

SISTEMA DE GESTIÓN DE JAVA ARCHIVES

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN
CIENCIAS INFORMÁTICAS**

FACULTAD 8

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

AUTOR: YUDISLANDRY AGUILA GONZALEZ

TUTOR: ING. ANGEL ALBERTO VAZQUEZ SÁNCHEZ

Co_TUTOR: ING. YENISLEIDY PILOTO LASTRA

“AÑO DEL 50 ANIVERSARIO DEL TRIUNFO DE LA REVOLUCIÓN” LA HABANA, CUBA,

JUNIO 2009

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco 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 _____.

Yudislandry Aguila Gonzalez

Ing. Angel Alberto Vazquez Sánchez

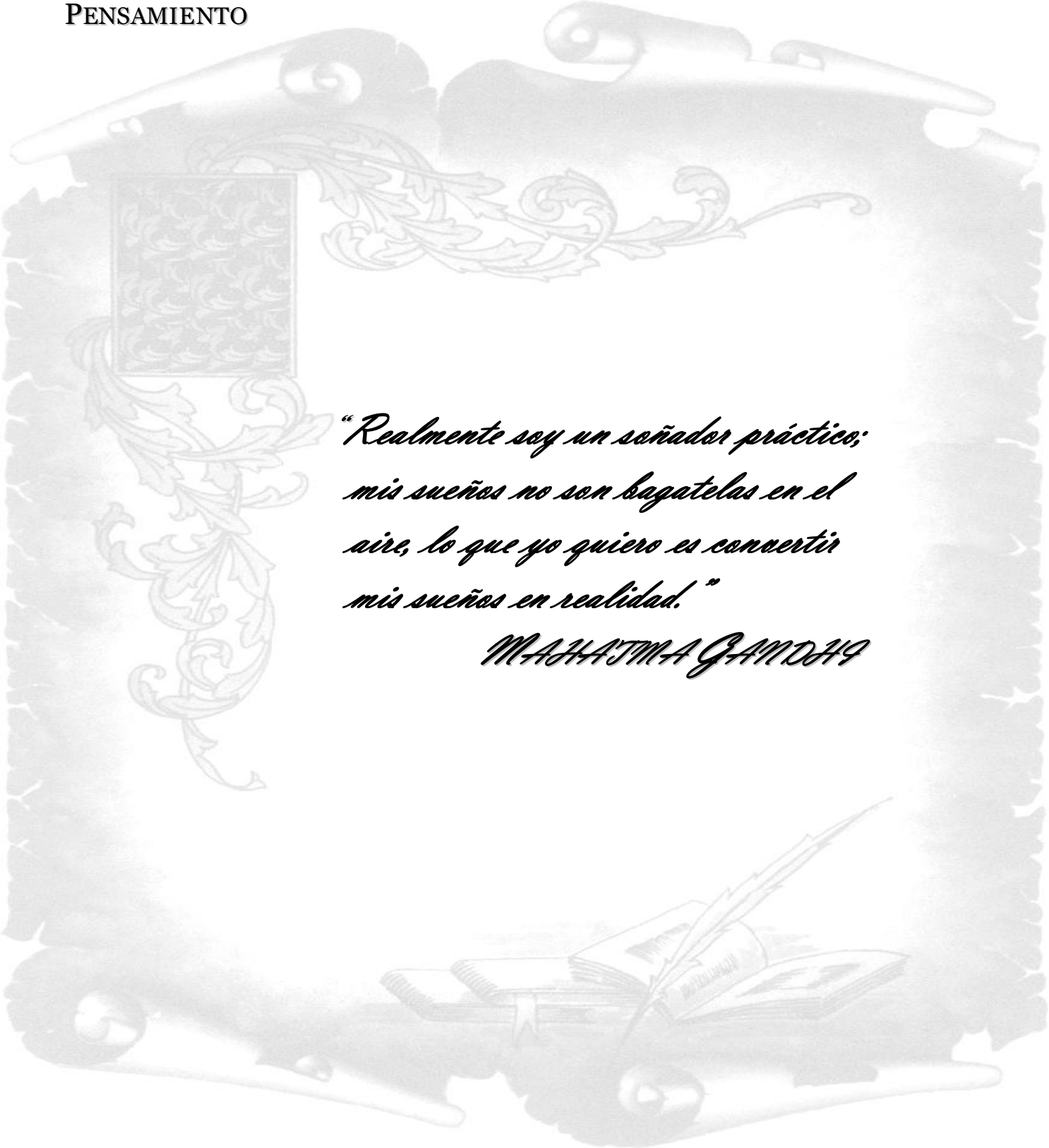
Firma del Autor

Firma del Tutor

Ing. Yenisleidy Piloto Lastra

Firma del Co_Tutor

PENSAMIENTO

A decorative border in a light gray tone frames the page. It features a scroll at the top, a square with intricate floral patterns on the left, and a stack of books with a quill pen at the bottom. The scrollwork consists of elegant, flowing lines and leaf-like motifs.

*“Realmente soy un soñador práctico;
mis sueños no son bagatelas en el
aire, lo que yo quiero es convertir
mis sueños en realidad.”*

MAHATMA GANDHI

AGRADECIMIENTOS

El trabajo de diploma enmarca la culminación de un período, la terminación de los estudios pertenecientes a la educación superior en Cuba, en el que todo joven concerniente a ella, logra obtener los conocimientos necesarios del profesional que personificará. Se puede decir que es un sueño, una utopía que se ha transformado tras cinco años de sacrificio, abnegación, responsabilidad y mucha consagración en una realidad perceptible.

Durante la elaboración de este trabajo se hace necesario agradecer a muchas personas que estuvieron involucradas directa o indirectamente, pero que sin duda alguna, brindaron su mano y su apoyo constante, he hicieron posible que este gran día llegase.

Ante todo quisiera agradecer a mis padres, esa figuras que sin titubeos son en mi vida lo mejor de este mundo, agradecerles por confiar en mí, por apoyarme como lo han hecho siempre, por educarme y forjar en mí un hombre de principios, por brindarme ese amor permanente y comprensión infinita... gracias de todo corazón y sepan siempre, que estoy orgulloso de ser su hijo.

A mi hermana, mi única hermana, por enseñarme a ser fuerte antes las adversidades, por convertirme en su guardián, en su apoyo invariable, por darme la felicidad de tener una linda sobrina.

A mi sobrinita del alma, a la princesita de mi vida, por inspirarme a seguir adelante, por conquistar mi corazón con su sonrisa y sus maldades, por ser el motivo real de lo que hoy soy capaz de brindar.

A mis abuelos todos, la cabecera de la familia, porque sin ellos la felicidad hoy sería incompleta.

A mis segundas madres, Argelia, Nuria, Coca, por educarme siempre y darme el cariño imperecedero que las caracteriza.

A mi tata Mary por empeñarse en la impresión de la tesis.

Agradecer en sentido general a toda la familia, sin dudas gigantesca, porque en ella se concentra el amor que nos inculcaran nuestros ancestros.

Quisiera agradecer a los amigos y vecinos del barrio y a alguien que aunque la vida no le dio la posibilidad de verme graduado, Jorge (Boniato de cariño) le estoy muy agradecido por comportarse como mi segundo papá.

A Yaima, por ser la confidente de toda la vida, por estar siempre cuando necesitaba un consejo o un regaño, por transitar conmigo en el espacio y en el tiempo aunque no estemos juntos, por ser hoy, la amiga insustituible.

A Dalaity por estar siempre a mi lado durante la carrera, por soportar todas mis malcriadece, por brindarme el apoyo necesario, por enseñarme a expresar lo que siento.

A Pico y Yuniór, compañeros de cuarto y además amigos, por poder contar siempre con su apoyo, por enseñarme que no existen dificultades si hay sacrificio.

A mi colectivo de amigos universitarios más cercanos, al grupo del carácter personalizado, Natalia, Yasniel, Yanisvel, Carlos, Rosalba, Reimis y nuevamente Pico y Dalaity por brindarme la seguridad y la confianza de seguir adelante.

A mis compañeros de toda la vida, a los viejos y nuevos, a los que ya no están, a esos compañeros de aula, de brigada, de comité de base, por instruirme y ayudarme cuando lo necesité, por estar siempre en las buenas y en las malas, por enseñarme a aceptar los diferentes tipos de personalidades que existen.

A los amigos que aparecieron en mi vida sin esperarlos y que me apoyaron con sinceridad y respeto María Elena (Madre Teresa de Calcuta), Inés (por su apoyo y preocupación), Greisy y Rosa (consejeras, estilistas y sobre todo la cabeza de mi familia universitaria), María Teresa (la compañera que nunca se ha podido aprender mi nombre) a Dianelys (Diana de cariño) por luchar junto a mí en esta batalla.

A Rolian y a Humberto por ser parte esencial del desarrollo íntegro del trabajo.

A todos los maestros y profesores que de una u otra forma han logrado formarme como un profesional íntegro.

A mi profesora de marxismo del IPVCE “Ernesto Che Guevara” por educarme para que en los momentos imprescindibles me comportara como tal, sé que si aún viviera, estaría orgullosa y feliz de verme graduado.

A mis tutores por depositar confianza en mí y brindarme su apoyo diario, por estar presentes en todo momento.

No por último, sería menos sentido, al agradecimiento por la Revolución Cubana, por su líder irrefutable, ese gigante de 7 leguas, nuestro Comandante en Jefe Fidel Castro, por tener la grandiosa idea de construir esta hermosa Universidad, por darme la posibilidad y la oportunidad de estudiar y formarme hoy como profesional de la informática en Cuba.

En fin, agradezco a todas las personas que con anterioridad fueron mencionadas porque siempre estuvieron en la misma trinchera de combate.

“Hasta la Victoria Siempre”

DEDICATORIA

Dedico este trabajo a mis padres, por consagrarse a mi formación profesional y personal sin recelos, por amarme como yo los amo, a mi madre, la intransigencia personalizada, por instruirme y demostrarme que la sinceridad, la combatividad y el sacrificio, son los que hacen crecer al ser humano, a mi padre, señor que todo lo puede, por tener siempre en el centro de su vida a sus dos hijos, por dedicarse completamente a la realización de un sueño, ver a su hijo más pequeño graduado, a mi hermana y a mi sobrinita por confiar siempre en mí y asentarme como ejemplo, a mis cuatro abuelitos que hoy pueden verme graduado y a mi familia toda por estar siempre apoyándome en todo momento. Y sepan que: "Ocurra lo que ocurra, aún en el día más borrascoso, los amaré siempre."



RESUMEN

Con el desarrollo de las Tecnologías de la Información y las Comunicaciones que devienen día tras día a nivel mundial, la utilización de las aplicaciones que se sustentan con tecnologías Web se convierten en una necesidad para empresas, instituciones sociales y demás entidades, debido a la posibilidad que estas aportan a sus cliente, entregando al mismo información actualizada así como rapidez y fácil manipulación del contenido que le es de su interés. El presente trabajo de diploma tiene como objetivo la implementación de una aplicación Web para la gestión de la información de Java Archives. El desarrollo de la misma estuvo guiado por las especificaciones que propone la metodología RUP lo que conllevó a la obtención de diferentes artefactos que se generan en cada uno de los flujos de trabajo por los que se transita, dada la metodología especificada. Para la implementación de la misma se utilizaron diferentes herramientas que se sustentan bajo la condición de libres y de código abierto, resultado de la investigación previamente realizada para poder asimilar el desarrollo de las tecnologías y tendencias actuales. El sistema posibilitará que se logre un adecuado proceso de gestión de la información de JAVA Archives en el desarrollo de software en los proyectos productivos de la Universidad de las Ciencias Informáticas, mediante una aplicación Web fácil de administrar y actualizar, donde todos los usuarios que requieran el uso de la información que se maneja, obtenga el beneficio esperado, según sus necesidades.

Palabras claves: aplicación Web, Java, gestión, tecnología.



ABSTRACT

With the development of Information and Communications Technology that arise every day worldwide, the use of applications that support Web technologies become a necessity for companies, social institutions and other entities because of the possibility these bring to their customers, delivering the same information as fast and easy manipulation of the content that it is of interest. This diploma work has the goal of implementing a Web application for managing information from Java Archives. The development was guided by the same specifications as proposed by the RUP methodology which led to the collection of different artifacts that are generated in each of the workflows for which route has given the methodology specified. To implement the application different tools were used, that are under the condition of been free and open source, resulting from researches previously done in order to assimilate the development of technologies and trends. The system will enable to achieve an appropriate process for management information of Java Archives in software development in the productive projects of the University of Informatics Science through a simple Web application to manage and update, where all the users that require the use of the handled information can obtain the expected benefits according to their needs.

Keywords: Web application, Java, management, technology.



ÍNDICE DE CONTENIDO

Introducción	1
Capítulo 1: Fundamentación Teórica.	5
1.1 Introducción	5
1.2 Estado del Arte	5
1.3 Tendencias y tecnologías	6
1.3.1 ¿Qué es una aplicación Web?	6
1.3.2 Modelo Cliente-Servidor	7
1.3.3 Servidor Web Apache	9
1.4.3 Arquitectura de 3 capas	9
1.4.4 Modelo Vista Controlador (MVC)	10
1.4.5 Lenguajes de programación para el proceso de desarrollo	11
1.4.5.1 JavaScript	12
1.4.5.2 PHP (Hiptertext Processor)	13
1.4.5.3 Java	14
1.4.5.3 Selección del lenguaje de programación a utilizar	15
1.4.6 Symfony, framework para el desarrollo Web	15
1.4.7 Lenguajes de Marcas	16
1.4.8 Ide de desarrollo (entorno de desarrollo Integrado)	18
1.4.9 Sistemas de Gestión de Bases de datos	18
1.4.9.2 Mysql	19
1.4.9.3 PostgreSQL	20
1.4.9.4 Selección del Sistema Gestor de Base de Datos	21
1.4.10 Metodologías de desarrollo	21
1.4.10.1 Proceso Unificado de Desarrollo (RUP)	22
1.4.10.2 Programación Extrema (XP)	23
1.4.10.3 Selección de la metodología a utilizar	24
1.4.11 Lenguaje Unificado de Modelado (UML)	24
1.4.12 Herramientas Case	25
1.4.12.1 Rational Rose Enterprise edition	26
1.4.12.2 Visual Paradigm	27
1.4.12.3 Selección de la herramienta case	27
1.4.13 Otras Herramientas de Apoyo para el desarrollo	27
1.5 Propuesta de solución	28
1.6 Conclusiones	29
Capítulo 2: Características del Sistema	30



2.1 Introducción	30
2.2 Modelo de Dominio	30
2.2.1 Glosario de términos	31
2.2.2 Diagrama de clases del dominio	32
2.3 Especificación de los requerimientos del Software	33
2.4 Descripción del sistema propuesto	35
2.4.1 Definición de los Casos de Uso	36
2.4.2 Definición de los Actores	36
2.4.3 Listado de los Casos de uso del sistema	36
2.4.4 Diagrama de casos de uso del sistema	39
2.4.5 Descripción textual de los casos de uso	40
2.5 Conclusiones	61
Capítulo 3: Análisis y Diseño	62
3.1 Introducción	62
3.2 Modelo de análisis	62
3.2.1 Clases del análisis, estereotipos y funciones	62
3.2.2 Diagramas de clases del análisis	63
3.3 Modelo de Diseño	67
3.3.1 Diagramas de interacción	67
3.3.2 Diseño del Sistema	78
3.3.2.1 Diagramas de clases del diseño	79
3.4 Diseño de la base de datos	85
3.4.1 Diagrama de clases persistentes	85
3.4.2 Modelo Entidad Relación	86
3.4.3 Descripción de las tablas	86
3.5 Conclusiones	88
Capítulo 4: Implementación	89
4.1 Introducción	89
4.2 Diagrama de Despliegue	89
4.3 Diagrama de Componentes	90
4.4 Conclusiones	94
Conclusiones	95
Recomendaciones	96
Referencias Bibliográficas	97
Bibliografía	98
Glosario de Términos	102



Anexos	105
Anexo 1: Diagrama de colaboración CU_Autenticar Usuario	105
Anexo 2: Diagrama de colaboración CU_Consultar Frameworks	105
Anexo 3: Diagrama de colaboración CU_Consultar Ficheros JAR	106
Anexo 4: Diagrama de colaboración CU_Descargar Frameworks	106
Anexo 5: Diagrama de colaboración CU_Descargar Ficheros JAR	107
Anexo 6: Diagrama de secuencia CU_Autenticar Usuario	107
Anexo 7: Diagrama de secuencia CU_Consultar Frameworks	108
Anexo 8: Diagrama de secuencia CU_Consultar Ficheros JAR	108
Anexo 9: Diagrama de secuencia CU_Descargar Frameworks	109
Anexo 10: Diagrama de secuencia CU_Descargar Fichero JAR	110



ÍNDICE DE FIGURAS

Figura 1: Diagrama de Clases del Dominio	32
Figura 2: Diagrama de casos de uso del Sistema.....	39
Figura 3: Diagrama de clases del análisis CU _ Autenticar Usuario.....	63
Figura 4: Diagrama de clases del análisis CU _ Gestionar Frameworks	63
Figura 5: Diagrama de clases del análisis del CU _ Gestionar Ficheros JAR	64
Figure 6: Diagrama de clases del análisis del CU _ Asignar Fichero JAR a un Framework	64
Figura 7: Diagrama de clases del análisis CU _ Consultar Ficheros JAR	65
Figura 8: Diagrama de clases del análisis del CU _ Consultar Frameworks	65
Figura 9: Diagrama de clases del análisis CU _ Descargar Ficheros JAR	66
Figura 10: Diagrama de clases del análisis CU _ Descargar Frameworks	66
Figura 11: Diagrama de colaboración CU _ Gestionar Frameworks Escenario: Adicionar Frameworks.....	68
Figura 12: Diagrama de colaboración CU _ Gestionar Frameworks Escenario_Eliminar Frameworks.....	68
Figura 13: Diagrama de colaboración CU _ Gestionar Frameworks Escenario _ Modificar Frameworks	69
Figura 14: Diagrama de colaboración CU _ Gestionar Frameworks Escenario _ Ver Frameworks.....	69
Figura 15: Diagrama de colaboración CU _ Gestionar ficheros JAR Escenario _ Adicionar Ficheros JA R.....	70
Figura 16: Diagrama de colaboración CU _ Gestionar Ficheros JAR Escenario _ Eliminar Ficheros JAR	70
Figura 17: Diagrama de colaboración CU_Gestionar Ficheros JAR Escenario _ Modificar Ficheros JAR.....	71
Figura 18: Diagrama de colaboración CU_Gestionar Ficheros JAR Escenario _ Ver Ficheros JAR	71
Figura 19: Diagrama de colaboración CU_Asignar Fichero JAR a un Framework.....	72
Figura 20: Diagrama de secuencia CU _ Gestionar Frameworks Escenario _ Adicionar Frameworks.....	72
Figura 21: Diagrama de secuencia CU _ Gestionar Frameworks Escenario _ Eliminar Frameworks	73
Figura 22: Diagrama de secuencia CU _ Gestionar Frameworks Escenario _ Modificar Frameworks	74
Figura 23: Diagrama de secuencia CU _ Gestionar Frameworks Escenario _ Ver Frameworks.....	75
Figura 24: Diagrama de secuencia CU _ Gestionar Fichero JAR Escenario _ Adicionar Fichero JAR	75
Figura 25: Diagrama de secuencia CU _ Gestionar Fichero JAR Escenario _ Eliminar Ficheros JAR	76
Figura 26: Diagrama de secuencia CU _ Gestionar Ficheros JAR Escenario _ Ver Ficheros JAR	76
Figura 27: Diagrama de secuencia CU _ Gestionar ficheros JAR Escenario_ Modificar Ficheros JAR	77
Figura 28: Diagrama de secuencia CU _ Asignar un Fichero JAR a un Framework	78
Figura 29: Diagrama de clases del diseño CU _ Autenticar Usuario	79
Figura 30: Diagrama de clases del diseño CU _ Asignar Fichero JAR a un Framework	80
Figura 31: Diagrama de clases del diseño CU _ Gestionar Frameworks	81
Figura 32: Diagrama de clases del diseño CU _ Gestionar Ficheros jar	82
Figura 33: Diagrama de clases del diseño CU _ Consultar Frameworks	83
Figura 34: Diagrama de clases del diseño CU_Descargar Frameworks	83
Figura 35: Diagrama de clases del diseño CU _ Consultar Ficheros JAR	84
Figura 36: Diagrama de clases del diseño CU_Descargar Ficheros JAR	84
Figura 37: Diagrama de clases persistentes	85
Figura 38: Modelo Entidad Relación.....	86
Figura 39: Diagrama de Despliegue	90
Figura 40: Diagrama de componentes Autenticarse	91
Figura 41: Diagrama de componentes Paquete: Gestionar Frameworks	91
Figura 42: Diagrama de componentes Paquete: Gestionar Ficheros JAR	92
Figura 43: Diagrama de componentes Paquete: Asignar Fichero JAR a un Framework	92
Figura 44: Diagrama de componentes Paquete: Descargar Framework	93
Figura 45: Diagrama de componentes Paquete: Descargar Ficheros JAR	93
Figura 46: Diagrama de colaboración del CU _ Autenticar Usuario	105
Figura 47: Diagrama de colaboración del CU _ Consultar Frameworks	105



Figura 48: Diagrama de colaboración del CU _ Consultar Fichero JAR	106
Figura 49: Diagrama de colaboración del CU_Descargar Frameworks	106
Figura 50: Diagrama de colaboración del CU_Descargar Ficheros JAR	107
Figura 51: Diagrama de secuencia del CU_Autenticar Usuario	107
Figura 52: Diagrama de secuencia del CU _ Consultar Frameworks	108
Figura 53: Diagrama de secuencia del CU _ Consultar Ficheros JAR	108
Figura 54: Diagrama de secuencia del CU_Descargar Frameworks	109
Figura 55: Diagrama de secuencia del CU_Descargar Fichero JAR	110



ÍNDICE DE TABLAS

Tabla 1: Actores del sistema	36
Tabla 2: Caso de uso Autenticar usuario	37
Tabla 3: Caso de uso Gestionar Frameworks.....	37
Tabla 4: Caso de uso Gestionar Ficheros JAR	37
Tabla 5: Caso de uso Asignar Fichero JAR a un Framework.....	37
Tabla 6: Caso de uso Consultar Ficheros JAR	38
Tabla 7: Caso de uso Consultar Frameworks.....	38
Tabla 8: Caso de uso Descargar Ficheros JAR.....	38
Tabla 9: Caso de uso Descargar Frameworks.....	38
Tabla 10: Descripción textual del CU _ Autenticar Usuario	41
Tabla 11: Descripción textual del CU_ Gestionar Frameworks	47
Tabla 12: Descripción textual del CU _ Gestionar Ficheros JAR	52
Tabla 13: Descripción textual del CU _ Asignar Fichero JAR a un Framework.....	54
Tabla 14: Descripción textual del CU _ Consultar Ficheros JAR	55
Tabla 15: Descripción textual del CU_ Consultar Frameworks	56
Tabla 16: Descripción textual del CU _ Descargar Ficheros JAR	58
Tabla 17: Descripción textual del CU _ Descargar Frameworks.....	60
Tabla 18: Clases del Análisis.....	62
Tabla 19: Clases del Diseño.....	79
Tabla 20: Descripción de la tabla Framework.....	86
Tabla 21: Descripción de la tabla Fichero JAR	87
Tabla 22: Descripción de la tabla Usuario.....	87



INTRODUCCIÓN

Actualmente el desarrollo mundial está sujeto al progreso y perfeccionamiento de las Tecnologías de la Investigación y las Comunicaciones. Es por ello que se evidencian notorios avances en la industria del software en países como Japón, China, Estados Unidos, Canadá, incluyendo además, algunos ejemplares de América Latina, dentro de los que se pueden citar: Argentina, Brasil, Chile... El beneficio de esta industria está estrechamente vinculado al avance de las tecnologías existentes en el mundo, al mejoramiento de las relaciones comerciales y a soslayar a gran escala los errores que comúnmente el hombre comete.

Cuba no puede estar aislada del desarrollo interno que provocan en una nación el avance de las tecnologías y, menos aún, incomunicada con la idea de ser ella misma quien con sus fuerzas de trabajo, con sus recursos humanos pueda ser partícipe del desarrollo de las tecnologías o ciencias aplicadas. Es por ello que se vienen tomando medidas para desplegar el avance tecnológico por todo el país y con ello se está logrando uno de los pilares fundamentales, primero distribuirla y luego desarrollar las ciencias de la computación, para lograr en un futuro insertar en el mercado internacional los productos nacionales que podrían suceder con el mismo progreso de los procesos tecnológicos.

