujo de pantallas se usan Diagramas de Estado los cuales permitirán mostrar el punto inicial y final de la secuencia de cambios de estados y transacciones del sistema. En el primer diagrama (Figura 3.8) muestra un panorama general del flujo de pantallas del sistema, en los diagramas siguientes se mostrará el funcionamiento de cada uno de las submáquinas de estados que se observan en este diagrama.

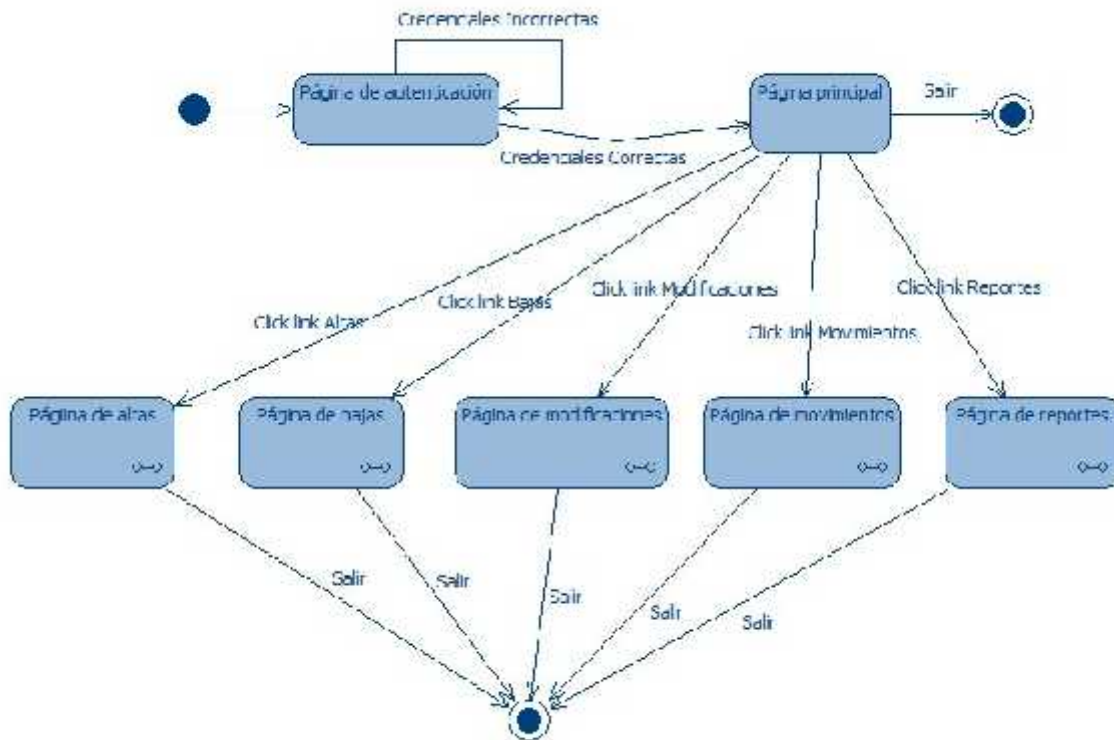


Figura 3.8 Diagrama general del flujo de pantallas

Capítulo 3

Desarrollo

Los flujos de pantalla correspondiente al modulo de altas, bajas, modificaciones, movimientos, reportes por Divisiones y del administrador se presentan en Anexo 1 Flujos de pantallas.

3.4.4 Diagramas de secuencias

Los diagramas de secuencias permiten visualizar la forma en cómo se comunican los objetos a través del tiempo. A diferencia de los diagramas de estados que también representan el flujo del sistema a través del tiempo, los diagramas de secuencias, muestran un nivel más detallado y técnico que permite a los desarrolladores implementar las operaciones del sistema.

En la figura 3.9 se presenta el diagrama de secuencia de visualizar inventarios para usuario División.

Los diagramas de secuencia correspondiente al modulo de altas, bajas, modificaciones, movimientos y reportes, se presentan en Anexo 2 Diagramas de secuencias.

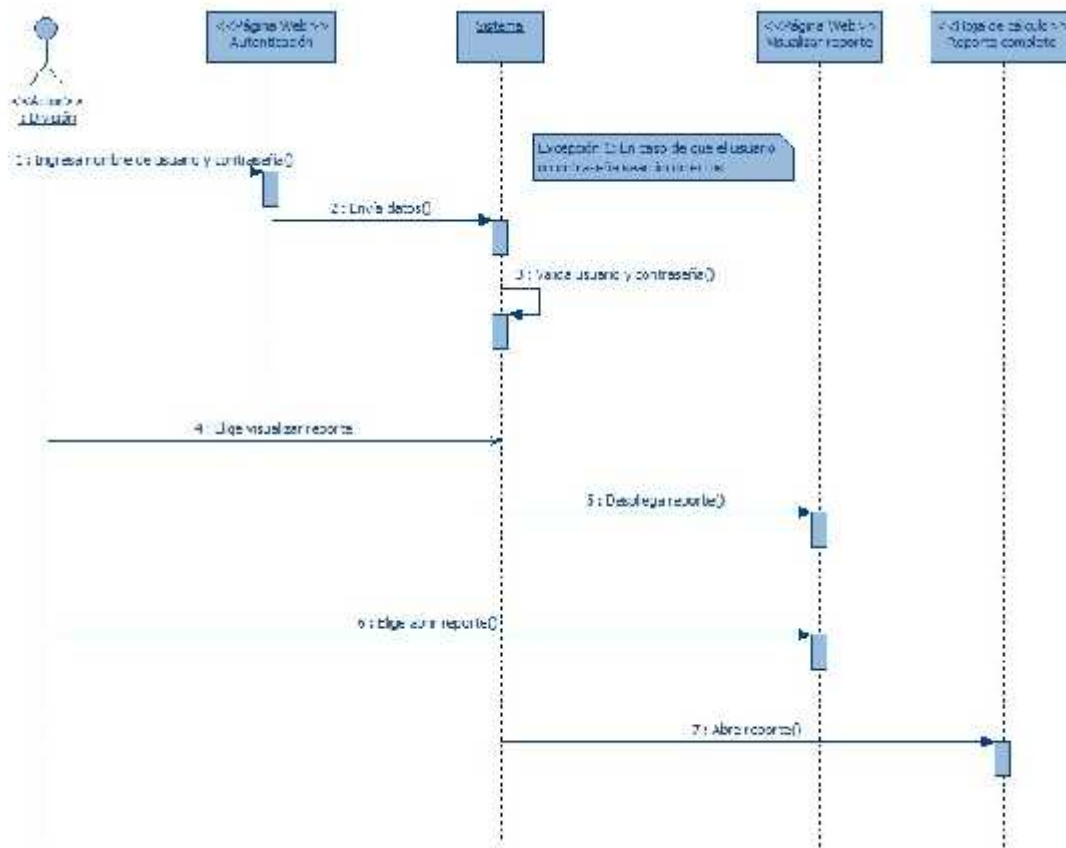


Figura 3.9 Visualizar inventarios

Capítulo 3

Desarrollo

3.4.5 Diccionario de datos

Un diccionario de datos es una lista de nombres ordenada alfabéticamente incluida en los modelos de los sistemas con el fin de complementarlos y explicarlos; debe reunir la descripción de cada dato utilizado en el sistema y sus atributos.

El siguiente diccionario permite gestionar, estandarizar y comprender la información de todos los diagramas presentados para este sistema. La tabla 3.8 muestra el nombre de cada entrada del diccionario junto con su descripción y el tipo, es decir, si es una entidad, relación, atributo o servicio.

Nombre	Descripción	Tipo
Administrador	Es el tipo de usuario que tiene privilegios para crear todo tipo de reportes de todas las divisiones.	Entidad
Alta	Es la operación que realiza un usuario en el sistema al ingresar los datos de un equipo en el sistema.	Servicio
Autenticar	Es el proceso que lleva a cabo el sistema para verificar las credenciales de un usuario y validar que existe.	Servicio
Aplicación	Es el conjunto archivos que permiten desplegar las páginas WEB con las que el usuario interactúa para realizar alguna operación en el sistema.	Entidad
Baja	Es una operación que realiza el usuario cuando elimina del sistema los datos de un bien o equipo.	Servicio
Bien	En el contexto del sistema un bien es todo aquel elemento físico de cómputo que cuenta con un número de inventario UNAM. En la tabla 3.1 de este capítulo se describe cada uno de los bienes que soporta el sistema.	Entidad
Campo	Un campo puede referirse a los campos de un reporte, es decir, cada una de las columnas que componen la tabla de un reporte o el campo de un formulario, que son las cajas de texto, combos o check boxes que este pueda contener.	Atributo
Censo	El censo o reporte del ceso, es un reporte especial que extrae y contabiliza por determinadas categorías los bienes de una división.	Entidad
Cliente	Se refiere a cada máquina donde corre la aplicación.	Entidad
Conector JDBC	Es un componente de software que permite que la aplicación interactúe con la base de datos.	Entidad
Contenedor Web	Es el componente de software que permite traducir las páginas JSP y gestionar las peticiones web de los clientes.	Entidad
División	Es el tipo de usuario que se asigna a cada una de las dependencias o divisiones de la Facultad. En la tabla 3.6 se muestran cada una de las dependencias que tienen asignado un usuario de este tipo.	Entidad

Capítulo 3

Desarrollo

Equipo	Un equipo está compuesto de uno o más bienes, por ejemplo: un equipo puede estar compuesto por un CPU, un Monitor y un Teclado.	Entidad
Excepción	Una excepción es una interrupción en el flujo normal de una operación, estas excepciones son manejadas por la aplicación para que se informe al usuario de que debe realizar determinada acción para continuar con el flujo normal.	Servicio
Formulario	Es el conjunto de elementos como cajas de texto, combos, radio botones, y check boxes que permiten al usuario manejar los datos de los elementos del sistema como bienes y responsables.	Entidad
Gestor de Bases de datos	Es el componente de software que permite almacenar los datos de los bienes, responsables y usuarios del sistema.	Entidad
Hoja de cálculo	Es el componente de software que permite desplegar y/o exportar los reportes.	Entidad
Librería	Es el conjunto de clases que realizan determinadas funciones en el sistema.	Entidad
Lista	Es el conjunto de datos que se despliegan en forma de lista en una página WEB. Estas listas son por lo general de equipos o bienes.	Entidad
Menú	Es la página principal de la aplicación, en esta página se encuentra un menú con todas las operaciones que puede realizar un usuario dentro del sistema.	Entidad
Modificación	Es una operación que realiza un usuario dentro del sistema para modificar datos de los bienes o responsables.	Servicio
Módulo	Es el conjunto de páginas WEB que permiten realizar determinada operación dentro del sistema, por ejemplo el módulo de Altas o el módulo de Modificaciones.	Entidad
Movimiento	Es la operación que un usuario lleva a cabo para crear un equipo a base de bienes que no están asignados a ninguna división, asignar un bien a un equipo o liberar un bien de la responsabilidad de una división.	Servicio
Opción	Una opción se refiere a cada una de las operaciones que puede elegir el usuario para realizar dentro del sistema.	Relación
Operación	Es una acción el usuario realiza dentro del sistema y que altera el estado de los datos del sistema. Existen las operaciones de Altas, Bajas, Modificaciones y Movimientos.	Servicio
Página	Una página WEB o simplemente página es la interfaz con la que interactúa el usuario con el sistema.	Entidad
Puerto	Es un canal de comunicación entre dos componentes del sistema.	Relación

Capítulo 3

Desarrollo

Reporte	Es el conjunto de datos de bienes de una o varias divisiones que se despliegan de manera organizada en una Lista u hoja de cálculo. Existen varios tipos de reportes dependiendo de la información que muestran o de la forma en que se despliegan.	Entidad
Reporte Completo	Un reporte completo contiene toda la información que se puede obtener de los bienes de una división (todos los campos). Se despliega a través de una hoja de cálculo.	Entidad
Reporte de Bajas	Este tipo de reporte solo despliega los bienes que han sido dados de baja por determinada división. Se despliega a través de una hoja de cálculo.	Entidad
Reporte Personalizado	Este tipo de reporte permite seleccionar los datos (campos) que se quieren extraer de los bienes. Se despliega a través de una hoja de cálculo.	Entidad
Reporte Web	Este tipo de reporte contiene los principales datos de los bienes y se despliega en una página WEB.	Entidad
Responsable	El responsable es la persona que está a cargo que utiliza un bien.	Atributo
Servidor	Un servidor es el equipo físico donde correrán los componentes de software del sistema. Para este sistema existen dos tipos de servidores, el de aplicaciones y el de bases d datos.	Entidad
Sistema	Es el conjunto de componentes de software y equipo físico que forman permiten almacenar y manipular los datos de los bienes, responsables y usuarios.	Entidad
Usuario	Un usuario es la entidad que puede modificar los datos del sistema. Cada usuario tiene asignada una contraseña con la cual se le permite realizar determinadas operaciones dentro del sistema. No debe confundirse a un usuario con un responsable, ya que un responsable es uno más de los datos que un usuario puede manipular.	Entidad
Tipo de usuario	Existen dos tipos de usuario: Administrador y División, los privilegios de cada uno obedecen principalmente en el tipo de información que el usuario puede extraer del sistema.	Atributo
Validar	Es el proceso mediante el cual es sistema comprueba que determinados datos que están por ingresar al sistema cumplen con determinadas características para mantener la consistencia de los datos.	Servicio

Tabla 3.8 Diccionario de datos

Capítulo 3

Desarrollo

3.4.6 Tecnologías de programación a implementar

En el capítulo 2 se describieron las herramientas de desarrollo para el proyecto, entre las que destaca el lenguaje de programación Java, en esta sección profundizaremos en Java EE que es la columna vertebral de este proyecto que, entre otras razones que se detallarán a continuación, fue elegido por ajustarse al modelo con que fue diseñado el sistema (Modelo Orientado a Objetos).

Además de Java EE también se hablará de otras dos tecnologías que sirven como soporte: Javascript y XML.

3.4.6.1 Java EE

El modelo de Java EE comienza con el lenguaje de programación Java y la máquina virtual de Java. La portabilidad, seguridad y productividad para desarrollo que provee forman parte las bases de este modelo de aplicación.

Este modelo define una arquitectura para implementar los servicios como aplicaciones multicapa que permiten desplegar la escalabilidad, accesibilidad y facilidad de administración necesarias para aplicaciones de alto nivel. El modelo divide el trabajo en dos partes:

- a) La lógica del negocio y presentación implementadas por el desarrollador.
- b) Los servicios estándar que provee la plataforma Java EE.

La plataforma Java EE utiliza un modelo de aplicación multiniveles distribuido. La lógica de la aplicación está dividida en componentes de acuerdo a sus funciones.

En la Figura 3.10 se muestran como dos aplicaciones Java EE divididas en los siguientes niveles:

- *Nivel del Cliente*, componentes que corren en la máquina del cliente.
- *Nivel Web*, componentes que corren en el servidor Java EE.
- *Nivel de Negocio*, componentes que corren en el servidor Java EE.

Capítulo 3

Desarrollo

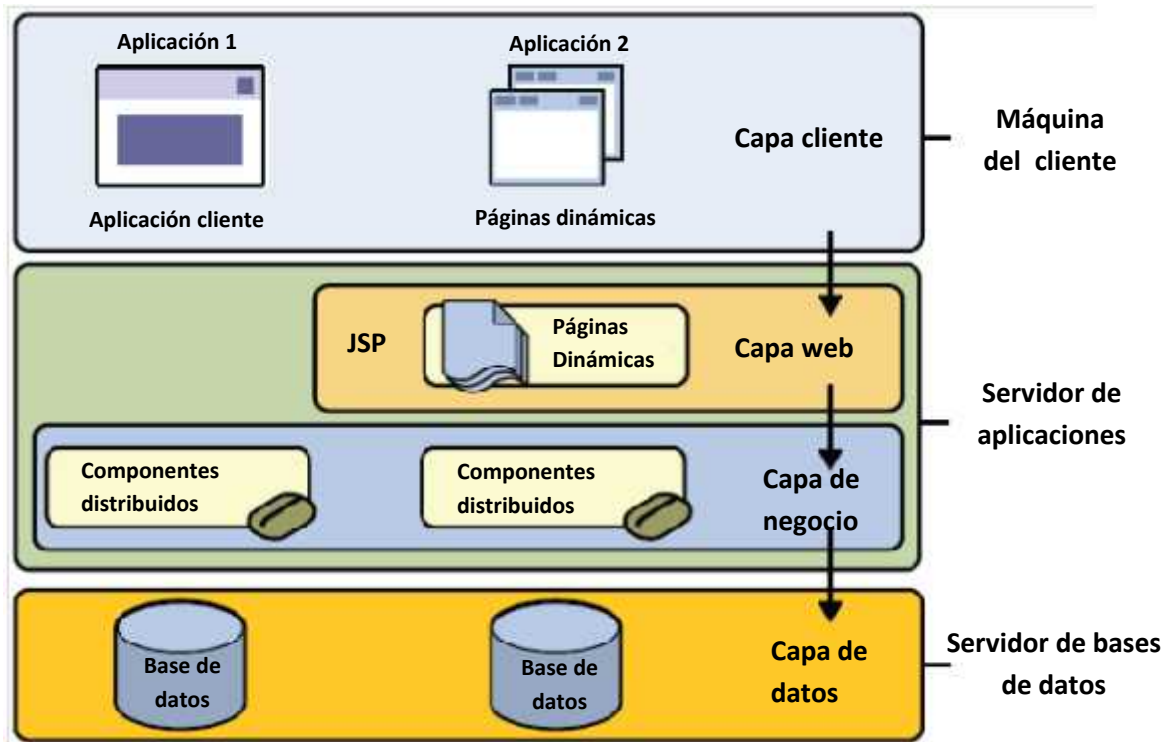


Figura 3.23 Plataforma Java EE¹⁵

Aunque una aplicación Java EE puede consistir en tres o cuatro niveles (el cuarto nivel consiste en el manejador de bases de datos), por lo general se dice que están implementadas en tres niveles ya que son distribuidas sobre tres diferentes locaciones: la máquina del cliente, el servidor Java EE, donde residen la capa Web y de Negocio y el servidor de bases de datos.

Las aplicaciones Java EE están hechas de varios componentes. Un componente Java EE es una unidad de software que es ensamblada en una aplicación Java EE con sus clases y archivos relacionados para comunicarse con otros componentes.

La especificación de la plataforma, define los siguientes componentes:

- Aplicaciones cliente y applets que corren del lado del cliente.
- Las tecnologías de Java Servlet, Java Server Faces y Java Server Pages (JSP) son componentes web que corren en el servidor.
- Los Enterprise JavaBeans (EJB) son componentes de negocio que también corren en el servidor.

En la figura 3.11 se muestran los diferentes componentes de una aplicación Java EE y cómo se comunican entre ellos.

¹⁵ Imagen tomada de The Java EE 5 tutorial For Sun Java System Application Server 9.1 By Sun Microsystems

Capítulo 3

Desarrollo

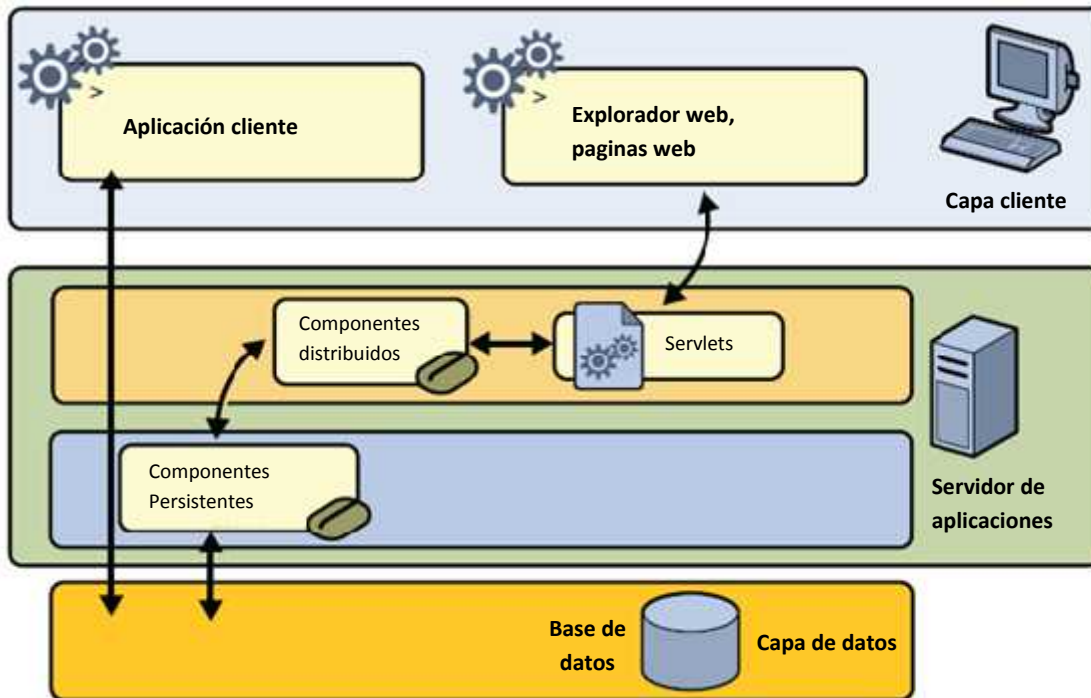


Figura 3.11 Componentes de una aplicación Java EE¹⁶

3.4.6.2 Javascript

Javascript es un lenguaje de programación interpretado. En este proyecto se utiliza en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web.

3.4.6.3 XML

Le Lenguaje de Marcas Extensible (XML) contiene, da forma, etiqueta, estructura y protege información. Esto lo hace con símbolos llamados marcas, embebidos en el texto. Las marcas mejoran el significado de la información que contienen en diferentes formas, identifican las cada dato y como se relacionan entre ellos.

XML es una parte importante de este proyecto ya que como se mostró en el capítulo 2, los archivos de configuración del contenedor Web (Apache Tomcat) se encuentran escritos en este lenguaje de marcas.

¹⁶ Imagen tomada de The Java EE 5 tutorial For Sun Java System Application Server 9.1 By Sun Microsystems

4

PRUEBAS

Debido a que es prácticamente imposible que el ser humano tenga una comunicación perfecta y que no se cometa ningún error en el ciclo de vida de software, debe existir una etapa que cumpla específicamente con la función de garantizar la calidad del producto, esta etapa siempre debe considerarse como crítica pues representa la revisión final de las especificaciones, del diseño y la codificación.

El costo de reparar un error aumenta conforme se avanza de etapa y si bien un error detectado en la etapa de pruebas puede resultar costoso siempre será más costoso repararlo cuando el sistema ya está en producción, es por esto que el presente capítulo tiene como propósito detectar defectos, comportamientos incorrectos o no deseados en el sistema, asimismo permitirá demostrar al cliente que el sistema satisface sus requerimientos.

Contenido:

- 4.1 Objetivos de las pruebas
- 4.2 Diseño de Casos de Prueba
- 4.3 Técnicas de Pruebas de caja blanca
- 4.4 Técnicas de Pruebas de caja negra
- 4.5 Casos de prueba

Capítulo 4

Pruebas

4.1 Objetivos de las pruebas

De acuerdo con Ian Sommerville en su libro: “Ingeniería del Software”, define dos principales objetivos de las pruebas:

1. *Demostrar al desarrollador y al cliente que el software satisface sus requerimientos.* Para el software a medida (como es el caso del SICICE), esto significa que debería haber al menos una prueba para cada requerimiento del sistema y del usuario. Algunos sistemas pueden tener una fase de pruebas de aceptación explícita en la que el cliente comprueba formalmente que el sistema cumple su especificación.
2. *Descubrir defectos en el software en el que el comportamiento de éste es incorrecto, no deseable o no cumple la especificación.* La prueba de defectos está relacionada con la eliminación de todos los tipos de sistema no deseables, tales como las caídas del sistema, interacciones no permitidas con otros sistemas, cálculos incorrectos y corrupción de datos.

Estos dos objetivos conducen a dos tipos básicos de pruebas: las pruebas de validación y las pruebas de defectos respectivamente. Para las pruebas de validación, una prueba de éxito es aquella en la que el sistema funciona correctamente. Para las pruebas de defectos, una prueba con éxito es aquella que muestra un defecto que hace que el sistema funciona incorrectamente.



Figura 4.1 Tipos básicos de pruebas.

Otro punto de vista presentado por Roger Pressman en su libro “Ingeniería de Software: Un Enfoque Práctico” sugiere tres objetivos de las pruebas:

1. La prueba es el proceso de ejecución de un programa con la intención de descubrir un error.
2. Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.

Capítulo 4

Pruebas

3. Una prueba tiene éxito si se descubre un error no detectado hasta entonces.

Los dos enfoques expuestos ayudan a eliminar la idea errónea de que una prueba sólo es exitosa si no se descubren errores, sin embargo el primer enfoque presenta una excepción: las pruebas de validación, las cuales se consideran exitosas si el sistema funciona correctamente, es decir, si cumple con las expectativas del cliente.

En conclusión las pruebas en general tienen por objetivo demostrar que el sistema es lo suficientemente bueno para poner en operación, es decir, son una herramienta para generar confianza en el software.

*Las pruebas sólo pueden demostrar la presencia de errores, no su ausencia.*¹⁷

4.2 Diseño de casos de prueba

En la figura 4.2 se muestra un ejemplo de modelo de proceso de pruebas tomado del libro “Ingeniería del Software” de Ian Somerville. Los bloques superiores representan las entradas y salidas de cada proceso de los bloques inferiores, la salida de uno es la entrada del siguiente. En el primer bloque, los casos de prueba son especificaciones de las entradas para la prueba y la salida esperada para el sistema más una afirmación de lo que se está probando. En el segundo bloque, los datos de prueba son las entradas que han sido identificadas para probar el sistema. En el tercer bloque, las salidas o resultados de las pruebas sólo pueden predecirse por personas que comprenden lo que debería hacer el sistema y el informe (cuarto bloque), debe realizarse tomando en cuenta el segundo y cuarto bloque.

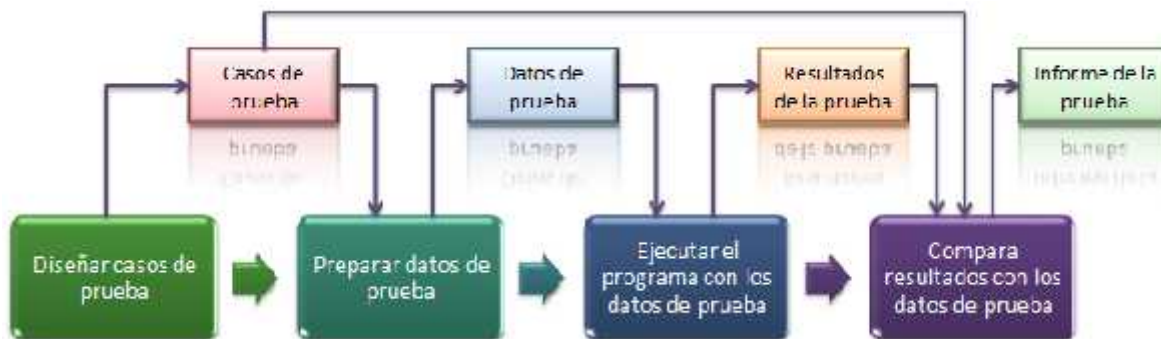


Figura 4.2 Diseño de casos de prueba.

Las pruebas exhaustivas, en las que cada posible secuencia de ejecución del programa es probada, son imposibles. Las pruebas por lo tanto, tienen que basarse en un subconjunto de posibles casos de prueba.

¹⁷ Edsger Dijkstra et al., 1972

Capítulo 4

Pruebas

El diseño de casos de prueba es una parte de las pruebas de componentes y sistemas en las que se diseñan los casos (entradas y salidas esperadas) para probar el sistema. El objetivo de este diseño es crear un conjunto de casos de prueba que sean efectivos descubriendo defectos en los programas y muestren que el sistema satisface sus requerimientos.

Para diseñar los casos de prueba hemos estudiado dos enfoques, el primero define tres aproximaciones:¹⁸

1. *Pruebas basadas en requerimientos*, en donde los casos de prueba se diseñan para probar los requerimientos del sistema. Para cada requerimiento, se identifica casos de prueba que puedan demostrar que el sistema satisface este requerimiento. Las pruebas basadas en requerimientos son pruebas de validación.
2. *Pruebas de particiones*, en donde se identifican particiones de entrada y salida y se diseñan pruebas para que el sistema ejecute entradas y genere salidas. Las particiones son grupos de datos que tienen características comunes, como todos los eventos provocados por la elección de operaciones en un menú y así sucesivamente.
3. *Pruebas estructurales*, en donde se utiliza el conocimiento de la estructura del programa para diseñar pruebas que ejecuten todas las partes del mismo. Esencialmente, cuando se prueba un programa, debería intentarse ejecutar cada sentencia al menos una vez. Las pruebas estructurales ayudan a identificar casos de prueba que pueden hacer esto posible. Esta aproximación se denomina a veces pruebas de <<caja blanca>>, de <<caja de cristal>> o de <<caja transparente>>.

El segundo enfoque define dos aproximaciones para el diseño de casos de uso:¹⁹

1. *Conociendo la función específica para la que fue diseñado el producto*, se pueden llevar a cabo pruebas que demuestren que cada función es completamente operativa y al mismo tiempo, buscando errores en cada función. A esta primera aproximación se le conoce como *prueba de caja negra* y se refiere a las pruebas que se llevan a cabo sobre la interfaz de software. Los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y se produce un resultado correcto, así como que la integridad de la información externa se mantiene. Este tipo de pruebas examinan algunos aspectos del modelo fundamental del sistema sin tener mucho en cuenta la estructura lógica interna del software.

¹⁸ Ingeniería del Software, Ian Sommerville, séptima edición, Pearson Addison Wesley.

¹⁹ Ingeniería de Software: Un Enfoque Práctico, Roger Pressman, quinta edición, Mc Graw Hill.

Capítulo 4

Pruebas

2. *Conociendo el funcionamiento del producto*, se pueden desarrollar pruebas que aseguren que todas las piezas encajan, o sea, que la operación interna se ajusta a las especificaciones y que todos los componentes internos se han comprobado de forma adecuada. A esta aproximación se le denomina *prueba de caja blanca* y se basa en el minucioso examen de los detalles procedimentales. Se prueban los caminos lógicos del software proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones y/o bucles. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coincide con el esperado o mencionado.

