

Universidad de las Ciencias Informáticas

Facultad 3



Título: Implementación del Sistema Centro de Encuadernación (SICEN)

Trabajo de Diploma para optar por el título de
Ingeniero Informático

Autor: Manuel Álvarez Alonso

Tutora: Lic. Arismayda Dorado Risco

LA HABANA, Junio 2011

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año 2011.

Manuel Álvarez Alonso

Lic. Arismayda Dorado Risco

DATOS DE CONTACTO

Síntesis del Tutor: Lic. Arismaida Dorado Risco.

Profesión: Licenciado en Ciencias de la Computación.

Años de graduado: 6.

RESUMEN

El objetivo de este trabajo de diploma consiste en crear una aplicación informática para gestionar los procesos de encuadernado en las oficinas móviles del Servicio Autónomo de Registros y Notarias (SAREN), para el control de la información generada, la definición de estrategias y el apoyo a la toma de decisiones del mismo.

El sistema gestionará los procesos dentro de las oficinas móviles. Esto garantizará un flujo de información adecuado y un mejor control de los procesos dentro de estas oficinas. En el desarrollo de esta aplicación se implementaron funcionalidades como las que definen y agrupan los materiales a utilizar así como el personal autorizado a interactuar con el sistema, además se desarrolló el procesamiento de las distintas actas con sus respectivos documentos o productos asociados y el proceso de exportar datos en formato PDF entre otros elementos.

El desarrollo del sistema estuvo guiado por la metodología de desarrollo Rational Unified Process (RUP) y el patrón de arquitectura Modelo Vista Controlador. En la definición de las herramientas se tomó como guía las solicitudes hechas por el cliente que especificaban la utilización del Sistema Gestor de Contenido (CMS) Drupal (6.17) y el lenguaje de programación PHP (5.3.2) así como el framework Ext.Js (3.2) unido al sistema de gestión de base de datos, PostgreSQL (8.4.4). Además se utilizó el ORM (object relational mapper) Doctrine (1.2) para el mapeo de la base de datos y la librería FPDF (1.6) para la exportación de datos en formato PDF. El sistema fue objeto de un proceso de validación con el que se pudo demostrar su funcionalidad.

Palabras Claves: Control de la Información, Gestión de Procesos.

ÍNDICE

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA..... 1

1.1 Introducción..... 1

1.2 Aplicaciones de Gestión 1

1.3 Centro de Encuadernación 4

1.4 Plan de Desarrollo..... 6

 1.4.1 Metodología de desarrollo..... 6

 1.4.1.1 Rational Unified Process (RUP)..... 6

 1.4.2 Herramienta Case..... 6

 1.4.2.1 Visual Paradigm (3.0)..... 7

 1.4.3 Lenguaje de Modelado..... 7

 1.4.3.1 UML(2.0) 7

 1.4.4 Programación Orientad a Objetos (POO) 7

 1.4.5 Lenguajes de Programación..... 8

 1.4.5.1 PHP(5.3.2)..... 8

 1.4.5.2 HTML 9

 1.4.5.3 JavaScript..... 9

 1.4.7 Otras Tecnologías..... 10

 1.4.7.1 JSON..... 10

 1.4.7.2 Lenguajes de hojas de estilos..... 11

 1.4.8 Sistema Gestor de Base de Datos..... 11

 1.4.8.1 PostgreSQL (8.4.4)..... 12

 1.4.9 Herramienta para la administración de base de datos 12

1.4.9.1	PGAdmin III (1.8).....	12
1.4.10	Generación de reportes	13
1.4.11	Servidor Web.....	13
1.4.11.1	Servidor Web Apache (2.2).....	14
1.4.12	Navegador Web.....	14
1.4.12.1	Mozilla Firefox	14
1.4.13	Herramientas Colaborativas	15
1.4.13.1	Subversion(1.4.5)	15
1.4.14	Framework.....	15
1.4.14.1	ExtJS (3.2).....	16
1.4.14.2	Doctrine (1.2).....	16
1.4.15	Sistema Gestor de Contenido.....	17
1.4.15.1	Drupal (6.17).....	17
1.4.16	Estándar de codificación.....	18
1.4.17	Tratamiento de excepciones.....	19
1.4.18	Patrones de diseño.....	19
1.4.19	Patrones arquitectónicos.	20
1.4.19.1	Modelo-Vista-Controlador.....	20
1.4.20	Pruebas de software.....	21
1.4.20.1	Pruebas de Caja Blanca	22
1.4.20.1	Herramientas para el desarrollo de pruebas unitarias	23
1.4.20.2	Pruebas de Caja Negra	23
1.5	Conclusiones del Capítulo	24
CAPITULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.....		26

2.1	Introducción.....	26
2.2	Valoración crítica de los artefactos propuestos por los analistas.....	26
2.3	Diagrama de Componente del Sistema	27
2.4	Diagrama de despliegue.....	29
2.5	Estándares de codificación.....	30
2.5.1	Nomenclatura aplicada a los elementos significativos.....	31
2.7	Descripción de las clases u operaciones necesarias.....	34
2.7.1	Descripción de algunas clases significativas.....	34
2.7.2	Descripción de algunas operaciones significativas.....	37
2.8	Tratamiento de Errores y validación en la entrada de datos.....	39
2.9	Aplicación de patrones de diseño.....	40
2.9.1	Patrones Generales de Software para Asignación de Responsabilidades (GRASP).....	40
2.9.2	Patrones Gang of Four (GoF).....	40
2.10	Seguridad del sistema.....	41
2.11	Conclusiones del Capítulo	42
CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA		43
3.1.	Introducción.....	43
3.2.	Diseño y Resultados obtenidos de Casos de Pruebas de Caja Blanca aplicados.....	43
3.3.	Diseño de Casos de Prueba de Caja Negra aplicados.....	47
3.3.1.	Caso de Prueba Gestionar Producto	47
3.4.	Resultado de las Pruebas de Caja Negra.....	55
3.5.	Conclusiones del Capítulo.....	59
CONCLUSIONES		60

RECOMENDACIONES.....	61
BIBLIOGRAFÍA.....	62
GLOSARIO DE TÉRMINOS	65

ÍNDICE DE IMÁGENES

Figura 1 Ejemplo del uso del CamelCase	19
Figura 2 Ejemplo del uso del Lower Camel.....	19
Figura 3 Flujo de datos en el patrón arquitectónico modelo vista controlador.	21
Figura 4 Diagrama de Componente del Sistema.....	28
Figura 5 Diagrama de despliegue	30
Figura 6 Interfaz de la funcionalidad Crear acta.....	38
Figura 7 Interfaz de la funcionalidad Crear Documento.	39
Figura 8 Grafo de flujo del método mostrarFuncionarios.....	44
Figura 9 Juegos de datos para el camino básico # 1.	45
Figura 10 Test aplicado al camino # 1.....	45
Figura 11 Juegos de datos para el camino básico # 2.	45
Figura 12 Test aplicado al camino # 2.....	45
Figura 13 Juegos de datos para el camino básico # 3.	46
Figura 14 Test aplicado al camino # 3.....	46
Figura 15 Resultados de los test aplicados al método motrarFuncionario.....	46
Figura 16 Cantidad de no conformidades por caso de prueba en la primera iteración del sistema	56
Figura 17 Distribución de las no conformidades en la primera iteración	57
Figura 18 Distribución de las no conformidades en la segunda iteración.	57
Figura 19 Cantidad de no conformidades por caso de prueba en la segunda iteración del sistema.	58

ÍNDICE DE TABLAS

Tabla 1 Descripción de la clase Util.	32
Tabla 2 Descripción de la clase GenericTable.	33
Tabla 3 Descripción de la clase NomencladorController.	34
Tabla 4 Descripción de la clase Dacta.	34
Tabla 5 Descripción de la clase DmovimientoproductoTable	36
Tabla 6 Descripción de la clase Dmovimientodocumento.	36
Tabla 7 Complejidad y posibles caminos del método mostrarFuncionarios.....	44
Tabla 8 Secciones a aprobar en el Caso de Uso.	47
Tabla 9 Descripción de las variables.....	48
Tabla 10 Matriz de datos adicionar producto.....	49
Tabla 11 Matriz de datos eliminar producto	51
Tabla 12 Matriz de datos modificar producto.....	52

INTRODUCCIÓN

Los Registros y Notarías son oficinas públicas encargadas de legalizar las distintas actividades que requieran una autenticidad o constancia dentro de la sociedad, por ejemplo los procesos de compra venta o los matrimonios. Estas instalaciones son cruciales dentro de cualquier sistema jurídico por ser el eslabón que lo enlaza con la sociedad. Un sistema de Registros y Notarías con un adecuado funcionamiento, permite a la sociedad vivir en un ambiente legal.

La República Bolivariana de Venezuela y su Ministerio del Poder Popular para las Relaciones Interiores y Justicia (MIJ) cuentan con una red de oficinas compuesta por Notarías y Registros dirigidas por SAREN, ente adscrito al MIJ. Estas instituciones diariamente realizan trámites que, para su posterior consulta, son plasmados en diversas actas creadas para tal efecto. Al acumularse un grupo de actas, son sometidas a un proceso de encuadernación, conformando libros compuestos por pliegos que han sido archivados.

En la actualidad el proceso de encuadernación es gestionado y desarrollado de manera independiente por las oficinas de SAREN, estas, son las encargadas de contratar personal para garantizar la encuadernación de sus actas. Las personas que desarrollan la actividad de encuadernación, en la actualidad no están adscriptas al MIJ, debido a que el Ministerio no cuenta con recursos humanos calificados para esta tarea. Tal situación pone en riesgos los tomos de las oficinas, una vez que son trasladados fuera de las entidades, para su encuadernación, se da margen a la elaboración de copias ilegales o a la sustracción de los mismos y no se garantiza la calidad y presentación uniforme de los tomos. Además, por no contarse con un mecanismo para verificar los gastos y costos reales del proceso, se presenta un aumento de los presupuestos que maneja el MIJ por conceptos relacionados al proceso. Los costos en algunas ocasiones, no son generados netamente por el proceso de encuadernación, incurriendo ocasionalmente en el uso ilegal de recursos financieros.

Para evitar los riesgos que se presentan en el movimiento de las actas y los gastos excesivos o desviaciones de recursos, se concibió la creación de oficinas móviles con personal capacitado, que permitan al SAREN proveer a sus oficinas de un servicio de encuadernación.

Con el surgimiento de las oficinas móviles surge la necesidad de contabilizar y administrar de manera

eficiente los procesos que se desarrollarán en su interior, que en gran medida facilitarán al SAREN mantener el control constante a los procesos así como el apoyo a la toma de decisiones acertadas en lo que respecta al proceso de encuadernación.

Para darle solución a la situación problemática antes descrita se define como **problema de la investigación**: ¿Cómo gestionar los procesos de encuadernado en las oficinas móviles de SAREN, para el control de la información generada, la definición de estrategias y el apoyo a la toma de decisiones?

El problema planteado se enmarca en el **objeto de estudio**: Proceso de desarrollo de software en los sistemas de gestión.

El objeto de estudio delimita el **campo de acción**: Implementación de los procesos de gestión para el control de los tomos empastados y gastos de los centros de encuadernación.

Se plantea como **objetivo general** de la investigación: implementar una aplicación informática para gestionar los procesos de encuadernado en las oficinas móviles de SAREN, para el control de la información generada, la definición de estrategias y el apoyo a la toma de decisiones.

Para dar cumplimiento al objetivo general propuesto se definen los siguientes **objetivos específicos**:

- ❖ Desarrollar el marco teórico de la investigación.
- ❖ Implementar el sistema del centro de encuadernación.
- ❖ Validar la propuesta del sistema.

Para dar cumplimiento a los objetivos propuestos se definen las siguientes **tareas**:

Análisis de los sistemas de gestión de recursos a nivel nacional e internacional.

- ❖ Valoración de las tendencias y metodologías relacionadas con las técnicas de programación actuales, así como de las plataformas de desarrollo que la soportan.
- ❖ Descripción de los lenguajes y herramientas a utilizar en la implementación del Sistema Centro de Encuadernación (SICEN).
- ❖ Análisis de los artefactos entregados por los analistas.

- ❖ Implementación de las interfaces a partir del prototipo entregado por los analistas.
- ❖ Implementación de la capa de negocio que dé respuesta a los requisitos propuestos por los analistas.
- ❖ Implementación de la capa de acceso a datos.
- ❖ Implementación de validaciones y excepciones.

La respuesta anticipada al problema se formula en la siguiente **idea a defender**: si se implementa una aplicación informática para gestionar los procesos de encuadernado en las oficinas móviles de SAREN, se podrá gestionar los procesos de encuadernado y permitirá controlar la información generada, la definición de estrategias y el apoyo a la toma de decisiones de SAREN.

Los **Métodos Teóricos** posibilitan el estudio de las características del objeto de investigación que no se pueden visualizar de forma directa y proporcionan la construcción de modelos e hipótesis de investigación. De ellos se utilizaron en el desarrollo de la investigación los siguientes:

Análisis – Síntesis: Se utilizó para el procesamiento de toda la información relacionada con el tema de investigación, analizando los documentos que permitieran extraer los elementos más significativos relacionados con el objeto de estudio.

Histórico - Lógico: Este método brinda la posibilidad de analizar la trayectoria histórica real del tema de investigación y constatar teóricamente cómo ha evolucionado el tema tratado durante un determinado período de tiempo.

Modelación: Para la creación de abstracciones que explican la realidad, por ejemplo, todos los modelos y diagramas presentados en las diferentes fases de desarrollo de la ontología.

Inductivo-Deductivo: Se utilizó para la realización de un estudio de los procesos que se gestionan en las oficinas móviles del SAREN.

Los **Métodos Empíricos** describen y explican las características de los objetos, representan un nivel de la investigación cuyo contenido proviene de la experiencia y es sometido a cierta elaboración racional. En el trabajo de diploma se hizo uso del **método experimental** para el estudio de un objeto en el cual el

investigador crea las condiciones o adapta las existentes para el esclarecimiento de las propiedades, leyes y relaciones del objeto, para verificar una hipótesis, una teoría o un modelo.

El documento de tesis quedará estructurado de la siguiente forma:

Capítulo 1: Fundamentación Teórica. Se realiza la fundamentación teórica de la investigación donde se incluye el estado del arte a nivel internacional y nacional; las técnicas, las tecnologías y software que se utilizaron o que sirvieron de apoyo para la solución del problema, así como los temas y conceptos relacionados con la investigación y con la arquitectura y patrones utilizados en el desarrollo de la aplicación.

Capítulo 2: Descripción y análisis de la solución propuesta: Se realiza una apreciación de los componentes ya existentes que pueden ser rehusados, una valoración crítica de los artefactos propuestos por los analistas, se realiza un análisis de algunos de los algoritmos no triviales a implementar y se analizan la complejidad de los mismos, se seleccionan las estructuras de datos apropiadas para la implementación de estos algoritmos y se describen las principales clases a utilizar en el desarrollo del Sistema Centro de Encuadernación (SICEN).

Capítulo 3: Validación de la solución propuesta: Se describen diversos casos de pruebas de caja negra, que le fueron aplicados al sistema y los resultados que arrojó este proceso en las diversas iteraciones que se ejecutaron. Además se analizan pruebas de caja blanca que le fueron aplicadas al sistema.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción.

La necesidad de un control y flujo de la información, unido a una administración eficiente, son elementos que tienen hoy un gran valor dentro de cualquier empresa o entidad, pues facilitan la toma de decisiones estratégicas con rapidez y dominio de toda la información existente, igualmente mejora el control de los recursos humanos y materiales dentro de la organización. Unido a lo anterior se presenta la necesidad que tiene el Gobierno de la República de Venezuela de conquistar la soberanía tecnológica en sus principales Ministerios y Empresas. Teniendo en cuenta los elementos anteriores, en el presente capítulo se presenta una breve descripción de los Sistemas de Gestión y se expondrán las principales características de las herramientas, tecnologías y metodologías que, por decisión del cliente, fueron utilizadas para el desarrollo de la aplicación.

1.2 Aplicaciones de Gestión

Se entiende como **Aplicación de Gestión** aquella que se diseña para sustituir uno o varios procedimientos, tanto comerciales como administrativos, que habitualmente realiza una persona en una empresa o institución de forma presencial, por una aplicación, que permite realizar al cliente los mismos procedimientos de forma más rápida y/o óptima.