La Industria Cubana del Software está llamada a convertirse en una significativa fuente de ingresos para el país, por lo que se han establecido estrategias con el objetivo de lograr su inserción en el mercado mundial. “La idea es convertir la informática en una de las ramas más productivas y recaudadora de recursos para la nación. Es el empleo a fondo de la inteligencia y del capital humano que tenemos y principalmente del que podemos crear casi como espina dorsal de la economía.” (Castro, 2006)

En el año 2002 se materializa en la Nación Cubana la creación del motor impulsor de tal encomienda, se crea la Universidad de Las Ciencias Informáticas, centro docente-productivo en el cual se forman profesionales de la informática de un alto nivel intelectual y profesional. El centro no sólo tiene como meta la formación de ingenieros competentes en materia de software sino que también pretende crear productos, servicios y soluciones informático-integrales para lograr una eficiente informatización de la sociedad nacional y extranjera, lo que se lleva a cabo a través de un amplio grupo de proyectos que existen en el centro universitario.

Esta institución educacional, a pesar de tener pocos años de experiencia ha obtenido numerosos lauros en el exterior, con la realización de los distintos proyectos productivos de los que se puede citar al proyecto Humanización Penitenciaria de Venezuela, además, el Cuerpo de Investigaciones Científicas Penales y Criminalísticas (CICPC) de la misma nación, entre otros.



Varios de estos proyectos son caracterizados como software de gestión: programas de ordenador creados para la gestión de grandes cantidades de datos. Estos utilizan diferentes lenguajes de programación, gestores de bases de datos, metodologías de desarrollo; los cuales reportan directamente a las funcionalidades que deben brindar el producto, herramientas y métodos que son seleccionados para darle una mejor solución al problema que se puede presentar.

Uno de los lenguajes de programación más utilizado para la creación de software es JAVA, teniendo en cuenta las ventajas que aportan sus características, las cuales ayudan a una mejor aplicación del mismo y encaminan el principio de funcionamiento a un éxito final. Los desarrolladores de JAVA utilizan en gran medida diferentes frameworks para el desarrollo de aplicaciones empresariales que gravitan alrededor de un mejor uso de la programación ya que estos son los Generadores de Aplicación que se relacionan directamente con un dominio específico, es decir, con una familia de problemas relacionados [1]. Estos frameworks almacenan sus clases en unos ficheros llamados **JAR** (Java **AR**chive). Por tanto, para utilizar cada uno de estos frameworks es necesario conocer cuáles ficheros JAR son requeridos para el mismo, tema que se dificulta por la gran cantidad existente y por la gran cuantía de ficheros JAR que necesite cada uno. Por lo anteriormente expresado surge la necesidad de crear un sistema que permita tener controlado qué ficheros JAR se requieren para ser usados en un framework determinado y, a su vez, tenerlos almacenados y brindar acceso a dichos ficheros JAR.

Después de analizar la situación antes expuesta, se define como **Problema a Resolver** la siguiente interrogante: ¿Cómo brindar información detallada sobre la relación existente entre frameworks y ficheros JAR, para lograr una utilización eficiente en el desarrollo de aplicaciones informáticas en los proyectos productivos de la Universidad de las Ciencias informáticas?

Se precisa como **Objeto de estudio** el proceso de Gestión de la Información referente a los frameworks de JAVA y sus ficheros JAR.

El **Campo de Acción** se enfoca en el Proceso de Gestión de la Información referente a los frameworks de JAVA y sus ficheros JAR en los proyectos productivos dentro de la Universidad de las Ciencias Informáticas.

La **Idea a Defender** se enfoca como sigue: Si se implementa una Aplicación Web que posibilite la Gestión de la Información de Java Archives entonces se obtendría una utilización eficiente de los frameworks de Java con sus ficheros JAR en el desarrollo de aplicaciones informáticas de los proyectos productivos de la Universidad de las Ciencias informáticas.

El **Objetivo General** que se puntualiza para dar respuesta al problema planteado es desarrollar una aplicación Web que permita la gestión y concentración de la información pertinente a los frameworks de



JAVA y sus ficheros JAR garantizando la calidad, seguridad, fiabilidad de los resultados, y consistencia de la información.

Para dar cumplimiento al objetivo general se plantean las siguientes **Tareas de Investigación**:

1. Definir los frameworks utilizados en los proyectos productivos UCI.
2. Realizar el estudio del arte a partir de aplicaciones anteriores o semejantes.
3. Asimilar las tendencias y tecnologías actuales para llevar a cabo el proceso de desarrollo.
4. Realizar el modelamiento del negocio.
5. Definir patrones de arquitectura a utilizar.
6. Hacer el levantamiento de los requisitos funcionales y no funcionales del sistema.
7. Realizar el análisis y el diseño de la propuesta de solución.
8. Confeccionar el modelo de implementación y despliegue.

El sistema informático propuesto será de gran utilidad para la gestión de las actividades diarias de estudiantes y profesores de la Universidad de las Ciencias Informáticas o de cualquier personal vinculado a los proyectos productivos y al desarrollo de software de gestión específicamente, debido a que con el mismo se podrán obtener los siguientes **aportes prácticos**:

- ✓ Un sistema informático que gestione toda la información concerniente a los framework de JAVA y los ficheros JAR que utiliza cada uno.
- ✓ Brindar un servicio único de almacenamiento de ficheros JAR y conjuntamente con ellos los diferentes framework utilizados en el centro universitario o a nivel mundial.
- ✓ Posibilitar la obtención de información fiable referente al tema tratado.
- ✓ Mejorar la toma de decisiones de los desarrolladores en cuanto a la utilización o no de un framework determinado.

El presente documento está compuesto por cuatro capítulos, que incluyen todo lo relacionado con el trabajo investigativo, así como el análisis, diseño y la implementación del sistema.

En el **Capítulo I** Fundamentación Teórica: Comprende un análisis de los sistemas que existen en la actualidad y se vinculan con la investigación, y el estado del arte de las tecnologías y herramientas a utilizar en el desarrollo de la aplicación.

En el **Capítulo II** Características del Sistema: Se refleja la investigación realizada con los procesos que tienen lugar en el negocio como objeto de estudio, la situación problemática existente además del objetivo general a cumplir durante el desarrollo del producto. Se describe la propuesta del sistema, se aborda lo



referente al funcionamiento del negocio, sus reglas, descripción y las mejoras que propone el mismo. Se describe además la solución propuesta, utilizando los requerimientos funcionales y no funcionales, los casos de uso, el diagrama de casos de uso del sistema y un prototipo de interfaz de usuario.

En el **Capítulo III** Análisis y Diseño del Sistema: Se realiza el análisis del sistema a desarrollar, con el propósito de refinar y estructurar los requisitos obtenidos con anterioridad para facilitar la comprensión, preparación, modificación y mantenimiento de los mismos. Describe los aspectos relacionados al diseño de la solución propuesta, se modelan los diagramas de clases del diseño y se especifican los principios para el diseño gráfico.

En el **Capítulo IV** Implementación: Trata los aspectos relacionados con la construcción de la solución propuesta, se modelan los diagramas de componentes y despliegue, aborda la descripción de los estándares de diseño, codificación y además del tratamiento de errores en la solución del sistema.



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

1.1 INTRODUCCIÓN

En el presente capítulo se hace referencia de los elementos teóricos que fundamentan el desarrollo del sistema informático propuesto. También contempla un análisis efectuado sobre el estudio del arte de las posibles herramientas y tecnologías que pudieran ser utilizadas para el desarrollo, posibilitando la toma de decisiones sobre qué es más factible utilizar en función de cumplimentar los objetivos generales de la investigación y solución del problema planteado.

1.2 ESTADO DEL ARTE

En la actualidad la gran mayoría de los desarrolladores de software optan por el uso de los frameworks¹ para lograr así un mejor tratamiento de los procesos ingenieriles de cualquier propuesta para la confección o mejoramiento de un software determinado.

Su uso cada día se hace más popular debido a que aporta grandes ventajas:

Modularidad y reducción de la complejidad: La aplicación está formada por subsistemas especializados en distintos aspectos fundamentales de toda aplicación (persistencia, presentación, manejo de Log²).

Fortaleza al cambio: Los módulos pueden ser evolucionados o cambiados conservando la arquitectura global de la aplicación.

Documentación: La documentación del framework promueve el uso correcto del mismo y disminuye el esfuerzo necesario para el mantenimiento.

Estructura: El desarrollo basado en frameworks establece una estructura sobre la cual las aplicaciones pueden ser construidas, liberando al desarrollador de tomar el 100% de las decisiones de diseño.

Distribución de funciones: Permite paralelizar el trabajo de desarrollo ya que la solución puede desplegarse como un conjunto de piezas independientes que encajarán en el framework usado.

Eficiencia: El desarrollador puede concentrarse en los requerimientos funcionales de la aplicación.

¹ *Un framework se puede definir como una estructura de soporte en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado, entre otros, para contribuir con el progreso de los diferentes componentes de un proyecto. Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio, provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.*

² *Un log es un registro oficial de eventos durante un periodo de tiempo en particular.*



Aplicaciones ricas: Facilitan mayor funcionalidad a los usuarios de la aplicación.

Se puede encontrar uso de frameworks en todos los lenguajes de programación como S_Char (C#), Hipertext Processor (PHP), Java, etc. Un framework puede ser algo tan grande como “.NET” pero el concepto se aplica en ámbitos más específicos, por ejemplo; dentro de Java, el de aplicaciones Web, posibilita encontrar los frameworks: Struts, “Java Server Faces”, o Spring. Estos frameworks de Java en la práctica son conjuntos de librerías (API's) para desarrollar aplicaciones Web, unido a librerías para su ejecución (o motor) y un conjunto de herramientas para facilitar esta tarea (debuggers, ambientes de desarrollo como Eclipse).

La aplicación DOCJAR [2] ha sido la más completa encontrada en el estudio realizado debido a las funcionalidades que brinda a cualquier usuario, posee un buscador con diferentes criterios de búsqueda relacionando los frameworks de Java y los ficheros JAR que utilizan, pero no ofrece una información puntualizada en el uso de los componentes mencionados. Por otra parte se especifica además la existencia de otro sitio con el nombre Find JAR [3] brindando las mismas funcionalidades que el DOCJAR pero con menor nivel de dificultad.

En Cuba, como en el resto del mundo, los desarrolladores de software utilizan frameworks para la implementación de cualquier sistema; muchos de estos programadores o ingenieros se pueden encontrar en la Universidad de las Ciencias Informáticas. Hoy en día no se cuenta con sitio Web ni aplicación que gestione lo que anteriormente se especifica.

Todos los sistemas citados poseen características que pueden agruparse en un solo software ofreciendo mayor funcionalidad, concentrando la información y gestionando la relación existente entre frameworks y ficheros JAR.

1.3 TENDENCIAS Y TECNOLOGÍAS

Durante el progreso de este epígrafe se exponen conceptos básicos muy utilizados en el campo de la informática, los cuales se encuentran estrechamente relacionados con arquitectura, metodología, lenguaje de modelado, herramientas, patrones de diseño, cliente, servidor, entre otras.

1.3.1 ¿QUÉ ES UNA APLICACIÓN WEB?

Una aplicación Web es una aplicación informática que los usuarios pueden utilizar accediendo a un servidor Web a través de Internet o de una intranet mediante un navegador. En otras palabras, es un software que se codifica en un lenguaje soportado por los navegadores Web (HTML, JavaScript, entre otros) en la que se confía la ejecución al navegador. Las aplicaciones Web son populares debido a lo



práctico del navegador como cliente ligero, así como a la facilidad para actualizar y mantener aplicaciones sin distribuir e instalar software a miles de usuarios potenciales. Se diferencian de las aplicaciones informáticas tradicionales (de escritorio), en que se manejan precisamente a través de un navegador. Son tan funcionales como el software de escritorio, en algunos casos más, y presentan numerosas ventajas:

- ✓ Para su utilización lo único que se necesita es disponer de conexión a una red.
- ✓ Son aplicaciones muy ligeras ya que el navegador de Internet no contiene el programa sino sólo las interfaces que permiten operar con él.
- ✓ Consumen muy pocos recursos del equipo en el que están instaladas.
- ✓ Son más fáciles de actualizar y mantener que el software convencional.
- ✓ Los usuarios pueden participar en la elaboración de los contenidos.
- ✓ Se pueden distribuir e instalar en miles de equipos.
- ✓ Su funcionalidad es independiente del sistema operativo instalado por el usuario.

1.3.2 MODELO CLIENTE-SERVIDOR

El modelo Cliente-Servidor empezó a ser aceptado a finales de los 80's. La separación entre cliente y servidor es de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores Web, los de archivo, los de correo, mientras que sus propósitos varían, la arquitectura básica seguirá siendo la misma.

Su principio de funcionamiento es sencillo pues se tiene una máquina cliente que requiere un servicio de una máquina servidor y éste realiza la función para la que está programado. Un único servidor típicamente sirve a una multitud de clientes, ahorrando a cada uno de ellos el problema de tener la información instalada y almacenada localmente.

Los sistemas Cliente-Servidor pueden ser de muchos tipos, dependiendo de las aplicaciones que el servidor pone a disposición de los clientes. Ejemplo de ellos:

- ✓ Servidor de Impresión: Mediante el cual los usuarios comparten impresoras.
- ✓ Servidor de Archivos: Con el cual los clientes comparten discos duros.
- ✓ Servidor de Bases de Datos: Donde existe una única base de datos.
- ✓ Servidor de Aplicaciones Web: Mediante el cual se comparte información entre servidores y usuarios a través de la red.



Existen diversos modelos comunes de máquinas clientes en entorno Cliente-Servidor. Uno de los clientes más populares es una computadora personal basada en Intel que ejecuta aplicaciones de DOS en un entorno Windows.

1.3.2.1 CARACTERÍSTICAS DEL CLIENTE

El cliente es el proceso que permite al usuario formular los requerimientos y pasarlos al servidor, es decir, requiere un servicio del mismo. Es una estación de trabajo o computadora que está conectada a una red a través de la cual puede acceder al servidor, normalmente tiene en cuenta todas las funciones relacionadas con la manipulación y despliegue de datos, por lo que están desarrollados sobre plataformas que permiten construir Interfaces Gráficas de Usuario (GUI), además de acceder a los servicios distribuidos en cualquier parte de una red.

1.3.2.2 FUNCIONES DEL CLIENTE

- ✓ Administrar la interfaz de usuario.
- ✓ Interactuar con el usuario.
- ✓ Procesar la lógica de la aplicación y hacer validaciones locales.
- ✓ Generar requerimientos de bases de datos.
- ✓ Recibir resultados del servidor.
- ✓ Formatear resultados.

1.3.2.3 CARACTERÍSTICAS DEL SERVIDOR

Es el proceso encargado de atender a múltiples clientes que realizan peticiones de algún recurso administrado por él, normalmente maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos. Es un programa que recibe una solicitud, realiza el servicio requerido y devuelve los resultados en forma de una respuesta.

1.3.2.4 FUNCIONES DEL SERVIDOR

- ✓ Aceptar los requerimientos de bases de datos que hacen los clientes.
- ✓ Procesar requerimientos de bases de datos.
- ✓ Formatear datos para transmitirlos a los clientes.
- ✓ Procesar la lógica de la aplicación y realizar validaciones a nivel de bases de datos.



1.3.3 SERVIDOR WEB APACHE

La piedra angular de cualquier portal es, con toda seguridad, el servidor Web; el software encargado de atender las peticiones de los clientes y enviarles las páginas Web solicitadas. Existen multitud de paquetes software para montar servidores Web, muchos de ellos distribuidos como software libre y siendo, sin lugar a dudas, el más popular de todos: Apache.

Una de las principales características que presenta Apache es que funciona en plataformas virtuales muy utilizadas.

Este presenta además muchas otras características, entre ellas un elaborado índice de directorios; un directorio de alias; negociación de contenidos; informe de errores HTTP configurable; ejecución SetUID de programas CGI; gestión de recursos para procesos hijos; integración de imágenes del lado del servidor; reescritura de las URL; comprobación de la ortografía de las URL; y manuales online.

Apache es un servidor Web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos (HTTP 1.1). Entre sus características destacan:

- ✓ Multiplataforma.
- ✓ Es un servidor de Web conforme al protocolo HTTP/1.1.
- ✓ Modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.
- ✓ Basado en hebras en la versión 2.0.
- ✓ Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.
- ✓ Se desarrolla de forma abierta.
- ✓ Extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.

Pero una de las características fundamentales de Apache es su flexibilidad a la hora de ser configurado. Dicha configuración está basada en módulos, que pueden ser añadidos o eliminados según interese su funcionalidad o no. Otra de las características más destacadas en Apache es que puede escuchar las peticiones para diferentes direcciones IP. Con ello, se puede centralizar el servicio de diferentes dominios Web con direcciones IP diferentes, en un único servidor de Web (hosting de varios dominios).

1.4.3 ARQUITECTURA DE 3 CAPAS

La programación por capas se basa en la separación de la capa de presentación, capa de negocio y la capa de acceso a datos. Su ventaja principal es que el desarrollo se puede llevar a cabo en varios



niveles y, en caso de que sobrevenga algún cambio, sólo se consulta a la capa que lo requiera sin tener que revisar todo el código. Además permite distribuir el trabajo de creación de una aplicación por niveles; cada grupo de trabajo está totalmente abstraído del resto, de forma que basta con conocer la API³ (Application Programming Interface) que existe entre niveles.

1.4.3.1 CAPAS O NIVELES

Capa de Presentación: Presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso. Se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser amigable para el usuario, generalmente se presentan como formularios.

Capa de Negocio: Se denomina capa de negocio o lógica del negocio porque es aquí donde se establecen todas las reglas que deben cumplirse. Recibe las peticiones del usuario y envía las respuestas tras el proceso. Se comunica con la capa de presentación para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él.

Capa de Acceso a Datos: Es donde residen los datos y encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, recibe solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

1.4.4 MODELO VISTA CONTROLADOR (MVC)

MVC es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones Web. [4]

Modelo: Incorpora la capa del dominio y persistencia, es la encargada de guardar los datos en un medio persistente (ya sea una base de datos, un archivo de texto, Extensible Markup Language en español Lenguaje de Marcas Ampliable (XML) o registro). Es donde se hace el levantamiento de todos los objetos que un sistema debe utilizar, es el proveedor de los recursos. Administra el comportamiento y

³ Es el conjunto de funciones y procedimientos o métodos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Ejemplos API for SCSI device interfacing, OpenGL, Microsoft Framework .NET, Symfony para PHP, Javascript-C de Mozilla.



los datos del dominio de aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador).

Vista: Se encarga de presentar la interfaz al usuario en sistemas Web, típicamente HTML, aunque pueden existir otros tipos de vistas. Maneja la visualización de la información.

Controlador: Es el que escucha los cambios en la vista y se los envía al modelo, el cual le regresa los datos a la vista. Es un ciclo donde cada acción del usuario origina que se inicie un nuevo ciclo. Controla el flujo entre la vista y el modelo (los datos).

1.4.4.1 VENTAJAS DEL PATRÓN MODELO-VISTA-CONTROLADOR

Soporte de múltiples vistas: Dado que la vista se halla separada del modelo y no existe una dependencia directa entre ambos, la interfaz de usuario puede mostrar múltiples vistas de los mismos datos simultáneamente. Por ejemplo, múltiples páginas de una aplicación Web pueden utilizar el mismo modelo de objetos mostrado de modos diferentes.

Adaptación al cambio: Los requerimientos de interfaz de usuario tienden a cambiar con mayor rapidez que las reglas del negocio. Dado que el modelo no depende de la vista, agregar nuevas opciones de presentación generalmente no afecta al mismo.

Desventaja:

El costo de actualizaciones frecuentes: Si el modelo experimenta cambios frecuentes podría desbordar las vistas con una lluvia de requerimientos de actualización.

1.4.5 LENGUAJES DE PROGRAMACIÓN PARA EL PROCESO DE DESARROLLO

Actualmente existen diferentes lenguajes de programación que han ido surgiendo debido a las tendencias y necesidades que presentan las plataformas. Desde los inicios de la Internet fueron floreciendo diferentes demandas por los usuarios y se dieron soluciones mediante lenguajes estáticos. Con el transcurso del tiempo las tecnologías han evolucionado germinando nuevos problemas. Esto dio lugar a ampliar lenguajes dinámicos de programación para la Web, que permitieran interactuar con los usuarios y utilizaran sistemas de Bases de Datos.



Los Lenguajes de Programación orientados a la Web se clasifican en lenguajes del lado del cliente y lenguajes del lado del servidor. Entre los lenguajes que trabajan del lado del servidor se pueden citar algunos como son PERL, Active Server Page (ASP), Hipertext Processor (PHP) y JavaServer Pages (JSP), que son los más usados en la actualidad. Desarrollan la lógica de negocio y además se encargan del acceso a los distintos Sistemas de Gestión de Bases de Datos. Del lado del cliente se encuentran principalmente JavaScript, JScript, Extensible Stylesheet Language, en español "Lenguaje Extensible de Hojas de Estilo (XSL) y el Visual Basic Script (VBScript). Estos son los encargados de aportar dinamismo a la aplicación en los navegadores.

1.4.5.1 JAVASCRIPT

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas Web dinámicas. Una página Web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario [5]. El navegador del cliente es el encargado de interpretar las instrucciones JavaScript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único con que cuenta este lenguaje es el propio navegador.

Es un lenguaje interpretado que no requiere compilación. Es similar a Java, aunque no es un lenguaje orientado a objetos, el mismo no dispone de herencias. La mayoría de los navegadores en sus últimas versiones interpretan código Javascript. Para evitar incompatibilidades el World Wide Web Consortium (W3C) diseñó un estándar denominado DOM (en inglés Document Object Model, en su traducción al español Modelo de Objetos del Documento).

1.4.5.1.1 VENTAJAS

- ✓ Lenguaje de script seguro y fiable.
- ✓ Los script tienen capacidades limitadas por razones de seguridad.
- ✓ El código se ejecuta en el cliente.

1.4.5.1.2 DESVENTAJAS

- ✓ Código visible por cualquier usuario.
- ✓ El código debe descargarse completamente.
- ✓ Puede poner en riesgo la seguridad del sitio, con el actual problema llamado XSS (significa en inglés Cross Site Scripting renombrado a XSS por su similitud con las hojas de estilo CSS).



1.4.5.2 PHP (HIPERTEXT PROCESSOR)

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas Web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor [6]. No necesita ser compilado para ejecutarse. Para su funcionamiento requiere tener instalado Apache o IIS con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas.

1.4.5.2.1 VENTAJAS

- ✓ Fácil de aprender.
- ✓ Se caracteriza por ser un lenguaje muy rápido.
- ✓ Soporta en cierta medida la orientación a objeto. Clases y herencia.
- ✓ Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- ✓ Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- ✓ Capacidad de expandir su potencial utilizando módulos.
- ✓ Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.
- ✓ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- ✓ Incluye gran cantidad de funciones.
- ✓ No requiere definición de tipos de variables ni manejo detallado del bajo nivel.
- ✓ Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- ✓ Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación.
- ✓ Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.

1.4.5.2.2 DESVENTAJAS

- ✓ Se necesita instalar un servidor Web.
- ✓ Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.
- ✓ La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- ✓ La programación orientada a objetos es aún muy deficiente para aplicaciones grandes.
- ✓ Dificulta la modularización y la organización por capas de la aplicación.



Está diseñado específicamente para ser un lenguaje más seguro para escribir programas CGI que Perl o C, y con la selección correcta de opciones de configuración en tiempos de compilación y ejecución, y siguiendo algunas prácticas correctas de programación.

1.4.5.3 JAVA

Java es un lenguaje de programación que se utiliza para la creación de aplicaciones informáticas, ha sido diseñado a modo de eliminar las complejidades de otros lenguajes como C y C++.

1.4.5.3.1 VENTAJAS

- ✓ Lenguaje orientado a objetos, lo que hace que los programas se construyan a partir de módulos independientes, y que esos módulos se pueden transformar o ampliar fácilmente.
- ✓ Posee bibliotecas de clases estándar predefinidas como son:
 - Manejo de archivos.
 - Comunicación de datos.
 - Acceso a la red Internet.
 - Acceso a bases de datos.
 - Interfaces gráficas.
- ✓ Es multiplataforma ya que los programas pueden ser ejecutados en cualquier plataforma sin necesidad de hacer cambios.
- ✓ Es compatible a cualquier plataforma a nivel de código compilado, a nivel de fuente y a nivel de biblioteca.
- ✓ Es robusto ya que proporciona numerosas comprobaciones en compilación y en tiempo de ejecución como revisar los índices al acceder un arreglo y revisión de tipos durante la compilación (al igual que C).
- ✓ Flexible pues combina la flexibilidad, robustez y legibilidad gracias a una revisión de tipos durante la compilación y durante la ejecución. Además pueden tener apuntadores a objetos de un tipo específico y también se pueden tener apuntadores a objetos de cualquier tipo.
- ✓ Administra automáticamente la memoria porque posee un recolector de basuras el cual determina cuando se puede liberar la memoria ocupada por un objeto.
- ✓ Puede ser interpretado y compilado a la vez.



