



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 10

Sistema Básico para Archivos Históricos

Trabajo de diploma para optar por el título de Ingeniería en Informática

Elaborado por: Rafael Barrios Leyva

Rismaris Calvo Suarez

Tutor: Ing. Adrian Cid Almaguer

Ing. Annia Surós Vicente

Ciudad de la Habana, Cuba.

Julio, 2008

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Rafael Barrios Leyva

Firma del Autor

Rismaris Calvo Suarez

Firma del Autor

Adrian Cid Almaguer

Firma del Tutor

Annia Surós Vicente

Firma del Tutor

AGRADECIMIENTOS

A todas las personas que de una forma u otra contribuyeron a la realización de este trabajo, a nuestros compañeros de estudios, a nuestros profesores y tutores, y familiares.

De corazón gracias.

DEDICATORIA

Rafael:

Dedico este trabajo principalmente a mi hijo y a mis padres, espero que queden orgullosos de verme convertido en un profesional después de muchos años de estudio, también le agradezco a todo aquel que ha brindado su ayuda, a mis tíos que siempre estuvieron dándome su mano, a mi esposa que siempre me dio esa bella compañía y un apoyo incondicional, a mis amigos y a mis tutores que siempre me apoyaron.

Rismaris:

A toda mi familia, amigos y tutores Que de una forma u otra me ayudaron Con la realización de este trabajo, Pero en especial se lo dedico a mis Dos abuelos que siempre me apoyaron Y que en este momento no pueden estar Entre nosotros Raúl Céspedes y Ángel Nemesio.

*“Da el primer paso con fe, no tienes que ver todas
las escaleras, tan solo da el primer paso.”*

Martin Luther King

Resumen.

En este Trabajo de Diploma se desarrollará la idea de cómo crear un sistema que resuelva los problemas existentes con la preservación y difusión de documentos históricos, que actualmente en muchos lugares se realiza de forma manual. Para ello se definirá el Modelo dominio para enmarcar la situación problemática que se planteará en la Introducción, luego se desarrollará un diagrama de clases del dominio es decir, que se les pueda dar una solución en un sistema informático. Posteriormente definidas estas se realizará el levantamiento de requisitos funcionales y no funcionales y se decidirá, analizando los mismos, como se agruparán para definir el diagrama de Casos de Uso del Sistema y quiénes serán los actores del mismo.

Siendo el análisis y el diseño los flujos de Trabajo que aportan mayor información, pues se ocupan en el núcleo técnico de la Ingeniería de Software, se definirá la arquitectura para desarrollar el sistema, los diagramas de clases del diseño web y de colaboración, se diseñará la Base de Datos que responderá a las necesidades del sistema con una descripción de las tablas definidas en la misma, culminándose con los diagramas de despliegue y componentes correspondientes al tema.

Índice

Introducción.....	1
Tareas a cumplir por estudiantes:.....	2
1.1 Introducción.....	5
1.2 Conceptos fundamentales y sus características.....	5
1.2.1 Sistema de archivo como modalidad de preservación y difusión.....	5
1.2.1.1 ¿Qué es un archivo?.....	5
1.2.1.2 ¿Qué es un archivo histórico?.....	5
1.2.2 Archivos 3000.....	6
1.2.3 Archivo Nacional de la República de Cuba.....	6
1.2.4 PARES.....	7
1.2.5 PAPIRO.....	7
1.2.6 AvilaDOC.....	7
1.2.7 Normas que se van a utilizar.....	8
1.2.7.1 ISAD (G) Alcance y objeto.....	8
¿Qué es ISAD (G)?.....	8
1.2.8 Lenguajes de Programación Web.....	9
1.2.8.1 PHP (Hypertext Preprocessor).....	10
1.2.8.2 Java.....	12
1.2.8.3 Perl.....	13
1.2.9 Servidores Web.....	14
1.2.9.1 ¿Qué es un servidor Web?.....	14
1.2.9.2 Apache.....	15
1.2.9.3 Internet Information Services.....	15
1.2.10 La arquitectura cliente / servidor.....	16
1.2.11 Sistemas de Gestión de Bases de Datos.....	16
1.2.11.1 PostgreSQL.....	16
1.2.11.2 MySQL.....	17
1.2.12 Lenguaje de Modelado.....	18
1.2.13 Herramientas a utilizar.....	19

1.2.14 Metodología de desarrollo del software.	19
1.3 Conclusiones	20
2. Descripción de la solución propuesta	21
2.1 Introducción	21
2.2 Modelo de Dominio	21
2.3 Descripción del Problema de Dominio	22
Diagrama de clase del dominio.	23
2.4 Requerimientos funcionales.	23
2.5 Requerimientos no funcionales.....	25
2.6 Actores del sistema.	26
Tabla 2-1.Actor del sistema.	26
Diagrama de caso de uso del sistema.	27
2.7 Casos de uso del sistema. (CUS).....	27
2.7.1 Tabla 2.2 Caso de Uso Gestionar descripción.	27
2.7.2 Tabla 2.3 Caso de uso Gestionar Ubicación Lógica.....	29
2.7.3 Tabla 2.4 Caso de uso Asignar imagen.....	31
2.7.4 Tabla 2.5 Caso de uso Gestionar Ubicación Física.	32
2.7.5 Tabla 2.6 Caso de uso Gestionar Caja.....	34
2.7.6 Tabla 2.7 Caso de uso Buscar archivo.....	36
2.7.7 Tabla 2.8 Caso de uso Autenticar usuario.	38
2.7.8 Tabla 2.9 Caso de uso Gestionar usuario.....	39
2.8 Análisis.....	43
2.8.1 Diagrama de clase del análisis. Gestionar Descripción.....	43
2.8.2 Diagrama de clase del análisis. Gestionar Ubicación Física.....	44
2.8.3 Diagrama de clase del análisis. Gestionar ubicación Lógica.	45
2.8.4 Diagrama de clase del análisis. Gestionar Caja.	46
2.8.5 Diagrama de clase del análisis. Autenticar Usuario.....	47
2.8.6 Diagrama de clase del análisis. Gestionar Usuario.	47
2.8.7 Diagrama de clase del análisis. Subir Archivo.	48
2.9 Diseño	48
2.9.1 Diagrama de clase. Gestionar Descripción.	49
2.9.2 Diagrama de clase. Crear Ubicación Lógica.....	50

2.9.3 Diagrama de clase. Asignar Imagen.....	51
2.9.4 Diagrama de clase. Eliminar Nivel Lógico	52
2.9.5 Diagrama de clase. Crear nivel Físico	53
2.9.6 Diagrama de clase. Eliminar nivel Físico.....	54
2.9.7 Diagrama de clase. Ubicar en caja.	55
2.9.8 Diagrama de clase. Buscar archivo.....	56
2.9.9 Diagrama de clase. Autentificar usuario.	57
2.9.10 Diagrama de clase. Gestionar usuario.....	58
2.9.11 Diagrama de clase. Registrar usuario.....	59
2.10 Diagramas de colaboración.....	59
2.10.1 Diagrama de colaboración. Adicionar Descripción.	60
2.10.2 Diagrama de colaboración. Actualizar Descripción.....	61
2.10.3 Diagrama de colaboración. Crear ubicación lógica.	62
2.10.4 Diagrama de colaboración. Eliminar ubicación lógica.	63
2.10.5 Diagrama de colaboración. Crear ubicación física.	64
2.10.6 Diagrama de colaboración. Eliminar ubicación física.....	65
2.10.7 Diagrama de colaboración. Buscar archivo.	66
2.10.8 Diagrama de colaboración. Asignar imagen.....	67
2.10.9 Diagrama de colaboración. Ubicar en Caja.....	68
2.10.10 Diagrama de colaboración. Adicionar Usuario.	69
2.10.11 Diagrama de colaboración. Autentificar Usuario.....	70
2.10.12 Diagrama de colaboración. Modificar Usuario.....	71
2.10.13 Diagrama de colaboración. Eliminar Usuario.	72
2.10.14 Diagrama de colaboración. Registrar Usuario.	73
2.10.15 Diagrama de colaboración. Buscar Usuario.....	74
2.10.16 Diagrama de colaboración. Asignar Documento.....	75
2.10.17 Diagrama de colaboración. Adicionar Caja.	76
2.10.18 Diagrama de colaboración. Abrir Caja.....	77
2.10.19 Diagrama de colaboración. Cerrar Caja.....	78
2.11 Diseño de la Base de Datos.	79
2.11.1 Diagrama Entidad Relación de la Base de Datos.	79
2.12 Descripción de las tablas de la Base de Datos.	80

Tabla 2.10	80
Tabla 2.11	81
Tabla 2.12	81
Tabla 2.13	82
Tabla 2.14	82
Tabla 2.15	83
Tabla 2.16	83
3. Implementación	84
3.1 Introducción	84
3.2 Modelo de Implementación	84
3.2.1 Diagrama de Componentes	85
3.3 Diagrama de Despliegue	87
Conclusiones Generales.	89
Recomendaciones.....	90
Referencias bibliográficas.	91
Bibliografía.	93
Glosario de términos.	95

Introducción.

Los archivos históricos juegan un papel fundamental en la preservación del patrimonio nacional de cada país, actualmente en muchos países existen problemas con la preservación y difusión de los documentos que constituyen los acervos históricos, los cuales no están bien organizados y en ocasiones casi la totalidad de las instituciones públicas y privadas no cuentan con estos archivos, no existen o no se aplican leyes adecuadas para la buena administración de documentos que eviten la destrucción y el saqueo de estos.

En nuestro país con el avance de las tecnologías son más las instituciones que, movidas por el ánimo de preservar y difundir su patrimonio están buscando en la captura digital una herramienta que les permita: preservar la información a largo plazo, posibilitar el acceso a ella desde lugares remotos, pero mientras más tiempo pasa los documentos y archivos siguen creciendo y más de la mitad de la información no estará en soporte de papel y el número de publicaciones electrónicas cuadruplicará las existente el día de hoy.

En ocasiones el proceso relacionado con el control de archivos y búsquedas de la información que conducen al buen desempeño del trabajo se realizan de forma manual, lo que provoca que la información en muchos casos se vea afectada en cuanto a tiempo de respuesta y en algunos casos deterioro de ella, y a su vez la eficiencia de la misma no es la esperada por las instituciones o personas vinculadas directamente con este proceso.

Tomando en cuenta la situación explicada anteriormente, se ha planteado el **Problema Científico**.
¿Cómo gestionar la preservación y difusión de documentos de un archivo histórico?

Se tomó como **Objeto de Estudio** de esta investigación los procesos que se realizan con los archivos históricos de una institución y el **Campo de Acción** la automatización de los diferentes procesos que se realizan en un sistema de archivo con los documentos históricos.

Para resolver el problema planteado el presente trabajo tiene como **Objetivo General** desarrollar una aplicación bajo tecnologías libres y multiplataforma, que permita la preservación y difusión de los documentos de archivo y como **Objetivos Específicos**:

Introducción.

- Estudiar los estándares internacionales para la descripción de documentos de archivos.
- Realizar un estudio bibliográfico sobre los diferentes Sistemas de Gestión de Documentos de Archivos hasta la fecha (Estado del Arte).
- Implementar un sistema que permita la preservación y difusión de los documentos de archivo, basado en estándares internacionales.

La Idea a Defender, si se realiza la aplicación Sistema Básico para Archivos Históricos se facilitará la preservación y difusión de los documentos históricos.

Tareas a cumplir por estudiantes:

o	Tarea	Responsable	Fecha Inicio	Fecha Fin	Observaciones
	Hacer una valoración de los estándares internacionales para la descripción de documentos de archivos.	Rafael Barrios Leyva	15/10/2007	15/11/2007	
	Estudiar las principales metodologías, sistemas y características de los archivos.	Rismaris Calvo Suarez	15/10/2007	15/11/2007	
	Valorar las características de los principales sistemas de archivos existentes en el mundo.	Rismaris Calvo Suarez	15/10/2007	10/12/2007	
	Definir el diseño teórico	Rafael Barrios	20/11/2007	7/12/2007	

de la investigación.	Leyva Rismaris Calvo Suarez			
Definir el diseño metodológico de la Investigación.	Rafael Barrios Leyva Rismaris Calvo Suarez	20/11/2007	7/12/2007	
Estudio del Gestor de Base de Datos MySQL	Rafael Barrios Leyva Rismaris Calvo Suarez	10/12/2007	20/02/2007	
Estudio del lenguaje de programación PHP en su versión 5.0	Rafael Barrios Leyva Rismaris Calvo Suarez	10/12/2007	20/02/2008	
Realizar análisis y diseño del Sistema Básico para Archivo Histórico.	Rafael Barrios Leyva	20/02/2008	30/03/2008	
Desarrollar la implementación del Sistema Básico para Archivo Histórico.	Rafael Barrios Leyva Rismaris Calvo Suarez	30/03/2008	25/06/2008	

Métodos Utilizados

- **Métodos Teóricos:**

Análisis: desglosando el problema para que, en pequeños problemas conduzca a una mejor solución y encontrar las relaciones entre ellos.

Modelación: permite realizar un estudio de las diferentes tecnologías existentes, sus cualidades y peculiaridades, para el correcto desarrollo del sistema

- **Métodos Empíricos:**

Observación: posibilita la obtención del conocimiento acerca del comportamiento de los sistemas existentes hasta el momento, además les permitió conocer la realidad mediante la percepción directa de los objetos y fenómenos.

Revisión de Documentos: se emplea para el reconocimiento del funcionamiento de las aplicaciones y las formas en que operan.

Nota: Todas las letras que se encuentran en negrita están en el glosario de término.

1. Fundamentación Teórica.

1.1 Introducción.

En el capítulo se explican algunos conceptos necesarios para el desarrollo de sistemas de archivos. Se abordan los distintos tipos de sistemas de archivos que existen en la actualidad. Se explican aspectos relacionados con las normas **ISAD (G)** e **ISSAR (CPF)** como los principios esenciales del quehacer archivístico. Estos principios están perfectamente capacitados para cubrir las necesidades de cualquier tipo de archivo.

Además, se tratan algunas de las **herramientas de autor** más usadas en la creación de este sistema. Por último se da una explicación acerca del uso de algunas metodologías para el desarrollo de sistemas informáticos, haciendo énfasis en la que se propone para el desarrollo de este trabajo.

1.2 Conceptos fundamentales y sus características.

1.2.1 Sistema de archivo como modalidad de preservación y difusión.

1.2.1.1 ¿Qué es un archivo?

Un archivo es el conjunto de documentos producidos o reunidos por una persona o una institución a lo largo de su vida. Estos documentos pueden tener muchos soportes: pueden estar en papel, pergamino, en soporte informático. Cuando se habla de archivo se habla de documentos, igual que cuando se refiere a una biblioteca pensamos en libros y cuando se hace de una hemeroteca se refiere a periódicos y revistas. Pero estos documentos no se han juntado porque sí, son el reflejo de la vida de una persona o de la historia de una institución u organismo, son los documentos que ha ido produciendo o recibiendo en su quehacer diario. [1]

1.2.1.2 ¿Qué es un archivo histórico?

Es uno o más conjuntos de documentos, independientemente de su fecha, su forma y soporte material, acumulados en un proceso natural por una persona o institución pública o privada en el transcurso de su gestión, y conservados para servir como testimonio e información a la persona o institución que los produce, o a los ciudadanos, o para ser utilizados como fuentes de historia.

Etapas del archivo:

Capítulo 1: Fundamentación Teórica.

- Archivo de gestión: bajo el control de la administración, también es conocido como archivo de oficina; debe permanecer en esta fase durante cinco años.
- Archivo administrativo: es una organización administrativa de gran volumen y complejidad; en cuanto las oficinas dejan de utilizarlo llegan a este archivo.
- Archivo intermedio: la documentación ha perdido prácticamente toda la utilidad que había tenido; durante otros 15 años se valora, se selecciona y se expurga.
- Archivo histórico: la documentación se conserva indefinidamente, debe ser factible de transmitir a las futuras generaciones. [2]

Los archivos históricos constituyen archivos de valor que tienen como misión la de garantizar la conservación de aquellos documentos considerados de interés y la de difundir la información contenida en ellos, tanto a los propios productores de los documentos como a los investigadores y al público en general.

Es importante destacar que la utilización de este módulo en instituciones archivísticas solo es aplicable para los archivos históricos.

1.2.2 Archivos 3000

Desarrollado por **3000 Informática S.L.**, es un software para la gestión de archivos que utiliza las normas ISAD (G) e ISAAR (CPF), y una adaptación del formato **USMARC21**, de las pautas **OCLC** para la introducción automatizada de datos y el nuevo formato de autoridades del Ministerio de Cultura de España. Además, utiliza la norma **ISO-15489** para la gestión de registros y la Especificación **MoReq** para registros electrónicos. Ofrece diversos productos para la gestión archivística, brindando desde las soluciones más básicas (Archivo 3000 W Premier) hasta las más complejas (Archivo 3000 W Suit). Cada uno de los productos, así como los servicios que brinda, poseen su valor monetario, además de que no revela su código fuente. [3]

1.2.3 Archivo Nacional de la República de Cuba

28 de enero de 1840, con el nombre de Archivo General de la Real Hacienda, el actual Archivo Nacional de la República de Cuba, fue la quinta de estas instituciones creadas en América Latina, luego de Argentina (1821), México (1823), Bolivia (1825) y Brasil (1838), para atesorar y proteger los documentos, siguiendo el espíritu de conservación documental que comenzara con el reinado de Felipe II en España y sus ordenanzas de 1569. [4]

Capítulo 1: Fundamentación Teórica.

Algunos de las provincias que conforman la división político-administrativa del país no se encuentran comunicados o interconectados unos con los otros para la búsqueda de documentos.

1.2.4 PARES

El Portal de Archivos Españoles es un proyecto del Ministerio de Cultura destinado a la difusión en Internet del Patrimonio Histórico Documental Español conservado en su red de centros. Como proyecto abierto y dinámico sirve de marco de difusión para otros proyectos archivísticos de naturaleza pública o privada, previamente establecido un marco de cooperación con el Ministerio de Cultura. PARES ofrece un acceso libre y gratuito, no solo al investigador, sino también a cualquier ciudadano interesado en acceder a los documentos con imágenes digitalizadas de los Archivos Españoles. [5]

Es un sitio bien diseñado pero resulta imposible reutilizar su código, ya que este es cerrado.

1.2.5 PAPIRO

PAPIRO, es un producto informático de uso libre que emplea herramientas igualmente libres y permite conservar documentación de valor histórico al evitar su manipulación; pues, al digitalizarse el documento, este se consulta en formato electrónico. Esta versión incluye también la gestión de publicaciones periódicas, especialmente revistas. La consulta de los documentos, puede ser realizada a través de una eficiente gestión que permite recuperar información por todas las variables de almacenamiento (campos de obligatoria presencia según la norma ISAD-G para intercambio internacional), en la base de datos; mientras, la posibilidad de colocar en la red de redes dichas bases de datos y las imágenes de los documentos, no sólo democratizan la accesibilidad documental; sino, que ahorra recursos materiales y tiempo, haciendo más eficaz y eficiente el proceso de búsqueda e investigación documental. Está concebido para correr bajo la plataforma Windows en las versiones 2000 y XP. Los formatos de imagen de los documentos están en pdf, estructura de uso común y gran poder de compresión, y para su consulta en la red se diseñó un método de gestión dinámica con páginas PHP.

Está concebido solamente para funcionar en Windows en la versión 2000 o XP y tampoco posee la funcionalidad de búsqueda a través de varios archivos interconectados. [6]

1.2.6 AvilaDOC

AvilaDOC es una aplicación web, desarrollada con una base de datos centralizada, destinada a la gestión, tramitación y resguardo de archivos electrónicos y digitales. Este proyecto posee como objetivos fundamentales: organizar el flujo documental, definir una estructura archivística y lograr un archivo digital centralizado, además de poseer diversas funcionalidades, como la posibilidad de búsquedas o

Capítulo 1: Fundamentación Teórica.

recuperación de información, pero no ofrece la funcionalidad de poder buscar o recuperar descripciones de archivos históricos en diferentes archivos interconectados al unísono. [7]

Algunos de estos productos son muy caros, no son multiplataforma, tienen el código cerrado, no tienen búsquedas bien implementadas y carecen de buena interconexión. Se decidió hacer este que tenga todas las características positivas que tienen ellos y corregir las negativas y ponerla en prácticas.

1.2.7 Normas que se van a utilizar.

1.2.7.1 ISAD (G) Alcance y objeto.

ISAD (G) - constituye una herramienta de gestión muy valiosa para los archivos, brindando especificaciones elementales para la elaboración de descripciones archivísticas compatibles a nivel nacional e internacional, que pueden aplicarse con independencia del tipo documental o soporte físico de los documentos de archivo, excepto para documentos especiales como sellos, registros sonoros o mapas.

Este conjunto de reglas generales es parte de un proceso que busca facilitar la integración de las descripciones de diferentes archivos en un sistema unificado de información. Es su intención servir a la elaboración de una primera aproximación a los fondos, en una visión sintética de conjunto.[8]

¿Qué es ISAD (G)?

La Norma Internacional General de Descripción Archivística constituye una herramienta primordial para la gestión de los archivos. ISAD (G) permite elaborar descripciones archivísticas estándares que pueden tener tanto nivel de detalle como se desee especificar. Cuenta con cuatro principios fundamentales:

- La descripción del archivo se origina de lo general a lo específico.
- La información debe ser relevante para el nivel de descripción.
- Las descripciones deben estar vinculadas entre niveles.
- No repetición de la información.

Además se conforma por 26 elementos agrupados en siete áreas de descripción:

- Área de identificación.
- Área de contexto.
- Área de contenido y estructura.

Capítulo 1: Fundamentación Teórica.

- Área de condiciones de acceso y uso.
- Área de documentación asociada.
- Área de notas.
- Área de control de la descripción.

De todos los elementos que conforman la norma, constituyen esenciales para el intercambio internacional de la información descriptiva:

- El código de referencia.
- El título.
- El productor.
- La fecha (s).
- La extensión de la unidad de descripción.
- El nivel de descripción.

1.2.8 Lenguajes de Programación Web.

Al irrumpir en el mundo la **WWW** y con el amplio crecimiento de la Internet y la Intranet, las aplicaciones y los sitios Web han adquirido gran popularidad a nivel mundial. Esta tecnología requiere de pocos recursos por parte del cliente, basta solamente con la presencia de un navegador de Internet.

La programación Web, parte de las siglas WWW, que significan telaraña mundial.

Para realizar una página con la programación Web, se deben tener claros, tres conceptos fundamentales los cuales son, el **URL**, es un sistema con el cual se localiza un recurso dentro de la red, este recurso puede ser una página Web, un servicio o cualquier otra cosa. En resumen el URL no es más que un nombre, que identifica una computadora, dentro de esa computadora un archivo que indica el camino al recurso que se solicita.

El siguiente concepto dentro de la programación Web, es el protocolo encargado de llevar la información que contiene una página Web por toda la red de Internet, como es el **HTTP**.

Y por último el lenguaje necesario cuya funcionalidad es la de representar cualquier clase de información que se encuentre almacenada en una página Web, este lenguaje es el **HTML**.

Capítulo 1: Fundamentación Teórica.

En la programación Web, el HTML es el lenguaje que permite codificar o preparar documentos de hipertexto, que viene a ser el lenguaje común para la construcción de una página Web.

Con el comienzo de Internet y la programación Web, se diferenciaron los diseños gráficos tradicionales, con lo que se empezaron a diseñar interfaces concretas para este medio, buscando ficheros pequeños para facilitar la carga de los mismos. La programación Web se orientaba a un diseño muy cargado e interactuando con el usuario, mientras que al empezar a competir con millones de Webs se ha optado más por el diseño sencillo y de fácil comprensión.

En programación Web se creó la necesidad de conocer a fondo diferentes lenguajes de programación como HTML, **JavaScript** y **DHTML**. [9]

1.2.8.1 PHP (Hypertext Preprocessor).

PHP, se conoce originalmente como **Personal Home Pages**, fue concebido en el otoño de 1994 por Rasmus Lerdorf. La primera versión salió en los comienzos de 1995, y fue ahí donde Rasmus se dio cuenta que haciendo en proyecto código-abierto, las personas arreglarían sus problemas. La primera versión fue muy precaria y tenía un **parser** que reconocía solo unas pocas **macros** y brindaba algunas utilidades que se usaban comúnmente en sitios web.

El parser fue reescrito a mediados de 1995 y se lo renombró a **PHP/FI** versión 2. El "FI" en esta versión quería decir Intérprete formal. Lo que Rasmus había agregado a PHP fue de acuerdo a las necesidades crecientes de los sitios web. El soporte para **MySQL** fué agregado. PHP/FI tuvo un crecimiento masivo, y otra gente empezó a contribuir programando regularmente.

A mediados de 1997 Zeev Suraski y Andi Gutmans reescribieron el parser principal, y PHP cambió de estar en manos de Rasmus a un grupo más orientado al proyecto. Esto formó las bases para que **PHP3**, fuere ahora llamado PHP.

La versión de **PHP4**, es otra reescritura de Suraski and Gutmans y está basada en el motor **Zend**. PHP ahora tiene doscientos contribuyentes regularmente trabajando en varias partes del proyecto. Tiene una cantidad muy grande extensiones, módulos y soporta todos los servidores más populares nativamente, y además tiene soporte para MySQL y **ODBC**.

Las últimas estadísticas muestran que PHP es actualmente usado por más de 5.5 millones de dominios, y ha tenido un gran crecimiento durante el último año. Apache, para dar alguna perspectiva,

Capítulo 1: *Fundamentación Teórica.*

Apache actualmente tiene un 60% del mercado de servidores de internet, y el servidor **IIS** (con soporte nativo para **ASP**) tiene menos de la mitad de esa proporción del mercado. [10]

Debido a la naturaleza **open-source** (código abierto) de PHP, si hay algo que actualmente no se pueda hacer en PHP no hay ningún impedimento para que se pueda escribir un módulo o una extensión en código **C** para extender la funcionalidad. Esto es posible por la buena documentación de la **API** que esta disponible para todos.

PHP4 tiene dos partes:

- El motor Zend

Es la parte del paquete PHP que mantiene los pedidos, los procesos de los archivos de script y maneja las variables y los recursos.

- PHP

Implementa el 90% de la funcionalidad que ve el usuario final. Brinda un amplio rango de módulos como el soporte para MySQL, ODBC and **XML**. [11]

Php5

El 13 de julio de 2004, fue lanzado PHP 5, utilizando el motor Zend Engine II (o Zend Engine 2). La versión más reciente de PHP es la 5.2.5 (8 de noviembre de 2007), que incluye todas las ventajas que provee el nuevo Zend Engine 2 como:

- Mejor soporte para la Programación Orientada a Objetos, que en versiones anteriores era extremadamente rudimentario, con PHP Data Objects.
- Mejoras de rendimiento.
- Mejor soporte para MySQL con extensión completamente reescrita.
- Mejor soporte a XML (XPath, DOM, etc.)
- Soporte nativo para SQLite.
- Soporte integrado para SOAP.
- Iteradores de datos.
- Manejo de excepciones. [12]

¿Por qué se utilizó php, ventajas?

- Es un lenguaje multiplataforma.

Capítulo 1: Fundamentación Teórica.

- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones. [13]

1.2.8.2 Java

Java es un lenguaje de programación con el que podemos realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Está desarrollado por la compañía Sun Microsystems con gran dedicación y siempre enfocado a cubrir las necesidades tecnológicas más punteras.

Una de las principales características por las que Java se ha hecho muy famoso es que es un lenguaje independiente de la plataforma. Eso quiere decir que si se hace un programa en Java podrá funcionar en cualquier ordenador del mercado. Es una ventaja significativa para los desarrolladores de software, pues antes tenían que hacer un programa para cada sistema operativo, por ejemplo Windows, Linux, Apple, etc. Esto lo consigue porque se ha creado una Máquina de Java para cada sistema que hace de puente entre el sistema operativo y el programa de Java y posibilita que este último se entienda perfectamente.

La independencia de plataforma es una de las razones por las que Java es interesante para Internet, ya que muchas personas deben tener acceso con ordenadores distintos. Pero no se queda ahí, Java está desarrollándose incluso para distintos tipos de dispositivos además del ordenador como móviles, agendas y en general para cualquier cosa que se le ocurra a la industria.

Pasado y presente de Java

Java fue pensado originalmente para utilizarse en cualquier tipo de electrodoméstico pero la idea fracasó. Uno de los fundadores de **Sun** rescató la idea para utilizarla en el ámbito de Internet y convirtieron a Java en un lenguaje potente, seguro y universal gracias a que lo puede utilizar todo el

Capítulo 1: Fundamentación Teórica.

mundo y es gratuito. Una de los primeros triunfos de Java fue que se integró en el navegador Netscape y permitía ejecutar programas dentro de una página web, hasta entonces impensable con el HTML.