Analizando detenidamente ambos enfoques se observa que en los dos, aunque presentados de manera distinta, toman en cuenta las pruebas de tipo funcional y estructural. En general, cuando se diseñan casos de prueba, deberían comenzar con las pruebas de más alto nivel a partir de los requerimientos y a continuación, de forma progresiva, añadir pruebas más detalladas utilizando pruebas estructurales y pruebas de particiones.

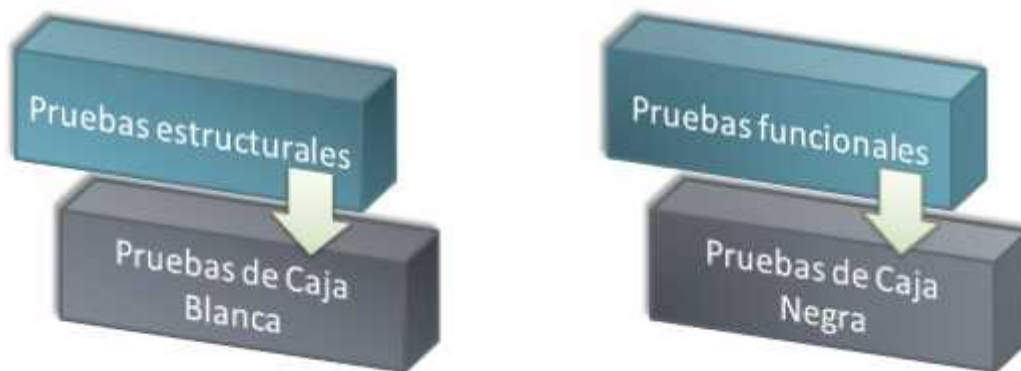


Figura 4.3 Pruebas de caja blanca y negra.

Para el SICICE se decidió diseñar casos de prueba que permiten evaluar los requerimientos funcionales del sistema y su estructura, ya que estos dos elementos permiten tener una visión amplia de la calidad del software y si es factible ponerlo en producción. Para evaluar estos elementos se toman en cuenta las técnicas de la caja negra y de caja blanca respectivamente.

En las siguientes secciones se examinará detalladamente los objetivos, técnicas y criterios para cada tipo de prueba que se aplicaron al SICICE y finalmente se presentan los formatos de casos utilizados.

Capítulo 4

Pruebas

4.2.1 Pruebas funcionales

Las pruebas de funcionalidad se deberían centrar en cualquier requisito que pueda ser trazado directamente de los casos de uso y reglas de negocio. El objetivo de estas pruebas es verificar la aceptación, procesamiento y recuperación de datos y la adecuada implementación de las reglas. Este tipo de pruebas están basadas en técnicas de caja negra, es decir, verificar la aplicación interactuando a través de las interfaces de usuario y analizando los resultados.

Objetivos de las Pruebas	Asegurar la navegación correcta de la aplicación, la entrada de datos, su procesamiento y recuperación.
Técnicas	Ejecutar cada flujo del caso de uso con datos válidos e inválidos para verificar los siguiente: <ul style="list-style-type: none">• Cuando se utilizan datos correctos se obtienen los resultados esperados.• Cuando los mensajes de error o advertencias son adecuadas.• Cada regla de negocio se ha aplicado correctamente.
Criterios de finalización	Todas las pruebas planificadas se han ejecutado. Todos los defectos identificados se ha considerando.

Tabla 4.1 Pruebas funcionales.

4.2.2 Pruebas de estructurales

Las pruebas de estructura deberán diseñarse por una persona que tenga conocimiento y comprensión de los algoritmos utilizados para implementar un componente. Debido a la complejidad de realizar una prueba exhaustiva ya que aún en proyectos pequeños, la cantidad de caminos lógicos se puede incrementar demasiado, las pruebas se centran en funciones clave del sistema.

Objetivos de las Pruebas	Ejercitar las decisiones lógicas en sus vertientes: verdadera y falsa. Ejecutar todos los bucles con sus límites operacionales. Ejercitar las estructuras internas de datos para asegurar su validez.
Técnicas	<ul style="list-style-type: none">• Pruebas de camino básico.• Pruebas de estructura de control.
Criterios de finalización	Todos los caminos lógicos planificados se han probado. Todos los defectos encontrados se han considerado.

Tabla 4.2 Pruebas estructurales.

4.2.3 Formato de casos de prueba

El siguiente formato (Figura 4.4) tiene por objetivo estandarizar y llevar el control de las pruebas para poder realizar un mejor análisis de los resultados.

Capítulo 4

Pruebas

SISTEMA DE CONTROL DE INVENTARIOS CASO DE PRUEBA			
Número	Fecha	Tipo de Prueba	Técnica empleada
Descripción:			
Condición de ejecución:			
<ul style="list-style-type: none">• Precondiciones• Valores de entrada• Observaciones• Resultados esperados• Poscondiciones			

Figura 4.4. Formato de caso de prueba

Los elementos que componen el formato se describen a continuación:

- *Número*. Es importante llevar el control del número de pruebas aplicadas.
- *Fecha*. Es la fecha en que se realizó la prueba.
- *Tipo de prueba*. Este campo se refiere a si es una prueba de tipo funcional o estructural.
- *Técnica empleada*. Campo que se utiliza para indicar la técnica específica que se utilizó en la prueba. Este campo junto con el de técnica empleada tienen por objetivo facilitar la identificación de la naturaleza de la prueba y al mismo tiempo proporcionar al personal de soporte una manera práctica de identificar la documentación que deben buscar si fuera necesario profundizar en la prueba.
- *Descripción del caso de prueba*. Es un párrafo en el que se narra brevemente en qué consiste la prueba, sus objetivos específicos y alcances.
- *Condición de ejecución*. Define el contexto en el que se desarrolla la prueba, se especifican los siguientes puntos para cada caso:
 - ✓ Precondiciones
 - ✓ Valores de entrada
 - ✓ Observaciones
 - ✓ Resultados esperados
 - ✓ Poscondiciones

Capítulo 4

Pruebas

4.3 Técnicas de pruebas de caja blanca

Para aplicar las pruebas estructurales es necesario primero determinar en qué consisten los métodos de este tipo de validaciones para poder elegir los que permiten evaluar de manera más óptima los elementos del sistema. En esta sección se describen y estudian algunas de las técnicas más comunes para pruebas de caja blanca.

En general las pruebas de caja blanca:

1. Garantizan que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo.
2. Ejercitan todas las decisiones lógicas en sus vertientes verdadera y falsa.
3. Ejecutan todos los bucles con sus límites operacionales.
4. Ejercitan las estructuras internas de datos para asegurar su validez.

Todos los métodos que se presentan a continuación son técnicas de pruebas de estructura de control la primera, la prueba del camino básico es sencilla y muy efectiva, sin embargo, no es suficiente por sí misma por lo que en esta sección se estudian otras variantes de la prueba de estructura de control que pueden ampliar y mejorar la calidad de las pruebas de caja blanca.

4.3.1 Prueba del camino básico

El método del camino básico permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. Los casos de prueba obtenidos del conjunto básico garantizan que durante la validación se ejecute por lo menos una vez cada sentencia del programa.

Este método se usa en secciones del código donde encontramos sentencias lógicas como if, while, case, etc., el objetivo es encontrar valores que permitan ejecutar los posibles caminos que se pueden tomar en esos segmentos de código. Para determinar que caminos debemos probar se usan como apoyo dos herramientas: la notación de grafo de flujo y la complejidad ciclomática.

- Notación de grafo de flujo

Para tener claro el concepto del camino básico primero es necesario representar el segmento de código de interés a través de un diagrama de flujo, este primer acercamiento proporciona una idea de los posibles caminos que se pueden ejecutar tomando en cuenta las decisiones lógicas que en él se encuentren. Para simplificar esta representación se utiliza la notación de grafo de flujo en el que cada círculo, denominado nodo del grafo, representa una o más sentencias. Un sólo nodo puede corresponder a una secuencia de cuadros de proceso y a un rombo de decisión. Las flechas del grafo denominadas aristas o enlaces, representan el flujo de control y son análogas a las flechas del diagrama de flujo. Una arista debe terminar en un nodo, incluso aunque éste no

Capítulo 4

Pruebas

represente ninguna sentencia. Las áreas delimitadas por aristas y nodos se denominan regiones. Cada nodo que contiene una condición se denomina nodo predicado y está caracterizado porque dos o más aristas emergen de él.

Más adelante cuando mostremos los resultados de las pruebas aplicadas al SICICE se observará con más detalle cómo se emplea esta notación.

- Complejidad ciclomática

La complejidad ciclomática es una métrica que define el número de caminos independientes del conjunto básico de un programa y establece un límite superior para el número de pruebas que se deben realizar para asegurar que se ejecuta cada sentencia al menos una vez.

Un camino independiente es cualquier camino del programa que introduce, por lo menos, un nuevo conjunto de sentencias de proceso o una nueva condición. En términos del grafo de flujo, un camino independiente está constituido por lo menos por una arista que no haya sido recorrida anteriormente a la definición del camino.

4.3.2 Prueba de condición

Este método de diseño de casos de prueba ejercita las condiciones lógicas contenidas en un módulo de un programa. Una condición simple es una variable lógica o una expresión relacional. Una expresión relacional es de la forma:

$$E_1 \text{ <operador relacional> } E_2$$

Donde E_1 y E_2 , son expresiones aritméticas y <operador relacional> puede ser alguno de los siguientes: <, <=, =, ≠, >, o >=. Una condición simple puede estar precedida también por un operador NOT. Una condición compuesta está formada por dos o más condiciones simples, operadores lógicos y paréntesis.

Si una condición es incorrecta, entonces es erróneo al menos un componente de la condición. Así los tipos de errores de una condición pueden ser los siguientes:

- error en operador lógico (existencia de operadores lógicos incorrectos /desaparecidos / sobrantes).
- error en variable lógica.
- error en paréntesis lógico.
- error en operador relacional.
- error en expresión aritmética.

Capítulo 4

Pruebas

La prueba de condiciones tiene las siguientes dos ventajas:

1. La cobertura de la prueba es sencilla.
2. Propicia la generación de pruebas adicionales al programa.

4.3.3 Prueba de bucles

Un elemento de suma importancia en la mayoría de los algoritmos que se implementan en los sistemas son los ciclos o bucles, es por esto que se vuelve imprescindible comprobar la validez de la construcción de dichos ciclos.

En la prueba de bucles se definen cuatro tipos para facilitar las pruebas de validez. Para cada tipo se especifica un enfoque diferente para realizar las pruebas. En la Figura 4.5 se muestra cada clase:

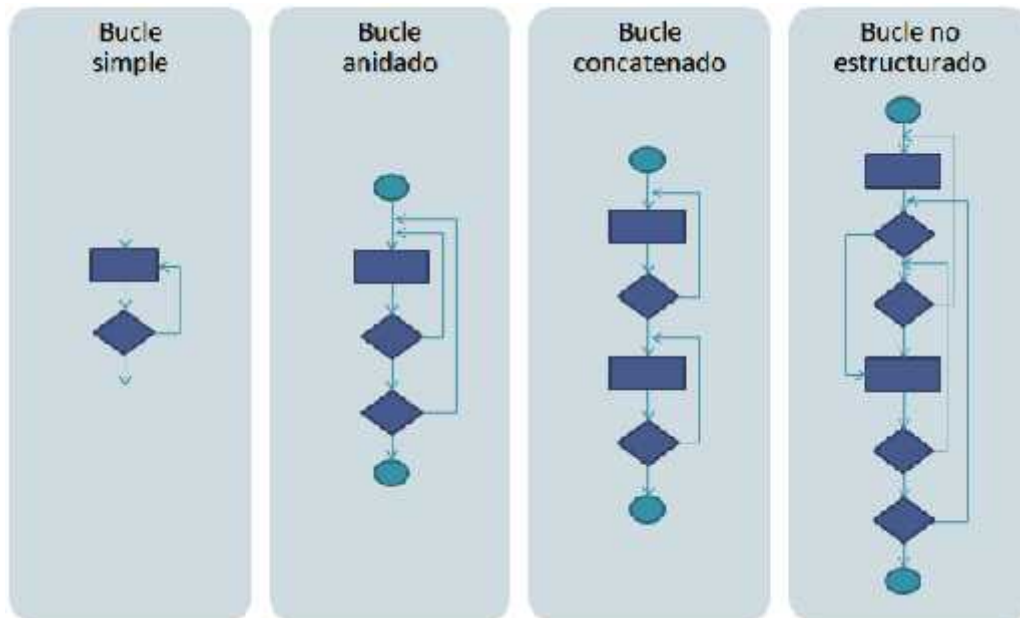


Figura 4.5. Tipos de bucles

- Bucle simple

A un bucle simple se le debe aplicar el siguiente conjunto de pruebas donde n es el número máximo de pasos permitidos por el bucle:

1. Pasar por alto totalmente el bucle.
2. Pasar una sola vez por el bucle.

Capítulo 4

Pruebas

3. Pasar dos veces por el bucle.
4. Hacer m pasos por el bucle con $m < n$.
5. Hacer $n-1$, y $n+1$ pasos por el bucle.

- Bucle anidado

Para reducir el número de pruebas que se puede incrementar de manera significativa con los bucles anidados existe el siguiente enfoque:

1. Comenzar por el bucle más interno. Establecer o configurar los demás bucles con sus valores mínimos.
2. Llevar a cabo las pruebas de bucles simples para el bucle más interno, mientras se mantienen los parámetros de iteración de los bucles externos en sus valores mínimos. Añadir otras pruebas para valores fuera de rango o excluidos.
3. Progresar hacia fuera, llevando a cabo pruebas para el siguiente bucle, pero manteniendo todos los bucles externos en sus valores mínimos y los demás bucles anidados en sus valores típicos.
4. Continuar hasta que se hayan probado todos los bucles.

- Bucle concatenado

Cuando los bucles concatenados son independientes se puede aplicar la misma prueba de los simples, sin embargo si en dos bucles concatenados se usa el controlador del bucle 1 como el valor inicial del bucle 2, entonces éstos no son independientes y por lo tanto se debe usar el enfoque de los anidados.

- Bucle no estructurado

Siempre que sea posible, esta clase de bucles se deben rediseñar para que se ajusten a las construcciones anteriores.

4.4 Técnicas de pruebas de caja negra

Para aplicar el enfoque de pruebas funcional (cabe aclarar que este es un enfoque complementario y no opcional al estructural) es necesario conocer las técnicas de caja negra que permiten encontrar los siguientes tipos de errores:

1. Funciones incorrectas o ausentes.
2. Errores de interfaz.

Capítulo 4

Pruebas

3. Errores en estructuras de datos o accesos a bases de datos.
4. Errores de rendimiento.
5. Errores de inicialización y terminación.

Existen diversas técnicas de pruebas de caja negra que permiten valorar distintos elementos del sistema y que se complementan unas a otras, sin embargo en las siguientes líneas sólo se explicarán brevemente las que se consideraron más importantes y más adecuadas para aplicar al SICICE.

4.4.1 Partición equivalente

La característica principal de este tipo de prueba es que se divide el campo de entrada de un programa en clases de datos, de esta manera se pueden descubrir clases de errores, reduciendo así el número total de casos de prueba a desarrollar.

El diseño de casos de prueba para la partición equivalente se basa en una evaluación de las clases de equivalencia para una condición de entrada. Una clase de equivalencia representa un conjunto de estados válidos y no válidos para condiciones de entrada. Por lo general una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica.

Roger Pressman en su libro “Ingeniería del Software” describe algunos criterios para determinar las clases de equivalencia:

1. Si una condición de entrada especifica un rango, se define una clase de equivalencia válida y dos no válidas.
2. Si una condición de entrada requiere un valor específico, se define una clase de equivalencia válida y dos no válidas.
3. Si una condición de entrada especifica un miembro de un conjunto, se define una clase de equivalencia válida y una no válida.
4. Si una condición de entrada es lógica, se define una clase de equivalencia válida y una no válida.

4.5 Casos de prueba

A través de este capítulo hemos estudiado los enfoques y técnicas que permiten diseñar exitosos casos de prueba, en esta sección finalmente se presentan los casos de prueba diseñados específicamente para el SICICE que permitirán evaluar diferentes aspectos del sistema. Como fue explicado anteriormente es prácticamente imposible realizar pruebas exhaustivas que permitan evaluar absolutamente todas las posibles entradas del sistema por lo que los siguientes casos de prueba se enfocan en los aspectos que se consideraron más críticos.

Capítulo 4

Pruebas

Se presentan casos de pruebas estructurales que se deben aplicar a los módulos de altas, bajas y modificaciones ya que es donde se eliminan o dan de alta registros en la base de datos; también se aplican en el módulo de reportes donde es crítico el manejo de bucles para reducir el tiempo de respuesta. Los casos de pruebas funcionales sólo se aplican en los módulos de altas y modificaciones donde es crítico validar los datos que introduce el usuario para ser dados de alta en la base de datos.

SISTEMA DE CONTROL DE INVENTARIOS CASO DE PRUEBA

Número	Fecha	Tipo de prueba	Técnica empleada
1	20-02-2011	Prueba estructural	Prueba de camino básico

Descripción:	El siguiente caso de prueba se aplica al módulo de altas, específicamente se evalúa el archivo de <i>insertaTrans.jsp</i> en el que se insertan en la base de datos los bienes que definió el usuario. Se presenta su notación de grafo de flujo para aplicar la prueba de camino básico. El objetivo es probar que con todos los caminos se conserva la consistencia de información en la base de datos.
---------------------	---

Condición de ejecución:	Se evalúa la porción de código del archivo <i>insertaTrans.jsp</i> donde se recorre cada elemento de un arreglo llamado <i>equipos</i> en el que se almacenan los tipos bienes que el usuario desea dar de alta, si el arreglo contiene determinado tipo de bien, se prepara la query para dar de alta su información en la tabla correspondiente en la base de datos.
--------------------------------	--

- **Precondiciones**

1. La principal precondición es tener el diagrama de flujo de la operación. En la Figura 4.6 se muestra dicho diagrama.
2. La segunda precondición es obtener la notación de grafo de flujo, mostrada en la figura 4.7.
3. Finalmente se obtiene la complejidad ciclomática del grafo obtenido.

La complejidad ciclomática, $V(G)$ de un grafo de flujo G está definida como:

$$V(G)=A-N+2 \quad \text{ó} \quad V(G)=P+1$$

Donde A es el número de aristas del grafo de flujo, N es el número de nodos y P el número de nodos predicados. Para este caso:

Capítulo 4

Pruebas

A=18
N=15
P=4

Por lo tanto, para este caso de prueba.

$$V(G)=18-15+2=5 \quad \text{ó} \quad V(G)=4+1=5$$

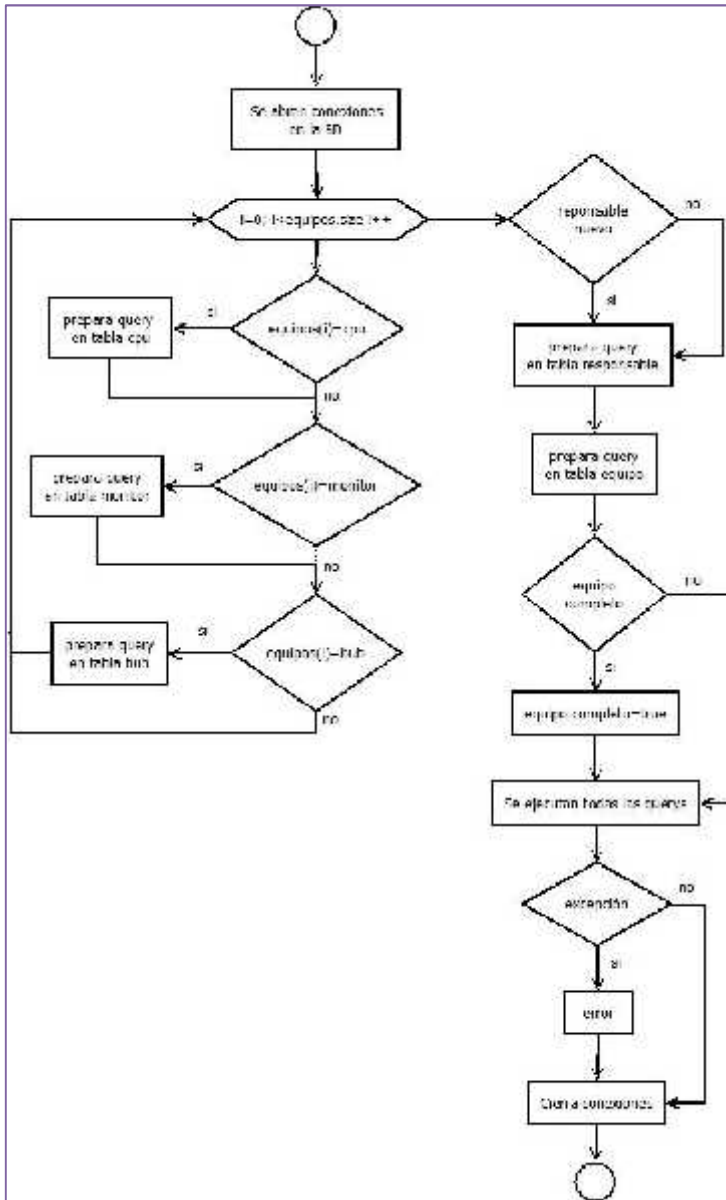


Figura 4.6 Diagrama de flujo

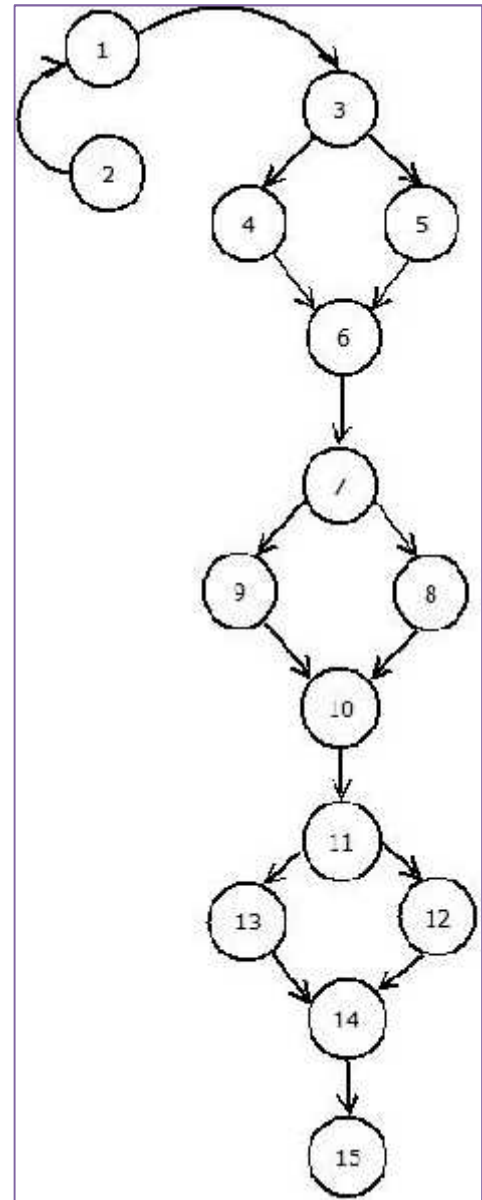


Figura 4.7 Notación de grafo

Capítulo 4

Pruebas

Los caminos independientes propuestos son los siguientes:

Camino 1: 1-2-3-4-6-7-8-10-11-12-14-15

Camino 2: 1-2-3-5-6-7-8-10-11-12-14-15

Camino 3: 1-2-3-4-6-7-9-10-11-12-14-15

Camino 4: 1-2-3-5-6-7-9-10-11-13-14-15

Camino 5: 1-3-5-6-7-9-10-11-12-14-15

- **Valores de entrada**

Los siguientes valores se deben probar para todos los caminos:

Tamaño o número de elementos del arreglo equipos:

- > Equipos.size=0
- > Equipos.size>0

Valor de la variable que determina si se va a dar de alta un nuevo responsable para los bienes que serán dados de alta o se tomará un responsable de los ya existentes:

- > Responsable= nuevo
- > Responsable= no nuevo

Valor que determina si el equipo está completo o sólo se está dando de alta un bien de manera independiente:

- > Equipo completo = si
- > Equipo completo = no

Valor que determina si ha ocurrido un error o excepción durante la actualización de información en la base de datos:

- > Excepción = true
- > Excepción = false

- **Observaciones**

Se simplifica el ciclo *for* debido a que es repetitivo y para poder enfocarse en los caminos que impliquen interacción con la base de datos.

Capítulo 4

Pruebas

- **Resultados esperados**

El valor de *equipos.size* en teoría nunca debe ser cero ya que existe una validación en otra parte del código que evita que se pueda llegar a este punto cuando no se han elegido ningún equipo para dar de alta, es decir *equipos.size*=0. El valor máximo de *equipos.size* debería ser 19 ya que es el número de tipos de bienes que existe (Ver tabla 3.2 en Capítulo 3).

Si el responsable es nuevo, debe preparar la query para dar de alta la información en la base de datos.

Si el equipo es completo se debe poner la bandera que indica que el equipo es nuevo como *true*.

Si ocurre una excepción el sistema debe avisar al usuario que ha ocurrido un problema al dar de alta la información en la base de datos y debe hacer un *rollback* para evitar inconsistencias.

- **Poscondiciones**

Verificar la consistencia de los datos cuando se haga *rollback* (en caso de presentarse un error) o cuando se hagan las inserciones en las diversas tablas en la base de datos.

Capítulo 4

Pruebas

SISTEMA DE CONTROL DE INVENTARIOS CASO DE PRUEBA

Número	Fecha	Tipo de prueba	Técnica empleada
2	20-02-2011	Prueba estructural	Prueba de camino básico

Descripción:	Este caso se aplica al módulo de bajas, en el cual un usuario puede dar de baja bienes del sistema. Se presenta una notación de grafo para aplicar la prueba de camino básico. El objetivo es comprobar que con todos los caminos se conserva la consistencia de información en la base de datos.
---------------------	---

Condición de ejecución:	Se evalúa la porción de código del archivo <i>bajas5.jsp</i> donde se elimina de la tabla <i>equipo</i> y se actualiza el campo de <i>estado</i> (que indica si un bien esta dado de baja o de alta) en los registros de las tablas de bienes cuando un usuario elige dar de baja del sistema uno o más bienes.
--------------------------------	---

- **Precondiciones**

1. La primera condición es realizar el diagrama de flujo para el segmento de código de interés, el cual se muestra en la Figura 4.8.
2. A partir de dicho diagrama de flujo se obtiene la notación de grafo representado en la Figura 4.9.
3. Finalmente, en base a la notación de grafo se calcula la complejidad ciclomática.