- ✓ Es multihebra soportando sincronización de múltiples hilos de ejecución (*multithreading*) a nivel de lenguaje, especialmente útiles en la creación de aplicaciones de red distribuidas.
- ✓ Dinámico
- ✓ Produce dos tipos de programas: aplicaciones independientes⁴ y applets⁵.

1.4.5.3.2 DESVENTAJAS

- ✓ Velocidad de procesamiento es lenta dado a que la máquina virtual de java es un intérprete y redonda en una falta de rendimiento con relación a aplicaciones equivalentes escritas en código máquina nativo.
- ✓ El poder reducir los problemas de acceso a memoria y liberación automática hacen de java un lenguaje poco apropiado para desarrollar aplicaciones de base como Sistemas Operativos.
- ✓ El recolector de basura es una sobrecarga adicional al procesador.
- ✓ La generalidad hace que no se aprovecha totalmente la potencia de la máquina y del sistema operativo.
- ✓ Para su comprensión y entendimiento con facilidad se requiere conocimientos básicos de C.

1.4.5.3 SELECCIÓN DEL LENGUAJE DE PROGRAMACIÓN A UTILIZAR

Para el desarrollo de la aplicación, de los lenguajes antes mencionados, aunque con similitudes, se selecciona PHP, porque además de sus características indicadas se puede enfatizar en que es un lenguaje fácil de aprender, muy rápido, posee capacidad de conexión con la mayoría de los servidores de bases de datos y posibilita asentar gran cantidad de funciones, no siendo así JAVA el cual requiere conocimientos de C para facilitar su uso, es lento y requiere de una tecnología robusta y sustentable, la cual no se posee para el desarrollo.

1.4.6 SYMFONY, FRAMEWORK PARA EL DESARROLLO WEB

Symfony, es un framework diseñado para optimizar el desarrollo de las aplicaciones Web, se encarga de separar la lógica de negocio, la lógica de servidor y la presentación de la aplicación Web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación Web

⁴ Las aplicaciones independientes se comportan como cualquier otro programa escrito en cualquier lenguaje, como por ejemplo el navegador de Web HotJava, escrito íntegramente en Java.

⁵ Las applets son pequeños programas que aparecen embebidos en las páginas Web, como aparecen los gráficos o el texto, pero con la capacidad de ejecutar acciones muy complejas, como animar imágenes, establecer conexiones de red, presentar menús y cuadros de diálogo para luego emprender acciones, etc.



compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

1.4.6.1 VENTAJAS

- ✓ Sencillo de usar.
- ✓ Fácil de instalar y configurar.
- ✓ Independiente del sistema gestor de bases de datos.
- ✓ Sigue la mayoría de mejores prácticas y patrones de diseño para la Web.
- ✓ Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- ✓ Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

1.4.6.2 Desventajas

- ✓ Posee una gran cantidad de generación de código y una gran cantidad de gastos generales de configuración.
- ✓ Requerido una estructura de directorios que se crea cuando se utiliza la herramienta de línea de comandos para crear módulos.
- ✓ Tiende a graves errores en la base de datos de Doctrina y al propulsar plugins.

1.4.7 LENGUAJES DE MARCAS

Un lenguaje de marcado o lenguaje de marcas es una forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación [7]. Son un estilo de hacer explícita la estructura de cierto tipo de información sin especificar su presentación. Suelen confundirse con lenguajes de programación, pero no son lo mismo, ya que el lenguaje de marcado no tiene las funciones aritméticas que tienen los lenguajes de programación, como las variables. Una de las principales ventajas de este tipo de codificación es que puede ser interpretada directamente, dado que son archivos de texto plano. Al tratarse únicamente de texto, los documentos son independientes de la plataforma, sistema operativo o programa con el que fueron creados.



1.4.7.1 LENGUAJE XHTML

XHTML, acrónimo inglés de eXtensible Hypertext Markup Language (lenguaje extensible de marcado de hipertexto), es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas Web. Es una familia de módulos y tipos de documentos que reproduce, engloba y extiende HTML 4.0. Los tipos de documentos de la familia XHTML están basados en XML, y diseñados fundamentalmente para trabajar en conjunto con aplicaciones de usuario basados en XML [8].

1.4.7.1.1 ¿POR QUÉ ES NECESARIO XHTML?

- ✓ Los desarrolladores de aplicaciones de usuario y documentos descubren constantemente nuevas formas de expresar sus ideas usando nuevas etiquetas, objetivo que se puede lograr con el uso de XHTML el cual lleva implícito la utilización de XML, lenguaje de marcado, que posibilita la creación de etiquetas personales.
- ✓ Constantemente se desarrollan nuevas formas de acceder a Internet y una posibilidad que brinda XHTML es que concibe la interoperabilidad con aplicaciones de usuario generales.
- ✓ Posee un nuevo mecanismo de especificación de documentos y aplicaciones de usuarios, los servidores, proxies, y aplicaciones de usuario finales pueden realizar una mejor transformación del contenido.
- ✓ Posibilita desarrollar contenido conforme a XHTML que sea utilizable por cualquier aplicación de usuario.

1.4.7.1.2 VENTAJAS RESPECTO DEL HTML

- ✓ Se pueden incorporar elementos de distintos espacios de nombres XML (como MathML y Scalable Vector Graphics).
- ✓ Un navegador no necesita implementar heurísticas para detectar qué quiso poner el autor, por lo que el parser puede ser mucho más sencillo.
- ✓ Como es XML se pueden utilizar fácilmente herramientas creadas para procesamiento de documentos XML genéricos (editores, XSLT).

1.4.7.1.3 DESVENTAJAS

- ✓ Algunos navegadores antiguos no son totalmente compatibles con los estándares, provocando que las páginas no se muestren correctamente.
- ✓ Varias herramientas de diseño Web aún no producen código XHTML correcto.



1.4.8 IDE DE DESARROLLO (ENTORNO DE DESARROLLO INTEGRADO)

1.4.8.1 ZEND STUDIO

Editor Web orientado a la programación de páginas PHP, con ayudas en la gestión de proyectos y depuración de código. Se trata de un programa de la casa Zend, impulsores de la tecnología de servidor PHP, orientada a desarrollar aplicaciones Web. [9]

El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración del código. El programa entero está escrito en Java, lo que a veces supone que no funcione tan rápido como otras aplicaciones de uso diario. Sin embargo, esto ha permitido a Zend lanzar con relativa facilidad y rapidez versiones del producto para Windows, Linux y MacOS, aunque el desarrollo de las versiones de este último sistema se retrase un poco más.

Zend Studio consta de dos partes en las que se dividen las funcionalidades del cliente y las del servidor. Ambas se instalan por separado, la del cliente contiene la interfaz de edición y la ayuda. Permite además hacer depuraciones simples de scripts, aunque para disfrutar de toda la potencia de la herramienta de depuración habrá que disponer de la parte del servidor, que instala Apache y el módulo PHP o, en caso de que estén instalados, los configura para trabajar juntos en depuración.

1.4.9 SISTEMAS DE GESTIÓN DE BASES DE DATOS

Un Sistema de Gestión de Bases de Datos (SGBD) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a esos datos. “El Objetivo primordial de un SGBD es proporcionar un entorno que sea a la vez conveniente y eficiente para ser utilizado al extraer y almacenar información de la base de datos”. (KORTH, 1995).

Son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan, su propósito general es manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para un buen manejo de los datos.

1.4.9.1 FUNCIONES DE UN SGBD

Un SGBD debe proporcionar:

- ✓ A los usuarios la capacidad de almacenar datos en la base de datos, acceder a ellos y actualizarlos.
- ✓ Un catálogo en el que se almacenen las descripciones de los datos y que sea accesible por los usuarios.



- ✓ Un mecanismo que garantice que todas las actualizaciones correspondientes a una determinada transacción se realicen, o que no se realice ninguna.
- ✓ Que asegure que la base de datos se actualice correctamente cuando varios usuarios la están modificando de forma concurrente.
- ✓ Capaz de recuperar la base de datos en caso de que ocurra algún suceso que la dañe.
- ✓ Que garantice que sólo los usuarios autorizados pueden acceder a la base de datos.
- ✓ Los medios necesarios para garantizar que tanto la información de la base de datos, como los cambios que se realizan sobre la misma, sigan ciertas reglas.
- ✓ Que se mantenga la independencia entre los programas y la estructura de la base de datos.
- ✓ Una serie de herramientas que permitan administrar la base de datos de modo efectivo.

1.4.9.2 MySQL

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. Fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.

Este gestor es, probablemente, el más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

1.4.9.2.1 VENTAJAS

- ✓ Aprovecha la potencia de sistemas multiprocesador gracias a su implementación multihilo.
- ✓ Soporta gran cantidad de tipos de datos para las columnas.
- ✓ Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP).
- ✓ Gran portabilidad entre sistemas.
- ✓ Soporta hasta 32 índices por tabla.
- ✓ Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.

1.4.9.2.2 ¿QUÉ LE FALTA?

Subconsultas: Tal vez ésta sea una de las características que más se echan en falta, aunque gran parte de las veces que se necesitan, es posible reescribirlas de manera que no sean necesarias.

SELECT INTO TABLE: Esta característica propia de Oracle todavía no está implementada.



Triggers y Procedures: Se tiene pensado incluir el uso de procedures almacenados en la base de datos, pero no el de triggers, ya que los triggers reducen de forma significativa el rendimiento de la base de datos, incluso en aquellas consultas que no los activan.

Transacciones: A partir de las últimas versiones ya hay soporte para transacciones, aunque no por defecto (se ha de activar un modo especial).

Integridad referencial: Aunque sí que admite la declaración de claves ajenas en la creación de tablas, internamente no las trata de forma diferente al resto de campos.

1.4.9.2.3 DESVENTAJAS

- ✓ Carece de soporte para transacciones, rollback's y subconsultas.
- ✓ El hecho de que no maneje la integridad referencial, hace de este gestor una solución pobre para muchos campos de aplicación, sobre todo para aquellos programadores que provienen de otros gestores que sí que poseen esta característica.
- ✓ No es viable para su uso con grandes bases de datos a las que se acceda continuamente, ya que no implementa una buena escalabilidad.

1.4.9.3 POSTGRESQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional (ORDBMS) basado en el proyecto POSTGRES de la Universidad de Berkeley. Es una derivación libre (OpenSource) de este proyecto y utiliza el lenguaje SQL92/SQL99. Fue el pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual, incluido más tarde en otros sistemas de gestión comerciales. Incluye características de la orientación a objetos, como puede ser: la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, no es un sistema de gestión de bases de datos puramente orientado a objetos.

1.4.9.3.1 VENTAJAS

- ✓ Implementación del estándar SQL92/SQL99.
- ✓ Soporta distintos tipos de datos, además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP), cadenas de bits, etc. También permite la creación de tipos propios.
- ✓ Incorpora una estructura de datos array.
- ✓ Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes.



- ✓ Permite la declaración de funciones propias, así como la definición de disparadores entre ellas:
 - Vistas.
 - Integridad transaccional.
 - Herencia de tablas.
 - Tipos de datos y operaciones geométricas.
- ✓ Soporta el uso de índices, reglas y vistas.
- ✓ Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que se le incluye entre los gestores objeto-relacionales.
- ✓ Permite la gestión de diferentes usuarios, así como los permisos asignados a cada uno de ellos.

1.4.9.3.2 DESVENTAJAS

- ✓ Consume gran cantidad de recursos.
- ✓ Tiene un límite de 8K por fila, aunque se puede aumentar a 32K, con una disminución considerable del rendimiento.
- ✓ Es de 2 a 3 veces más lento que MySQL.

1.4.9.4 SELECCIÓN DEL SISTEMA GESTOR DE BASE DE DATOS

Se define PostgreSQL como Sistema Gestor de Base de Datos por sus características en la orientación a objetos, sus tipos de datos, sus funciones, restricciones, disparadores e integridad transaccional. Porque soporta el uso de índices, reglas y vistas. Además permite la gestión de diferentes usuarios, así como los permisos asignados a cada uno de ellos, a pesar de consumir más recursos y ser más lento que el MySQL su condición de ser libre lo hace más sustentable.

1.4.10 METODOLOGÍAS DE DESARROLLO

Se entiende por metodología de desarrollo una colección de documentación formal referente a los procesos, las políticas y los procedimientos que intervienen en el desarrollo del software. La finalidad de una metodología de desarrollo es garantizar la eficacia logrando con ello el cumplimiento de los requisitos iniciales y minimizar las pérdidas de tiempo en el proceso de generación de software. **[10]**



1.4.10.1 PROCESO UNIFICADO DE DESARROLLO (RUP)

El Proceso Unificado de Desarrollo (RUP) es una metodología para la ingeniería de software que va más allá del mero análisis y diseño orientado a objetos, para proporcionar una familia de técnicas que soportan el ciclo completo de desarrollo de software. El resultado es un proceso basado en componentes, dirigido por casos de uso, iterativo e incremental y centrado en la arquitectura.

Es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software, sin embargo el RUP es más que un simple proceso, es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, en diferentes áreas de aplicación, tipos de organizaciones, niveles de aptitud y tamaños de proyecto.

El Proceso Unificado se basa en componentes, lo que significa que el sistema en construcción está hecho de componentes de software interconectados por medio de interfaces bien definidas. Usa el Lenguaje Unificado de Modelado (UML) en la preparación de todos los planos del sistema.

Entre sus principales características se pueden encontrar:

- ✓ Unifica los mejores elementos de metodologías anteriores.
- ✓ Preparado para desarrollar grandes y complejos proyectos.
- ✓ Orientado a Objetos.
- ✓ Utiliza el UML como lenguaje de representación visual.

Como es un proceso en su modelación describe entre sus principales elementos a trabajadores, artefactos, actividades y flujo de actividades o sea quién, qué, cómo y cuándo.

1.4.10.1.1 EL CICLO DE VIDA DE RUP SE CARACTERIZA POR

Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.

Centrado en la arquitectura: La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. La arquitectura se representa a través de vistas en las que se incluyen los diagramas de UML.



Iterativo e Incremental: RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desenvuelve fundamentalmente algunos más que otros.

1.4.10.2 PROGRAMACIÓN EXTREMA (XP)

La Programación Extrema, o eXtreme Programming, es otra de las metodologías de desarrollo de software que existen en la actualidad. Intenta como metodología ágil reducir la complejidad del software por medio de un trabajo orientado directamente al objetivo, basado en las relaciones interpersonales y la velocidad de reacción.

XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.

El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega (Release), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.

Lo fundamental en este tipo de metodología es:

- ✓ La comunicación, entre los usuarios y los desarrolladores.
- ✓ La simplicidad, al desarrollar y codificar los módulos del sistema.
- ✓ La retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.

(SÁCHEZ, 2004)



1.4.10.3 SELECCIÓN DE LA METODOLOGÍA A UTILIZAR

Para desarrollar la propuesta que presenta este trabajo, se ha decidido utilizar como metodología el Proceso Unificado de Desarrollo (RUP), porque utiliza UML como lenguaje de modelado.

Representa una propuesta de proceso cualitativamente superior a XP puesto que combina lo mejor de esta y algunas otras metodologías para garantizar la elaboración de todas las fases de desarrollo de un producto de software orientado a objeto. Finalmente agregar que debido a la estandarización de RUP como guía en el proceso de desarrollo de software, práctica a la que se suma nuestro país, se decide considerarlo como la metodología a utilizar, pues la experiencia ha demostrado que combinar UML con RUP maximiza los resultados a obtener.

1.4.11 LENGUAJE UNIFICADO DE MODELADO (UML)

UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software [11]. Entrega una forma de modelar cosas conceptuales como lo son procesos de negocio y funciones de sistema, además de cosas concretas como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables. Es un sistema notacional destinado a los sistemas de modelado que utilizan conceptos orientados a objetos. Puede usarse para modelar distintos tipos de sistemas: de software, de hardware y organizaciones del mundo real.

1.4.11.1 CARACTERÍSTICAS

Divide cada proyecto en un número de diagramas que representan las distintas vistas del proyecto y juntos representan la arquitectura del mismo. Permite describir un sistema en diferentes niveles de abstracción, simplificando la complejidad sin perder información, para que los usuarios y desarrolladores comprendan las características de la aplicación.

Ofrece nueve diagramas para modelar sistemas:

- ✓ De Casos de Uso para modelar los procesos de negocio (“business” en ingles).
- ✓ De Secuencia para modelar el paso de mensajes entre objetos.
- ✓ De Colaboración para modelar interacciones entre objetos.
- ✓ De Estado para modelar el comportamiento de los casos de uso, objetos y operaciones.
- ✓ De Actividad para modelar el comportamiento de los casos de uso, objetos u operaciones.
- ✓ De Clases para modelar la estructura estática de las clases en el sistema.
- ✓ De Objetos para modelar la estructura estática de los objetos en el sistema.
- ✓ De Componentes para modelar componentes.



- ✓ De Implementación para modelar la distribución del sistema.

1.4.11.2 VENTAJAS

Hoy en día, UML está consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Mientras más complejo es el sistema que se desea crear más beneficios presenta el uso de UML, las razones de esto son evidentes:

- ✓ Diseño y documentación.
- ✓ Código reutilizable.
- ✓ Descubrimiento de fallas.
- ✓ Ahorro de tiempo en el desarrollo del software.
- ✓ Mucho más fáciles las modificaciones.
- ✓ Más fácil comunicación entre programadores.

UML es un lenguaje expresivo, claro y uniforme, que no garantiza el éxito de los proyectos, pero si mejora sustancialmente el desarrollo de los mismos, al permitir una nueva y fuerte integración entre las herramientas, los procesos y los dominios.

1.4.12 HERRAMIENTAS CASE

CASE es una sigla, que corresponde a las iniciales Computer Aided Software Engineering; y en su traducción al español significa Ingeniería de Software Asistida por Computación. El concepto de CASE es muy amplio; y una buena definición genérica sería la de considerar a la Ingeniería de Software Asistida por Computación (CASE), [12] como la aplicación de métodos y técnicas a través de las cuales se les hace útil a las personas comprender las capacidades de las computadoras por medio de programas, de procedimientos y su respectiva documentación. Las herramientas CASE representan una forma que permite modelar los procesos del negocio de las empresas y desarrollar los Sistemas de Información Gerenciales.

Debido a la gran demanda que tienen las CASE la exigencia en cuanto a su uso ha ido aumentando, por lo que toda CASE debe entre otras cosas:

- ✓ Proporcionar topologías de aplicaciones flexibles y portátiles.
- ✓ Brindar un control de versiones.
- ✓ Crear código compilado en el servidor.
- ✓ Dar un soporte multiusuario.
- ✓ Ofrecer seguridad.



1.4.12.1 RATIONAL ROSE ENTERPRISE EDITION

Es una de las más poderosas herramientas de modelado visual para el análisis y diseño de sistemas basados en objetos. Se utiliza para modelar un sistema antes de proceder a construirlo. Cubre todo el ciclo de vida de un proyecto:

- ✓ Concepción y formalización del modelo.
- ✓ Construcción de los componentes.
- ✓ Transición a los usuarios.
- ✓ Certificación de las distintas fases.

Es la herramienta CASE que comercializan los desarrolladores de UML, la cual permite el completamiento de una gran parte de los flujos fundamentales del RUP como son:

- ✓ Modelado del negocio.
- ✓ Captura de requisitos.
- ✓ Análisis y Diseño.
- ✓ Implementación.
- ✓ Gestión de Configuración y Control de Cambios.

1.4.12.1.1 CARACTERÍSTICAS

- ✓ Permite especificar, analizar, diseñar el sistema antes de codificarlo.
- ✓ Mantiene la consistencia de los modelos del sistema de software.
- ✓ Chequeo de la sintaxis UML.
- ✓ Generación y documentación automática.
- ✓ Generación de código a partir de los modelos.
- ✓ Ingeniería Inversa.
- ✓ Esta herramienta contiene 4 vistas:
 - Vista de Caso de Uso.
 - Vista Lógica.
 - Vista de Componente.
 - Vista de Despliegue.



1.4.12.2 VISUAL PARADIGM

El Visual Paradigm es una herramienta CASE que utiliza UML como lenguaje de modelado, con el uso del acercamiento orientado al objeto. Esta herramienta apoya los estándares más altos de las notaciones de Java y de UML. Está dotada de una buena cantidad de productos o módulos para facilitar el trabajo durante la confección de un software, lo cual garantiza la calidad del producto final.

1.4.12.2.1 CARACTERÍSTICAS

- ✓ Entorno de creación de diagramas para UML.
- ✓ Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- ✓ Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- ✓ Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- ✓ Disponibilidad en múltiples plataformas.
- ✓ Es amigable.
- ✓ Contiene facilidades para redactar Especificaciones de Casos de Uso del Sistema.
- ✓ Sincronización entre Diagramas de Entidad Relación y Diagramas de Clases.
- ✓ Generación de documentos.
- ✓ Interoperabilidad con otras Aplicaciones.
- ✓ Integración con distintos Ambientes de Desarrollo Integrados (IDE).

1.4.12.3 SELECCIÓN DE LA HERRAMIENTA CASE

Se concreta la utilización de Visual Paradigm como herramienta case por su integración a UML además de portar con las características de ser multiplataforma y no propietario, amigable en su uso y poseer interoperabilidad con otras aplicaciones e integración con distintos Ambientes de Desarrollo Integrado (IDE).

1.4.13 OTRAS HERRAMIENTAS DE APOYO PARA EL DESARROLLO

1.4.13.1 MACROMEDIA DREAMWEAVER 8.0

Macromedia Dreamweaver ha sido históricamente y es una de las herramientas más poderosas y flexibles para el diseño de sitios Web, posibilita mayor facilidad para trabajar con los aspectos del diseño de las páginas.



1.4.13.2 MACROMEDIA FIREWORKS 8.0

Macromedia Fireworks es una aplicación en forma de estudio pero con más parecido a un taller destinado para el manejo híbrido de gráficos vectoriales con gráficos en mapa de bits y ofrece un ambiente eficiente para la creación rápida de prototipos de sitios Web e interfaces de usuario como para la creación y optimización de imágenes. Fireworks está enfocado en la creación y edición de gráficos para la Internet, diseñado para integrarse con otros productos de Adobe, como Dreamweaver y Flash.

1.4.13.3 MACROMEDIA FLASH 8.0

Esta herramienta es la líder mundial en la confección de animaciones para banners. Es importante decir que la mayoría de los sitios Web que existen en el mundo utilizan la tecnología flash para promocionar o divulgar información y realizar anuncios más elegantes, así como para hacer presentaciones corporativas o de instituciones.

1.5 PROPUESTA DE SOLUCIÓN

Como propuesta de solución y partiendo del análisis previo realizado sobre las diferentes herramientas y tecnologías que se pueden utilizar para el desarrollo de las aplicaciones Web y como premisa de construir un software con políticas de migración al Software Libre, se define el uso de PostgreSQL como gestor de base de datos por sus características anteriormente mencionadas; Visual Paradigm como herramienta CASE y RUP como metodología de desarrollo, teniendo en cuenta las ventajas y características expuestas con anterioridad. Se precisa UML como lenguaje de modelado, también se puntualiza el uso de PHP como lenguaje de programación partiendo de las características que posee como lenguaje, aludidas precedentemente, integrado al ZendStudio y haciendo uso del framework Symfony portador de características que ayudan al desarrollador, brindando la posibilidad de que este, centre su atención en las funcionalidades esenciales de la aplicación. Se incluye además el uso de diferentes herramientas de apoyo como Adobe Fireworks, Macromedia Dreamweaver y Macromedia Flash 8.0 para el tratamiento de imágenes, el diseño de las páginas HTML y la creación de banners respectivamente, que ayudarán al mejoramiento de la interfaz de la aplicación presentada al usuario. También se define el uso de XHTML como lenguaje de marcado para lograr una mejor funcionalidad y diseño de la aplicación que se desea realizar.