Actualmente Java se utiliza en un amplio abanico de posibilidades. Con Java podemos programar páginas web dinámicas, con accesos a bases de datos, utilizando XML, con cualquier tipo de conexión de red entre cualquier sistema. En general, cualquier aplicación que se desee hacer con acceso a través web se puede hacer utilizando Java.

1.2.8.3 Perl

Perl, Lenguaje Práctico para la Extracción e Informe es un lenguaje de programación diseñado por Larry Wall creado en 1987. Perl toma características del C, del lenguaje interpretado shell), AWK, sed, Lisp y, en un grado inferior, de muchos otros lenguajes de programación.

Estructuralmente, Perl está basado en un estilo de bloques como los del C o **AWK**, y fue ampliamente adoptado por su destreza en el procesado de texto y no tener ninguna de las limitaciones de los otros lenguajes de script.

Características

La estructura completa de Perl deriva ampliamente del lenguaje C. Perl es un lenguaje imperativo, con variables, expresiones, asignaciones, bloques de código delimitados por llaves, estructuras de control y subrutinas.

Perl también toma características de la programación shell. Todas las variables son marcadas con un signo precedente (sigil). Los sigil identifican inequívocamente los nombres de las variables, permitiendo a Perl tener una rica sintaxis. Notablemente, los sigil permiten interpolar variables directamente dentro de las cadenas de caracteres (strings). Como en los shell, Perl tiene muchas funciones integradas para tareas comunes y para acceder a los recursos del sistema.

Perl toma las listas del **Lisp**, **hash** (memoria asociativa) del AWK y expresiones regulares del sed. Todo esto simplifica y facilita todas las formas del análisis sintáctico, manejo de texto y tareas de gestión de datos.

En Perl 5, se añadieron características para soportar estructuras de datos complejas, funciones de primer orden y un modelo de programación orientada a objetos. Éstos incluyen referencias, paquetes y una ejecución de métodos basada en clases y la introducción de variables de ámbito léxico, que hizo más fácil escribir código robusto (junto con el programa script). Una característica principal introducida en Perl 5 fue la habilidad de empaquetar código reutilizable como módulos. Larry Wall indicó más adelante que

"la intención del sistema de módulos de Perl 5 era apoyar el crecimiento de la cultura Perl en vez del núcleo de Perl".

Todas las versiones de Perl hacen el tipificado automático de datos y la gestión de memoria. El intérprete conoce el tipo y requerimientos de almacenamiento de cada objeto en el programa; reserva y libera espacio para ellos según sea necesario. Las conversiones legales de tipo se hacen de forma automática en tiempo de ejecución; las conversiones ilegales son consideradas errores fatales. [14]

1.2.9 Servidores Web

1.2.9.1 ¿Qué es un servidor Web?

Un servidor web es un programa que implementa el protocolo HTTP (hypertext transfer protocol). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML (hypertext markup language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.

Es un programa que se ejecuta continuamente en un ordenador (también se emplea el término para referirse al ordenador que lo ejecuta), manteniéndose a la espera de peticiones por parte de un cliente (un navegador de Internet) y que responde a estas peticiones adecuadamente, mediante una página web que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error.

Es un tipo de software que realiza ciertas tareas en nombre de los usuarios. El término servidor ahora también se utiliza para referirse al ordenador físico en el cual funciona ese software, una máquina cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos.

Este uso dual puede llevar a confusión. Por ejemplo, en el caso de un servidor web, este término podría referirse a la máquina que almacena y maneja los sitios web, y en este sentido es utilizada por las compañías que ofrecen **hosting** u hospedaje. Alternativamente, el servidor web podría referirse al software, que funciona en la máquina y maneja la entrega de los componentes de las páginas web como respuesta a peticiones de los navegadores de los clientes.

Los archivos para cada sitio de Internet se almacenan y se ejecutan en el servidor. Hay muchos servidores en Internet y muchos tipos de servidores, pero comparten la función común de proporcionar el acceso a los archivos y servicios.

Un servidor sirve información a los ordenadores que se conecten a él. Cuando los usuarios se conectan a un servidor pueden acceder a programas, archivos y otra información del servidor.

Capítulo 1: Fundamentación Teórica.

En la web, un servidor web es un ordenador que usa el protocolo http para enviar páginas web al ordenador de un usuario cuando el usuario las solicita.

Los servidores web, servidores de correo y servidores de bases de datos son a lo que tiene acceso la mayoría de la gente al usar Internet.

Algunos servidores manejan solamente correo o solamente archivos, mientras que otros hacen más de un trabajo, ya que un mismo ordenador puede tener diferentes programas de servidor funcionando al mismo tiempo.

Los servidores se conectan a la red mediante una interfaz que puede ser una red verdadera o mediante conexión vía línea telefónica o digital. [15]

1.2.9.2 Apache

Apache es el servidor más popular en Internet desde abril de 1996. En Noviembre del 2005, una encuesta sobre servidores Web de **Netcraft** arrojó que más del 70% de los sitios Web en Internet usaban apache lo que lo hace ser preferido por sobre otros servidores. Entre sus características sobresalen:

- **Fiabilidad:** Alrededor del 90% de los servidores con más alta disponibilidad funcionan con Apache.
- **Gratuidad:** Apache es totalmente gratuito, y se distribuye bajo la licencia Apache Software License, que permite la modificación del código.
- **Extensibilidad:** Se pueden añadir módulos para ampliar las amplias capacidades de Apache. Existe una amplia variedad de módulos, que permiten desde generar contenido dinámico con PHP, Java, Perl, entre otros, monitorizar el rendimiento del servidor. Estos módulos pueden ser creados por cualquier persona con conocimientos de programación.

El proyecto de Servidor HTTP Apache es un esfuerzo para desarrollar y mantener un servidor HTTP abierto para modernos sistemas operativos como **UNIX** y Windows NT. El objetivo de este proyecto es ofertar un servidor seguro, eficiente y extensivo que de servicios HTTP en sincronismo con los estándares actuales.

1.2.9.3 Internet Information Services

Internet Information Services, IIS, es una serie de servicios para los ordenadores que funcionan con Windows. Originalmente era parte del **Option Pack** para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server

Capítulo 1: Fundamentación Teórica.

2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: **FTP, SMTP, NNTP y HTTP/HTTPS.**

Este servicio convierte a un ordenador en un servidor de Internet o Intranet es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas Web tanto local como remotamente (servidor Web).

El servidor Web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas, por ejemplo Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl. [16]

1.2.10 La arquitectura cliente / servidor.

En el ámbito de enviar y recibir información la arquitectura Cliente-Servidor es una de las más importantes y utilizadas. Esta arquitectura es la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo y/o, a través de la organización, en múltiples plataformas. Es una herramienta potente para guardar los datos en una base de datos como servidor.

Los elementos principales que tiene este modelo son el elemento llamado cliente(es un consumidor de servicios) y el otro elemento llamado servidor(es un proveedor de servicios). Ambos elementos pueden estar conectados a una red local o una red amplia, como la que se puede implementar en una empresa o a una red mundial como lo es la Internet. Bajo este modelo cada usuario tiene la libertad de obtener la información que requiera en un momento dado proveniente de una o varias fuentes locales o distantes y de procesarla como según le convenga.

El Cliente-Servidor es una relación entre procesos corriendo en máquinas separadas e interactúan por un mecanismo de pasaje de mensajes con un pedido de servicios y una respuesta. Esta arquitectura es el resultado de la integración de dos culturas. Por un lado, la del **Mainframe** que aporta capacidad de almacenamiento, integridad y acceso a la información y, por el otro, la del computador que aporta facilidad de uso, bajo costo, presentación atractiva y una amplia oferta en productos y aplicaciones. [17]

1.2.11 Sistemas de Gestión de Bases de Datos.

1.2.11.1 PostgreSQL.

PostgreSQL es un servidor de base de datos relacional orientada a objetos de software libre, liberado bajo la licencia BSD.

Capítulo 1: Fundamentación Teórica.

Como muchos otros proyectos open source, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG (**PostgreSQL Global Development Group**).

Características

Algunas de sus principales características son:

Alta concurrencia

Mediante un sistema denominado **MVCC** (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último que se hace en la tabla. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

Amplia variedad de tipos nativos

PostgreSQL provee nativamente soporte para:

- Números de precisión arbitraria.
- Texto de largo ilimitado.
- Figuras geométricas (con una variedad de funciones asociadas)
- Direcciones IP (IPv4 e IPv6).
- Bloques de direcciones estilo CIDR.
- Direcciones MAC.
- Arrays. [18]

1.2.11.2 MySQL

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual. MySQL AB pertenece a **Sun Microsystems** desde enero de 2008.

Por un lado se ofrece bajo la **GNU GPL** para cualquier uso compatible con esta licencia, pero las empresas que quieran incorporarlo en productos privativos pueden comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en **ANSI C** .

Al contrario que proyectos como Apache, donde el software es desarrollado por una comunidad pública y el copyright del código está en poder del autor individual, MySQL es propiedad y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.

Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet. MySQL AB fue fundado por David Axmark, Allan Larsson, y Michael Widenius. [19]

1.2.12 Lenguaje de Modelado

UML

Es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir. UML capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. Un sistema se modela como una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia a un usuario externo. El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. UML no es un lenguaje de programación.

Las herramientas pueden ofrecer generadores de código de UML para una gran variedad de lenguajes de programación, así como construir modelos por ingeniería inversa a partir de programas existentes. Es un lenguaje de propósito general para el modelado orientado a objetos y el modelado visual que permite una abstracción del sistema y sus componentes.

Existían diversos métodos y técnicas Orientadas a Objetos, con muchos aspectos en común pero utilizando distintas notaciones, se presentaban inconvenientes para el aprendizaje, aplicación, construcción y uso de herramientas, etc., además de pugnas entre enfoques, lo que generó la creación del UML como estándar para el modelado de sistemas de software, pero con posibilidades de ser aplicado a todo tipo de proyectos.

Hoy en día el lenguaje de modelado visual está consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Mediante UML es posible establecer la serie de requerimientos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código.

Capítulo 1: Fundamentación Teórica.

El UML estándar está compuesto por tres partes: bloques de construcción (tales como clases, objetos, mensajes), relaciones entre los bloques (tales como asociación, generalización) y diagramas (por ejemplo, diagrama de actividad), entonces partiendo que se utilizó como herramienta Case el Visual Paradigm se hace necesario entonces el UML como lenguaje de modelado.

1.2.13 Herramientas a utilizar

Para solucionar los problemas expuesto anteriormente se utilizaron las siguientes herramientas:

Case, herramienta basada en UML, que permite realizar la modelación del sistema a través de diagramas que se obtienen como parte de la documentación del sistema. De las herramientas CASE nosotros utilizaremos Visual Paradigm.

EMS SQL Manager 2005, herramienta utilizada para la gestión de la base de datos, ya que es gratuita y de fácil utilización.

1.2.14 Metodología de desarrollo del software.

Rational Unified Process (RUP)

Proceso de desarrollo propuesto por “Rational Software Corporation” resultado del esfuerzo de las tres últimas décadas en desarrollo de software y de la experiencia de sus creadores Ivar Jacobson, Grady Booch y James Rumbaugh.

Su antecedente más importante lo ubicamos en 1967 con la Metodología Ericsson (Ericsson Approach), ésta es una aproximación de desarrollo basada en componentes, que introdujo el concepto de caso de uso; entre los años de 1987 a 1995 Jacobson funda la compañía “Objectory AB” y lanza el proceso de desarrollo Objectory (abreviación de Object Factory), posteriormente en 1995 “Rational Software Corporation” adquiere “Objectory AB” y es entre 1995 y 1997 que se desarrolla “Rational Objectory Process (ROP)” fruto del encuentro y evolución de Objectory 3.8 y la Metodología Rational (Rational Approach) que adopta por primera vez UML (Universal Modeling Language) como Lenguaje de Modelación Universal.

A principios de los noventas, la guerra de los métodos hizo evidente la necesidad de unificar criterios, es así como Grady Booch autor del método Booch y James Rumbaugh (desarrollador para General Electric) se unieron en Rational en 1994, después en 1995 se une Jacobson y gracias al esfuerzo de varias compañías y metodologistas evolucionó UML hasta ser un estándar en 1997, el cual es adoptado en todos los modelos del ROP. Desde ese entonces y a la cabeza de Booch, Jacobson y Rumbaugh. [20].

1.3 Conclusiones

Al concluir este capítulo se ha podido apreciar todo lo referente a la base teórica que fundamenta esta aplicación, acercando al lector a los conceptos fundamentales, el estado del arte, los lenguajes de programación para la Web, los servidores Web, los sistemas de gestión de bases de datos y por último la metodología de ingeniería del software a utilizar.



2. Descripción de la solución propuesta

2.1 Introducción

En este capítulo se describirá el flujo de trabajo de los procesos llevados a cabo para desarrollar el sistema, dándole solución al problema de investigación planteado. Se detallarán los distintos pasos de la metodología propuesta en el capítulo anterior (RUP) para el desarrollo de la aplicación. Primero se realiza un modelo de dominio, donde se capturan las clases más importantes en el contexto del sistema. Se hace un levantamiento de los requisitos no funcionales y funcionales, estos últimos se estructuran mediante los Casos de Uso del sistema, de los cuales se ofrece una descripción textual, el Modelo del Diseño y el diagrama Entidad Relación de la Base de Datos.

Además se detallará la solución propuesta para la realización del sistema básico para Archivos Históricos.

2.2 Modelo de Dominio

Hay por lo menos dos aproximaciones para expresar el contexto de un sistema en una forma utilizable para los desarrolladores de software: modelado del dominio y modelado del negocio. Un modelo del dominio describe los conceptos importantes del contexto como objetos del dominio y enlaza estos objetos unos con otros. La identificación y asignación de un nombre para estos objetos ayuda a desarrollar un glosario de términos que permitirán comunicarse mejor a los que trabajen en un sistema.

”Las entidades del negocio se derivan a partir de los clientes del negocio. Las clases del dominio se obtienen de la base del conocimiento de unos pocos expertos del dominio, o posiblemente del conocimiento (otras clases del dominio, especificaciones de requisitos, etc.) asociado con sistemas similares al que se está desarrollando.”.

Capítulo 2: Descripción de la solución propuesta.

2.3 Descripción del Problema de Dominio

Un sistema básico para archivos históricos constituye una aplicación que describirá la posibilidad a un usuario de buscar documentos históricos, preservarlos y difundirlos.

La aplicación esta formada por cuatro interfaces, una destinada a la búsqueda y visualización de los documentos históricos, otra destinada a crear la ubicación lógica del documento, así como subirle archivos y ubicarlos en una caja, la tercera está destinada para crear la ubicación física del documento y la última para la administración del sistema.

La interfaz de administración contiene el formulario para la autenticación del usuario, de no existir alguno se procede a la creación de un nuevo usuario.

El sistema brindará las siguientes funcionalidades:

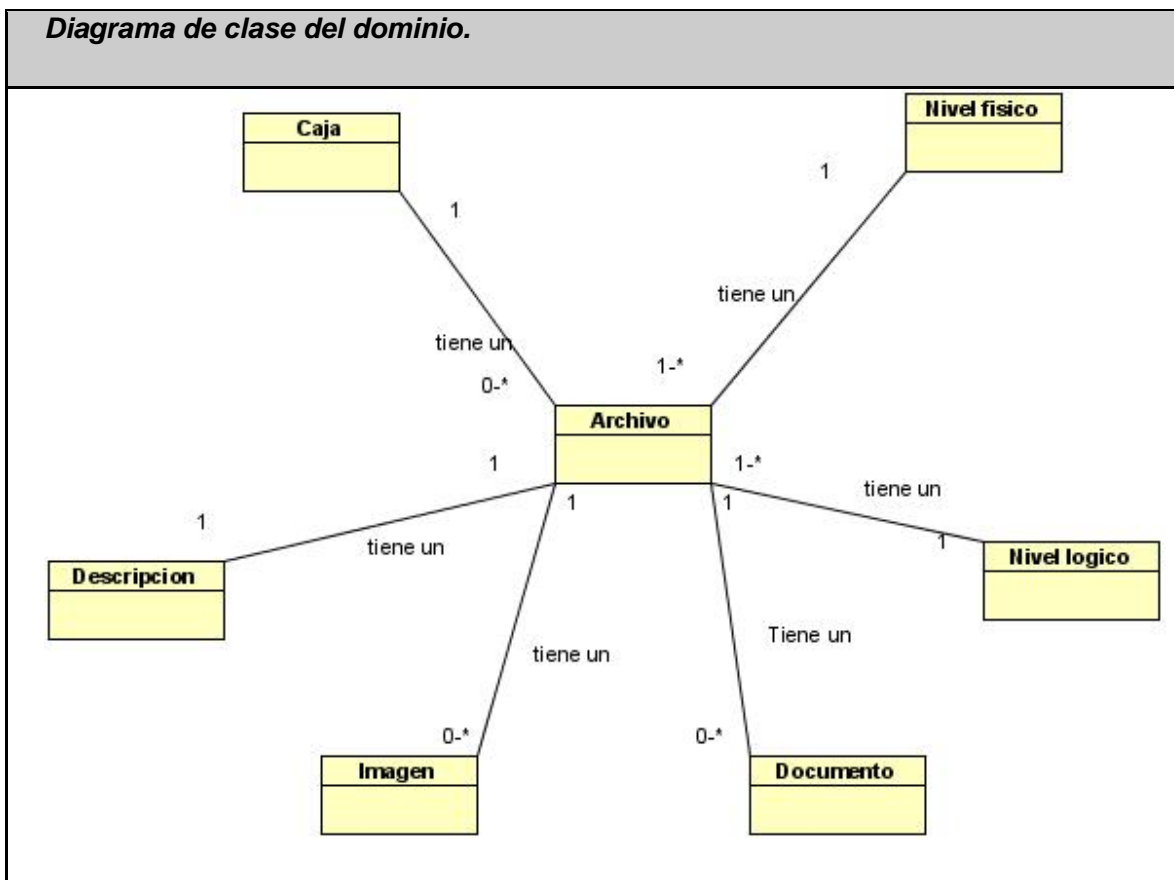
1- Búsquedas de archivos:

Cualquier usuario puede realizar la búsqueda de archivos históricos, existe una búsqueda avanzada por (fecha, título, autor, nivel de descripción, unidad de descripción, nombre del productor) que permitirá al usuario listar una serie de documentos que se encuentren algunas de estas categoría y él escogerá el que necesite.

2- Gestionar descripción:

Brindará la posibilidad de guardar y actualizar las descripciones que contienen los archivos históricos.

Diagrama de clase del dominio.



2.4 Requerimientos funcionales.

RF1 - Gestionar descripción

RF1.1-Adicionar descripción

RF1.2- Obtener descripción

RF1.3- Actualizar descripción

RF2 – Gestionar ubicación lógica.

RF2.1- Crear ubicación Lógica.

RF2.2- Eliminar ubicación Lógica.

RF3 -Asignar imagen.

Capítulo 2: Descripción de la solución propuesta.

RF4-Gestionar ubicación física.

RF4.1- Crear ubicación física.

RF4.2- Eliminar ubicación física.

RF5- Gestionar caja.

RF5.1 Ubicar en caja.

RF5.2- Adicionar caja.

RF5.3- Cerrar caja.

RF6 – Buscar archivo.

RF6.1-Buscar archivo por fecha.

RF6.2-Buscar archivo por título.

RF6.3-Buscar archivo por autor.

RF6.4-Buscar archivo por nivel de descripción.

RF6.5-Buscar archivo por unidad de descripción.

RF6.6-Buscar archivo por el nombre del productor.

RF7- Autenticar usuario.

RF8 - Gestionar usuario.

RF8.1- Modificar usuario.

RF8.2- Adicionar usuario.

RF8.3- Eliminar usuario.

RF8.4- Registrar usuario.

2.5 Requerimientos no funcionales.

- **Software**

- La aplicación requiere tener instalado el lenguaje de programación PHP 5.0
- La aplicación utilizará como base de datos MySQL.
- Se utilizará como servidor web Apache 2.0.

- **Seguridad y Privacidad**

- Restringir el acceso a los usuarios al área de administración.

- **Portabilidad**

- La aplicación está diseñado para soportar su funcionalidad en sistemas operativos libre como propietario.

- **Hardware**

- Requiere al menos de RAM 256 MB.
- El disco duro requiere al menos 5 GB para almacenar la Base de Datos.

- **Restricciones en el diseño y la implementación**

- Lenguaje de programación PHP.

- **Legales**

- La aplicación y toda la documentación generada pertenecen al proyecto Archivos Históricos y la Universidad de las Ciencias Informáticas.

- **Confiabilidad**

- La aplicación debe estar disponible de forma tal que se pueda acceder a todas sus funcionalidades.

Capítulo 2: Descripción de la solución propuesta.

2.6 Actores del sistema.

Tabla 2-1. Actor del sistema.

Actores del sistema	Justificación
Usuario lógico	Es el encargado de adicionar descripción, actualizar descripción, crear ubicación lógica entre otras.
Administrador del sistema	Es aquel que se encarga de adicionar, eliminar o modificar a los usuarios.
Usuario físico	Es el encargado de darle la dirección física al documento
Usuario Externo	Es el encargado de buscar los archivos que necesite.

Capítulo 2: Descripción de la solución propuesta.

Diagrama de caso de uso del sistema.

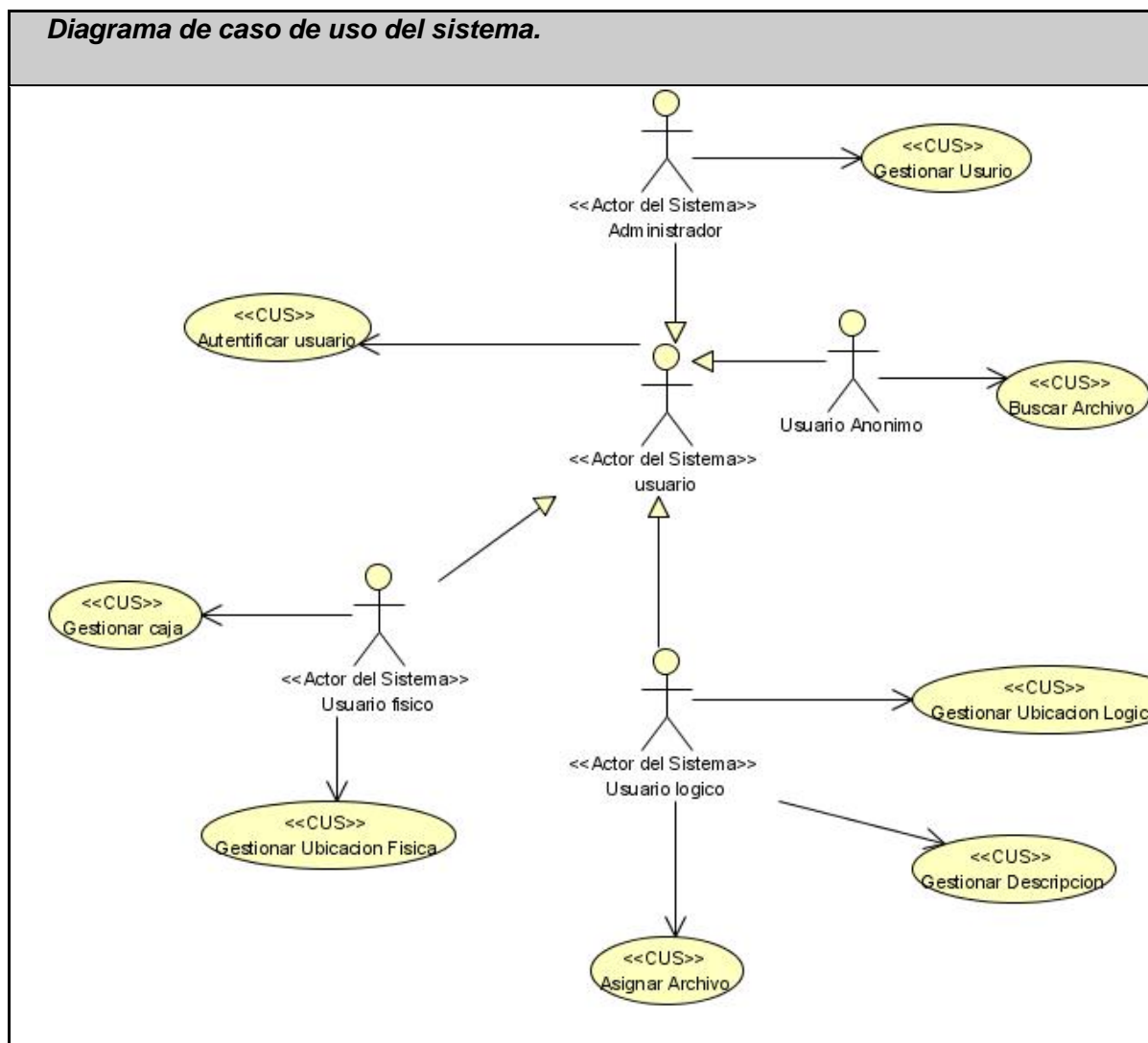


Fig. 2-1. Diagrama de CUS

2.7 Casos de uso del sistema. (CUS)

2.7.1 Tabla 2.2 Caso de Uso Gestionar descripción.

CU-1	Gestionar Descripción.
Actor(s)	Usuario Lógico

Capítulo 2: Descripción de la solución propuesta.

Descripción	El mismo se inicia cuando el actor invoca el servicio que permite realizar las acciones relacionada con la gestión de descripción, tales como adicionar descripción, obtener descripción y actualizar descripción, el sistema devuelve un mensaje diciendo si se realizo o no la acción. Y finaliza así el caso de uso.	
Referencia	RF1 (RF1.1,RF1.2)	
Flujo Normal de Eventos		
Sección Adicionar descripción		
Acción del Actor	Respuesta del Sistema	
1. El usuario lógico selecciona la opción guardar descripción.	2. El sistema solicita los datos necesarios (código de referencia, titulo, productor, fecha, extensión de la unidad de descripción y nivel de descripción).	
3. El usuario lógico introduce los datos.	4. El sistema verifica los datos.	
	5. El sistema guarda los datos.	
Flujo Alternativo		
Acción del Actor	Respuesta del Sistema	
	6. Si el usuario quiere guardar datos que ya existen, el sistema notifica con un mensaje de error.	
Sección Actualizar descripción		
Acción del Actor	Respuesta del Sistema	
1. El usuario lógico selecciona la opción de actualizar una descripción.	2. El sistema solicita los datos necesarios (código de referencia, titulo, productor, fecha, extensión de la unidad de	

Capítulo 2: Descripción de la solución propuesta.

	descripción y nivel de descripción).
3. El usuario lógico introduce sus datos.	4. El sistema verifica sus datos.
	5. El sistema actualiza los datos.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
	6. Si los datos ya existen, el sistema le notifica un error.
Prioridad	Crítica

2.7.2 Tabla 2.3 Caso de uso Gestionar Ubicación Lógica.

CU-2	Gestionar Ubicación Lógica
Actor(s)	Usuario Lógico
Descripción	El mismo se inicia cuando el actor invoca el servicio que permite realizar las acciones relacionada con la gestión de ubicación lógica, tales como crear ubicación, eliminar ubicación, el sistema devuelve un mensaje diciendo si se realizo o no la acción. Y finaliza así el caso de uso.
Referencia	RF2 (RF2.1,RF2.2)
Sección Crear Ubicación Lógica	
Acción del Actor	Respuesta del Sistema

Capítulo 2: Descripción de la solución propuesta.

1. El usuario lógico selecciona la opción crear subnivel.	2. El sistema le muestra el subnivel que puede crear.
3. El usuario lógico crea el subnivel.	4. El sistema guarda la ubicación.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema

Sección Eliminar Nivel Lógico	
Acción del Actor	Respuesta del Sistema
1. El usuario lógico selecciona la opción eliminar nivel.	2. El sistema le muestra una lista de niveles.
3. El usuario lógico selecciona la opción eliminar nivel.	4. El sistema verifica que no tenga ningún archivo guardado el.
	5. El sistema elimina el nivel.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema

Capítulo 2: Descripción de la solución propuesta.

	6. Si tiene un archivo guardado en el nivel de documentación simple el sistema muestra error.
Prioridad	Crítica

2.7.3 Tabla 2.4 Caso de uso Asignar imagen.

CU-3	Asignar imagen	
Propósito:	Permite al usuario lógico asignar una imagen del archivo al nivel de documentación simple.	
Actores:	Usuario lógico (Inicia)	
Resumen:	El caso de uso inicia cuando el usuario lógico se autentifica, luego recorre la ubicación lógica hasta llegar a la unidad de documentación simple que es donde se ubicará la imagen, esto se realizará en caso de que el archivo contenga una imagen. Terminando así el caso de uso.	
Referencia:	RF3	
Precondiciones:	El usuario lógico debe estar autenticado.	
Poscondiciones:	Guardar la imagen.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El usuario lógico selecciona el nivel donde va a insertar la imagen.	2. El sistema le muestra una lista de opciones.	

Capítulo 2: Descripción de la solución propuesta.