1.2.1 *Aplicación Web*

Una aplicación web o webapp (del inglés, Web Application) es una aplicación a la que se accede mediante un navegador web a través de una red como Internet. Es una aplicación de software de computadora que se codifica en un navegador apoyado en lenguajes (tales como HTML, JavaScript, CSS, PHP, Java, entre otros) y dependen de un navegador web (como Internet Explorer, Mozilla, Netscape) para hacer que la aplicación sea ejecutada.

Este tipo de aplicaciones posee ventajas como: el desarrollo de aplicaciones multiusuario bajo el concepto de compartición de información debido a que se accede a través de Internet. Además tiene la capacidad

de actualizar y mantener aplicaciones web sin distribuir e instalar software en miles de equipos clientes, lo que constituye una razón clave para su popularidad. Basta con una conexión a una red como Internet o una intranet para acceder a éstas y disponer de sus funcionalidades. Así como los clientes no requieren de grandes especificaciones de hardware en sus ordenadores para acceder a este tipo de aplicaciones.

Sin embargo, posee desventajas como: la dependencia entre la calidad de la conexión a la red y la comunicación constante con el servidor de la aplicación y este a su vez debe soportar la conexión concurrente de múltiples usuarios para que la ejecución de sus funciones sea de manera fluida, por lo que debe reunir condiciones de hardware óptimas.

1.2.2 Aplicación de gestión de inventarios extranjeros

1.2.2.1 SAP

SAP ERP propone gran cantidad de funcionalidades que le permiten analizar el negocio, optimizar las finanzas, gestionar los recursos humanos, operaciones y servicios corporativos. Además, proporciona soporte a la gestión de sistemas de administración de usuarios, gestión de configuración, gestión de datos centralizada y gestión de servicios de web. Es un sistema costoso, que brinda además las posibilidades de mantenimiento. SAP ERP El subsistema Inventario de SAP permite reducir los costos de almacenamiento, transporte, cumplimiento de pedidos y manipulación de materiales, a su vez mejora el servicio al cliente. Entre los beneficios adicionales de la gestión de inventarios se encuentran la mejora del flujo de caja, la visibilidad y el apoyo a la toma de decisiones. Para la gestión de almacenes, puede realizarse un seguimiento de la cantidad y el valor de todos sus materiales, realizar un inventario físico y optimizar los recursos del almacén. Los empleados pueden planificar, introducir y documentar movimientos de almacén interno gestionando las entradas y las salidas de mercancías, el almacenamiento, la recogida, el embalaje y los traslados físicos. (1)

1.2.2.2 CONDOR Enterprise

Sistema automatizado de alta complejidad y seguridad que abarca todos los aspectos del proceso contable de una entidad, tales como la dualidad de moneda y el pago por resultados. Este brinda mayor

autonomía al cliente para efectuar cambios de estructura sin necesidad de la intervención de especialistas, quedando registrados de forma que puedan ser auditados. Incluye la contabilidad utilizando varias monedas. (6) Entre sus módulos se encuentran la gestión de inventarios, que permite entre otras funcionalidades: tener un control de los esquemas de depósitos y almacenes, realizar inventarios rotativos, serializar los productos y generar transacciones de costeo. (2)

1.2.3 Aplicación de gestión de inventarios cubanas.

1.2.3.1 Gestión de Inventarios (ERP-Cuba)

Es un producto realizado por el ERP-Cuba que se desarrolla en la Universidad de las Ciencias Informáticas (UCI) denominado Cedrux. Es una aplicación web implementada en software libre, multiplataforma, cuenta con diferentes módulos como son: Nomencladores, Ajuste, Inventarios físicos, Configuración, Apertura, Devolución al almacén, Traspasos entre áreas, Documento de salida, Solicitudes, Planes de venta y Distribución. Rompe con todos los estándares que tenía el país en esta rama.

1.2.3.2 Rodas XXI. Versión 3.0

Sistema multiempresa y multiusuario creado por CIMATEL para automatizar la gestión empresarial. Contiene diferentes módulos que pueden usarse integrados o independientes. El módulo inventario de Rodas XXI le permite tener un control detallado de los inventarios de su entidad, realizado en el mismo momento que se registra un movimiento, así como su contabilización. Se pueden realizar operaciones de entradas y salidas de los almacenes con facilidad en el momento que se desee, generando el documento asociado al movimiento de que se trate de forma automática previa configuración del sistema.

Es posible operar con varios almacenes trabajando cada uno de forma independiente. Este módulo permite además, visualizar la información correspondiente a períodos anteriores, aunque en dichos períodos no se permite realizar cambios. Esta característica es compartida por todos los módulos de RODAS XXI. (3)

1.2.4 Valoración crítica de los sistemas analizados

A partir del estudio realizado a los diferentes sistemas de gestión de entidades, tanto nacionales como

internacionales que posean procesos de Control de Inventario, se determinó que las soluciones internacionales (SAP y Condor) usan tecnologías riesgosas, pues su realización está basada en la plataforma J2EE¹ y su máquina virtual es propiedad de la SUN². Además realizan los reportes de inventario de forma estática, cuando se debe brindar la posibilidad al usuario de filtrar los datos según las características de la recuperación, por otro lado la tecnología que utiliza es propietaria y su uso aumentaría los costos de la aplicación, e iría en contra de la política que en materia informática ha trazado el cliente donde define como lenguaje de programación a utilizar PHP. El pago de las patentes o licencias es un aspecto a tener en cuenta, debido a que elevaría los costos de desarrollo del sistema. La solución cubana (Rodas XXI) está desarrollada sobre software propietario también, ésta es una aplicación de escritorio, por lo que se hace necesaria la instalación en todas las computadoras de la entidad. Sin embargo, con una aplicación web y acceso a la red se tendrá acceso a la aplicación en cuestión. En el caso de Cedrux se puede decir que esta aplicación cumple con todas las expectativas con que fue realizada, solo se le puede señalar que carece de funcionalidades para su uso en otras entidades. En el caso de los cuatro sistemas analizados no cuentan con los medios para dar solución a la totalidad de las funcionalidades que necesita la solución del problema planteado, ni a los requerimientos tecnológicos pedidos por el cliente.

1.3 Centro de Encuadernación

Es un local u oficina donde se lleva a cabo el proceso de encuadernar libros. En el caso del centro de encuadernación del SAREN, las actas elaboradas como resultado del funcionamiento diario de las Notarias y Registros Principales, son los pliegos o cuadernos que serán encuadernados.

1.3.1 *Encuadernación*

La encuadernación, es la acción de coser o pegar varios pliegos o cuadernos de texto y ponerles

¹ **J2EE:** *Java Platform, Enterprise Edition o Java EE es una plataforma de programación para desarrollar y ejecutar aplicaciones en lenguaje de programación Java*

² **SUN:** *Stanford University Network, Red de la Universidad de Stanford.*

cubiertas; y tiene como objetivo dar al libro tres características o ventajas fundamentales: una mayor perdurabilidad de pliegos o cuadernos y con ellos la información que contienen. Facilitar su consulta, debido a que son organizados siguiendo un orden o regla dispuesto previamente y que se mantienen en todos los casos posteriores y proveer a los libros de una mejor presentación y en el caso de los Registros y Notarías, de una presentación estándar.

1.3.2 Actas

Un acta es una relación escrita de lo sucedido, tratado o acordado en una junta, certificación, testimonio, asiento o constancia oficial de un hecho. En el caso de SICEN son los documentos emitidos por los órganos adscriptos al SAREN y que clasificamos para su tratamiento dentro del centro de encuadernación en:

- 1 Acta de Entrega de documentos: Se registran los documentos que han sido encuadernados y/o restaurados que serán devueltos a las oficinas notariales y registrales.
- 2 Acta de Inventario parcial: Registra el estado de la existencia física de uno o varios productos del almacén.
- 3 Acta de Inventario general: Registra el estado de la existencia física de todos los productos del almacén.
- 4 Acta de producción: Registra el estado de la producción del centro en cuanto a cantidad de productos utilizados.
- 5 Acta de Recepción de productos: Tiene por finalidad registrar la entrada al centro de los productos necesarios para que el centro pueda ofrecer sus servicios a las oficinas notariales y
- 6 Acta de Registro de documento: Tiene por finalidad registrar la entrada al centro de los documentos que envían las oficinas notariales y registrales para encuadernar o restaurar. En la misma se reflejan los documentos que pueden ser procesados por el centro y cuáles no.

1.4 Plan de Desarrollo.

1.4.1 Metodología de desarrollo.

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software. Una metodología permite conocer paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además qué personas deben participar en el desarrollo de las actividades y qué rol desempeñan. Además, detallan la información que se debe producir como resultado de una actividad y la necesaria para comenzarla. (4)

1.4.1.1 Rational Unified Process (RUP).

RUP es una infraestructura flexible de desarrollo de software que proporciona prácticas recomendadas, probadas y una arquitectura configurable. RUP se encuentra entre las metodologías más utilizadas en el mundo por su probada capacidad de guiar exitosamente el desarrollo de disímiles proyectos garantizando abundante documentación y definiendo con exactitud las tareas y roles que intervienen en cada momento del desarrollo.

Para el desarrollo del SICEN se seleccionó como metodología de desarrollo de software RUP, ya que se cuenta con una mayor experiencia en el trabajo con esta metodología dentro del equipo de desarrollo. RUP es un proceso bien definido y estructurado; provee de un marco de desarrollo que abarca todo el ciclo de desarrollo del proyecto y es adaptable a sus especificaciones. Tiene como particularidad que en cada ciclo de iteración, se hace exigente el uso de artefactos, que pueden mejorarse en cada iteración posterior; y genera una documentación amplia sobre el software en desarrollo, permitiendo que los miembros del proyecto que continuarán trabajando en los siguientes flujos, tengan mayor documentación para entender el trabajo realizado hasta ese momento. También se genera gran cantidad de entregables para los clientes, aumentando su conocimiento sobre el software que se está produciendo. (5)

1.4.2 Herramienta Case

Se puede definir a las Herramientas CASE como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de

desarrollo de un software. Los estados en el Ciclo de Vida de desarrollo de un software son: Investigación Preliminar, Análisis, Diseño, Implementación e Instalación. (6)

1.4.2.1 Visual Paradigm (3.0).

Es una herramienta CASE que utiliza “UML”: como *lenguaje* de modelaje. Soporta la notación UML 2.1, ingeniería inversa, generación de código, importación desde Rational Rose, exportación/importación XMI, generador de informes, editor de figuras, integración con MS Visio, plug-in, integración IDE con Visual Studio, IntelliJ IDEA, Eclipse, NetBeans y otros. Entre sus características se incluyen el modelado colaborativo con CVS y Subversion, interoperabilidad con modelos UML2 a través de XML, entre otros. Está disponible en varias ediciones, cada una destinada a las necesidades: Enterprise, Professional, Community, Standard, Modeler y Personal. (7)

1.4.3 Lenguaje de Modelado.

El lenguaje de modelado es la notación (principalmente gráfica) que usan los métodos para expresar un diseño. (8)

1.4.3.1 UML(2.0)

El Lenguaje de Modelado Unificado (UML:Unified Modeling Language) es la sucesión de una serie de métodos de análisis y diseño orientadas a objetos que aparecen a fines de los 80's y principios de los 90's. UML es llamado lenguaje de modelado, no método. Los métodos surgen de la unión de un lenguaje de modelado y de un proceso. El UML, fusiona los conceptos de la orientación a objetos aportados por Booch, OMT y OOSE. UML incrementa la capacidad de lo que se puede hacer con otros métodos de análisis y diseño orientados a objetos. Los autores de UML apuntaron también al modelado de sistemas distribuidos y concurrentes para asegurar que el lenguaje maneje adecuadamente estos dominios (8).

1.4.4 Programación Orientad a Objetos (POO)

La Programación Orientada a Objetos (POO u OOP por sus siglas en inglés) es el paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones y programas de

computadora, estos objetos colaboran entre ellos para desarrollar tareas. (9) Está basada en varias técnicas como: El encapsulamiento, la herencia y el polimorfismo.

1.4.5 Lenguajes de Programación.

Un conjunto de sintaxis y reglas semánticas que definen los programas del computador. Es una técnica estándar de comunicación para entregarle instrucciones al computador. Un lenguaje le da la capacidad al programador de especificarle al computador, qué tipo de datos actúan y que acciones tomar bajo una variada gama de circunstancias, utilizando un lenguaje relativamente próximo al lenguaje humano. (10)

1.4.5.1 PHP(5.3.2)

PHP (acrónimo de *PHP: Hypertext Preprocessor*) es un lenguaje de código abierto muy popular especialmente adecuado para desarrollo web y que puede ser incrustado en HTML. En lugar de usar muchos comandos para mostrar HTML (como en C o Perl), páginas PHP contienen HTML con código incluido en el mismo El código PHP está entre medio de etiquetas de comienzo y final especiales `<?php` y `?>` que permitirán entrar y salir del "modo PHP".

Lo que distingue a PHP de algo, lado-cliente, como Javascript, es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá los resultados de ejecutar el script, sin ninguna posibilidad de determinar qué código ha producido el resultado recibido. El servidor web puede ser incluso configurado para que procese todos los archivos HTML con PHP. Lo mejor de usar PHP es que es extremadamente simple para el principiante, pero a su vez, ofrece muchas características avanzadas para los programadores profesionales. (11)

Posee ventajas como: fácil de aprender, se caracteriza por ser un lenguaje muy rápido, soporta en cierta medida la orientación a objeto; Clases y herencia. Es un lenguaje multiplataforma: Linux, Windows, entre otros. Posee capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras. Capacidad de expandir su potencial utilizando módulos. Es libre, por lo que se presenta como una alternativa de fácil acceso para todos. Incluye gran cantidad de funciones y no requiere definición de tipos de variables ni manejo detallado del bajo nivel.

Sin embargo posee desventajas como que: se necesita instalar un servidor web, todo el trabajo lo realiza el servidor y no delega al cliente, por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número. La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP y dificulta la organización por capas de la aplicación

1.4.5.2 *HTML*

Hyper Text Markup Language, es un lenguaje de programación muy sencillo que se utiliza para crear los textos y las páginas web. Es un lenguaje que se basa en las marcas para crear los hipertextos. Permite que se creen enlaces entre distintas partes del mismo documento o entre distintas fuentes de información a través de hiperenlaces o hipervínculos, e incluso insertar otros elementos como imágenes y sonidos.
(12)

HTML es un lenguaje de composición de documentos y especificación de ligas de hipertexto que define la sintaxis y coloca instrucciones especiales que no muestra el navegador, aunque si le indica cómo desplegar el contenido del documento, incluyendo texto, imágenes y otros medios soportados. HTML también indica cómo hacer un documento interactivo a través de ligas especiales de hipertexto, las cuales conectan diferentes documentos, ya sea en la computadora, o con otros recursos de Internet, como FTP
(13)

1.4.5.3 *JavaScript*

Es un lenguaje de programación scripting (guión o archivo de órdenes o archivo de procesamiento por lotes) basado en objetos no tipeado y liviano, utilizado para acceder a objetos en aplicaciones. Principalmente, se utiliza integrado en un navegador web permitiendo el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas. JavaScript es un dialecto de ECMAScript y se caracteriza por ser un lenguaje basado en prototipos, con entrada dinámica y con funciones de primera clase. JavaScript ha tenido influencia de múltiples lenguajes y se diseñó con una sintaxis similar al lenguaje de programación Java, aunque más fácil de utilizar para personas que no programan.

Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas web.

Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del DOM.
(14)

Posee ventajas como: Los programas escritos en este lenguaje no requieren de mucha memoria ni tiempo adicional de transmisión, por ser pequeños y compactos. No requiere un tiempo de compilación; ya que los scripts se pueden desarrollar en un período de tiempo relativamente corto. Es independiente de la plataforma hardware o sistema operativo, y funciona correctamente siempre y cuando exista un navegador con soporte JavaScript. Asegura la permanencia de una operación realizada (Durabilidad), y aunque falle el sistema esta no podrá deshacerse.

Sin embargo existen desventajas como que: Los script tienen capacidades limitadas, por razones de seguridad, el código es visible y puede ser leído por cualquiera y debe descargarse completamente antes de poderse ejecutar.

1.4.6 Herramienta de desarrollo

1.4.6.1 NetBeans (6.8)

Netbean 6.8 es un entorno de desarrollo integrado (IDE) libre y gratuito sin restricciones de uso, que permite a los desarrolladores crear aplicaciones web, enterprise, desktop y mobile de manera rápida y sencilla utilizando Java, JavaFX, PHP, JavaScript y Ajax, Ruby y Ruby on Rails, Groovy y Grails, y C/C++. Existe además un número importante de módulos para extender el NetBeans IDE.