1.6 CONCLUSIONES

En la actualidad existe una gran variedad de tecnologías de desarrollo Web, las cuales poseen en particular una serie de características que las definen como más o menos apropiadas para llevar a cabo un determinado proyecto. Utilizar adecuadamente sus características depende en gran medida de los requisitos que debe cumplir la aplicación a desarrollar, de los objetivos que persigue y la posibilidad de brindar los mejores servicios a los clientes que interactúen con ella.

Entonces se puede concluir que, si con el estudio realizado, aún no existe un software que sea capaz de gestionar la información que existe entre frameworks de Java y los ficheros JAR que estos utilizan, con la confección de la aplicación propuesta se podría lograr tales objetivos incluyendo también un beneficio relevante para los desarrolladores de software en Cuba.



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1 INTRODUCCIÓN

En el presente capítulo se define la propuesta de solución y se describen los procesos básicos que influyen en el objeto de estudio y hacen válido el problema a resolver anteriormente puntualizado. Para poseer un mejor conocimiento y concebir el funcionamiento del sistema, teniendo como premisa la insuficiente estructuración de los procesos, se hace necesaria la definición de diferentes conceptos que se agrupan en un Modelo de Dominio. Además se enumeran los requisitos funcionales y no funcionales, características y cualidades que el sistema debe cumplir y tener respectivamente, permitiendo una noción general del mismo e identificar mediante Diagramas de Casos de Uso las relaciones existentes entre los actores que se involucran con el sistema y las secuencias de acciones con las que interactúa teniendo los casos de uso identificados como de base.

2.2 MODELO DE DOMINIO

Cuando se hace uso de la ingeniería de software y se alude al empleo de alguna metodología de desarrollo, en este caso RUP, en lo primero que se piensa es en realizar un Modelo de Negocio, pero para ello se hace necesario la identificación de los procesos del mismo, siendo este un conjunto de actividades que responden directamente a un proceso en sí.

En el estudio realizado del problema que se plantea, no fue posible identificar claramente los procesos que se llevan a cabo en el negocio, dado el escaso balance de la estructura de estos, por tanto se decidió la realización de un Modelo de Dominio.

El Modelo de Dominio ayuda a la interpretación precisa del sistema, posibilita obtener correctamente los requerimientos y entender el objeto de estudio para darle una solución adecuada al problema a resolver. Hay que tener presente además que el este modelo se realiza cuando no se puede delimitar bien un negocio, pero a la vez la presencia del mismo no exime la representación de un negocio cualquiera, es decir, se puede realizar Modelo de Dominio y Modelo de Negocio en conjunto. En esencia y en el caso que se observa, el Modelo de Dominio tiene como objetivo entender y describir los objetos más importantes dentro del contexto del sistema y las relaciones entre ellos que, en lo concreto, sería la comprensión básica del problema a resolver en relación a su contexto.



2.2.1 GLOSARIO DE TÉRMINOS

El glosario de términos se realizó con los principales conceptos que se desdoblaron en el contexto del problema y que se utilizarán una vez definidos, en la confección del Modelo de Dominio. Los mismos tienen como objetivo principal la obtención de un lenguaje común entre desarrolladores, clientes y usuarios finales.

Administrador: Persona que se encargará de velar por la seguridad de los datos de la aplicación así como trabajar directamente con la misma lo que se concreta en la actualización, modificación de la información.

Búsqueda: Acción que se lleva a cabo con el propósito de obtener como resultado la información que sea de interés al usuario esta puede ser efectuada siguiendo un criterio de búsqueda especificado.

Criterio de Búsqueda: Lo constituyen especificaciones que se realizan antes de acometer la búsqueda en concreto como lo son: nombre de frameworks, nombre de fichero JAR, tamaño del fichero JAR, entre otros.

Dirección de Producción UCI: Conjunto de personas facultadas para la toma de decisiones en la UCI.

Dirección de Tecnología: Lo constituye un grupo de personas capaces de tomar las decisiones relacionado a las tecnologías a utilizar en el centro universitario.

Equipo de trabajo: Grupo de estudiantes y profesores que forman parte de un proyecto productivo y que constituyen un eslabón fundamental en el mismo.

Estudiantes: Personal que realiza estudios en la educación superior.

Jefe de Proyecto: Persona encargada del cumplimiento de todo lo referente a un proyecto productivo.

Jefe de Polo Productivo: Persona facultada para exigir el cumplimiento de todo lo referente al polo productivo.

Laboratorio: Local designado para la docencia o la producción que posee la tecnología adecuada para el avance académico y/o productivo.

Producto: Resultados de los proyectos informáticos (software).

Personal: Estudiantes, profesores y trabajadores que forman parte de un proyecto productivo.

Polo Productivo: Espacio natural para ejecutar proyectos temáticos a partir de la integración de los procesos fundamentales de formación, investigación, producción y comercialización.

Proyecto Productivo: Actividad que involucra a un grupo seleccionado de estudiantes y profesores con un objetivo único el cual será la obtención de un producto de software final.

Profesores: Personal que contribuyen a la formación de los estudiantes en una institución determinada.



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Vicerrector de Producción: Persona capacitada para exigir el cumplimiento de todo lo relacionado con producción en la UCI.

Vicerrector de Tecnología: Persona delegada para controlar todo lo pertinente a las tecnologías en la Universidad.

UCI: Universidad de Las Ciencias Informáticas, centro universitario que contribuye a la formación de intelectuales de la informática vinculando el estudio con la producción y que por ende funciona además como centro de producción de software.

Usuario: Estudiantes y profesores de la UCI que utilizan la aplicación para lograr obtener los resultados esperados.

2.2.2 DIAGRAMA DE CLASES DEL DOMINIO

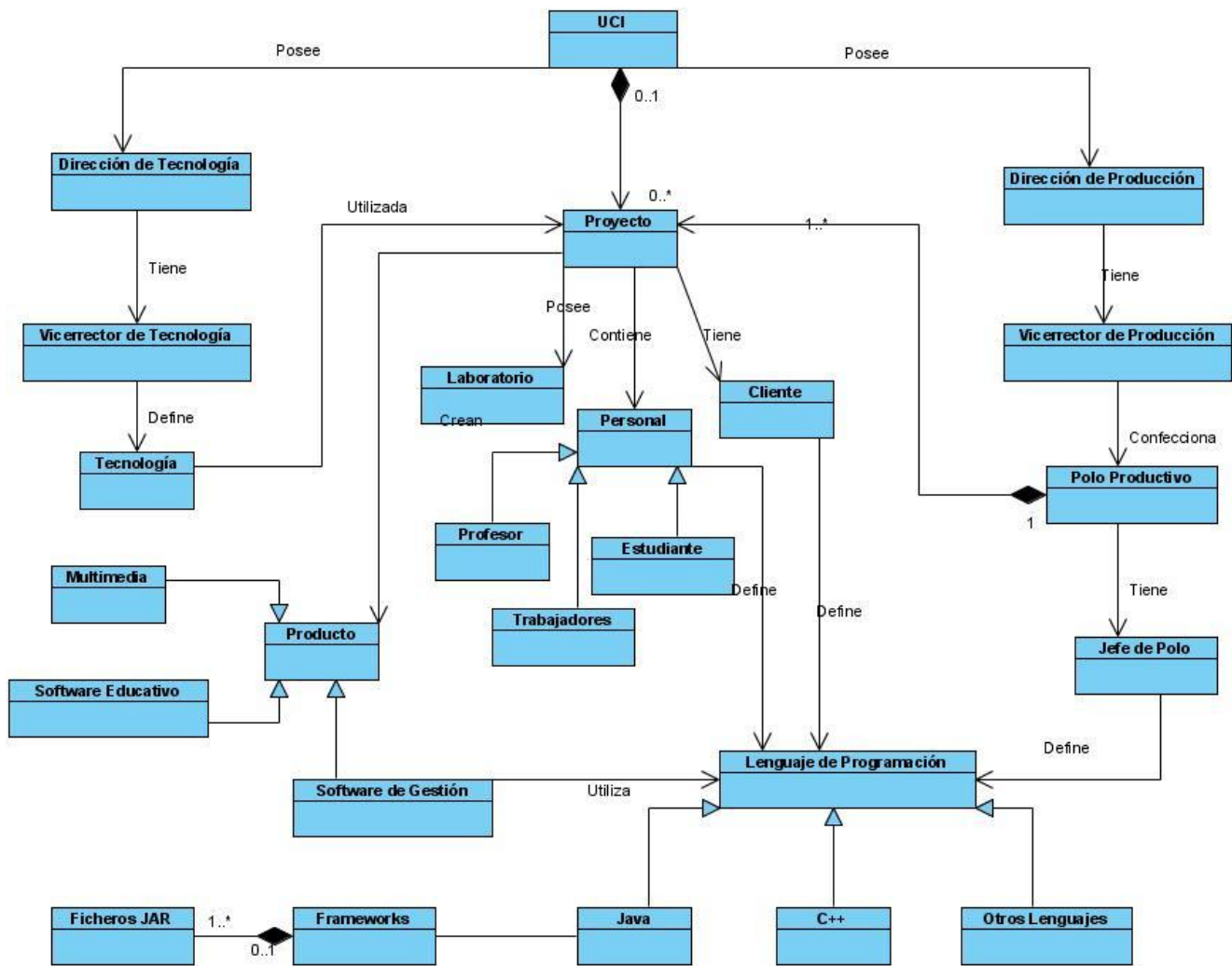


Figura 1: Diagrama de Clases del Dominio



2.3 ESPECIFICACIÓN DE LOS REQUERIMIENTOS DEL SOFTWARE

Los requerimientos del software son las características y cualidades que el sistema debe tener. Estos se agrupan en dos grupos, los funcionales y los no funcionales. Una vez definidos e identificados los mismos, se puede transitar al modelamiento del sistema. Además es posible afirmar que es uno de los flujos más relevantes en la fase de inicio definida por RUP, ya que aporta la visión general de lo que se quiere hacer en el sistema.

Listado de los Requisitos Funcionales (RF)

Los requisitos funcionales se definen como las condiciones o capacidades que el sistema debe cumplir.

Se desea que el sistema permita:

RF1: Autenticación de usuarios.

RF2: Gestionar Frameworks.

RF2.1: Adicionar frameworks.

RF2.2: Eliminar frameworks.

RF2.3: Modificar frameworks.

RF2.4: Ver frameworks.

RF3: Gestionar ficheros JAR.

RF3.1: Adicionar ficheros JAR.

RF3.2: Eliminar ficheros JAR.

RF3.3: Actualizar ficheros JAR.

RF3.4: Ver ficheros JAR

RF4: Asignar un fichero JAR a un framework.

RF5: Consultar frameworks

RF6: Consultar ficheros JAR

RF7: Descargar framework.

RF8: Descargar ficheros JAR.



Listado de los Requisitos no Funcionales (RNF)

Los requisitos no funcionales se definen como las cualidades o propiedades que el producto o sea el software debe tener, estos representan las características del producto.

Usabilidad

RNF1: El sistema será ampliamente usado por todos sus clientes.

RNF2: Debe consentir el manejo de cualquier usuario posibilitando confianza y seguridad en cuanto a su uso.

Interfaz

RNF3: La aplicación debe ser amigable y muy profesional permitiendo una interacción amena.

RNF4: El diseño de la interfaz de usuario llamar la atención al mismo.

Rendimiento

RNF5: Debe ser rápida, eficiente tanto en los tiempos de respuesta como en la velocidad de procesamiento de 0.25 segundos y 0.45 segundos respectivamente.

Confidencialidad

RNF6: La información debe estar protegida contra accesos no autorizados.

RNF7: La información debe ser publicada por una persona con los privilegios asignados, o sea el administrador.

Soporte

RNF8: La aplicación debe ser multiplataforma.

RNF9: Debe ofrecer la posibilidad de un fácil mantenimiento.

RNF10: Se hallará implementado sobre tecnología Web lo que proporcionará su uso mediante la red.

Funcionalidad

RNF11: Reducir al mínimo el tiempo de carga de la aplicación.

Seguridad

RNF12: Tener protección contra acciones no autorizadas y que puedan afectar la integridad, confidencialidad y seguridad de los datos.



RNF13: Asegurar que el usuario sea autenticado antes de realizar cualquier operación sobre la aplicación.

RNF14: El sistema deberá estar disponible brindando una posibilidad de uso de 24 horas.

RNF15: El sistema solo puede ser modificado o actualizado por el administrador del mismo.

Software

La aplicación debe correr en una estación de trabajo que conste con los diferentes software que se exponen a continuación:

RNF16: Navegador compatible con Mozilla Firefox.

RNF17: Macromedia Dreamweaver 8.

RNF18: Macromedia Flash 8.

RNF19: Macromedia Fireworks MX.

RNF20: PostgreSQL 8.2.

RNF21: Apache 2.2.

RNF22: PHP 5.0.

2.4 DESCRIPCIÓN DEL SISTEMA PROPUESTO

Para lograr satisfactoriamente cumplimentar los objetivos propuestos de este trabajo y además contar con los requerimientos expuestos con anterioridad, los cuales responden directamente a dichos objetivos porque constituyen las características necesarias para lograr las funcionalidades básicas que debe brindar el sistema, se propone el desarrollo de una Aplicación Web para gestionar la relación referente a los frameworks de JAVA y a sus ficheros JAR.

Se considera la existencia de dos roles, el rol de usuario anónimo y el rol de administrador del sistema, este último se encargará de la gestión de la información tanto de frameworks como ficheros JAR lo que engloba adicionar, eliminar y modificar los elementos especificados, así como la asignación de un fichero JAR a un framework determinado.

El rol de usuario anónimo sólo obtendrá el beneficio de la información que se maneja en la aplicación y la posibilidad de descargar tanto frameworks como ficheros JAR como estime pertinente, el mismo puede ser utilizado por cualquier estudiante y profesor perteneciente a la UCI que necesite de la información referente a los frameworks de JAVA y a las ficheros JAR.



2.4.1 DEFINICIÓN DE LOS CASOS DE USO

Los casos de uso son fragmentos de funcionalidad que el sistema ofrece y que aportan un resultado para sus actores, los cuales son, aquellas personas clasificados como terceros que se encuentran fuera del sistema y que trabajan con él. Un caso de uso puede ser inicializado por un solo actor, no siendo así la relación con los mismos, estos especifican una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de la misma.

Es posible afirmar que los casos de usos, en concreto, son los requerimientos funcionales del sistema aludidos con anterioridad, agrupados según su funcionalidad, independientes o relacionados entre sí formando un conjunto de actividades que llevan implícito funcionalidades básicas propiamente del sistema.

Para la especificación de un caso de uso se parte necesariamente de los requisitos funcionales y se siguen tres pasos fundamentales que dan lugar a la correcta identificación de los mismos:

1. Se define un caso de uso por requisito funcional.
2. Se agrupan los casos de usos fuertemente relacionados.
3. Se dividen los casos de usos sobrecargados o complejos.

Una vez que se tiene la conceptualización de los casos de uso se definen a continuación los actores que intervienen en el sistema.

2.4.2 DEFINICIÓN DE LOS ACTORES

Actores del sistema

Actores	Justificación
Administrador	Persona que se encargará de la configuración de la aplicación así como de la gestión de eventos y la administración de forma general de la misma.
Usuario Anónimo	Persona que utiliza la aplicación para la obtención de información referente a la relación existente entre frameworks de JAVA y ficheros JAR, así como posee la posibilidad de descargar, o sea, obtener físicamente los elementos definidos.

Tabla 1: Actores del sistema

2.4.3 LISTADO DE LOS CASOS DE USO DEL SISTEMA

A continuación se muestra un listado de los casos de uso y su relación con el actor del sistema que lo inicia.



CU1 Autenticar Usuario	
Actor	Administrador
Descripción	El administrador se autentica en la aplicación.
Referencia	RF1

Tabla 2: Caso de uso Autenticar usuario

CU2 Gestionar frameworks	
Actor	Administrador
Descripción	El administrador solicita adicionar, ver, modificar o eliminar frameworks.
Referencia	RF2(2.1, 2.2, 2.3, 2.4)

Tabla 3: Caso de uso Gestionar Frameworks

CU3 Gestionar ficheros JAR	
Actor	Administrador
Descripción	El administrador requiere adicionar, ver, modificar o eliminar ficheros JAR.
Referencia	RF3(3.1, 3.2, 3.3, 3.4) CU6

Tabla 4: Caso de uso Gestionar Ficheros JAR

CU4 Asignar un fichero JAR a un framework	
Actor	Administrador
Descripción	El administrador asigna un fichero JAR a un framework determinado.
Referencia	RF4

Tabla 5: Caso de uso Asignar Fichero JAR a un Framework



CU5 Consultar ficheros JAR	
Actor	Usuario Anónimo
Descripción	El usuario anónimo solicita consultar los ficheros JAR que se encuentran en la base de datos y puede además solicitar una información especificando algún criterio.
Referencia	RF6

Tabla 6: Caso de uso Consultar Ficheros JAR

CU6 Consultar frameworks	
Actor	Usuario Anónimo
Descripción	El usuario anónimo solicita consultar los frameworks que se encuentran en la base de datos y puede además solicitar una información especificando algún criterio.
Referencia	RF5

Tabla 7: Caso de uso Consultar Frameworks

CU7 Descargar fichero JAR	
Actor	CU_Consultar ficheros JAR
Descripción	El usuario anónimo requiere descargar un fichero JAR.
Referencia	RF8, CU5

Tabla 8: Caso de uso Descargar Ficheros JAR

CU8 Descargar framework	
Actor	CU_Consultar frameworks
Descripción	El usuario anónimo solicita descargar un framework.
Referencia	RF7, CU6

Tabla 9: Caso de uso Descargar frameworks



2.4.4 DIAGRAMA DE CASOS DE USO DEL SISTEMA

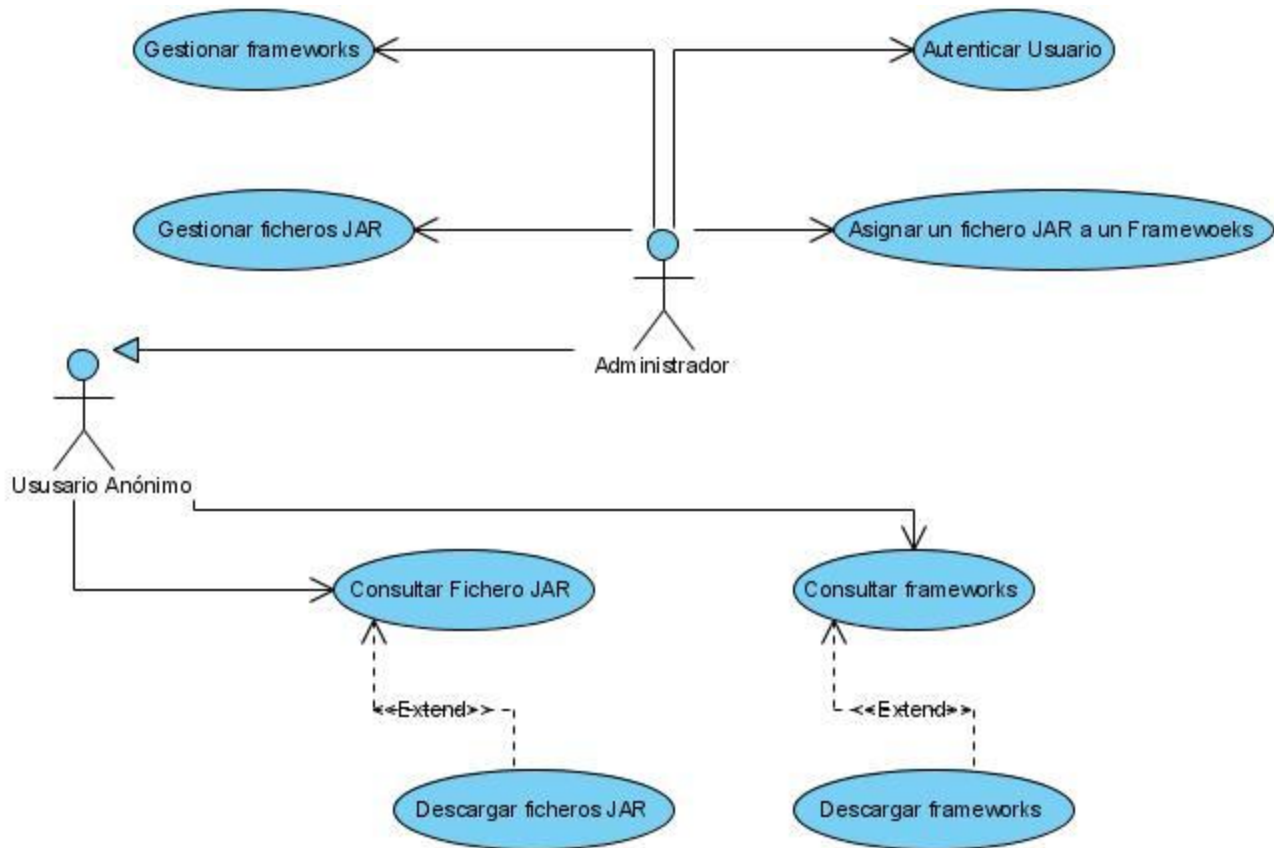


Figura 2: Diagrama de casos de uso del Sistema



2.4.5 DESCRIPCIÓN TEXTUAL DE LOS CASOS DE USO

En el siguiente epígrafe se describen detalladamente los casos de uso presentes en el Diagrama de Casos de Uso del Sistema. Se situará el propósito general de cada caso de uso, el actor que le da inicio al mismo, un pequeño resumen de su funcionamiento así como las precondiciones y las poscondiciones, o sea, las condiciones necesarias para que el caso de uso en sí pueda lograr el funcionamiento para el cual fue creado y el estado en que queda el sistema luego de su inicialización y ejecución total, respectivamente.

CU-1	Autenticar Usuario	
Actor	Administrador	
Propósito	Autenticarse en el sistema	
Resumen	El caso de uso inicia cuando el administrador accede a la aplicación para autenticarse introduciendo sus datos personales como usuario y contraseña y luego se comprueba la existencia o no del usuario registrado.	
Referencia	RF1	
Precondiciones	El usuario que accede debe estar registrado en el sistema como administrador.	
Poscondiciones	EL usuario se autenticó en el sistema y se habilitaron las funcionalidades según sus privilegios.	
Curso Normal de los Eventos		
Acción del Actor	Respuesta del Sistema	
1. Selecciona la opción de Autenticar.	2. El sistema brinda la posibilidad de insertar los siguientes datos : ❖ Usuario ❖ Contraseña.	
3. Introduce (Usuario, Contraseña).	2. Busca el usuario y comprueba contraseña.	
	4 Se asignan los permisos correspondientes.	



Flujo Alterno	
3.a* Introduce Usuario y/o contraseña incorrectos	
	3.a.1 Muestra un mensaje de error.
Nivel	Usuario
Prioridad	Crítico

Tabla 10: Descripción textual del CU _ Autenticar Usuario

CU-2	Gestionar frameworks
Actor	Administrador
Propósito	Adicionar, Modificar, Ver y Eliminar datos referentes a los frameworks.
Resumen	El caso de uso inicia cuando el usuario accede a la aplicación y selecciona la opción de Gestionar frameworks, luego se le da la posibilidad de escoger cuatro opciones válidas: Adicionar, Ver, Eliminar o Modificar frameworks.
Referencia	RF2(2.1, 2.2, 2.3, 2.4)
Precondiciones	El usuario que accede debe estar autenticado en el sistema como administrador. Antes de Ver, Eliminar o modificar se debe Consultar frameworks.



Poscondiciones	Se adicionó, visualizó, eliminó o modificó algún framework.
Curso Normal de los Eventos	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción: Gestionar frameworks.	2. Brinda cuatro opciones posibles para seleccionar:
	2.1 Adicionar frameworks. 2.2 Eliminar frameworks. (Ver Sección1) 2.3 Modificar frameworks. (Ver Sección2) 2.4 Ver frameworks. (Ver Sección3)
Sección: Adicionar frameworks	
3. Selecciona la opción: Adicionar frameworks.	4. Muestra una interfaz para que el usuario inserte los datos correspondientes al framework que adicionará, especificando: <ul style="list-style-type: none"> • Nombre del framework. • Versión. • Grupo de ficheros JAR asociados. • Grupo de ficheros JAR de terceros asociados.
5. Introduce los datos del framework y presiona el botón adicionar.	6. Valida los datos.
	7. Muestra un mensaje informando si desea culminar la operación. ¿Desea terminar el proceso?