3. El usuario lógico selecciona la opción insertar imagen.	4. El sistema pide la dirección de donde se encuentra la imagen.
	5. El sistema guarda la imagen.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
Prioridad	Crítica

2.7.4 Tabla 2.5 Caso de uso Gestionar Ubicación Física.

CU-4	Gestionar Ubicación Física
Actor(s)	Usuario Físico
Descripción	El mismo se inicia cuando el actor invoca el servicio que permite realizar las acciones relacionada con la gestión de ubicación física, tales como crear ubicación, eliminar ubicación, el sistema devuelve un mensaje diciendo si se realizo o no la acción. Y finaliza así el caso de uso.
Referencia	RF4 (RF4.1,RF4.2)
Flujo Normal de Eventos	
Sección Crear Ubicación Física	
Acción del Actor	Respuesta del Sistema

Capítulo 2: Descripción de la solución propuesta.

1. El usuario físico selecciona la opción crear subnivel.	2. El sistema le muestra el subnivel que puede crear.
3. El usuario físico selecciona el subnivel.	4. El sistema guarda el subnivel.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
Flujo Normal de Eventos	
Sección Eliminar Ubicación Física	
Acción del Actor	Respuesta del Sistema
1. El usuario lógico selecciona la opción eliminar nivel.	2. El sistema le muestra una lista de niveles.
3. El usuario físico selecciona el nivel que va a eliminar.	4. El sistema elimina el nivel.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
Prioridad	Crítica

Capítulo 2: Descripción de la solución propuesta.

2.7.5 Tabla 2.6 Caso de uso Gestionar Caja.

CU-5	Gestionar Caja	
Actores:	Usuario físico (Inicia)	
Resumen:	El caso de uso inicia cuando el usuario físico invoca las operaciones relacionada con gestionar caja, tales como Ubicar en caja, adicionar caja, cerrar caja. El sistema devuelve un mensaje diciendo si se realizó la operación terminando así el caso de uso.	
Referencias:	RF5(RF5.1, RF5.2, RF5.3)	
Flujo Normal de Eventos		
Sección Ubicar en caja		
Acción del Actor	Respuesta del Sistema	
1. El usuario físico selecciona lo opción ubicar en caja.	2. El sistema le muestra una lista cajas.	
3. El usuario lógico selecciona la caja en la que va a guardar la ubicación física.	4. El sistema guarda la ubicación en la caja seleccionada.	
Flujo Alternativo		
Acción del Actor	Respuesta del Sistema	
Flujo Normal de Eventos		
Sección Adicionar caja		
Acción del Actor	Respuesta del Sistema	

Capítulo 2: Descripción de la solución propuesta.

1. El usuario físico selecciona la opción adicionar caja.	2. El sistema le pide que entre el número de la caja.
3. El usuario entra el número de la caja.	4. El sistema adiciona la caja.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
Flujo Normal de Eventos	
Sección Cerrar Caja	
Acción del Actor	Respuesta del Sistema
1. El usuario físico selecciona la opción cerrar caja.	2. El sistema le muestra una lista de cajas.
3. El usuario lógico selecciona la caja que va a cerrar.	4. El sistema cierra la caja.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
Prioridad	Crítica

Capítulo 2: Descripción de la solución propuesta.

2.7.6 Tabla 2.7 Caso de uso Buscar archivo.

CU-6	Buscar Archivo	
Propósito:	Permite al usuario lógico eliminar un nivel.	
Actores:	Usuario externo (Inicia)	
Resumen:	El caso de uso inicia cuando el usuario necesita tener alguna información sobre algún archivo, este se autentifica y procede a buscar la información desea por algún criterio de búsqueda. El caso de uso termina cuando el usuario se marcha con la información buscada.	
Referencias:	RF6(RF6.1,6.2,6.3,6.4,6.5,6.6)	
Flujo Normal de Eventos		
Sección Buscar por fecha		
Acción del Actor	Respuesta del Sistema	
1. El usuario selecciona la opción buscar.	2. El sistema le pide que entre la fecha.	
3. El usuario entra la fecha.	4. El sistema verifica que existan archivos en esa fecha.	
	5. El sistema busca el archivo.	
Flujo Alternativo		
Acción del Actor	Respuesta del Sistema	
	6. Si no existe ningún archivo en esa fecha, muestra un mensaje, no existe archivo en esa fecha.	

Capítulo 2: Descripción de la solución propuesta.

Sección Buscar por Título	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción buscar.	2. El sistema le pide que entre el título.
3. El usuario entra el título.	4. El sistema verifica que verifica que existan archivos con ese título.
	5. El sistema busca el archivo.
	6. El sistema muestra el archivo.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
	7. Si no existe ningún archivo en esa fecha, muestra un mensaje, no existe archivo con ese título.
Sección Buscar por nombre del Productor	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción buscar.	2. El sistema le pide que entre el nombre del productor.
3. El usuario entra el nombre del productor.	4. El sistema verifica que verifica que existan archivos con ese nombre de productor.
	5. El sistema busca el archivo.

Capítulo 2: Descripción de la solución propuesta.

		6. El sistema muestra el archivo.
Flujo Alternativo		
Acción del Actor		Respuesta del Sistema
		7. Si no existe ningún archivo en esa fecha, muestra un mensaje, no existe archivo que contenga ese nombre del productor.
Prioridad	Crítica	

2.7.7 Tabla 2.8 Caso de uso Autenticar usuario.

CU-7	Autenticar usuario
Propósito:	Permite al usuario anónimo acceder a la interfaz de administración.
Actores:	Usuario anónimo (Inicia)
Resumen:	El caso de uso inicia cuando el usuario anónimo desea acceder a la sección administrativa del sistema y proporciona su nombre y contraseña. El sistema verifica si son válidos, de no suceder así, le muestra un mensaje de error y le da la opción de acceder nuevamente, de lo contrario le da acceso al sistema. El caso de uso finaliza cuando el usuario anónimo se ha autenticado.
Referencias:	RF7
Precondiciones:	El usuario debe estar registrado.

Capítulo 2: Descripción de la solución propuesta.

Poscondiciones:	El administrador debe estar autenticado.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El usuario anónimo selecciona la opción de autenticarse.	2. El sistema solicita los datos necesarios (usuario/contraseña).	
3. El usuario anónimo introduce sus datos.	4. El sistema verifica sus datos. 5. El sistema lo redirecciona a la interfaz de usuario.	
Flujo Alternativo		
Acción del Actor	Respuesta del Sistema	
	6. Si el usuario no entra correctamente los datos se notifica un error.	
Prioridad	Crítica	

2.7.8 Tabla 2.9 Caso de uso Gestionar usuario.

CU-8	Gestionar usuario
Propósito:	Permite al administrador adicionar, actualizar, eliminar y registrar un usuario al sistema.
Actores:	Administrador (Inicia)
Resumen:	El caso de uso inicia cuando el administrador elige una de las opciones (adicionar, actualizar, eliminar o registrar) un usuario e introduce los datos del usuario. El caso de uso finaliza cuando se realice la opción escogida correctamente.

Capítulo 2: Descripción de la solución propuesta.

Referencias:	RF8(8.1, 8.2,8.3,8.4)	
Flujo Normal de Eventos		
Sección Adicionar Usuario		
Acción del Actor	Respuesta del Sistema	
1. El administrador selecciona la opción de adicionar un nuevo usuario.	2. El sistema solicita los datos necesarios (usuario, contraseña, nombre, apellidos).	
3. El administrador introduce sus datos.	4. El sistema verifica sus datos. 5. El sistema adiciona un nuevo administrador al sistema.	
Flujo Alternativo		
Acción del Actor	Respuesta del Sistema	
	6. Si el administrador quiere adicionar un usuario que ya existe el sistema le notifica un error.	
Sección Eliminar Usuario		
Acción del Actor	Respuesta del Sistema	
1. El administrador selecciona la opción de eliminar un usuario.	2. El sistema le muestra la lista de usuario del sistema.	
3. El administrador selecciona los usuarios que va a eliminar.	4. El sistema elimina los usuarios seleccionados.	
Flujo Alternativo		

Capítulo 2: Descripción de la solución propuesta.

Acción del Actor	Respuesta del Sistema
Sección Modificar Usuario	
Acción del Actor	Respuesta del Sistema
1. El administrador selecciona la opción de modificar un usuario.	2. El sistema le muestra la lista de usuario del sistema.
3. El administrador selecciona el usuario que va a modificar.	4. El sistema pide los datos del usuario.
5. el administrador entra los datos.	6. El sistema verifica los datos que no sean repetidos. 7. El sistema actualiza los datos del usuario seleccionado.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
Sección Registrar Usuario	
Acción del Actor	Respuesta del Sistema

Capítulo 2: Descripción de la solución propuesta.

1. El usuario anónimo accede por primera vez al sistema.	2. El sistema le brinda la posibilidad de registrarse.
3. El usuario anónimo introduce sus datos.	4. El sistema registra al usuario.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
Prioridad	Crítica

2.8 Análisis

2.8.1 Diagrama de clase del análisis. Gestionar Descripción.

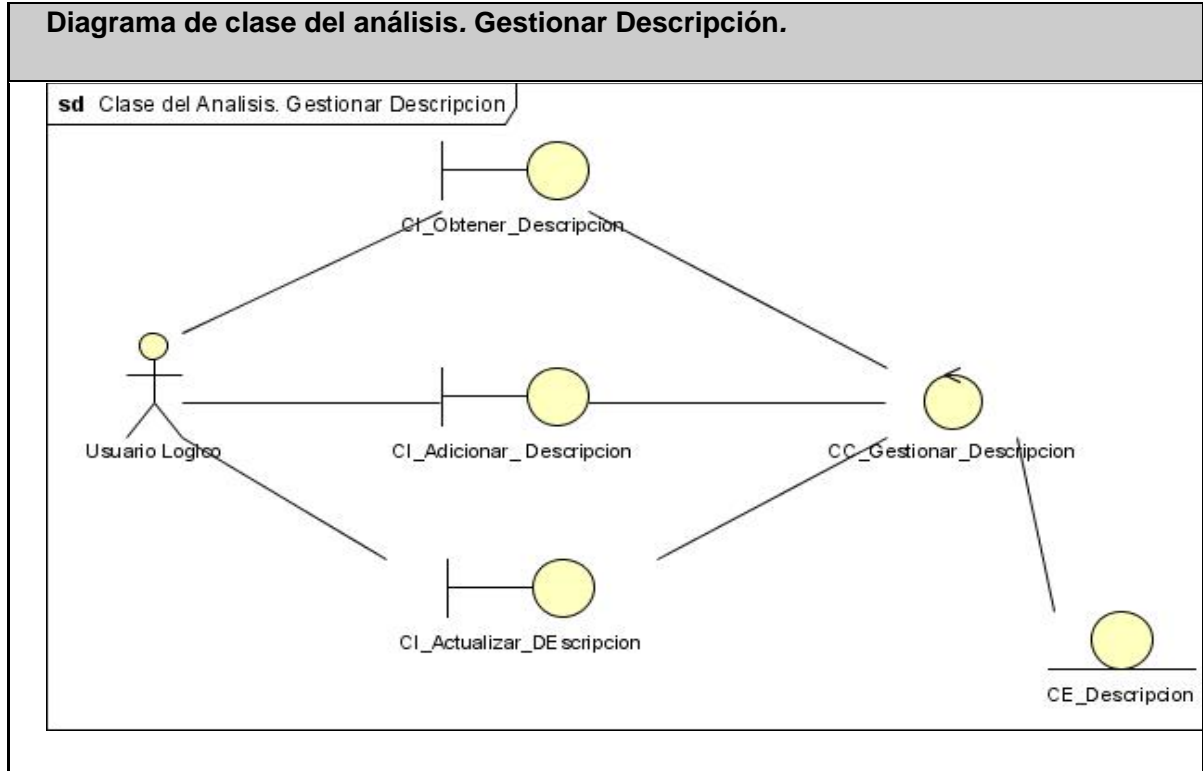


Fig. 2-2. Diagrama de clase del análisis. Gestionar Descripción.

2.8.2 Diagrama de clase del análisis. Gestionar Ubicación Física.

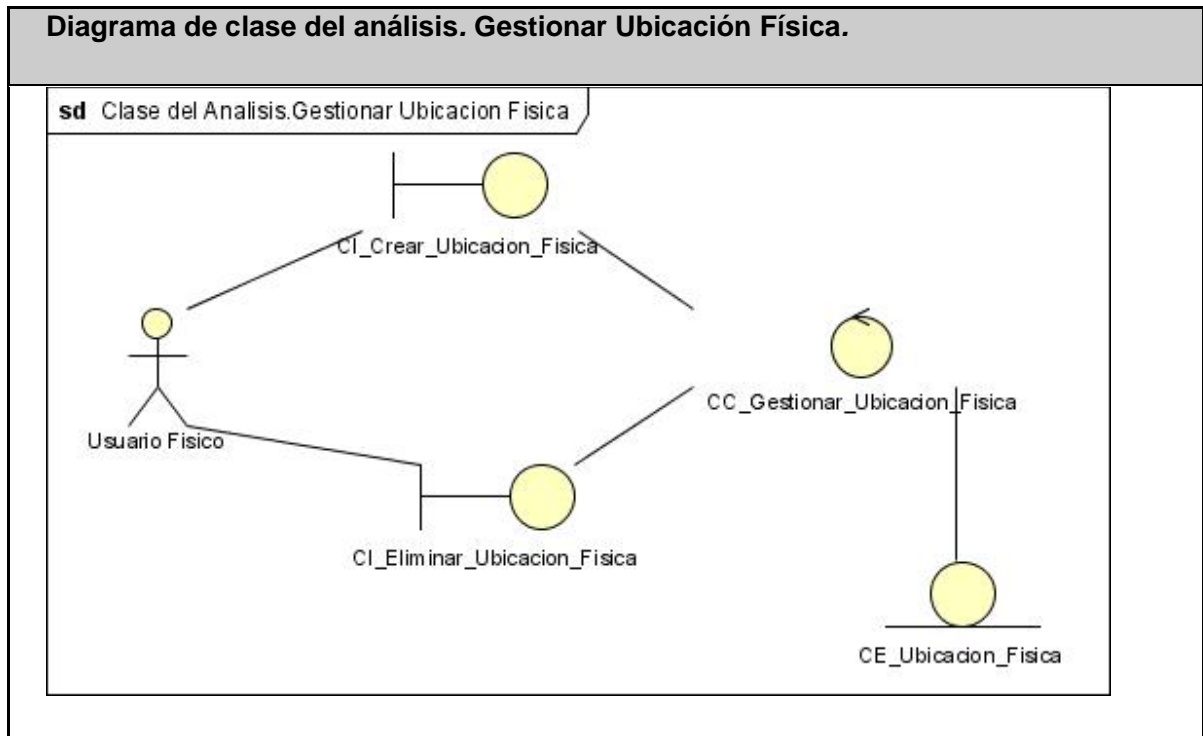


Fig. 2-3. Diagrama de clase del análisis. Gestionar Ubicación Física.

2.8.3 Diagrama de clase del análisis. Gestionar ubicación Lógica.

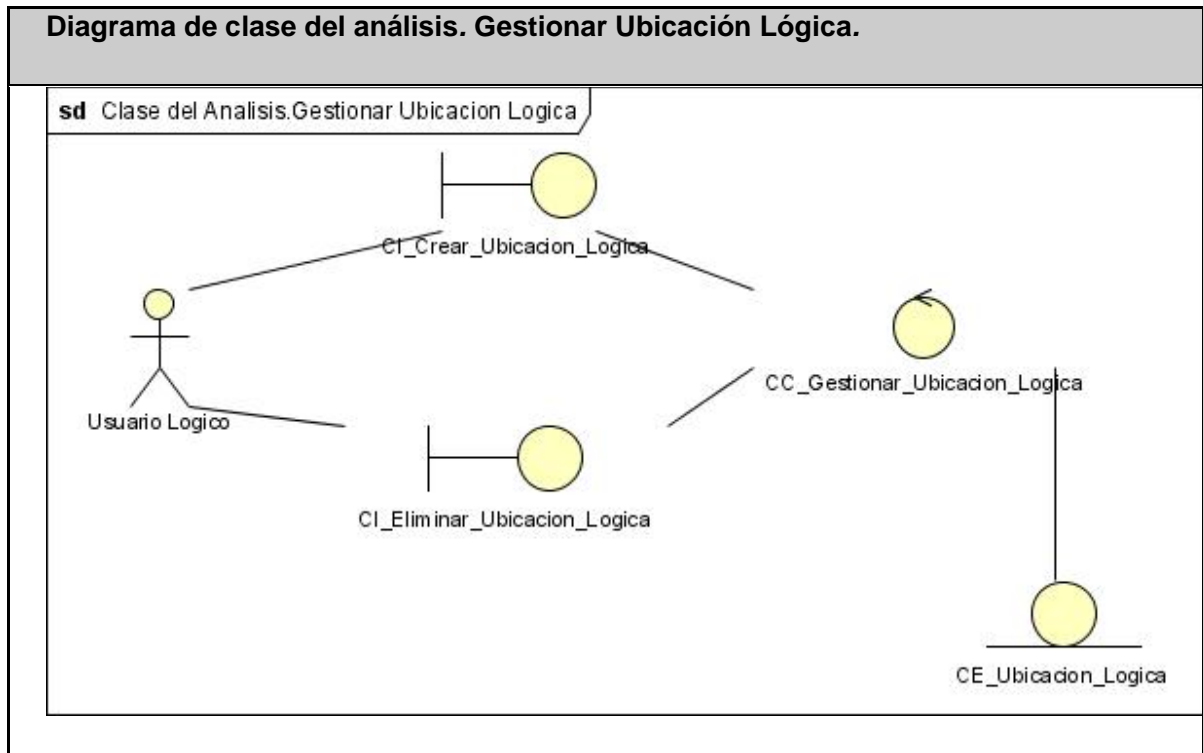


Fig. 2-4. Diagrama de clase del análisis. Gestionar Ubicación Lógica.

2.8.4 Diagrama de clase del análisis. Gestionar Caja.

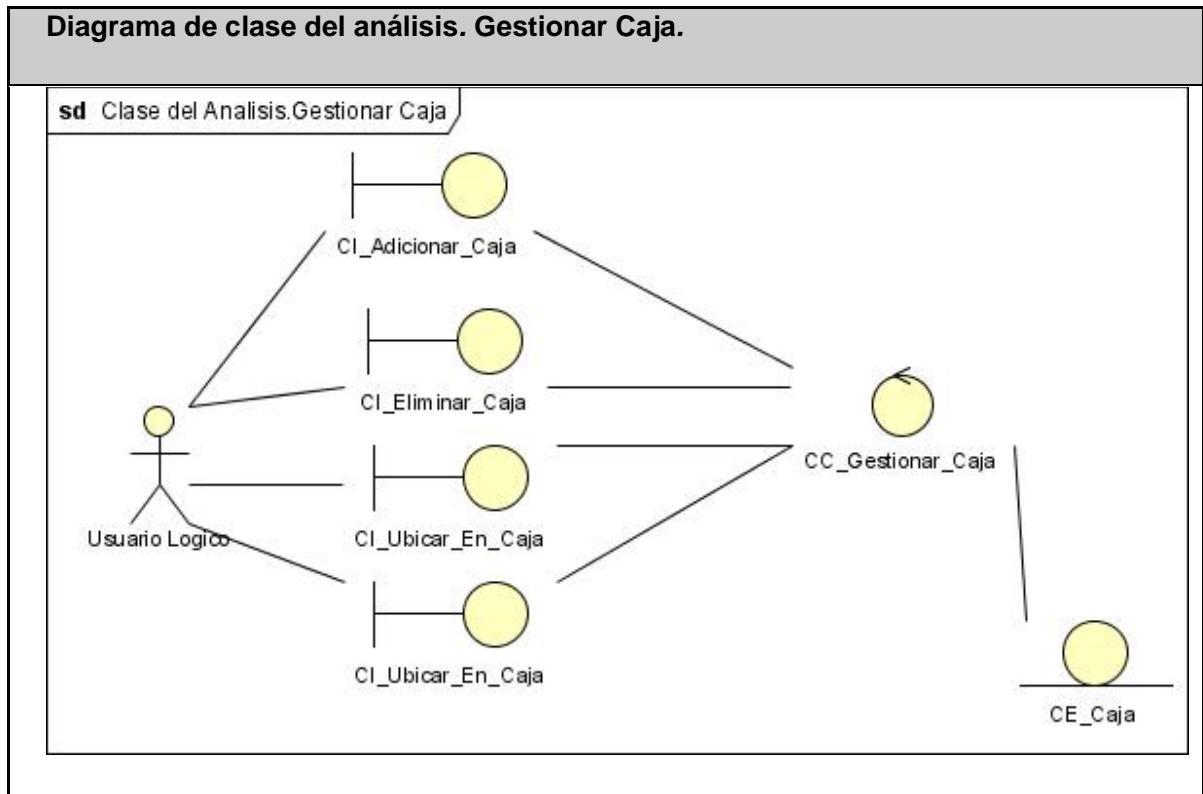


Fig. 2-5. Diagrama de clase del análisis. Gestionar Caja.

2.8.5 Diagrama de clase del análisis. Autenticar Usuario.

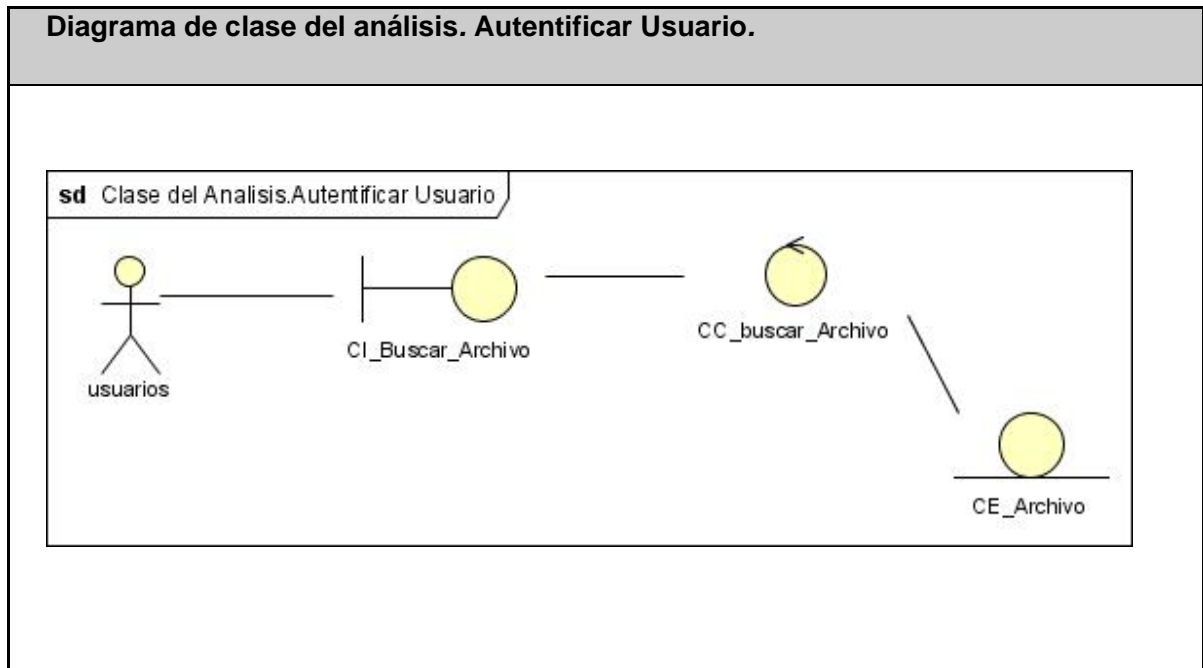


Fig. 2-6. Diagrama de clase del análisis. Autenticar Usuario.

2.8.6 Diagrama de clase del análisis. Gestionar Usuario.

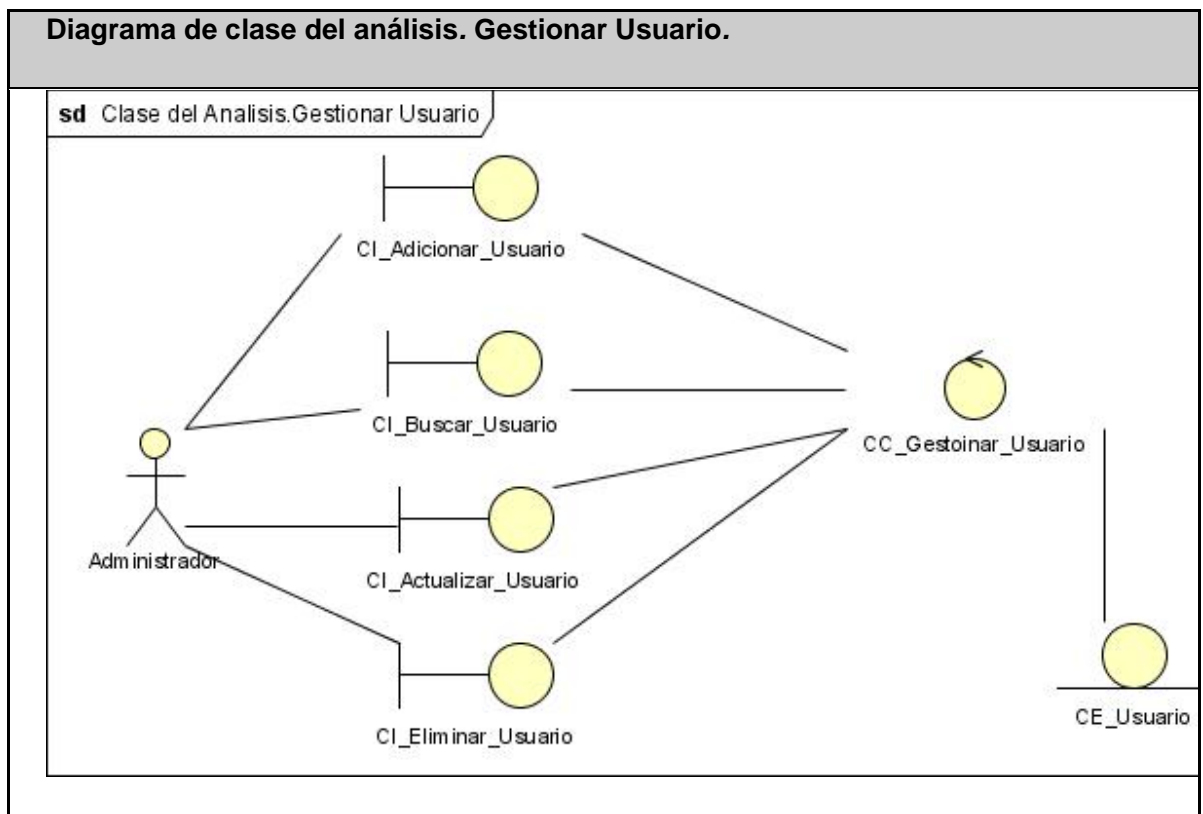


Fig. 2-7. Diagrama de clase del análisis. Gestionar Usuario.

2.8.7 Diagrama de clase del análisis. Subir Archivo.

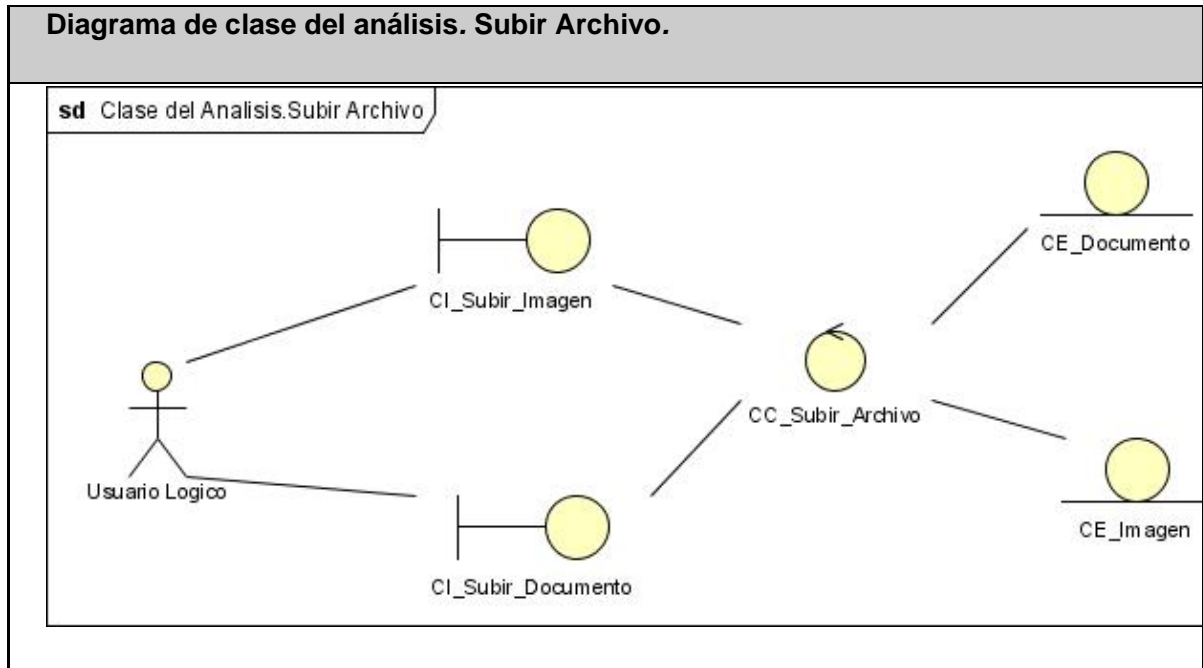


Fig. 2-8. Diagrama de clase del análisis. Subir Archivo.