En el desarrollo con PHP puede, a partir de las fuentes de Drupal, crearse un nuevo proyecto web y configurarse para Apache WebServer. Desde la versión 6.5 de Netbeans, para Drupal 6.x posee plugins configurables para desarrollo de módulos de Drupal, que autogeneran múltiples ficheros de configuración de módulos, temas y hooks.

1.4.7 Otras Tecnologías.

1.4.7.1 JSON

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Está constituido por dos estructuras:

- ❖ Una colección de pares de nombre/valor. En varios lenguajes se conoce como objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.
- ❖ Una lista ordenada de valores. En la mayoría de los lenguajes, se implementa como arreglos, vectores, listas o secuencias.

Estas son estructuras universales que son soportadas virtualmente por todos los lenguajes de programación, de una forma u otra. (15)

JSON es una forma de intercambio de datos y la ventaja que tiene, es que desde el lado de Javascript puede ser considerado como un objeto. Cuando se trate desde el lado del servidor, debe hacerse como si fuera un string. (16)

1.4.7.2 Lenguajes de hojas de estilos

CSS (Cascading Style Sheets,) es un formato usado en las páginas web para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Los estilos definen la forma de mostrar los elementos HTML y XML CSS permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas sus páginas vinculadas. (17)

Entre los beneficios concretos de CSS se encuentran: el control de la presentación de muchos documentos desde una única hoja de estilo. Control más preciso de la presentación y la aplicación de diferentes presentaciones en diferentes tipos de medios (pantalla, impresión).

1.4.8 Sistema Gestor de Base de Datos

Consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a esos datos. El Objetivo primordial de un SGBD es proporcionar un entorno que sea a la vez conveniente y eficiente para ser utilizado al extraer y almacenar información de la base de datos. (18). El gestor utilizado es PostgreSQL.

1.4.8.1 PostgreSQL (8.4.4)

Es un gestor de base de datos que proporciona un gran número de características que normalmente solo se encontraban en bases de datos comerciales como Oracle. Estas son algunas características presentes a partir de la versión de PostgreSQL 7.1.x:

DBMS Objeto – Relacional: PostgreSQL aproxima los datos a un modelo objeto-relacional y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son las consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, transactions, optimización de consultas, herencia, y arrays.

Altamente Extensible: Soporta operadores, métodos de acceso y tipos de datos definidos por el usuario.

Soporte SQL Comprensivo: Incluye características avanzadas tales como las uniones (joins).

Integridad Referencial: Utilizada para garantizar la validez de los datos de la base de datos.

API Flexible: La flexibilidad del API de PostgreSQL ha permitido a los vendedores proporcionar soporte al desarrollo fácilmente para el RDBMS PostgreSQL. Estas interfaces incluyen Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++, y Pike.

MVCC (Multi-Version Concurrency Control): Control de concurrencia Multi-Versión, es la tecnología que PostgreSQL utiliza para evitar bloqueos innecesarios. (19)

1.4.9 Herramienta para la administración de base de datos

1.4.9.1 PGAdmin III (1.8)

PGAdmin III es una aplicación gráfica para gestionar el SGBD PostgreSQL, siendo la más completa y popular con licencia Open Source. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usar en Linux, FreeBSD, Solaris, Mac OS X y Windows. Es capaz de gestionar versiones a partir de la PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL como Pervasive Postgres, EnterpriseDB, Mammoth Replicator y SRA PowerGres.

PGAdmin III está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. Su interfaz gráfica soporta todas las características de PostgreSQL y facilita enormemente la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados, soporte para el motor de replicación Slony-I, entre otros. La conexión al servidor puede hacerse mediante conexión TCP/IP o Unix Domain Sockets, y puede encriptarse mediante SSL para mayor seguridad. (20)

1.4.10 Generación de reportes

FPDF es una librería escrita en PHP que permite generar documentos PDF directamente desde PHP. Esta librería es de código abierto y gratuito. FPDF tiene entre sus principales características: elección de la unidad de medida, formato de página y márgenes. Gestión de cabeceras y pies de página. Salto de página automático. Salto de línea y justificación del texto automático. Admisión de imágenes (JPEG, PNG y GIF). Colores. Enlaces. Admisión de fuentes TrueType, Type1 y codificación, además la compresión de página.

FPDF solo necesita de zlib para activar la compresión y GD para soporte a GIF como extensiones para PHP y funciona con PHP4 ($\geq 4.3.10$) y PHP5. Para la selección de esta librería se tuvo en cuenta que esta debía ser de ser libre y garantizar la generación en un tiempo razonable de los reportes partiendo de PHP. Además de FPDF, se estudiaron DOMPDF y TCPDF. En el caso de la primera presenta como inconveniente el hecho de que su generación es a partir de HTML por lo que este debe ser generado antes de obtener el PDF final, lo que implica más tiempo de desarrollo en estas funcionalidades. En el caso de TCPDF es una librería basada en FPDF pero el equipo de desarrollo no cuenta con mucha experiencia en esta librería y su uso implicaría una capacitación al personal que utilizara esta librería y no se cuenta con el factor tiempo a favor. (21)

1.4.11 Servidor Web

Básicamente, un servidor web sirve contenido estático a un navegador, carga un archivo y lo sirve a través de la red al navegador de un usuario. Este intercambio es mediado por el navegador y el servidor que

hablan el uno con el otro mediante HTTP. (22)

1.4.11.1 Servidor Web Apache (2.2)

El servidor web Apache fue desarrollado por el Apache Server Project (Proyecto Servidor Apache). Su función principal es analizar todos los archivos solicitados por un navegador y mostrar los resultados correctos de acuerdo con el código dentro de ese archivo. Es un servidor web muy potente y flexible, y puede realizar prácticamente cualquier tarea que el usuario requiera. Cuenta con una arquitectura modular que le permite personalizarse mejor para las necesidades de cada sitio web y facilita a los usuarios adicionar cómodamente funcionalidad a sus ambientes específicos. Es robusto y seguro. Es multiplataforma, pues trabaja sobre todas las versiones recientes de UNIX y Linux, Windows, BeOs, mainframes. Facilita la visualización de código HTML. Una de las grandes ventajas que posee es su capacidad de autenticación, lo que proporciona el control de acceso de los usuarios y estaciones de trabajo a varios sitios Web; esto permite una regulación en la Internet en cuanto a los usuarios que están capacitados o no de conocer la información de una organización (23)

1.4.12 Navegador Web

Es una aplicación software que permite al usuario recuperar y visualizar documentos de hipertexto, comúnmente descritos en HTML, desde servidores web de todo el mundo a través de Internet. Esta red de documentos es denominada World Wide Web (WWW). Cualquier navegador actual permite mostrar o ejecutar gráficos, secuencias de vídeo, sonido, animaciones y programas diversos además del texto y los hipervínculos o enlaces. (24)

La aplicación estará optimizada para el navegador Mozilla Firefox aunque podrá ser visualizada por los navegadores más usados en la actualidad, Safari, Opera, Chrome y Explore.

1.4.12.1 Mozilla Firefox

Mozilla Firefox es un navegador de Internet, con interfaz de usuario gráfica, desarrollado por la Corporación Mozilla y un gran número de voluntarios externos. Es un navegador Web multiplataforma, que está disponible en versiones para Microsoft Windows y Linux. (25)

Posee diversas características como son: barra de direcciones inteligente, identificación instantánea del sitio web, Anti-malware, Anti-phishing, control de contenido, programas antivirus, gestor de contraseñas, actualización automática, bloqueador de ventanas emergentes, gestor de complementos, gestor de descargas, corrector ortográfico, restauración de sesiones, sugerencias de búsqueda, búsqueda en la web integrada, entre otros elementos. (26)

1.4.13 Herramientas Colaborativas

Las herramientas colaborativas, básicamente son los sistemas que permiten acceder a ciertos servicios que facilitan a los usuarios comunicarse y trabajar conjuntamente sin importar que estén reunidos en un mismo lugar físico. En general con ellos se puede compartir información en determinados formatos (audio, texto, video) y en algunos casos producir conjuntamente nuevos materiales producto de la colaboración. Muchos de ellos poseen avanzadas funcionalidades que facilitan tareas como publicación de información, búsquedas, filtros, accesos y privilegios (27). Como herramienta colaborativa se utilizo la siguiente:

1.4.13.1 Subversion(1.4.5)

Subversion es un sistema de control de versiones libre y de código fuente abierto que maneja ficheros y directorios a través del tiempo. La información se mantiene mediante un árbol de ficheros en un repositorio central. El repositorio es como un servidor de ficheros ordinario, excepto porque recuerda todos los cambios hechos a sus ficheros y directorios. Esto le permite recuperar versiones antiguas de sus datos, o examinar el historial de cambios de los mismos. Subversión puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintos ordenadores. A cierto nivel aporta la capacidad para que varias personas puedan modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones, fomenta la colaboración. (28)

1.4.14 Framework.

Un framework es un conjunto de interfaces de programación de aplicaciones (API) por sus siglas en ingles y herramientas destinadas a la construcción de un determinado tipo de aplicaciones de manera generalista. Algunos pueden llegar al detalle de definir los nombres de ficheros, su estructura, las

convenciones de programación, etc. (29) Los frameworks utilizados son:

1.4.14.1 ExtJS (3.2)

ExtJS 3.2 tiene dos tipos de licencias, LGPL y comercial. Este framework para JavaScript es muy utilizado en el desarrollo de aplicaciones Web con AJAX. Tiene una librería inmensa que permite configurar las interfaces Web de manera semejante a aplicaciones de escritorio. Tiene incluidos la mayoría de los controles de los formularios Web incluyendo Grids para mostrar datos y elementos semejantes a la programación de escritorio, como los formularios, paneles, barras de herramientas, menús y muchos otros. Dentro de su librería de componentes incluye componentes para el manejo de datos, lectura de XML, lectura de datos JSON e implementaciones basadas en AJAX. Presenta el uso de JavaScript con una programación orientada a objetos. (30)

Posee ventajas como: código reutilizable, independiente o adaptable a otros frameworks, orientada a la programación de interfaces tipo escritorio en la Web. El API es homogenizado independientemente del adaptador usado. Los controles siempre se verán igual. Posee soporte comercial y extensa comunidad de usuarios.

1.4.14.2 Doctrine (1.2)

Doctrine es un potente y completo sistema ORM (object relational mapper) para PHP 5.2+ con un DBAL (database abstraction layer) incorporado. Entre muchas otras cosas se tiene la posibilidad de exportar una base de datos existente a sus clases correspondientes y también a la inversa.

Su principal ventaja radica en poder acceder a la base de datos utilizando la programación orientada a objetos (POO) debido a que Doctrine utiliza el patrón Active Record para manejar la base de datos, tiene su propio lenguaje de consultas y trabaja de manera rápida y eficiente. Es fácilmente integrado a los principales frameworks de desarrollo utilizados actualmente. (30)

Utilizar un ORM tiene una serie de ventajas que facilitan tareas comunes y de mantenimiento:

- ❖ **Reutilización:** permitiendo llamar a los métodos de un objeto de datos desde distintas partes de la aplicación e incluso desde diferentes aplicaciones.

- ❖ **Encapsulación:** encapsula la lógica de los datos pudiendo hacer cambios que afectan a toda la aplicación únicamente modificando una función.
- ❖ **Portabilidad:** el uso de una capa de abstracción permite cambiar, en mitad de un proyecto, de una base de datos MySQL a una Oracle, debido a que no se utiliza una sintaxis MySQL, Oracle o SQLite para acceder al modelo, sino una sintaxis propia del ORM utilizado que es capaz de traducir a diferentes tipos de bases de datos.
- ❖ **Seguridad:** se implementan mecanismos de seguridad que protegen la aplicación de los ataques más comunes como SQL Injections.
- ❖ **Mantenimiento del código:** gracias a la correcta ordenación de la capa de datos, es sencillo modificar y mantener el código.

Además de todas las ventajas que conlleva un ORM, uno de sus puntos fuertes es su lenguaje **DQL** (Doctrine Query Language) inspirado en el **HQL** de Hibernate. En base al modelo de datos definido, se puede hacer **ingeniería inversa** de la base de datos existente generando el esquema, o si se empieza la aplicación desde cero, crear el modelo en la sintaxis específica que propone Doctrine y luego generar toda la base de datos.

1.4.15 Sistema Gestor de Contenido

Un CMS es un sistema de software para ordenador que permite organizar y facilitar la creación de documentos y otros contenidos de un modo cooperativo. Con frecuencia, un CMS es una aplicación web usada para gestionar sitios web y contenidos web. (31)

1.4.15.1 Drupal (6.17)

Drupal es un sistema de gestión de contenido modular y muy configurable. Es un programa de código abierto, con licencia GNU/GPL, escrito en PHP, desarrollado y mantenido por una activa comunidad de usuarios. Destaca por la calidad de su código y de las páginas generadas, el respeto de los estándares de la web, y un énfasis especial en la usabilidad y consistencia de todo el sistema.

El diseño de Drupal es especialmente idóneo para construir y gestionar comunidades en Internet. No

obstante, su flexibilidad y adaptabilidad, así como la gran cantidad de módulos adicionales disponibles, hace que sea adecuado para realizar muchos tipos diferentes de sitio web. Cuenta con una amplia comunidad que le aporta gran números de módulos y proponen y soluciona problemas que pueda presentar. (32)

Posee ventajas como que: los clientes están familiarizado con esta tecnología, el código fuente de Drupal está libremente disponible bajo los términos de la licencia GNU/GPL, los administradores de Drupal no tienen que establecer permisos para cada usuario. En lugar de eso, pueden asignar permisos a un 'rol' y agrupar los usuarios por roles, todo el contenido en Drupal es totalmente indexado en tiempo real y se puede consultar en cualquier momento, usa el mod_rewrite de Apache para crear URLs que son manejables por los usuarios y los motores de búsqueda e incorpora una capa de abstracción de base de datos que actualmente está implementada y mantenida para MySQL y PostgreSQL. Puede funcionar con Apache o Microsoft IIS como servidor web y en sistemas como Linux, BSD, Solaris, Windows y Mac OS X.

1.4.16 Estándar de codificación.

Los estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. (33)

Existen numerosos estándares de codificación con sus características propias y es muy frecuente la combinación de algunos de estos dentro de un mismo sistema. Dentro de esta gama de estándares encontramos el CamelCase y el Lower Camel definidos de la siguiente forma.

- ❖ **CamelCase** es un término usado para describir una técnica de escritura que aprovecha el principio de cada palabra y elimina cualquier espacio, dando a las palabras que el aspecto de las jorobas en la espalda de un camello.

```
public static function PrintCodeBar()  
{  
  
}
```


Figura 1 Ejemplo del uso del CamelCase

- ❖ **Lower Camel** similar al Camel Case sólo que la primera letra de la primera palabra es también en minúscula.

```
public static function printCodeBar()  
{  
  
}
```

Figura 2 Ejemplo del uso del Lower Camel

1.4.17 Tratamiento de excepciones.

Para garantizar el correcto funcionamiento del sistema se debe tener en cuenta un tratamiento de excepciones, el sistema captura todas aquellas excepciones que son lanzadas y se le facilita un tratamiento para que no colapse.

De esta manera se da solución a las problemáticas de los lanzamientos de excepciones que se dan en el sistema según las peticiones del usuario, capturando los tipos de errores lanzados y mostrando de manera visible al usuario mensajes de confirmación, esto le dará una idea de qué es lo que está incorrecto y cómo solucionarlo. Para el trabajo de excepciones fueron utilizadas las herramientas que para este fin define el lenguaje de programación PHP.

El lenguaje de programación PHP 5 tiene un modelo de excepciones similar al de otros lenguajes de programación. Una excepción puede ser lanzada (*thrown*), y atrapada ("*caught*") dentro de PHP.