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

8. Selecciona la opción: Aceptar.	9. Adiciona el framework con sus datos correspondiente.
	10. Muestra un mensaje de verificación y/o información al actor: “Se ha adicionado con éxito el framework deseado”
	11. Culmina la funcionalidad especificada.
Flujo Alterno	
5.a* Introduce los datos del framework incompletos.	
	5.a.1 Muestra un mensaje de error: “No puede existir ningún campo de los señalados en blanco o vacío”
	5.a.2 Muestra una señalización sobre los campos que debe llenar.
	5.a.3 Continúa con el paso 11 del curso normal de los eventos.
5.b* introduce los datos del framework más de una vez.	
	5.b.1 Muestra un mensaje de error: ” El framework que desea añadir ya se encuentra registrado, solamente puede modificar sus datos”
	5.b.2 Continúa con el paso 11 del curso normal de los eventos.
Sección1: Eliminar frameworks	
Curso Normal de los Eventos	
1 Selecciona la opción de Eliminar framework.	2. Muestra mensaje: “Debe Consultar frameworks primeramente”
	3. Muestra dos opciones: <ul style="list-style-type: none"> ❖ Consultar frameworks ❖ Cancelar



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

4. Selecciona la opción: Consultar Frameworks.	5. Muestra la IU del CU Consultar frameworks.
6. Selecciona un framework determinado.	7. Muestra un mensaje brindado dos opciones posibles para que se realice la eliminación. ¿Seguro que desea eliminar el framework señalado? ❖ Aceptar ❖ Cancelar
8. Selecciona la opción: Aceptar.	9. Elimina el framework.
	10. Muestra un mensaje informando al usuario: “El framework señalado se ha eliminado correctamente”
	11. Culmina la funcionalidad especificada.
Flujo Alternativo	
4.a* Selecciona la opción: Cancelar	
	4.a.1 Regresa a la IU anterior.
	4.a.2 Continúa con el paso 11 del flujo normal de los eventos.
8.a.* Selecciona la opción Cancelar	
	8.a.1 Regresa a la IU anterior.
	8.a.2 Continúa con el paso 11 del flujo normal de los eventos.
Sección 2: Modificar frameworks	
Curso Normal de los Eventos	
1. Selecciona la opción: Modificar frameworks.	2. Muestra mensaje: “Debe Consultar frameworks primeramente”
	3. Muestra dos opciones:



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	<ul style="list-style-type: none"> ❖ Consultar frameworks ❖ Cancelar
3. Selecciona la opción: Consultar frameworks.	4. Muestra una IU del CU Consultar frameworks.
5. Selecciona el framework a modificar.	
6. Selecciona la opción de ver datos del framework. (Ver en la Sección3: Ver frameworks)	<p>7. Muestra los datos del framework seleccionado:</p> <ul style="list-style-type: none"> • Nombre del framework. • Versión. • Grupo de ficheros JAR asociados. Grupo de ficheros JAR de terceros asociados.
8. Modifica los datos necesarios.	<p>9. Muestra opciones para efectuar operación de modificación:</p> <ul style="list-style-type: none"> ❖ Actualizar Campos ❖ Cancelar
10. Selecciona la opción de Actualizar campos del framework.	11. Valida los datos.
	12. Muestra un mensaje al usuario: ¿Se actualizarán los campos que sufrieron cambios, desea continuar?
13. Selecciona la opción Aceptar.	14. Muestra mensaje de información al usuario: “Los campos han sido actualizados satisfactoriamente”.
	<p>15. Muestra los datos del framework modificado.</p> <p>(Ver Sección4: Ver frameworks)</p>
	16. Termina la funcionalidad señalada.
Flujo Alternativo	



3.a* Selecciona la opción: Cancelar	
	3.a.1 Muestra IU anterior.
	3.a.2 Continúa con el paso 16 del flujo normal de los eventos.
10.a.* Selecciona la opción: Cancelar.	
	10.a.1 Elimina los datos.
	10.a.2 Muestra IU anterior.
	10.a.3 Continúa con el paso 16 del flujo normal de los eventos.
Sección4: Ver frameworks	
Curso Normal de los Eventos	
1. Selecciona la opción: Ver frameworks.	2. Muestra un mensaje: “Debe seleccionar Consultar frameworks primeramente”.
	4. Muestra dos posibles opciones: <ul style="list-style-type: none"> ❖ Consultar frameworks ❖ Cancelar
5. Selecciona la opción Consultar frameworks.	6. Muestra la IU del CU Consultar framework
7. Selecciona un framework determinado.	8. Muestra los datos específicos del framework seleccionado: <ul style="list-style-type: none"> • Nombre del framework. • Versión. • Grupo de ficheros JAR asociados. • Grupo de ficheros JAR de terceros asociados.
9. Selecciona la opción de salir de la vista actual.	10. Muestra la IU anterior.



	11. Culmina la funcionalidad especificada.
Flujo Alternativo	
5.a* Selecciona la opción Cancelar.	
	5.a.1 Continúa con el paso 10 del curso normal de los eventos.
Nivel	Usuario
Prioridad	Crítico

Tabla 11: Descripción textual del CU_Gestionar Frameworks

CU-3	Gestionar ficheros JAR	
Actor	Administrador	
Propósito	Adicionar, Modificar, Ver y Eliminar datos referentes a los ficheros JAR.	
Resumen	El caso de uso inicia cuando el usuario accede a la aplicación y selecciona la opción de Gestionar ficheros JAR, luego se le da la posibilidad de escoger cuatro opciones válidas: Adicionar, Ver, Eliminar o Modificar ficheros JAR.	
Referencia	RF3(3.1, 3.2, 3.3, 3.4)	
Precondiciones	El usuario que accede debe estar autenticado en el sistema como administrador. Antes de Ver, Eliminar o modificar se debe Consultar ficheros JAR.	
Poscondiciones	Se adicionó, visualizó, eliminó o modificó algún fichero JAR.	
Curso Normal de los Eventos		
Acción del Actor	Respuesta del Sistema	
1. Selecciona la opción: Gestionar ficheros JAR.	2. Brinda cuatro opciones posibles para seleccionar:	
	2.1 Adicionar ficheros JAR. 2.2 Eliminar ficheros JAR (Ver Sección1) 2.3 Modificar ficheros JAR. (Ver Sección2)	



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	2.4 Ver ficheros JAR. (Ver Sección3)
Sección: Adicionar ficheros JAR	
3. Selecciona la opción: Adicionar fichero JAR.	4. Muestra una interfaz para que el usuario inserte los datos correspondientes al fichero JAR que adicionará, especificando: <ul style="list-style-type: none"> • Nombre del fichero JAR . • Tamaño. • Estado. • Framework al que pertenece.
5. Introduce los datos del fichero JAR y presiona el botón adicionar.	6. Valida los datos.
	7. Muestra un mensaje informando si desea culminar la operación. ¿Desea terminar el proceso?
8. Selecciona la opción: Aceptar.	9. Adiciona el fichero JAR con sus datos correspondiente.
	10. Muestra un mensaje de verificación y/o información al actor: “Se ha adicionado con éxito el fichero JAR deseado”
	11. Culmina la funcionalidad especificada.
Flujo Alterno	
5.a* Introduce los datos del fichero JAR incompletos.	
	5.a.1 Muestra un mensaje de error: “No puede existir ningún campo de los señalados en blanco o vacío”
	5.a.2 Muestra una señalización sobre los campos que debe llenar.
	5.a.3 Continúa con el paso 11 del curso normal de los eventos.



5.b* Introduce los datos del fichero JAR más de una vez.	
	5.b.1 Muestra un mensaje de error: " El ficheros JAR que desea añadir ya se encuentra registrado, solamente puede modificar sus datos"
	5.b.2 Continúa con el paso 11 del curso normal de los eventos.
Sección1: Eliminar ficheros JAR	
Curso Normal de los Eventos	
1 Selecciona la opción de Eliminar ficheros JAR.	2. Muestra mensaje: "Debe Consultar ficheros JAR primeramente"
	3. Muestra dos opciones: <ul style="list-style-type: none"> ❖ Consultar ficheros JAR ❖ Cancelar
4. Selecciona la opción: Consultar ficheros JAR.	5. Muestra la IU del CU Consultar ficheros JAR.
6. Selecciona un fichero JAR determinado.	7. Muestra un mensaje brindado dos opciones posibles para que se realice la eliminación. ¿Seguro que desea eliminar el fichero JAR señalado? <ul style="list-style-type: none"> ❖ Aceptar ❖ Cancelar
8. Selecciona la opción: Aceptar.	9. Elimina el fichero JAR.
	10. Muestra un mensaje informando al usuario: "El fichero JAR señalado se ha eliminado correctamente"
	11. Culmina la funcionalidad especificada.
Flujo Alternativo	



4.a* Selecciona la opción: Cancelar	
	4.a.1 Regresa a la IU anterior.
	4.a.2 Continúa con el paso 11 del flujo normal de los eventos.
8.a.* Selecciona la opción Cancelar	
	8.a.1 Regresa a la IU anterior.
	8.a.2 Continúa con el paso 11 del flujo normal de los eventos.
Sección2: Modificar ficheros JAR	
Curso Normal de los Eventos	
1. Selecciona la opción: Modificar ficheros JAR.	2. Muestra mensaje: “Debe Consultar ficheros JAR primeramente”
	3. Muestra dos opciones: <ul style="list-style-type: none"> ❖ Consultar ficheros JAR ❖ Cancelar
3. Selecciona la opción: Consultar ficheros JAR.	4. Muestra una IU del CU Consultar ficheros JAR.
5. Selecciona el fichero JAR a modificar.	
6. Selecciona la opción de ver datos del fichero JAR. (Ver en la Sección3: Ver frameworks)	7. Muestra los datos del fichero JAR seleccionado: <ul style="list-style-type: none"> • Nombre del fichero JAR. • Tamaño. • Estado. • Framework al que pertenece.
8. Modifica los datos necesarios.	9. Muestra opciones para efectuar operación de modificación: <ul style="list-style-type: none"> ❖ Actualizar Campos



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	❖ Cancelar
10. Selecciona la opción de Actualizar campos del fichero JAR.	11. Valida los datos.
	12. Muestra un mensaje al usuario: ¿Se actualizarán los campos que sufrieron cambios, desea continuar?
13. Selecciona la opción Aceptar.	14. Muestra mensaje de información al usuario: “Los campos han sido actualizados satisfactoriamente”.
	15. Muestra los datos del fichero JAR modificado. (Ver Sección4: Ver ficheros JAR)
	16. Termina la funcionalidad señalada.
Flujo Alternativo	
3.a* Selecciona la opción: Cancelar	
	3.a.1 Muestra IU anterior.
	3.a.2 Continúa con el paso 16 del flujo normal de los eventos.
10.a.* Selecciona la opción: Cancelar.	
	10.a.1 Elimina los datos.
	10.a.2 Muestra IU anterior.
	10.a.3 Continúa con el paso 16 del flujo normal de los eventos.
Sección4: Ver ficheros JAR	
Curso Normal de los Eventos	
1. Selecciona la opción: Ver ficheros JAR.	2. Muestra un mensaje: “Debe seleccionar



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	Consultar ficheros JAR primeramente”.
	5. Muestra dos posibles opciones: ❖ Consultar ficheros JAR ❖ Cancelar
5. Selecciona la opción Consultar ficheros JAR.	6. Muestra la IU del CU Consultar ficheros JAR
7. Selecciona un fichero JAR determinado.	8. Muestra los datos específicos del fichero JAR seleccionado: • Nombre del ficheros JAR. • Tamaño. • Estado. • Frameworks al que pertenece.
9. Selecciona la opción de salir de la vista actual.	10. Muestra la IU anterior.
	11. Culmina la funcionalidad especificada.
Flujo Alternativo	
5.a* Selecciona la opción Cancelar.	
	5.a.1 Continúa con el paso 10 del curso normal de los eventos.
Nivel	Usuario
Prioridad	Crítico

Tabla 12: Descripción textual del CU _ Gestionar Ficheros JAR

CU-4	Asignar un fichero JAR a un framework
Actor	Administrador
Propósito	Asignar un fichero JAR a un framework determinado.



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Resumen	El caso de uso se inicia cuando el usuario accede a la aplicación para asignar un fichero JAR a un framework específico o a un grupo de frameworks, se actualiza el estado del fichero en caso de no estar previamente asignado.	
Referencias	RF4	
Precondiciones	El usuario que accede a la operación debe estar autenticado como administrador.	
Poscondiciones	Se asignó un fichero JAR a un framework o a un grupo de estos, se actualizó el conjunto de ficheros JAR pertenecientes al framework(s) al que se le asignó.	
Curso Normal de los Eventos		
Acción del Actor	Respuesta del Sistema	
1. Se selecciona la opción Asignar fichero JAR.	2. Muestra mensaje: “Debe Consultar ficheros JAR y Frameworks, previamente”. Brinda dos opciones: <ul style="list-style-type: none">❖ Consultar Frameworks y Ficheros JAR❖ Cancelar	
3. Selecciona la opción: Consultar Frameworks y Ficheros JAR.	4. Muestra una IU con ficheros y JAR existente	
5. Selecciona fichero JAR y framework al que le hará la asignación.	6. Si el fichero no estaba asignado anteriormente, cambia la propiedad de asignación por “Asignado”.	
	7 Muestra un mensaje informando al actor del proceso: ¿Desea terminar la asignación?	
	8 Muestra dos opciones posibles: <ul style="list-style-type: none">❖ Aceptar❖ Cancelar	
9. Selecciona la opción de Aceptar.	10. Guarda la información e la base de datos.	



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	11. Muestra un mensaje al usuario: “La operación ha efectuado correctamente”.
	12. Concluye el caso de uso.
Flujo Alterno	
5.a* No se encuentra el framework al que se le hará la asignación en la aplicación	
5.a.1 Selecciona la opción de Cancelar.	5.a.2 No guarda la información.
	5.a.3 Muestra un mensaje al usuario: “La operación ha sido cancelada”.
	5.a.4 Continúa con el paso 12 del curso normal de los eventos.
2.a* Selecciona la opción Cancelar	
	5.a.2 Muestra un mensaje al usuario: “La operación ha sido cancelada”.
	5.a.3 Continúa con el paso 12 del curso normal de los eventos.
10.a* Selecciona la opción Cancelar	
	10.a.1 No guarda la información.
	10.a.2 Muestra un mensaje al usuario: “La operación ha sido cancelada”.
	10.a.3 Continúa con el paso 12 del curso normal de los eventos.
Nivel	Usuario
Prioridad	Crítico

Tabla 13: Descripción textual del CU _ Asignar Fichero JAR a un Framework



CU-5	Consultar ficheros JAR	
Actor	Usuario Anónimo	
Propósito	Mostrar un listado de los ficheros JAR almacenados en el sistema.	
Resumen	El caso de uso inicia cuando el usuario accede a la aplicación para obtener información referente a la existencia o no de ficheros JAR en el sistema.	
Referencias	RF6	
Precondiciones	Debe existir al menos un fichero JAR registrado en la base de datos.	
Poscondiciones	Se obtuvo el listado de todos los ficheros que se encuentran en la aplicación.	
Curso Normal de los Eventos		
Acción del Actor	Respuesta del Sistema	
1. Selecciona la opción: Consultar Ficheros JAR.	2. Accede a la base de datos.	
	3. Muestra un IU con el listado de los ficheros JAR registrados.	
4. Selecciona la opción de continuar.	6. Retorna a la IU inicial.	
	7. Termina el caso de uso.	
Flujo Alternativo		
.....		
Nivel	Subfunción	
Prioridad	Auxiliar	

Tabla 14: Descripción textual del CU _ Consultar Ficheros JAR



CU-6	Consultar frameworks	
Actor	Usuario Anónimo	
Propósito	Mostrar un listado de los frameworks almacenados en el sistema.	
Resumen	El caso de uso inicia cuando el usuario accede a la aplicación para obtener información referente a la existencia o no de frameworks en el sistema.	
Referencias	RF5	
Precondiciones	Debe existir al menos un frameworks registrado en la base de datos.	
Poscondiciones	Se obtuvo el listado de todos los frameworks que se encuentran en la aplicación.	
Curso Normal de los Eventos		
Acción del Actor	Respuesta del Sistema	
1. Selecciona la opción: Consultar Frameworks.	2. Accede a la base de datos.	
	3. Muestra un IU con el listado de los frameworks registrados.	
4. Selecciona la opción de continuar.	6. Retorna a la IU inicial.	
	7. Termina el caso de uso.	
Flujo Alternativo		
.....		
Nivel	Subfunción	
Prioridad	Auxiliar	

Tabla 15: Descripción textual del CU_Consultar Frameworks



CU-7	Descargar ficheros JAR	
Actor	CU_Consultar ficheros JAR	
Propósito	Obtener de forma física el o los ficheros JAR deseados.	
Resumen	El caso de uso inicia cuando el usuario accede al sistema para obtener de forma física el o los fichero JAR útil e importante para su desarrollo pero antes debe consultar los ficheros que existen.	
Referencias	RF8, CU5	
Precondiciones	Debe existir como mínimo un fichero JAR en la base de datos.	
Poscondiciones	Se obtuvo el o los ficheros JAR deseados por el usuario anónimo.	
Curso Normal de los Eventos		
Acción del Actor	Respuesta del Sistema	
1. Selecciona la opción: Consultar ficheros JAR.	2. Muestra la IU referente al CU Consultar ficheros JAR.	
3. Selecciona los ficheros que desea descargar.	4. Muestra un mensaje de información al usuario: ¿Seguro que desea descargar los ficheros seleccionado? Y muestra dos posibles opciones: <div style="text-align: center;"> ❖ Aceptar ❖ Cancelar </div>	
5. Selecciona la opción de Aceptar.	5 Muestra una IU con la dirección donde se guardarán los ficheros seleccionados.	



	6. Brinda dos opciones: ❖ Salvar ❖ Cancelar
7. Selecciona la opción de Salvar.	8. Muestra un mensaje de información al usuario: “La operación fue realizada con éxitos”
	9. Retorna a la IU principal.
	10. Finaliza el caso de uso.
Flujo Alterno	
5.a* Selecciona la opción Cancelar	
	5.a.1 Elimina los datos seleccionados.
	5.a.2 Continúa con el paso 9 del curso normal de los eventos.
7.a* Selecciona la opción Cancelar	
	7.a.1 Muestra un mensaje de información al usuario: “La operación ha sido cancelada”
	7.a.2 Continúa con el paso 9 del curso normal de los eventos.
Nivel	Usuario
Prioridad	Secundario

Tabla 16: Descripción textual del CU _ Descargar Ficheros JAR



CU-8	Descargar frameworks	
Actor	CU_Consultar frameworks	
Propósito	Obtener de forma física el o los frameworks deseados.	
Resumen	El caso de uso inicia cuando el usuario accede al sistema para obtener de forma física el o los frameworks útil e importante para su desarrollo pero antes de acceder a descargar tiene que consultar los framework que existen en la aplicación.	
Referencias	RF7, CU6	
Precondiciones	Debe existir como mínimo un frameworks en la base de datos.	
Poscondiciones	Se obtuvo el o los frameworks deseados por el usuario anónimo.	
Curso Normal de los Eventos		
Acción del Actor	Respuesta del Sistema	
1. Selecciona la opción: Consultar Frameworks.	2. Muestra la IU referente al CU Consultar frameworks.	
3. Selecciona los frameworks que desea descargar.	4. Muestra un mensaje de información al usuario: ¿Seguro que desea descargar los frameworks seleccionado? Y muestra dos posibles opciones: <ul style="list-style-type: none"> ❖ Aceptar ❖ Cancelar 	
5. Selecciona la opción de Aceptar.	5 Muestra una IU con la dirección donde se guardarán los frameworks seleccionados.	



	6. Brinda dos opciones: ❖ Salvar ❖ Cancelar
7. Selecciona la opción de Salvar.	8. Muestra un mensaje de información al usuario: “La operación fue realizada con éxitos”
	9. Retorna a la IU principal.
	10. Finaliza el caso de uso.
Flujo Alterno	
5.a* Selecciona la opción Cancelar	
	5.a.1 Elimina los datos seleccionados.
	5.a.2 Continúa con el paso 9 del curso normal de los eventos.
7.a* Selecciona la opción Cancelar	
	7.a.1 Muestra un mensaje de información al usuario: “La operación ha sido cancelada”
	7.a.2 Continúa con el paso 9 del curso normal de los eventos.
Nivel	Usuario
Prioridad	Secundario

Tabla 17: Descripción textual del CU _ Descargar Frameworks



2.5 CONCLUSIONES

En este capítulo se realizó el estudio de los procesos del negocio donde se pudo definir la no existencia de los mismos, por lo que fue necesario realizar un diagrama de clases del domino para lograr un entendimiento preciso de lo que se requería. Asociado a ello se efectuó el sondeo del sistema, lo que dio lugar a la identificación los casos de usos necesarios para definir las funcionalidades básicas de la aplicación, teniéndose en cuenta los requerimientos funcionales especificados y representados en el Diagrama de Casos de Uso del Sistema. También se enmarca dentro del capítulo las acciones que tienen lugar entre actores y casos de usos detallando estas acciones en las descripciones textuales de los casos de usos definidos en el sistema.



CAPÍTULO 3: ANÁLISIS Y DISEÑO

3.1 INTRODUCCIÓN

El contenido del presente capítulo está determinado por la propuesta de solución final para resolver el problema planteado. Es contenido del mismo se enfoca en observar las clases que intervienen en el proceso a través de los diferentes diagramas de clases ya sea del análisis como del diseño. También es objetivo del mismo, visualizar las interacciones entre los objetos, mediante los diagramas de interacción que surgen al calor del desarrollo de este flujo de trabajo.

3.2 MODELO DE ANÁLISIS.

El modelo de análisis contiene clases y objetos, dichas clases se centran en los requisitos funcionales y son evidentes en el dominio del problema porque representan conceptos y relaciones del dominio. Los diagramas de clases del análisis muestra la interacción actor – sistema de modo conceptual, o sea, visualizan la información a través de estereotipos, exponiendo al usuario un esquema donde se evidencia las interrelaciones que se produce entre actores y clases, es decir, es una representación visual de los objetos, resaltando las relaciones entre ellos, además se utiliza específicamente para la representación de las clases tres estereotipos básicos que se desglosan a continuación.

3.2.1 CLASES DEL ANÁLISIS, ESTEREOTIPOS Y FUNCIONES.

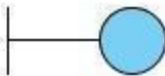


Clase	Estereotipo	Función
Interfaz	 CI_Ejemplo	Modelan la interacción: Actor-Sistema. Ejemplos: Ventanas, Formularios, comunicación con otros sistemas o dispositivos.
Control(Controladora)	 CC_Ejemplo	Coordinan el trabajo de las clases. Encapsulan comportamiento de un CU. Funciones complejas
Entidad	 CE_Ejemplo	Modelan la información del Sistema. Modelan el comportamiento asociado a una información.

Tabla 18: Clases del Análisis.



La realización o no del modelo del análisis está en dependencia de cómo lograr un mejor entendimiento, entre los miembros del equipo de desarrollo. Estudios realizados han demostrado que es más efectivo cuando se unen al desarrollo de software el análisis y el diseño como un único flujo de trabajo, ya que este suaviza la transición al diseño, apoya el cambio a otras plataformas, sirve para tener una visión general de la propuesta de sistema y apoya la aplicación de reingeniería inversa a aplicaciones ya existentes. Por otra parte los diagramas de clases del análisis se estructuran por cada caso de uso, por tanto para cada caso de uso se obtendrá un diagrama de clases del análisis.