2.9 Diseño

Para una mejor comprensión y dada la importancia que tiene la fase de diseño, se deben utilizar mecanismos de diseño, artefacto de RUP que agrupa un conjunto de clases de diseño, colaboraciones e incluso subsistema del modelo de diseño que lleva a cabo requisitos comunes que persisten, distribución, seguridad y funcionamiento. Para la documentación de estos mecanismos se realizaron diagramas de vista estática (diagrama de clase de del diseño Web) y diagramas de interacción (Colaboración).

2.9.1 Diagrama de clase. Gestionar Descripción.

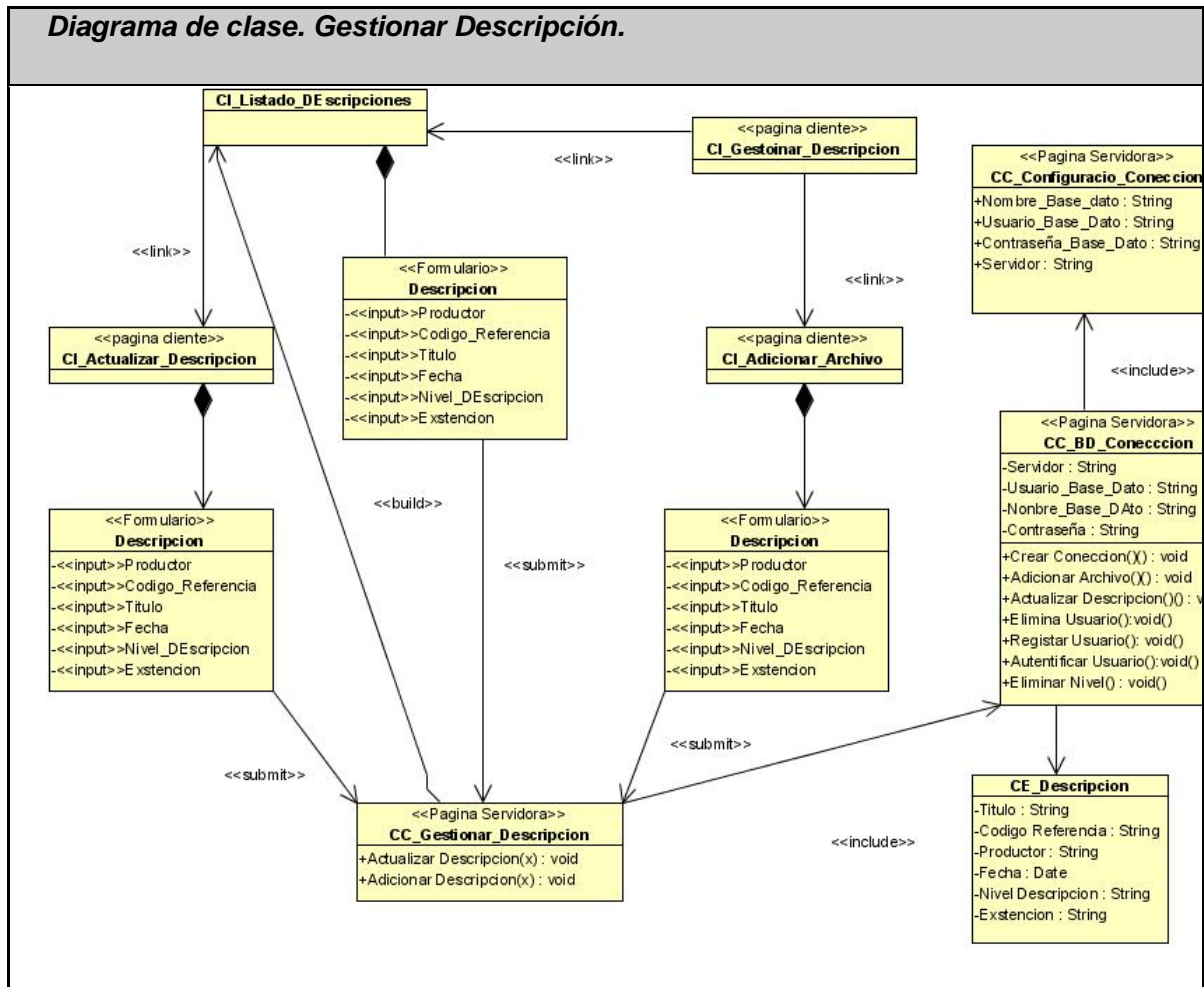


Fig. 2-9. Diagrama clase. Gestionar descripción.

2.9.2 Diagrama de clase. Crear Ubicación Lógica

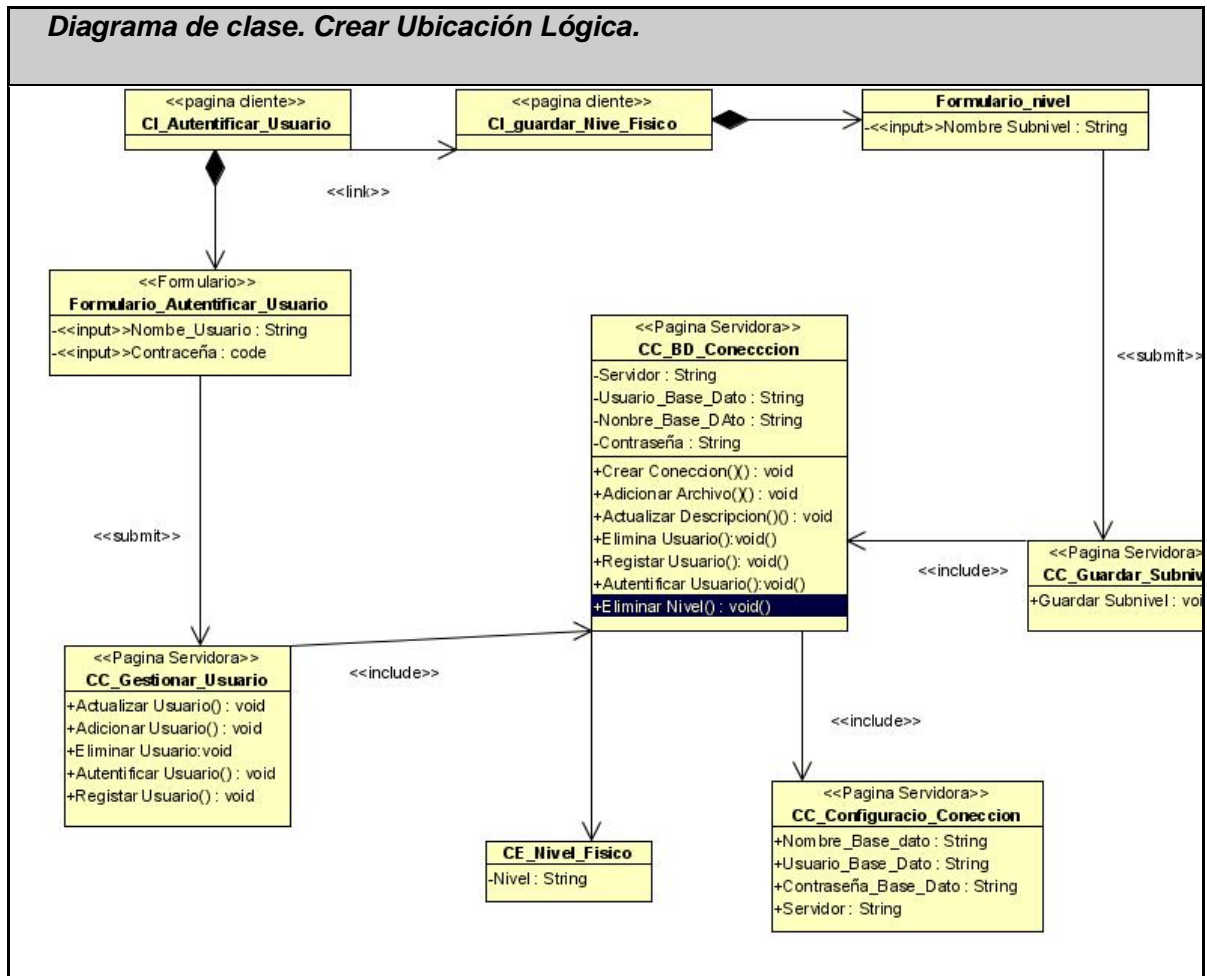


Fig. 2-10. Diagrama clase. Crear Ubicación Lógica

2.9.3 Diagrama de clase. Asignar Imagen

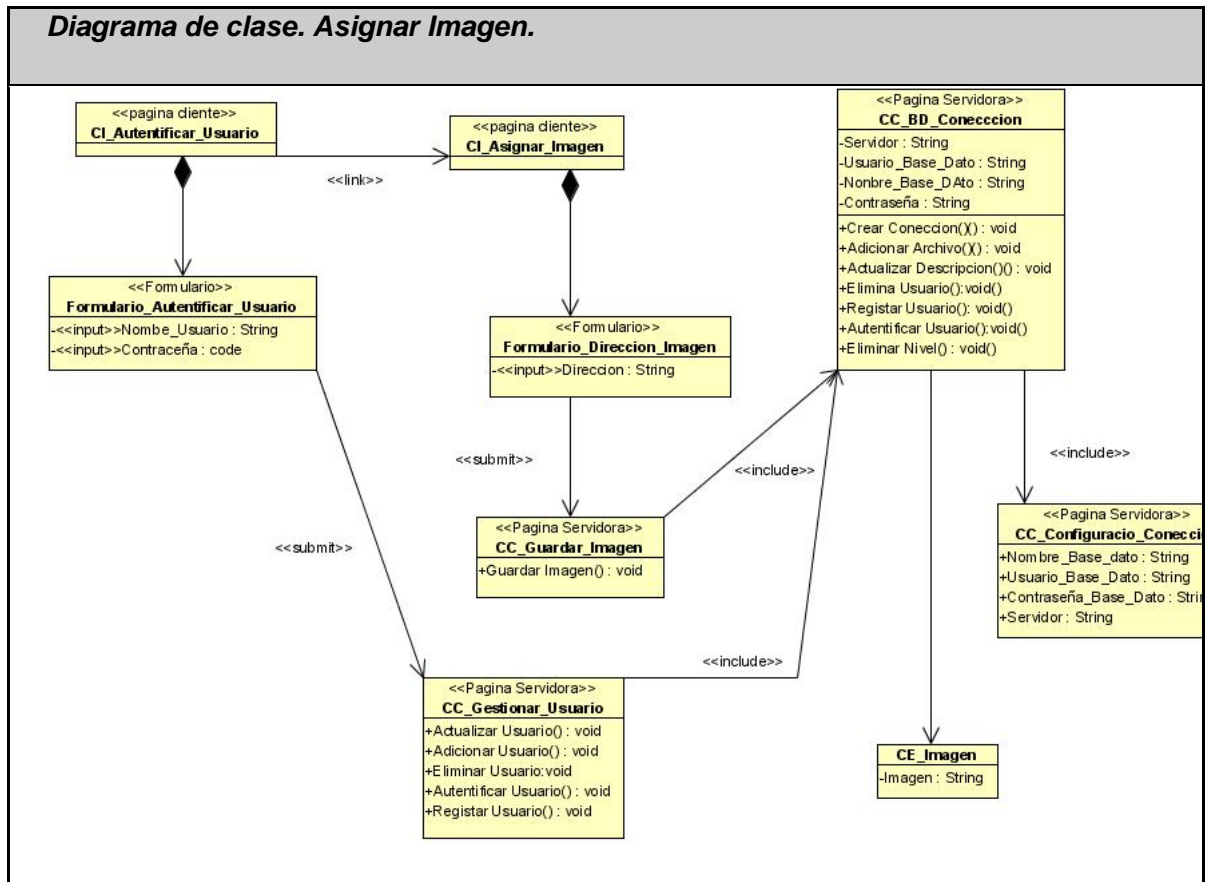


Fig. 2-11. Diagrama clase. Asignar imagen

2.9.4 Diagrama de clase. Eliminar Nivel Lógico

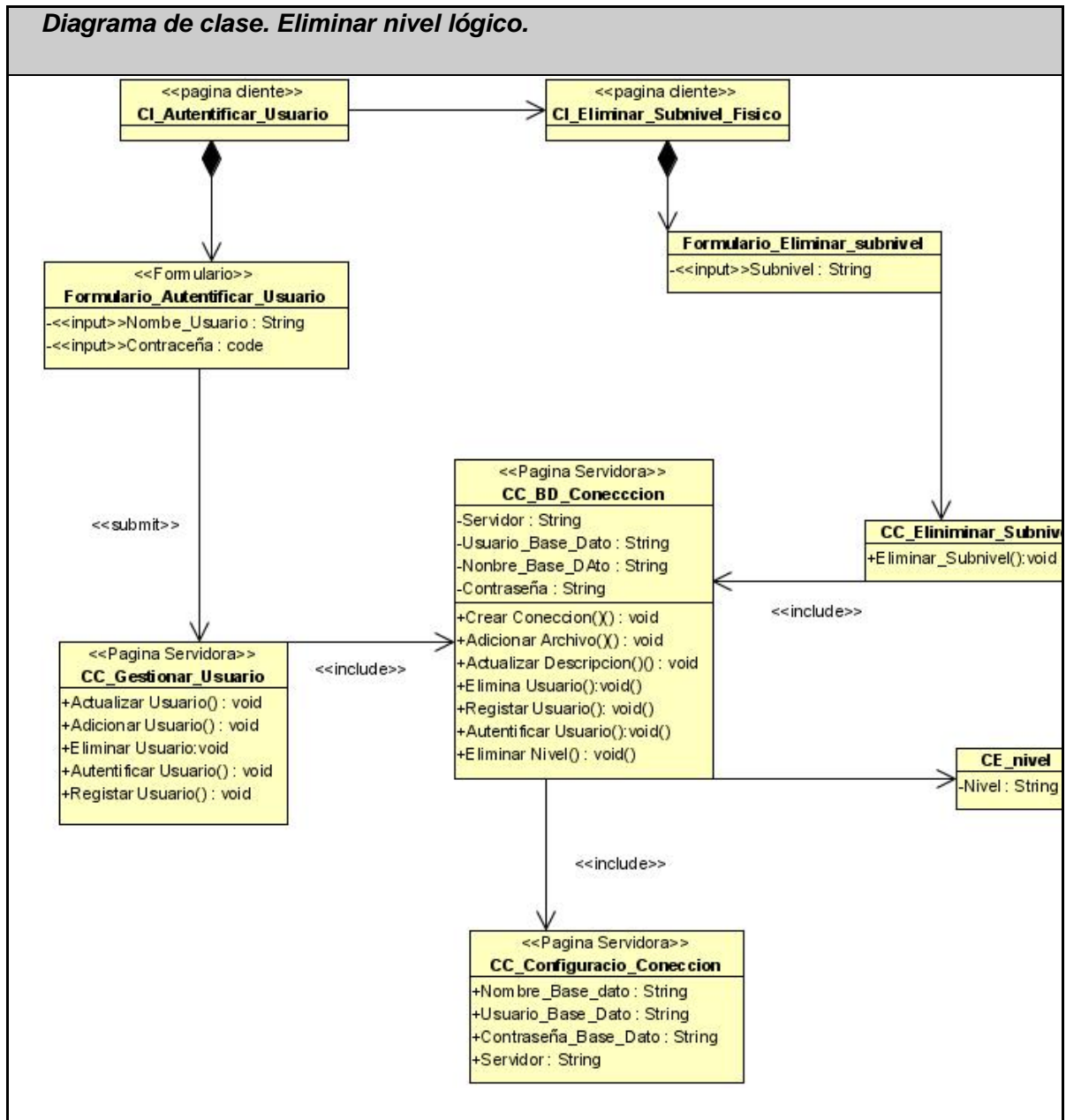


Fig. 2-12. Diagrama clase. Eliminar Nivel Lógico.

2.9.5 Diagrama de clase. Crear nivel Físico

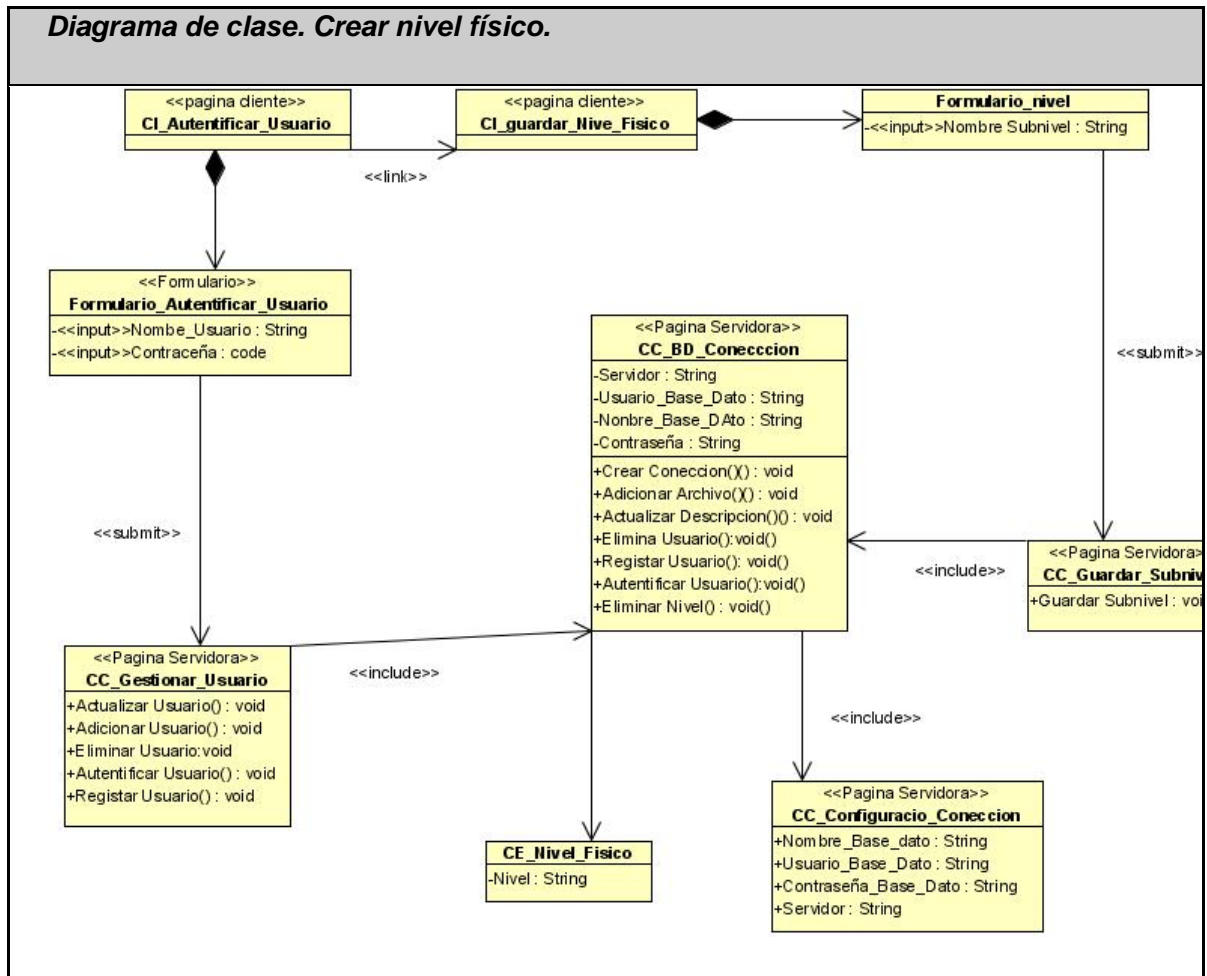


Fig. 2-13. Diagrama clase. Crear nivel físico.

2.9.6 Diagrama de clase. Eliminar nivel Físico.

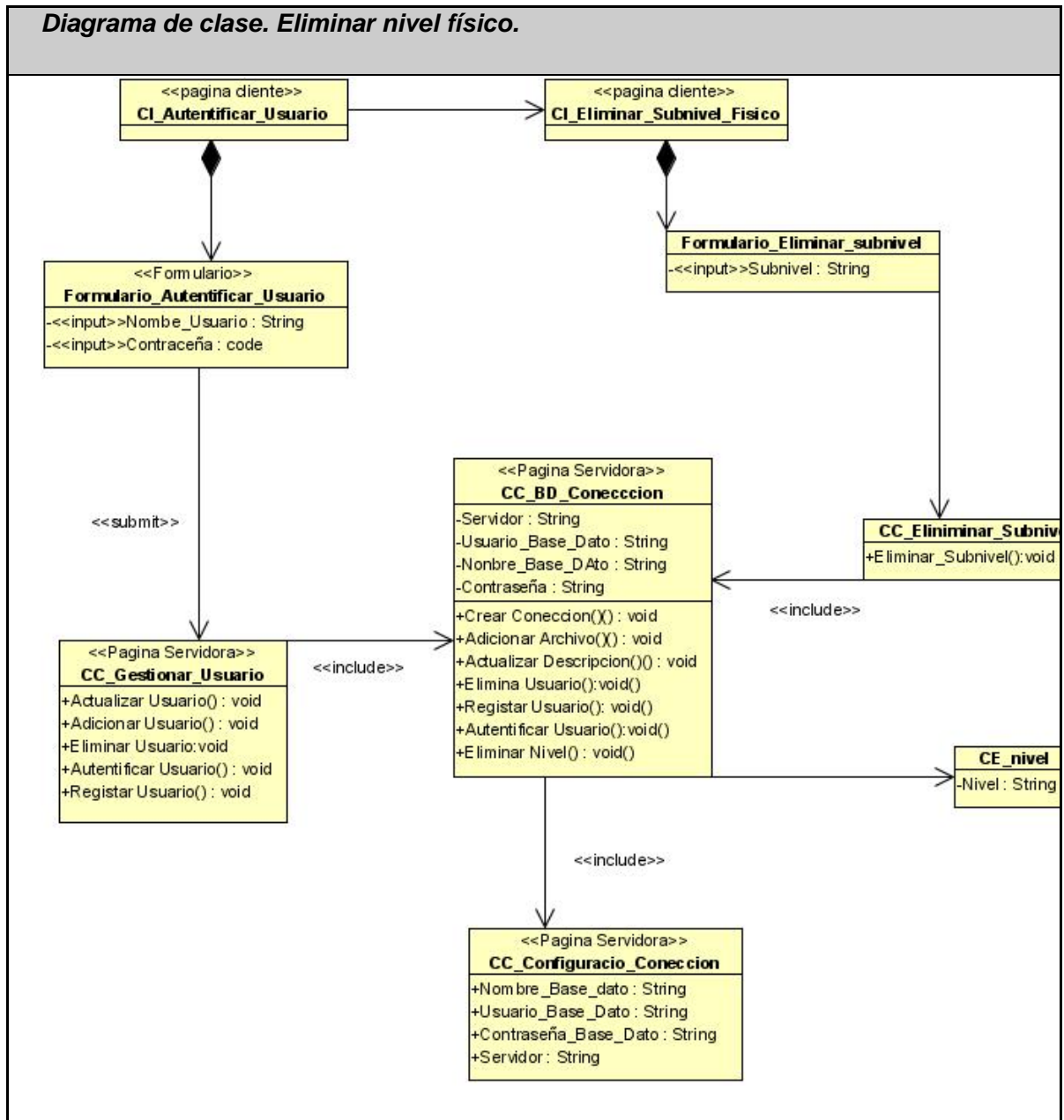


Fig. 2-14. Diagrama clase. Crear nivel físico.

2.9.7 Diagrama de clase. Ubicar en caja.

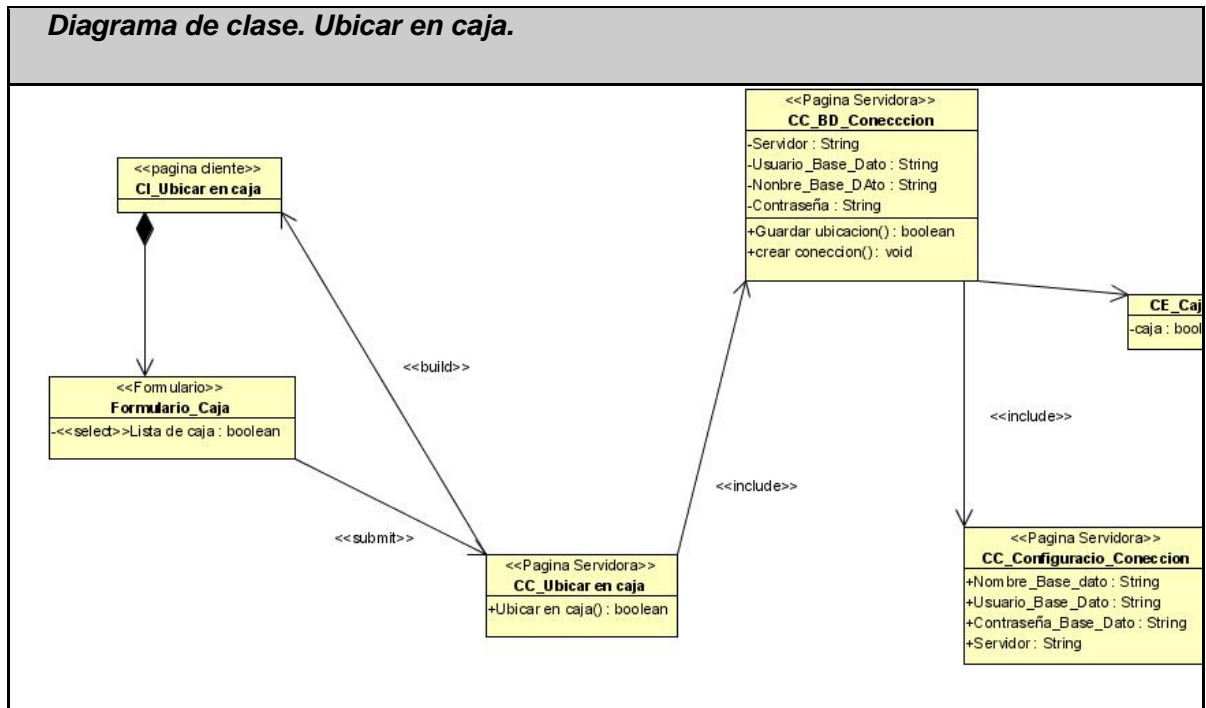


Fig. 2-15. Diagrama clase. Ubicar en caja.

2.9.8 Diagrama de clase. Buscar archivo.

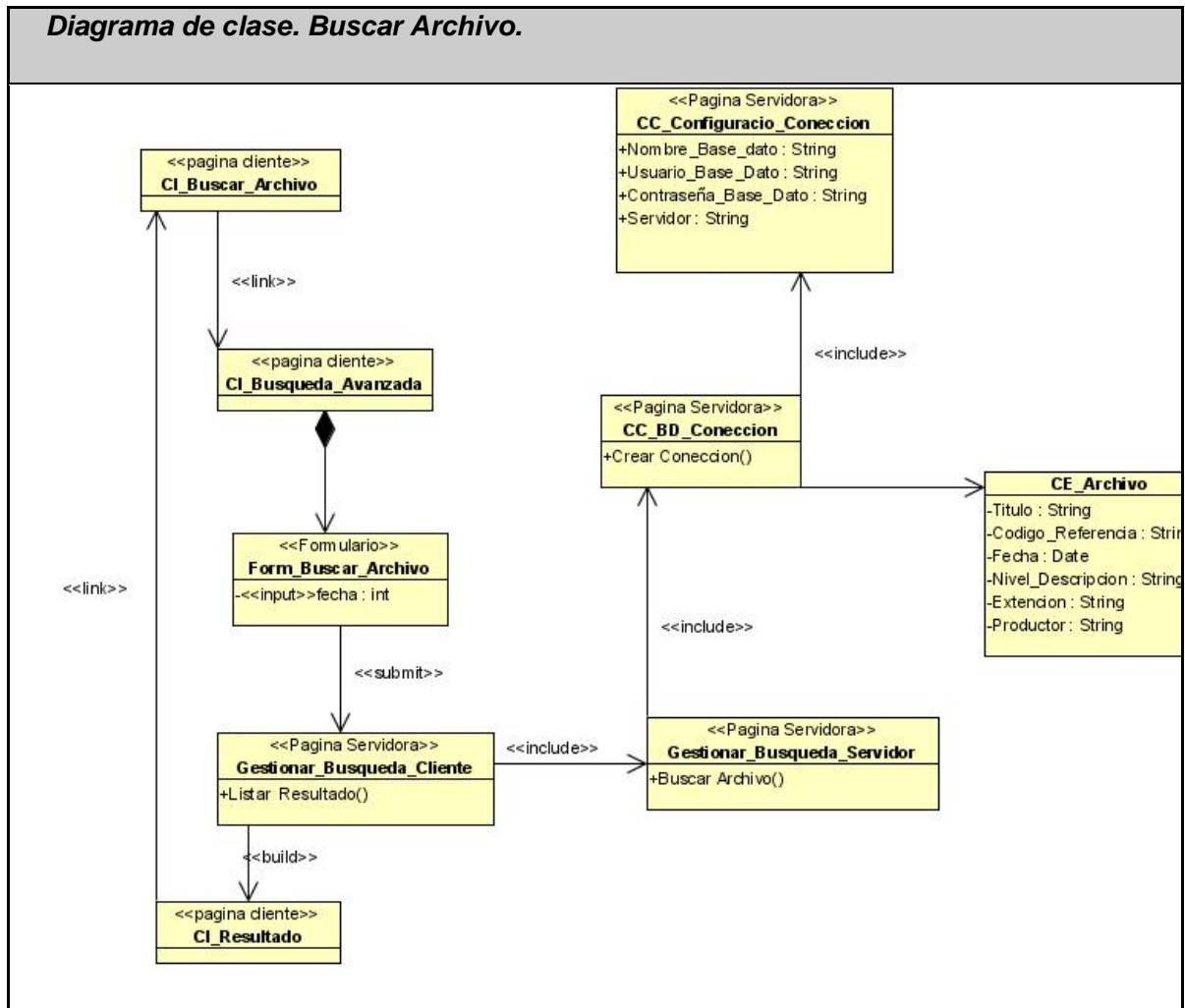


Fig. 2-16. Diagrama clase. Buscar Archivo.