Cuando una excepción es lanzada, el código siguiente a la declaración no será ejecutado, y PHP intentará encontrar el primer bloque *catch* coincidente. Si una excepción no es capturada, se emitirá un Error Fatal de PHP con un mensaje "*Uncaught Exception*" ("Excepción No Capturada"). (33)

1.4.18 Patrones de diseño.

Son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos,

basadas en la experiencia y que se ha demostrado que funcionan. (34)

Los patrones de diseño se pueden agrupar en varios grupos entre los que se encuentran los GoF o Banda de los Cuatro por sus siglas en inglés. Ellos fueron los creadores de los patrones Singleton, Fachada, Adapter, Proxy, Composite, Abstract Factory, Factory method y Observer / Publicar-Suscribir entre otros. (35)

Los patrones GRASP (General Responsibility Assignment Software Patterns) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en formas de patrones. El nombre se eligió para indicar la importancia de captar estos principios, si se quiere diseñar eficazmente el software orientado a objetos. Dentro de los patrones que agrupan se encuentran Experto en información, Controlador y Creador, entre otros. (36)

1.4.19 Patrones arquitectónicos.

Definen los aspectos fundamentalmente de la estructura de un sistema o software, especifican un conjunto predefinido de subsistemas con sus responsabilidades y una serie de recomendaciones para organizar los distintos componentes. (37)

1.4.19.1 Modelo-Vista-Controlador.

El patrón conocido como Modelo-Vista-Controlador (MVC) separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario en tres elementos diferentes. El patrón MVC organiza el código en base a su función; separa el código en tres capas principales: (39)

La capa del **modelo** define la lógica de negocio. Encapsula los datos y las funcionalidades. Es independiente de cualquier representación de salida y/o comportamiento de entrada. Esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos.

La **vista** es lo que utilizan los usuarios para interactuar con la aplicación, los gestores de plantillas pertenecen a esta capa. Este presenta el modelo en un formato adecuado para interactuar, usualmente la

interfaz de usuario.

El **controlador** recibe las entradas, traducidas a solicitudes de servicio para el modelo. Es un bloque de código que realiza llamadas al modelo para obtener los datos y se los pasa a la vista para que los muestre al usuario. Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

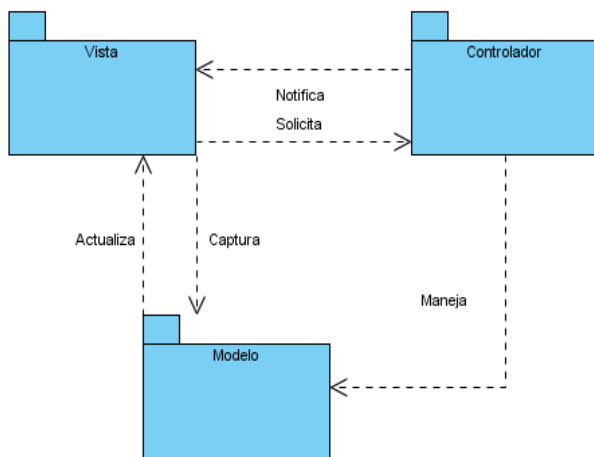


Figura 3 Flujo de datos en el patrón arquitectónico modelo vista controlador.

1.4.20 Pruebas de software.

Las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente. La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación. La creciente inclusión del software como un elemento más de muchos sistemas y la importancia de los costos asociados a un fallo del mismo, han motivado la creación de pruebas más minuciosas y bien planificadas.

Objetivos de las pruebas:

- ❖ La prueba es un proceso de ejecución de un programa con la intención de descubrir un error.
- ❖ Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.
- ❖ Una prueba tiene éxito si descubre un error no detectado hasta entonces

Los software pueden ser probado de varias formas que podemos agrupar por sus objetivos en dos, en un primer grupo estarían situadas las que persiguen comprobar que todos los componentes internos encajan y se ajustan a las especificaciones previamente definidas, en un segundo grupo estarían las pruebas encaminadas a comprobar las distintas funcionalidades que debe desarrollar el sistema. Al primer grupo de pruebas se les denomina pruebas de caja blanca y al segundo pruebas de caja negra.

1.4.20.1 Pruebas de Caja Blanca

La prueba de caja blanca se basa en el diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivarlos. Mediante la prueba de la caja blanca el ingeniero del *software* puede obtener casos de prueba que: garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada modulo, programa o método. Ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa. Ejecuten todos los bucles en sus límites operacionales y ejerciten las estructuras internas de datos para asegurar su validez.

Es por ello que se considera a la prueba de Caja Blanca como uno de los tipos de pruebas más importantes que se le aplican a los *software*, logrando como resultado que disminuya en un gran porcentaje el número de errores existentes en los sistemas y por ende una mayor calidad y confiabilidad.

Prueba del camino básico.

La prueba del camino básico es una técnica de prueba de la Caja Blanca propuesta por Tom McCabe. Esta técnica permite obtener una medida de la complejidad lógica de un diseño y usar esta medida como guía para la definición de un *conjunto básico*.

La idea es derivar casos de prueba a partir de un conjunto dado de caminos independientes por los cuales puede circular el flujo de control. Para obtener dicho conjunto de caminos independientes se construye el

Grafo de Flujo asociado y se calcula su complejidad ciclomática. Los pasos que se siguen para aplicar esta técnica son:

1. A partir del diseño o del código fuente, se dibuja el grafo de flujo asociado.
2. Se calcula la complejidad ciclomática del grafo.
3. Se determina un conjunto básico de caminos independientes.
4. Se preparan los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico.

Los casos de prueba derivados del conjunto básico garantizan que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa. (38)

1.4.20.1 Herramientas para el desarrollo de pruebas unitarias.

PHPUnit (3.4) es una herramienta para la ejecución de pruebas al desarrollo de programas/sistemas efectuados específicamente en PHP. Se puede aplicar al proceso de desarrollo de sistemas con la idea central de fortalecer la fase de pruebas y poder garantizar de una mejor forma al cliente de los productos al finalizar el desarrollo de un proyecto. Asume una programación orientada a objetos y existen versiones para PHP 4 y para PHP 5. (5)

1.4.20.2 Pruebas de Caja Negra

Se refiere a las pruebas que se llevan a cabo sobre la interfaz del software, por lo que los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta, así como que la integridad de la información externa se mantiene. Esta prueba examina algunos aspectos del modelo fundamentalmente del sistema sin tener mucho en cuenta la estructura interna del software.

Entre los principales problema que posibilitan detectar las pruebas de caja negra se encuentran: funciones incorrectas o ausentes, errores de interfaz, en estructuras de datos o en accesos a las Bases de Datos externas, de rendimiento y errores de inicialización y terminación.

Para desarrollar la prueba de caja negra existen varias técnicas, entre ellas están:

- ❖ Técnica de la Partición de Equivalencia: esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del *software*.
- ❖ Técnica del Análisis de Valores Límites: esta Técnica prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.
- ❖ Técnica de Grafos de Causa-Efecto: es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones. (38)

Dentro de las técnicas antes definidas la más utilizada es la de partición equivalente, pues esta permite evaluar entradas validas e inválidas al software permitiendo encontrar problemas en las funcionalidades que usando otras técnicas requerirían más tiempo y pruebas, por estos elementos esta fue la técnica seleccionada para desarrollar las pruebas al sistema.

Una partición equivalente es una técnica de prueba de Caja Negra que divide el dominio de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. El diseño de estos casos de prueba para la partición equivalente se basa en la evaluación de las clases de equivalencia.

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 o inválidos para condiciones de entrada. Regularmente, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica. (38)

1.5 Conclusiones del Capítulo

Los sistemas de gestión existentes nacional e internacionalmente no cuentan con todas las funcionalidades necesarias para el Sistema Centro de Encuadernación (SICEN) o no las soportan.

Después de realizar un estudio de las herramientas de inventario y gestión de almacenes, se concluye que en la actualidad no existe un sistema informático integral de gestión que cumpla con la totalidad de los requerimientos de funcionalidad y seguridad que se esperan de este sistema y además que esté

desarrollado con la tecnología acorde con las políticas del cliente.

Se define como lenguaje de programación a utilizar PHP (5.3.2), el trabajo en la parte visual se hará con ExtJS (3.2), para el ORM (Object Relation Mapper) Doctrine (1.2), se utilizará además Drupal y PostgreSQL (8.4.4), como herramientas para el desarrollo del sistema se definió NetBeans 6.8, PGAdmin III, Apache (2.2). Se utilizará la metodología de desarrollo de software Proceso Racional Unificado (RUP) y como patrón arquitectónico el Modelo-Vista-Controlador.

CAPITULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

2.1 Introducción.

En este capítulo se relacionan varios aspectos del desarrollo de la solución informática. Se comienza realizando una valoración de los artefactos propuestos por los analistas del sistema. Se describen los estándares de codificación utilizados. Se muestran las clases que son reutilizadas durante la implementación, con el objetivo de facilitar la comprensión del funcionamiento de los componentes implementados. También se describe algunos algoritmos que son considerados de peso dentro del sistema y se muestran las interfaces asociadas a estos. Se hace un análisis del diagrama de componente del sistema y por último son descritos los patrones del diseño aplicados durante el desarrollo.

2.2 Valoración crítica de los artefactos propuestos por los analistas.

La ingeniería de requisitos ayuda a los ingenieros de software a entender mejor el problema en cuya solución trabajan. Incluye el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio, que es lo que el cliente quiere y como interactuaran los usuarios finales con el software (39).

Uno de los artefactos generados durante esta etapa es la Especificación de Requisitos, el cual fue propuesto por los analistas del sistema con el objetivo de describir los servicios que ha de ofrecer el sistema y las restricciones asociadas a su funcionamiento. En este artefacto fueron descritos los Requisitos Funcionales y los Requisitos No Funcionales

Los Requerimientos funcionales especifican acciones que el sistema debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física, los requerimientos funcionales especifican el comportamiento de entrada y salida del sistema y surgen de la razón fundamental de la existencia del producto. Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable (40).

CAPITULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Se considera que el artefacto recibido cumple con todos los parámetros, por lo que se ha podido recopilar la información necesaria para establecer la funcionalidad que se quiere alcanzar con el sistema, así como han sido reflejadas todas las necesidades de los clientes para esta versión del producto.

2.3 Diagrama de Componente del Sistema

A continuación se muestra el diagrama de componente del Sistema Centro de Encuadernación. En la vista están incluidos el conjunto de ficheros necesarios para la presentación de las interfaces, básicamente incluye ficheros de imágenes, estilos (.CSS), JavaScript (.js) y ficheros del Theme system de Drupal (.tpl.php, .info). La capa controlador agrupa los ficheros que garantizan la comunicación con el core de drupal, según las especificaciones del framework. Encapsula las tareas comunes de la lógica de cada módulo en hooks que escuchan ante las acciones para las que son definidos. Además funciona como controlador que prepara las configuraciones y atiende las peticiones de la vista, en caso necesario consulta el modelo para satisfacer las necesidades de información. Define un punto de entrada para todo el módulo e impide el acceso indebido. La capa de modelo contiene las ficheros de configuraciones funciones y clases que permiten la comunicación con el subsistema Doctrine.

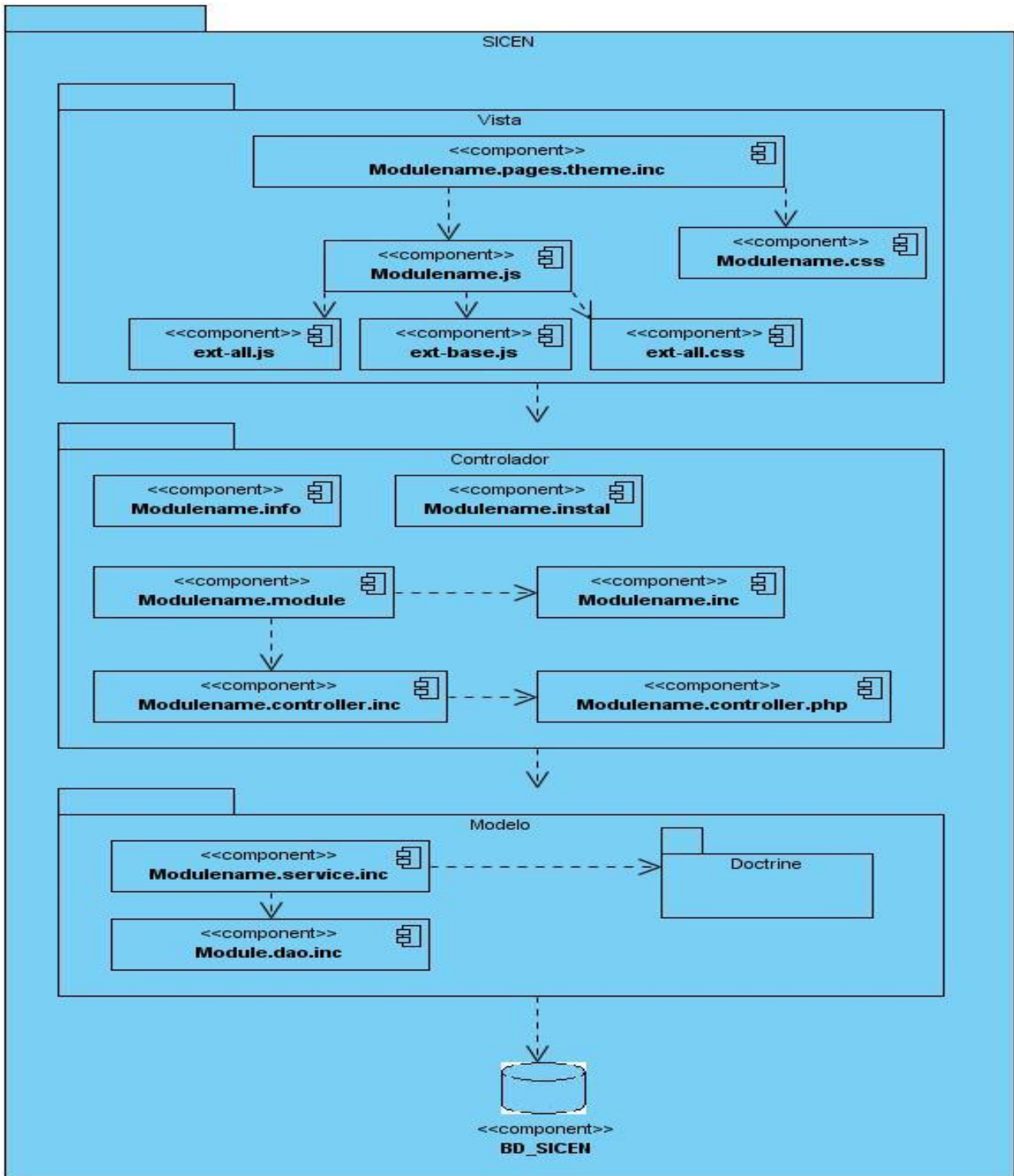


Figura 4 Diagrama de Componente del Sistema

2.4 Diagrama de despliegue.

A continuación se describen los elementos que estarán involucrados en el despliegue de la solución informática.

Impresora: Este dispositivo estará a disposición de la estación de trabajo cliente para visualizar en una hoja impresa los datos requeridos por el cliente.

Estaciones de trabajo con la Aplicación Cliente: Se refiere a las estaciones de trabajo que el usuario utilizará para acceder a la aplicación Web.

Servidor de aplicación: Servidor de Aplicaciones utilizado para la publicación de la aplicación; y para lograr la conexión del sistema con la PC Cliente se utiliza HTTP como protocolo de comunicación.

Servidor de Datos: Se refiere al servidor que radica en cada nodo del Servidor de aplicación, donde van a estar centralizados los datos recopilados por la aplicación.

Descripción de elementos e interfaces de comunicación

- ❖ HTTP: La PC cliente accede al servidor mediante una la red LAN, sin acceso a internet. El protocolo de transferencia de hipertexto (HTTP, HyperText Transfer Protocol) es el protocolo usado en cada transacción de la Web (WWW).
- ❖ USB: Para brindar el servicio de impresión se utiliza el protocolo USB para la conexión entre la estación de trabajo y el dispositivo impresora. El USB, Universal Serial Bus (bus universal en serie) o Conductor Universal en Serie (CUS), abreviado comúnmente USB, es un puerto que sirve para conectar periféricos a un ordenador.