A continuación se muestran los diagramas de clases del análisis, que se organizan como anteriormente se listaron los casos de uso:

3.2.2 DIAGRAMAS DE CLASES DEL ANÁLISIS

3.2.2.1 DIAGRAMA DE CLASES CU_AUTENTICAR USUARIO

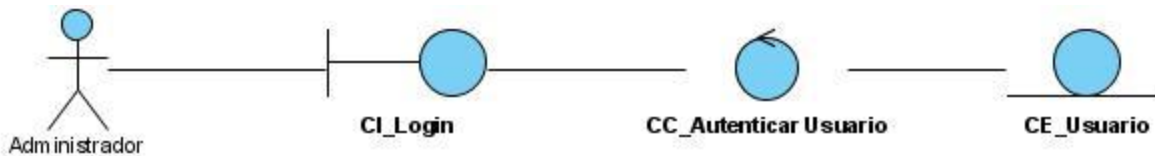


Figura 3: Diagrama de clases del análisis CU _ Autenticar Usuario

3.2.2.2 DIAGRAMA DE CLASES CU _ GESTIONAR FRAMEWORKS

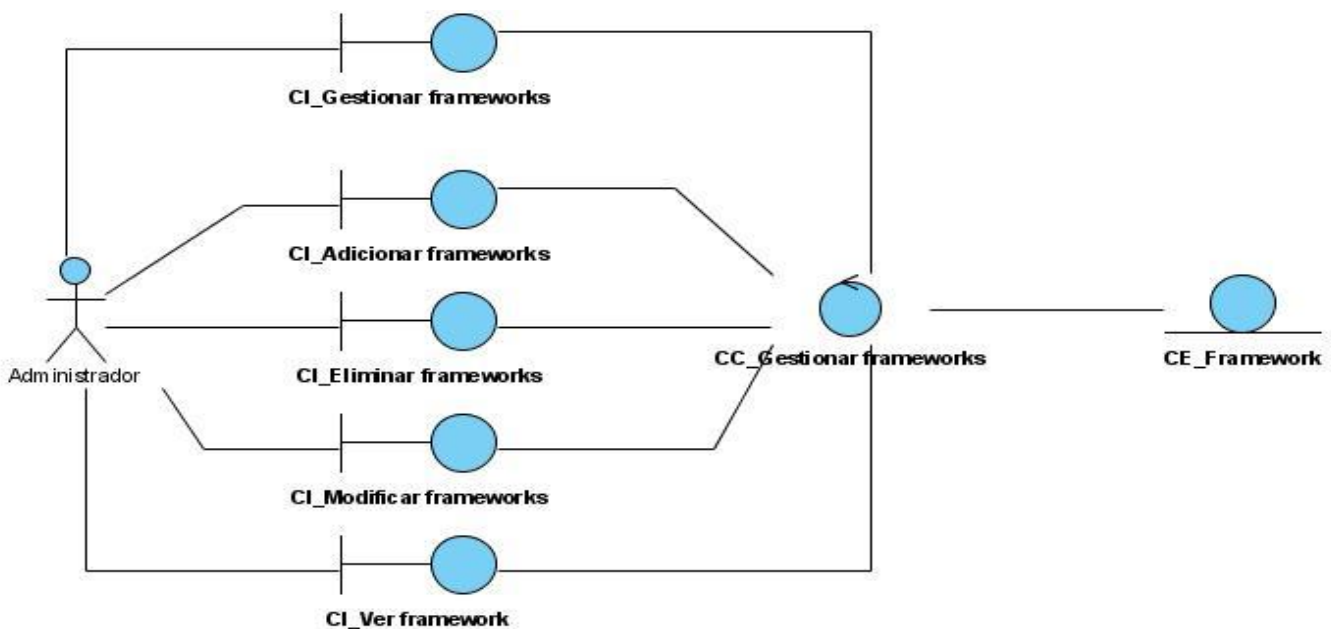


Figura 4: Diagrama de clases del análisis CU _ Gestionar Frameworks



3.2.2.3 DIAGRAMA DE CLASES CU_GESTIONAR FICHEROS JAR

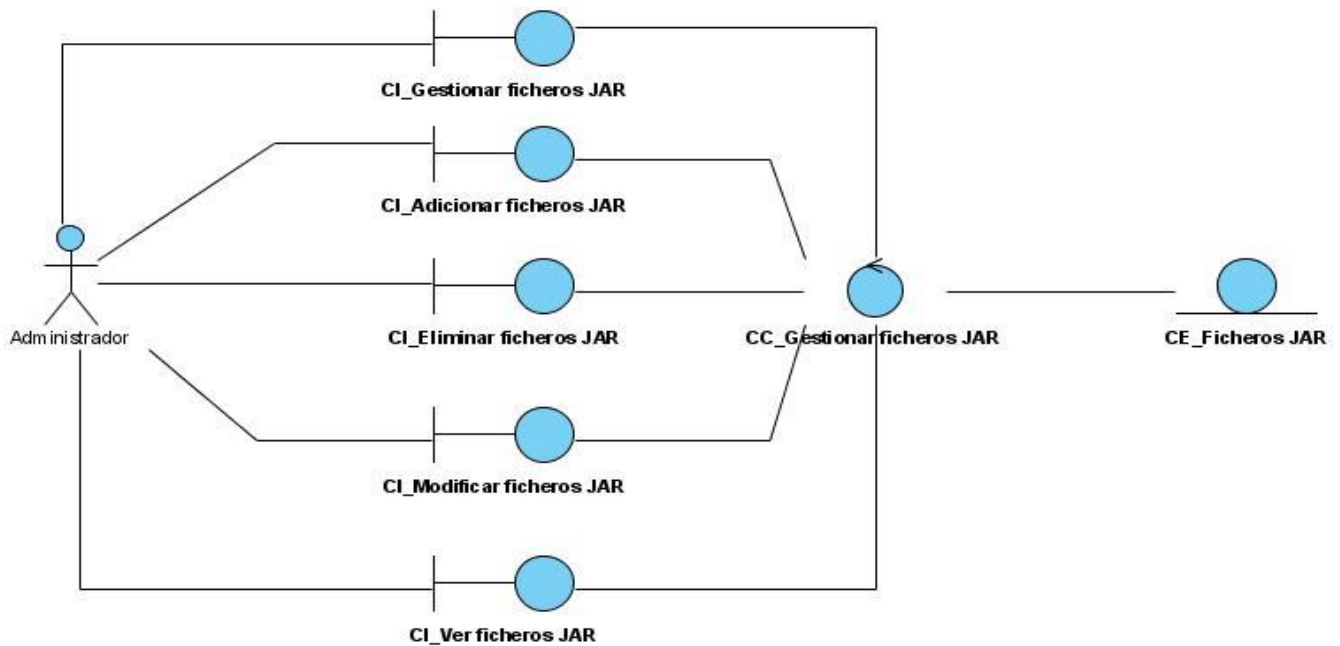


Figura 5: Diagrama de clases del análisis del CU _ Gestionar Ficheros JAR

3.2.2.4 DIAGRAMA DE CLASES CU_ASIGNAR UN FICHERO JAR A UN FRAMEWORKS

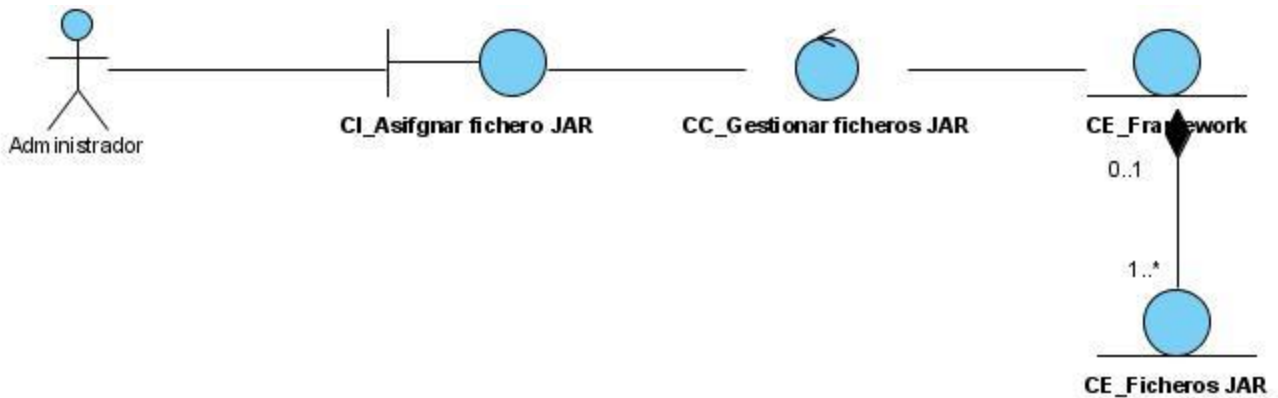


Figura 6: Diagrama de clases del análisis del CU _ Asignar Fichero JAR a un Framework



3.2.2.5 DIAGRAMA DE CLASES CU_CONSULTAR FICHEROS JAR

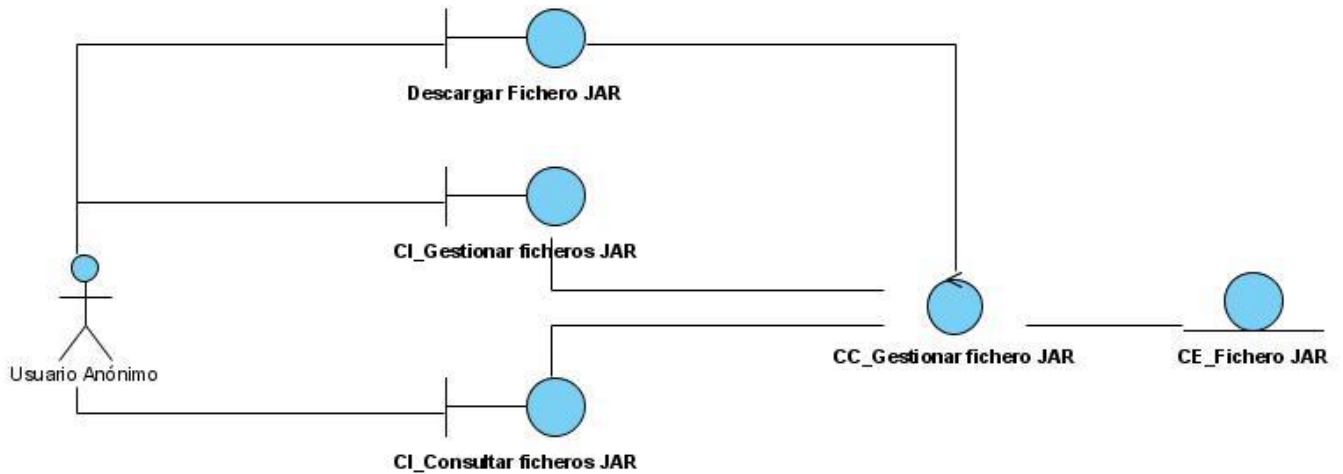


Figura 7: Diagrama de clases del análisis CU _ Consultar Ficheros JAR

3.2.2.6 DIAGRAMA DE CLASES CU_CONSULTAR FRAMEWORKS

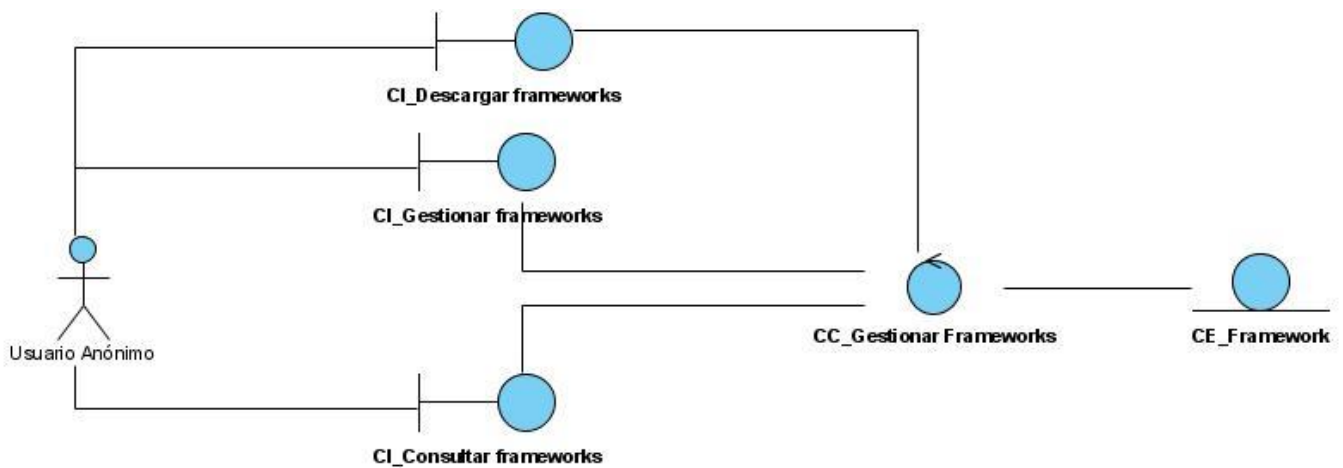


Figura 8: Diagrama de clases del análisis del CU _ Consultar Frameworks



3.2.2.7 DIAGRAMA DE CLASES CU_DESCARGAR FICHEROS JAR

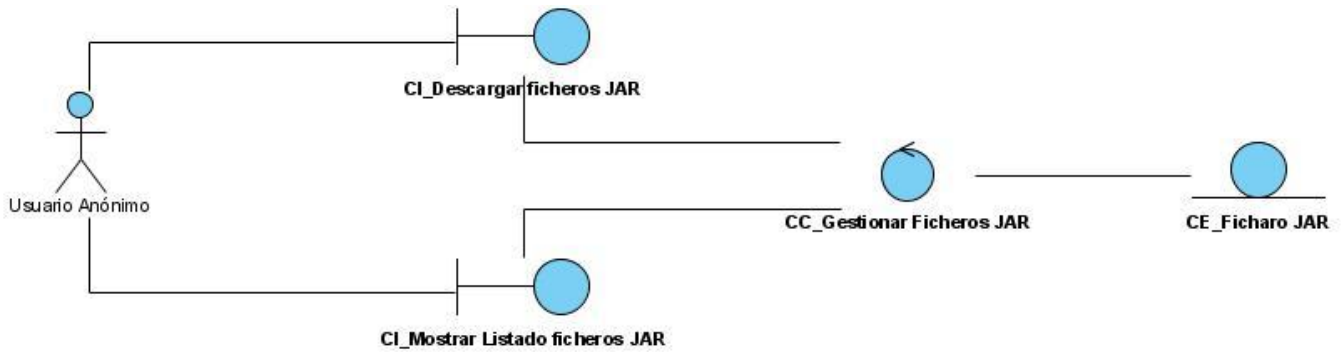


Figura 9: Diagrama de clases del análisis CU _ Descargar Ficheros JAR

3.2.2.7 DIAGRAMA DE CLASES CU_DESCARGAR FRAMEWORKS

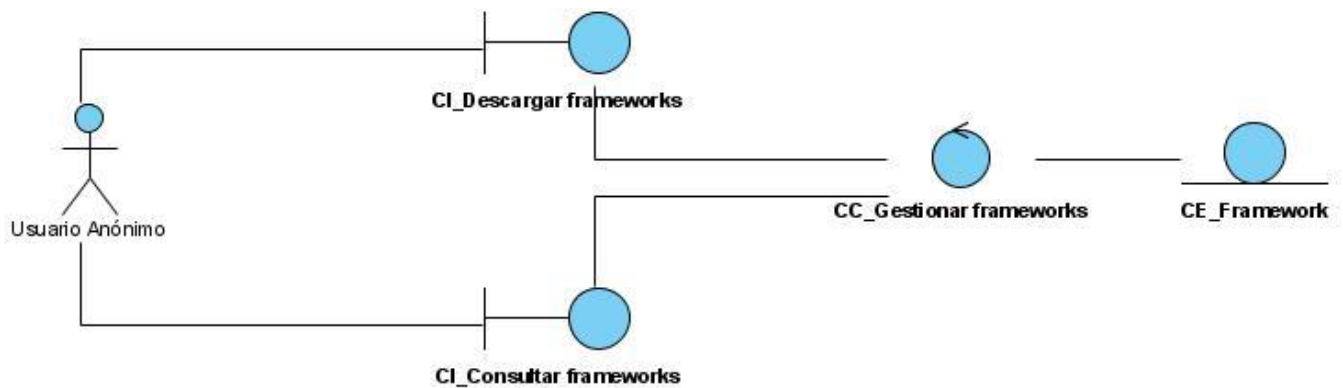


Figura 10: Diagrama de clases del análisis CU _ Descargar Frameworks



3.3 MODELO DE DISEÑO

En el diseño se modela el sistema, incluyendo su arquitectura, para que soporte todos los requisitos. Una entrada esencial es el resultado del análisis visto anteriormente. La esencia de esta etapa es la elaboración de los diagramas de interacción, que muestran gráficamente como los objetos se comunican entre ellos a fin de cumplir con los requerimientos.

3.3.1 DIAGRAMAS DE INTERACCIÓN

UML como lenguaje de de modelado define tres grandes grupos de diagramas: de estructuras estáticas o estáticas, de comportamiento o dinámicas y de implementación. Específicamente en los diagramas dinámicos se engloban los de actividades, los de estado y los de interacción, donde estos últimos se clasifican en diagramas de colaboración o secuencia, pero aunque difieren en su organización o distribución, coinciden en querer visualizar o mostrar la interacción entre objetos.

Los diagramas de colaboración destacan la organización de los objetos que participan así como el paso de mensajes entre los mismos. Los diagramas de secuencia resaltan el tiempo de vida de cada uno de los objetos que intervienen en el proceso de interacción así como el orden temporal de los mensajes que se intercambian entre objetos lo que ofrece una visión clara del flujo a lo largo del tiempo. Cada uno de ellos poseen gran importancia para los desarrolladores porque mediante ellos se muestra una representación visual de lo que se quiere lograr una vez culminado el sistema o la aplicación Web, por ello, aunque no sea imprescindible la presencia de ambos, en el trabajo se exponen los dos en conjunto, debido a la significancia que poseen por separados. Además se debe destacar que los diagramas de colaboración se confeccionaron, en este trabajo, lo más cercano al diseño posible y que ambos son modelados teniendo en cuenta cada escenario de un caso de uso, donde se define como escenario a las diferentes acciones que se pueden realizar en un mismo caso de uso, pudiendo pertenecer a un escenario más de una transacción, que no es más que el par de acción-reacción, o sea, acción del actor- respuesta del sistema.

A continuación se exhibirán los diferentes diagramas de interacción, comenzando por los de colaboración y luego los de secuencia. En este capítulo solo se mostrarán los de los casos de uso arquitectónicamente significativos, aquellos que acumulan la mayor parte de las funcionalidades del sistema y son relevantes para la arquitectura. Los demás diagramas estarán incluidos en los anexos de este trabajo.



3.3.1.1 DIAGRAMAS DE COLABORACIÓN

3.3.1.1.1 DIAGRAMAS DE COLABORACIÓN CU_GESTIONAR FRAMEWORKS (4 ESCENARIOS)

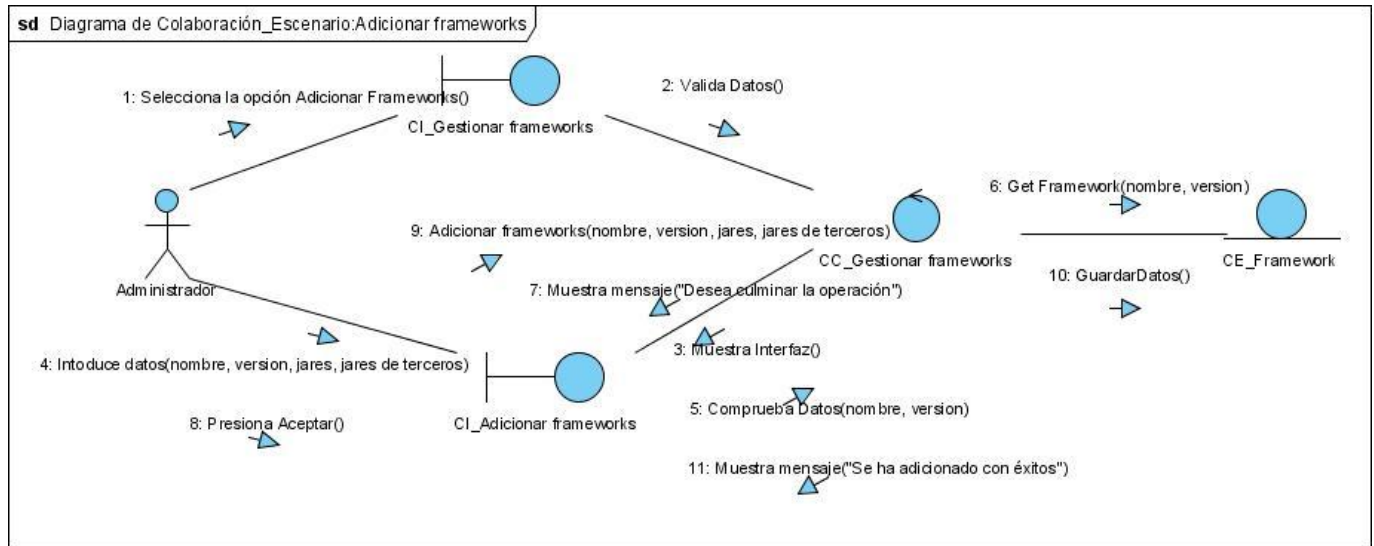


Figura 11: Diagrama de colaboración CU _ Gestionar Frameworks Escenario: Adicionar Frameworks

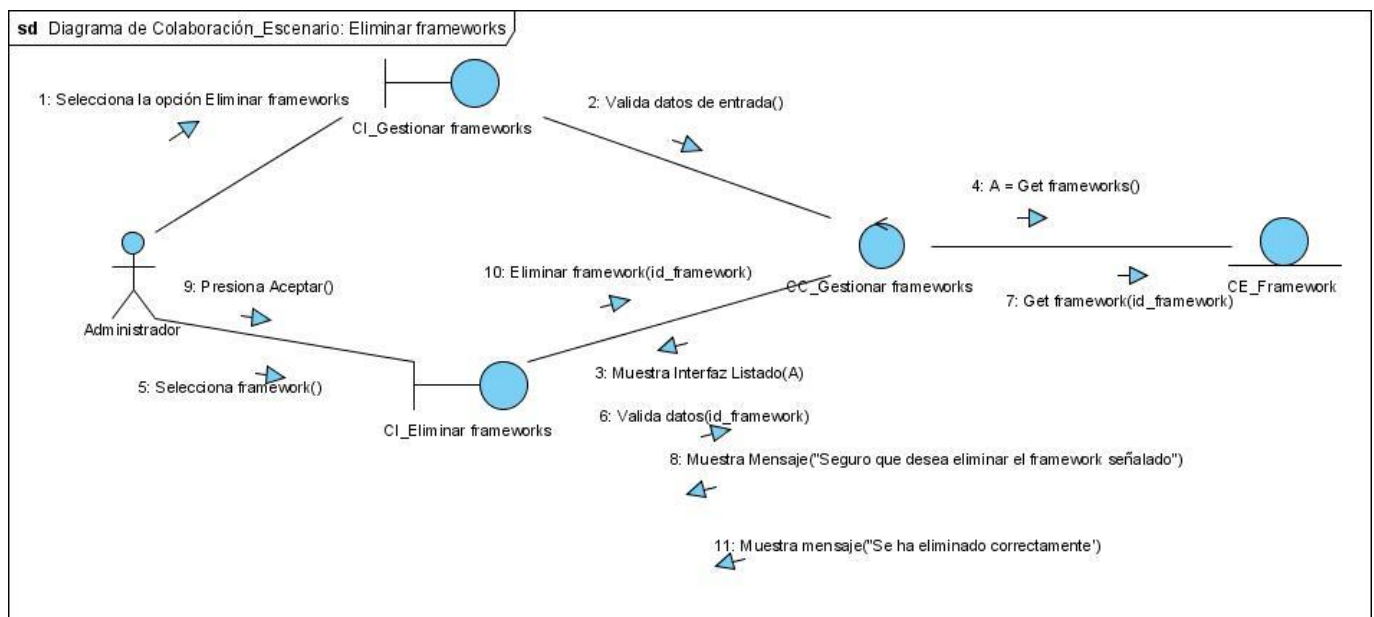


Figura 12: Diagrama de colaboración CU _ Gestionar Frameworks Escenario_Eliminar Frameworks