Capítulo 2: Descripción de la solución propuesta.

2.9.9 Diagrama de clase. Autenticar usuario.

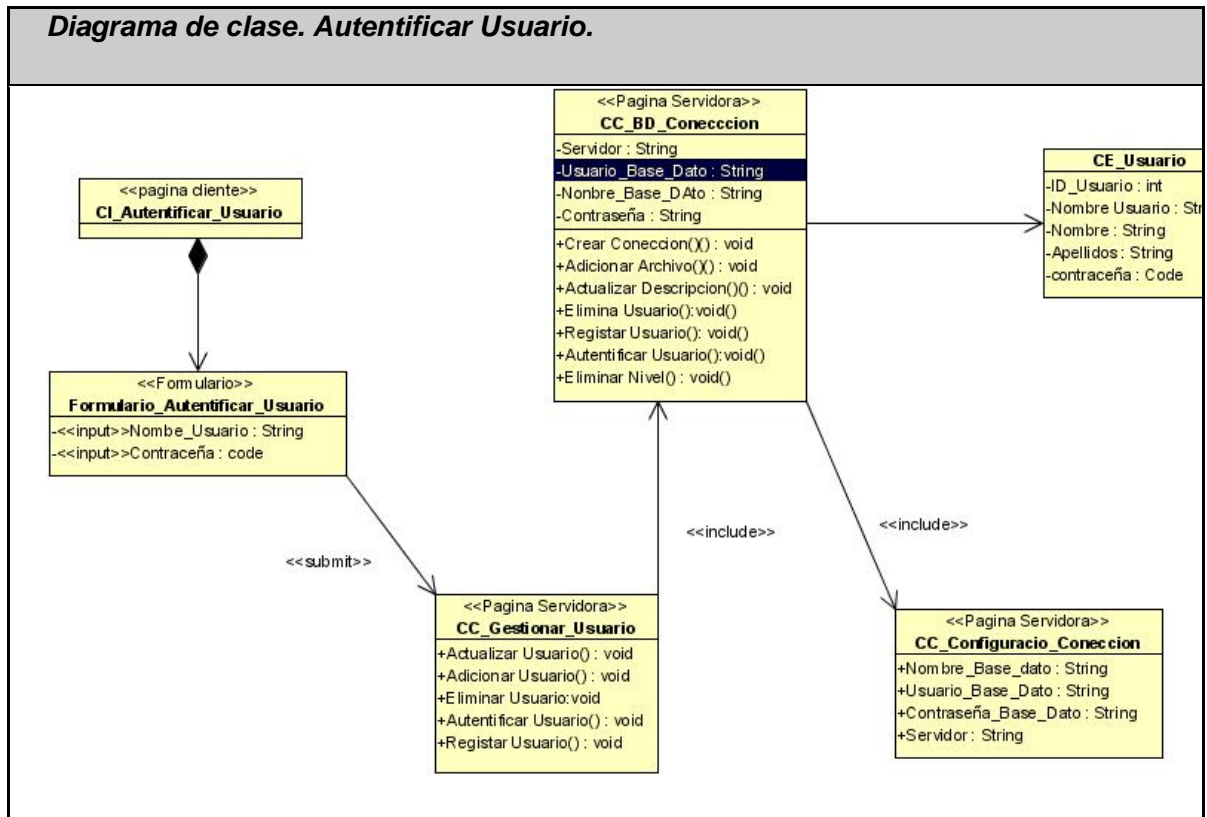


Fig. 2-17. Diagrama clase. Autenticar usuario.

2.9.10 Diagrama de clase. Gestionar usuario

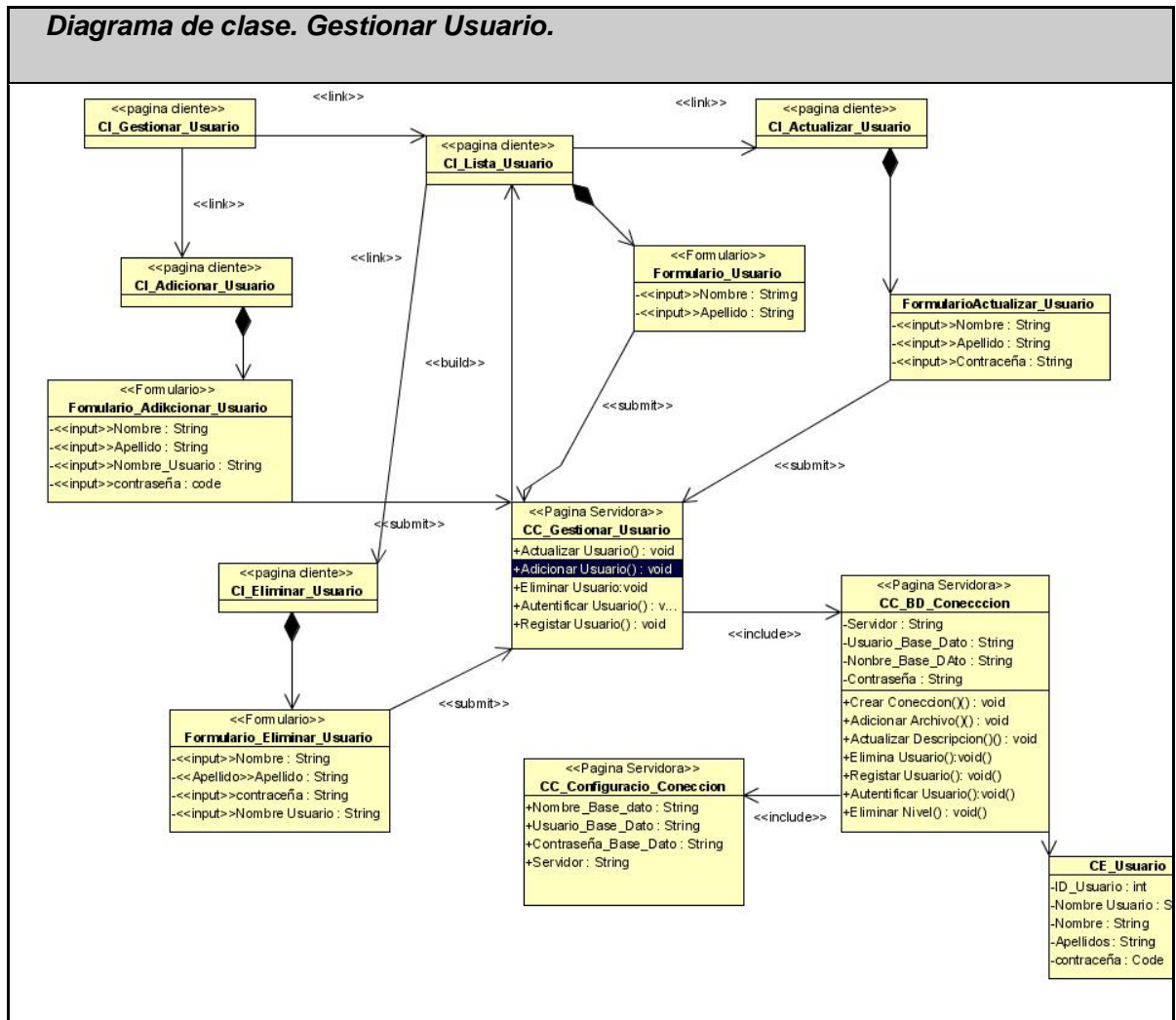


Fig. 2-18. Diagrama clase. Gestionar usuario.

2.9.11 Diagrama de clase. Registrar usuario.

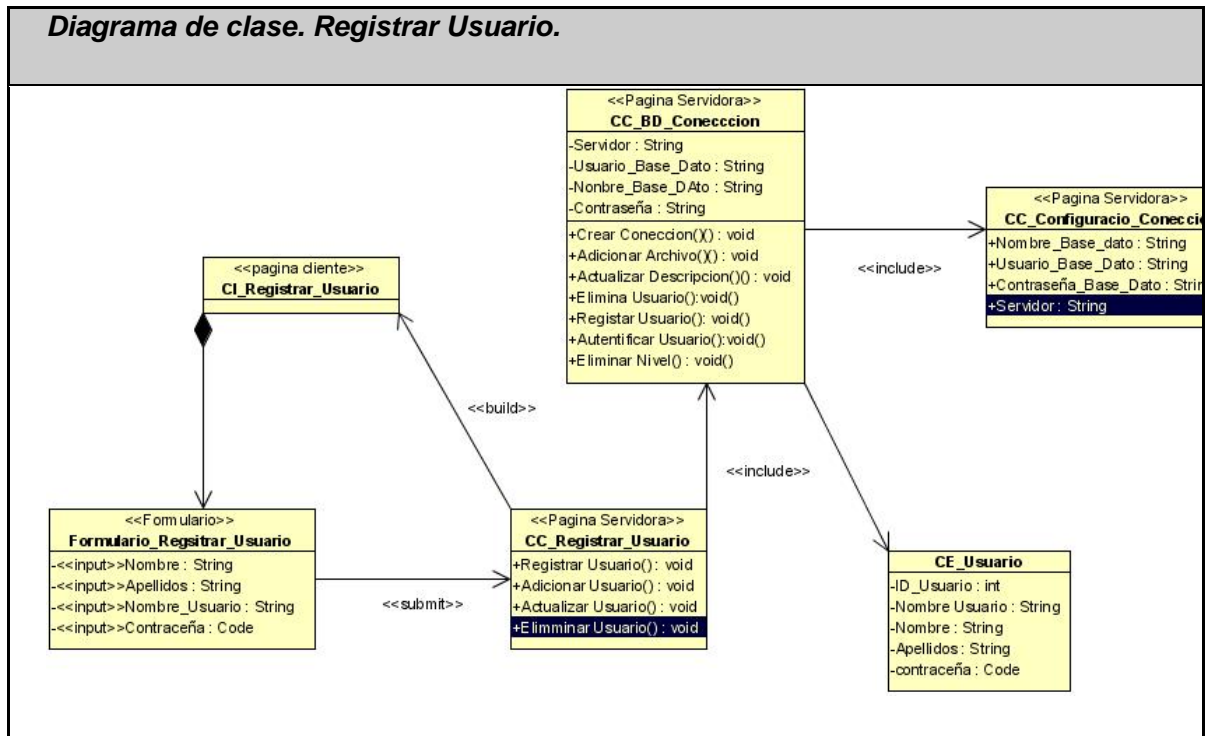


Fig. 2-19. Diagrama clase. Registrar usuario.

2.10 Diagramas de colaboración.

Estos Diagramas muestran las interacciones entre los objetos organizadas en una secuencia temporal. En particular muestran los objetos participantes en la interacción y la secuencia de mensajes intercambiados. Representan una interacción, un conjunto de comunicaciones entre objetos organizadas visualmente.

2.10.1 Diagrama de colaboración. Adicionar Descripción.

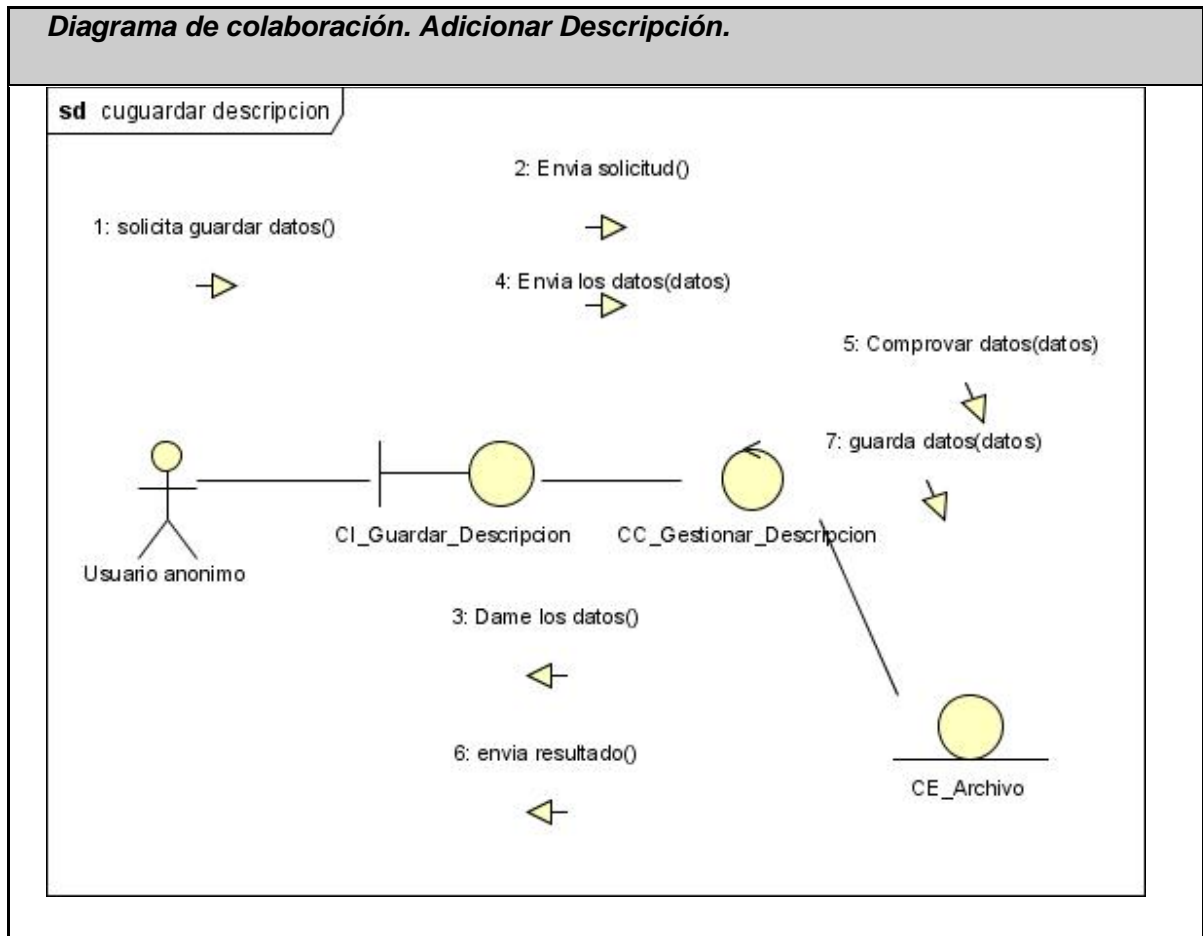


Fig. 2-20. Diagrama colaboración. Adicionar descripción.

2.10.2 Diagrama de colaboración. Actualizar Descripción.

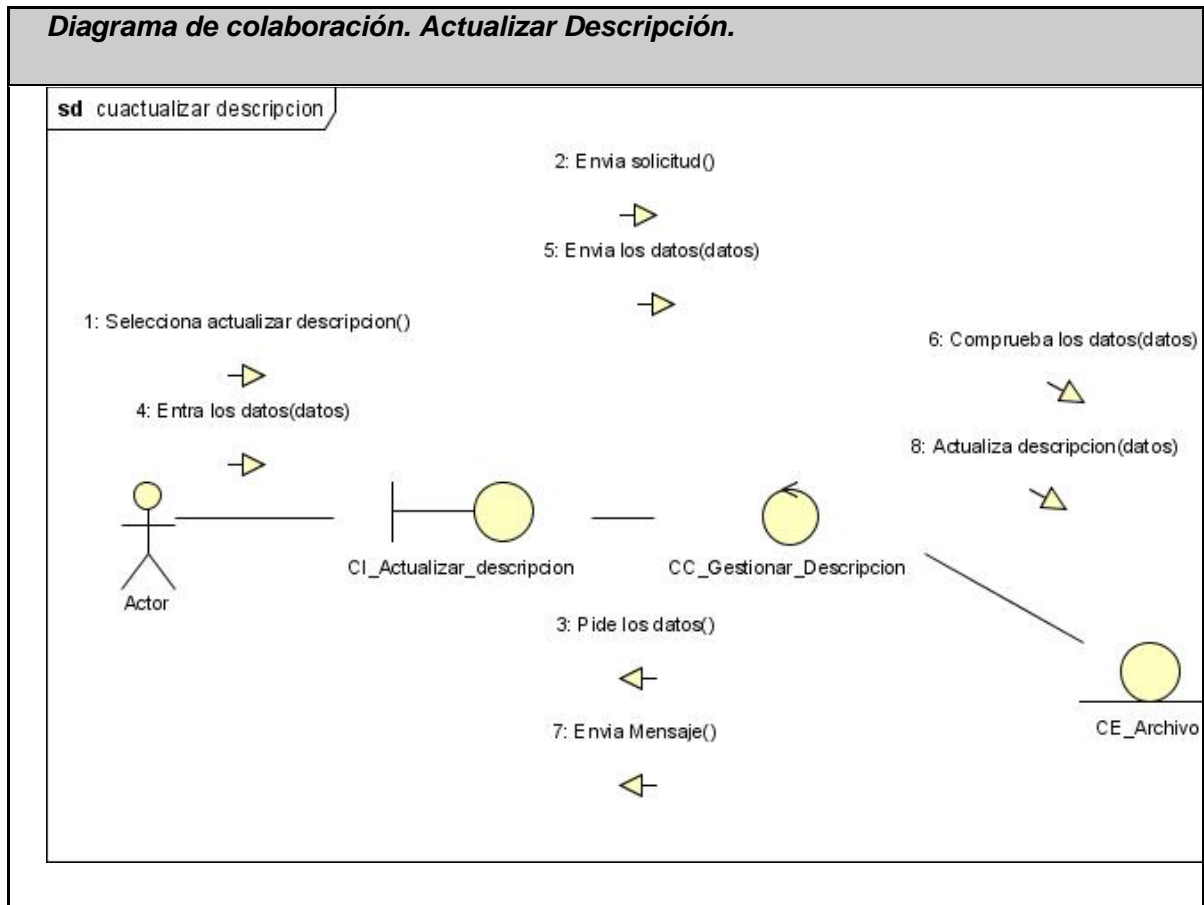


Fig. 2-21. Diagrama colaboración. Actualizar descripción.

2.10.3 Diagrama de colaboración. Crear ubicación lógica.

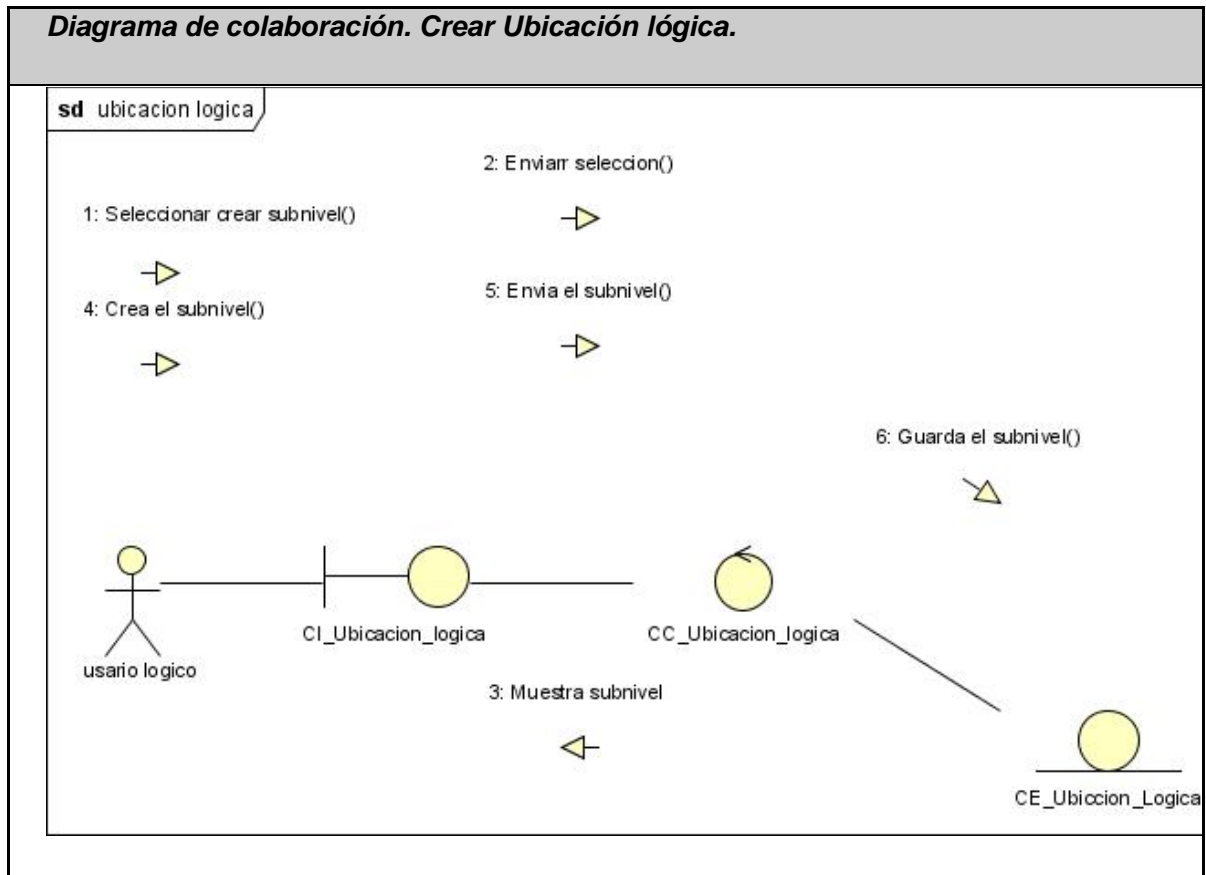


Fig. 2-22. Diagrama colaboración. Crear Ubicación lógica.

2.10.4 Diagrama de colaboración. Eliminar ubicación lógica.

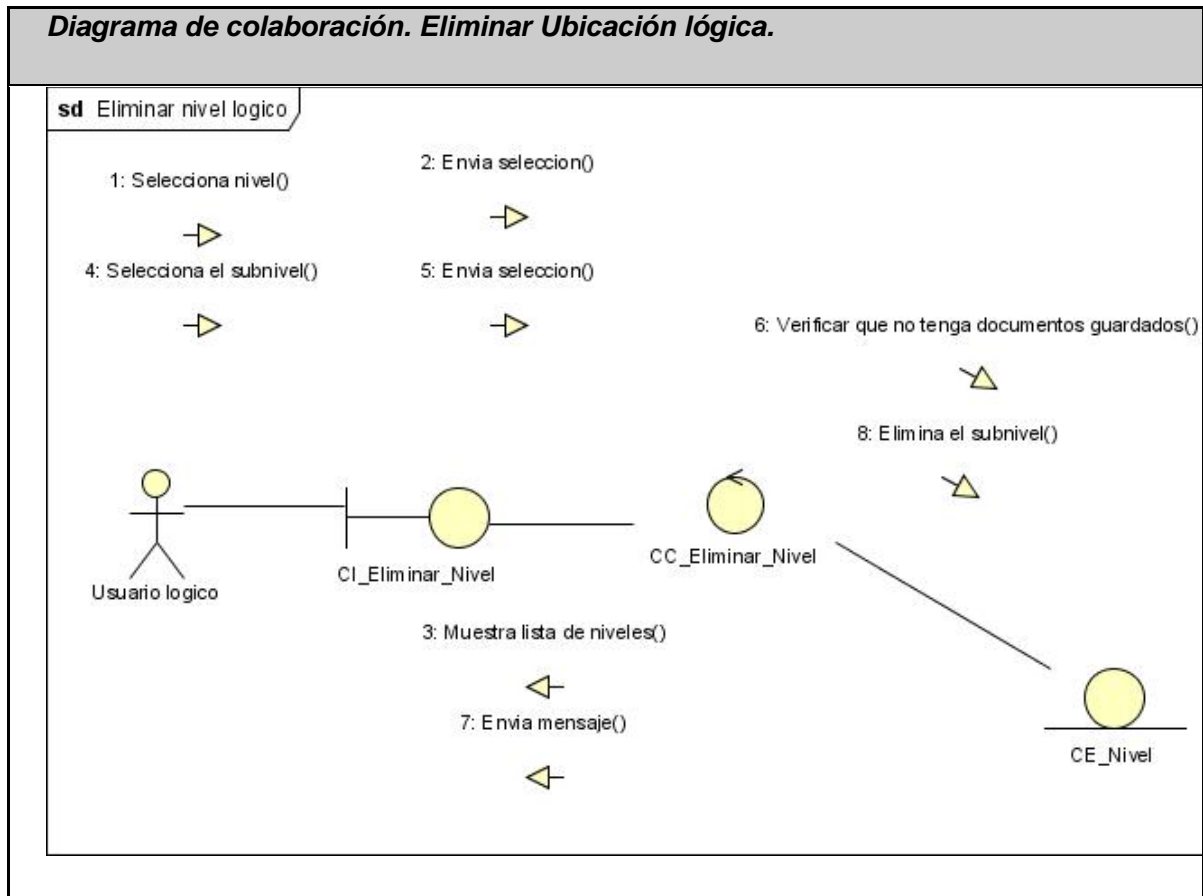


Fig. 2-23. Diagrama colaboración. Eliminar Ubicación lógica.

2.10.5 Diagrama de colaboración. Crear ubicación física.

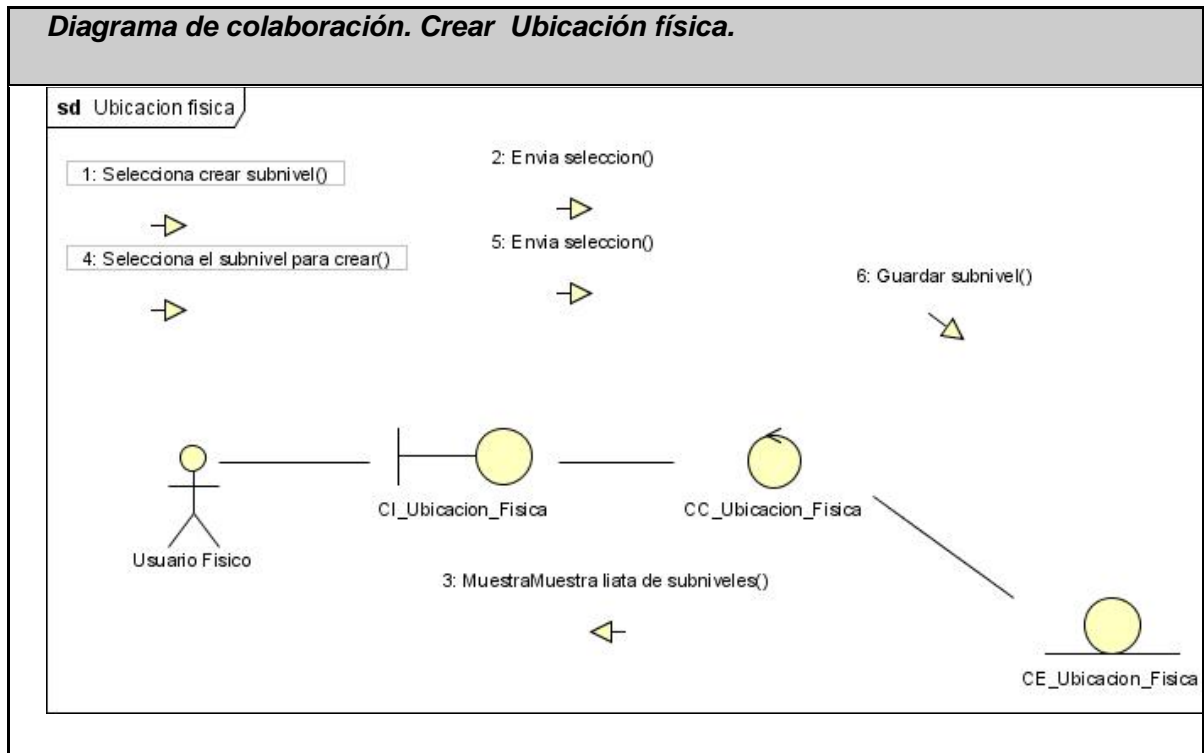


Fig. 2-24. Diagrama colaboración. Crear Ubicación física.