La complejidad ciclomática, $V(G)$ de un grafo de flujo G está definida como:

$$V(G)=A-N+2 \quad \text{ó} \quad V(G)=P+1$$

Donde A es el número de aristas del grafo de flujo, N es el número de nodos y P el número de nodos predicados. Para este caso:

$$\begin{aligned} A &= 26 \\ N &= 19 \\ P &= 8 \end{aligned}$$

Por lo tanto, para este caso de prueba.

$$V(G)=26-19+2=9 \quad \text{ó} \quad V(G)=8+1=9$$

Capítulo 4

Pruebas

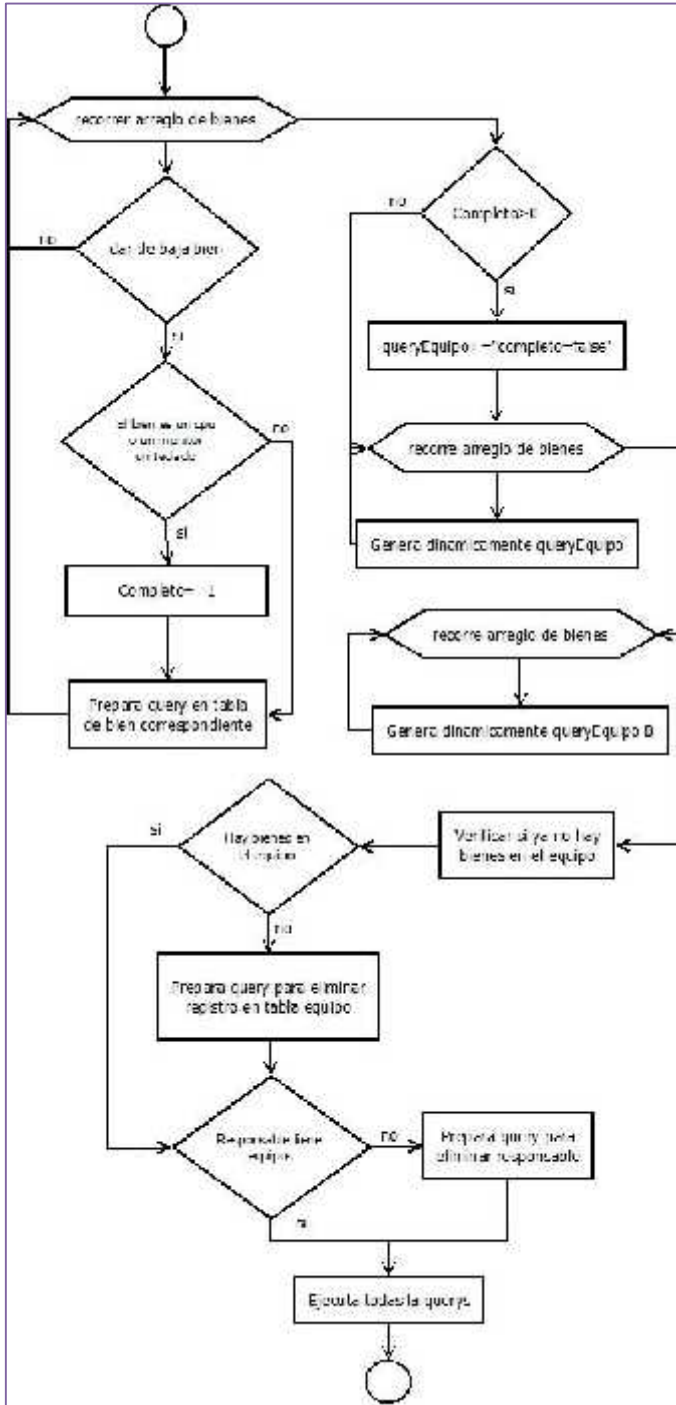


Figura 4.8 Diagrama de flujo

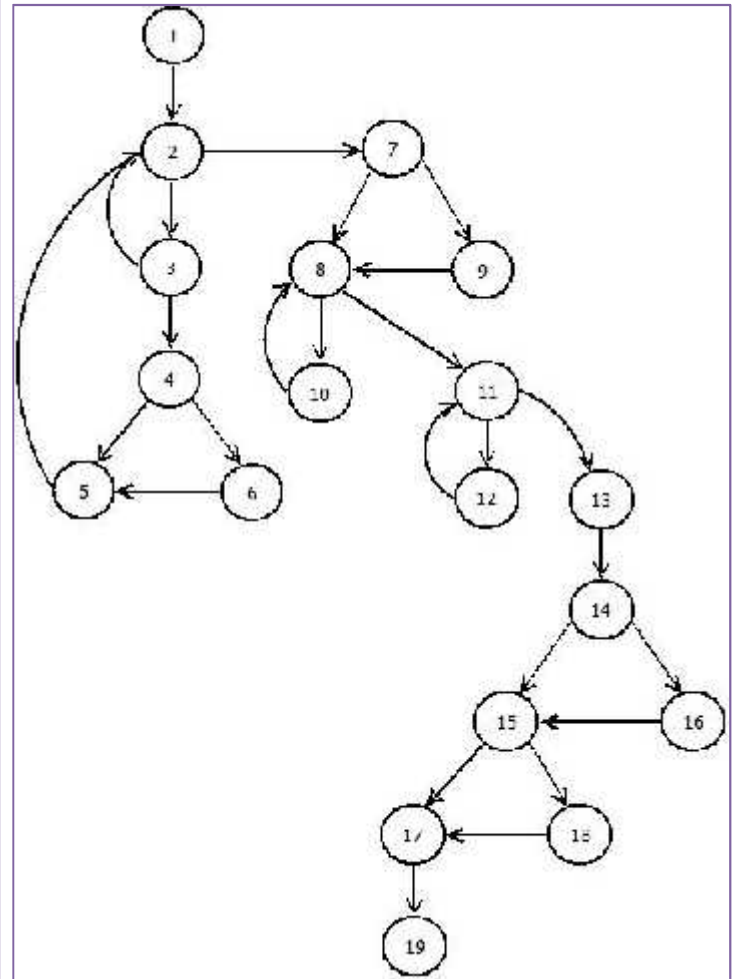


Figura 4.9 Notación de grafo

Capítulo 4

Pruebas

Los caminos independientes propuestos son los siguientes:

Camino 1: 1-2-3-4-5-2-7-8-11-13-14-15-17-19

Camino 2: 1-2-7-8-11-13-14-15-17-19

Camino 3: 1-2-3-2-7-9-8-11-13-14-16-15-18-17-19

Camino 4: 1-2-3-4-6-5-2-7-8-10-8-11-12-11-13-14-15-17-19

Camino 5: 1-2-3-4-5-2-7-9-8-11-12-11-13-14-16-15-17-19

Camino 6: 1-2-7-8-10-8-11-12-11-13-14-16-15-18-17-19

Camino 7: 1-2-7-9-8-11-12-11-13-14-15-17-19

Camino 8: 1-2-7-9-8-11-13-14-15-17-19

Camino 9: 1-2-3-2-7-8-10-8-11-12-11-13-14-16-15-18-17-19

- **Valores de entrada**

Los siguientes valores se deben probar para todos los caminos:

Cuando se analiza determinado tipo de bien:

- > Dar de baja bien=si
- > Dar de baja bien=no

Cuando se determina si un equipo debe ser marcado como completo:

- > El bien es CPU, monitor o teclado=si
- > El bien es CPU, monitor o teclado=no

Variable que guarda el valor que indica si un equipo está completo:

- > completo=0
- > completo>0

Cuando se verifica si un equipo tiene al menos un bien asignado:

- > Hay bienes en el equipo=si
- > Hay bienes en el equipo=no

Cuando si un responsable tiene asignado al menos un equipo:

- > Responsable tiene equipos=si
- > Responsable tiene equipo=no

Capítulo 4

Pruebas

- **Observaciones**

Debido a la complejidad del algoritmo el diagrama de flujo de la figura 4.8 simplifica muchos de los procesos implicados con determinadas entradas, por ejemplo, para determinar el valor de entrada que evalúa si un responsable tiene asignados equipos es necesario realizar una query a una base de datos. Estos detalles se tendrán que tomar en cuenta para determinar los valores de entrada.

- **Resultados esperados**

Cuando se evalúa el arreglo de tipos de bienes, por lo menos un equipo debe resultar dado de baja, esto significará que la validación que impide que un usuario pueda continuar el proceso de baja, si no es seleccionado ningún bien ha funcionado correctamente.

Cuando un bien sea de tipo *CPU, monitor o teclado*, el valor de la variable completo deberá aumentarse en una unidad.

Cuando la variable *Completo* sea mayor a 0 deberá actualizarse en la base de datos el registro del equipo en cuestión en el campo de completo por: *FALSE*.

Cuando un registro tiene como valor 0 en todos los id's de bienes, entonces querrá decir que ese equipo no tiene asociado ningún bien y por lo tanto debe ser dado de baja.

Si después de dar de baja uno o más bienes y al consultar la tabla de responsables resulta que algún registro ya no tiene asociado ningún equipo, entonces el registro de dicho responsable también debe ser dado de baja.

- **Poscondiciones**

Verificar la consistencia de los datos cuando se haga *rollback* (en caso de presentarse un error) o cuando actualicen o eliminen registros en las diversas tablas en la base de datos.

Capítulo 4

Pruebas

SISTEMA DE CONTROL DE INVENTARIOS CASO DE PRUEBA

Número	Fecha	Tipo de prueba	Técnica empleada
3	20-02-2011	Prueba estructural	Prueba de camino básico

Descripción:	Esta prueba se aplica al módulo de modificaciones, en el archivo de <i>finalr.jsp</i> en el que se actualizan los datos de los bienes especificados por el usuario. El objetivo es comprobar que se conserva la consistencia de datos en todos los caminos independientes.
---------------------	--

Condición de ejecución:	Se evalúa el segmento de código del archivo <i>finalr.jsp</i> donde se actualizan los datos de un registro específico en una de las diferentes tablas de bienes cuando un usuario elige modificar los datos de un bien determinado.
--------------------------------	---

- **Precondiciones**

1. La primera condición es realizar el diagrama de flujo para el segmento de código de interés, el cual se muestra en la Figura 4.10
2. A partir de dicho diagrama de flujo se obtiene la notación de grafo representado en la Figura 4.11
3. Finalmente, en base a la notación de grafo se calcula la complejidad ciclomática.

La complejidad ciclomática, $V(G)$ de un grafo de flujo G está definida como:

$$V(G)=A-N+2 \quad \text{ó} \quad V(G)=P+1$$

Donde A es el número de aristas del grafo de flujo, N es el número de nodos y P el número de nodos predicados. Para este caso:

$$A=13$$

$$N=11$$

$$P=3$$

Por lo tanto, para este caso de prueba.

$$V(G)=13-11+2=4 \quad \text{ó} \quad V(G)=3+1=4$$

Capítulo 4

Pruebas

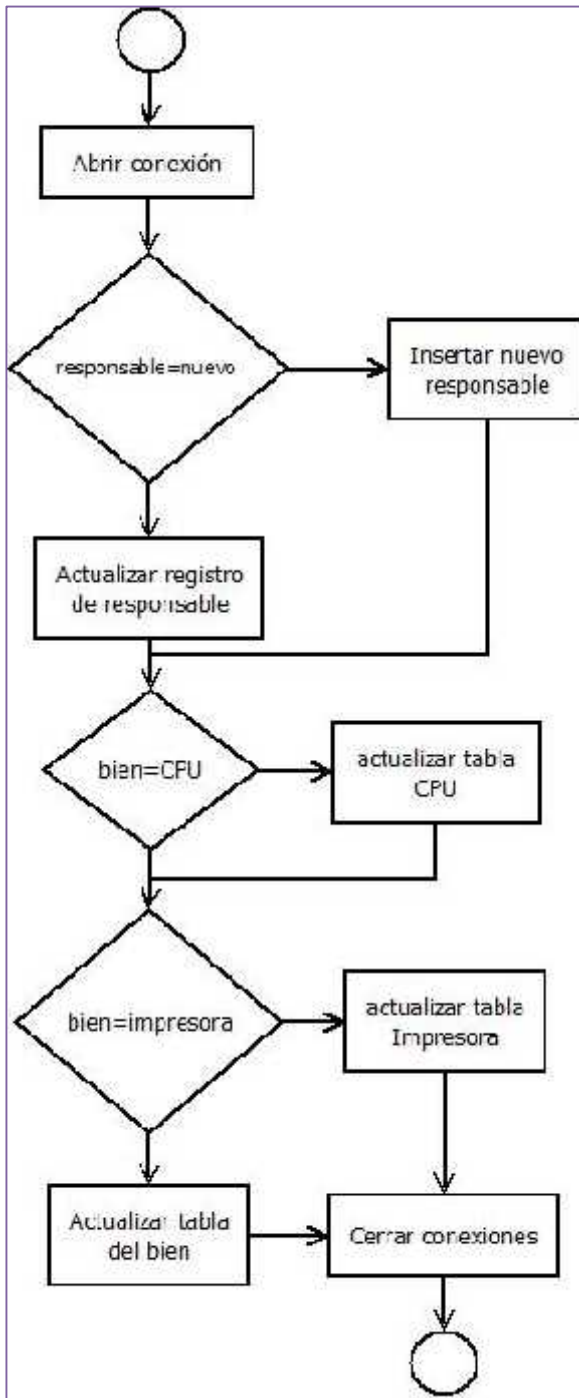


Figura 4.10 Diagrama de flujo

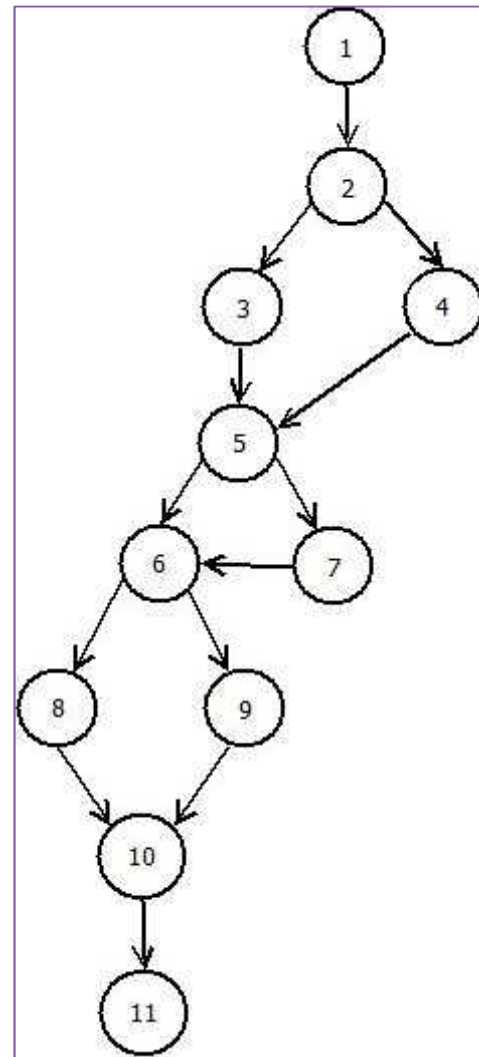


Figura 4.11 Notación de grafo

Capítulo 4

Pruebas

Los caminos independientes propuestos son los siguientes:

Camino 1: 1-2-3-5-6-8-10-11

Camino 2: 1-2-4-5-7-6-8-10-11

Camino 3: 1-2-3-5-7-6-9-10-11

Camino 4: 1-2-4-5-6-9-10-11

- **Valores de entrada**

Los siguientes valores se deben probar para todos los caminos:

El valor de la variable de sesión que indica si el responsable de un bien a modificar será nuevo:

- > Responsable nuevo=si
- > Responsable nuevo=no

Variable de sesión que contiene el tipo de bien, determinar si es de tipo CPU:

- > El bien es CPU=si
- > El bien es CPU=no

Variable de sesión que contiene el tipo de bien, determinar si el de tipo Impresora:

- > El bien es Impresora=si
- > El bien es Impresora=no

- **Resultados esperados**

Las actualizaciones deben realizarse en tablas diferentes dependiendo del tipo de bien, así mismo se deberá crear un nuevo registro en la tabla de responsables cuando el usuario ha decidido asignar un nuevo responsable.

- **Poscondiciones**

Verificar que las actualizaciones se hayan hecho en las tablas correctas y de manera correcta.

Capítulo 4

Pruebas

SISTEMA DE CONTROL DE INVENTARIOS CASO DE PRUEBA

Número	Fecha	Tipo de prueba	Técnica empleada
4	20-02-2011	Prueba estructural	Prueba de bucles

Descripción:	Este caso de prueba está diseñado para el módulo de reportes. El objetivo es evaluar cada uno de los bucles presentes en el archivo <i>DemoJExcel.java</i> , dicho archivo es uno de los más importantes para el módulo de reportes ya que contiene funciones primordiales para la generación de los mismos. Los bucles a evaluar son los siguientes: I. Bucle <i>while</i> para leer de la base de datos los bienes de cada división II. Bucle <i>for</i> para escribir un reporte personalizado de la división UNICA
---------------------	--

Condición de ejecución:	Se evalúan para cada bucle las precondiciones, valores de entrada, resultados esperados y poscondiciones de manera independiente.
--------------------------------	---

I. BUCLE WHILE PARA LEER DE LA BASE DE DATOS LOS BIENES DE CADA DIVISIÓN

- **Precondiciones**

1. Identificar el tipo de bucle de acuerdo a la sección 4.4.2 del capítulo 4.

En este caso es en bucle de tipo simple.

2. Establecer el número máximo "n" de pasos permitidos por el bucle.

En este caso "n" está definida por el número de registros en la tabla división.

- **Valores de entrada**

1. Pasar por alto totalmente el bucle.
2. Pasar una sola vez por el bucle.
3. Pasar dos veces por el bucle.
4. Hacer m pasos por el bucle con $m < n$.
5. Hacer $n-1$ y $n+1$ pasos por el bucle.

Capítulo 4

Pruebas

- **Resultados esperados**

Pasar por alto el bucle debería arrojar como resultado un reporte vacío.

Pasar una sola vez por el bucle arrojaría sólo los bienes de una división.

Pasar dos veces por el bucle imprimiría los bienes de las dos primeras divisiones.

Pasar m veces imprimiría los datos de las m divisiones.

Pasar n-1 veces imprimiría los bienes de todas las divisiones excepto la última.

Hacer n+1 veces imprimiría los bienes de todas las divisiones y a la n+1 vez ya no entra al ciclo.

- **Poscondiciones**

Existe un reporte que contiene los bienes de cada una de las divisiones registradas en la base de datos.

II. BUCLE FOR PARA ESCRIBIR REPORTE PERSONALIZADO PARA LA DIVISIÓN “UNICA”

- **Precondiciones**

1. *Identificar el tipo de bucle de acuerdo a la sección 4.4.2 del capítulo 4.*

En este caso tenemos un bucle anidado.

2. *Establecer el número de bucles interiores.*

En este caso se tienen dos bucles interiores anidados.

3. *Establecer el número máximo de pasos permitidos para cada bucle.*

Para el primer bucle (bucle más exterior) n_1 =número de departamentos de UNICA.

Para el segundo bucle (segundo bucle interior) n_2 =número de equipos de cada departamento.

Para el tercer bucle (bucle más interior) n_3 =número de columnas para cada equipo.

Capítulo 4

Pruebas

- **Valores de entrada**

1. Con el bucle más exterior en su valor mínimo y el segundo bucle interior también en su valor mínimo, y probar para el bucle más interior los siguientes valores:
 - a. Pasar por alto totalmente el bucle.
 - b. Pasar una sola vez por el bucle.
 - c. Pasar dos veces por el bucle.
 - d. Hacer m_3 pasos por el bucle con $m_3 < n_3$.
 - e. Hacer $n_3 - 1$ y $n_3 + 1$ pasos por el bucle.
2. Con el bucle más exterior en su valor mínimo y el bucle más interior con valor mínimo de 4 hacer las siguientes pruebas para el segundo bucle interior:
 - a. Pasar por alto totalmente el bucle.
 - b. Pasar una sola vez por el bucle.
 - c. Pasar dos veces por el bucle.
 - d. Hacer m_2 pasos por el bucle con $m_2 < n_2$.
 - e. Hacer $n_2 - 1$ y $n_2 + 1$ pasos por el bucle.
3. Con el bucle más interior con valor de 4 y el segundo bucle interior también con su valor mínimo, hacer las siguientes pruebas para el bucle más exterior:
 - a. Pasar por alto totalmente el bucle.
 - b. Pasar una sola vez por el bucle.
 - c. Pasar dos veces por el bucle.
 - d. Hacer m_2 pasos por el bucle con $m_2 < n_2$.
 - e. Hacer $n_2 - 1$ y $n_2 + 1$ pasos por el bucle.

- **Resultados esperados**

Con el bucle más exterior y el segundo bucle interior en sus valores mínimos dependiendo de las veces que se pasen por el bucle, se deberían imprimir todos los datos de un bien del primer departamento de UNICA.

Con el bucle más exterior en su valor mínimo y el segundo bucle interior con valor de 4, dependiendo de las veces que se pase por el bucle, debería imprimir el id de división, nombre de división, id de equipo y nombre de todos los bienes del primer departamento de UNICA.

Capítulo 4

Pruebas

Con el bucle más interior con valor de 4 y el segundo bucle interior con su valor mínimo se debería obtener los datos id de división, nombre de división, id de equipo y nombre de bien de todos los departamentos de UNICA.

- **Poscondiciones**

Existe un reporte con los bienes de cada departamento de UNICA.

Capítulo 4

Pruebas

SISTEMA DE CONTROL DE INVENTARIOS CASO DE PRUEBA

Número	Fecha	Tipo de prueba	Técnica empleada
5	20-02-2011	Prueba funcional	Partición equivalente

Descripción:	<p>Este caso de prueba está diseñado para evaluar los valores de entrada en los campos de un formulario para registrar un bien en el módulo de altas. Se selecciona una partición equivalente para cada situación que se desea evaluar. Las pruebas que se aplicarán al formulario son la siguientes:</p> <ol style="list-style-type: none">I. Seleccionar los tipos de bienes que serán dados de alta.II. Valores de entrada para número del inventario.III. Valores de entrada para el nombre de un responsable.IV. Valores de entrada para la ubicación.V. Valores de entrada para el correo electrónico de un responsable.VI. Valores de entrada para el teléfono de un responsable.VII. Valores de entrada para la fecha de garantía.VIII. Valores de entrada la velocidad de procesador para un CPU.
---------------------	---

Condición de ejecución:	<p>Para hacer cada una de las pruebas descritas anteriormente accedemos a la opción de altas en cualquiera de los menús que presenta el sistema. Para cada prueba se describirán las precondiciones, valores de entrada, observaciones, resultados esperados y precondiciones de manera independiente.</p>
--------------------------------	--

I. SELECCIONAR LOS TIPOS DE BIENES QUE SERÁN DADOS DE ALTA

- **Precondiciones**

El usuario debió haber ingresado sus credenciales en el sistema y haber elegido la opción de altas en el menú. Se debe encontrar en la primera página del módulo donde se presentan controles de selección relacionados con cada tipo de bien que es posible dar de alta.

Capítulo 4

Pruebas

- **Valores de entrada**

Valores válidos: Se selecciona uno o más controles de selección de cada tipo de bien.

Valores inválidos: No se selecciona ningún de control de selección.

- **Resultados esperados**

Cuando se selecciona uno o más controles al dar click en continuar, el sistema debe presentar en una misma página un formulario de datos para cada tipo de bien que el usuario haya elegido a través éstos.

Cuando no se selecciona ningún control al dar click en continuar, el sistema no debe permitir pasar de esta página y deberá presentar un mensaje de aviso al usuario para que elija al menos un tipo de bien.

- **Poscondiciones**

Ninguna.

II. VALORES DE ENTRADA PARA EL NÚMERO DE INVENTARIO

- **Precondiciones**

El usuario debió haber registrado sus credenciales en el sistema, haber elegido la opción de altas en el menú y haber elegido al menos un tipo de bien y por último encontrarse en la página de formulario de datos del o de los tipos de bienes elegidos.

- **Valores de entrada**

Valores válidos: Números de máximo 15 cifras que no estén ya registrados en la base de datos.

Valores inválidos: Letras o cualquier carácter que no sea dígito.

- **Resultados esperados**

Cuando un usuario introduce información en el campo de *número de inventario* y da click en continuar, si ha llenado correctamente los demás campos del formulario y el número no ha sido registrado con anterioridad en la base de datos, el bien se dará de alta y se informará al usuario que los datos han sido registrados.

Cuando un usuario introduce información en el campo de *número de inventario* y da click en continuar pero el número de inventario ya ha sido dado de alta con anterioridad en la base de datos, el sistema deberá enviar una notificación y no permitirá registrar el bien.

Capítulo 4

Pruebas

Cuando el usuario introduce letras o cualquier carácter inválido el sistema no permitirá dar de alta la información en la base de datos y enviará una notificación al usuario para indicarle el error.

El sistema evitará que el usuario intente introducir números mayores a 15 cifras bloqueando la caja de texto.

- **Poscondiciones**

Hay un nuevo registro en la base de datos por cada bien especificado por el usuario, con sus respectivos números de inventario asignados.

III. VALORES DE ENTRADA PARA EL NOMBRE DE UN RESPONSABLE

- **Precondiciones**

El usuario debe haber registrado credenciales en el sistema, haber elegido la opción de altas en el menú, haber elegido al menos un tipo de bien, haber elegido la opción de responsable nuevo y finalmente encontrarse en la página de formulario de datos.

- **Valores de entrada**

Valores válidos: Cadena no mayor a 65 caracteres de letras mayúsculas minúsculas y espacios.

Valores inválidos: Números o cualquier otro carácter diferente a letras o espacios.

- **Resultados esperados**

Cuando el usuario introduce caracteres válidos en el campo de nombre de responsable, si ha llenado los demás campos de formulario correctamente y el responsable no existe en la base de datos, el sistema dará de alta al nuevo registro y notificará al usuario.

Cuando el usuario introduce caracteres válidos en el nombre de responsable pero éste ya existe en la base de datos, el sistema no permitirá dar de alta el registro y notificará al usuario que el responsable ya existe.

Cuando el usuario introduce caracteres inválidos en el campo de nombre de responsable el sistema no permitirá dar de alta el registro en la base de datos y notificará al usuario que ha introducido caracteres inválidos.

Cuando el usuario intente introducir cadenas mayores a 65 caracteres en el campo de nombre de responsable el sistema bloquea la caja de texto, para permitir un máximo de 50 caracteres.

Capítulo 4

Pruebas

- **Poscondiciones**

Hay un nuevo registro de responsable con el nombre especificado por el usuario.

IV. VALORES DE ENTRADA PARA LA UBICACIÓN

- **Precondiciones**

El usuario debe haber registrado credenciales en el sistema, haber elegido la opción de altas en el menú, haber elegido al menos un tipo de bien, haber elegido la opción de responsable nuevo o existente y finalmente encontrarse en la página de formulario de datos.

- **Valores de entrada**

Valores válidos: Cadena no mayor a 85 caracteres de letras mayúsculas minúsculas, números y espacios y puntos.

Valores inválidos: Cualquier otro carácter no especificado en los valores válidos.

- **Resultados esperados**

Cuando el usuario introduce caracteres válidos en el campo de ubicación y ha llenado correctamente los demás campos, el sistema registra los datos en la base y notifica al usuario.

Cuando el usuario introduce caracteres inválidos no permite dar de alta los datos y notifica al usuario que algunos caracteres no son válidos.

Cuando el usuario intenta introducir cadenas mayores a 85 caracteres en el campo de ubicación el sistema bloquea la caja de texto limitándola a un máximo de 85 caracteres.

- **Poscondiciones**

Existe en la base de datos la ubicación que especificó el usuario.

V. VALORES DE ENTRADA PARA EL CORREO ELETRÓNICO DE UN RESPONSABLE

- **Precondiciones**

El usuario debe haber registrado credenciales en el sistema, haber elegido la opción de altas en el menú, haber elegido al menos un tipo de bien, haber elegido la opción de responsable nuevo y finalmente encontrarse en la página de formulario de datos.

Capítulo 4

Pruebas

- **Valores de entrada**

Valores válidos: Cadena no mayor a 60 caracteres que se ajusten a la expresión regular:

`^[a-zA-Z]+((\._|-|_)*[a-zA-Z0-9]+)*@[a-z]+(\.\[_a-z]+)+$`

Valores inválidos: Cualquier otro carácter no especificado en los valores válidos.

- **Resultados esperados**

Cuando el usuario introduce caracteres válidos en el campo de correo electrónico y ha llenado correctamente los demás campos, el sistema registra los datos en la base y notifica al usuario.

Cuando el usuario introduce caracteres inválidos no permite dar de alta los datos y notifica al usuario que algunos caracteres no son válidos.

Cuando el usuario intenta introducir cadenas mayores a 60 caracteres en el campo de correo electrónico el sistema bloquea la caja de texto limitándola a un máximo de 60 caracteres.

- **Poscondiciones**

Existe en la base de datos el correo electrónico que especificó el usuario.

VI. VALORES DE ENTRADA PARA EL TELÉFONO DE UN RESPONSABLE

- **Precondiciones**

El usuario debe haber registrado credenciales en el sistema, haber elegido la opción de altas en el menú, haber elegido al menos un tipo de bien, haber elegido la opción de responsable nuevo y finalmente encontrarse en la página de formulario de datos.

- **Valores de entrada**

Valores válidos: Cadena no mayor a 20 caracteres de números y/o guiones.

Valores inválidos: Cualquier otro carácter no especificado en los valores válidos.

- **Resultados esperados**

Cuando el usuario introduce caracteres válidos en el campo de teléfono y ha llenado correctamente los demás campos, el sistema registra los datos en la base y notifica al usuario.

Cuando el usuario introduce caracteres inválidos el sistema no permite dar de alta los datos y notifica al usuario que algunos caracteres no son válidos.

Cuando el usuario intenta introducir cadenas mayores a 20 caracteres en el campo de teléfono el sistema bloquea la caja de texto limitándola a un máximo de 20 caracteres.

Capítulo 4

Pruebas

- **Poscondiciones**

Existe en la base de datos el teléfono que especificó el usuario.

VII. VALORES DE ENTRADA PARA LA FECHA DE GARANTÍA

- **Precondiciones**

El usuario debió haber registrado sus credenciales en el sistema, haber elegido la opción de altas en el menú y haber elegido al menos un tipo de bien y por último encontrarse en la página de formulario de datos del o de los tipos de bienes elegidos.

- **Valores de entrada**

Valores válidos: Cadena con la forma "aaaa-mm-dd" donde, dd es un número de 1 a 31 expresado con dos dígitos para el día, mm es un número de 1 a 12 expresado en dos dígitos para el mes y aaaa es el número del año expresado con cuatro dígitos.

Valores inválidos: Cualquier otro carácter no especificado en los valores válidos.

- **Observaciones**

El sistema proporciona un calendario en el que el usuario puede elegir la fecha que desea y sistema automáticamente generará el formato correcto para registrar la información en la base, no obstante el usuario es libre de poner cualquier carácter en el campo de fecha de garantía.

- **Resultados esperados**

Cuando el usuario introduce caracteres válidos en el campo de fecha de garantía y ha llenado correctamente los demás campos, el sistema da de alta los datos en la base de datos y notifica al usuario.

Cuando el usuario introduce caracteres inválidos el sistema no permite dar de alta los datos y notifica al usuario que algunos caracteres no son válidos.

- **Poscondiciones**

Existe en la base de datos la fecha de garantía especificada por el usuario.

VIII. VALORES DE ENTRADA PARA LA VELOCIDAD DE PROCESADOR DE UN CPU

- **Precondiciones**

El usuario debió haber registrado sus credenciales en el sistema, haber elegido la opción de altas en el menú y haber elegido al menos el tipo de bien CPU y por último encontrarse en la página de formulario de datos del o de los tipos de bienes elegidos.

- **Valores de entrada**

Valores válidos: Cadena de sólo 5 dígitos.

Valores inválidos: Cualquier otro carácter no especificado en los valores válidos.

- **Resultados esperados**

Cuando el usuario introduce caracteres válidos en el campo de velocidad y ha llenado correctamente los demás campos, el sistema registra los datos en la base y notifica al usuario.

Cuando el usuario introduce caracteres inválidos el sistema no permite dar de alta los datos y notifica al usuario que el campo sólo acepta números

- **Poscondiciones**

Existe en la base de datos la velocidad de procesador para el CPU especificada por el usuario.

Capítulo 4

Pruebas

SISTEMA DE CONTROL DE INVENTARIOS CASO DE PRUEBA

Número	Fecha	Tipo de prueba	Técnica empleada
6	20-02-2011	Prueba funcional	Partición equivalente

Descripción:	<p>Este caso de prueba está diseñado para evaluar los valores de entrada en los campos de un formulario para modificar los datos de un bien en el módulo de modificaciones. Se selecciona una partición equivalente para cada situación que se desea evaluar. Las pruebas que se aplicarán al formulario son la siguientes:</p> <ul style="list-style-type: none">I. Valores de entrada para número del inventario.II. Valores de entrada para el nombre de un responsable.III. Valores de entrada para la ubicación.IV. Valores de entrada para el correo electrónico de un responsable.V. Valores de entrada para el teléfono de un responsable.VI. Valores de entrada para la fecha de garantía.
---------------------	--

Condición de ejecución:	<p>Para hacer cada una de las pruebas descritas anteriormente accedemos a la opción de modificaciones en cualquiera de los menús que presenta el sistema y en la primera página del módulo que pregunta si se desea modificar un equipo o un responsable, se debe elegir la opción de equipo, después se debe buscar el bien ya sea por responsable o con algún dato como número de inventario o de serie.</p> <p>Para cada prueba se describirán las precondiciones, valores de entrada, observaciones, resultados esperados y precondiciones de manera independiente.</p>
--------------------------------	---

I. VALORES DE ENTRADA PARA NÚMERO DE INVENTARIO

- **Precondiciones**

El usuario debió haber registrado sus credenciales en el sistema, haber elegido la opción de modificaciones en el menú y haber elegido la opción para modificar un equipo, buscar el bien de interés y por último encontrarse en la página de formulario de datos del bien elegido.