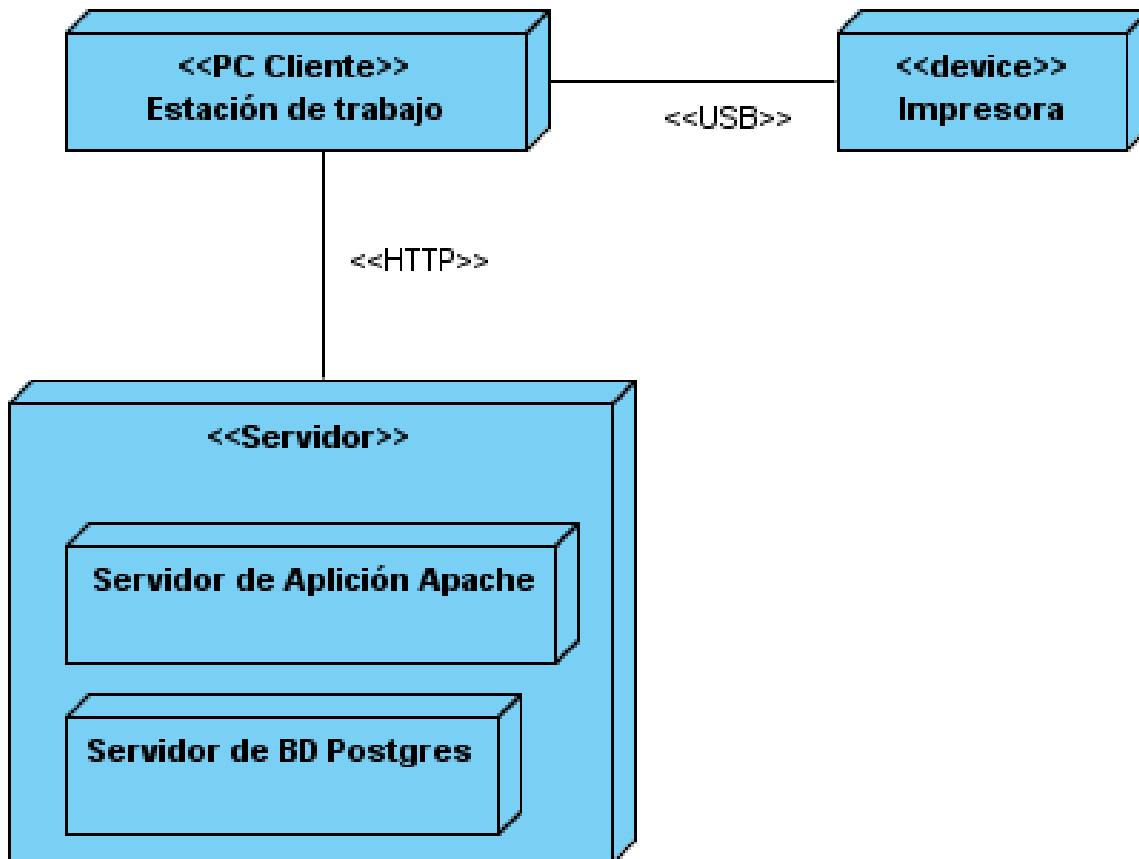


Figura 5 Diagrama de despliegue

2.5 Estándares de codificación.

Los estándares de codificación en el marco del Sistema Centro de Encuadernación van a permitir una mejor integración entre las líneas de producción y se establecerán las pautas que conlleven a lograr un código más legible y reutilizable. Para el caso de SICEN nos basamos en el estándar seleccionado por Drupal debido a que el sistema soportado en este CMS y resultar de interés para el sistema mantener una homogeneidad en el código que compondrá la solución. Estas normas de codificación se aplican a

códigos dentro de Drupal y sus módulos y se basa libremente en la codificación de las normas de PEAR³. (41) Dentro de su estándar de codificación, Drupal, utiliza otros estándares como son CamelCase y lowerCamelCase.

2.5.1 Nomenclatura aplicada a los elementos significativos.

Nomenclatura de las clases

Los nombres de las clases son escritas con Mayúscula y sin utilización de caracteres especiales entre una palabra y otra en caso de tenerlas.

Ejemplo: InventarioGeneral

Nomenclatura según el tipo de clases:

- ❖ Clases controladoras: Las clases controladoras después del nombre llevan la palabra “Controller”.

Ejemplo: GrupoPropiedadController

- ❖ Las clases que se encuentran dentro del directorio Generated el nombre comienza con la palabra: “Base” y seguido el nombre de la tabla en la Base de Datos. Ejemplo: BaseDacta.

- ❖ Las clases del negocio se encuentran dentro del directorio Models, si representan tablas de datos inician con la letra D y son nomencladores con N y ambas terminan con la palabra Table. Ejemplo: DmovimientoinventarioTable y NfasedocumentoTable

Nomenclatura de las funciones

El nombre a emplear para las funciones se escribe con la primera palabra en minúscula, en caso de que

³ PEAR es un framework de distribución de componentes reutilizables para PHP.

CAPITULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

sea un nombre compuesto se empleará notación lowerCamelCase. Ejemplo: actaDetails

En caso de ser una acción de la clase controladora se le pone el nombre y seguida la palabra: "Action".

Ejemplo: readUsersAction

Nomenclatura de las variables

El nombre a emplear para las variables se escribe con la primera palabra en minúscula, en caso de que sea un nombre compuesto se empleará guión bajo para separarla. Ejemplo: \$ data_header

Nomenclatura de las constantes

El nombre a emplear para las constantes se escribe con todas las letras en mayúscula. Ejemplo: TEXT

Nomenclatura de los atributos

El nombre a emplear para los atributos se escribe con la primera palabra en minúscula, en caso de que sea un nombre compuesto se unirán mediante el carácter "_". Además en caso de ser un objeto se comienza con: "_" y después se escribe el nombre. Ejemplo: \$data_table , _listado_acta

2.6 Análisis de clases reutilizadas en la implementación.

Util

Esta clase contiene métodos que son utilizados para configuración de distintos módulos y dentro de ellos el trabajo con url y el paginado entro otras.

Tabla 1 Descripción de la clase Util.

Nombre: Util	
Tipo de clase: Modelo	
Para cada responsabilidad	
Nombre	Descripción
arrayPager(\$array = array(), \$limit = null, \$offset = null)	Permite dar un formato estándar a un arreglo

CAPITULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

generateId()	Permite generar ID
generateCode(\$cantidad)	Permite generar código
addUrlToJs(\$key, \$url, \$module)	Permite agregar la url
addGobalVar(\$key, \$value)	Permite adicionar una variable global
initModule(\$module)	Inicializa un módulo
checkConfiguration()	Chequea la configuración del módulo Doctrini

GenericTable

Esta clase contiene algunas de las funcionalidades más utilizadas durante el acceso a datos adaptadas a las características del sistema.

Tabla 2 Descripción de la clase GenericTable.

Nombre: GenericTable	
Tipo de clase: Modelo	
Para cada responsabilidad	
Nombre	Descripción
findTableBy(\$table,\$fields,\$values,\$idValue)	Permite buscar en una tabla según varios criterios
getRecords(\$table,\$limit=null,\$offset=null,\$query,\$sql=null,\$count = true)	Permite obtener registros provenientes de una tabla
getRecordsMultiTable(\$table,\$limit=null,\$offset=null,\$query = null,\$sql=null,\$count=true,\$tableNames = array(),\$asoc = array())	Permite obtener registros provenientes de varias tablas.
getTotalRecords(\$table)	Permite obtener el total de los registros de la tabla.
createRecord(\$className,\$values)	Permite crear un nuevo registro
updateRecord(\$table,\$id,\$values)	Permite actualizar un registro
deleteRecord(\$table,\$id,\$idname = null)	Permite eliminar un registro

NomencladorController

Esta clase contiene las funcionalidades que comparten todo los nomencladores dentro del proyecto.

Tabla 3 Descripción de la clase NomencladorController.

Nombre: NomencladorController	
Tipo de clase: Controladora	
Para cada responsabilidad	
Nombre	Descripción
readAction(Request \$request)	Permite obtener los nomencladores
getDetallesAction(Request \$request)	Permite obtener los detalles de un nomenclador
createAction(Request \$r)	Permite crear un nomenclador
updateAction(Request \$r)	Permite actualizar un nomenclador
destroyAction(Request \$r)	Permite eliminar un nomenclador

2.7 Descripción de las clases u operaciones necesarias.

2.7.1 Descripción de algunas clases significativas

A continuación se realiza una breve descripción de las funcionalidades implementadas en las clases de mayor peso dentro del sistema.

Clase del modelo : Dacta.

Esta clase contiene funcionalidades para el trabajo con las actas.

Tabla 4 Descripción de la clase Dacta.

Nombre: Dacta
Tipo de clase: Modelo

CAPITULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Para cada responsabilidad	
Nombre	Descripción
read(\$limit,\$offset,\$query)	Permite obtener las actas existentes
getActa(\$idacta)	Permite obtener un acta
mostrarDetalles(\$idacta,\$limit=null,\$offset=null,\$query=null)	Permite obtener los detalles de un acta
mostrarFuncionarios(\$idacta)	Permite obtener los funcionarios asociados a un acta
readMunicipios(\$table,\$limit,\$offset,\$query)	Permite obtener municipios asociados
readOficinas(\$table,\$limit,\$offset,\$query)	Permite obtener municipios asociados
comprobarFuncionario(\$user)	Permite obtener los datos del funcionario que está autenticado
create(\$data,\$funcionario)	Permite adicionar un acta
update(\$data)	Permite actualizar los datos de un acta
comprobarMunicipio(\$datos,\$id)	Permite comprobar la existencia de un municipio
updateFuncionario(\$funcionario)	Permite actualizar los datos de un funcionario
compruebaFase(\$id)	Permite obtener la fase en la que se encuentra el acta
terminarActaprod(\$datos)	Permite cerrar el acta de producto
terminarActadocum(\$datos)	Permite cerrar acta de documentos
aprobar(\$datos,\$funcionario)	Permite aprobar el acta
cambiarFase(\$datos)	Permite cambiar la fase en la que se encuentra el acta
destroy(\$id)	Permite eliminar actas
cancelarActa(\$datos)	Permite cancelar un acta

Clase del modelo: DmovimientoproductoTable.

Esta clase contiene funcionalidades para el trabajo con los movimientos a los que puede ser sometido un producto.

CAPITULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Tabla 5 Descripción de la clase DmovimientoproductoTable

Nombre: DmovimientoproductoTable	
Tipo de clase: Modelo	
Para cada responsabilidad	
Nombre	Descripción
eliminarVarios(\$ids)	Permite eliminar productos con sus movimientos asociados
getProducta(\$idacta)	Permite obtener los productos asociados a un acta
getProducto(\$idMov)	Permite obtener los productos asociados a un movimiento
getMovimientos(\$idActa,\$limit=null,\$offset=null,\$query=null)	Permite obtener los movimientos asociados a un acta
cancelarVarios(\$ids)	Permite cancelar uno o varios movimientos
findByIdMov(\$idMov)	Permite buscar productos a través del movimiento

Clase del modelo: Dmovimientodocumento.

Esta clase contiene funcionalidades para el trabajo con los movimientos a los que puede ser sometido un documento.

Tabla 6 Descripción de la clase Dmovimientodocumento.

Nombre: Dmovimientodocumento	
Tipo de clase: Modelo	
Para cada responsabilidad	
Nombre	Descripción
getDocumentos(\$idActa,\$limit=null,\$offset=null,\$query=null)	Permite obtener los documentos asociados a un acta
actualizarDocumento(\$documentData,\$funcionarios)	Permite actualizar los documentos asociados a un acta
eliminar(\$deleteDoc)	Permite eliminar los documentos asociados a un acta

CAPITULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

cancelar()	Permite cancelar los documentos asociados a un acta
aprobarByActa(\$idacta)	Permite aprobar los documentos asociados a un acta

2.7.2 Descripción de algunas operaciones significativas.

Crear acta

Una de las principales operaciones que se realizan con las actas es la de su creación, luego de realizar una serie de pasos lógicos que se detallan a continuación:

1. Se obtiene de parámetro **request** los datos de la nueva acta.
2. Se comprueba si existe un funcionario de entrega, en caso que exista se obtiene sus datos.
3. Se comprueba si existe el id del estado y de ser así se construye un arreglo que estará compuesto por los id del estado, el municipio y la oficina.
4. Se comprueba la existencia del usuario que está adicionando el acta.
5. Si el Usuario es válido y no es nulo ni él ni su id se crea el acta.
6. Se envía a la vista los resultados de la inserción.

The image shows a software window titled "Adicionar acta". The window contains several input fields and dropdown menus. At the top, there are icons for back, forward, and search. The main form area includes:

- Número de acta: (empty text field)
- Fecha de confección: 19/05/2011
- Fase: Elaboración
- Tipo de acta: (dropdown menu)
- Datos de la oficina:
 - Estado: (dropdown menu)
 - Oficina: (dropdown menu)
 - Municipio: (dropdown menu)
- Funcionario de la oficina registral o notarial:
 - Cédula: (text field)
 - Segundo apellido: (text field)
 - Primer nombre: (text field)
 - Teléfono: (text field)
 - Segundo nombre: (text field)
 - Cargo: (dropdown menu)
 - Primer apellido: (text field)
 - Dirección: (text field)

At the bottom of the window, there are three buttons: "Aceptar" (with a green checkmark), "Aplicar" (with a green checkmark), and "Cancelar" (with a red X).

Figura 6 Interfaz de la funcionalidad Crear acta

Crear Documento

Dentro los principales elementos con los que se trabaja se encuentran los Documentos y dentro de las tareas que con este se desarrollan se encuentra su creación, luego de realizar una serie de pasos lógicos que se detallan a continuación:

1. Se obtiene de parámetro **request** los datos del nuevo documento.
2. Se comprueba si los datos que están obtenidos son un arreglo de no ser así se obtiene del **request** las observaciones son incluidas dentro de los parámetros obtenidos en el paso uno y se da formato de arreglo a los datos
3. Se obtiene de parámetro **request** los datos del acta a la que está asociada este documento.

4. Se construye un arreglo compuesto por los datos de los funcionarios que están asociados al acta que fue obtenida en el paso 3.
5. Se crea el movimiento asociando el nuevo documento al acta.

Adicionar documento al acta de registro de documentos

Datos del acta
Número de acta: 3 Estado: Barinas
Fecha de confección: 23/03/2011 Municipio: Barinas
Fase: Elaboración Nombre de la oficina: REGISTRO PÚBLICO DEL MUNICIPIO BARINAS ESTADO BARINAS
Tipo de acta: Registro de

Documento Funcionario acta de apertura Funcionario acta de cierre Observaciones

Tipo de documento: Cantidad documentos:
Protocolo: Cantidad páginas:
Número documento: Tipo Servicio:
Fecha: Clasificación:
Folios útiles: Motivo:

Aceptar Aplicar Cancelar

Figura 7 Interfaz de la funcionalidad Crear Documento.

2.8 Tratamiento de Errores y validación en la entrada de datos

Mediante la interfaz Web se impedirá que el usuario introduzca información optando siempre que sea posible porque este seleccione los datos de una lista previamente elaborada, para esto se contará con cuadros de opción, menú de selección, lo cual facilitará la entrada de datos. La información que requiera ser adicionada por el usuario se validará mediante funciones o expresiones regulares que garanticen que sea válida y que el cuadro de texto no esté vacío si es obligatorio llenarlo. Si hay un error en la información le saldrá al usuario un mensaje en pantalla indicándole el error, también se validarán las opciones correspondientes a la extracción o modificación de datos del servidor de base datos. Si se desea eliminar algún elemento de la BD se preguntará al usuario si está seguro de realizar dicha acción, al igual

que cuando desee modificar alguna información, antes de actualizarla se le preguntará si desea realizarla o no.

2.9 Aplicación de patrones de diseño.

2.9.1 *Patrones Generales de Software para Asignación de Responsabilidades (GRASP).*

Controlador: El patrón **Controlador** define que todas las peticiones web son manejadas por un solo controlador que es la única entrada a la aplicación. Cuando este módulo recibe una solicitud, utiliza el sistema de mapeo de URL para asociar el nombre de una acción y el nombre de un módulo encargado de atender la solicitud del usuario.

Alta cohesión: Con la aplicación de este patrón se logró que la información que se almacena una clase fuera coherente y estuviera (en la medida de lo posible) relacionada con la clase. Este patrón potencia la organización de los elementos dentro del sistema.

Bajo acoplamiento: Este patrón se aplicó para lograr tener las clases lo menos ligadas entre sí. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de las clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases.

2.9.2 *Patrones Gang of Four (GoF).*

Patrones de comportamiento

Observer: El patrón **Observer** es generalizado en Drupal. Cuando una modificación es hecha a un vocabulario en el sistema de taxonomía de Drupal, el hook taxonomy es llamado en todos los módulos que lo implementan. Mediante la aplicación del hook, se han registrado como observadores del objeto vocabulario; cualquier cambio a esto puede entonces actuar como es apropiado. (42)

Chain of Responsibility: El sistema de menús de Drupal sigue el patrón **Chain of Responsibility**. En cada solicitud de la página, el menú del sistema determina si hay un módulo para gestionar la solicitud, si el usuario tiene acceso a los recursos solicitados, y que la función se llama para hacer el trabajo. Para ello,

CAPITULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

el mensaje se pasa a la opción del menú correspondiente a la vía de la solicitud. Si el elemento de menú no puede manejar la petición, se pasa de la cadena. Esto continúa hasta que un módulo se encarga de la petición, un módulo niega el acceso para el usuario, o la cadena se ha agotado.