2.10.6 Diagrama de colaboración. Eliminar ubicación física.

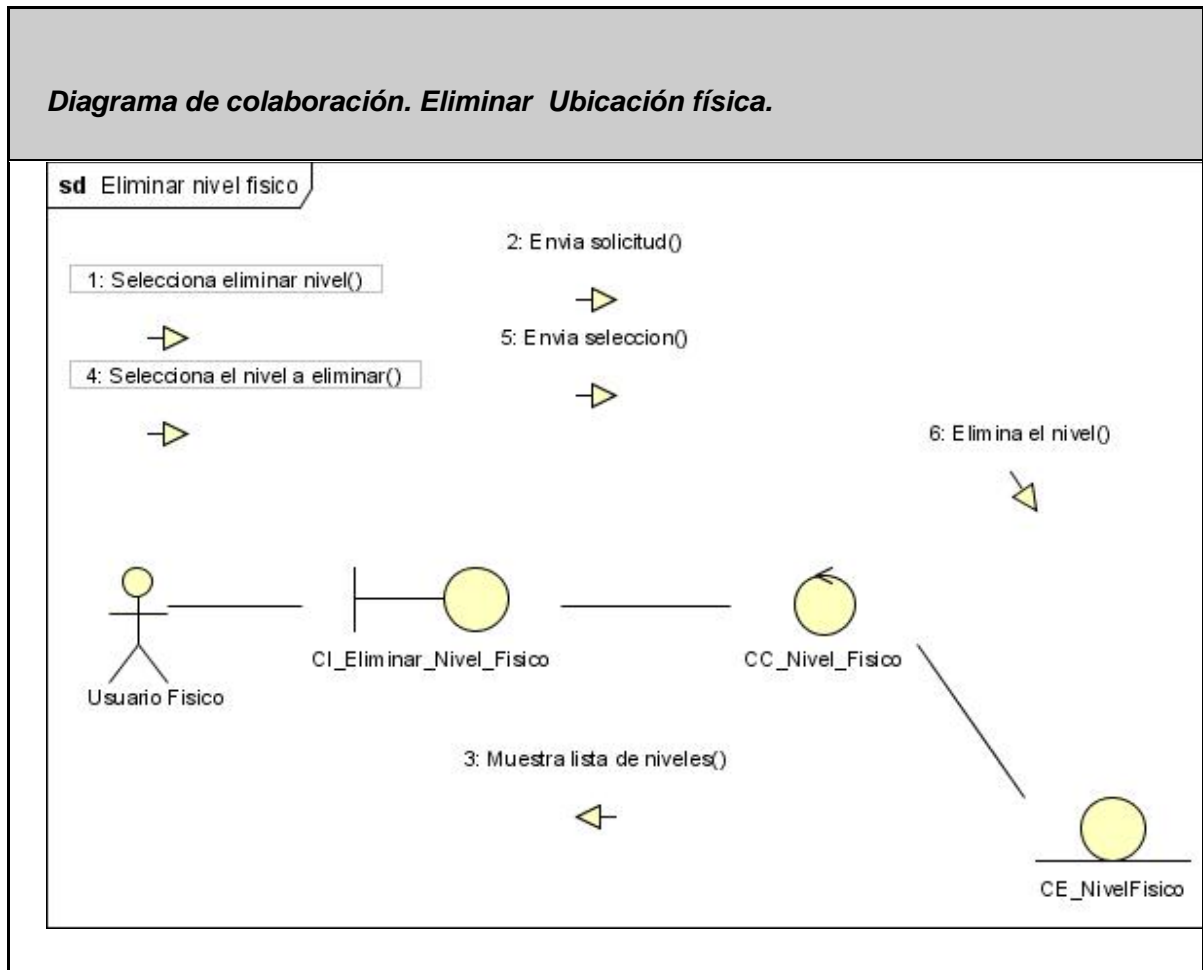


Fig. 2-25. Diagrama colaboración. Eliminar Ubicación física.

2.10.7 Diagrama de colaboración. Buscar archivo.

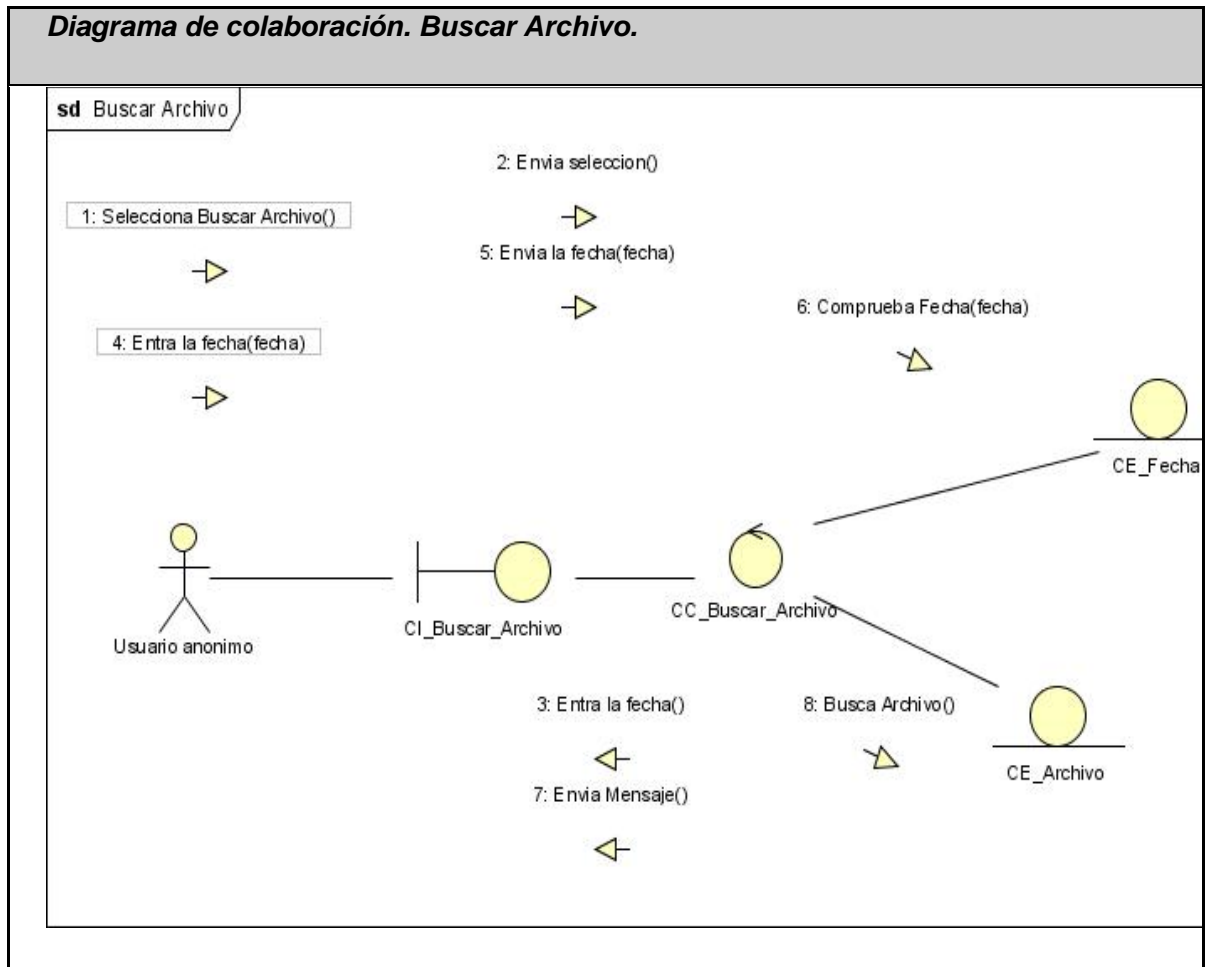


Fig. 2-26. Diagrama colaboración. Buscar archivo.

2.10.8 Diagrama de colaboración. Asignar imagen.

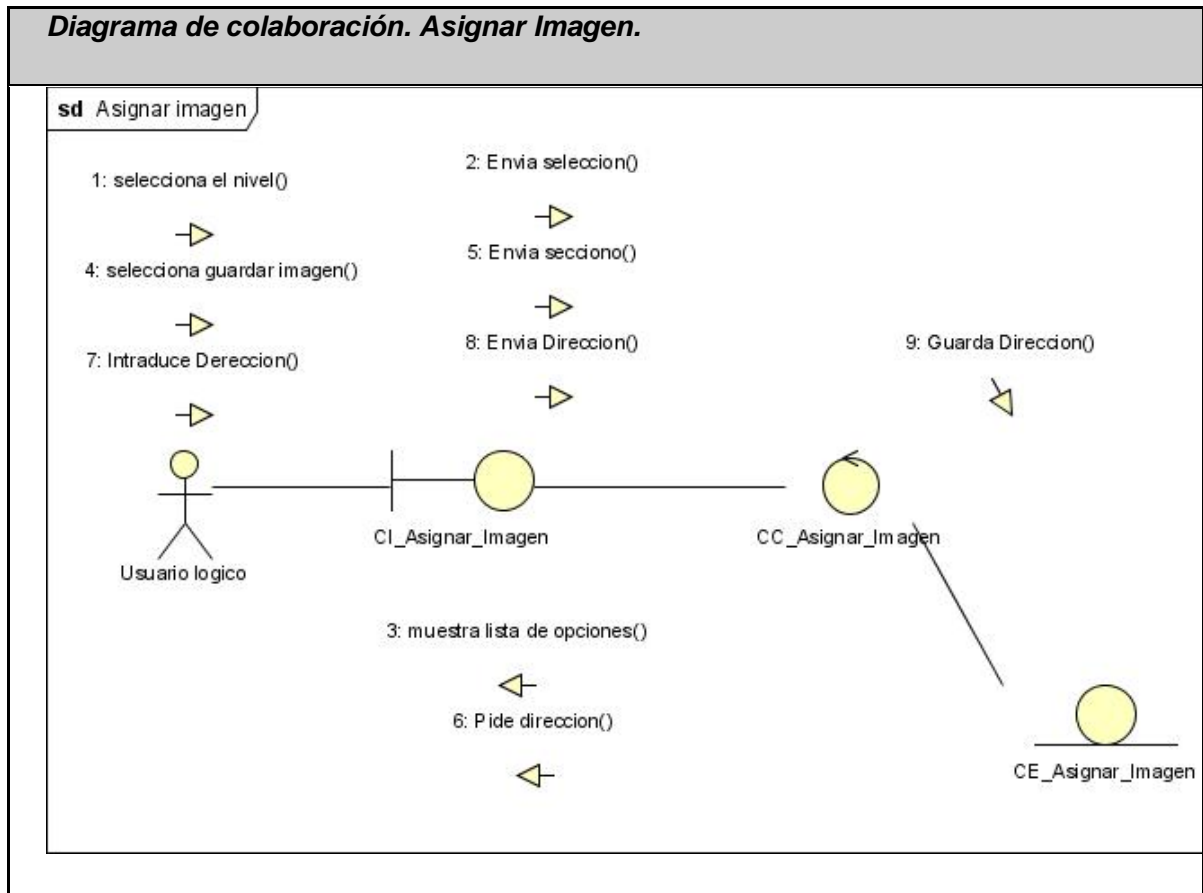


Fig. 2-27. Diagrama colaboración. Asignar imagen.

2.10.9 Diagrama de colaboración. Ubicar en Caja.

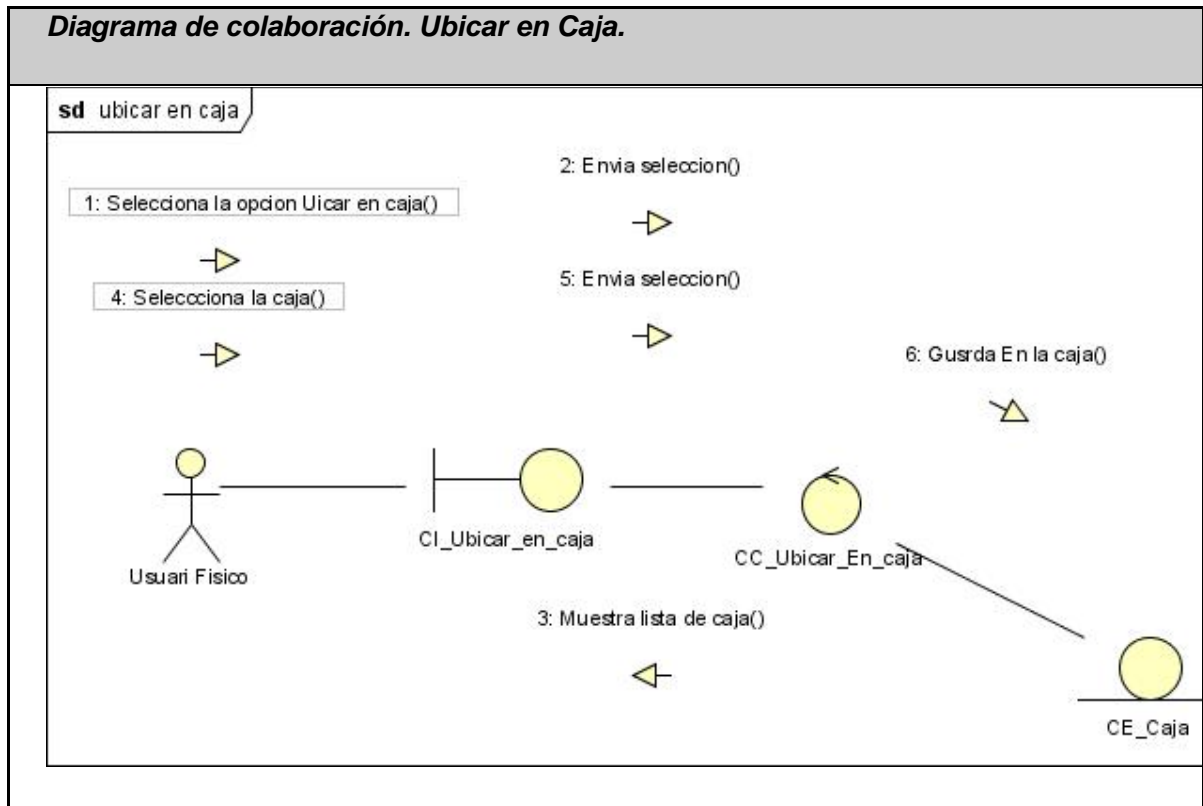


Fig. 2-28. Diagrama colaboración. Ubicar en caja.

2.10.10 Diagrama de colaboración. Adicionar Usuario.

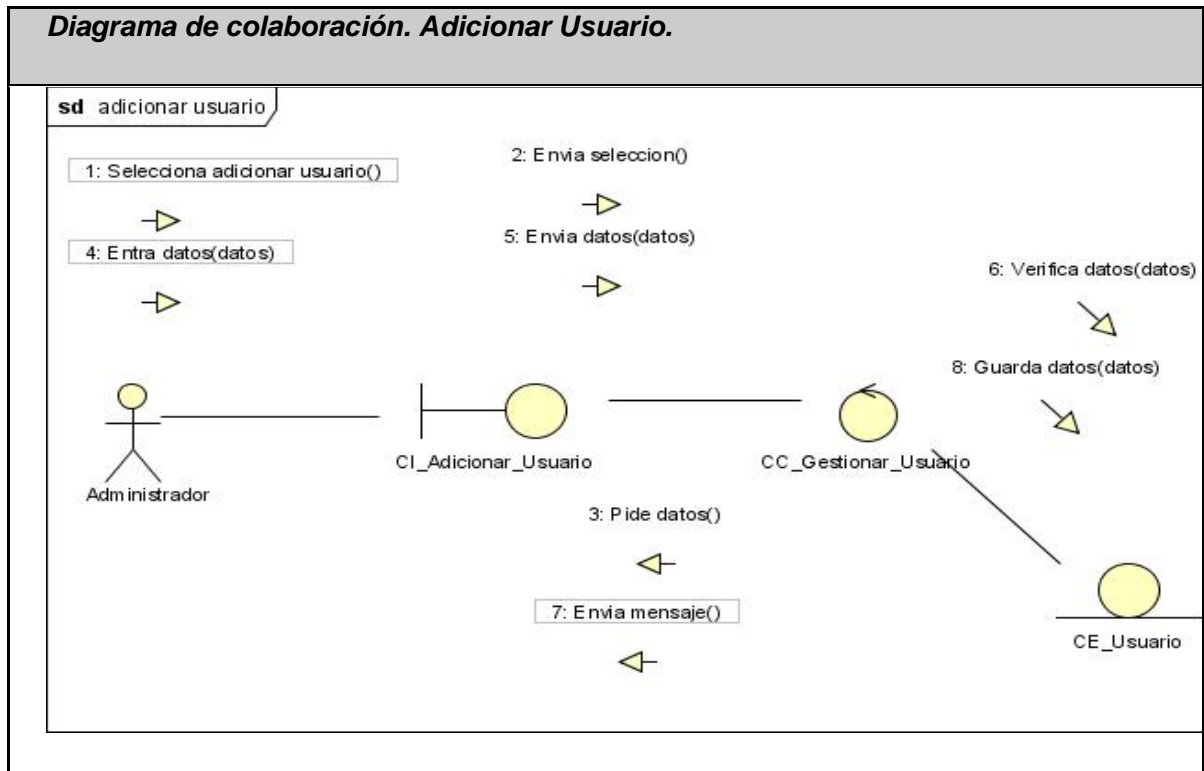


Fig. 2-29. Diagrama colaboración. Adicionar usuario.

2.10.11 Diagrama de colaboración. Autenticar Usuario.

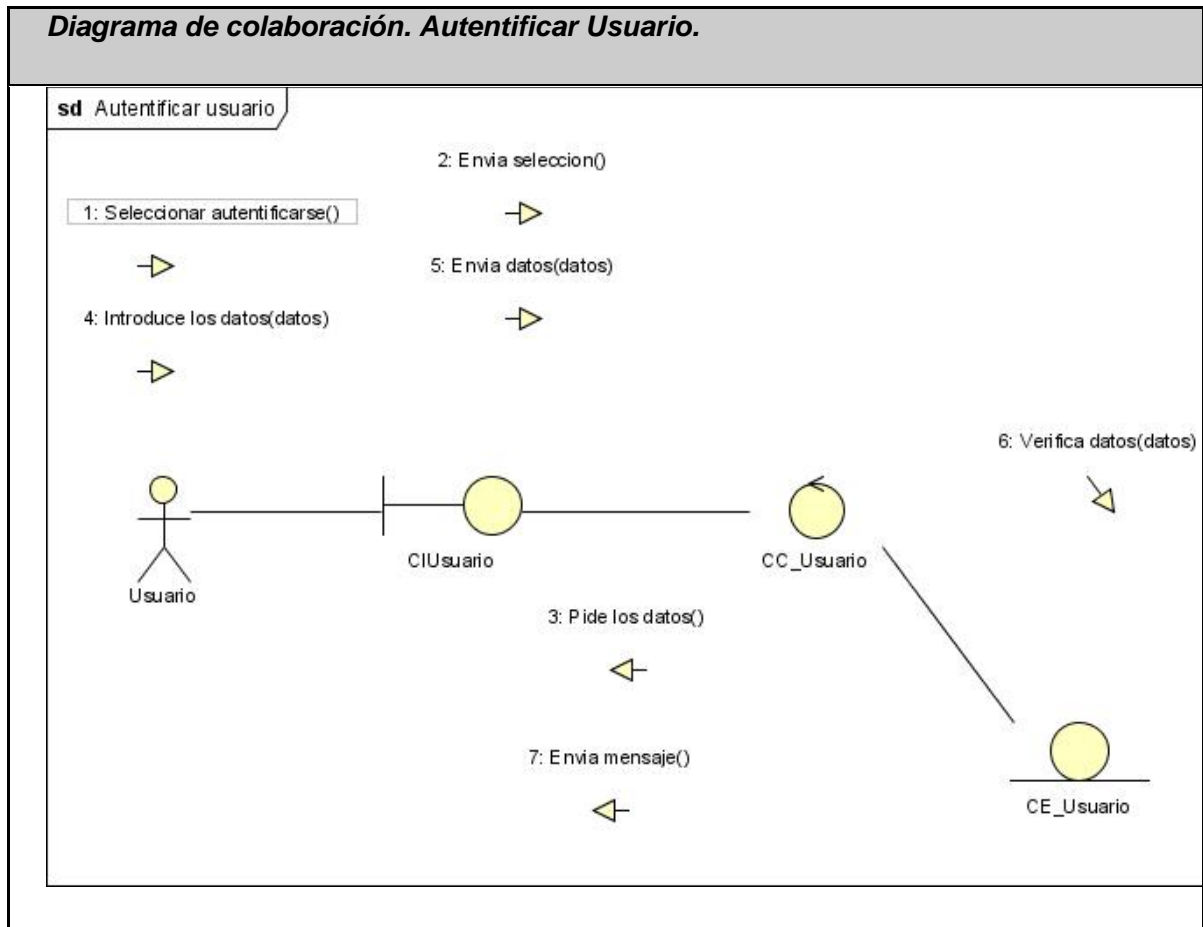


Fig. 2-30. Diagrama colaboración. Autenticar usuario.

2.10.12 Diagrama de colaboración. Modificar Usuario.

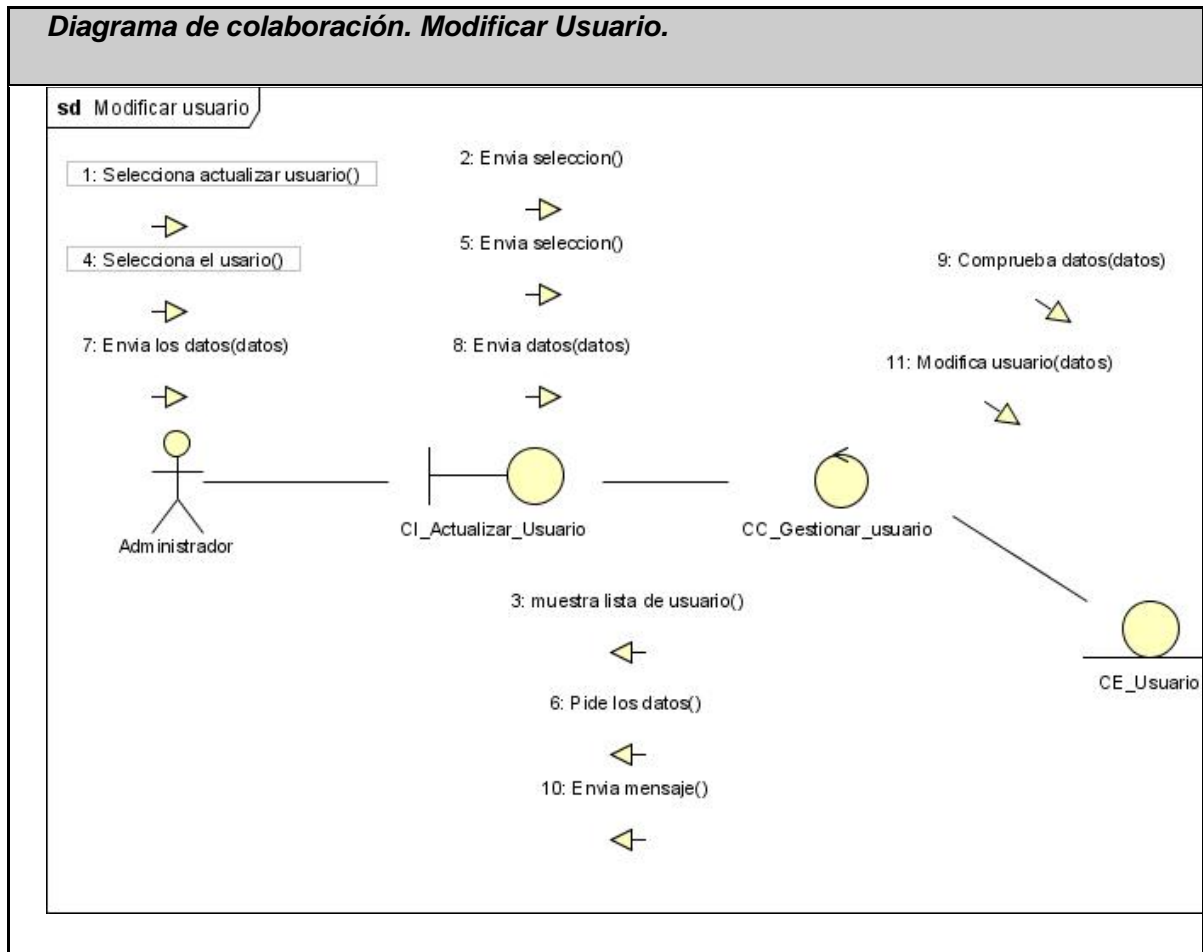


Fig. 2-31. Diagrama colaboración. Modificar usuario.

2.10.13 Diagrama de colaboración. Eliminar Usuario.

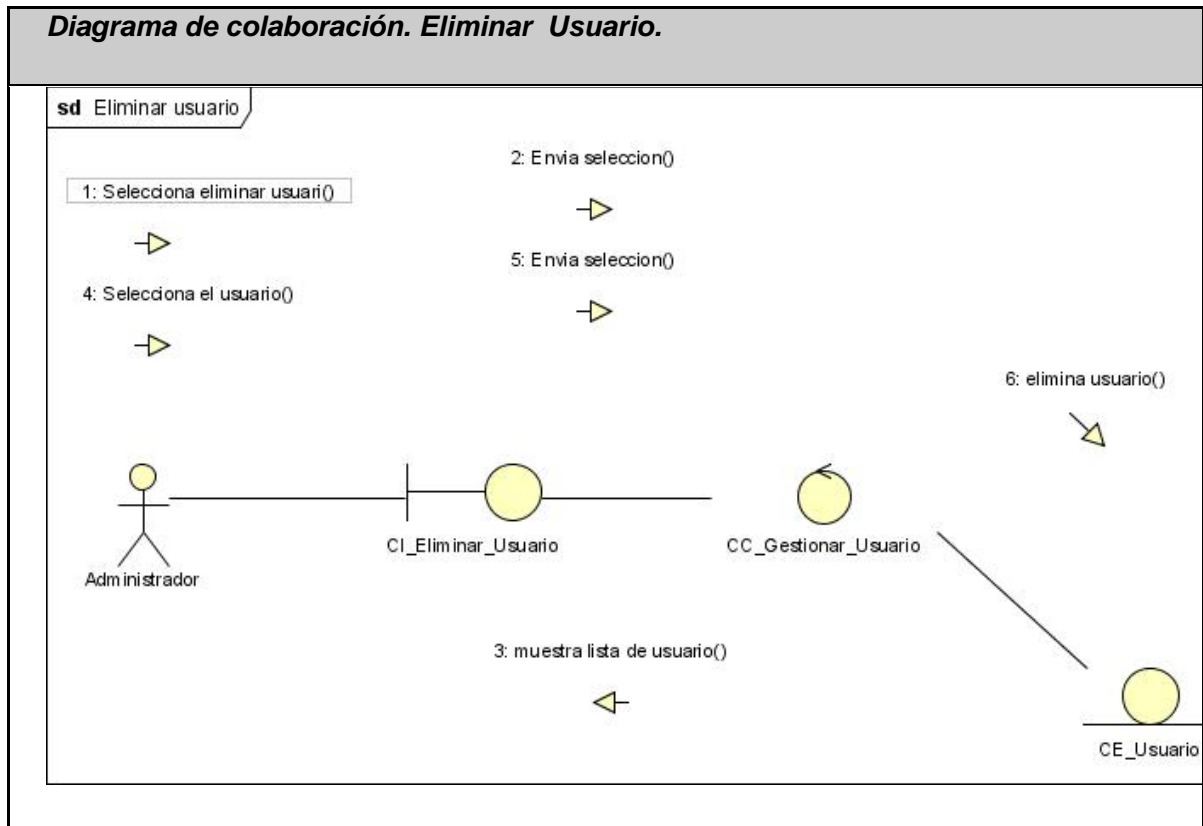


Fig. 2-32. Diagrama colaboración. Eliminar usuario.