- **Valores de entrada**

Capítulo 4

Pruebas

Valores válidos: Números de máximo 15 cifras que no estén ya registrados en la base de datos

Valores inválidos: Letras o cualquier carácter que no sea dígito

- **Resultados esperados**

Cuando un usuario introduce datos en el campo de *número de inventario* y da click en continuar, si ha llenado correctamente los demás campos del formulario y el número no ha sido registrado con anterioridad en la base de datos, el bien se actualizará y se informará al usuario que los datos han sido actualizados

Cuando un usuario introduce datos en el campo de *número de inventario* y da click en continuar pero el número de inventario ya ha sido dado de alta con anterioridad en la base de datos, el sistema deberá enviar una notificación y no permitirá actualizar la información.

Cuando el usuario introduce letras o cualquier carácter inválido el sistema no permitirá dar de alta los datos en la base y enviará una notificación al usuario para indicarle el error al usuario.

El sistema evitará que el usuario intente introducir números mayores a 15 cifras bloqueando la caja de texto.

- **Poscondiciones**

Los datos del bien se han actualizado en la base de datos.

II. VALORES DE ENTRADA PARA NOMBRE DE RESPONSABLE

- **Precondiciones**

El usuario debió haber registrado sus credenciales en el sistema, haber elegido la opción de modificaciones en el menú y haber elegido la opción para modificar un equipo, buscar el bien de interés y en la página de resultados de la búsqueda elegir la opción de responsable nuevo y por último encontrarse en la página de formulario de datos del bien elegido.

- **Valores de entrada**

Valores válidos: Cadena no mayor a 65 caracteres de letras mayúsculas minúsculas y espacios

Valores inválidos: Números o cualquier otro carácter diferente a letras o espacios.

Capítulo 4

Pruebas

- **Resultados esperados**

Cuando el usuario introduce caracteres válidos en el campo *de nombre de responsable*, si ha llenado los demás campos de formulario correctamente y el responsable no existe en la base de datos el sistema actualizará el nuevo responsable a cargo del bien y notificará al usuario.

Cuando el usuario introduce caracteres válidos en el *nombre de responsable* pero el responsable ya existe en la base de datos el sistema no permitirá actualizar el campo y notificará al usuario que el registro ya existe.

Cuando el usuario introduce caracteres inválidos en el campo de *nombre de responsable* el sistema no permitirá actualizar el campo en la base de datos y notificará al usuario que ha introducido caracteres inválidos.

Cuando el usuario intente introducir cadenas mayores a 65 caracteres en el *campo de nombre de responsable* el sistema bloquea la caja de texto limitándola para un máximo de 50 caracteres.

- **Poscondiciones**

El bien tiene asignado un nuevo responsable que no existía anteriormente.

III. VALORES DE ENTRADA PARA LA UBICACIÓN

- **Precondiciones**

El usuario debió haber registrado sus credenciales en el sistema, haber elegido la opción de modificaciones en el menú y haber elegido la opción para modificar un equipo, buscar el bien de interés y por último encontrarse en la página de formulario de datos del bien elegido.

- **Valores de entrada**

Valores válidos: Cadena no mayor a 85 caracteres de letras mayúsculas minúsculas, números y espacios y puntos.

Valores inválidos: Cualquier otro carácter no especificado en los valores válidos.

- **Resultados esperados**

Cuando el usuario introduce caracteres válidos en el *campo de ubicación* y ha llenado correctamente los demás campos el sistema actualiza los datos en la base y notifica al usuario.

Capítulo 4

Pruebas

Cuando el usuario introduce caracteres inválidos no permite dar de alta los datos y notifica al usuario que algunos caracteres no son válidos.

Cuando el usuario intenta introducir cadenas mayores a 85 caracteres en el *campo de ubicación* el sistema bloquea la caja de texto limitándola a un máximo de 85 caracteres.

- **Poscondiciones**

Los datos del bien se han actualizado en la base de datos.

IV. VALORES DE ENTRADA PARA CORREO ELECTRÓNICO DE UN RESPONSABLE

- **Precondiciones**

El usuario debió haber registrado sus credenciales en el sistema, haber elegido la opción de modificaciones en el menú y haber elegido la opción para modificar un equipo, buscar el bien de interés y en la página de resultados de la búsqueda elegir la opción de responsable nuevo y por último encontrarse en la página de formulario de datos del bien elegido.

- **Valores de entrada**

Valores válidos: Cadena no mayor a 60 caracteres que se ajusten a la expresión regular:
`^[a-zA-Z]+((\._|-|_)*[a-zA-Z0-9]+)*@[a-z]+(\.\[_a-z]+)+$`

Valores inválidos: Cualquier otro carácter no especificado en los valores válidos.

- **Resultados esperados**

Cuando el usuario introduce caracteres válidos en el *campo de correo electrónico* y ha llenado correctamente los demás campos el sistema actualiza los datos en la base y notifica al usuario.

Cuando el usuario introduce caracteres inválidos no permite actualizar los datos y notifica al usuario que algunos caracteres no son válidos.

Cuando el usuario intenta introducir cadenas mayores a 60 caracteres en el *campo de correo electrónico* el sistema bloquea la caja de texto limitándola a un máximo de 60 caracteres.

- **Poscondiciones**

Los datos del bien se han actualizado en la base de datos.

V. VALORES DE ENTRADA PARA TELÉFONO DE UN RESPONSABLE

- **Precondiciones**

El usuario debió haber registrado sus credenciales en el sistema, haber elegido la opción de modificaciones en el menú y haber elegido la opción para modificar un equipo, buscar el bien de interés y en la página de resultados de la búsqueda elegir la opción de responsable nuevo y por último encontrarse en la página de formulario de datos del bien elegido.

- **Valores de entrada**

Valores válidos: Cadena no mayor a 20 caracteres de números y/o guiones.

Valores inválidos: Cualquier otro carácter no especificado en los valores válidos.

- **Resultados esperados**

Cuando el usuario introduce caracteres válidos en el *campo de teléfono* y ha llenado correctamente los demás campos, el sistema actualiza los datos en la base y notifica al usuario.

Cuando el usuario introduce caracteres inválidos el sistema no permite actualizar los datos y notifica al usuario que algunos caracteres no son válidos.

Cuando el usuario intenta introducir cadenas mayores a 20 caracteres en el *campo de teléfono* el sistema bloquea la caja de texto limitándola a un máximo de 20 caracteres.

- **Poscondiciones**

Los datos del bien se han actualizado en la base de datos.

VI. VALORES DE ENTRADA PARA LA FECHA DE GARANTÍA

- **Precondiciones**

El usuario debió haber registrado sus credenciales en el sistema, haber elegido la opción de modificaciones en el menú y haber elegido la opción para modificar un equipo, buscar el bien de interés y por último encontrarse en la página de formulario de datos del bien elegido.

Capítulo 4

Pruebas

- **Valores de entrada**

Valores válidos: Cadena con la forma "aaaa-mm-dd" donde, dd es un número de 1 a 31 expresado con dos dígitos para el día, mm es un número de 1 a 12 expresado en dos dígitos para el mes y aaaa es el número del año expresado con cuatro dígitos.

Valores inválidos: Cualquier otro carácter no especificado en los valores válidos.

- **Observaciones**

El sistema proporciona un calendario en el que el usuario puede elegir la fecha que desea y sistema automáticamente generará el formato correcto para dar de alta en la base de datos, no obstante el usuario es libre de poner cualquier carácter en el campo de fecha de garantía.

- **Resultados esperados**

Cuando el usuario introduce caracteres válidos en el campo de fecha de garantía y ha llenado correctamente los demás campos, el sistema actualiza los datos en la base y notifica al usuario.

Cuando el usuario introduce caracteres inválidos el sistema no permite actualizar los datos y notifica al usuario que algunos caracteres no son válidos.

- **Poscondiciones**

Los datos del bien se han actualizado en la base de datos.

5

IMPLEMENTACIÓN

El presente capítulo presentan dos manuales que son la piedra angular para la implementación de cualquier producto de software:

El Manual de instalación muestra paso a paso como se debe hacer el despliegue del sistema, el cual involucra la establecimiento de un servidor de bases de datos y un servidor web, realizar la configuración de las conexiones a estos servidores.

Y Finalmente el Manual de usuario instruye sobre el uso correcto de cada una de las funciones del sistema.

Contenido:

- 5.1 Manual de Instalación
- 5.2 Manual de Usuario

Capítulo 5

Implementación

5.1 Manual de Usuario

Ingresando al sistema

Lo primero que el sistema hace, es verificar que exista un usuario válido y permite acceder al sistema (Figura 5.1).

- 1.-El usuario ingresa su clave.
- 2.-El usuario ingresa su contraseña.
- 3.-El sistema verifica que el usuario y contraseña sean válidos
- 4.-El sistema presenta las opciones a realizar de acuerdo con el tipo de usuario.
- 5.-En caso de que el usuario y/o la contraseña sean incorrectos, el sistema alerta al usuario y le permite ingresar los datos nuevamente.

Figura 5.1 Campos para ingresar al sistema.

Dentro del sistema se identifican cuatro partes principales que se muestran en la figura 5.2:

- Menú Superior
- Menú Lateral
- Área Principal
- Barra ayuda



Figura 5.2 Menú principal.

Capítulo 5

Implementación

Menú Superior

En la figura 5.3, se puede observar dicho menú, el cual será visible para el usuario en todas las secciones que se visiten dentro del sistema, permite dirigirse a los distintos módulos del sistema.



Figura 5.3 Menú superior.

Menú Lateral

Además de permitir seleccionar los módulos del sistema, también ofrece la opción de generación de reportes y cerrar la sesión de usuario.



Figura 5.4 Menú lateral.

Área principal

Esta será la sección que cambiara dependiendo del Módulo que se seleccione.

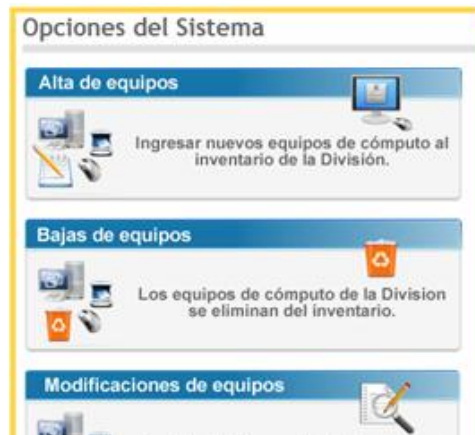


Figura 5.5 Área Principal.

Barra de ayuda

Esta sección será una guía para el usuario, ya que muestra los pasos a seguir en cada pantalla. en cada

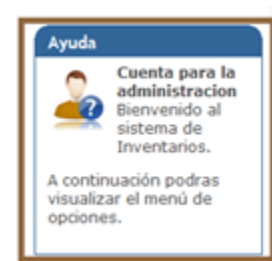


Figura 5.6 Barra de ayuda.

Capítulo 5

Implementación

Módulo de Altas

Dar de alta un equipo

1.- Una vez que se ingresa al sistema hay tres formas de iniciar al módulo de altas:

- Utilizando el menú superior
- Utilizando el menú lateral
- Utilizando el menú que se localiza en el área principal.

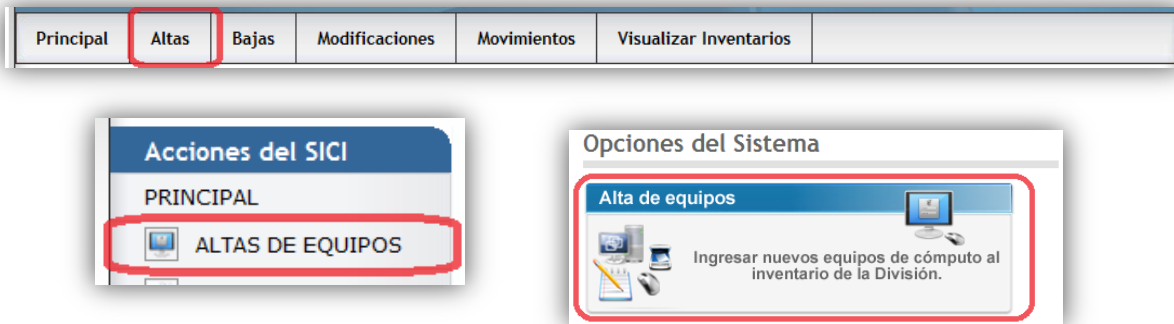


Figura 5.7 Ingresar al módulo de Altas.

2.- El usuario elige los bienes que formarán parte del nuevo *EQUIPO*, estos pueden ser uno o más *bienes* (ó dispositivos de cómputo).

IMPORTANTE: *Seleccionar más de un dispositivo, producirá que el sistema registre un nuevo EQUIPO y éste será conformado por cada uno de los bienes seleccionados, esto es, cada uno de los dispositivos seleccionados serán referenciados por un numero UNICO INTERNAMENTE.*

3.-El usuario elige la forma de asignar un responsable al equipo:

Asignar un responsable existente.- Seleccionar esta opción si el responsable del equipo de computo ya fue registrado en el sistema y ya tienen equipos a su cargo.

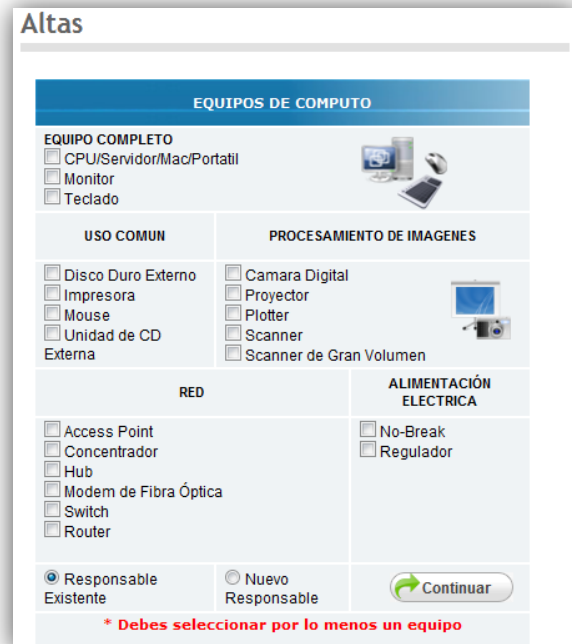


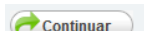
Figura 5.8 Selección de bienes.

Capítulo 5

Implementación

Asignar un responsable nuevo.- Esta opción permite dar de alta al responsable del equipo si éste no ha sido registrado previamente.

*Nota: Todo equipo de cómputo debe tener una persona que se haga responsable de dicho bien, por lo tanto deberá proporcionarse la información del encargado del equipo.
Un EQUIPO se refiere a un conjunto de dispositivos (o bienes) que se utilizan de manera conjunta.*

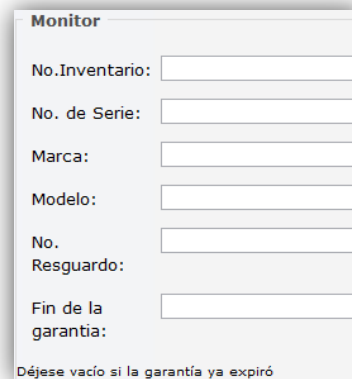
Una vez seleccionados los equipos que se desean registrar, se continúa con el alta seleccionando dando click el botón de continuar. 

Seleccionando la opción de:

- *Asignar un responsable existente*
- El sistema despliega los campos correspondientes para dar de alta los datos de cada bien (Figura 5.9).
- El usuario ingresa los datos de cada bien que componen el equipo

Nota. El número de inventario es único y en caso de ingresar un número ya registrado el sistema lo indicará.

Si el equipo no tiene garantía el campo correspondiente se deberá dejar en blanco.



Monitor

No. Inventario:

No. de Serie:

Marca:

Modelo:


No. Resguardo:

Fin de la garantía:

Déjese vacío si la garantía ya expiró

Figura 5.9 Campos de captura.

- El sistema despliega un combo con la lista de responsables dados de alta en el sistema (Figura 5.10).
- El usuario elige de la lista de usuarios existentes un responsable para el equipo
- El usuario ingresa la ubicación física del equipo.



Datos del Responsable

Responsable:

Ubicación:

Datos del Responsable

Responsable:

Ubicación:

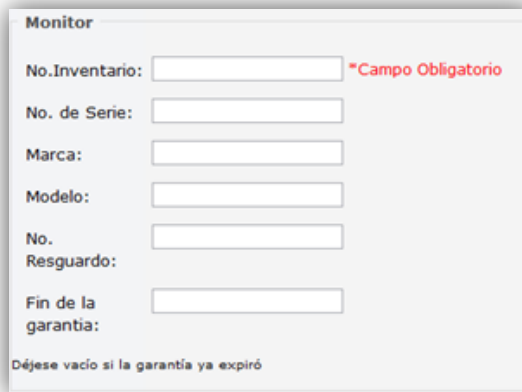
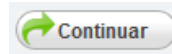
Morales Crispin Julio Cesar
Barragan Paz Ma. Rosario
Gallo Rivera Diana
Baltazar Galvez Jose Miguel
Rangel Guerrero Alberto
Figueroa Figueroa Jorge
García Castillo Gamar Zaid
Durán Campos Victor
Sandoval Vazquez Ricardo
Garcia Silva Adan Alberto
C. Cecilia Espinosa
Martinez Espinoza Jose Giovanni
Velazquez Sedan Nelida
Alvarez Chimal Leonardo

Figura 5.10 Listado de responsables

Capítulo 5

Implementación

Se validan los datos haciendo click en el boton continuar:



Monitor

No.Inventario: *Campo Obligatorio

No. de Serie:

Marca:

Modelo:

No. Resguardo:

Fin de la garantía:

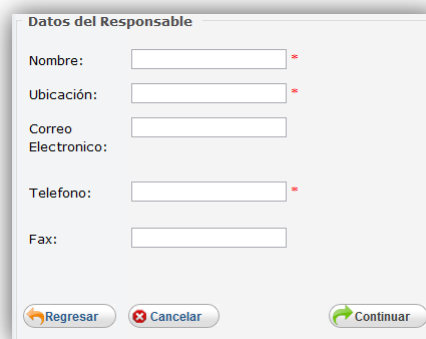
Déjese vacío si la garantía ya expiró

Nota. Si alguno de los datos obligatorios no fue ingresado o alguno de los campos no cumple las especificaciones de validación, el sistema envía una alerta al usuario y no permitirá continuar con el proceso.

Figura 5.11 Validación de información

Seleccionando la opción de:

- *Asignar un responsable nuevo*
- El sistema despliega los campos correspondientes para dar de alta los datos de cada bien (Figura 5.9).
- El usuario ingresa los datos de cada bien que componen el equipo.
- El sistema despliega los campos para dar de alta los datos del responsable nuevo (Figura 5.12).
- El usuario da ingresa el nombre, correo, teléfono y fax del responsable del equipo
- El usuario ingresa la ubicación física del equipo.



Datos del Responsable

Nombre: *

Ubicación: *

Correo Electronico:

Telefono: *

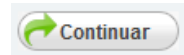
Fax:

Figura 5.12 Campos para datos de responsables

Capítulo 5

Implementación

Se validan los datos haciendo click en el boton continuar



Datos del Responsable

Nombre: *Campo Obligatorio

Ubicacion: *Campo Obligatorio

Correo Electronico:

Telefono: *Campo Obligatorio

Fax:

Nota. Si alguno de los datos obligatorios no fue ingresado o alguno de los campos no cumple las especificaciones de validación, el sistema envía una alerta al usuario y no permitirá continuar con el proceso.

Figura 5.13 Validación de información

4.-Una vez que los datos han sido ingresados, el sistema los validará y si todo es correcto mostrara la pantalla de confirmación de dicha información (Figura 5.14).

El sistema muestra la pagina de confirmación de datos y permite:

- Continuar
- Modificar
- Cancelar

Altas

Monitor

idMonitor	2624
No. de Inventario:	234234234
No. de Serie:	SERIEX4546546
Marca:	DELL
Modelo:	F900
No. de Resguardo:	
Fecha de Ingreso:2011-02-05	Fecha de mov:2011-02-05
Movimiento:A	Estado:A
Fin de la garantia: 2012-02-01	

Ayuda

Estos son los datos que fueron ingresados.

Favor de verificarlos.

Se pueden realizar las siguientes acciones:

- MODIFICAR:** Regresar para cambiar algun datos.
- CANCELAR:** Anula toda la operación.
- CONTINUAR:** Se ingresa la información y se termina con la tarea.

Figura 5.14 Confirmación de información de bienes

Capítulo 5

Implementación

Responsable	
Nombre:	Carlos Aleman
Ubicacion:	Edificio Principal
Telefono:	56581106
E-mail:	caleman@gmail.com
Fax:	

Figura 5.15 Confirmación de información de responsable

Seleccionando la opción de: **Continuar**

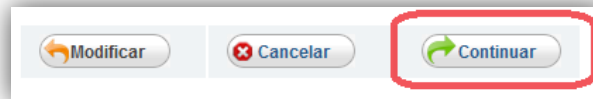



Figura 5.16 Confirmar los datos

- Se registra la información en la base de datos.
- Se notifica al usuario que los datos han sido dados de alta exitosamente (Figura 5.17).

Altas

 Alta realizada satisfactoriamente.

[Regresar a Altas](#)

Ayuda

Alta realizada satisfactoriamente.

Figura 5.17 Registro exitoso

Capítulo 5

Implementación

Módulo de Bajas

Dar de baja un equipo

Descripción: Permite eliminar del sistema los datos de un equipo

1.- Una vez que se ingresa al sistema hay tres formas de ingresar al módulo de bajas:

- Utilizando el menú superior
- Utilizando el menú lateral
- Utilizando el menú que se localiza en el área principal.



Figura 5.18 Ingresar al módulo de bajas.

2.- El usuario elige la forma de buscar el bien que desea eliminar (Figura 5.19):

- *Buscar por Responsable del equipo*
- *Buscar por Equipo*

El formulario 'Baja de equipos' muestra la opción 'Nombre del Responsable' seleccionada con un botón 'Continuar' habilitado y un botón 'Cancelar' deshabilitado. La opción 'Datos del Equipo' no está seleccionada.

Ayuda

1.-Para dar de baja, se debe especificar un equipo.

Hay dos formas de buscar el equipo:

Eligir una opción:

El nombre del Responsable:
Selecciona el nombre de un responsable, para ver todos lo equipos que tiene a su cargo.

Datos del Equipo:
Busca el equipo por su número de inventario, o su numero de serie.

Figura 5.19 Seleccionar tipo de búsqueda de equipo

Capítulo 5

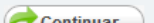
Implementación

Seleccionando la opción de:

- *Buscar por Responsable del equipo*
- En esta opción se mostraran todos los responsables que tienen equipos a su cargo dentro de la división.



Figura 5.20 Listado de responsables

3.- Una vez seleccionado el responsable, se muestran todos los equipos que tiene a su cargo haciendo click en el boton 

Se muestra un listado de todos los equipos que tiene el responsable seleccionado, en esta pantalla se selecciona el equipo que se quiere eliminar del inventario.

Equipos del responsable: Carlos Aleman					
Bien	No. DE INVENTARIO	No. DE SERIE	MARCA	MODELO	
MONITOR	234234234	SERIEX4546546	DELL	F900	<input checked="" type="radio"/>
MONITOR	435435435	SX5444SD6FGG	DELL	F900	<input type="radio"/>
IMPRESORA	8798797699	SDD5WRGVGT4GX	HP		<input type="radio"/>

Figura 5.21 Listado de equipos

Un equipo no puede estar armado con más de un bien del mismo tipo, es decir, no puede tener dos monitores, o dos CPU'S. En estos casos se debe dar de alta cada bien por separado y asignarlo a un mismo responsable.

Capítulo 5

Implementación

Como se muestra en la figura 5.22, si tienen dos monitores pero el sistema los identificara como monitores que pertenecen a dos equipos de cómputo distintos.

4.- Para eliminar un equipo se debe seleccionar el *EQUIPO* (ver nota) o dispositivo que se desea, y dar click en continuar:

Equipos del responsable: Carlos Aleman					
Bien	No. DE INVENTARIO	No. DE SERIE	MARCA	MODELO	
MONITOR	234234234	SERIEX4546546	DELL	F900	<input type="radio"/>
MONITOR	435435435	SX5444SD6FGG	DELL	F900	<input checked="" type="radio"/>
IMPRESORA	8798797699	SDD5WRGVGT4GX	HP		<input type="radio"/>

[Regresar](#) [Continuar](#)

Figura 5.22 Listado de equipos encontrados

Nota. Un responsable puede tener más de un equipo.

En la figura 5.23, se muestra como se deberán *seleccionar* los bienes que forman parte del equipo seleccionado previamente, esto es, se muestran los dispositivos que se dieron de alta como parte de un solo equipo y no como bienes individuales.

SELECCIONA LOS BIENES QUE SERÁN DADOS DE BAJA					
Bien	No. DE INVENTARIO	No. DE SERIE	MARCA	MODELO	
MONITOR	435435435	SX5444SD6FGG	DELL	F900	<input type="checkbox"/>
IMPRESORA	8798797699	SDD5WRGVGT4GX	HP		<input type="checkbox"/>

Al "Confirmar Baja" se eliminaran los elementos seleccionados

[Regresar](#) [Confirmar Baja](#)

Figura 5.23 Selección de equipos para dar de baja

5.- Seleccionar los bienes que se desean de eliminar y confirmar Baja

Nota. Si no se elige ningún dispositivo el sistema enviara el siguiente mensaje.

Debes de seleccionar una elemento

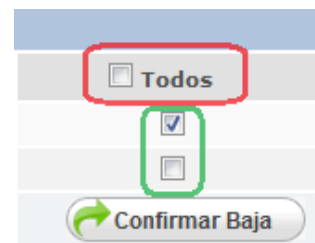


Figura 5.24 Confirmación para dar de baja

Capítulo 5

Implementación

- El sistema elimina el bien o bienes seleccionados.
- El sistema envía un mensaje para confirmar que se han dado de baja el o los bienes elegidos (Figura 5.25).

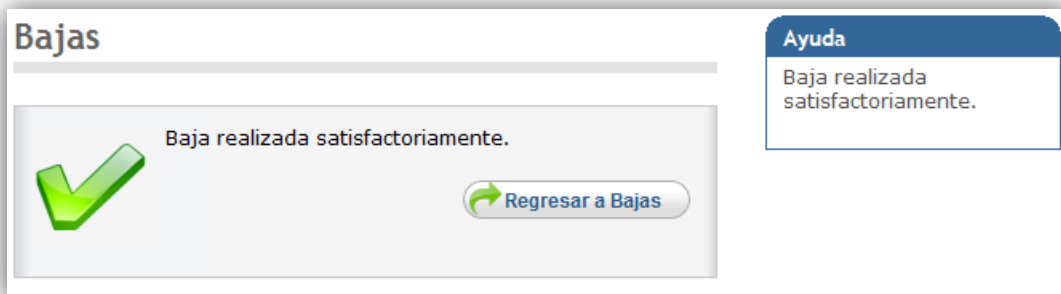


Figura 5.25 Baja exitosa

Seleccionando la opción de:

- *Buscar por Datos del Equipo*

3.- Para este caso, se busca un equipo ingresando una de los siguientes datos conocidos del equipo (Figura 5.26):

- Numero de inventario.
- Numero de serie
- Numero de resguardo.
- Modelo

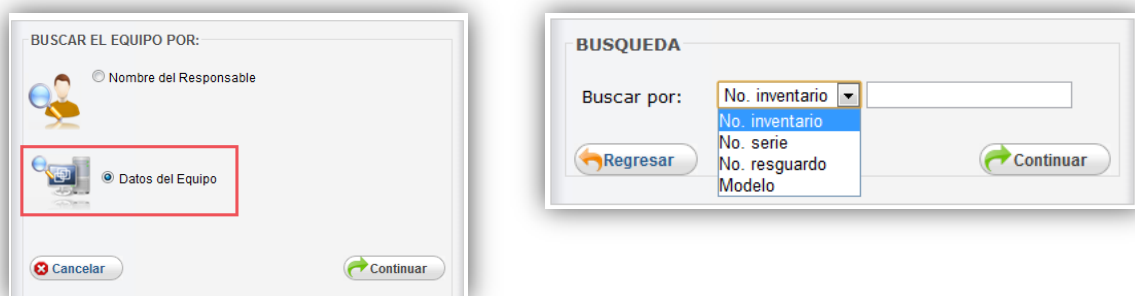


Figura 5.26 Búsqueda ingresando datos de un equipos

Capítulo 5

Implementación

Una vez que se ingresa algún dato se da click en continuar:

BUSQUEDA

Buscar por: No. serie

Figura 5.27 Confirmación de datos

Si la búsqueda tiene éxito, se muestran los dispositivos encontrados.

4.- Para eliminar un equipo se debe seleccionar el *EQUIPO* o dispositivo que se desea, y dar click en continuar:

Bajas

Equipos con no_serie = SDD5WRGVGT4GX

Bien	No. DE INVENTARIO	No. DE SERIE	MARCA	MODELO	
IMPRESORA	8798797699	SDD5WRGVGT4GX	HP		<input type="radio"/>

Figura 5.28 Listado de equipos encontrados

5.- Seleccionar los bienes que se desean de eliminar y confirmar Baja

SELECCIONA LOS BIENES QUE SERÁN DADOS DE BAJA

Bien	No. DE INVENTARIO	No. DE SERIE	MARCA	MODELO	<input type="checkbox"/> Todos
IMPRESORA	8798797699	SDD5WRGVGT4GX	HP		<input type="checkbox"/>

Al "Confirmar Baja" se eliminarán los elementos seleccionados

Figura 5.29 Selección de equipos para dar de baja

Nota. Si no se elige ningún dispositivo el sistema enviara el siguiente mensaje.

Debes de seleccionar una elemento

Todos

Figura 5.30 Confirmación para dar de baja

Capítulo 5

Implementación

- El sistema elimina el bien o bienes seleccionados.
- El sistema envía un mensaje para confirmar que se han dado de baja el o los bienes elegidos (Figura 5.31).