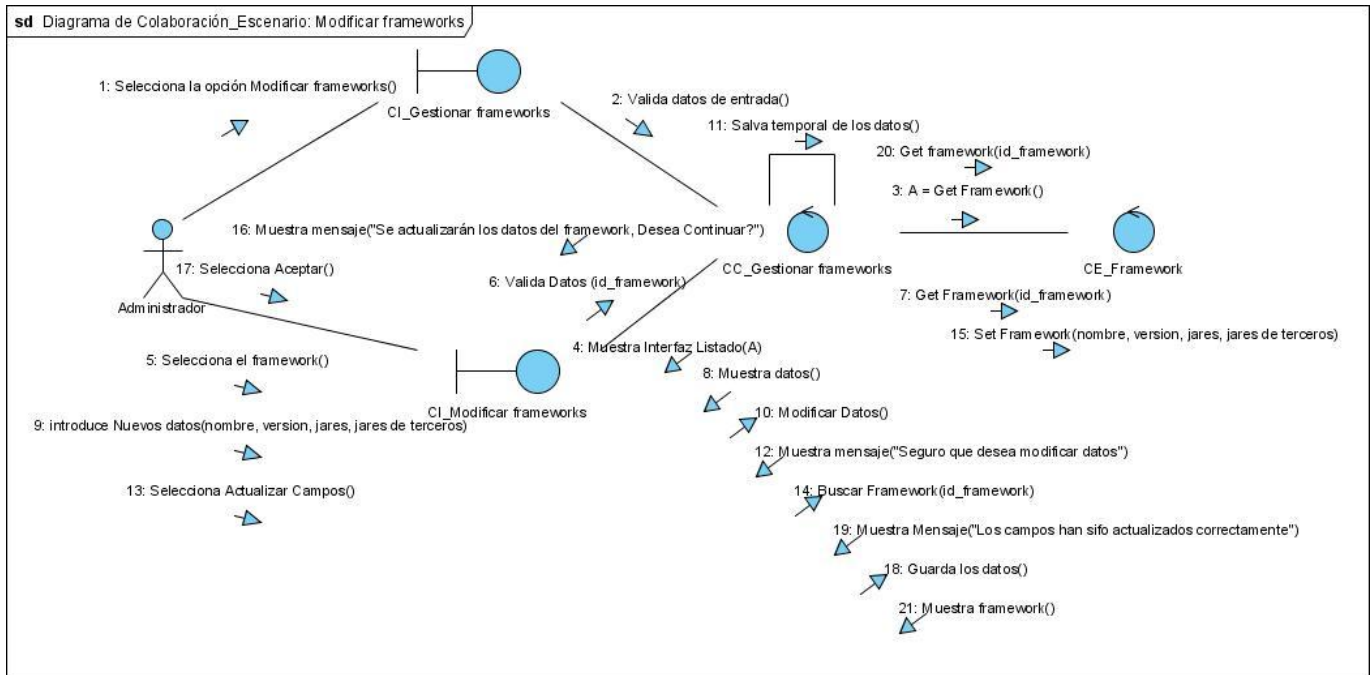


Figura 13: Diagrama de colaboración CU _ Gestionar Frameworks Escenario _ Modificar Frameworks

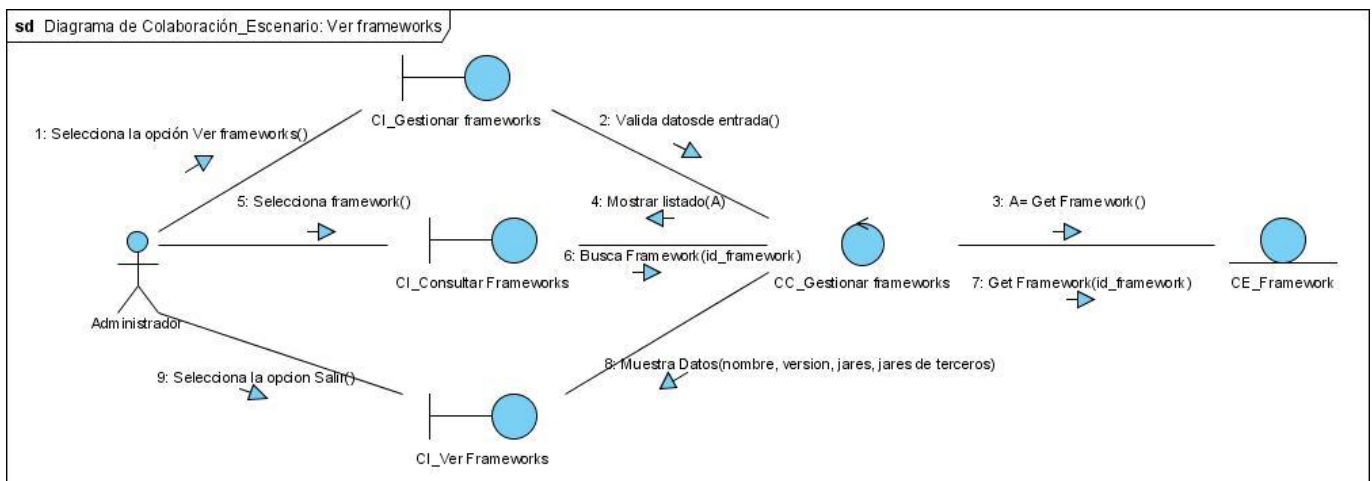


Figura 14: Diagrama de colaboración CU _ Gestionar Frameworks Escenario _ Ver Frameworks



3.3.1.1.2 DIAGRAMAS DE COLABORACIÓN CU_GESTIONAR FICHEROS JAR(4 ESCENARIOS)

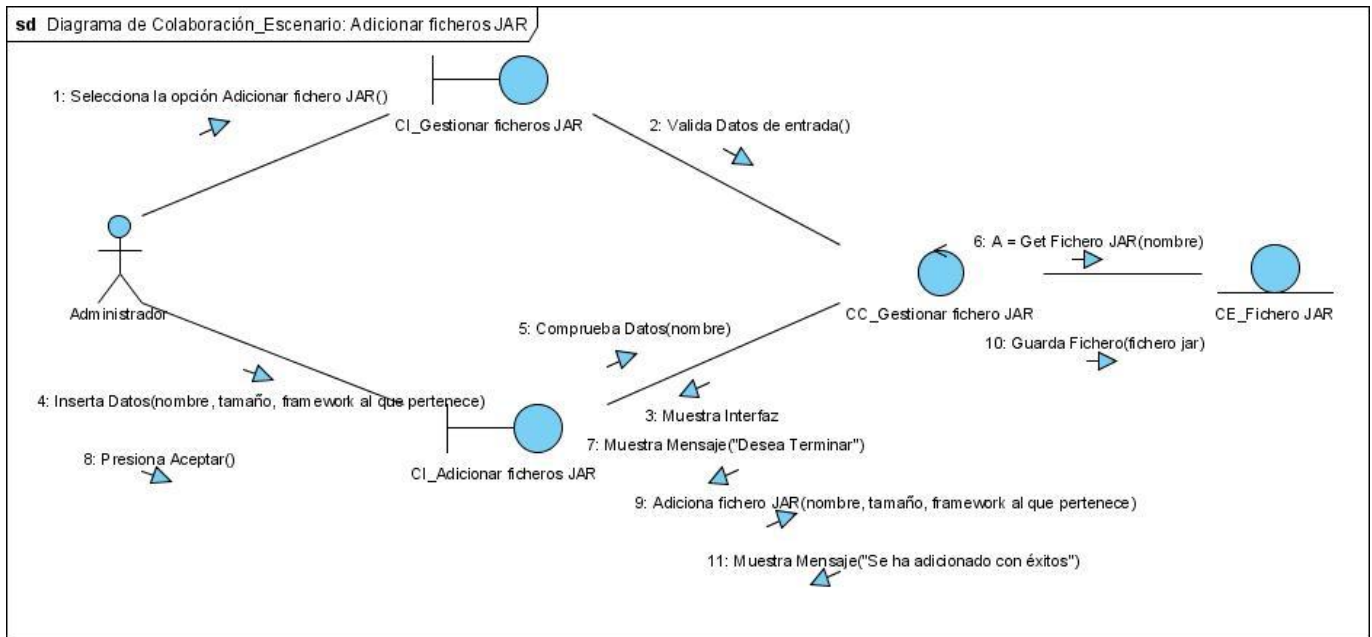


Figura 15: Diagrama de colaboración CU _ Gestionar ficheros JAR Escenario _ Adicionar Ficheros JAR

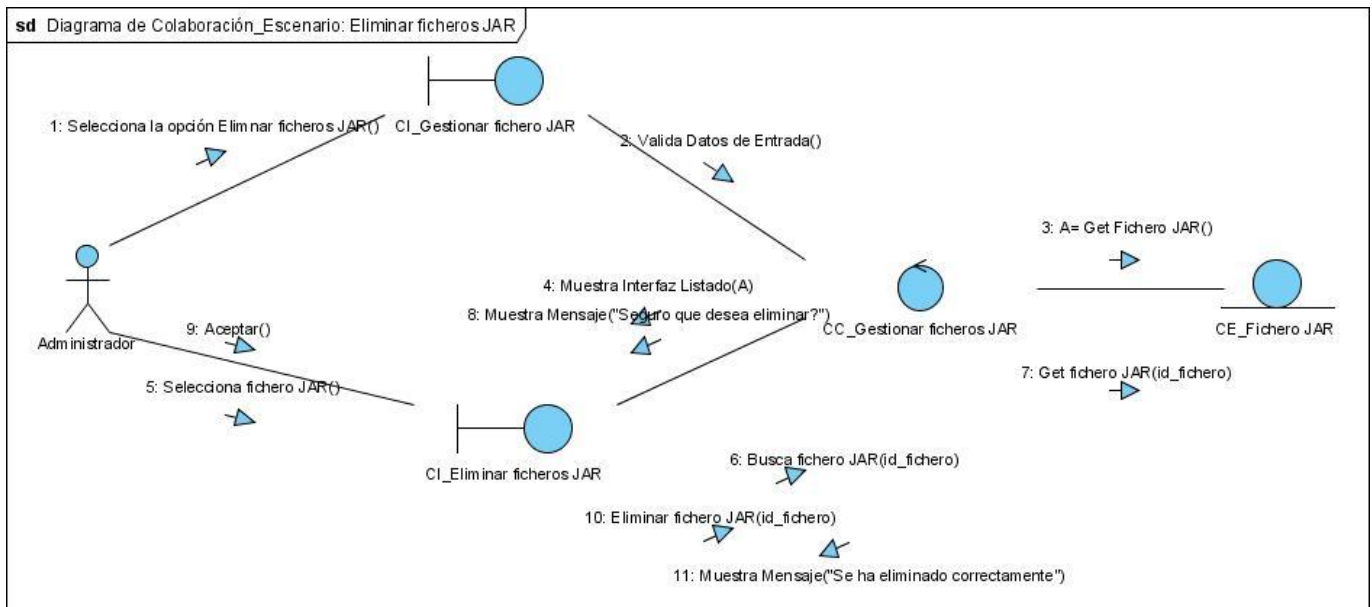


Figura 16: Diagrama de colaboración CU _ Gestionar Ficheros JAR Escenario _ Eliminar Ficheros JAR

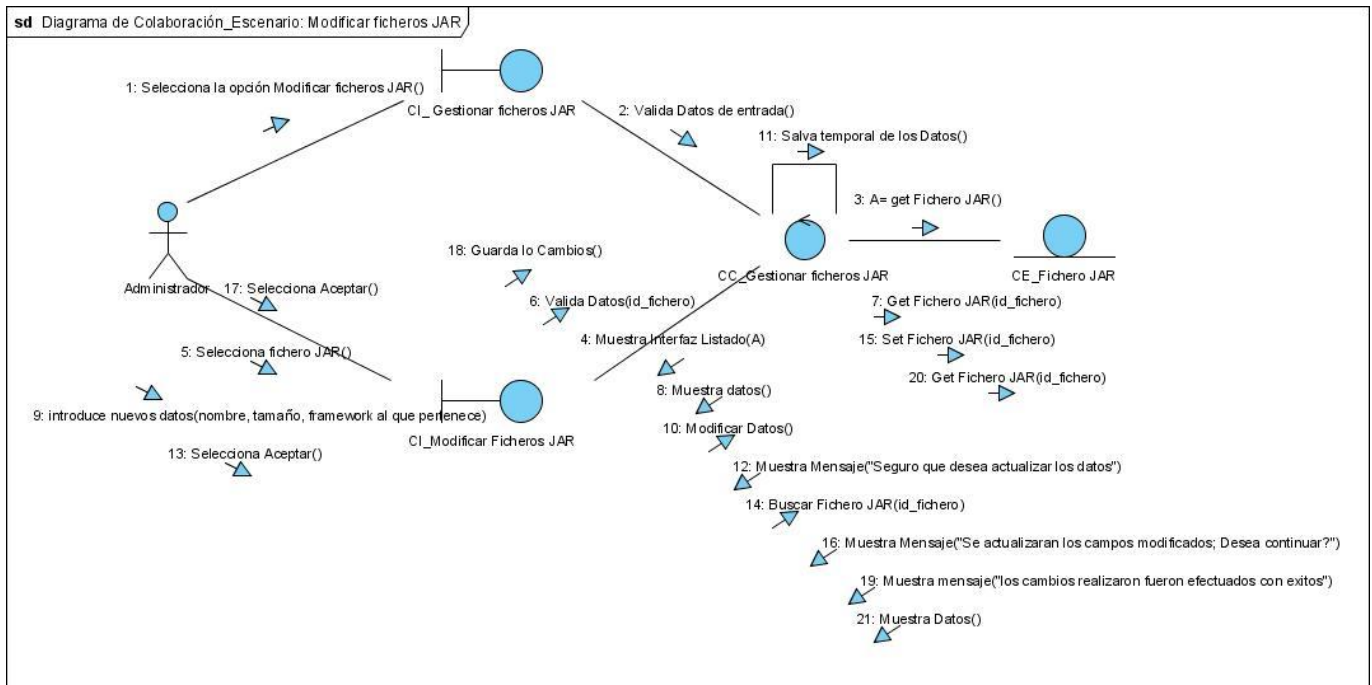


Figura 17: Diagrama de colaboración CU_Gestionar Ficheros JAR Escenario _ Modificar Ficheros JAR

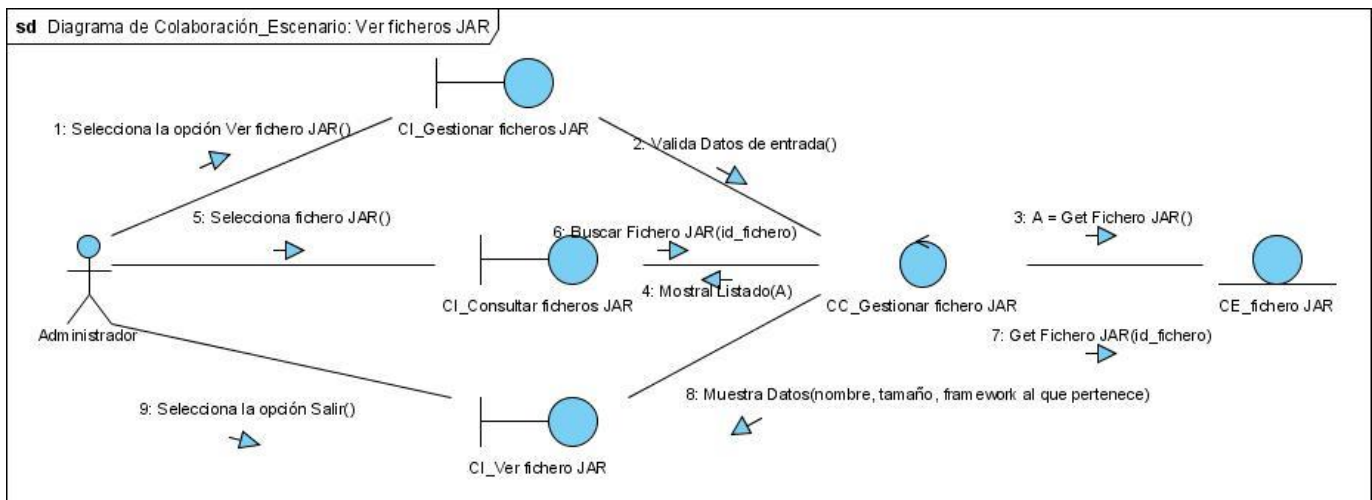


Figura 18: Diagrama de colaboración CU_Gestionar Ficheros JAR Escenario _ Ver Ficheros JAR



3.3. 1.1.3 DIAGRAMA DE COLABORACIÓN CU_ASIGNAR UN FICHERO JAR A UN FRAMEWORKS

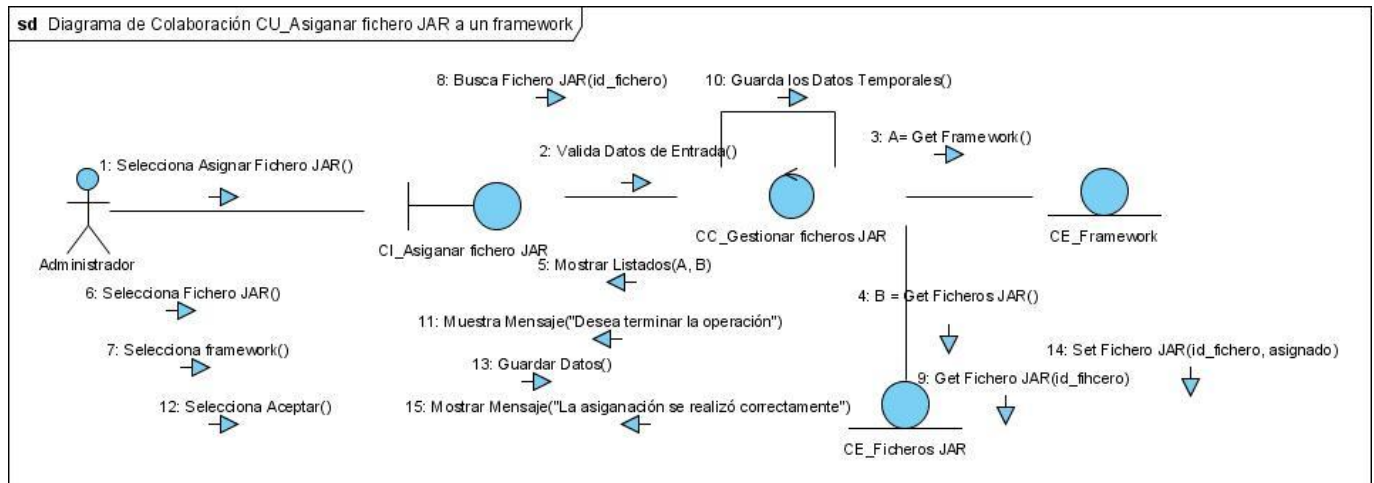


Figura 19: Diagrama de colaboración CU_Assignar Fichero JAR a un Framework

Los restantes diagramas de colaboración de cada uno de los casos de usos que no se exponen en el epígrafe se localizan en los anexos del trabajo. ([Anexo 1 – Anexo 5](#))

3.3.1.2 DIAGRAMAS DE SECUENCIA

3.3. 1.2.1 DIAGRAMAS DE SECUENCIA CU_GESTIONAR FRAMEWORKS (4 ESCENARIOS)

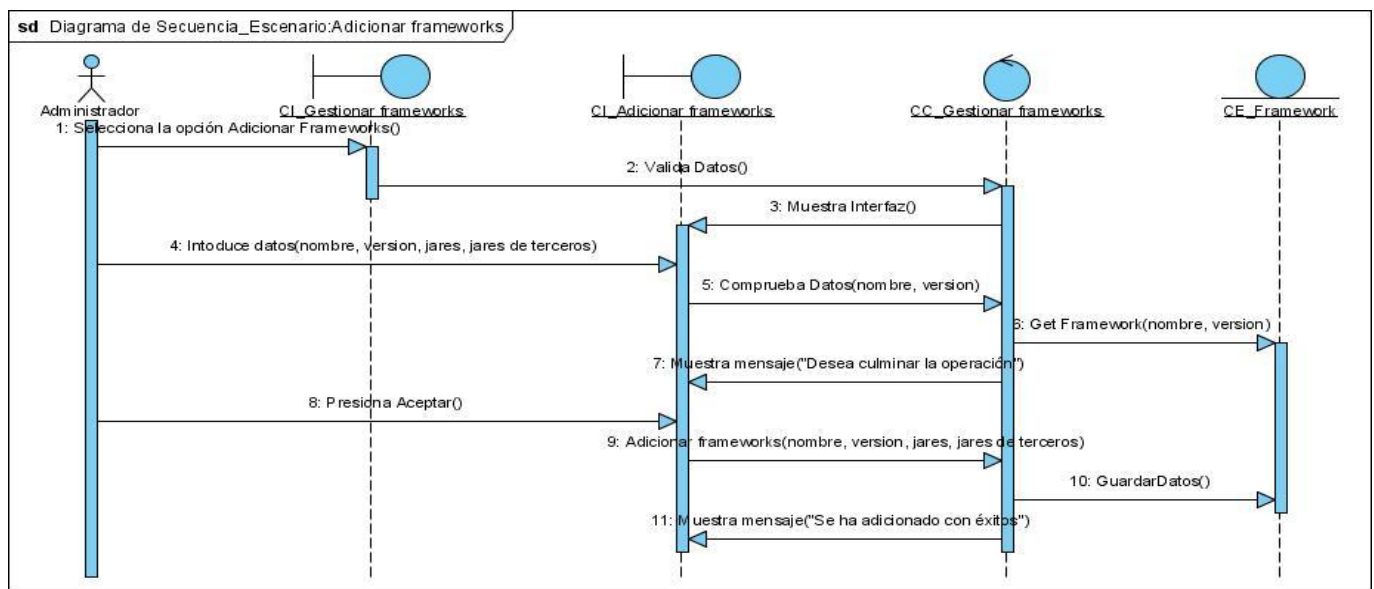


Figura 20: Diagrama de secuencia CU_Gestionar Frameworks Escenario Adicionar Frameworks

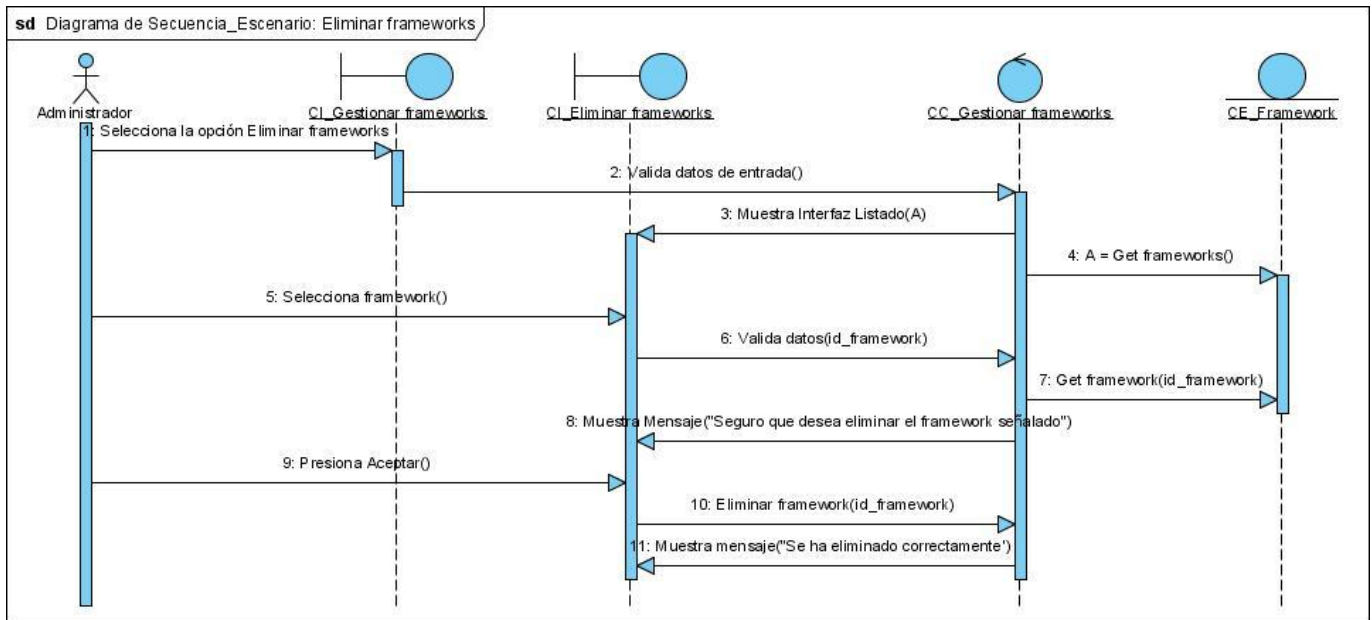


Figura 21: Diagrama de secuencia CU _ Gestionar Frameworks Escenario _ Eliminar Frameworks