Command: Muchos de los hooks de Drupal utilizan el patrón **Command** para reducir el número de funciones que son necesarias para la aplicación, pasando por la operación como un parámetro, junto con los argumentos. De hecho, el propio sistema de gancho utiliza este modelo, a fin de que los módulos no tengan que definir cada hook, sino solo los que vaya aplicar. (42)

Patrones estructurales

Decorator: Drupal hace amplio uso del patrón **decorator**. Un ejemplo es el polimorfismo de nodo de objetos, y el uso de hook_nodeapi, que permite a los módulos extender arbitrariamente el comportamiento de todos los nodos.

Patrones creacionales

Singleton: El patrón **Singleton** es aplicado en Drupal para la definición de módulos y los temas. Garantizándose la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia.

2.10 Seguridad del sistema.

A la hora de desarrollar cualquier Sistema de Gestión la seguridad es un tema delicado, este elemento toma mayor relevancia cuando se gestiona información legal como en este caso , este elemento por tanto es prioridad del cliente y por consiguiente del equipo de desarrollo.

Como elemento más importante en lo que a seguridad respecta, se encuentra la asignación de roles a cada uno de los funcionarios autorizados a trabajar con la aplicación y la designación a estos roles de los distintos privilegios que necesitan para desarrollar su trabajo garantizando que cada usuario tenga solamente los permisos estrictamente necesarios. Este proceso fue gestionado haciendo uso del módulo usser que para este fin provee el CMS Drupal.

Otro elemento que se concibió para garantizar la seguridad del sistema es la adecuada actualización de las tecnologías que lo componen y así lo requieren de manera que se garantice la solución de errores o

CAPITULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

posibles vulnerabilidades que presenten, en este caso se encuentra el CMS Drupal y el servidor Apache, este último además fue configurado para restringir el acceso a este solo al personal autorizado. Unido a lo anteriormente expuesto se utilizaron funcionalidades que para este fin poseen algunas de las tecnologías que se utilizan. En este caso tenemos el ORM Doctrine que evita alguno de los ataques más comunes como puede ser el SQL inyección.

El sistema cuenta como se explicó en otros epígrafes con un proceso de validación de los datos que son ingresados a este, tanto a través de la capa de presentación como en la configuración de Drupal que evita ataques frecuentes a este tipo de CMS como son la inserción de código java script no autorizado con el fin de afectar la disponibilidad del sistema o la seguridad de los datos que maneja.

2.11 Conclusiones del Capítulo

Al término de este capítulo, se obtiene el artefacto propuesto. Se definió un estándar de codificación para garantizar la homogeneidad y legibilidad del código y la estructura de datos que mejor resuelva las necesidades del proyecto. Se implementaron las interfaces a partir de los prototipos entregados por los analistas. Se implementó además el negocio que da respuesta a los requisitos especificados por los analistas, así como el acceso a los datos, las validaciones y excepciones.

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

3.1. Introducción.

En este capítulo se relacionan los elementos concernientes a la validación de la solución. Se comienza realizando un análisis del diseño de casos de pruebas de caja blanca aplicado en el sistema. Como parte de las pruebas de caja negra que se le hace al sistema se describe un caso de prueba aplicado a la solución informática y por último se explican los resultados de la aplicación de los casos de pruebas elaborados para la validación de la solución.

3.2. Diseño y Resultados obtenidos de Casos de Pruebas de Caja Blanca aplicados.

A continuación se muestra la prueba del camino básico aplicada al método `mostrarFuncionarios` de la clase `BaseDacta.php` para la cual se siguieron una serie de pasos lógicos.

- Se construye el grafo de flujo a partir del código fuente del método a probar.
- Se determina la complejidad ciclomática (V) del grafo (G). Se puede calcular de tres formas:
 - ✓ $V(G)$ coincide con el número de regiones del grafo de flujo (internas y externas).
 - ✓ $V(G) = A - N + 2$, donde A es el número de aristas del grafo de flujo y N es el número de nodos del mismo.
 - ✓ $V(G) = P + 1$, donde P es el número de nodos predicado contenidos en el grafo de flujo. Un nodo predicado es cada nodo que contiene una condición y se caracteriza gráficamente porque dos o más aristas emergen de él.
- Se construye una secuencia de $V(G)$ caminos linealmente independientes en G .
- Se prepara un caso de prueba por camino hallado en el paso anterior.
 - ✓ Determinar los datos a proporcionar como entrada para ejecutar el camino hallado.
 - ✓ Usando la especificación funcional del método, indicar cuál es el resultado esperado.

Descripción del método `mostrarFuncionarios`.

Permite obtener los funcionarios asociados a un acta.

CAPITULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Tabla 7 Complejidad y posibles caminos del método mostrarFuncionarios

Complejidad Ciclomática	Posibles Caminos
$V(G) = (A - N) + 2$	1. 1-2-3-4-5-6-7-12-5-6-8-9-12-5-6-8-10-11-12-5-13-16
$V(G) = (18 - 16) + 2$	2. 1-2-3-4-5-13-16
$V(G) = 4$	3. 1-2-14-15

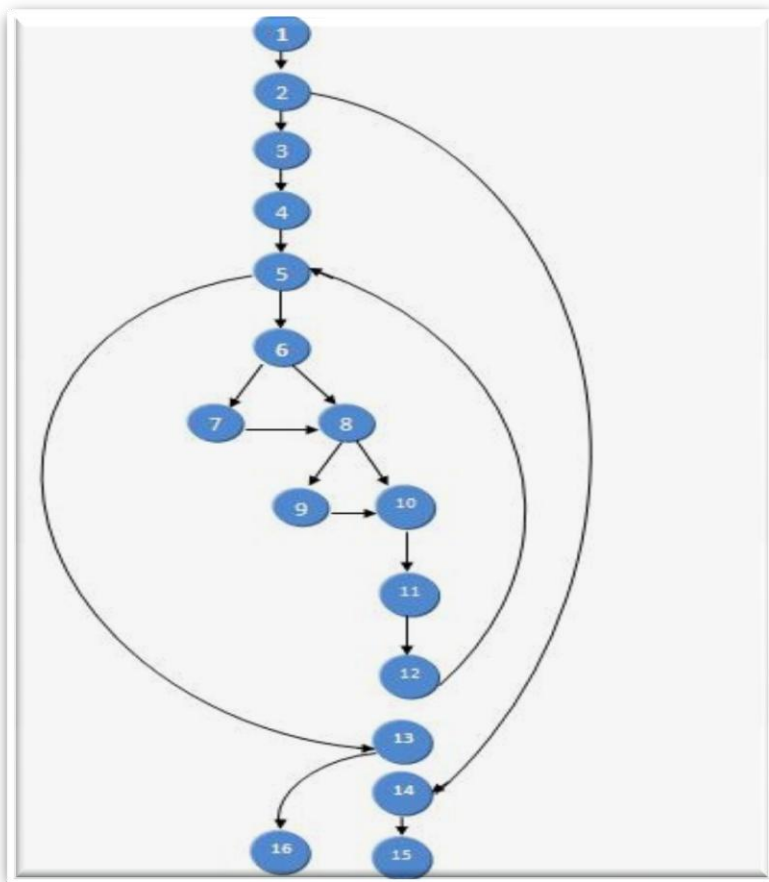


Figura 8 Grafo de flujo del método mostrarFuncionarios

Caso de prueba para el camino básico #1:

Descripción: Se obtienen los datos que son pasados desde la clase controladora correspondiente.

```
$idacta = '1000000000000009';
```

Figura 9 Juegos de datos para el camino básico # 1.

```
public function testMostrarfuncionariocamino1() {  
  
    $resultado= array('idfuncionario'=>"1000000000000010", 'cedula'=>'3123131313', 'nombre'=>'Manuel',  
    'segundonombre'=>NULL, 'primerapellido'=>"Alvarez", 'segundoapellido'=>'Alonso', 'idcargo'=>'Registrador',  
    'direccion'=>"", 'descripcion'=>"", 'centro'=>'true', 'modo'=>'true', 'telefonos'=>"41265666",  
    'identidad'=>"1000000000", 'id'=>'1000000000000010', 'ridfuncionario'=>'1000000000000010',  
    'ridusers'=>'4', 'uid'=>'Manuel');  
  
    $obj = new Dacta();  
    $this->assertEquals($obj->mostrarFuncionario(1000000000000009),$resultado);  
}
```

Figura 10 Test aplicado al camino # 1

Obtiene: Arreglo con los datos de los funcionarios asociados al acta identificada con el id que se le pasó al método.

Caso de prueba para el camino básico #2:

Descripción: Se obtienen los datos que son pasados desde la clase controladora correspondiente.

```
$idacta = 1000000000000004;
```

Figura 11 Juegos de datos para el camino básico # 2.

```
public function testMostrarfuncionariocamino2() {  
  
    $resultado = array(NULL);  
    $obj = new Dacta();  
    $this->assertEquals($obj->mostrarFuncionario(1000000000000004),$resultado);  
}
```

Figura 12 Test aplicado al camino # 2

CAPITULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Obtiene: Arreglo vacío pues el acta identificada con el id que se le pasó al método no posee funcionarios asociados.

Caso de prueba para el camino básico #3:

Descripción: Se obtienen los datos que son pasados desde la clase controladora correspondiente.

```
$idacta = '1000000';
```

Figura 13 Juegos de datos para el camino básico # 3.

```
public function testMostrarfuncionariocamino3() {  
    $resultado = 'Error al obtener los datos.';  
    $obj = new Dacta();  
    $this->assertEquals($obj->mostrarFuncionario(100000000), $resultado);  
}
```

Figura 14 Test aplicado al camino # 3

Obtiene: Mensaje: Error al obtener los datos.



Figura 15 Resultados de los test aplicados al método motrarFuncionario

Como resultado de las pruebas de caja blanca aplicada se pudo garantizar la ejecución de todos los caminos independientes en al menos una ocasión de manera que no exista código aislado o que no se ejecute. Se ejercitan todas las decisiones lógicas detectando errores que pudieran existir tanto en su vertiente verdadera como en la falsa. Se analizó la complejidad ciclomática de los métodos comprobando

CAPITULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

la cantidad de caminos independientes que existen en los mismos y por consiguiente definió el límite superior de la cantidad de pruebas a ejecutar.

3.3. Diseño de Casos de Prueba de Caja Negra aplicados

Durante la validación de la solución propuesta se utilizó también el método de Caja Negra encaminado a garantizar el cumplimiento de los requisitos funcionales del sistema. Para el desarrollo de estas pruebas se utilizan como material inicial casos de pruebas que describen las funcionalidades a probar en cada caso. A continuación se describe uno de los 23 casos de pruebas que se le aplicaron a la aplicación informática. Este caso de prueba tiene como objetivo comprobar la funcionalidad Gestionar producto, se escogió este caso de prueba por ser los productos junto a las actas y los documentos los elementos más significativos del sistema.

3.3.1. Caso de Prueba Gestionar Producto

Tabla 8 Secciones a aprobar en el Caso de Uso.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: Adicionar.	EC 1.1: Adicionar producto exitosamente.	Se adiciona un nuevo producto insertando datos válidos.
	EC 1.2: Seleccionar la opción Cancelar.	Mediante este escenario se cancela la creación de un nuevo grupo, subgrupo o producto.
	EC 1.3: Existen datos incompletos del producto.	Al insertar datos pertenecientes a un nuevo grupo, al menos uno se encuentra en blanco.

CAPITULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

	EC 1.4: Existen datos incorrectos del producto.	Al insertar datos pertenecientes a un subgrupo o producto, existen campos incorrectos.
SC 2: Eliminar producto.	EC 2.1: Eliminar uno o varios productos exitosamente.	Este escenario se encarga de eliminar uno o varios productos seleccionados.
	EC 2.2: Seleccionar la opción Cancelar.	Mediante este escenario se cancela la eliminación de uno o varios productos.
SC 3: Modificar producto.	EC 3.1 Modificar un producto exitosamente.	Se modifica el producto seleccionado exitosamente.
	EC 3.2: Seleccionar la opción Cancelar.	Mediante este escenario se cancela la modificación del producto.
	EC 3.3: Existen datos incompletos.	Al insertar datos pertenecientes a un producto, al menos uno se encuentra en blanco.
	EC 3.4: Existen datos incorrectos.	Al insertar datos pertenecientes a un producto, existen campos incorrectos.

Tabla 9 Descripción de las variables

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
[1]	Código	Campo de texto	No	Números.
[2]	Denominación	Campo de texto	No	Acepta letras, números y todo tipo de caracteres especiales.
[3]	Descripción	Campo de texto	Si	Acepta letras, números y todo tipo de caracteres especiales.

CAPITULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

[4]	Es Producto	Campo de selección	Si	Booleano.
[5]	UM(Unidad de medida)	Lista desplegable	Si	Letras y números.

Tabla 10 Matriz de datos adicional producto.

Escenario	Código	Denominación	Descripción	Es Producto	UM(Unidad de medida)	Respuesta Esperada	Resultado de la Prueba	Flujo Central
EC 1.1: Adicionar producto exitosamente.	V 20	V Hilo	V Producto para coser los tomos.	V	V M	El sistema adiciona los datos del nuevo producto y muestra un mensaje de confirmación.	Satisfactorio	<ol style="list-style-type: none"> 1. Seleccionar Nomenclador. 2. Seleccionar Producto. 3. Seleccionar la carpeta Grupos. 4. Selecciona el grupo perteneciente a materiales para coser 5. Seleccionar Adicionar. 6. Introducir valor en los campos. 7. Seleccionar Aceptar o Aplicar.

CAPITULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

EC 1.2: Seleccionar la opción Cancelar.	N/A	N/A	N/A	N/A	N/A	El sistema cancela las operaciones realizadas.	Satisfactorio	<ol style="list-style-type: none"> 1. Seleccionar Nomenclador. 2. Seleccionar Producto. 3. Seleccionar la carpeta Grupos. 4. Seleccionar Adicionar. 5. Seleccionar Cancelar.
EC 1.3: Existen datos incompletos del producto.	I	V Hilo	V Producto para coser los tomos	V	V M	El sistema señala el campo incorrecto y muestra un mensaje de error.	Satisfactorio	<ol style="list-style-type: none"> 1. Seleccionar Nomenclador. 2. Seleccionar Producto. 3. Seleccionar la carpeta Grupos. 4. Seleccionar Adicionar. 5. Introducir valores en los campos. 6. Seleccionar Aceptar o Aplicar.
	20	I	V Producto para coser los tomos	V	V M			

CAPITULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Tabla 11 Matriz de datos eliminar producto

Escenario	Código	Denominación	Descripción	Es Producto	UM(Unidad de medida)	Respuesta Esperada	Resultado de la Prueba	Flujo Central
EC 2.1: Seleccionar la opción Eliminar.	N/A	N/A	N/A			El sistema elimina uno o varios productos y muestra un mensaje de confirmación.	Satisfactorio	<ol style="list-style-type: none"> 1. Seleccionar Nomenclador. 2. Seleccionar Producto. 3. Dar doble clic encima de una carpeta. 4. Seleccionar uno o varios productos. 5. Seleccionar Eliminar. 6. Seleccionar Aceptar.
EC 2.2: Seleccionar la opción Cancelar.	N/A	N/A	N/A	N/A	N/A	El sistema cancela las operaciones realizadas.	Satisfactorio	<ol style="list-style-type: none"> 1. Seleccionar Nomenclador. 2. Seleccionar Producto. 3. Dar doble clic encima de una carpeta. 4. Seleccionar una o varias filas del grid. 5. Seleccionar Eliminar. 6. Seleccionar Cancelar.

CAPITULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Tabla 12 Matriz de datos modificar producto.