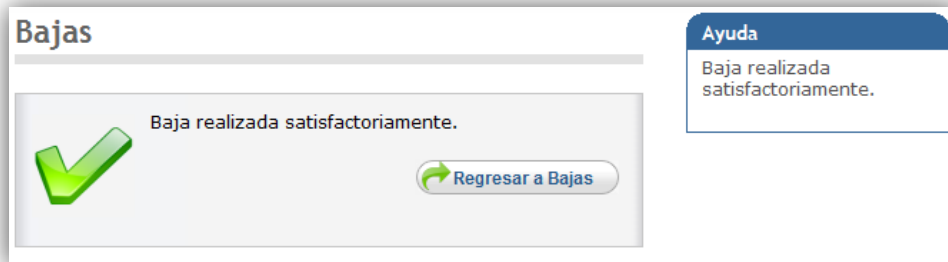


Figura 5.31 Baja exitosa

Capítulo 5

Implementación

Módulo de Modificaciones

Descripción: Permite cambiar los datos de un equipo y reasignar equipos a un responsable.

Una vez que se ingresa al sistema hay tres formas de ingresar al módulo de modificaciones:

- Utilizando el menú superior
- Utilizando el menú lateral
- Utilizando el menú que se localiza en el área principal.



Figura 5.32 Ingresar al módulo de modificaciones.

1.- El usuario elige la opción para modificar los datos de un equipo.

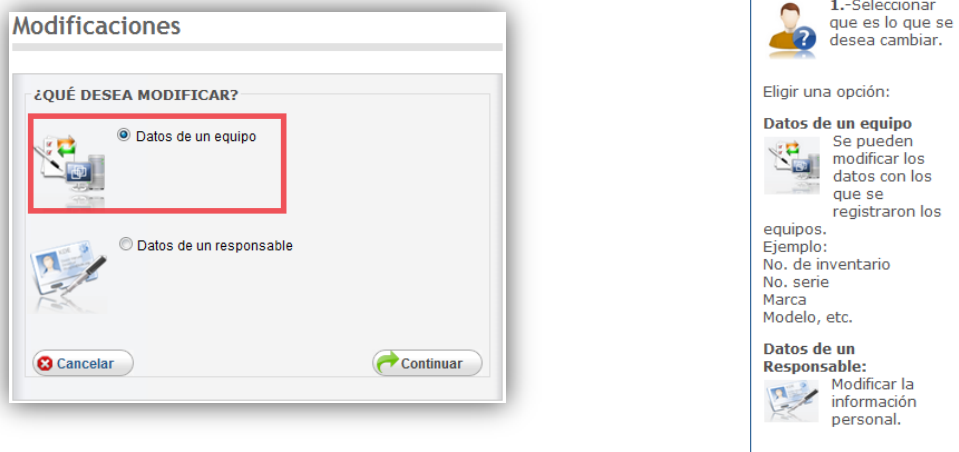


Figura 5.33 Modificar datos de un equipo.

Capítulo 5

Implementación

2.- El sistema despliega las opciones para buscar el equipo a modificar.

Se elige la forma de buscar el bien que desea modificar:

- *Buscar por Responsable del equipo*
- *Buscar por algún dato conocido del equipo*

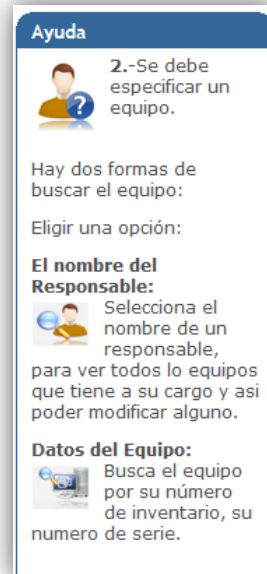


Figura 5.34 Seleccionar tipo de búsqueda de equipo.

Nota. Las pantallas de búsqueda son las mismas en cada uno de los diferentes módulos del sistema.

- Buscar por Responsable del equipo
- Buscar por algún dato conocido del equipo

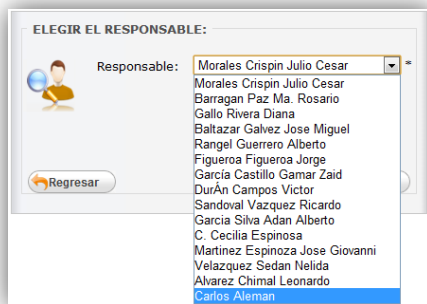


Figura 5.35 Listado de responsables.

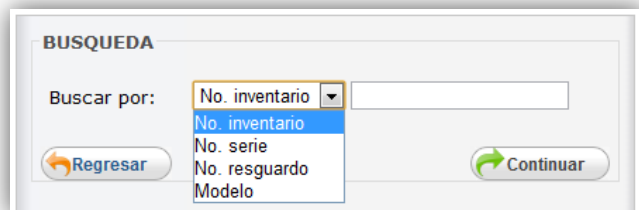


Figura 5.36 Búsqueda ingresando datos de un equipos.

Capítulo 5

Implementación

3.- Se despliega la lista de los bienes que coinciden con la búsqueda y las opciones para reasignar un responsable si así se deseara (Figura 5.37):

The screenshot shows a web interface titled "Equipos que tiene a su cargo: Velazquez Sedan Nelida". Below the title is a red instruction: "SELECCIONA EL ELEMENTO QUE SE DESEA MODIFICAR". A table lists equipment items with columns: Bien, No. DE INVENTARIO, No.DE SERIE, MARCA, MODELO, and a selection radio button. The items are: CPU (1877352, 0003985, LUFAC, 7JFQD), MONITOR (1877408, AQ15HCERA08831BMX, SAMSUNG, 551V), TECLADO (1877408, 14015677, aopen, universal), MOUSE (030415564, HP, M042KO), and IMPRESORA (1919638, USGT080430, HEWLETT PACKARD, C4170A). Below the table, a grey box contains the text "El equipo seleccionado estará acargo de:" followed by two radio buttons: "El Responsable Actual" (selected) and "Un Nuevo Responsable". At the bottom right are two buttons: "Regresar" and "Continuar".

Bien	No. DE INVENTARIO	No.DE SERIE	MARCA	MODELO	
CPU	1877352	0003985	LUFAC	7JFQD	<input checked="" type="radio"/>
MONITOR		AQ15HCERA08831BMX	SAMSUNG	551V	<input type="radio"/>
TECLADO	1877408	14015677	aopen	universal	<input type="radio"/>
MOUSE		030415564	HP	M042KO	<input type="radio"/>
IMPRESORA	1919638	USGT080430	HEWLETT PACKARD	C4170A	<input type="radio"/>

El equipo seleccionado estará acargo de:
 El Responsable Actual
 Un Nuevo Responsable

[Regresar](#) [Continuar](#)

Figura 5.37 Resultado de la búsqueda.

4.- El usuario elige uno de los bienes de la lista

- El usuario elige la forma de asignar un responsable
- Click en continuar:

This screenshot is identical to the previous one, but the "MONITOR" row in the table is highlighted with a red border, and its radio button is selected. Additionally, the grey box at the bottom containing the selection options is also highlighted with a red border.

Bien	No. DE INVENTARIO	No.DE SERIE	MARCA	MODELO	
CPU	1877352	0003985	LUFAC	7JFQD	<input type="radio"/>
MONITOR		AQ15HCERA08831BMX	SAMSUNG	551V	<input checked="" type="radio"/>
TECLADO	1877408	14015677	aopen	universal	<input type="radio"/>
MOUSE		030415564	HP	M042KO	<input type="radio"/>
IMPRESORA	1919638	USGT080430	HEWLETT PACKARD	C4170A	<input type="radio"/>

El equipo seleccionado estará acargo de:
 El Responsable Actual
 Un Nuevo Responsable

[Regresar](#) [Continuar](#)

Figura 5.38 Selección de equipos a modificar.

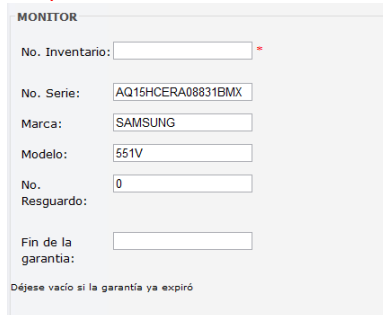
Capítulo 5

Implementación

Opciones posibles:

- *El Responsable Actual.*- Permite modificar los datos del equipo y seleccionar un responsable existente en caso de que se que quiera cambiar.

IMPORTANTE: Si el equipo está conformado por más de un bien, todos estos bienes serán reasignados al nuevo responsable.



MONITOR

No. Inventario: *

No. Serie:

Marca:

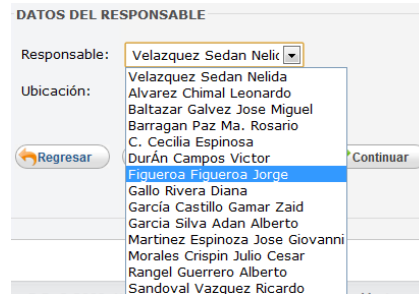
Modelo:

No. Resguardo:

Fin de la garantía:

Déjese vacío si la garantía ya expiró

Figura 5.39 Campos de información para modificar.



DATOS DEL RESPONSABLE

Responsable: *

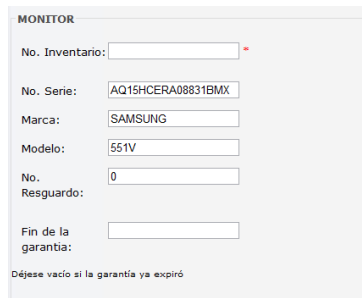
Ubicación:

- Velazquez Sedan Nelida
- Alvarez Chimal Leonardo
- Baltazar Galvez Jose Miguel
- Barragan Paz Ma. Rosario
- C. Cecilia Espinosa
- Durán Campos Victor
- Figueroa Figueroa Jorge**
- Gallo Rivera Diana
- García Castillo Gamar Zaid
- Garcia Silva Adan Alberto
- Martinez Espinoza Jose Giovanni
- Morales Crispin Julio Cesar
- Rangel Guerrero Alberto
- Sandoval Vazquez Ricardo

Figura 5.40 Listado de Responsables.

- *Un Nuevo Responsable.*-Permite modificar los datos del equipo y al mismo tiempo dar de alta un nuevo responsable del equipo.

IMPORTANTE: Si el equipo está conformado por más de un bien, todos estos bienes serán reasignados al nuevo responsable.



MONITOR

No. Inventario: *

No. Serie:

Marca:

Modelo:

No. Resguardo:

Fin de la garantía:

Déjese vacío si la garantía ya expiró

Figura 5.41 Campos de información para modificar.



DATOS DEL RESPONSABLE

Nombre: *

Correo:

Teléfono: *

Fax:

Ubicación: *

Figura 5.42 Campos para los datos del nuevo responsable.

5.- Pulsar el boton de *continuar*.

- Se validan los datos y se pide la confirmación de los cambios.
- El sistema envía un mensaje para confirmar que se ha modificado el bien elegido (Figura 5.43).

Capítulo 5

Implementación

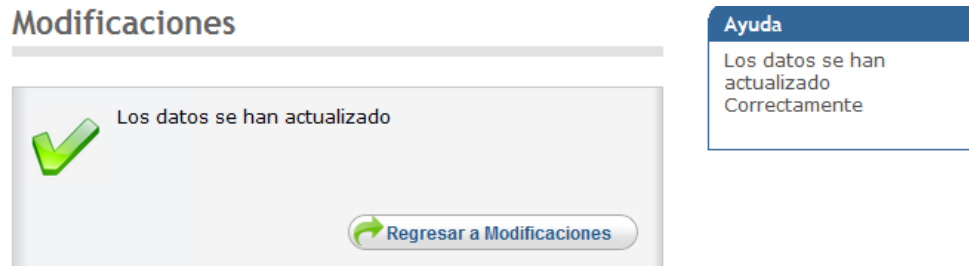


Figura 5.43 Modificación exitosa.

Modificar datos de un responsable.- Este módulo permite cambiar los de un responsable existente
1.- Seleccionar la opción para modificar los datos de un responsable



Figura 5.44 Modificar datos de un responsable.

2.- El sistema despliega la lista con los responsables de la división para modificar:

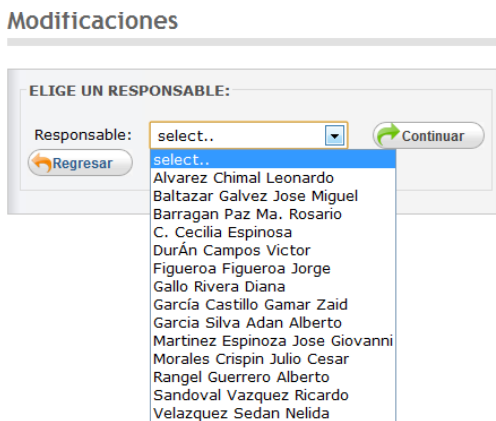


Figura 5.45 Listado de Responsables.

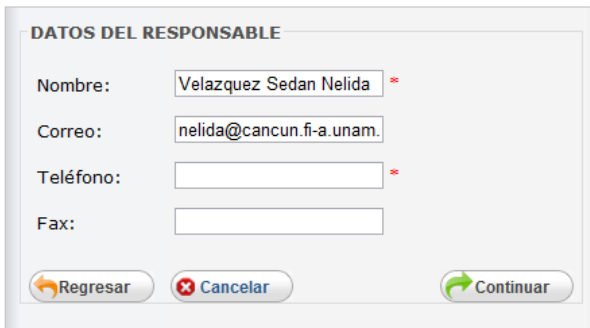
Nota. Es importante señalar que los responsables que se muestran, son únicamente los responsables dentro de la división que inicio sesión.

Capítulo 5

Implementación

3.- Se muestran los campos con los datos actuales del responsable, para que puedan ser modificados (Figura 5.46).

Modificaciones



DATOS DEL RESPONSABLE

Nombre: *


Correo:

Teléfono: *


Fax:

[Regresar](#) [Cancelar](#) [Continuar](#)

Figura 5.46 Campos a modificar.



Ayuda

 Modifica los datos necesarios, los campos con asteriscos son obligatorios.

4.- El sistema validará la información y muestra la pagina de confirmación de datos (Figura 5.47).



Modificaciones

Reponsable

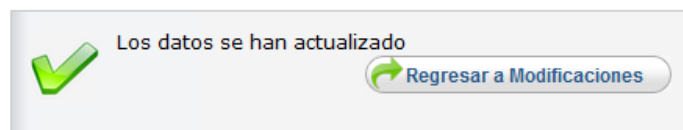
Nombre:	Velazquez Sedan Nelida
Correo:	nelida@cancun.fi-a.unam.mx
Telefono:	565811111
Fax:	


¿Los datos están correctos?

[Modificar](#) [Cancelar](#) [Confirmar](#)

Figura 5.47 Confirmación de cambios.

Modificaciones



 Los datos se han actualizado

[Regresar a Modificaciones](#)

Figura 5.48 Modificación exitosa.

Capítulo 5

Implementación

Módulo de Movimientos

Poner un bien en movimiento

Descripción

Esta opción permite quitar un equipo del inventario de una división, con la posibilidad de que otras divisiones puedan usarlo y agregarlo entre sus equipos de cómputo.

Una vez que se ingresa al sistema hay tres formas de ingresar al módulo de movimientos:

- Utilizando el menú superior
- Utilizando el menú lateral
- Utilizando el menú que se localiza en el área principal.



Figura 5.49 Ingresar al módulo de movimientos.

1.- El usuario elige la opción de: *Poner un bien en movimiento* y selecciona el tipo de búsqueda del equipo:

La imagen muestra la interfaz de usuario para el movimiento de equipos. El título es 'Movimiento de equipos'. Hay una sección 'ELEGIR UNA OPCIÓN:' con un menú desplegable que muestra 'Poner un bien en movimiento' seleccionado y resaltado con un recuadro rojo. Debajo de esto, hay un campo de texto con el texto 'Poner un bien en movimiento' y 'Asignar un bien en Movimiento'. La sección 'BUSCAR EL EQUIPO POR:' tiene dos opciones de radio: 'Nombre del Responsable' (seleccionada) y 'Datos del Equipo'. Hay un botón 'Cancelar' y un botón 'Continuar' resaltado con un recuadro rojo. En la parte inferior, hay un campo 'Crear un Equipo nuevo' y un botón 'Armar el equipo'.

Figura 5.50 Seleccionar tipo de búsqueda de equipo y la opción de movimiento.

Capítulo 5

Implementación

2.- Después que la búsqueda fue realizada, se muestran los equipos encontrados y se selecciona uno de los bienes mostrados:

Estos son los equipos que tiene a cargo: Velazquez Sedan Nelida					
Bien	No. DE INVENTARIO	No.DE SERIE	MARCA	MODELO	
CPU	1877352	0003985	LUFAC	7JFQD	<input checked="" type="radio"/>
MONITOR	565656	AQ15HCERA08831BMX	SAMSUNG	551V	<input type="radio"/>
TECLADO	1877408	14015677	aopen	universal	<input type="radio"/>
MOUSE		030415564	HP	M042KO	<input type="radio"/>
IMPRESORA	1919638	USGT080430	HEWLETT PACKARD	C4170A	<input type="radio"/>
				Regresar	Continuar

Figura 5.51 Resultados de la búsqueda.

3.-Una vez seleccionado el bien, el sistema busca todos los dispositivos vinculados a él, permitiendo poner en movimiento cada uno de los bienes (Figura 5.52):

SELECCIONA LOS BIENES QUE SERÁN PUESTOS EN MOVIMIENTO					
Bien	No. DE INVENTARIO	No.DE SERIE	MARCA	MODELO	<input type="checkbox"/> Todos
CPU	1877352	0003985	LUFAC	7JFQD	<input type="checkbox"/>
MONITOR	565656	AQ15HCERA08831BMX	SAMSUNG	551V	<input checked="" type="checkbox"/>
TECLADO	1877408	14015677	aopen	universal	<input type="checkbox"/>
MOUSE		030415564	HP	M042KO	<input type="checkbox"/>
IMPRESORA	1919638	USGT080430	HEWLETT PACKARD	C4170A	<input type="checkbox"/>
Selecciona un elemento				Regresar	Continuar

Figura 5.52 Listado de equipos

4.- El sistema pone en movimiento el bien o bienes seleccionados:

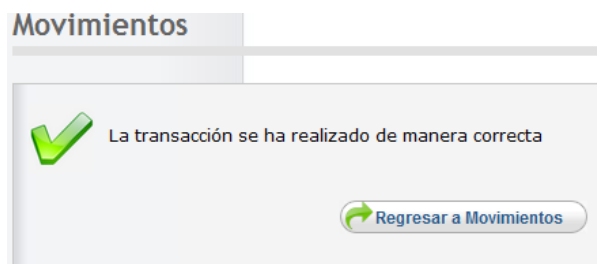


Figura 5.53 Movimiento exitoso.

Por último, se envía un mensaje al usuario para informar que los bienes han sido puestos en movimiento.

Nota. Si ocurre un problema el sistema muestra un mensaje de error.

Capítulo 5

Implementación

Importante: El bien sigue existiendo en la base de datos pero no pertenece a ninguna división por lo tanto no puede ser dado de baja ni hacer modificaciones en sus datos hasta que sea asignado nuevamente a una división.

Asignar un bien en movimiento

Descripción

Con esta opción se pueden elegir elementos que se encuentran en 'movimiento' y agregarlos a los equipos que actualmente se tienen en la división.

1.- El usuario elige la opción de: *Asignar un bien en Movimiento (Figura 5.54)*.

Importante: Lo que se desea es asignar un bien a un equipo ya existente, es decir, que ya cuenta con algunos dispositivos, por lo tanto, se debe buscar primero el equipo al que se le quieren adicionar bienes.

Movimiento de equipos

ELEGIR UNA OPCIÓN:

Poner un bien en movimiento
Poner un bien en movimiento
Asignar un bien en Movimiento

BUSCAR EL EQUIPO POR:

Nombre del Responsable

Datos del Equipo

Cancelar Continuar

Crear un Equipo nuevo

Armar el equipo

Nota. Se puede utilizar la búsqueda del equipo existente (Equipo al que se le agregan nuevos dispositivos) utilizando la búsqueda por responsable, o con algún dato conocido del equipo.

Figura 5.54 Seleccionar tipo de búsqueda de equipo y la opción de asignar un bien en movimiento.

Capítulo 5

Implementación

2.- Después que la búsqueda fue realizada, se muestran los equipos encontrados y se selecciona uno de los bienes mostrados para poder agregarle uno nuevo:

Estos son los equipos que tiene a cargo:Velazquez Sedan Nelida					
Bien	No. DE INVENTARIO	No.DE SERIE	MARCA	MODELO	
CPU	1877352	0003985	LUFAC	7JFQD	<input checked="" type="radio"/>
TECLADO	1877408	14015677	aopen	universal	<input type="radio"/>
MOUSE		030415564	HP	M042KO	<input type="radio"/>
IMPRESORA	1919638	USGT080430	HEWLETT PACKARD	C4170A	<input type="radio"/>
				Regresar	Continuar

Figura 5.55 Resultados de la búsqueda.

Una vez seleccionado el bien, el sistema busca todos los dispositivos vinculados a él, y los muestra dentro de una tabla con el título EQUIPO SELECCIONADO (Figura 5.56):

EQUIPO SELECCIONADO				
Bien	No. DE INVENTARIO	No.DE SERIE	MARCA	MODELO
CPU	1877352	0003985	LUFAC	7JFQD
TECLADO	1877408	14015677	aopen	universal
MOUSE		030415564	HP	M042KO
IMPRESORA	1919638	USGT080430	HEWLETT PACKARD	C4170A

ELIGE LOS BIENES QUE DESEAS ASIGNAR AL EQUIPO SELECCIONADO					
Bien	No. DE INVENTARIO	No.DE SERIE	MARCA	MODELO	Todos
MONITOR	1589684	0070579-9650HI0926	SUN	NCE447L	<input type="radio"/>
MONITOR	1917809	MY02923616	HP	D8896	<input type="radio"/>
MONITOR	1274594	TA33400928	DIGITAL	BT510-B4	<input type="radio"/>
MONITOR	1917810	MY02928913	HP	D8896	<input type="radio"/>
MONITOR	1917823	MY02925197	HP	D8896	<input type="radio"/>
MONITOR	1917816	MY02925139	HP	D8896	<input type="radio"/>
MONITOR	1917820	MY03026266	HP	D8896	<input type="radio"/>

Figura 5.56 Bienes del equipo seleccionado y equipos en movimiento.

En la misma pantalla, pero en la parte inferior se hace un listado **de todos los dispositivos** que el sistema ha detectado en estado de **movimiento**, permitiendo seleccionarlos para que sean vinculados al equipo existente que previamente fue seleccionado.

Capítulo 5

Implementación

3.- Seleccionar los bienes que se desean añadir al equipo existente, de la tabla con el título (Figura 5.57):

“ELIGE LOS BIENES QUE DESEAS ASIGNAR AL EQUIPO SELECCIONADO”

ELIGE LOS BIENES QUE DESEAS ASIGNAR AL EQUIPO SELECCIONADO					
Bien	No. DE INVENTARIO	No.DE SERIE	MARCA	MODELO	Todos
MONITOR	1589684	0070579-9650HI0926	SUN	NCE447L	<input type="radio"/>
MONITOR	1917809	MY02923616	HP	D8896	<input type="radio"/>
MONITOR	1274594	TA33400928	DIGITAL	BT510-B4	<input type="radio"/>
MONITOR	1917810	MY02928913	HP	D8896	<input type="radio"/>
MONITOR	1917823	MY02925197	HP	D8896	<input type="radio"/>
MONITOR	1917816	MY02925139	HP	D8896	<input type="radio"/>
MONITOR	1917820	MY03026266	HP	D8896	<input type="radio"/>
MONITOR	1917822	MY02925833	HP	D8896	<input type="radio"/>
MONITOR	1917828	MY02925204	HP	D8896	<input type="radio"/>

Figura 5.57 Equipos en estado de movimiento.

Un equipo no puede estar armado con más de un bien del mismo tipo, es decir, no puede tener dos monitores, o dos CPU'S. Por lo tanto, el sistema solo muestra los bienes en estado de movimiento, que puedan ser agregados al equipo ya existente.

El sistema solo permite seleccionar un bien de cada uno de los diferentes tipos, es decir, un solo monitor, un solo teclado, etcétera.

Equipos seleccionados (Figura 5.58):

MONITOR	565656	AQ15HCERA08831BMX	SAMSUNG	551V	<input checked="" type="radio"/>
MONITOR	222	222	prueba	prueba	<input type="radio"/>
SCANNER	555	555	prueba	prueba	<input checked="" type="radio"/>

Figura 5.58 Selección de equipos.

4.- El sistema asigna el bien o bienes seleccionados al equipo existente (Figura 5.59):

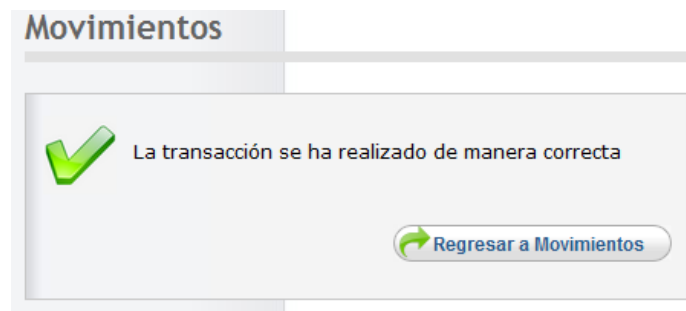


Figura 5.59 Asignación exitosa.

Capítulo 5

Implementación

Una manera de verificar que se realizó el cambio es;

- Seleccionar la opción **de Poner un bien en movimiento**.
- Buscar nuevamente el equipo que ya existía
- Observar que se le asignaron los elementos que antes estaban en movimiento (Figura 5.60).

Poner en Movimiento

Estos son los equipos que tiene a cargo: Velazquez Sedan Nelida

Bien	No. DE INVENTARIO	No.DE SERIE	MARCA	MODELO	
CPU	1877352	0003985	LUFAC	7JFQD	<input checked="" type="radio"/>
MONITOR	565656	AQ15HCERA08831BMX	SAMSUNG	551V	<input type="radio"/>
TECLADO	1877408	14015677	aopen	universal	<input type="radio"/>
MOUSE		030415564	HP	M042KO	<input type="radio"/>
IMPRESORA	1919638	USGT080430	HEWLETT PACKARD	C4170A	<input type="radio"/>
SCANNER	555	555	prueba	prueba	<input type="radio"/>

Figura 5.60 Equipo con nuevos bienes.

Crear un nuevo equipo

Descripción:

Con esta última opción se forman nuevos equipos con elementos que se encuentran en estado de movimiento.

1.-Ir la sección de *Crear un Equipo nuevo* y seleccionar el botón: *Armar el equipo*:

Movimiento de equipos

ELEGIR UNA OPCIÓN:

Poner un bien en movimiento ▾

BUSCAR EL EQUIPO POR:

Nombre del Responsable

Datos del Equipo

Cancelar Continuar

Crear un Equipo nuevo

Armar el equipo




Figura 5.61 Crear un nuevo equipo.

2.-Seleccionar quien será el responsable del equipo, indicando alguna de las siguiente opciones:

Movimiento de equipos

ASIGNAR EL EQUIPO:

A un Responsable existente

A un Responsable Nuevo

Cancelar Continuar



Figura 5.62 Seleccionar responsable

- *A un Responsable existente.*- Esta opción busca los responsables que existen en la división, permitiendo seleccionar alguno (Figura 5.63).
- *A un Responsable nuevo.*- En esta opción se registra los datos del nuevo responsable (Figura 5.64).

Capítulo 5

Implementación

A un Responsable existente

ELEGIR EL RESPONSABLE:

Responsable:

Ubicacion:

Figura 5.63 Seleccionar responsable existente.

A un Responsable nuevo

Datos del Responsable

Nombre: *

Ubicación: *

Correo Electronico:

Telefono: *

Fax: *

Figura 5.64 Seleccionar responsable nuevo.

3.-El sistema desplegara un listado de los equipos en estado de movimiento disponibles para seleccionar un bien de cada tipo.

ELIGE LOS BIENES					
Bien	No. DE INVENTARIO	No.DE SERIE	MARCA	MODELO	
CPU	1917496	MX05125535	HP	VECTRA	<input type="radio"/>
CPU	1917471	MX05125561	HP	VECTRA	<input type="radio"/>
CPU	1917487	MX05125520	HP	VECTRA	<input type="radio"/>
MOUSE	917801	1LZE00409737	HP	PS/2	<input type="radio"/>
MOUSE	131313	131313	prueba	prueba	<input type="radio"/>
IMPRESORA	777	777	prueba	prueba	<input type="radio"/>
PLOTTER	0000001	0000001			<input type="radio"/>
DD_EXTERNO	1570267	62460913	Lectura de Cinta Ext. SUN	611	<input type="radio"/>
DD_EXTERNO	141414	141414	prueba	prueba	<input type="radio"/>
CDWR_EXTERNO	161616	161616	prueba	prueba	<input type="radio"/>
HUB	666	666	prueba	prueba	<input type="radio"/>
NOBREAK	888	888	prueba	prueba	<input type="radio"/>
REGULADOR	1893423		TRIPP LITE		<input type="radio"/>
REGULADOR	1393234	35007951	VICA		<input type="radio"/>
REGULADOR	1185769	11064011	VICA		<input type="radio"/>
REGULADOR	1553199	5601515	LIGHT II		<input type="radio"/>
REGULADOR	171717	171717	prueba	prueba	<input type="radio"/>
REGULADOR	898098098098	0890809809809			<input type="radio"/>
ROUTER	111	111	prueba	prueba	<input type="radio"/>
CAMARA_DIGITAL	23434234				<input type="radio"/>

Selecciona un elemento

Figura 5.65 Equipos en estado de movimiento

Capítulo 5

Implementación

4.- Una vez que se seleccionan los dispositivos, se da click en continuar y el sistema indicara si el equipo fue creado correctamente (Figura 5.66).