2.10.14 Diagrama de colaboración. Registrar Usuario.

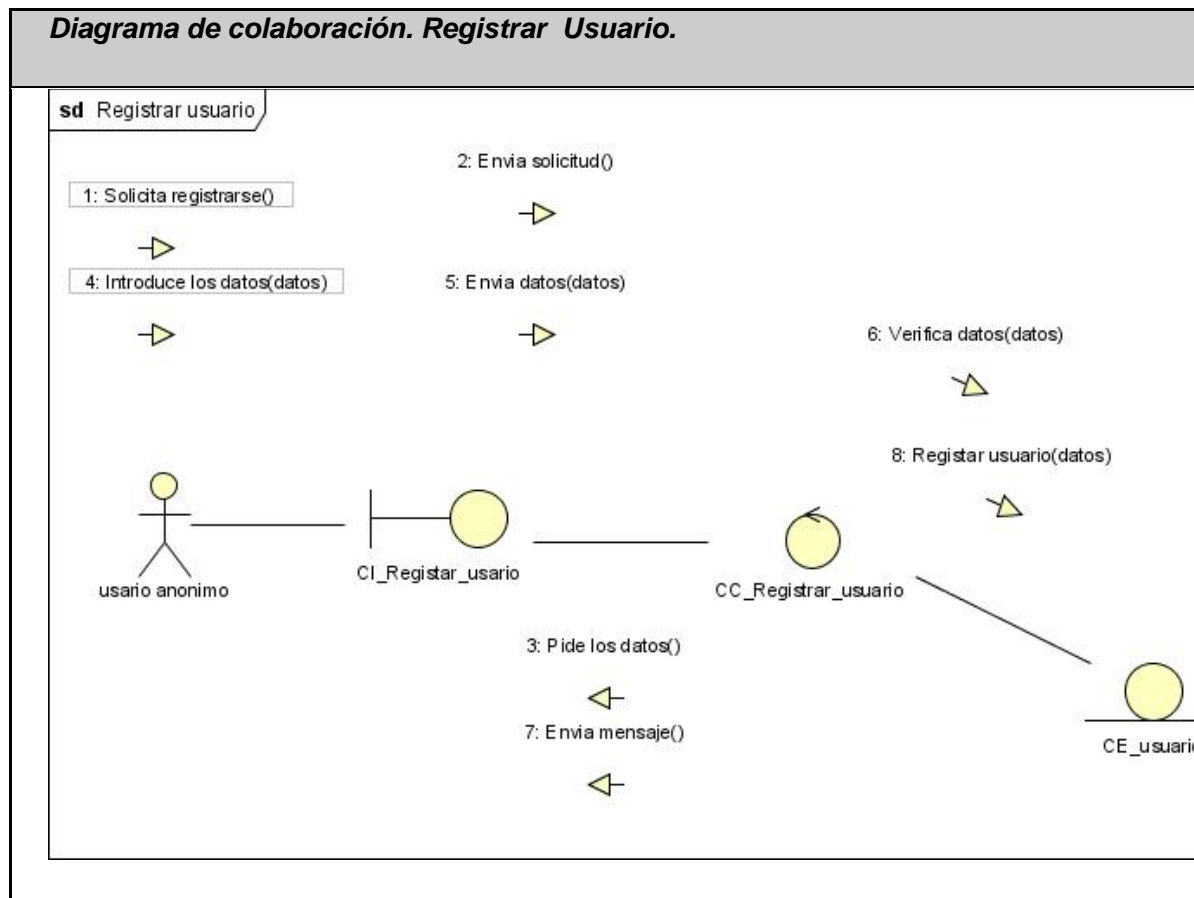


Fig. 2-33. Diagrama colaboración. Registrar usuario.

2.10.15 Diagrama de colaboración. Buscar Usuario.

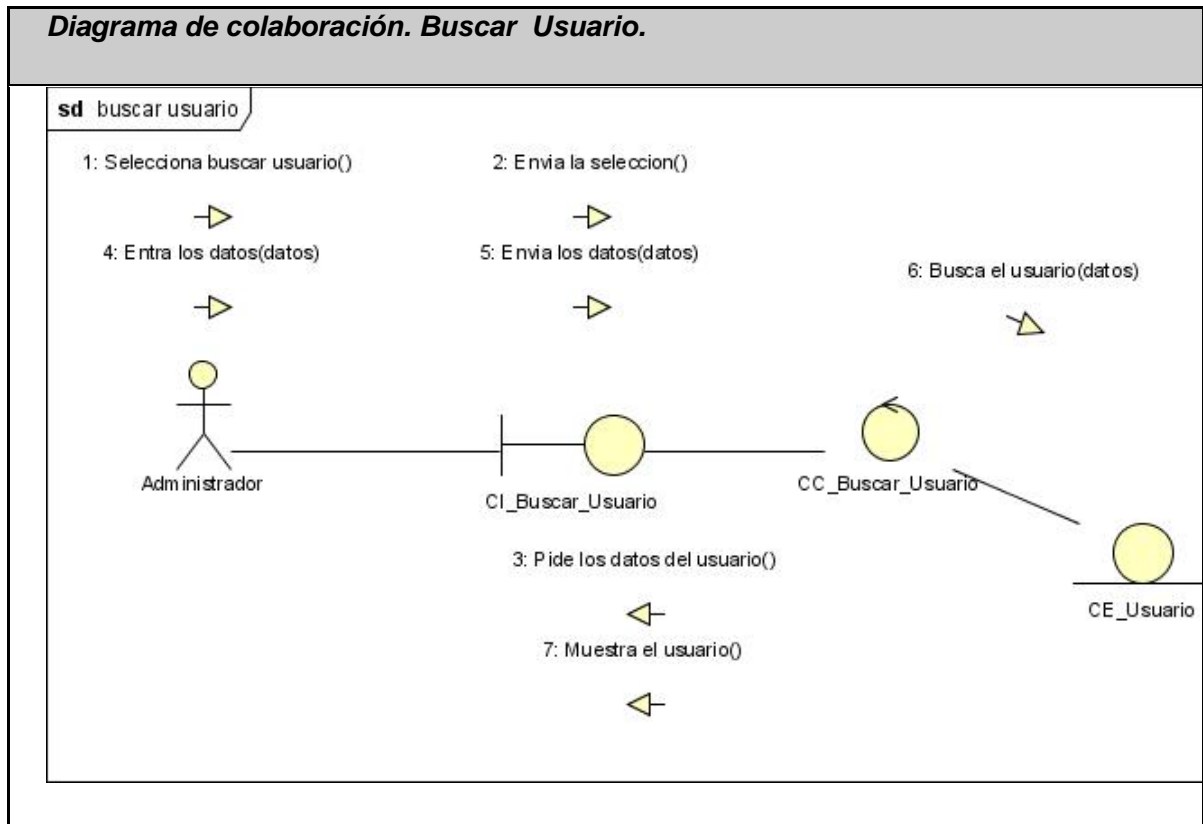


Fig. 2-34. Diagrama colaboración. Buscar usuario.

2.10.16 Diagrama de colaboración. Asignar Documento.

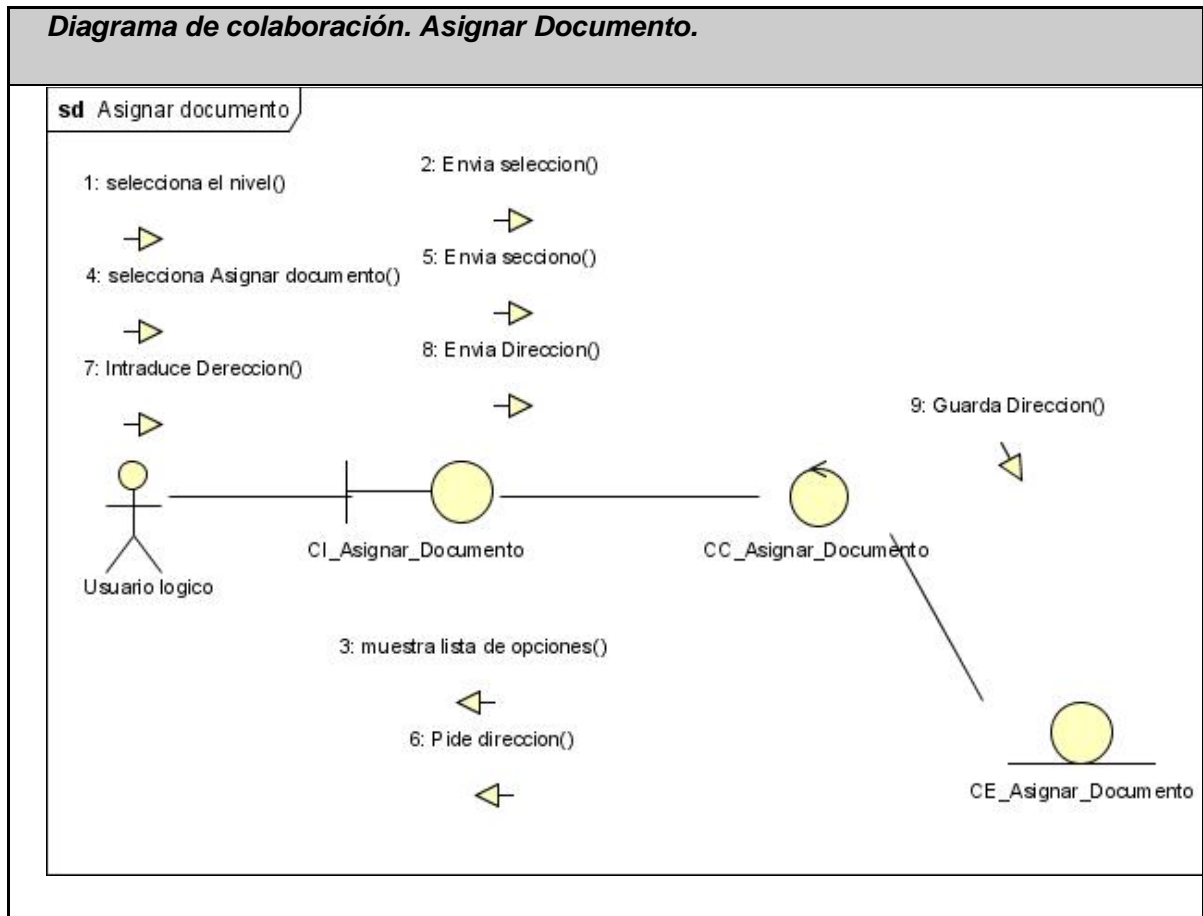


Fig. 2-35. Diagrama colaboración. Asignar documento.

2.10.17 Diagrama de colaboración. Adicionar Caja.

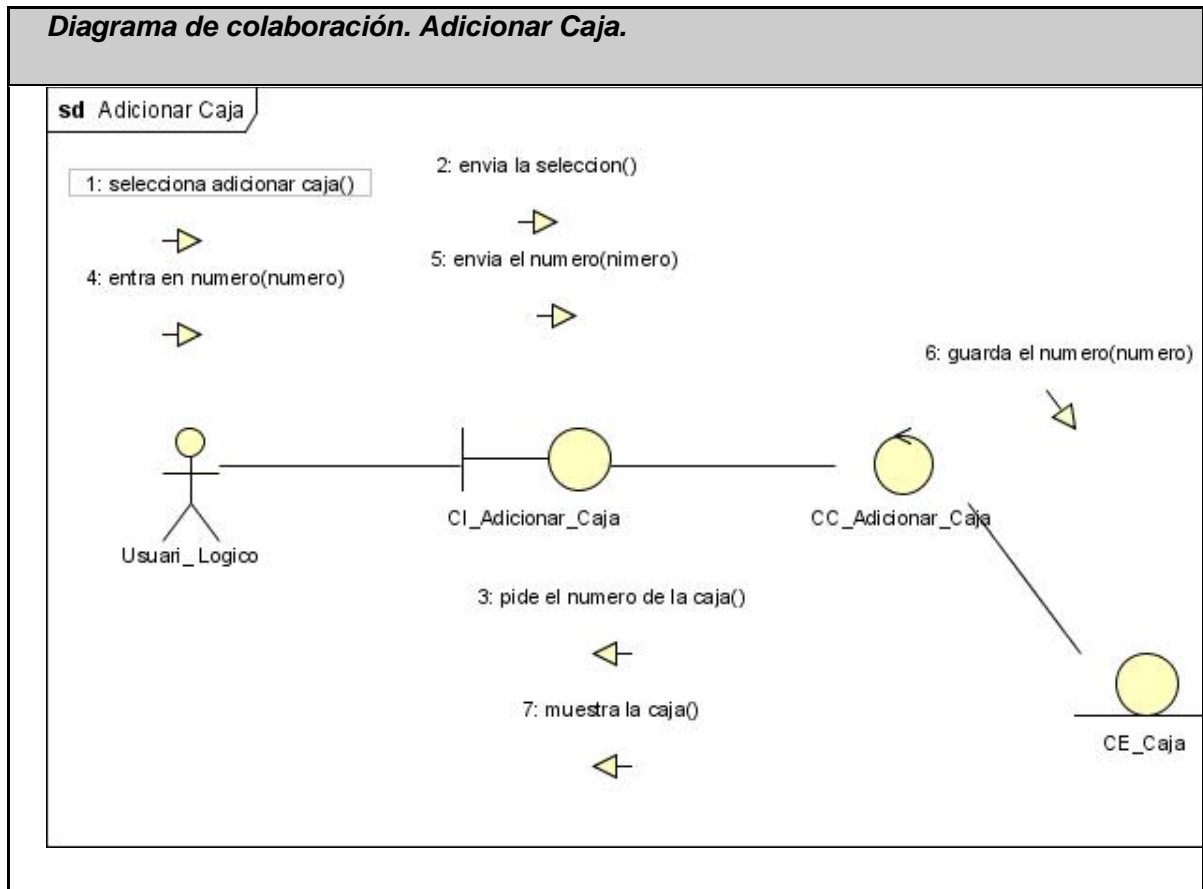


Fig. 2-36. Diagrama colaboración. Adicionar caja.

2.10.18 Diagrama de colaboración. Abrir Caja.

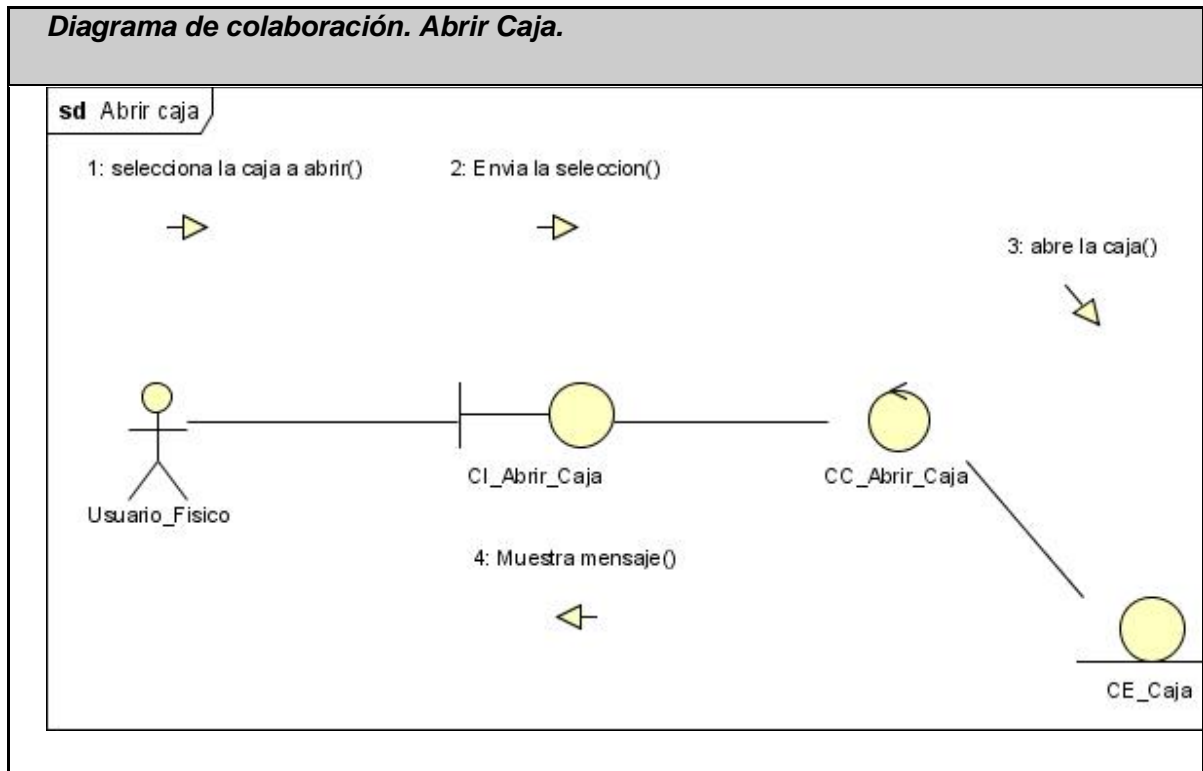


Fig. 2-37. Diagrama colaboración. Abrir caja.

2.10.19 Diagrama de colaboración. Cerrar Caja.

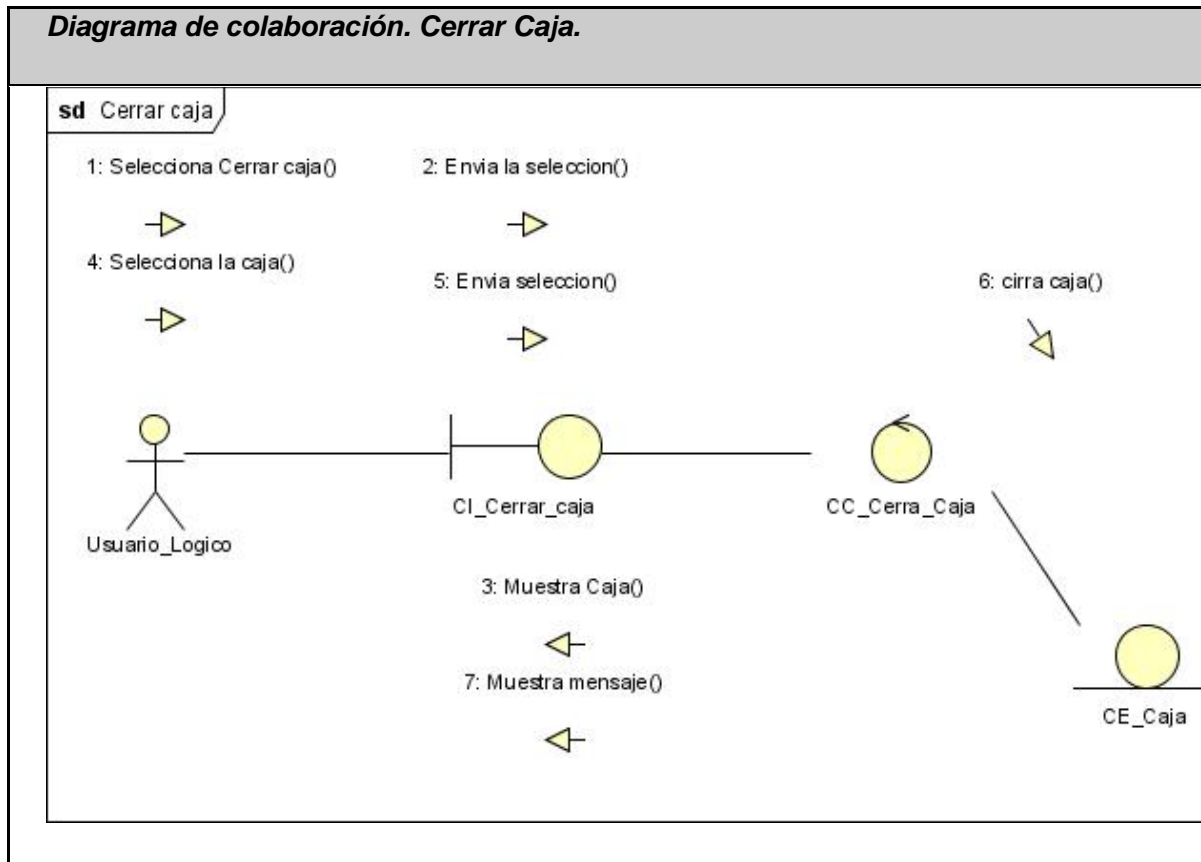
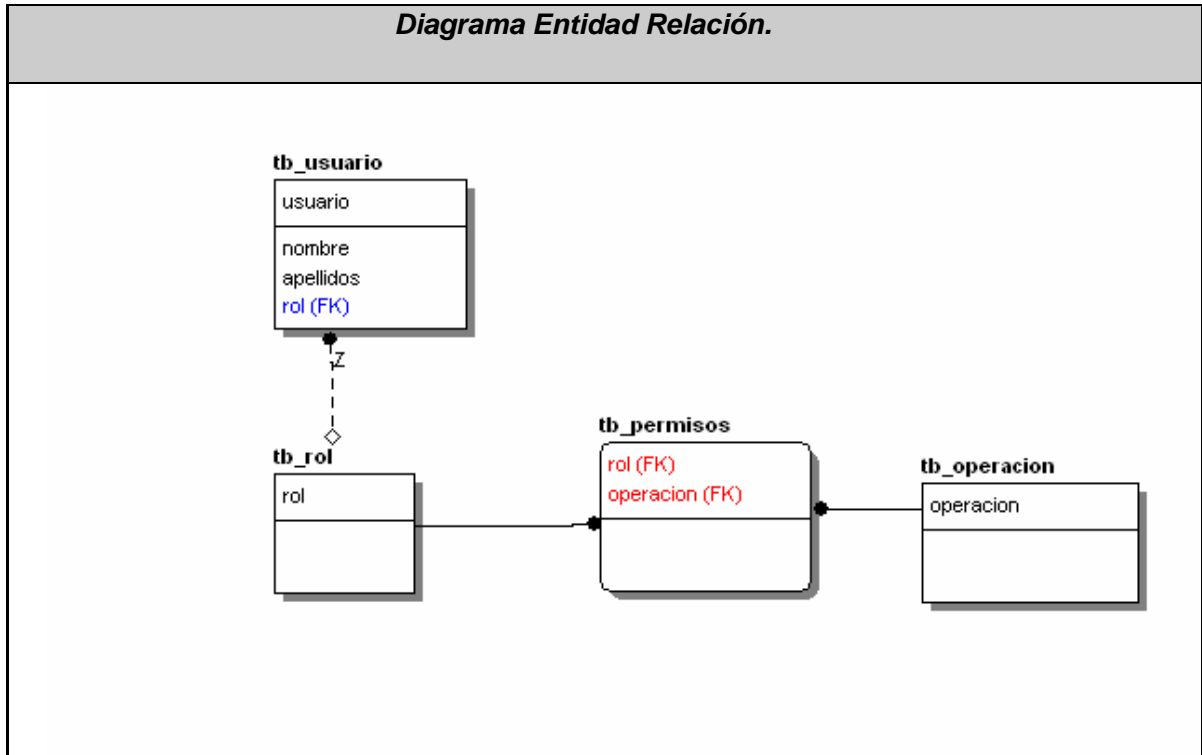


Fig. 2-38. Diagrama colaboración. Cerrar caja.

2.11 Diseño de la Base de Datos.

2.11.1 Diagrama Entidad Relación de la Base de Datos.



Capítulo 2: Descripción de la solución propuesta.

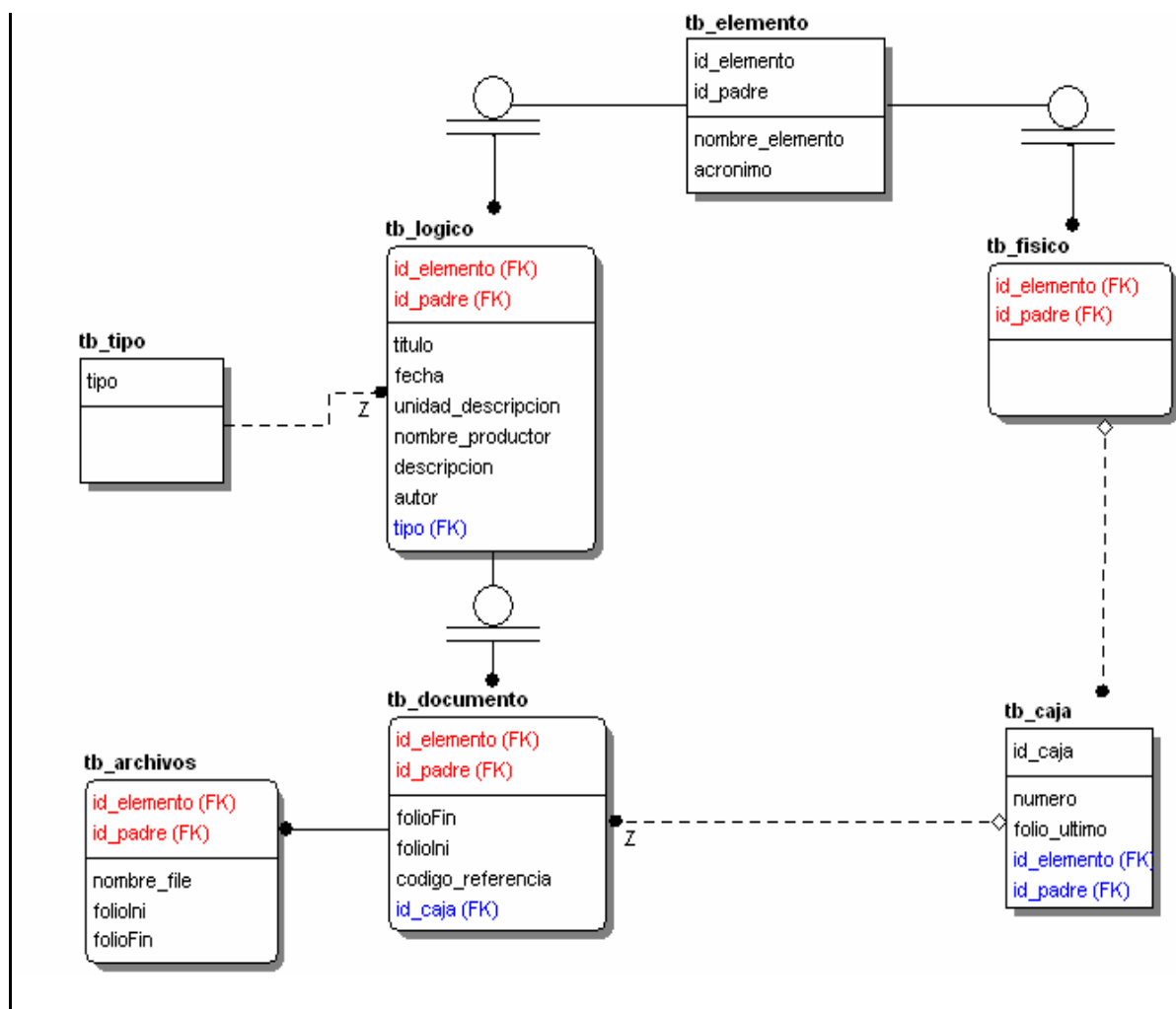


Fig. 2-39. Diagrama Entidad Relación.

2.12 Descripción de las tablas de la Base de Datos.

Tabla 2.10

Nombre: tb_logico.		
Descripción: para guardar lo datos de los documentos.		
Atributo	Tipo	Descripción
Id_elemento	int	Identificador de Elemento.
Id_padre	int	Identificador del padre.

Capítulo 2: Descripción de la solución propuesta.

titulo	varchar	Denominar la unidad de descripción.
descripción	varchar	Identificar el nivel de organización de la unidad de descripción.,
fecha	date	Fecha de la unidad de descripción.
unidad de descripción	varchar	Identificar y describir el soporte de la unidad de descripción.
Nombre_ productor	varchar	Identificar el productor.
Id_tipo	int	Identificador de la Tipo.

Tabla 2.11

Nombre:tb_elemento.		
Descripción: Ubicación física.		
Atributo	Tipo	Descripción
id_elemento	int	Identificador de Elemento.
id_padre	int	Identificador del padre.
nombre_elemento	varchar	Nombre del elemento.
acrónimo	varchar	Breve descripción de la ubicación.

Tabla 2.12

Nombre: tb_caja		
Descripción: para guardar la ubicación física.		

Capítulo 2: Descripción de la solución propuesta.

Atributo	Tipo	Descripción
Id_caja	int	Identificador de la caja.
Numero	int	Numero de la caja.
Id_elemento	int	Identificador de la ubicación física.
Id_padre	Int	Identificador del padre.

Tabla 2.13

Nombre: tb_documento.		
Descripción: guardar la ubicación lógica.		
Atributo	Tipo	Descripción
id_elemento	int	Identificador del elemento.
Id_padre	Int	Identificador del padre.
Codigo_referencia	int	Identificar de modo único la unidad de descripción.
Imagen	varchar	Imagen del documento que se va a guardar.
Id_caja	int	Identificador de la caja.

Tabla 2.14

Nombre: tb_tipo		
Descripción: Tipo de documento.		
Atributo	Tipo	Descripción
tipo	int	Identificador del tipo.

Capítulo 2: Descripción de la solución propuesta.