Escenario	Código	Denominación	Descripción	Es Producto	UM(Unidad de medida)	Respuesta Esperada	Resultado de la Prueba	Flujo Central
EC 3.1: Modificar producto exitosamente.	V 22	V Televisores	V Subgrupo al cual pertenece el Televisor LG de 32 pulgadas	V	V Unidad	El sistema adiciona los datos del nuevo producto y muestra un mensaje de confirmación.	Satisfactorio	<ol style="list-style-type: none"> 1. Seleccionar Nomenclador. 2. Seleccionar Producto. 3. Dar doble clic encima de una carpeta. 4. Seleccionar un producto 5. Seleccionar Modificar producto. 6. Introducir valor en los campos. 7. Seleccionar Aceptar.
	V 48	V Televisor LG	V Televisor LG de 32 pulgadas	V	V Unidad			

CAPITULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Escenario	Código	Denominación	Descripción	Es Producto	UM(Unidad de medida)	Respuesta Esperada	Resultado de la Prueba	Flujo Central
EC 3.2: Seleccionar la opción Cancelar.	N/A	N/A	N/A		N/A	El sistema cancela las operaciones realizadas.	Satisfactorio	<ol style="list-style-type: none"> 1. Seleccionar Nomenclador. 2. Seleccionar Producto. 3. Dar doble clic encima de una carpeta. 4. Seleccionar un producto 5. Seleccionar Modificar producto. 6. Selecciona Cancelar

CAPITULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Escenario	Código	Denominación	Descripción	Es Producto	UM(Unidad de medida)	Respuesta Esperada	Resultado de la Prueba	Flujo Central
EC 3.3: Existen datos incompletos.	I	V	V	V	V	El sistema señala el campo incorrecto y deshabilita el botón de Aceptar.	Satisfactorio	<ol style="list-style-type: none"> 1. Seleccionar Nomenclador. 2. Seleccionar Producto. 3. Dar doble clic encima de una carpeta. 4. Seleccionar una fila del grid. 5. Seleccionar Modificar producto. 6. Introducir valor en los campos.
	ab	Televisor LG	Televisor LG de 32 pulgadas		Unida			
	V	I	V	V	V			
	48		V	V	V			
		V	V	V	I			
	48	Televisor LG	Televisor LG de 32 pulgadas		m			

CAPITULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Escenario	Código	Denominación	Descripción	Es Producto	UM(Unidad de medida)	Respuesta Esperada	Resultado de la Prueba	Flujo Central
EC 3.4: Existen datos incorrectos.	V 48	V Televisores	i	V		El sistema señala el campo incorrecto y deshabilita el botón de Aceptar.	Satisfactorio	<ol style="list-style-type: none"> 1. Seleccionar Nomenclador. 2. Seleccionar Producto. 3. Dar doble clic encima de una carpeta. 4. Seleccionar una fila del grid. 5. Seleccionar Modificar producto. 6. Introducir valor en los campos.

3.4. Resultado de las Pruebas de Caja Negra.

Como resultado de la aplicación, de la totalidad de los caso de pruebas fueron detectados 21 no conformidades, de manera que podemos calificar los casos de pruebas desarrollado de acertado pues cumplieron con el objetivo de encontrar deficiencias en los ámbitos donde fueron aplicados. El sistema como resultado de estas no conformidades y con el objetivo de resolverlas iteró hasta obtener la liberación por parte de CALISOFT, entidad encargada de la ejecución de las pruebas y de la liberación del producto. CALISOFT como resultado del proceso de liberación emitió un certificado acreditativo que se adjunta en el Anexo 1. Como resultado de la primera iteración fueron encontradas 17 no conformidades de las cuales 10 fueron funcionales y 11 de validación.

CAPITULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

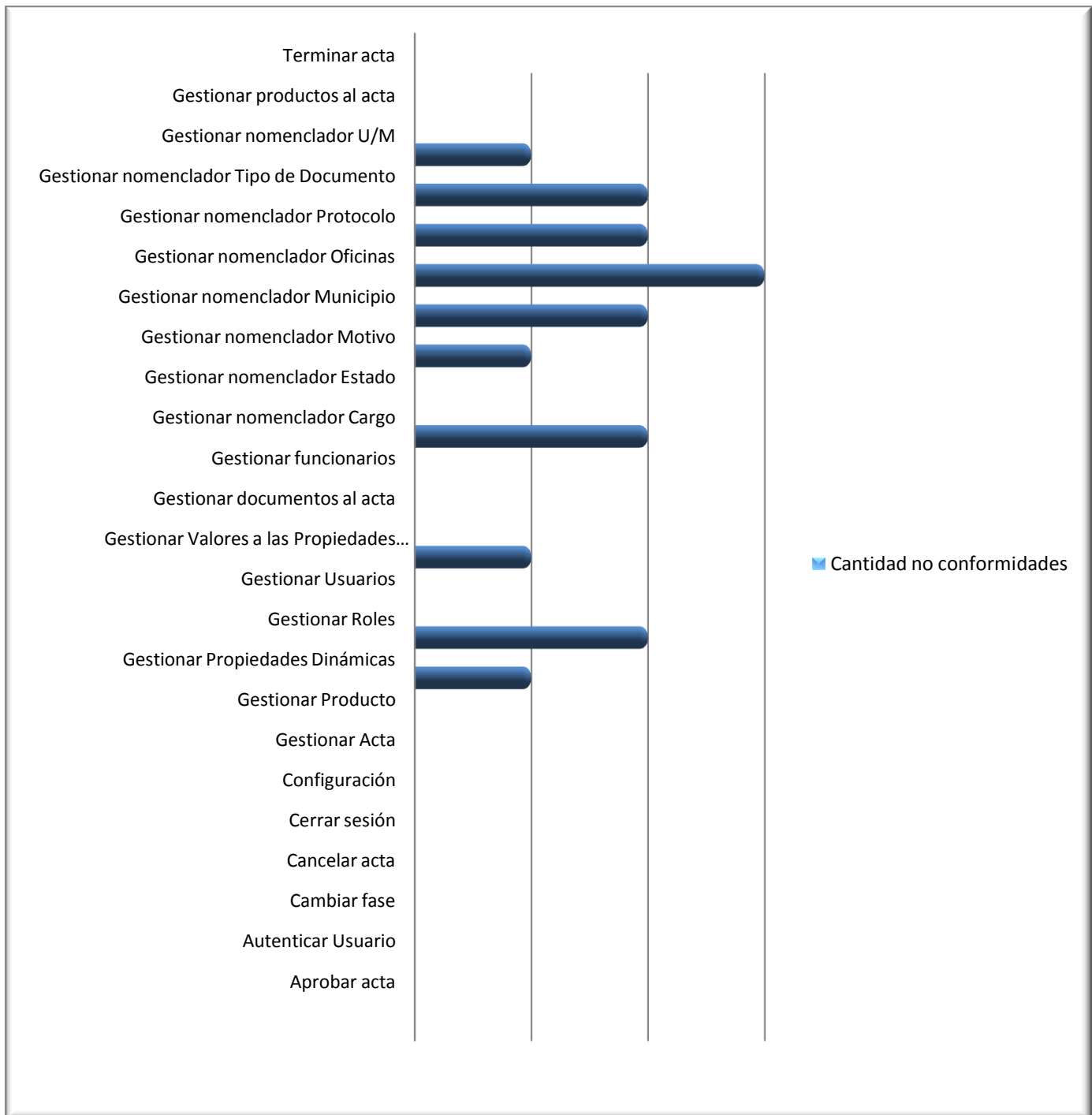


Figura 16 Cantidad de no conformidades por caso de prueba en la primera iteración del sistema

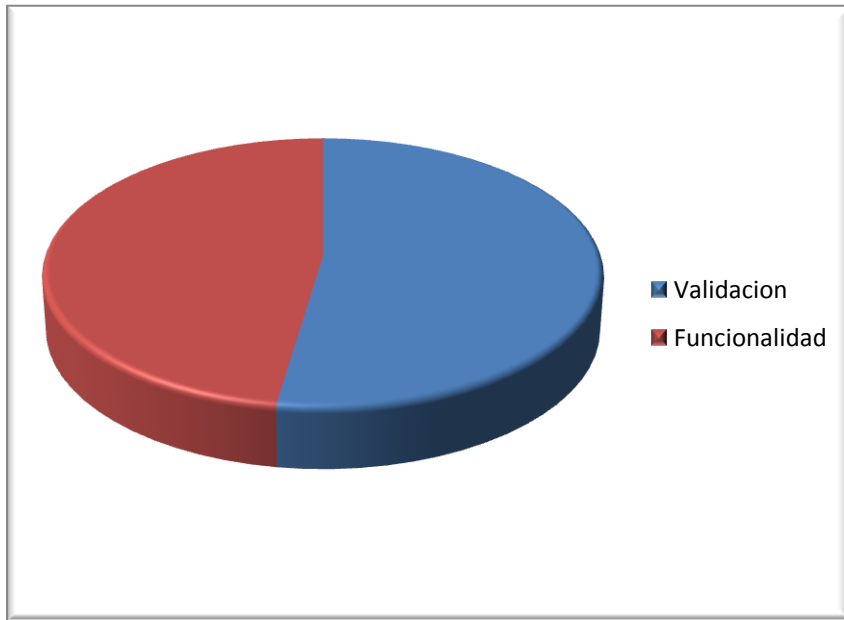


Figura 17 Distribución de las no conformidades en la primera iteración

En el caso de la 2da iteración se obtuvieron 4 no conformidades de las cuales 2 correspondieron a la validación de datos y 2 funcionalidades.

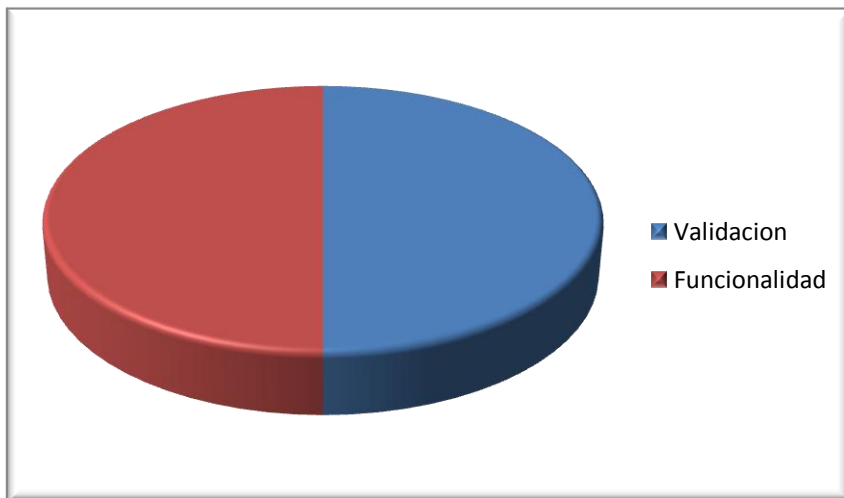


Figura 18 Distribución de las no conformidades en la segunda iteración.

CAPITULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

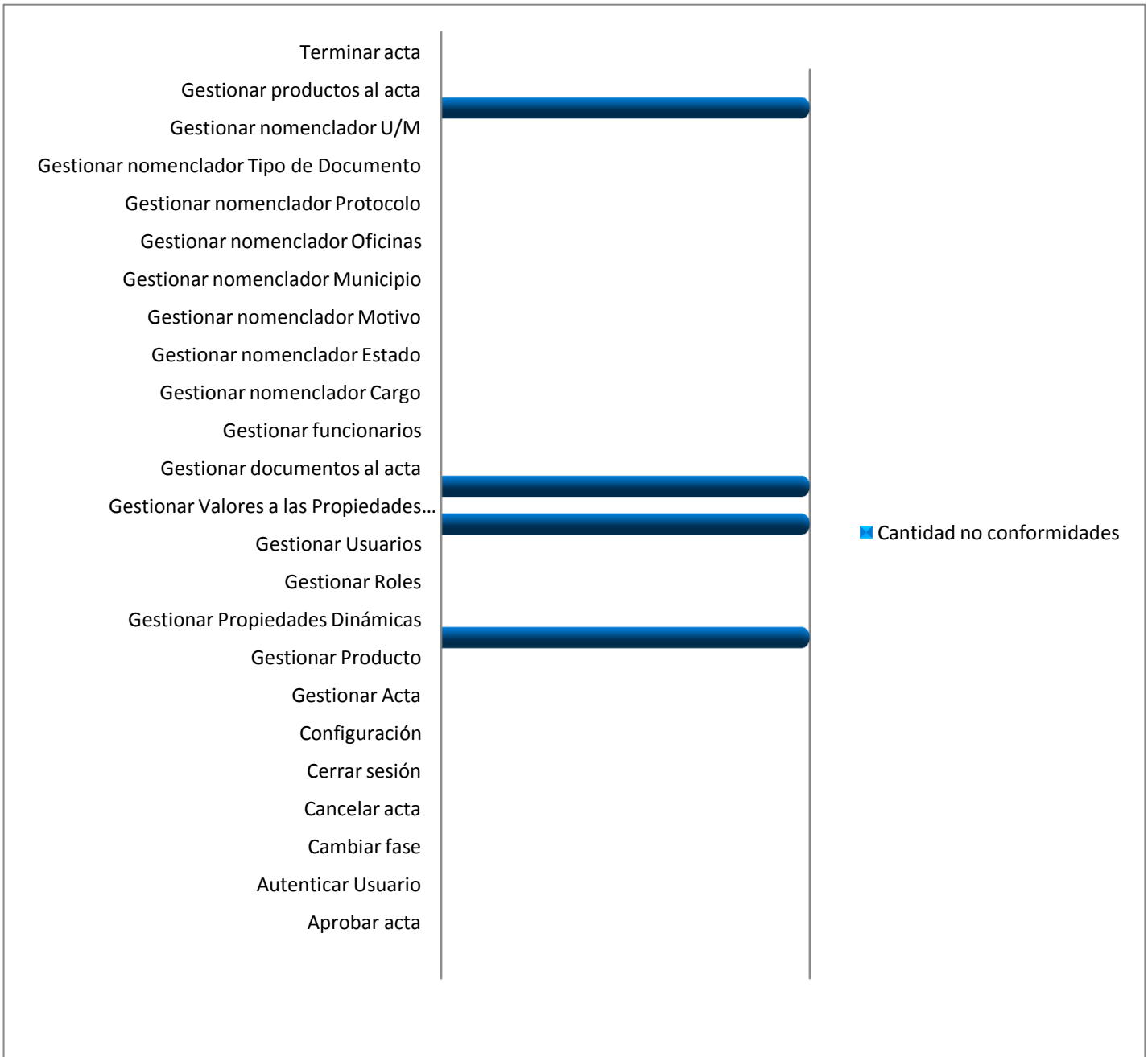


Figura 19 Cantidad de no conformidades por caso de prueba en la segunda iteración del sistema.

3.5. Conclusiones del Capitulo.

En este capítulo se analizaron elementos concernientes a la aplicación de las pruebas de software, tanto de caja blanca como de caja negra, describiendo algunos escenarios de pruebas a los que fue sometido el sistema y analizando los resultados que arrojaron las pruebas con la aplicación de la totalidad de los casos de pruebas, concluyendo que estos fueron satisfactorios pues permitieron la detección de no conformidades que posteriormente fueron resueltas, garantizando el correcto funcionamiento del sistema.

CONCLUSIONES

- ❖ La metodología de desarrollo RUP y la arquitectura Modelo-Vista-Controlador fueron las más adecuadas para el desarrollo del sistema SICEN y se cumplió con las exigencias en cuanto a las tecnologías y herramientas a usar que definió el cliente.
- ❖ Se dotó a las oficinas móviles de encuadernación del SAREN de un sistema para la gestión del proceso de encuadernación y el control de la información generada por estas, la definición de estrategias y toma de decisiones.
- ❖ El sistema desarrollado fue objeto de pruebas de caja negra, en 23 escenarios que abarcan las funcionalidades que se esperan este desarrollo y de caja blanca en sus métodos comprobándose que cumple con las especificaciones de requisitos hechas por el cliente previamente.

RECOMENDACIONES

- ❖ Realizar pruebas de eficiencia al sistema en una oficina de encuadernación.
- ❖ Desarrollar una nueva versión del sistema que contemple la existencia de oficinas fijas con un mayor número de personal, pues estas instituciones por la dinámica que hoy sigue el SAREN deben crecer en número de funcionarios con sus correspondientes roles.