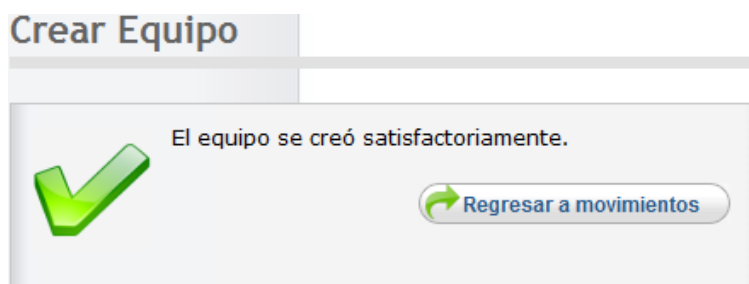


Figura 5.66 Equipo creado correctamente.

Capítulo 5

Implementación

Módulo de Visualizar Inventarios

Descripción: Permite visualizar la información completa de todos los bienes que pertenecen a la división en formato web.

1.- El usuario elige la opción de visualizar Inventarios:

Una vez que se ingresa al sistema hay tres formas de ingresar al módulo de modificaciones:

- Utilizando el menú superior
- Utilizando el menú lateral
- Utilizando el menú que se localiza en el área principal.



Figura 5.67 Ingresar al módulo de visualizar inventarios.

2.-El sistema despliega una lista con los datos de los bienes bajo la custodia del usuario (Figura 5.68):

Equipos actuales en la División:						
Bien	ID	No. DE INVENTARIO	No.DE SERIE	MARCA	MODELO	RESPONSABLE
CPU	8	1917262	CN03750190	HP	NETSERVER E200(WODAM)	Barragan Paz Ma. Rosario
CPU	12	2075596	00045-419-827-756	DELL	0932RY	Garcia Silva Adan Alberto
CPU	24	2041505		eCOIN	PLATINIUM	Gallo Rivera Diana
CPU	499	1589682	702V003F	SUN	ULTRA ENTERPRISE	nuevo
CPU	1017	1959695	MX10835268	HP	VECTRA VL 400	Alvarez Chimal Leonardo
CPU	1040	1877352	0003985	LUFAC	7JFQD	Velazquez Sedan Nelida
CPU	1600	2227645	1HSPQB1	Dell	optiplex gx620	Durán Campos Victor
CPU	1610	2117274	1XPX731	DELL	OPTIPLEX GX270	Baltazar Galvez Jose Miguel
CPU	1622	2117277	8WPX731	DELL	OPTIPLEX GX270	Rangel Guerrero Alberto
CPU	2169	2180197	32H0351	Dell	Optiplex GX 620	Barragan Paz Ma. Rosario
CPU	2192	1564687				Barragan Paz Ma. Rosario
CPU	2194	1959181	MY10835249	HEWLETT PACKARD	HP VECTRA VL400	C. Cecilia Espinosa
CPU	2283	1959794	MX10835147	HP	VECTRA VL 400	Martinez Espinoza Jose Giovanni
CPU	2385	2211120	HXILJ91	DELL	Dimension 5150	Figueroa Figueroa Jorge
CPU	2495	2196804		HP	ProLiant ML350 (TORK)	Barragan Paz Ma. Rosario
CPU	2554	2226299	33707826397	DELL	LATITUDE D620	Barragan Paz Ma. Rosario

Figura 5.68 Listado de equipos

Capítulo 5

Implementación

Al final de la lista el sistema presenta la opción de generar un reporte en Excel con la información que acaba de ser presentada.

IMPRESORA	159	1919637	USGT086744	HP	LASER JET 2100	Barragan Paz Ma. Rosario
SCANNER	88	555	555	prueba	prueba	Velazquez Sedan Nelida
NOBREAK	121	1891800	F02516080	Tripp-lite	E136916	Barragan Paz Ma. Rosario
NOBREAK	122	2227740	SUA1500	APC	SMART-UPS-1500	Sandoval Vazquez Ricardo
NOBREAK	123	2105767	AS0231330846	APC	Smart-UPS 1500	Barragan Paz Ma. Rosario
CAMARA_DIGITAL	2	23434234				Velazquez Sedan Nelida
ACCESSP	1	101010	101010	prueba	prueba	nuevo

[Regresar](#) [Reporte](#)

D.R. © 2009. Unidad de Servicios de Cómputo Académico.
Última actualización Agosto 2009 | Versión 2.1

Figura 5.69 Opción de exportar a Excel

SISTEMA DE INVENTARIOS DE EQUIPOS DE CÓMPUTO DE LA FACULTAD DE INGENIERÍA

Principal Altas Bajas Modificaciones Movimientos Visualizar Inventarios

Viernes 11 de Febrero 2011 Investigación y desarrollo Cerrar Sesión

Descargar Reporte

Regresar al inicio

Figura 5.70 Descargar el reporte.

Para generar directamente el reporte en Excel esta la opción de “reporte completo” (Figura 5.71):
En la sección de Anexos se muestra el formato del Reporte General.

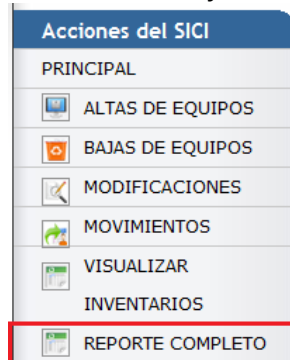


Figura 5.71 Descargar el reporte directamente del menú lateral.

Capítulo 5

Implementación

Personalizar Reporte

Permite especificar que datos de los bienes se necesitan generar en el reporte.

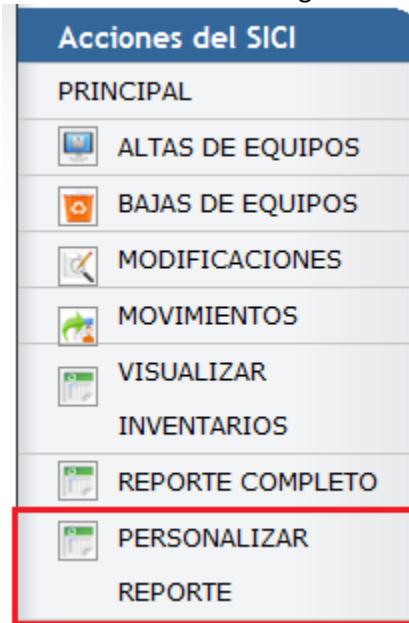


Figura 5.72 Menú de personalizar reporte.

La figura 5.73 muestra los filtros que permitirán al usuario, definir que campos deberán aparecer en el reporte.

PERSONALIZAR REPORTE						
<input type="checkbox"/> Nombre Division	<input type="checkbox"/> ID de equipo	<input type="checkbox"/> Nombre del Bien	<input type="checkbox"/> Número de Resguardo	<input type="checkbox"/> Inventario UNAM	<input type="checkbox"/> Movimiento	<input type="checkbox"/> Fecha de Movimiento
<input type="checkbox"/> Marca	<input type="checkbox"/> Número de Serie	<input type="checkbox"/> Modelo	<input type="checkbox"/> Tipo Procesador	<input type="checkbox"/> Velocidad	<input type="checkbox"/> Capacidad en Disco Duro	<input type="checkbox"/> Cantidad de Memoria
<input type="checkbox"/> Accesorios	<input type="checkbox"/> Garantía/Mantenimiento	<input type="checkbox"/> Ubicación	<input type="checkbox"/> Nombre del Responsable	<input type="checkbox"/> E-mail	<input type="checkbox"/> Teléfono	<input type="checkbox"/> Tipo de Equipo
<input type="checkbox"/> Sistema Operativo	<input type="checkbox"/> Alta / Baja	<input type="checkbox"/> Dirección IP	<input type="checkbox"/> Dirección Física MAC	<input type="checkbox"/> Seleccionar todos los campos	<input type="button" value="Generar Reporte"/>	

Figura 5.73 Campos para generar el reporte.

En la sección de Anexos se muestra el formato del Reporte General.

Capítulo 5

Implementación

Reporte del censo

Permite generar un reporte que contiene información resumida y contabilizada de acuerdo a los siguientes criterios:

- Por tipo de máquinas
- Por tipo de usuario
- Por tipo de sistema operativo
- Por tipo de conexión
- Por equipos de impresión
- Por equipos digitales
- Por equipos de telecomunicaciones

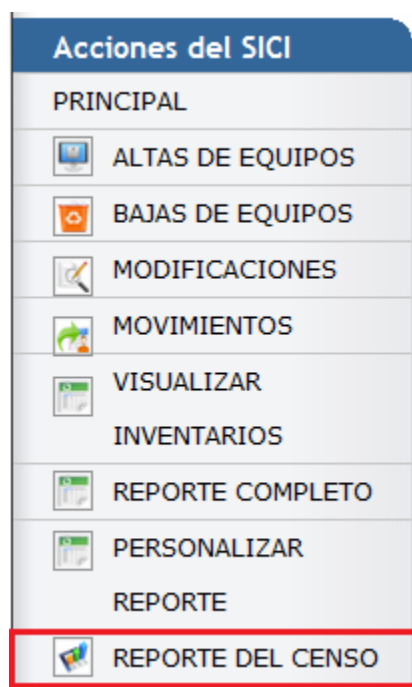


Figura 5.74 Menú para reportes del Censo.

En el Anexo2 se muestran los formatos para reportes del censo

Capítulo 5

Implementación

Obtener reportes como usuario administrador

Descripción: Permite visualizar y generar diversos tipos de reportes de los bienes de cualquier división.

Importante: *Únicamente el usuario administrador podrá visualizar este tipo de reportes.*

Estos reportes pueden ser completos o personalizados de los bienes que están dados de alta o que han sido dados de baja, así mismo permite crear reportes de tipo censo que entregan información resumida y contabilizada de los bienes de las divisiones

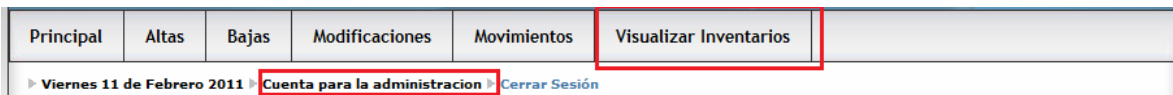


Figura 5.75 Usuario administrador.

1. El usuario elige la opción de Visualizar Inventarios (Figura 5.75).
2. El sistema despliega distintas agrupaciones de divisiones para facilitar la búsqueda al usuario (Figura 5.76).
3. El usuario elige la agrupación en la que le conviene buscar para generar el reporte que desea.
 - a. Reportes por cada división
 - b. Secretaría General
 - c. Unidad de Servicios de Cómputo Académico
 - d. Facultad de Ingeniería
 - e. Secretaría de Apoyo a la Docencia
 - f. Coordinación de Vinculación Productiva y Social

Capítulo 5

Implementación

<h4>REPORTES POR CADA DIVISIÓN</h4> <p>Obtiene la información de toda una división.</p> <p>Nombre de la división: Jefatura</p> <p>Generar Reporte</p> <ul style="list-style-type: none"><input type="radio"/> Reporte Bajas<input checked="" type="radio"/> Reporte Excel<input type="radio"/> Reporte Web<input type="radio"/> Reporte Censo	<h4>SECRETARIA GENERAL</h4> <p>Se obtiene la información de una división que pertenece a Secretaría General, o bien, de toda la Secretaría.</p> <p>Nombre de la división: ---TODA SECRETARIA GENERAL---</p> <p>Generar Reporte</p> <ul style="list-style-type: none"><input type="radio"/> Reporte Bajas<input checked="" type="radio"/> Reporte Excel<input type="radio"/> Reporte Web<input type="radio"/> Reporte Censo
<h4>UNIDAD DE SERVICIOS DE COMPUTO ACADEMICO</h4> <p>Se obtiene la información de un departamento, o bien, de toda la UNIDAD</p> <p>Nombre de la división: --TODOS LOS DEPARTAMENTOS--</p> <p>Generar Reporte</p> <ul style="list-style-type: none"><input checked="" type="radio"/> Reporte Bajas<input type="radio"/> Reporte Excel<input type="radio"/> Reporte Web<input type="radio"/> Reporte Censo	<h4>FACULTAD DE INGENIERIA</h4>  <p>INFORME DE TODA LA FACULTAD</p> <p>Generar Reporte</p> <ul style="list-style-type: none"><input checked="" type="radio"/> Reporte Altas<input type="radio"/> Reporte Bajas <p>EQUIPOS TOTALES EN CADA DEPARTAMENTO Se muestra el numero totales de equipos, en cada uno de los departamentos.</p>
<h4>SECRETARIA DE APOYO A LA DOCENCIA</h4> <p>Se obtiene la información de una división que pertenece a la Secretaría, o bien, de toda.</p> <p>Nombre de la división: ---SECRETARIA APOYO A LA DOCENCIA---</p> <p>Generar Reporte</p> <ul style="list-style-type: none"><input type="radio"/> Reporte Bajas<input checked="" type="radio"/> Reporte Excel<input type="radio"/> Reporte Web<input type="radio"/> Reporte Censo	<h4>COORD. DE VINCULACIÓN PRODUCTIVA Y SOCIAL</h4> <p>Se obtiene la información de una división que pertenece a Secretaría, o bien, de toda. (CUAL ES LA OFICINA D VINCULACION)</p> <p>Nombre de la división: ---VINCULACIÓN PRODUCTIVA Y SOCIAL---</p> <p>Generar Reporte</p> <ul style="list-style-type: none"><input type="radio"/> Reporte Bajas<input checked="" type="radio"/> Reporte Excel<input type="radio"/> Reporte Web<input type="radio"/> Reporte Censo

Figura 5.76 Opciones para generar reportes por divisiones.

Capítulo 5

Implementación

5.2 Manual de Instalación

Introducción

El sistema de inventarios es una aplicación Web que consiste en un conjunto de servlets, páginas jsp, ficheros html, clases Java de apoyo empaquetadas o no en ficheros jar y otro tipo de recursos tales como ficheros de imágenes, de sonidos, de texto, etc.

Al ser una aplicación web puede existir de dos modos:

- *Mediante un fichero de extensión war (Web Application Resource, a veces también se le suele llamar Web ARchive) que engloba a todo su contenido.*
Se crea mediante la herramienta jar incluido en el J2SE, del mismo modo que un fichero jar. Este empaquetamiento se produce en la etapa de producción, es decir, cuando la aplicación ha sido comprobada y depurada para su comercialización.
- *Mediante una estructura de directorios basada en la especificación definida por Sun para los Servlets.* Dentro de esta estructura deben ubicarse de forma adecuada los componentes de la aplicación.
Es el modo de trabajo habitual en la etapa de desarrollo de la aplicación, es decir, cuando se realizan pruebas y modificaciones constantes en sus componentes.

Modo de trabajo habitual en las aplicaciones web se divide en dos fases:

- *Etapa de desarrollo:* Se usa una estructura de directorios. Se prevén constantes cambios en los componentes de la aplicación (Figura 5.77).
- *Etapa de producción:* se usa un archivo war. Los objetivos de la aplicación se han logrado y no se prevén modificaciones en sus componentes. Todos los recursos de la aplicación se empaquetan en un war para que pueda ser desplegada en cualquier tipo de servidor J2EE compatible con la especificación.

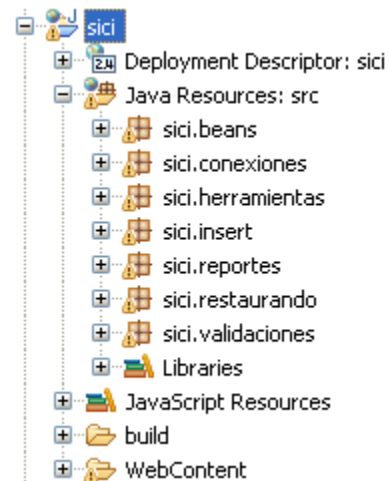


Figura 5.77 Estructura de directorios.

Capítulo 5

Implementación

Instalación

Requisitos:

- **Servidor de aplicaciones y Java Development Kit (JDK)**
- **Servidor de Base de datos**
- **Archivo de extensión war**

Pasos de instalación:

1.-Configurar los datos de la conexión:

Dentro de los archivos del sistema, hay que modificar los datos del archivo llamado Resources.properties, esto con la finalidad de configurar los parámetros de la base de datos (Figura 5.78):

Parámetros de la base de datos:

El nombre del conector:

```
sql.driver=org.postgresql.Driver
```

La dirección de la base de datos varía de acuerdo al manejador:

```
sql.url=jdbc:postgresql://127.0.0.1:5432/SIICI
```

Nombre del usuario de la base de datos:

```
sql.user=postgres
```

Contraseña de la base de datos:

```
sql.password=postgres
```

Una vez que los datos son configurados, se puede empaquetar el proyecto como .war para colocarlo en el servidor de aplicaciones.

2.-Generar paquete .war

- Utilizando el entorno de trabajo de eclipse Galileo seleccionamos el proyecto principal y en el menú contextual seleccionamos la opción: Export->WAR file (Figura 5.79)



Figura 5.79 Generar archivo war.

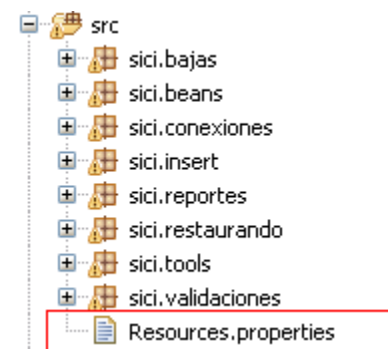


Figura 5.78 Archivo de configuración

Capítulo 5

Implementación

- Ingresar un nombre al paquete `sici.war` y el lugar del sistema en donde se desea guardar.

3.- Despliegue mediante war

Para ilustrar el proceso de despliegue de una aplicación web, se va a suponer que se han cumplido las dos etapas mencionadas anteriormente, se dispone de un fichero war de nombre `sici.war` que engloba todos sus componentes y que se le ha hecho llegar al administrador del servidor J2EE en el que se va a desplegar.

- El administrador asignará a la aplicación un context path o ruta de contexto coincidente con el nombre del fichero war, que indica el URL mediante el que cualquier cliente puede acceder a la misma. Cada aplicación web estará asociada a un contexto y todos sus componentes existirán en relación a ese contexto

En este caso, el contexto será `/sici`. El contexto debe empezar por `/`. Esta barra significa directorio raíz del servidor a nivel de aplicaciones web desplegadas en él.

Para el caso del sistema de inventarios se utilizó como servidor de aplicaciones Apache Tomcat, el cual cuenta la siguiente estructura de directorios:

La jerarquía de directorios de instalación de Tomcat incluye:

- `bin` - arranque, cierre, y otros scripts y ejecutables
- `common` - clases comunes que pueden utilizar Catalina y las aplicaciones web
- `conf` - ficheros XML y los correspondientes DTD para la configuración de Tomcat
- `logs` - logs de Catalina y de las aplicaciones
- `server` - clases utilizadas solamente por Catalina
- `shared` - clases compartidas por todas las aplicaciones web
- `webapps` - directorio que contiene las aplicaciones web
- `work` - almacenamiento temporal de ficheros y directorios

Importante: El fichero con extensión `.war` deberá ser colocado dentro del directorio `webapps` del servidor de aplicaciones.

Si fuera necesario, se requiere que el servidor de aplicaciones sea reiniciado.

Capítulo 5

Implementación

Una vez que es colocada la aplicación dentro del servidor, se puede ingresar mediante la ruta establecida, por ejemplo:



Figura 5.80 Dirección web del sistema

Si todo es correcto se mostrara la pantalla de inicio del Sistema de Inventarios de la Facultad de Ingeniería. (Figura 5.81.)

A screenshot of the SICI web application interface. The page has a blue header with the text "SISTEMA DE INVENTARIOS DE EQUIPOS DE CÓMPUTO DE LA FACULTAD DE INGENIERÍA" and logos for "INGENIERÍA" and "Unica". Below the header, there is a date "Sabado 5 de Febrero 2011". The main content area is divided into three sections: a left sidebar titled "Equipos de Cómputo" with a list of equipment types (Access Point, Camaras Digitales, Concentradores, Modem de Fibra Óptica, Plotters, Router, Monitores, Mouse, Teclados, Unidades de CD, Impresoras, Escaners, Discos Duros, Reguladores, Plotters, Hub, Switchs, No-Breaks, Proyectoros, CPU, Portátiles, Mac, PC, WS); a central section titled "Sistema de Inventarios" with a sub-section "SICI" containing a paragraph of text about the system's history and functionality; and a right sidebar titled "Login" with input fields for "Usuario:" and "Password:" and an "Ingresar" button. At the bottom of the page, there is a footer with the text "D.R. © 2009. Unidad de Servicios de Cómputo Académico. Última actualización Agosto 2009 | Versión 2.1".

Figura 5.81 Página de inicio del sistema.

CONCLUSIONES

A lo largo de esta tesis hemos presentado la documentación del proceso de análisis y desarrollo de un proyecto que surge de la necesidad de una completa renovación de un sistema llamado SICI, toda la información contenida en dicho documento permite, entre otras cosas, apreciar claramente el cumplimiento amplio de los objetivos planteados en el capítulo 1.

El objetivo principal era proporcionar un mecanismo que permitiera automatizar, facilitar y mantener actualizado el inventario de equipo de cómputo de la Facultad de Ingeniería, como resultado se desprendió la segunda versión de el Sistema de Control de Inventarios (SICI) denominada ahora “SISTEMA DE CONTROL DE INVENTARIOS Y CENSO DE EQUIPOS DE CÓMPUTO DE LA FACULTAD DE INGENIERÍA” (SICICE) el cual se presenta con diversas mejoras en cuanto a estructura e interfaz gráfica pero sobre todo se presenta con una funcionalidad completamente nueva que es la generación de reportes generales y personalizados en hojas de cálculo, los cuales permiten el fácil intercambio y análisis de los datos de los bienes que se tienen inventariados y representan un apoyo que resulta primordial para realización del Censo de dichos bienes.

El mismo documento que ciñe esta tesis fue un instrumento para el cumplimiento de algunos objetivos específicos como son la elaboración de manuales de instalación, usuario, mismos que permiten dar soporte y mantenimiento al sistema tal como se especifica en los requerimientos y objetivos del proyecto.

Más allá del alcance satisfactorio de los objetivos planteados para este proyecto cabe destacar tres aspectos fundamentales sobre los que descansa gran parte de este proyecto:

- 1) Tal como se explicó en la introducción de este trabajo los avances en las tecnologías de la información han permitido ampliar las posibilidades para aplicar técnicas para el control y optimización de inventarios, sin embargo no basta realizar y/o implementar cualquier sistema para que se logren cumplir las expectativas deseadas, se debe tener perfectamente claro dichas expectativas y realizar un análisis detallado para ofrecer la mejor solución, en ocasiones resulta también importante tener visión y tomar en cuenta la constante evolución en las necesidades del usuario que cada día exige más y más de las tecnologías de información; si bien un análisis se deben definir objetivos y alcances también debe diseñarse para ser flexible, lo cual nos lleva al segundo punto.
- 2) En la mayoría de los desarrollos de software saber elegir el modelo de proceso correcto se convierte en la piedra angular ya que de ello dependerá el éxito o fracaso de un proyecto. Es indispensable, a la hora de elegir, entender que debido a la diversidad de modelos, no existe uno solo que sea ideal y es por ello que en ocasiones se podrá y tendrá que combinar y adaptar diversos modelos para alcanzar los objetivos planteados. Por otra parte,

Conclusiones

independientemente del modelo o modelos elegidos, siempre es recomendable apegarse a uno o varios estándares que nos permitan asegurar la calidad del producto.

- 3) Por último, resulta conveniente destacar la importancia del software libre no sólo por las herramientas que se emplearon para desarrollar el sistema, sino porque en sí mismo el sistema nació con la filosofía de software libre, creemos enérgicamente tal como lo indica la Free Software Foundation que la tecnología debe trabajar en beneficio de las comunidades y juzgamos no sólo recomendable sino ineludible que un proyecto de esta naturaleza debe estar basado en una ideología que permita garantizar el acceso y beneficio para la comunidad universitaria sobre todo en la Universidad Nacional Autónoma de México donde se brinda una formación altamente humanista y socialmente responsable.

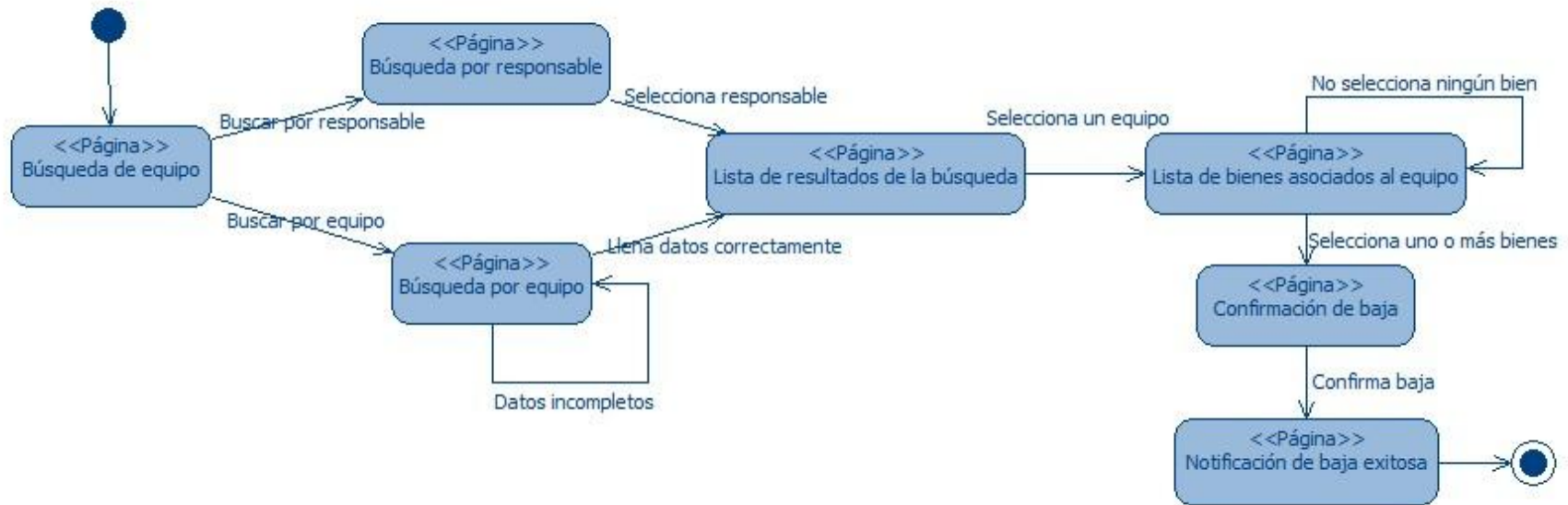
ANEXO 1

Flujos de pantallas

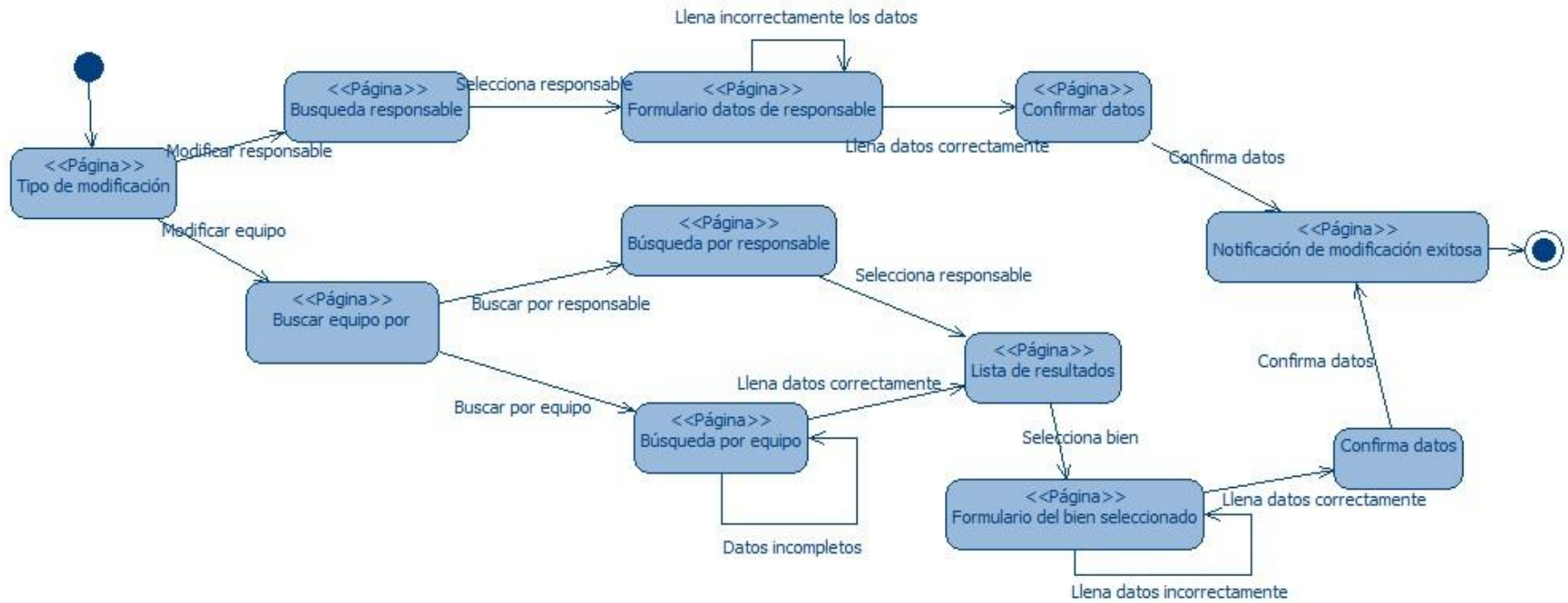
- Submáquina de estados: Página de altas