Tabla 2.15

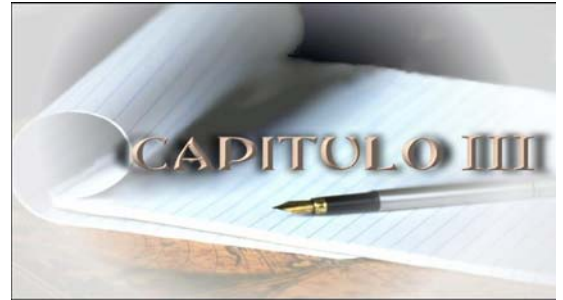
Nombre: tb_físico.		
Descripción: Guardar la Ubicación física del documento.		
Atributo	Tipo	Descripción
Id_elemento	int	Identificador de elemento.
Id_padre	Int	Identificador del padre.

Tabla 2.16

Nombre: tb_archivos.		
Descripción: para guardar los archivos.		
Atributo	Tipo	Descripción
Id _ elemento	int	Identificador de Elemento.
Id _ padre	int	Identificador del padre.
Nombre _ fichero	varchar	Para el nombre del fichero.
foliini	int	Para el folio de inicio.
Folifin	int	Para el folio final.

2.13 Conclusiones del Capítulo.

En este capítulo se mostraron las principales clases del dominio del problema para un mayor entendimiento del negocio, se describieron propuestas de solución para el desarrollo del software a través de la muestra de varias vistas para llevar a cabo el proceso de análisis, diseño así como el diagrama de entidad relación de la base de datos y la descripción de sus tablas.



3. Implementación

3.1 Introducción

El objetivo fundamental de este capítulo es definir cómo desarrollar la arquitectura prevista durante el diseño, representando el sistema como un todo. Se implementan las clases y subsistemas definidos en el capítulo anterior en términos de componentes. Se modelan los Diagramas de Componentes y de Despliegue, quedando conformado el Modelo de Implementación del sistema, dando una visión de cómo quedará construida y distribuida la aplicación.

3.2 Modelo de Implementación

El Modelo de Implementación es comprendido por un conjunto de componentes y subsistemas que constituyen a la composición física de la implementación del sistema. Este modelo describe la relación que existe desde los paquetes y clases del Modelo de Diseño a subsistemas y componentes físicos, cómo se organizan estos últimos de acuerdo con los mecanismos de estructuración, lenguajes utilizados y cómo dependen unos de otros.

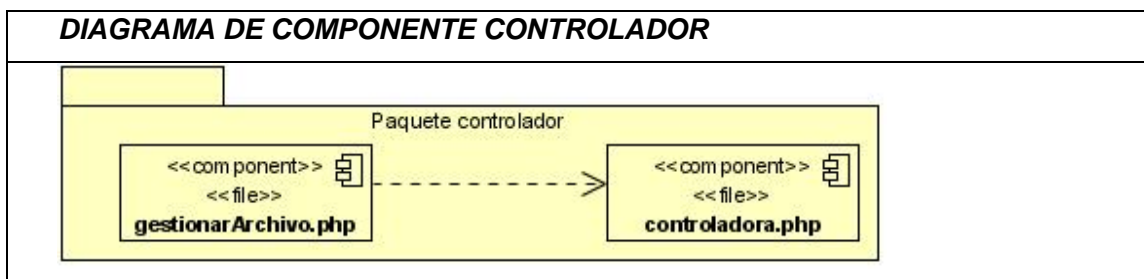
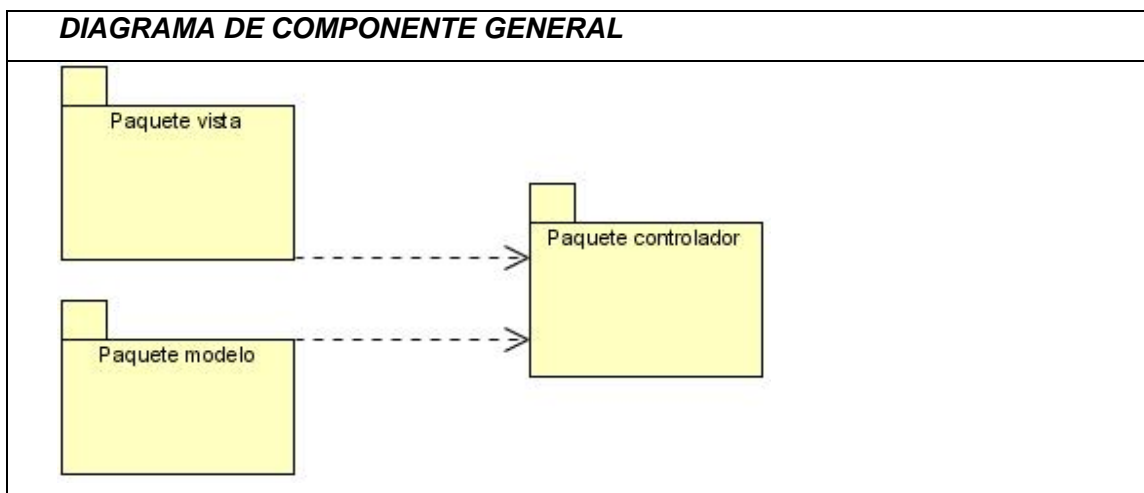
El Modelo de Implementación del Sistema de Archivo Histórico, expone una organización en capas, jerarquías de paquetes y subsistemas de implementación. Los cuales contienen componentes y sus relaciones, dividiendo al sistema en partes más manejables y en partes más pequeñas. Posibilitando la reutilización, la implementación de componentes por separado y disminución del impacto que pueda traer consigo un cambio.

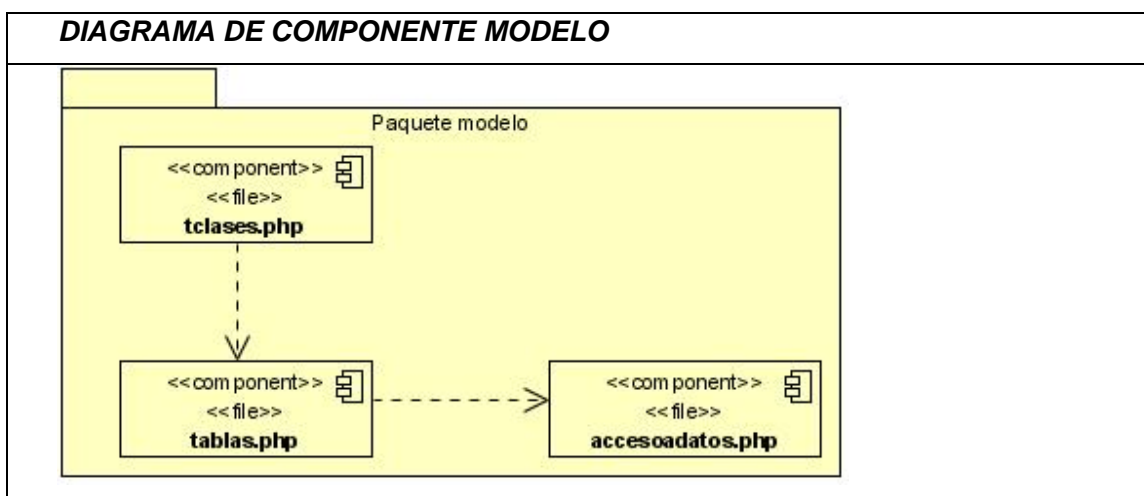
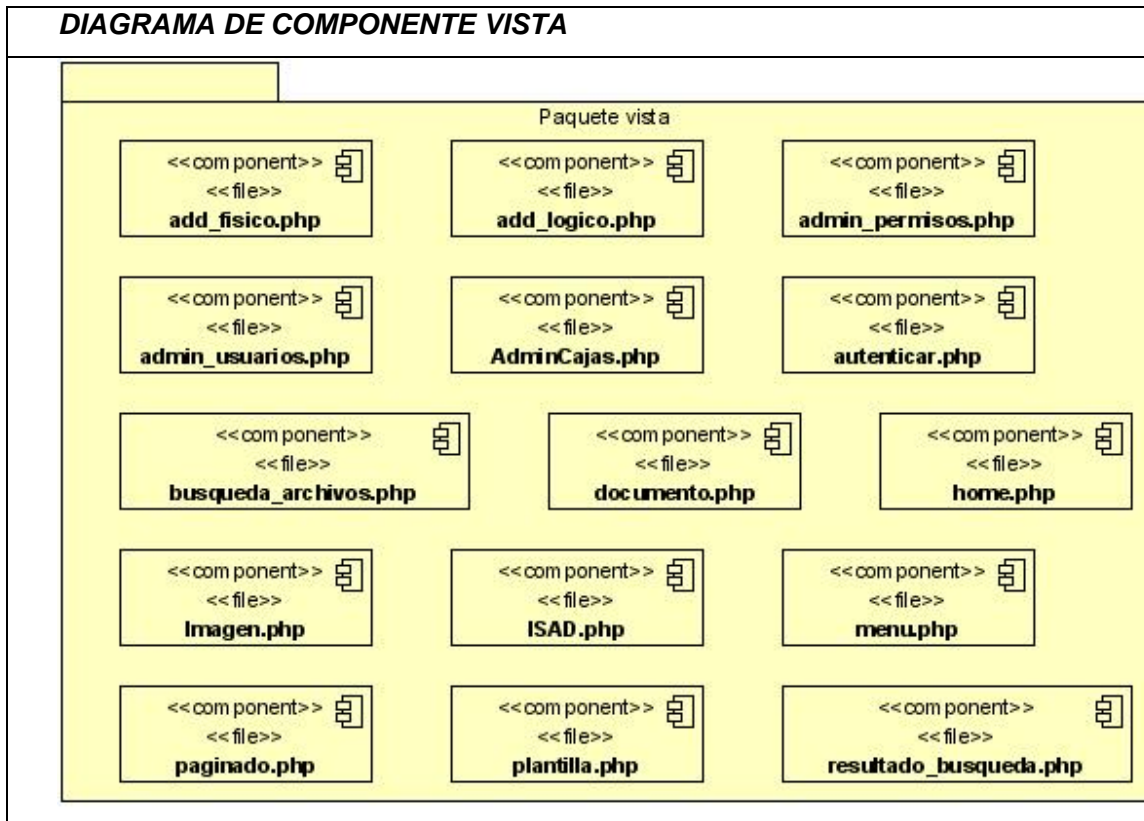
3.2.1 Diagrama de Componentes

Con el objetivo de lograr una mejor comprensión de los componentes que forman el sistema, se presentan los Diagramas de Componentes que describen los elementos del sistema y sus relaciones.

Componente es: el empaquetamiento físico de los elementos de un modelo, como son las clases en el Modelo del Diseño. Los componentes pueden ser simples archivos, paquetes o bibliotecas cargadas dinámicamente. Las relaciones de dependencia se utilizan, en los Diagramas de Componentes, para indicar que un componente utiliza los servicios ofrecidos por otro componente, las mismas pueden ser de *Import* entre subsistemas y de *Compilation* entre paquetes de componentes o componentes.

El diagrama que se presenta a continuación, tiene como objetivo, figurar la estructura y organización general del Sistema de Planificación, en términos de componentes:





Capítulo 3: Implementación.

El sistema se encuentra estructurado en paquetes, los cuales contienen componentes de implementación.

Dentro del paquete general, SArchivos se encuentran tres subpaquetes fundamentales, modelo, vista, controlador. Dentro del paquete vista se encuentran todas las páginas que mostraran las interfaces del usuario. Dentro del paquete modelo se encontrar aquellas clases se encargaran de realizar las consultas y manipulación directamente con la base de datos, el componente Acceso a Datos, se encargara de forma abstracta del acceso al servidor mientras que el componente Tabla se encargara de forma mas objetiva de funcionalidades comunes hacia las tablas, como insertar, modificar, buscar, eliminar, pudiendo ser estos especializados mediante herencia. En el paquete controlador, se encontraran todas aquellas clases que se encargaran de controlar todo el flujo de información entre la vista y el controlador, pudiendo acceder a las funciones de la modelo y construyendo las vistas en correspondencia con las necesidades de cada funcionalidad. Aquí dentro se puede mencionar el componente controlador, el brindara un conjunto de funcionalidades comunes para el control de los procesos y eventos, pudiendo ser estos igualmente especializados por herencia.

3.3 Diagrama de Despliegue

El diagrama de despliegue describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos. Los nodos representan recursos de cómputos: procesadores o dispositivos de hardware. La distribución de la aplicación quedará de la siguiente manera:

DIAGRAMA DE DESPLIEGUE

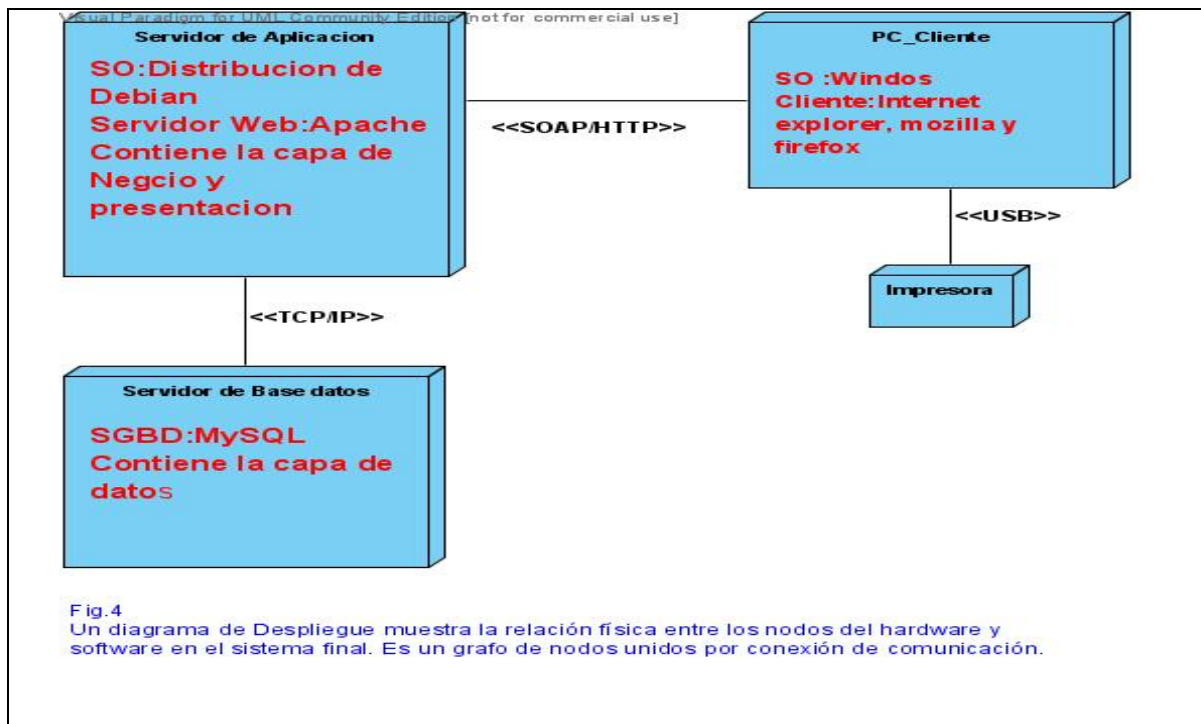


Figura 4.2. Diagrama de Despliegue

En el Servidor de Aplicación se encuentra montado el Sistema de Archivo Histórico específicamente la Capa de Presentación y la Capa de Negocio, junto con el servidor web Apache que posibilita su publicación. La capa de Datos esta localizada en el Servidor de Base de Datos de la aplicación, en ella se guarda la información que se gestiona a través del sistema.

En el nodo que representa la PC_Cliente, se encuentra el navegador web, a través del cual se puede ejecutar la aplicación. Para realizar la impresión de las fotos de los documentos, se utiliza una impresora, representada con el dispositivo llamado Impresora.

3.4 Conclusiones

En este capítulo se mostraron los resultados obtenidos durante la etapa de implementación; quedando conformados los Diagramas de Componentes y Despliegue, que representan cómo construir y distribuir el sistema, permitiendo la obtención de un producto que cumpla con las funcionalidades propuestas.

Conclusiones Generales.

En este trabajo se analizaron aspectos relacionados con la construcción, la necesidad de crear y gestionar un sistema de archivo histórico, basados en las normas ISAD (G), Con la realización de esta investigación se arriba a las siguientes conclusiones:

- La necesidad de desarrollar e implementar un Sistema Básico de Archivos Históricos que fuera libre, factible y rápido.
- Los sistemas de archivos históricos funcionales en nuestro país tienen problemas de interconexión y no son multiplataforma.
- Como resultado del presente trabajo se logró desarrollar una aplicación Web que facilita la difusión y preservación de documento históricos en una institución. Se considera que han sido cumplidos los objetivos planteados.

Recomendaciones

Sobre la presente investigación recomendamos:

- Someter a una prueba de calidad rigurosa al software para su posterior uso.
- Ampliar las funcionalidades del sistema desarrollado.
- Hacer un estudio de la implementación de un Sistema de Archivo.
- Realizar la autenticación al sistema mediante el dominio de la institución.
- Implementar un servicio Web que brinde búsquedas de documentos para los clientes interesados.
- Agregar funcionalidades para gestionar las trazas de los usuarios en el sistema.

Referencias bibliográficas.

[1]. Sitio Web:

<http://cmisapp.ayto-zaragoza.es/hemeroteca/visualizar.jsp?modulo=archivo&pagina=quees>

[Citado el 20 enero del 2008].

[2]. NUÑEZ CONTRERAS, Luis: "Concepto de documento", en Archivística. Estudios básicos.

Sevilla: Diputación [Citado el 20 enero del 2008].

Provincial, 1983, 31.

[Citado el 20 enero del 2008].

[3]. Sitio Web: <http://archivo3000.com/general/archivo3000.htm#pestanas> [Citado el 20 enero del 2008].

[4]. Sitio Web: <http://www.arnac.cu/index.htm> [Citado el 20 enero del 2008].

[5]. Sitio Web: <http://pares.mcu.es/> [Citado el 22 enero del 2008].

[6]. Sitio Web: http://www.ahm.granma.inf.cu/programas/software_papiro/soft_papiro.php

[Citado el 22 enero del 2008].

[7]. Sitio Web: [Citado el 10 febrero del 2008].

http://www.informaticahabana.com/evento_virtual/?q=node/233&ev=III%20Simposio%20Internacional%20de%20Gobierno%20en%20L%C3%ADnea [Citado el 10 febrero del 2008].

[8]. Sitio Web: <http://www.msinfo.info/propuestas/documentos/isad-g.pdf> [Citado el 10 febrero del 2008].

[9]. Sitio Web: <http://www.lenguajes-de-programaion.com/programacion-web.shtml>

[Citado el 10 febrero del 2008].

[10]. Sitio Web: <http://gtk.php.net/manual1/es/html/intro.whatishistory.html>

[Citado el 10 febrero del 2008].

[11]. Sitio Web: <http://gtk.php.net/manual1/es/html/intro.whatishistory.html>

[Citado el 10 febrero del 2008].

[12]. Sitio Web: http://es.wikipedia.org/wiki/.php#PHP_5 [Citado el 17 febrero del 2008].

[13]. Sitio Web: <http://es.wikipedia.org/wiki/.php#Ventajas> [Citado el 17 febrero del 2008].

[14]. Sitio Web: <http://es.wikipedia.org/wiki/Perl> [Citado el 17 febrero del 2008].

[15]. Sitio Web: <http://www.masadelante.com/faq-servidor.htm> [Citado el 7 marzo del 2008].

[16]. Sitio Web: http://es.wikipedia.org/wiki/Internet_Information_Services

[Citado el 7 marzo del 2008].

- [17]. Sitio Web: <http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml> [Citado el 7 marzo del 2008].
- [18]. Sitio Web: <http://es.wikipedia.org/wiki/PostgreSQL> [Citado el 28 marzo del 2008].
- [19]. Sitio Web: <http://es.wikipedia.org/wiki/MySQL> [Citado el 18 abril del 2008].
- [20]. Jacobson, I.; Booch, G.; Rumbaugh J. El proceso unificado de desarrollo de software. Félix Varela, La Habana 2004 [Citado el 28 abril del 2008].

Bibliografía.

Sitio Web: <http://cmisapp.ayto-zaragoza.es/hemeroteca/visualizar.jsp>

NUÑEZ CONTRERAS, Luis: "Concepto de documento", en Archivística. Estudios básicos. Sevilla: Diputación

Provincial, 1983, 31.

Sitio Web: <http://archivo3000.com/general/archivo3000.htm>

Sitio Web: <http://archivo3000.com/general/archivo3000.htm#pestanas>

Sitio Web: <http://archivo3000.com/general/archivo3000.htm#pestanas>

Sitio Web: <http://archivo3000.com/general/a3wp.htm>

Sitio Web: <http://archivo3000.com/general/a3wd.htm>

Sitio Web: <http://archivo3000.com/general/a3wv.htm>

Sitio Web: <http://archivo3000.com/general/a3ws.htm>

Sitio Web: <http://www.arnac.cu/index.htm>

Sitio Web: <http://www.arnac.cu/nuestrai.htm>

Sitio Web: <http://pares.mcu.es/>

Sitio Web: http://www.ahm.granma.inf.cu/programas/software_papiro/soft_papiro.php

Sitio Web:

http://www.informaticahabana.com/evento_virtual/?q=node/233&ev=III%20Simposio%20Internacional%20de%20Gobierno%20en%20L%C3%ADnea

Sitio Web: <http://www.msinfo.info/propuestas/documentos/isad-g.pdf>

Sitio Web: <http://www.mcu.es/archivos/docs/isaar.pdf>

Sitio Web: <http://www.lenguajes-de-programacion.com/programacion-web.shtml>

Sitio Web: <http://gtk.php.net/manual1/es/html/intro.whatism.php.history.html>

Sitio Web: <http://gtk.php.net/manual1/es/html/intro.whatism.php.whatdoes.html>

Sitio Web: http://es.wikipedia.org/wiki/.php#PHP_5

Sitio Web: <http://es.wikipedia.org/wiki/.php#Ventajas>

Sitio Web: <http://es.wikipedia.org/wiki/Perl>

Sitio Web: <http://www.masadelante.com/faq-servidor.htm>

Sitio Web: http://es.wikipedia.org/wiki/Internet_Information_Services

Sitio Web:

<http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml>

Sitio Web: <http://es.wikipedia.org/wiki/PostgreSQL>

Sitio Web: <http://es.wikipedia.org/wiki/MySQL>

Jacobson, I.; Booch, G.; Rumbaugh J. El proceso unificado de desarrollo de software. Félix Varela, La Habana 2004.

Norma ISAD-G.

NUÑEZ CONTRERAS, Luís: "Concepto de documento", en *Archivística. Estudios básicos*. Sevilla: Diputación Provincial, 1983, 31.

Libro: Descripción archivística codificada, directrices de aplicación *Elaboradas por el Encoded Archival Description Working Group de la Society of American Archivist*.

<http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>

<http://www.aspfacil.com/articulos/278001.asp>

<http://www.w3c.es>

<http://www.w3schools.com>

Glosario de términos.

Término	Definición
ANSI	(American National Standards Institute - Instituto Nacional Americano de Estándares)
API	API (del inglés Application Programming Interface-Interfaz de Programación de Aplicaciones) es el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta librería para ser utilizado por otro software como una capa de abstracción.
ASP	Servidor de Página Activa (ASP) es una tecnología del lado servidor de Microsoft para páginas web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Services
AWK	Es un lenguaje de programación diseñado para procesar datos basados en texto, ya sean ficheros o flujos de datos.
CASE	Computer Aided Software Engineering (Ingeniería de Software Asistida por Ordenador), aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.
CUS	Caso de uso del sistema.
C/C++	Lenguaje de programación compilado.

DOM	Modelo de objetos en documentos.
FTP	Protocolo de Transferencia de Archivos. Es el protocolo de comunicaciones que permite la interconexión entre ordenadores y la transferencia de ficheros.
Hash	Se refiere a una función o método para generar claves o llaves que representen de manera casi unívoca a un documento.
Herramientas de Autor	Son aplicaciones que permiten un trabajo multimedia y constructivista para generar un entorno de aprendizaje dinámico, dentro de las funcionalidades que este tipo de herramientas presentan se puede destacar la posibilidad de crear actividades o pequeñas aplicaciones desde la misma herramienta.
HTML	Lenguaje de Marcación de Hipertexto. Es un lenguaje de marcas diseñado para estructurar textos y representarlos en forma de hipertexto, que es el formato estándar de las páginas web.
HRM	Gestión de Recursos Humanos.
HTTP	Protocolo de Transferencia de Hipertexto. Protocolo para transferir archivos o documentos hipertexto a través de la red.
HTTPS	Protocolo Seguro de Transferencia de Hipertexto. Garantiza la seguridad de las comunicaciones entre el usuario y el servidor web al que este se conecta.
Hosting	Servicio de alojamiento de las páginas web de otras empresas.
IIS	Servidor Web para la plataforma de Windows.

ISO	Organización Internacional para la Estandarización.
Interoperabilidad	La condición mediante la cual sistemas heterogéneos pueden intercambiar procesos o datos.
JavaScript	Es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web.
Lisp	Es el segundo lenguaje de programación.
Macros	Es una serie de instrucciones que se almacenan para que se puedan ejecutar de forma secuencial mediante una sola llamada.
Mainframe	Es una computadora grande, generalmente el ordenador central de una empresa u organismo.
MoReq	Modelo de Requisitos para la Gestión Electrónica de Documentos de Archivo
MySQL	Servidor de bases de datos recomendable para desarrollos que necesiten manejar numerosos registros y sesiones simultáneas.
MVCC	Control de las Multi Ocurrencia de Versiones.
Netcraft	Netcraft es una compañía de servicios de Internet con sede en Bath, Inglaterra.
NNTP	Protocolo empleado en Internet para transferir artículos en los grupos de noticias
NuSOAP	Es un grupo de las clases escritas en PHP que permiten a los desarrolladores crear y consumir servicios web a través del protocolo

	SOAP.
OCLC	Librería central de Computadoras en línea.
ODBC	Interfaz normalizada, o intermedia, para acceder a una base de datos desde un programa.
Open-Source	Código abierto.
Option Pack	Es un tipo de multimedia.
Parser	Es un módulo, biblioteca o programa que se ocupa de transformar un archivo de texto en una representación interna.
PHP	Lenguaje de programación interpretado.
Personal Home Pages	Lenguaje para crear páginas dinámicas para la Web.
PostgreSQL Global Development Group	Grupo global de desarrollo de base de datos.
GTK	Es una biblioteca, la cual contiene los objetos y funciones para crear la interfaz gráfica de usuario.
GIS	Sistema de Información Geográfica.
GiST	Es un biblioteca de gráficos científicos desarrollada en lenguaje de programación C.

GNU GPL	La Licencia Pública General.
REST	Transferencia de Estado Representacional. Estilo arquitectónico para sistemas distribuidos basado en el protocolo HTTP.
RUP	El Proceso Unificado de Racional es un proceso de desarrollo de software
RPC	Llamada a Procedimiento Remoto. Permite que un programa ejecute una subrutina o procedimiento en un espacio de direcciones diferente (comúnmente otro ordenador) sin que el programador tenga que codificar explícitamente los detalles de la interacción remota.
SMTP	Simple Mail Transfer Protocol (SMTP), o protocolo simple de transferencia de correo electrónico. Protocolo de red basado en texto utilizado para el intercambio de mensajes de correo electrónico entre computadoras o distintos dispositivos.
SOAP	Protocolo Simple de Acceso a Objeto. Protocolo para el intercambio de mensajes XML entre redes, comúnmente utilizado sobre el protocolo HTTP.
Shell	Intérprete de instrucciones o comandos.
SQLite	Es un sistema de gestión de bases de datos relacional compatible con ACID, y que está contenida en una relativamente pequeña.
Sun Microsystems	Es una empresa informática de Silicón Valley, fabricante de semiconductores y software.
USMARC21	Es un software para la gestión de archivos que utilizando las normas

	ISAD(G) e ISAAR(CPF).
UML	Lenguaje Unificado de Modelado, lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos.
UNIX	Es un sistema operativo portable, multitarea y multiusuario.
URL	Es el Localizador Uniforme de Recursos, o dicho mas claramente, es la dirección que localiza una información dentro de Internet.
WWW	Red mundial amplia.
WS	Servicio web. Conjunto de estándares y protocolos para el intercambio de datos entre aplicaciones.
W3C	World Wide Web Consortium. Desarrolla tecnologías inter-operables (especificaciones, guías, software y herramientas) para maximizar el potencial de la web.
XPath	Es un lenguaje que permite construir expresiones que recorren y procesan un documento XML.
XML	Extensible Markup Language (lenguaje de marcas extensible) es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C).
XML- RPC	XML-RPC es un protocolo de llamada a procedimiento remoto que usa XML para codificar las llamadas y HTTP como mecanismo de transporte.
Zend	Editor web orientado a la programación de páginas PHP.