BIBLIOGRAFÍA

1. **AG, SAP.** SAP GLOBAL. [En línea] [Citado el: 7 de Febrero de 2011.] <http://www.sap.com/index.epx>.
2. **Condor.** ERP Softwarer. [En línea] [Citado el: 7 de Febrero de 2011.] <http://www.erpsoftwaredownload.com/erpsoftware/condor-enterprise.html>.
3. **Rodas.** Rodas XXI. [En línea] [Citado el: 7 de Febrero de 2011.] <http://www.rodasxxi.cu..>
4. **Menéndez-Barzanallana, Rafael.** Universidad de Murcia . [Online] <http://www.um.es/docencia/barzana/LAGP/lagp2.html>.
5. **Batista, Ing. Lester Omar Bello.** *Arquitectura de softwarer Registros y Notarias fase III.* 2010.
6. **Camones, Rafael Reyna.** Scribd. [Online] 2011. <http://es.scribd.com/doc/3062020/Capitulo-I-HERRAMIENTAS-CASE>.
7. **Alonso, Evelyn Menéndez.** Plusformacion.com. [Online] 2011. http://www.plusformacion.com/Recursos/r/Herramientas-CASE-para-proceso-desarrollo-Software?quicktabs_ofertas_relacionadas_quicktab=1#resumena.
8. **Cornejo, Ing José Enrique González.** Doclrs. [Online] 2011. <http://www.docirs.cl/uml.htm>.
9. **Garcés, Anderson Xavier Terres.** *Diseño y construccion de un robot de batalla controlado mediante bluetooth.* 2010.
10. **Nacional, Universidad Tecnológica.** Universidad Tecnológica Nacional. [Online] <http://ft.utn.edu.ar/sistemas/paradigmas/lenguajes.htm>.
11. **Group, PHP.** PHP. [Online] 2011. <http://www.php.net/manual/es/intro-whatism.php>.
12. **Lanzillotta, Analía.** MASTERMAGAZINE. [Online] 2004. [Cited: Febrero 7, 2011.] <http://www.mastermagazine.info/termino/5286.php>.
13. **Kemedy, Chuck Musciano y Bill.** *HTML La Guía Completa.* Mexico : Litografica Ingramex, 1999.
14. **Santos, Víctor M. Rivas.** GENURA. [Online] 2010. [Cited: Enero 23, 2011.] http://geneura.ugr.es/~victor/cursillos/javascript/js_intro.html.
15. **Group, JSON.** JSON. [Online] 2011. <http://www.json.org/json-es.html>.
16. **Delgado, José.** Ubuntulife. [Online] 2011. <http://ubuntulife.wordpress.com/2008/09/06/introduccion-a-json/>.
17. **Pergaminovirtua.** Pergaminovirtua. [Online] 2011.

<http://www.pergaminovirtual.com.ar/definicion/CSS.html>.

18. **Molpeceres, Alberto**. Abcdatos. [Online] 2011. <http://www.abcdatos.com/tutoriales/tutorial/l6455.html>.
19. **González, Carlos D.** Usabilidad Web. [Online] 2011. <http://www.usabilidadweb.com.ar/postgre.php>.
20. **Guia-Ubunto**. Guia-Ubunto. [Online] 2011. http://guia-ubuntu.org/index.php?title=PgAdmin_III.
21. **Maidana], Esteban**. FPDF library. [Online] <http://www.fpdf.org/>.
22. **Masadelante.com**. Mas Adelante.com. [Online] 2011. <http://www.masadelante.com/faqs/servidor-web>.
23. **Foundation, The Apache Software**. Apache. [Online] 2011. <http://httpd.apache.org/>.
24. **Sevilla, Universidad de**. Taller de accesibilidad web. [Online] 2011. <http://moodle.forpas.us.es/mod/resource/view.php?id=1349>.
25. **Pajón, Javier**. Javinet. [Online] 2011. <http://www.javinet.com.ar/firefox.htm>.
26. **Europe, Mozilla**. Mozilla Europe. [Online] [Cited: Diciembre 6, 2010.] [http://www.mozilla-europe.org/es/firefox/features/..](http://www.mozilla-europe.org/es/firefox/features/)
27. **Bustamante, Miranda Patricio**. wordpress.com. [Online] 2006. [http://patob2000.wordpress.com/2006/09/28/funcionamiento-general-de-las-herramientas-colaborativas/..](http://patob2000.wordpress.com/2006/09/28/funcionamiento-general-de-las-herramientas-colaborativas/)
28. **Crespo, C.** eprints. [Online] 2011. <http://eprints.ucm.es/11295/1/SistemasInform%C3%A1ticos.pdf>.
29. **Sánchez, Jordi**. Jordisan.net. [Online] 2011. <http://jordisan.net/blog/2006/que-es-un-framework>.
30. *Revista de Software Libre Atix*. **Proenza, Yuniel Eliades Arias**. 2009, Vol. VIII.
31. **White, Steve**. *Manual del Usuario Joomla! 1.0.x*. 2006.
32. **Gersbach, Martin**. Drupal Hispano. [Online] 2011. <http://drupal.org.es/drupal>.
33. **DEPARTAMENTO DE ELECTRÓNICA, SISTEMAS E INFORMÁTICA**. Coordinación docente de lenguajes de programación . [Online] <http://progra.iteso.mx/estandares/estandar%20codificacion%20c++/estandardcodificacion.pdf>.
34. **Group, The PHP**. PHP. [Online] <http://php.net/manual/es/language.exceptions.php>.
35. **Gracia, Joaquin**. IngenieroSoftware. [Online] <http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>.
36. **Valls, Ignasi Pérez**. JavaDabbaDoo.org. [Online] 2009. <http://www.javadabbadoo.org/cursos/infosintesis.net/javase/paqawt/selectorcolores/paso03patrones.html>.
37. **Grosso, Andres**. Prácticas de Software. [Online] <http://www.practicadesoftware.com.ar/2011/03/patrones-grasp/>.
38. **Sevilla, Departamento de Lenguajes y Sistemas Informáticos Universidad de**. Departamento de

- Lenguajes y Sistemas Informáticos Universidad de Sevilla. [Online]
<http://www.lsi.us.es/docencia/get.php?id=1130>.
39. **Cauca, Facultad de Ingeniería Electrónica y Telecomunicaciones Universidad de.** Comunidad Virtual Universidad de Cauca . [Online]
http://pis.unicauca.edu.co/moodle/file.php/291/Patron_Disenio_MVC.pdf.
40. **Pressman, Roger.** *Ingeniería de Software Un Enfoque Práctico*. 2000.
41. **Chaves, Michael Arias.** Revista de la Universidad de Costa Rica. [Online] 2011.
http://www.latindex.ucr.ac.cr/intersedes10/10-art_11.pdf.
42. **Softwarer, Departamento Central Ingeniería de.** *Rational RequisitePro*. 2009.
43. **Drupal.** Drupal. [Online] <http://drupal.org/coding-standards>.
44. **Martín, Liber Matos.** Se soluciona programando. [Online] <http://jitcode.blogspot.com/2010/06/poo-y-patrones-de-diseno-en-drupal.html>.

GLOSARIO DE TÉRMINOS

[A]

Acta Un acta es una relación escrita de lo sucedido, tratado o acordado en una junta, certificación, testimonio, asiento o constancia oficial de un hecho.

Actores (ISW): El término actor significa el rol que algo o alguien juega cuando interactúa con el negocio o sistema.

Algoritmo: Es un conjunto finito de instrucciones o pasos que sirven para ejecutar una tarea o resolver un problema.

Apache Http Server: Apache es programa de servidor HTTP Web de código abierto. Fue desarrollado en 1995 y actualmente es uno de los servidores web más utilizados en la red.

[C]

C : Es un lenguaje de programación creado en 1972 por Dennis M. Ritchie en los Laboratorios Bell como evolución del anterior lenguaje B, a su vez basado en BCPL. Al igual que B, es un lenguaje orientado a la implementación de Sistemas Operativos, concretamente Unix. C es apreciado por la eficiencia del código que produce y es el lenguaje de programación más popular para crear software de sistemas, aunque también se utiliza para crear aplicaciones.

C++: Es un lenguaje de programación diseñado a mediados de los años 1980 por Bjarne Stroustrup. La intención de su creación fue el extender al exitoso lenguaje de programación C con mecanismos que permitan la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido.

Casos de uso: Representa a un proceso, por lo que se corresponde con una secuencia de acciones que producen un resultado observable para ciertos actores. Desde la perspectiva de un actor individual, define un flujo de trabajo completo que produce resultados deseables.

CITMATEL: Empresa de Tecnologías de la Información y Servicios Telemáticos Avanzados de Cuba. Contribuye con el desarrollo científico y tecnológico de la sociedad cubana con relevante liderazgo en la introducción de tecnologías de primera línea en el país. Como proveedores de Internet en Cuba, son

pioneros en el desarrollo de redes de transmisión de datos, y poseen la red CENIAInternet con alcance internacional que se extiende a todo el territorio nacional.

Componente: Un componente es una parte no trivial, casi independiente, y reemplazable de un subsistema que llena claramente una funcionalidad dentro de un contexto en una arquitectura bien definida. Un componente se conforma y provee la realización física por medio de un conjunto de interfaces.

CSS: Es el acrónimo de (Cascading Style Sheets,) es un formato usado en las páginas web para dar estilo a documentos HTML y XML, separando el contenido de la presentación.

[D]

Doctrine: Es un ORM para PHP 5.2.3 y posterior. Además de todas las ventajas que conlleva un ORM, uno de sus puntos fuertes es su lenguaje DQL (Doctrine Query Language). Esta desarrolla bajo la licencia GPL.

[E]

Encapsulamiento: Significa reunir a todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción. Esto permite aumentar la cohesión de los componentes del sistema. Algunos autores confunden este concepto con el principio de ocultación, principalmente porque se suelen emplear conjuntamente.

Encuadernación: Se llama **encuadernación** a la acción de coser o pegar varios pliegos o cuadernos de texto y ponerles cubiertas. Las encuadernaciones tienen por objeto procurar al libro tres ventajas: su conservación, su fácil manejo y su presentación artística.

ERP: Los sistemas de **planificación de recursos empresariales**, o **ERP** (por sus siglas en inglés, *Enterprise resource planning*) son sistemas de información gerenciales que integran y manejan muchos de los negocios asociados con las operaciones de producción y de los aspectos de distribución de una compañía en la producción de bienes o servicios.

[G]

GNU/Linux: Sistema Operativo. El sistema conjunto (herramientas GNU y núcleo Linux) forma GNU/Linux.

GPL: La licencia pública general de GNU o más conocida por su nombre en inglés GNU General Public License o simplemente su acrónimo del inglés GNU GPL, es una licencia creada por la Free Software Foundation a mediados de los 80, y está orientada principalmente a proteger la libre distribución,

modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

[H]

Herencia: Las clases no están aisladas, sino que se relacionan entre sí, formando una jerarquía de clasificación. Los objetos heredan las propiedades y el comportamiento de todas las clases a las que pertenecen. La herencia organiza y facilita el polimorfismo y el encapsulamiento permitiendo a los objetos ser definidos y creados como tipos especializados de objetos preexistentes

Hooks: Son disparadores con los que cuenta Drupa, que se activan cuando algo que ellas escuchan ocurre, estas tiene una estricta sintaxis para ser reconocidas por el sistema. Además contienen funciones de ayuda, para uso interno o para exportar sus funcionalidades a otros módulos.

HTML: Son las siglas de HyperText Markup Language (*Lenguaje de Marcado de Hipertexto*), es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML

[I]

Informatización s. f. Implantación de medios informáticos para el desarrollo de una actividad o trabajo.

Ingeniería de Requerimientos: Es un enfoque sistémico para recolectar, organizar y documentar los requerimientos del sistema; es también el proceso que establece y mantiene acuerdos sobre los cambios de requerimientos, entre los clientes y el equipo del proyecto.

[J]

JavaScript: Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente (client-side)

JSON: Es el acrónimo de *JavaScript Object Notation*, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

[M]

MySQL: Es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta

licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

[N]

NetBeans: Se refiere a una plataforma para el desarrollo de aplicaciones de escritorio y web.

Nomenclador: Catálogo de reglas estipuladas a nivel central o de entidad.

[O]

Objeto: Entidad provista de un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidad. Corresponden a los objetos reales del mundo que nos rodea, o a objetos internos del sistema.

On-line: Se refiere a estar conectado a internet.

Open Source: El software OpenSource se define por la licencia que lo acompaña, que garantiza a cualquier persona el derecho de usar, modificar y redistribuir el código libremente.

Oracle®: Sistema de administración de base de datos (ORDBMS por el acrónimo en inglés de Relational Data Base Management System), fabricado por Oracle Corporation. Soporta acceso por SQL y por lenguajes de programación.

ORM (Object Relational Mapper): Es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional.

[P]

Pascal: Es un lenguaje de programación desarrollado por el profesor suizo Niklaus Wirth entre los años 1968 y 1969 y publicado en 1970. Su objetivo era crear un lenguaje que facilitara el aprendizaje de programación a sus alumnos, utilizando la programación estructurada y estructuración de datos. Sin embargo con el tiempo su utilización excedió el ámbito académico para convertirse en una herramienta para la creación de aplicaciones de todo tipo.

Perl: Es un lenguaje de programación diseñado por Larry Wall en 1987. Perl toma características del lenguaje C, del lenguaje interpretado shell (sh), AWK, sed, Lisp y, en un grado inferior, de muchos otros lenguajes de programación.

PHP: Es un acrónimo de *Hypertext Preprocessor* es un lenguaje de programación interpretado, diseñado

originalmente para la creación de páginas web dinámicas. Es usado para la interpretación del lado del servidor (*server-side scripting*).

PHPmyadmin: Es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas webs, utilizando Internet. Actualmente puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 50 idiomas. Se encuentra disponible bajo la licencia GPL.

Polimorfismo: Comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre, al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando.

PostgreSQL™: Sistema de gestión de bases de datos objeto-relacional (ORDBMS) basado en el proyecto POSTGRES, de la universidad de Berkeley. Es una derivación libre (OpenSource) de este proyecto, y utiliza el lenguaje SQL92/SQL99.

Programación orientada a objetos (POO): Es un paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones y programas de computadora. Está basado en varias técnicas, incluyendo herencia, modularidad, polimorfismo y encapsulamiento.

Python: Es un lenguaje de programación de alto nivel cuya filosofía hace hincapié en una sintaxis muy limpia y que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico, es fuertemente tipado y multiplataforma.

[R]

RUP: Es el **Proceso Racional Unificado** (*Rational Unified Process* en inglés, habitualmente resumido como **RUP**) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Ruby es un lenguaje de programación interpretado, reflexivo y orientado a objetos, creado por el programador japonés Yukihiro "Matz" Matsumoto, quien comenzó a trabajar en Ruby en 1993, y lo presentó públicamente en 1995. Combina una sintaxis inspirada en Python y Perl con características de programación orientada a objetos similares a Smalltalk.

Requerimientos: Una condición o capacidad que debe estar presente en un sistema o componentes de

sistema para satisfacer un contrato, estándar, especificación u otro documento formal.

[S]

Sistema de software: Es un programa o aplicación de software que permite a los usuarios el control o realización de varias tareas y que hacen que el trabajo sea más cómodo, rápido y eficiente.

SQL: El Lenguaje de consulta estructurado (Structured Query Lenguaje) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas.

SQLite: Es una librería escrita en lenguaje C que implementa un manejador de base de datos SQL embebido.

Subsistema: Son las partes que forman un sistema. Cada sistema está compuesto de “subsistemas”, los cuales a su vez son parte de otros subsistemas; cada subsistema es delineado por sus límites.

Switch: o conmutador, interconecta dos o más partes de una red, funcionando como un puente que transmite datos de un segmento a otro. Su empleo es muy común cuando existe el propósito de conectar múltiples redes entre sí para que funcionen como una sola. Un conmutador suele mejorar el rendimiento y seguridad de una red de área local.

[T]

TCP/IP: Sistema de protocolos establece las bases de la mayor parte del universo de Internet. El TCP se encarga en fragmentar la información en pequeños paquetes para después volverlos a juntar en un destino final. El IP tiene como función el checar que estos paquetes vayan dirigidos correctamente hacia un mismo destino.

[U]

UML: Lenguaje Unificado de Modelado, es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software.

[Y]

YUI (Yahoo! User Interface): Serie de bibliotecas escritas en JavaScript, para la construcción de aplicaciones interactivas (RIA). Liberadas bajo licencia BSD por parte de la compañía Yahoo. Dichas bibliotecas son utilizadas para el desarrollo web específicamente para ser usadas como la programación de aplicaciones de escritorio, con componentes vistosos y personalizables y con una amplia implementación con AJAX.

ANEXO



Acta de Liberación de Productos Software

Fecha de liberación: 28/01/2011

Emitida a favor de: Sistema Centro de Encuademación (SICEN)

1. Datos del producto

Artefacto	Versión	Estado final	Cantidad liberaciones	Tipos de pruebas realizadas
Aplicación SICEN	1.0	0	3	Funcionalidad

Validar
de reconocida
Dignidad agente
Yuan Trujillo Reyna
Cm. 30122-11
Rector: Oscar
Oficina de CALISOFT
Calle 100

Yuan Trujillo Reyna
Responsable Calisoft

Arlampyda Dorado Risco
Responsable Proyecto