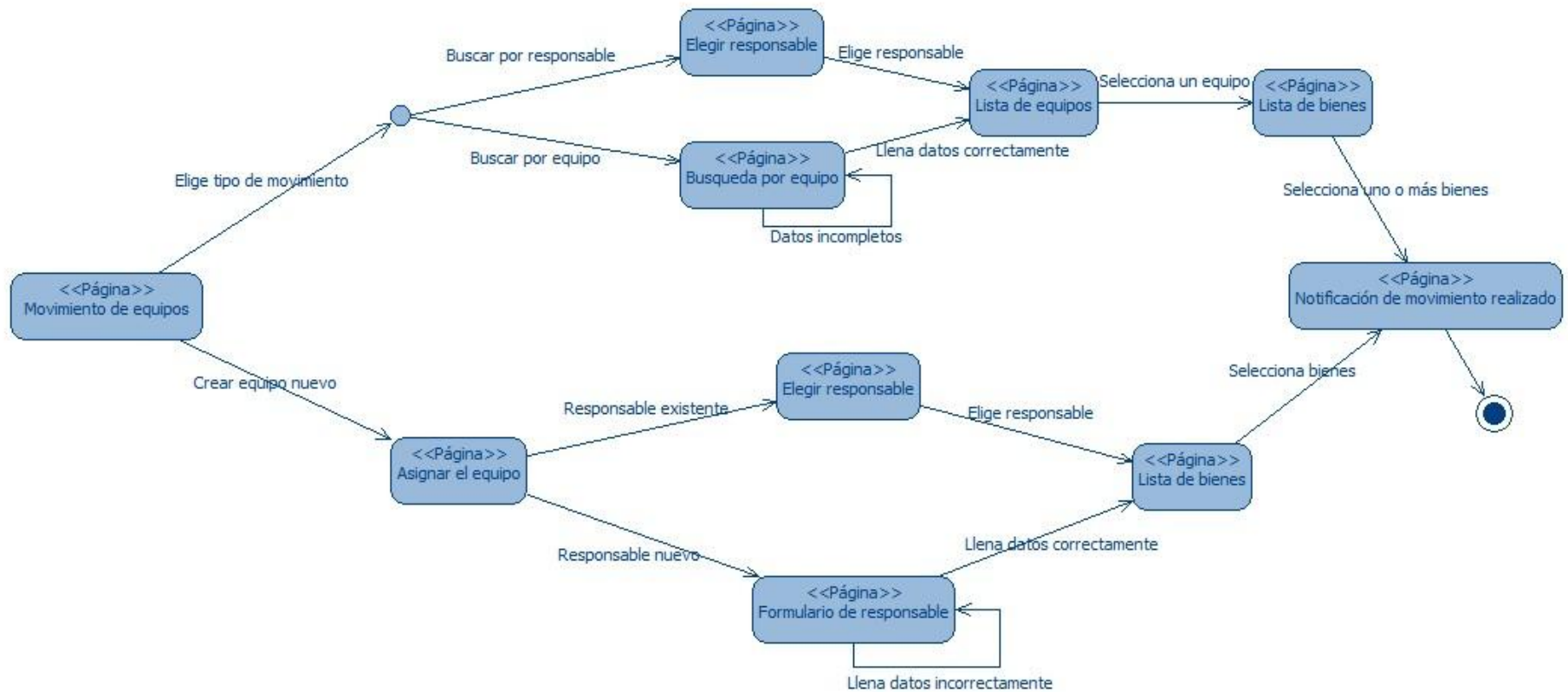
- Submáquina de estados: Página de bajas



- Submáquina de estados: Página de modificaciones



- Submáquina de estados: Página de movimientos



- Submáquina de estados: Página de reportes para usuario División



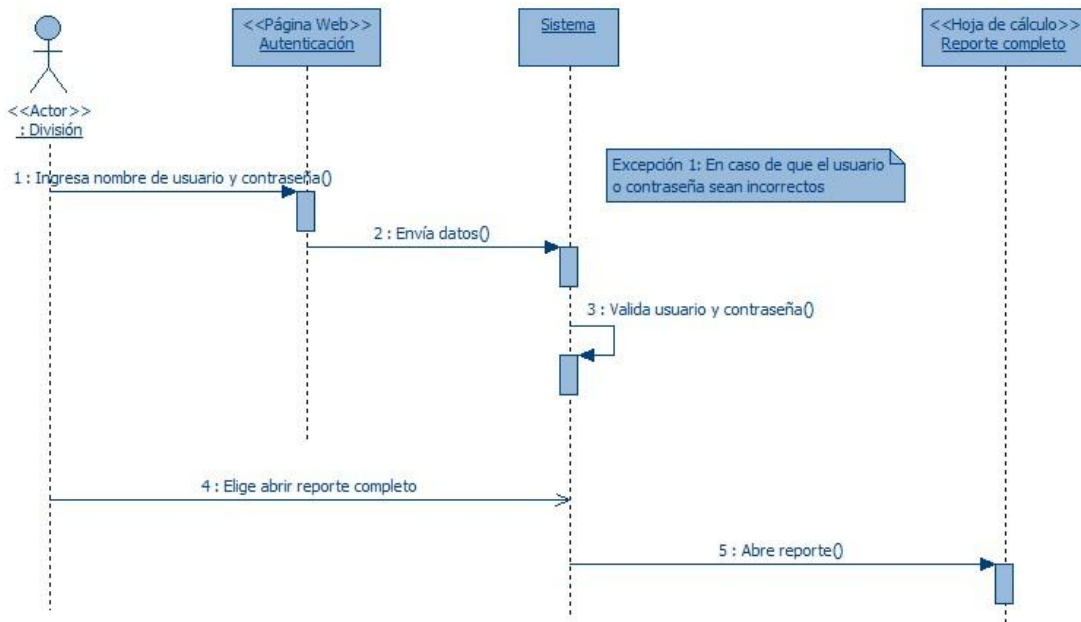
- Submáquina de estados: Página de reportes para usuario Administrador



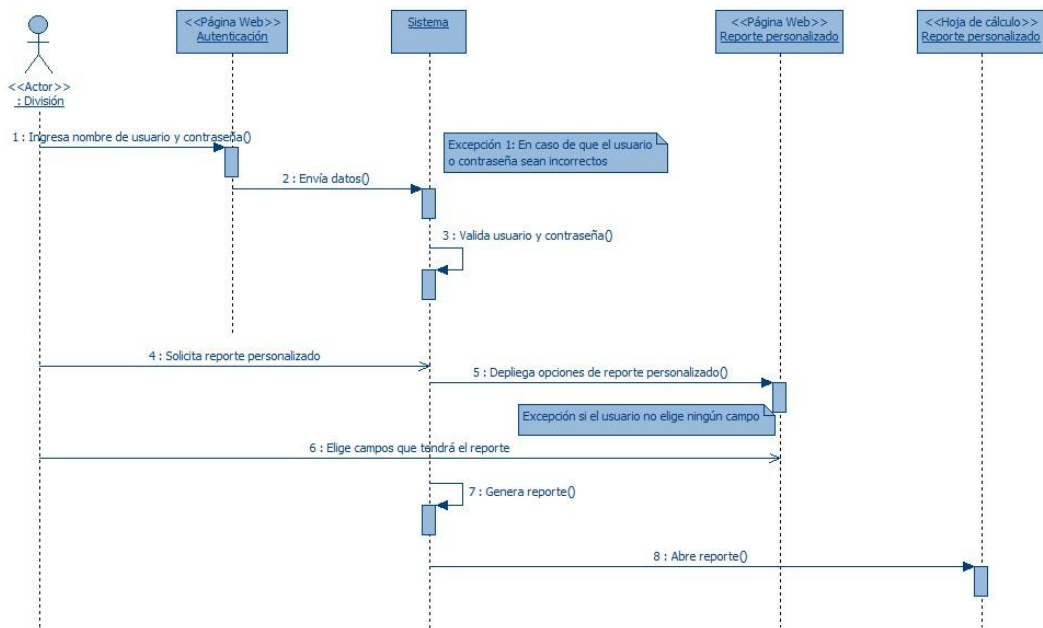
ANEXO 2

Diagramas de secuencia

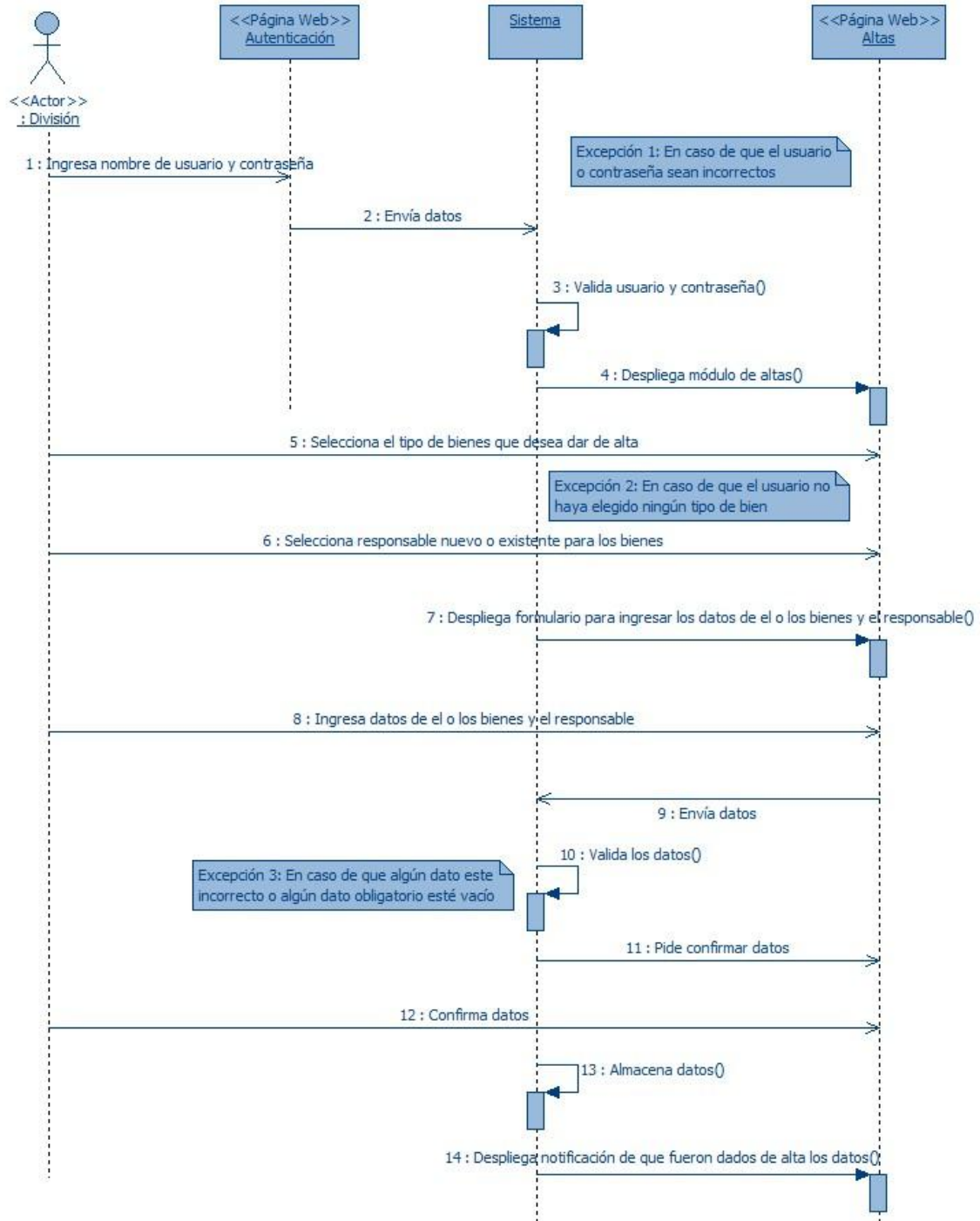
Reporte completo para usuario División



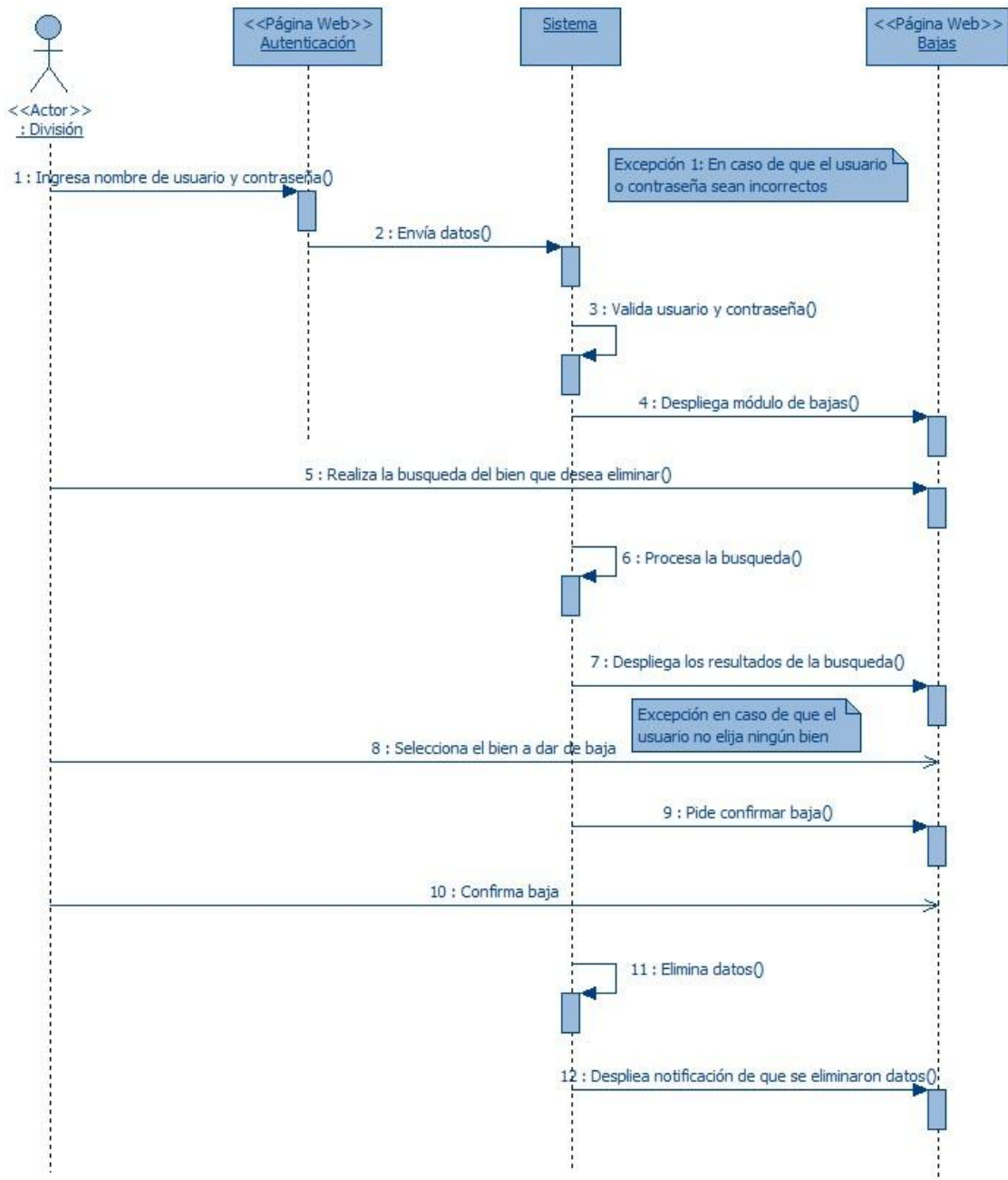
Reporte personalizado para usuario División



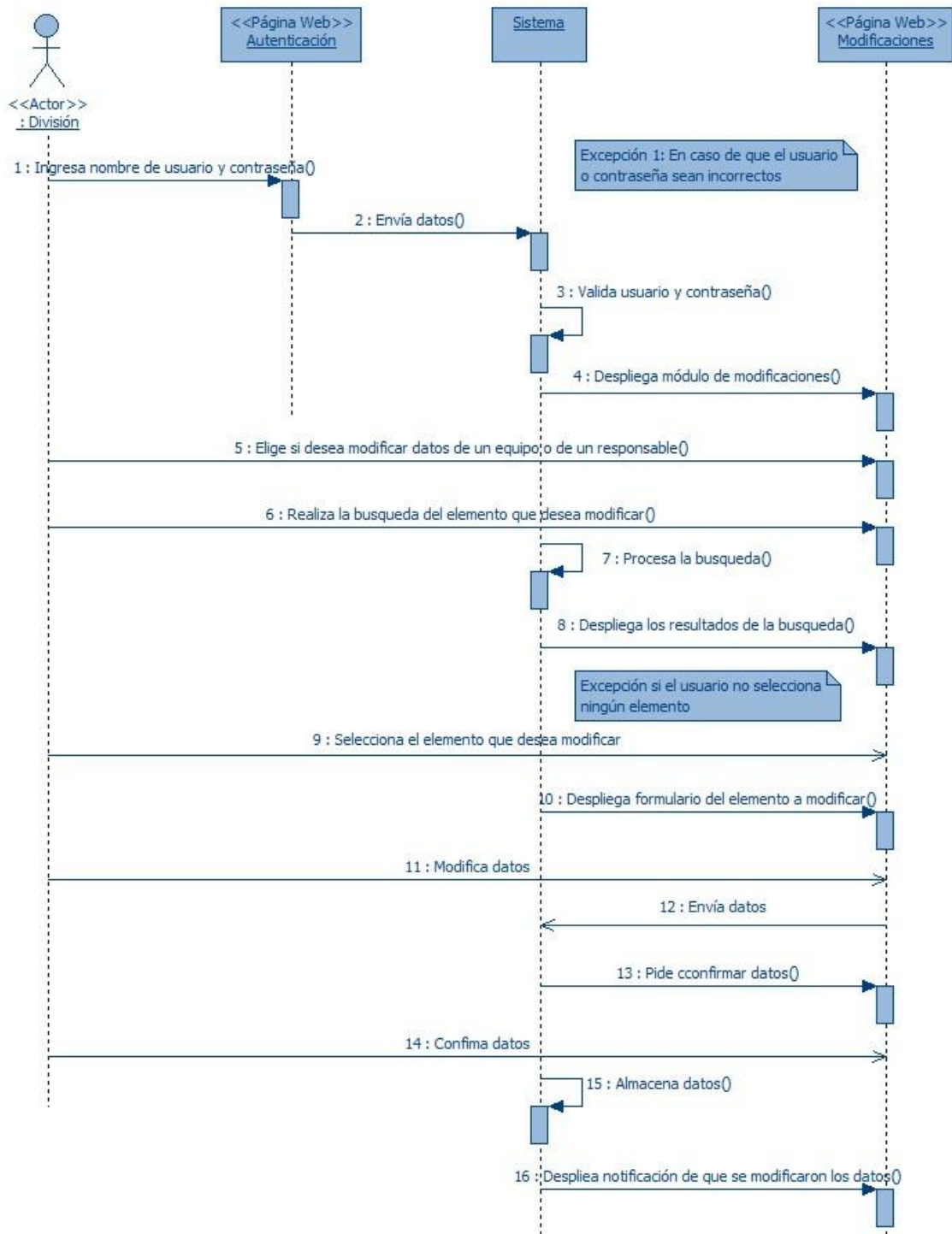
Altas



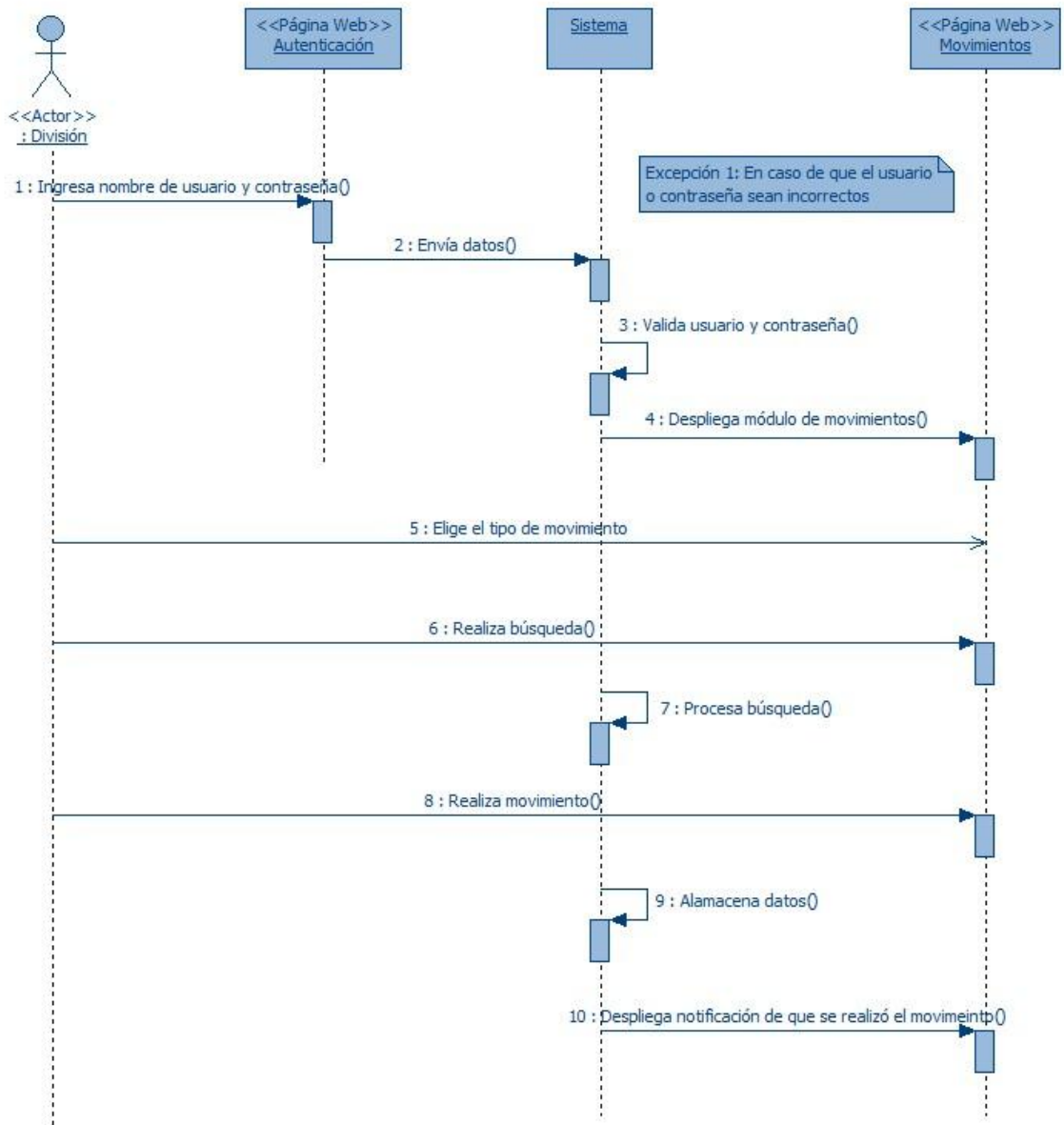
Bajas



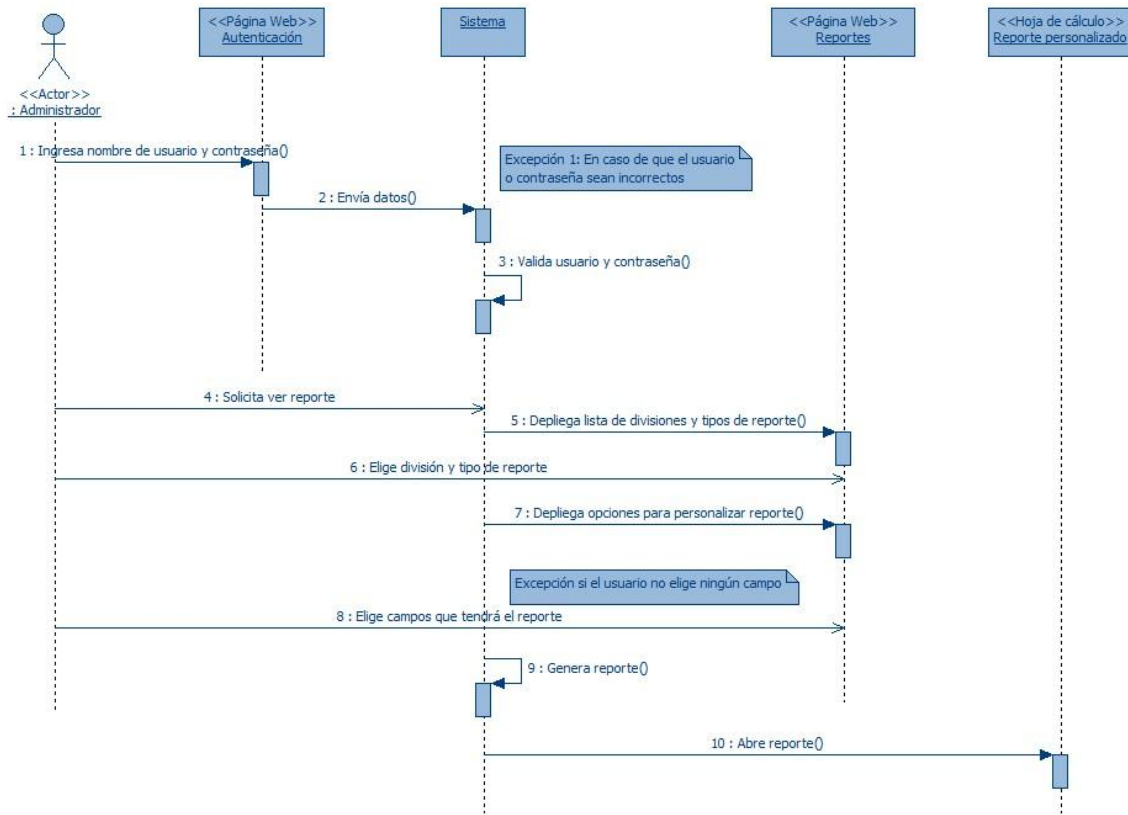
Modificaciones



Movimientos



Reportes para usuario Administrador



ANEXO 3

Formatos para reportes de bienes por División

FACULTAD DE INGENIERIA
SECRETARIA GENERAL
UNIDAD DE SERVICIOS DE COMPUTO ACADEMICO

REPORTE GENERAL

Fecha de Actualizacion: 2010-09-18

Departamento de Seguridad				
DIVISION	ID EQUIPO	NOMBRE DEL BIEN	MARCA	NO. SERIE
Departamento de Seguridad	85	CPU	DELL	3FQG621
Departamento de Seguridad	118	CPU	DELL	F4QTL051
Departamento de Seguridad	118	MONITOR	DELL	MYOX37824760344JB58
Departamento de Seguridad	118	TECLADO	DELL	TH07N1243717143802QX
Departamento de Seguridad	118	MOUSE	DELL	HCD40841979
Departamento de Seguridad	119	CPU	DELL	5RTL051
Departamento de Seguridad	119	MONITOR	DELL	MYOX37I8244760344JB54
Departamento de Seguridad	119	TECLADO	DELL	TH07N124371714380206
Departamento de Seguridad	119	MOUSE	DELL	HCD40818344
Departamento de Seguridad	120	CPU	DELL	4STL051
Departamento de Seguridad	120	MONITOR	DELL	MYOX37824760344J
Departamento de Seguridad	120	TECLADO	DELL	TH07N124371714390174
Departamento de Seguridad	120	MOUSE	DELL	HCD40818334
Departamento de Seguridad	124	CPU	ARMADA	
Departamento de Seguridad	124	MONITOR	SANSUNG	HCEHCO1151N
Departamento de Seguridad	124	TECLADO		9A083631

RESUMEN DE RESULTADOS	
DESCRIPCION	NUMERO
CPU	5
MONITOR	4
TECLADO	4
MOUSE	3
IMPRESORA	null
SCANNER	null
PLOTTTER	null
DD EXTERNO	null
CDWR EXTERNO	null
HUB	null
SWITCH	null
NO-BREAK	null
REGULADOR	null
ACCESS POINT	null
CAMARA DIGITAL	null
CONCENTRADOR	null
MODEL FIBRA OPTICA	null
PROYECTOR	null
ROUTER	null
SCANNER DE GRAN VOLUMEN	null

ANEXO 4

Formatos para reportes de Censo

DIRECCION GENERAL DE SERVICIOS DE COMPUTO ACADEMICO
CONSEJO ASESOR DE COMPUTO
CENSO DE COMPUTO

2010-09-18

DEPENDENCIA: Divisiones con Secretaria General

Cuadro 1:

EQUIPO DE COMPUTO -POR ESTADO-			
TIPO DE EQUIPO	ESTADO ACTUAL DEL EQUIPO		
	EN USO	DESUSO	TOTAL
COMPUTADORAS PERSONALES			
CELERON D Y EQUIVALENTES	0	0	0
CELERON.PENTIUM II,PENTIUM III y ANTERIORES	0	0	0
PENTIUM D, CORE DUO,CORE 2 DUO Y EQUIVALENTES	0	0	0
PENTIUM IV y EQUIVALENTES	0	0	0
MACINTOSH			
CORE DUO, CORE 2 DUO	0	0	0
POWERPC G3,G4,G5	0	0	0
PORTATILES			
CELERON M, PENTIUM Y EQUIVALENTES	0	0	0
CORE DUO MOVIL, CORE 2 DUO MOVIL Y EQUIVALENTES	0	0	0
CORE SOLO MOVIL Y EQUIVALENTES	0	0	0
OTROS (PDAs)	0	0	0
ALTO RENDIMIENTO			
EQUIPO DE ALTO RENDIMIENTO	0	0	0
TOTAL	0	0	0

EQUIPO DE COMPUTO -POR USUARIO FINAL-									
TIPO EQUIPO	TOTAL	ALUMNOS	PERSONAL ACADEMICO	PERSONAL ADMVO.	INVESTIGADORES	BIBLIOTECAS		SERVIDORES	
						USUARIOS	ADMVO.	GENERALES	WEB
COMPUTADORAS PERSONALES									
CELERON D Y EQUIVALENTES	0	0	0	0	0	0	0	0	0
CELERON.PENTIUM II,PENTIUM III y ANTERIORES	0	0	0	0	0	0	0	0	0
PENTIUM D, CORE DUO,CORE 2 DUO Y EQUIVALENTES	0	0	0	0	0	0	0	0	0
PENTIUM IV y EQUIVALENTES	0	0	0	0	0	0	0	0	0
MACINTOSH									
CORE DUO, CORE 2 DUO	0	0	0	0	0	0	0	0	0
POWERPC G3,G4,G5	0	0	0	0	0	0	0	0	0
PORTATILES									
CELERON M, PENTIUM Y EQUIVALENTES	0	0	0	0	0	0	0	0	0
CORE DUO MOVIL, CORE 2 DUO MOVIL Y EQUIVALENTES	0	0	0	0	0	0	0	0	0
CORE SOLO MOVIL Y EQUIVALENTES	0	0	0	0	0	0	0	0	0
OTROS (PDAs)	0	0	0	0	0	0	0	0	0
ALTO RENDIMIENTO									
EQUIPO DE ALTO RENDIMIENTO	0	0	0	0	0	0	0	0	0
TOTAL	0	0	0	0	0	0	0	0	0

EQUIPO DE COMPUTO -POR SISTEMA OPERATIVO-							
TIPO EQUIPO	TOTAL	WIN ME Y ANTERIORES	WIN 2000/XP	WIN SERVER	MACOS	UNIX/LINUX	OTROS
				STA			
COMPUTADORAS PERSONALES							
CELERON D Y EQUIVALENTES	0	0	0	0	0	0	0
CELERON.PENTIUM II,PENTIUM III y ANTERIORES	0	0	0	0	0	0	0
PENTIUM D, CORE DUO,CORE 2 DUO Y EQUIVALENTES	0	0	0	0	0	0	0
PENTIUM IV y EQUIVALENTES	0	0	0	0	0	0	0
MACINTOSH							
CORE DUO, CORE 2 DUO	0	0	0	0	0	0	0
POWERPC G3,G4,G5	0	0	0	0	0	0	0
PORTATILES							
CELERON M, PENTIUM Y EQUIVALENTES	0	0	0	0	0	0	0
CORE DUO MOVIL, CORE 2 DUO MOVIL Y EQUIVALENTES	0	0	0	0	0	0	0
CORE SOLO MOVIL Y EQUIVALENTES	0	0	0	0	0	0	0
OTROS (PDAs)	0	0	0	0	0	0	0
ALTO RENDIMIENTO							
EQUIPO DE ALTO RENDIMIENTO	0	0	0	0	0	0	0
TOTAL	0	0	0	0	0	0	0

EQUIPO DE COMPUTO -CONEXION A RED UNAM-				
TIPO EQUIPO	TOTAL	CONEXION DIRECTA		VIA MODEM
		CABLEADO	INALAMBRICA	
COMPUTADORAS PERSONALES				
CELERON D Y EQUIVALENTES	0	0	0	0
CELERON.PENTIUM II,PENTIUM III y ANTERIORES	0	0	0	0
PENTIUM D, CORE DUO,CORE 2 DUO Y EQUIVALENTES	0	0	0	0
PENTIUM IV y EQUIVALENTES	0	0	0	0
MACINTOSH				
CORE DUO, CORE 2 DUO	0	0	0	0
POWERPC G3,G4,G5	0	0	0	0
PORTATILES				
CELERON M, PENTIUM Y EQUIVALENTES	0	0	0	0
CORE DUO MOVIL, CORE 2 DUO MOVIL Y EQUIVALENTES	0	0	0	0
CORE SOLO MOVIL Y EQUIVALENTES	0	0	0	0
OTROS (PDAs)	0	0	0	0
ALTO RENDIMIENTO				
EQUIPO DE ALTO RENDIMIENTO	0	0	0	0
TOTAL	0	0	0	0

EQUIPO DE IMPRESION -POR ESTADO ACTUAL-			
TIPO DE EQUIPO	ESTADO ACTUAL DEL EQUIPO		
	EN USO	DESUSO	TOTAL
INYECCION DE TINTA	0	0	0
LASER ALTO VOLUMEN B/N	0	0	0
LASER ALTO VOLUMEN COLOR	0	0	0
LASER PERSONAL B/N	0	0	0
LASER PERSONAL COLOR	0	0	0
MATRIZ	0	0	0
OTROS	0	0	0
TOTAL	0	0	0
OTROS EQUIPOS DIGITALES -POR ESTADO ACTUAL-			
TIPO DE EQUIPO	ESTADO ACTUAL DEL EQUIPO		
	EN USO	DESUSO	TOTAL
CAMARA DIGITAL	0	0	0
PROYECTORES	0	0	0
SCANNER CAMA PLANA PARA OFICINA	0	0	0
SCANNER GRAN VOLUMEN	0	0	0
TOTAL	0	0	0
EQUIPOS DE TELECOMUNICACIONES -POR ESTADO ACTUAL-			
TIPO DE EQUIPO	ESTADO ACTUAL DEL EQUIPO		
	EN USO	DESUSO	TOTAL
ACCES POINT	0	0	0
CONCENTRADORES	0	0	0
MODEMS DE FIBRA OPTICA	0	0	0
RUTEADORES	0	0	0
SWITCHES	0	0	0
TOTAL	0	0	0

ANEXO 5

Estándar de herramientas tecnológicas para desarrollo de sistemas

- Estándar de herramientas tecnológicas para desarrollo de sistemas



Facultad de Ingeniería
Secretaría General
Unidad de Servicios de Cómputo Académico

ESTÁNDAR DE HERRAMIENTAS TECNOLÓGICAS PARA DESARROLLO DE SISTEMAS

1. INTRODUCCIÓN

Este documento tiene como objetivo describir la propuesta y justificación del estándar de herramientas tecnológicas para el desarrollo de software con que cuenta la Unidad de Servicios de Cómputo Académico de la Facultad de Ingeniería (UNICA).

2. PROPUESTA

Con esto se intenta instaurar un estándar de herramientas tecnológicas para la Facultad de Ingeniería dada la amplia experiencia y recursos humanos con que cuenta la Facultad en las siguientes herramientas.

2.2. Lenguaje de modelado unificado (UML)

2.3. Manejador de base de datos POSTGRESQL

2.4. Lenguaje JAVA

2.5. Sistema operativo LINUX

3. JUSTIFICACION

3.2. LENGUAJE DE MODELADO UNIFICADO (UML)

UML es un lenguaje para hacer modelos y es independiente de los métodos de análisis y diseño. Existen diferencias importantes entre un método y un lenguaje de modelado. Un *método* es una manera explícita de estructurar el pensamiento y las acciones de cada individuo. Además, el método le dice al usuario qué hacer, cómo hacerlo, cuándo hacerlo y por qué hacerlo; mientras que el lenguaje de modelado carece de estas instrucciones. Los métodos contienen modelos y esos modelos son utilizados para describir algo y comunicar los resultados del uso del método.

Un modelo es expresado en un *lenguaje de modelado*. Un lenguaje de modelado consiste de vistas, diagramas, elementos de modelo y un conjunto de mecanismos generales o reglas que indican cómo utilizar los elementos. Las reglas son sintácticas, semánticas y pragmáticas

Con UML se fusiona la notación de estas técnicas para formar una herramienta compartida entre todos los ingenieros software que trabajan en el desarrollo orientado a objetos.

- 3.2.1.** UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware y organizaciones del mundo real. UML ofrece nueve diagramas en los cuales modelar sistemas.

ESTÁNDAR DE HERRAMIENTAS TECNOLÓGICAS PARA DESARROLLO DE SISTEMAS

- 3.2.1.1.** *Diagramas de comportamiento:* Especifican las partes dinámicas de un sistema tales como estados del sistema, flujo de control de actividades, secuencia de mensajes y más.
- 3.2.1.2.** *Diagramas de clases:* Conjunto de clases, interfaces y colaboraciones, y las relaciones entre ellas.
- 3.2.1.3.** *Diagramas de objetos:* Instantáneas de las instancias de los elementos encontrados en los diagramas de clases.
- 3.2.1.4.** *Diagramas de componentes:* Conjunto de componentes y sus relaciones.
- 3.2.1.5.** *Diagramas de despliegue:* Conjunto de nodos y sus relaciones.
- 3.2.1.6.** *Diagramas de casos de uso:* Conjunto de casos de uso y actores y sus relaciones. Son importantes para organizar y modelar el sistema.
- 3.2.1.7.** Diagramas de interacción:
- 3.2.1.8.** *Diagramas de secuencia:* Conjunto de objetos y los mensajes enviados y recibidos por ellos. Resalta ordenación temporal de los mensajes.
- 3.2.1.9.** *Diagramas de colaboración:* Resalta organización estructural de objetos que envían y reciben mensajes.
- 3.2.1.10.** *Diagramas de estados:* Representan máquinas de estados, construida por estados, transiciones, eventos y actividades. Útiles para modelar sistemas reactivos.
- 3.2.1.11.** *Diagramas de actividades:* Muestran el flujo de actividades de un sistema. Importantes para modelar la función de un sistema, así como para resaltar el flujo de control entre objetos.
- 3.2.2.** Cada diagrama usa la anotación pertinente y la suma de estos diagramas crean las diferentes vistas. Las vistas existentes en UML son:
- 3.2.2.1.** *Vista casos de uso:* Muestra el comportamiento del sistema tal y como es percibido por usuarios, analistas y encargados de pruebas. Se forma con los diagramas de casos de uso, colaboración, estados y actividades.
- 3.2.2.2.** *Vista de diseño:* Comprende el vocabulario del problema y su solución, y soporta los requisitos funcionales del sistema (servicios que el sistema debería proporcionar a los usuarios finales). Se forma con los diagramas de clases, objetos, colaboración, estados y actividades.
- 3.2.2.3.** *Vista de procesos:* Hilos y procesos que forman mecanismos de sincronización y concurrencia del sistema. Se forma con los diagramas de la vista de diseño; recalcando las clases y objetos referentes a procesos.
- 3.2.2.4.** *Vista de implementación:* Componentes y archivos que se utilizan para ensamblar y hacer disponible el sistema físico. Se forma con los diagramas de componentes, colaboración, estados y actividades.

ESTÁNDAR DE HERRAMIENTAS TECNOLÓGICAS PARA DESARROLLO DE SISTEMAS

3.2.2.5. *Vista de despliegue:* Nodos que forman la topología hardware sobre la que se ejecuta el sistema. Distribución, entrega e instalación de las partes. Se forma con los diagramas de despliegue, interacción, estados y actividades.

3.2.3. UML también intenta solucionar el problema de propiedad de código que se da con los desarrolladores, al implementar un lenguaje de modelado común para todos los desarrollos se crea una documentación también común, que cualquier desarrollador con conocimientos de UML será capaz de entender, independientemente del lenguaje utilizado para el desarrollo.

3.2.4. UML permite la modificación de todos sus miembros mediante estereotipos y restricciones. Un estereotipo nos permite indicar especificaciones del lenguaje al que se refiere el diagrama de UML. Una restricción identifica un comportamiento forzado de una clase o relación, es decir mediante la restricción estamos forzando el comportamiento que debe tener el objeto al que se le aplica.

3.2.5. UML es el primer método en publicar un meta-modelo en su propia notación, incluyendo la notación para la mayoría de la información de requisitos, análisis y diseño. Se trata pues de un meta-modelo auto-referencial (cualquier lenguaje de modelado de propósito general debería ser capaz de modelarse a sí mismo).

3.2.6. Los principales beneficios de UML son:

3.2.6.1. Mejores tiempos totales de desarrollo (de 50 % o más).

3.2.6.2. Modelar sistemas (y no sólo de software) utilizando conceptos orientados a objetos.

3.2.6.3. Establecer conceptos y artefactos ejecutables.

3.2.6.4. Encaminar el desarrollo del escalamiento en sistemas complejos de misión crítica.

3.2.6.5. Crear un lenguaje de modelado utilizado tanto por humanos como por máquinas.

3.2.6.6. Mejor soporte a la planeación y al control de proyectos.

3.2.6.7. Alta reutilización y minimización de costos

3.2.7. Fases del desarrollo de un sistema

Las fases del desarrollo de sistemas que soporta UML son: *Análisis de requerimientos, Análisis, Diseño, Programación y Pruebas.*

3.2.7.1. Análisis de Requerimientos

UML tiene casos de uso (use-cases) para capturar los requerimientos del cliente. A través del modelado de casos de uso, los actores externos que tienen interés en el sistema son modelados con la funcionalidad que ellos requieren del sistema (los casos de uso). Los actores y los casos de uso son modelados con relaciones y

ESTÁNDAR DE HERRAMIENTAS TECNOLÓGICAS PARA DESARROLLO DE SISTEMAS

tienen asociaciones entre ellos o éstas son divididas en jerarquías. Los actores y casos de uso son descritos en un diagrama use-case. Cada use-case es descrito en texto y especifica los requerimientos del cliente: lo que él (o ella) espera del sistema sin considerar la funcionalidad que se implementará. Un análisis de requerimientos puede ser realizado también para procesos de negocios, no solamente para sistemas de software.

3.2.7.2. Análisis

La fase de análisis abarca las abstracciones primarias (clases y objetos) y mecanismos que están presentes en el dominio del problema. Las clases que se modelan son identificadas, con sus relaciones y descritas en un diagrama de clases. Las colaboraciones entre las clases para ejecutar los casos de uso también se consideran en esta fase a través de los modelos dinámicos en UML. Es importante notar que sólo se consideran clases que están en el dominio del problema (conceptos del mundo real) y todavía no se consideran clases que definen detalles y soluciones en el sistema de software, tales como clases para interfaces de usuario, bases de datos, comunicaciones, concurrencia, etc.

3.2.7.3. Diseño

En la fase de diseño, el resultado del análisis es expandido a una solución técnica. Se agregan nuevas clases que proveen de la infraestructura técnica: interfaces de usuario, manejo de bases de datos para almacenar objetos en una base de datos, comunicaciones con otros sistemas, etc. Las clases de dominio del problema del análisis son agregadas en esta fase. El diseño resulta en especificaciones detalladas para la fase de programación.

3.2.7.4. Programación

En esta fase las clases del diseño son convertidas a código en un lenguaje de programación orientado a objetos. Cuando se crean los modelos de análisis y diseño en UML, lo más aconsejable es trasladar mentalmente esos modelos a código.

3.2.7.5. Pruebas

Normalmente, un sistema es tratado en pruebas de unidades, pruebas de integración, pruebas de sistema, pruebas de aceptación, etc. Las pruebas de unidades se realizan a clases individuales o a un grupo de clases y son típicamente ejecutadas por el programador. Las pruebas de integración integran componentes y clases en orden para verificar que se ejecutan como se especificó. Las pruebas de sistema ven al sistema como una "caja negra" y validan que el sistema tenga la funcionalidad final que le usuario final espera. Las pruebas de aceptación conducidas por el cliente verifican que el sistema satisface los requerimientos y son similares a las pruebas de sistema.

ESTÁNDAR DE HERRAMIENTAS TECNOLÓGICAS PARA DESARROLLO DE SISTEMAS

3.3. LENGUAJE JAVA¹

Características del lenguaje java

Los siguientes puntos muestran las principales características del lenguaje de programación Java.

3.3.1. Simple. Java ofrece toda la funcionalidad de un lenguaje de programación simplificando algunas tareas que en otros lenguajes requieren un esfuerzo adicional por parte del programador. Un ejemplo de lo anterior es el manejo de la memoria. La máquina virtual de Java (JVM) se encarga de su administración haciendo uso de un hilo de ejecución de baja prioridad encargado de detectar y liberar bloques de memoria libres de referencias (Garbage Collector). Al eliminar características como la aritmética de punteros, los registros struct, operaciones de reserva de memoria (malloc() en C), liberación de memoria (como free() en C), existencia de referencias en lugar de punteros, etc., reduce considerablemente los errores de programación generados al delegar la administración de la memoria al programador.

3.3.2. Orientado a objetos. Java es un lenguaje totalmente orientado a objetos: encapsulación, herencia, polimorfismo, etc. Todos los programas en Java son clases, y su representación en memoria son las instancias de una clase. Java incorpora la sobrecarga y sobre escritura de métodos. No soporta herencia múltiple, sin embargo, incorpora la implementación de interfaces.

3.3.3. Distribuido. Java cuenta con capacidades de interconexión TCP/IP. Existen librerías para interactuar con protocolos como *http* y *ftp*. Esto permite a los programadores acceder a la información a través de la red con facilidad. Java proporciona las herramientas y librerías necesarias para la ejecución de sistemas distribuidos mediante Java RMI, JNDI, RMI-IIOP, JNI.

3.3.4. Robusto. Java proporciona diversos mecanismos con la finalidad de minimizar los errores de una aplicación en tiempo de compilación y en tiempo de ejecución algunos de ellos:

3.3.4.1. Tiempo de compilación.

- Comprobación de tipos de datos
- Requiere declaración explícita de métodos

3.3.4.2. Tiempo de Ejecución

- Verificación de la integridad del archivo de bytcodes
- Manejo de memoria

¹ JAVA (nombre de un tipo de café, originario del este de Asia, de la isla del mismo nombre), aunque hay algunos que afirman que el nombre deriva de las siglas de James Gosling, Arthur Van Hoff, y Andy Bechtolsheim.

ESTÁNDAR DE HERRAMIENTAS TECNOLÓGICAS PARA DESARROLLO DE SISTEMAS

- Operaciones de comprobación de límites de índices en arreglos evitando la posibilidad de sobrescribir o corromper memoria como resultado de punteros que señalan a zonas erróneas.
- Manejo de Excepciones

3.3.5. Portable. Java es un lenguaje multiplataforma en el sentido de que una aplicación no requiere ser re-compilada ó modificada para que pueda ser ejecutada en cualquier plataforma para la cual exista una maquina virtual (JVM).

3.3.6. Seguro. Antes de ejecutar un programa en Java, el código generado pasa por diversas pruebas con la finalidad de detectar posibles anomalías ó alteraciones de la integridad del código en bytecodes a interpretar:

- 3.3.6.1.** El código no debe producir desbordamiento de operandos en la pila.
- 3.3.6.2.** El tipo de los parámetros de todos los códigos de operación deben ser conocidos y correctos.
- 3.3.6.3.** No debe existir alguna conversión ilegal de datos, por ejemplo, convertir enteros en punteros.
- 3.3.6.4.** El acceso a los atributos de un objeto debe ser legal : public, private, protected
- 3.3.6.5.** El código debe cumplir con las reglas de acceso y seguridad establecidas.

De forma similar, en tiempo de compilación, existen mecanismos que evitan la generación de código inseguro ó vulnerable, En Java no existen instrucciones para la manipulación de la memoria eliminando la posibilidad de acceder a áreas ajenas ó la posibilidad de desbordamientos.

- 3.3.6.6.** En cuanto a código proveniente de una fuente remota (red), Java cuenta con mecanismos de seguridad. Algunos de ellos:
- 3.3.6.7.** Esquema de seguridad empleado por los Applets (Sandbox) ,
- 3.3.6.8.** Verificación de una llave digital por parte del cargador de clases antes de instanciar cualquier clase, etc.
- 3.3.6.9.** Separación de las clases provenientes de una fuente externa y las locales con la finalidad de prevenir el indebido reemplazo de una clase externa por una local. (El cargador primero carga las clases locales y después las remotas).
- 3.3.6.10.** Un aspecto considerado inseguro por los programadores es la posibilidad de la generación del código fuente a partir del archivo en bytecodes (.class) mediante el comando *javap*.

3.3.7. Multihilos. En java es posible ejecutar varias aplicaciones a la vez mediante el empleo de hilos de ejecución, lo que permite incrementar y optimizar el desempeño de una aplicación, al permitir la distribución de las tareas a realizar asociadas a hilos de ejecución debidamente organizadas y sincronizadas. La programación con multihilos es fundamental para el desarrollo de interfaces gráficas (swing), aplicaciones distribuidas, etc.

ESTÁNDAR DE HERRAMIENTAS TECNOLÓGICAS PARA DESARROLLO DE SISTEMAS

Para el modulo de Operación y control del pago del SIAR, esta característica es importante, como se verá mas adelante, debido principalmente a la cantidad de tareas que se requieren ejecutar al calcular una nómina.

3.3.8. Dinámico. Java es dinámico debido a que la carga de las clases se realiza cuando estas son requeridas, ya sea de forma local ó de algún punto de la red empleando algún protocolo de comunicación.

3.4. MANAJADOR BASES DE DATOS POSTGRESQL

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) que ha sido desarrollado de varias formas desde 1977. Comenzó como un proyecto denominado *Ingres* en la Universidad Berkeley de California. *Ingres* fue más tarde desarrollado comercialmente por la *Relational Technologies/Ingres Corporation*.

En 1986 otro equipo dirigido por *Michael Stonebraker* de Berkeley continuó el desarrollo del código de *Ingres* para crear un sistema de bases de datos objeto-relacionales llamado *Postgres*. En 1996, debido a un nuevo esfuerzo de código abierto y a la incrementada funcionalidad del software, *Postgres* fue renombrado a *PostgreSQL*, tras un breve periplo como *Postgres95*. El proyecto *PostgreSQL* sigue actualmente un activo proceso de desarrollo a nivel mundial gracias a un equipo de desarrolladores y contribuidores de código abierto.

3.4.1. Características PostgreSQL

PostgreSQL está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo.

PostgreSQL dispone de las siguientes características:

- 3.4.1.1. Transacciones**
- 3.4.1.2. Subselects**
- 3.4.1.3. Triggers**
- 3.4.1.4. Vistas**
- 3.4.1.5. Integridad referencial de claves externas**
- 3.4.1.6. Un sofisticado sistema de bloqueos**

Además, también presenta algunas funcionalidades que no encontramos en otras bases de datos comerciales, como tipos de datos definibles por el usuario, herencia, reglas, y control de concurrencia multi-versión que permite reducir el bloqueo de conexión.

3.4.2. A continuación se presenta una comparación de PostgreSQL con los demás motores:

Es posible instalarlo un número ilimitado de veces sin temor a sobrepasar la cantidad de licencias, la principal preocupación de muchos proveedores de bases de datos comerciales.

**ESTÁNDAR DE HERRAMIENTAS TECNOLÓGICAS
PARA DESARROLLO DE SISTEMAS**

- **Velocidad y rendimiento**
- **Flexibilidad para extenderse según se requiera**
- **Diseño escalable**
- **Mínimos requerimientos de administración**
- **Sigue estándares ANSI**
- **Excelente Atomicidad, Consistencia, Aislamiento y Durabilidad (Prueba del ACID)**

El siguiente cuadro ilustra el comportamiento de PostgreSQL contra otros motores de datos.

	MySQL	Microsoft	Oracle	PostgreSQL
Conexiones simultáneas (instalación default)	101	1000	41	32
Columnas en tabla	2819	1024	1000	1600
Máximo tamaño de renglón en tabla (omitiendo blobs)	65534	8036	255000	103275
Tamaño del renglón en tabla sin nulos (omitiendo blobs)	65502	8036	255000	103275
Tamaño query	1048574	16777216	16777216	16777216
Valor FALSE	0			0
Valor TRUE	1			1

SISTEMA OPERATIVO DE LIBRE DISTRIBUCION²

3.5. Linux

El sistema operativo Linux es compatible con UNIX el cual es de gran uso por sus características como servidor, además cuenta con dos características que lo hacen diferente a los demás sistemas comerciales las cuales son:

- 3.5.1.** Es de libre distribución **General Public License (GNU)**, esto significa que no se tiene que pagar una licencia
- 3.5.2.** Se distribuye junto con el **código fuente** por lo que si deseamos realizar correcciones o adecuaciones para alguna tarea específica podemos hacerlo directamente.
- 3.5.3.** El sistema es formado por el núcleo del sistema (**KERNEL**) y una serie de programas y librerías, los cuales han sido diseñados e implantados por una gran cantidad de programadores a lo largo del mundo, lo que lo hace un sistema en constante desarrollo.

² Software libre vs software propietario Ventajas y desventajas Culebro Juárez, Montserrat. Gomez Herrera, Wendy Guadalupe. Torres Sanchez, Susana. México, Mayo 2006.

ESTÁNDAR DE HERRAMIENTAS TECNOLÓGICAS PARA DESARROLLO DE SISTEMAS

3.5.4. Características

Dentro de las características más significativas que distinguen a Linux están:

- 3.5.4.1.** Multitarea
- 3.5.4.2.** Multiusuario
- 3.5.4.3.** Multiplataforma
- 3.5.4.4.** Multiprocesador
- 3.5.4.5.** Memoria Virtual
- 3.5.4.6.** Consolas virtuales múltiples
- 3.5.4.7.** Shells programables

3.5.5. Ventajas del software libre.

El software libre presenta una serie de ventajas sobre el software propietario por los derechos que otorga a sus usuarios. Algunas de estas ventajas pueden ser más apreciadas por los usuarios particulares, otras por las empresas, y otras por las administraciones públicas.

Principales ventajas.

3.5.5.1. Bajo costo de adquisición y libre uso. El software, como mercadería, por lo general no está a la venta. Lo que el usuario adquiere, a través de una erogación monetaria o sin ella, es una licencia respecto de los usos que puede dar a los programas en cuestión. El software no sólo cuesta un precio de adquisición de licencia. También cuesta mantenerlo, operarlo, ajustarlo.

3.5.5.2. Innovación tecnológica. El software libre, tiene como objetivo principal compartir la información, trabajando de manera cooperativa. Este es principalmente el modelo sobre el que la humanidad ha innovado y avanzado. La ideología de los defensores del software libre, es que el conocimiento le pertenece a la humanidad, sin hacer distinciones. Por lo tanto, los usuarios tienen un destacado papel al influir decisivamente en la dirección hacia donde evolucionan los programas: votando los errores que quieren que sean corregidos, proponiendo nueva funcionalidad al programa, o contribuyendo ellos mismos en el desarrollo del software

3.5.5.3. Requisitos de hardware menores y durabilidad de las soluciones. Aunque resulta imposible generalizar, si existen casos documentados que demuestran que las soluciones de software libre tienen unos requisitos de hardware menor, y por lo tanto son más baratas de implementar. Por ejemplo, los sistemas Linux que actúan de servidores pueden ser utilizados sin la interfaz gráfica, con la consecuente reducción de requisitos de hardware necesarios.

También es importante destacar que en el software propietario el autor puede decidir en un momento dado no continuar el proyecto para una cierta plataforma, para un hardware que considera antiguo, o descontinuar el soporte para una versión de su software. En las aplicaciones de software libre, estas decisiones no pueden ser tomadas por una empresa o individuo sino por toda una comunidad, con diferentes intereses. Lo que se traduce en un

ESTÁNDAR DE HERRAMIENTAS TECNOLÓGICAS PARA DESARROLLO DE SISTEMAS

mejor soporte -de manera general- para las versiones antiguas de software y de plataformas de hardware o software minoritarias.

3.5.5.4. Escrutinio publico. El modelo de desarrollo de software libre sigue un método a través de la cual trabajan de forma cooperativa los programadores que en gran parte son voluntarios y trabajan coordinadamente en Internet.

Lógicamente, el código fuente del programa esta a la vista de todo el mundo, y son frecuentes los casos en que se reportan errores que alguien ha descubierto leyendo o trabajando con ese código.

El proceso de revisión publica al que esta sometido el desarrollo del software libre imprime un gran dinamismo al proceso de corrección de errores. Los usuarios del programa de todo del mundo, gracias a que disponen del código fuente de dicho programa, pueden detectar sus posibles errores, corregirlos y contribuir a su desarrollo con sus mejoras. Son comunes los casos en que un error de seguridad en Linux se hace publico y con el la solución al mismo.

Con el software propietario la solución de los errores no llega hasta que el fabricante del programa puede asignar los recursos necesarios para solventar el problema y publicar la solución.

3.5.5.5. Independencia del proveedor. El software libre garantiza una independencia con respecto al proveedor gracias a la disponibilidad del código fuente. Cualquier empresa o profesional, con los conocimientos adecuados, puede seguir ofreciendo desarrollo o servicios para nuestra aplicación.

En el mundo del software propietario, sólo el desarrollador de la aplicación puede ofrecer todos los servicios, con el software libre, como su denominación lo indica, su uso es libre: todo aquel que lo tiene en su poder puede usarlo cuantas veces quiera, en cuantas maquinas quiera, a los fines que quiera.

De esta manera, utilizándolo, el usuario se libera de toda dependencia de un proveedor único, y puede administrar su crecimiento y operación con total autonomía, sin temor de costos ocultos ni extorsiones. Uno de los grandes problemas en la industria del software propietario es la dependencia que se crea entre el fabricante y el cliente.

BIBLIOGRAFÍA

[1] **SOMERVILLE**, Ian. *Ingeniería de Software*. Séptima edición. Editorial Pearson, Addison Wesley. España 2005

[2] **BOEHM, B. W.**, *Software Engineering., IEEE Transactions on Computers*, C-25, núm. 12.

[3] **PRESSMAN**, Roger S, *Ingeniería del Software, Un enfoque práctico*, Quinta edición, Editorial Mc Graw Hill, España 2002

[4] **BAUER, F. L.**, *Software Engineering, Information Processing, 71*, North Holland Publishing Co., Amsterdarn, 1972

[5] **SCMULLER**, Joseph. *Aprendiendo UML en 24 Horas*. Editorial Prentice Hall.

[6] **BARCLAY**, K y Savage J. *Object-Oriented Design with UML and Java*. Elsevier Butterworth Heinemann. Gran Bretaña, 2004.

[7] **BRANDON**, Daniel M. *Software Engineering For Modern Web Applications, Methodologies And Technologies*. Information Science Reference. United States of America. 2008

[8] **CADENHEAD**, Roger y Lemay Laura. *JAVA 6 in 21 Days*, 5th edition, Sams Publishing, U.S 2007.

[9] **TUYA**, Javier; Ramos Isabel, et.al., *Técnicas Cuantitativas para la gestión en la Ingeniería de Software*, Netbilbio, España 2007.

[10] **C. WORSLEY**, John y D. Drake Joshua, *Practical PostgreSQL*, First Edition, O'Reilly & Associates, Inc, U.S 2002.

[11] **MOODIE**, Matthew, *Pro Apache Tomcat 6*, Apress Publishing, U.S 2007

[12] **BOND**, Martin y Law Debbie. *Tomcat Kick Start*. Sams Publishing. First Edition, US, 2002

[13] **T. RAY**, Erick, *Learning XML*, O'Reilly Media, Second Edition, United States of America 2003

[14] **TERSINE**, Richard J, *Principles of Inventory and Materials Management*, p 2-3, 1999, Ed. North-Holland

[15] **DIAZ** De, Santos. *Compras e Inventarios*, 1996.

[16] *The Java EE 5 Tutorial*, Sun Microsystems, Octubre 2008

REFERENCIAS

Última revisión : 06/02/2012

The 1968/69 NATO Software Engineering Reports

<http://homepages.cs.ncl.ac.uk/brian.randell/NATO/NATOREports/index.html>

The Humble Programmer By Edsger W. Dijkstra

<http://userweb.cs.utexas.edu/~EWD/transcriptions/EWD03xx/EWD340.html>

About IEEE

<http://www.ieee.org/about/index.html>

Standars IEEE

<http://standards.ieee.org/>

What is free software?

<http://www.gnu.org>

Free Software

<http://www.fsf.org>