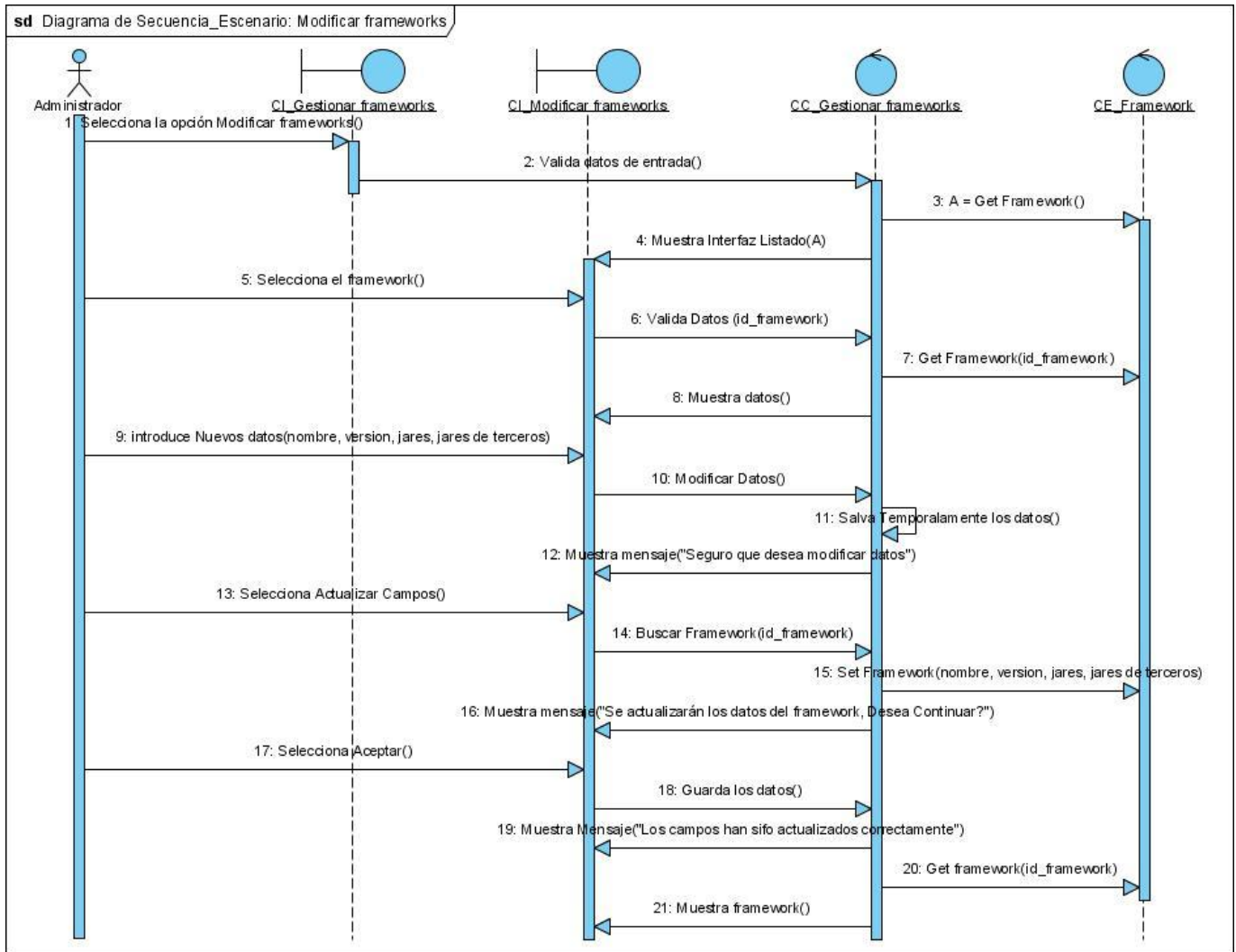


Figura 22: Diagrama de secuencia CU _ Gestionar Frameworks Escenario _ Modificar Frameworks

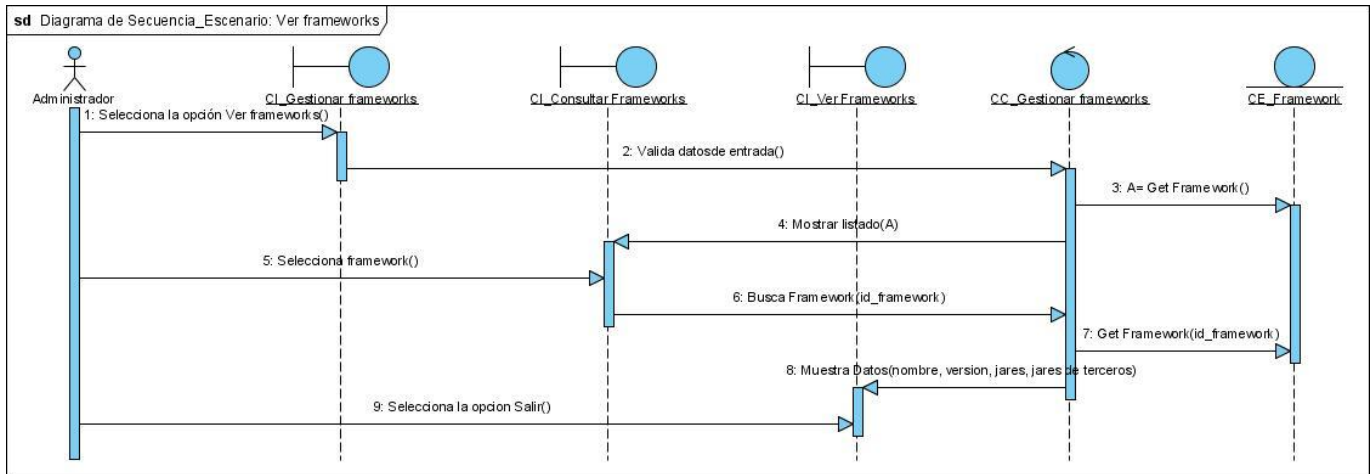


Figura 23: Diagrama de secuencia CU _ Gestionar Frameworks Escenario _ Ver Frameworks

3.3. 1.2.2 DIAGRAMAS DE SECUENCIA CU_GESTIONAR FICHEROS JAR

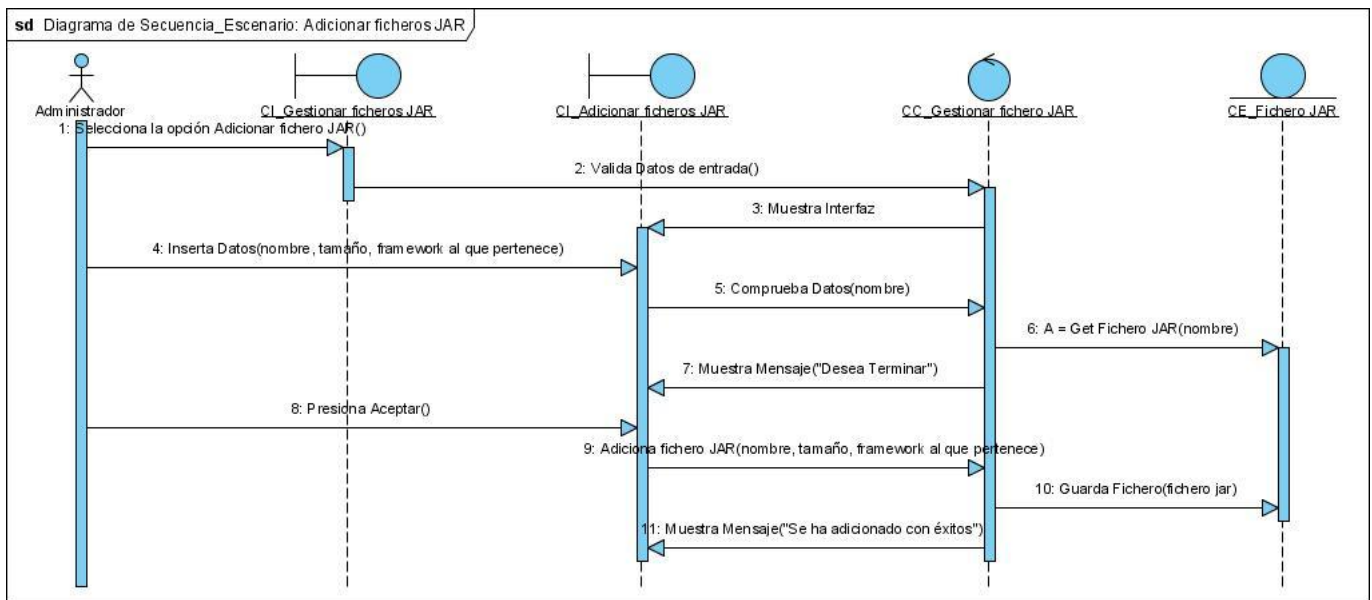


Figura 24: Diagrama de secuencia CU _ Gestionar Fichero JAR Escenario _ Adicionar Fichero JAR

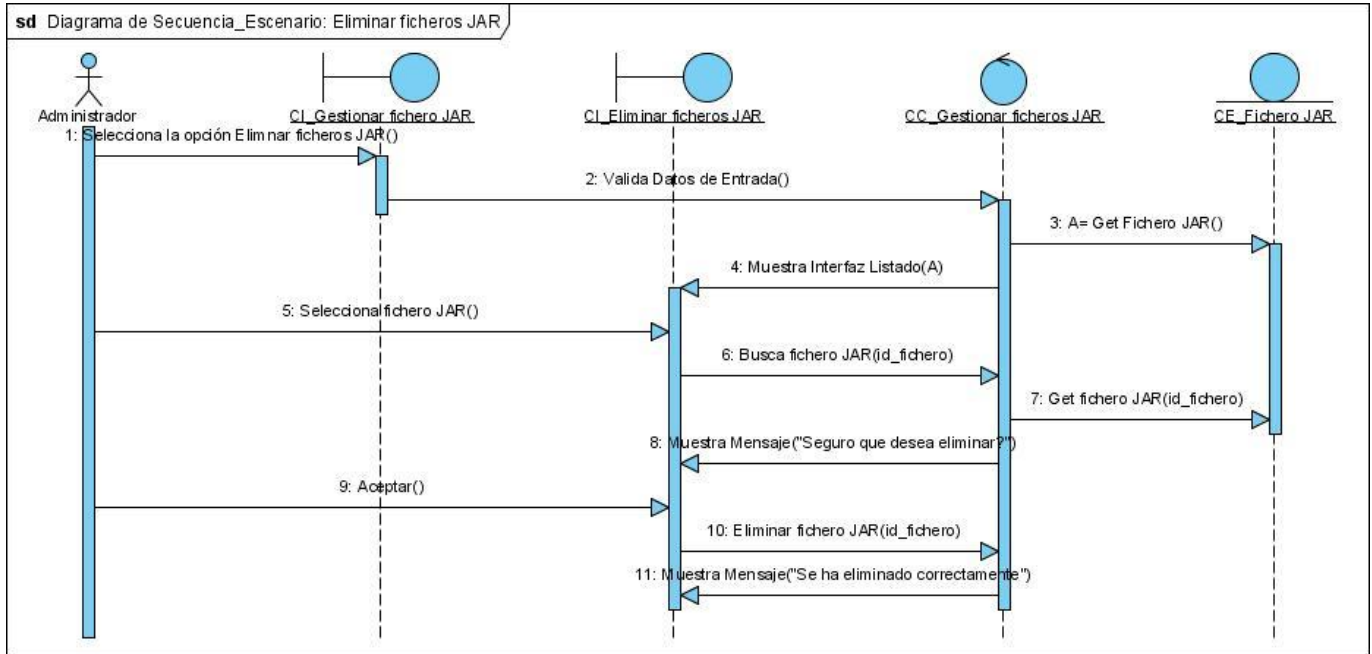


Figura 25: Diagrama de secuencia CU _ Gestionar Fichero JAR Escenario _ Eliminar Ficheros JAR

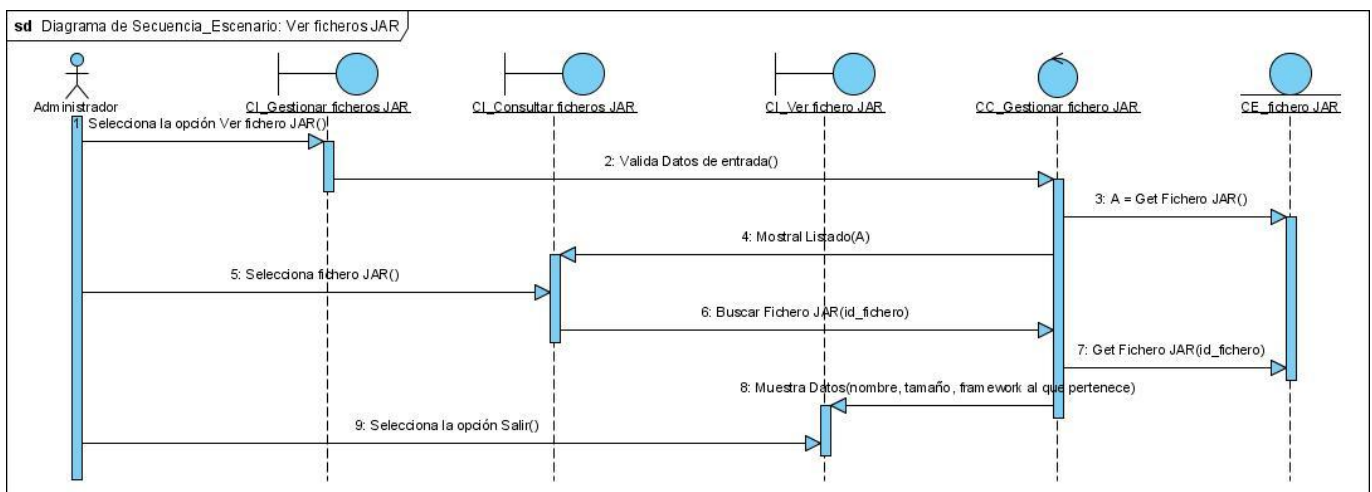


Figura 26: Diagrama de secuencia CU _ Gestionar Ficheros JAR Escenario _ Ver Ficheros JAR

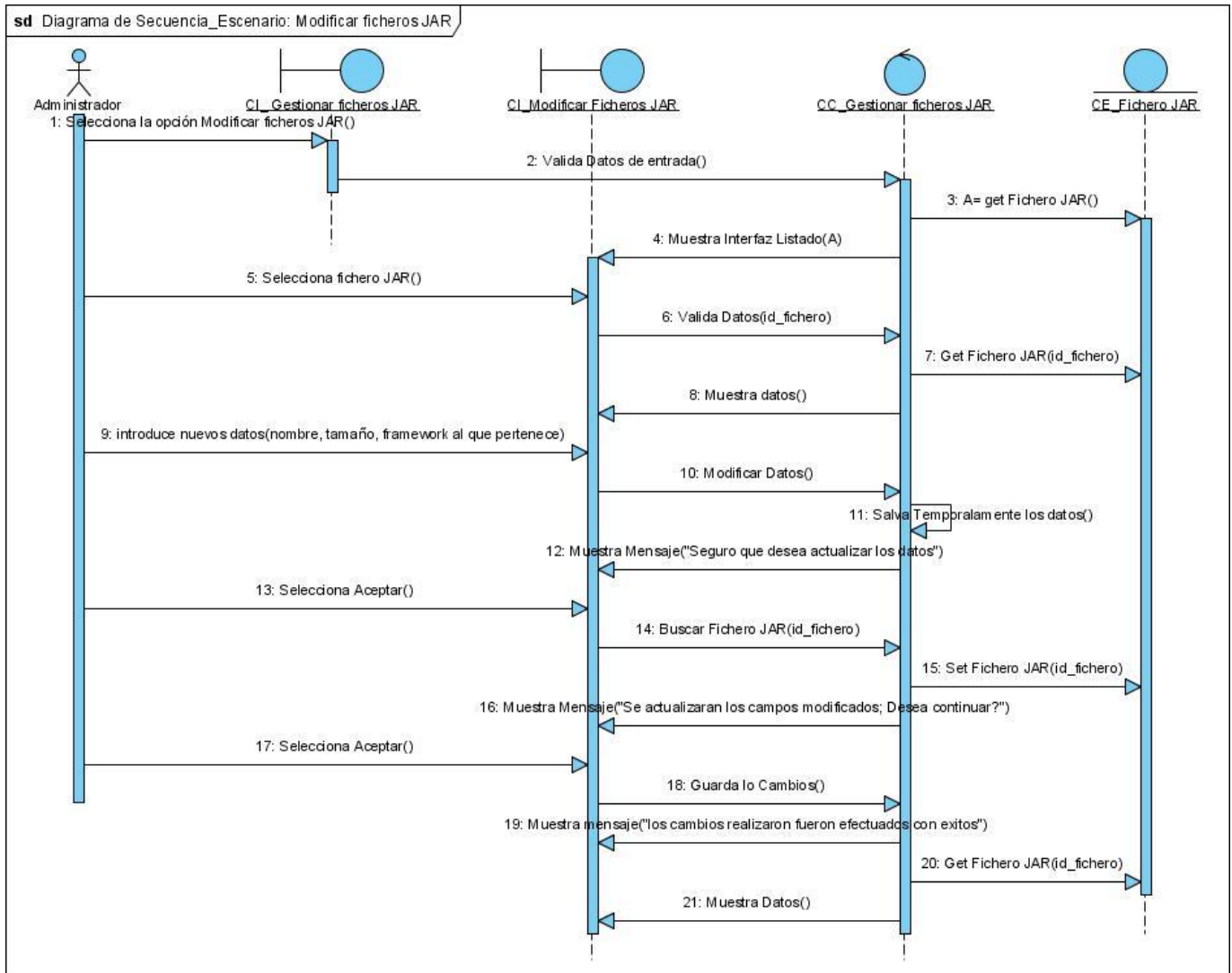


Figura 27: Diagrama de secuencia CU _ Gestionar ficheros JAR Escenario_ Modificar Ficheros JAR



3.3. 1.2.3 DIAGRAMA DE SECUENCIA CU _ ASIGNAR UN FICHERO JAR A UN FRAMEWORK

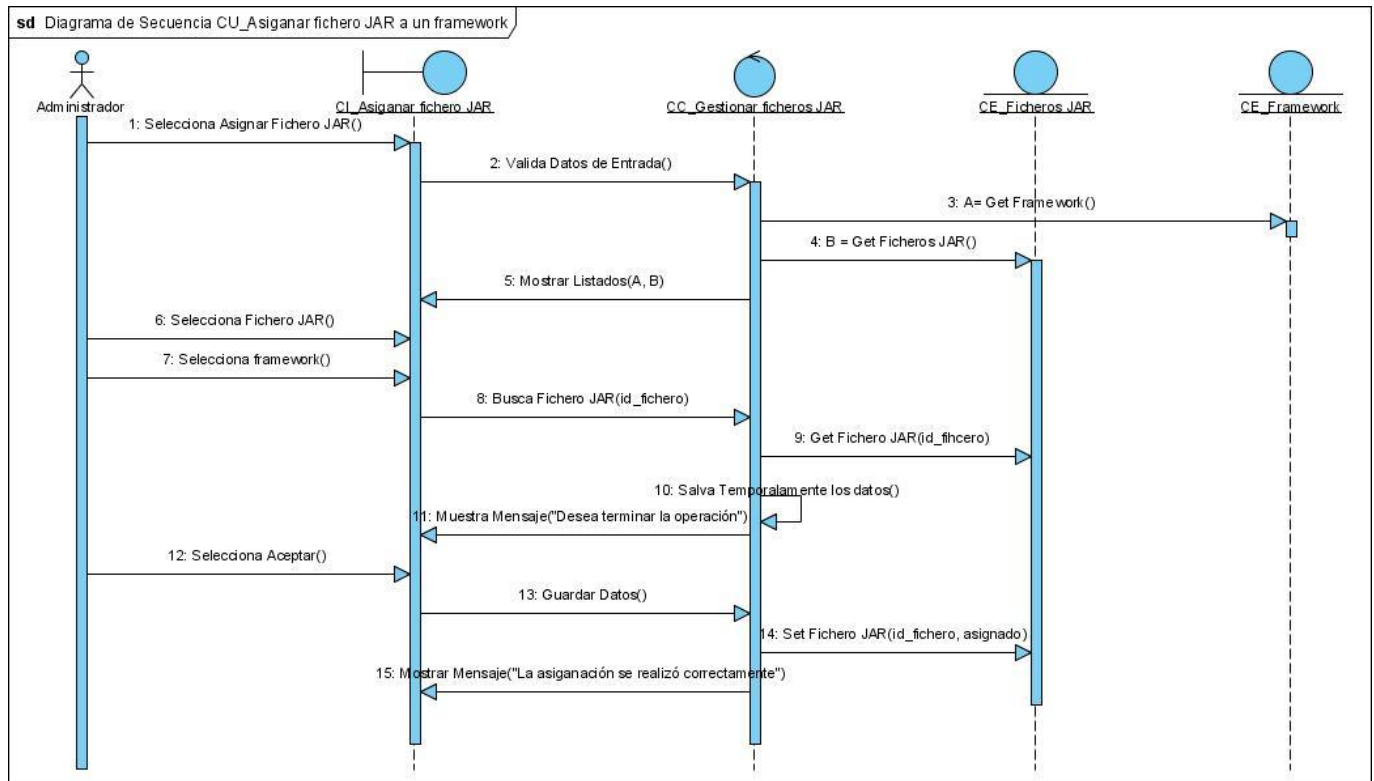


Figura 28: Diagrama de secuencia CU _ Asignar un Fichero JAR a un Framework

Los restantes diagramas de secuencia de cada uno de los casos de usos que no se exponen en el epígrafe se localizan en los anexos del trabajo. ([Anexo 6 – Anexo10](#))

3.3.2 DISEÑO DEL SISTEMA

El diagrama de clases del diseño es un modelo de objetos que describe la realización de los casos de uso, centrándose en los requisitos funcionales y no funcionales. Su propósito es adquirir una comprensión de los requisitos funcionales y las restricciones. Mediante él se crea una entrada apropiada y un punto de partida para la implementación de la aplicación. Es un modelo físico, dinámico y no genérico, es el que da forma al sistema mientras intenta preservar la estructura definida por el modelo de análisis.



A continuación se muestra una tabla donde se muestra las terminologías utilizadas en los diagramas de clases del diseño para un mejor entendimiento de los mismos.

Clases	Estereotipo	Función
Server Page [SP] (en español Página Servidora)		Representa una clase para el acceso a datos, su principal función es construir la CP.
Client Page [CP] (en español Página Cliente)		Representa una clase que interactúa con el usuario, su función es visualizar, interactuar y mostrar lo que el usuario necesita.
Form (en español Formulario)		Es una clase que interactúa directamente con el usuario y su principal función es enviar los datos entrados por el usuario a la SP.

Tabla 19: Clases del Diseño

3.3.2.1 DIAGRAMAS DE CLASES DEL DISEÑO

3.3.2.1.1 DIAGRAMA DE CLASES DEL DISEÑO CU _ AUTENTICAR USUARIO

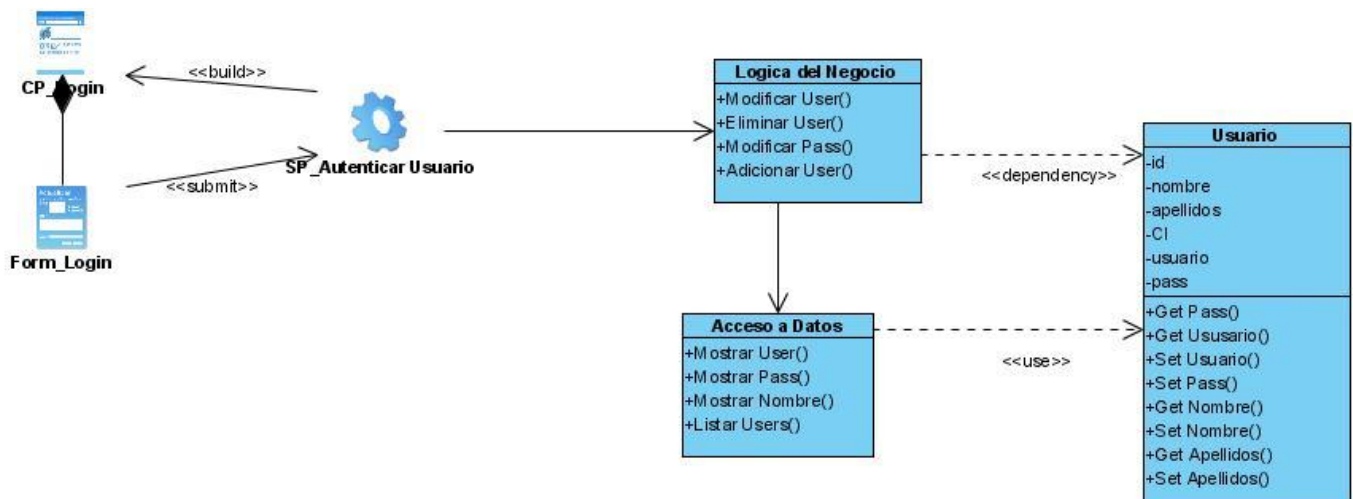


Figura 29: Diagrama de clases del diseño CU _ Autenticar Usuario



3.3.2.1.2 DIAGRAMA DE CLASES DEL DISEÑO CU_ASIGNAR UN FICHERO JAR A UN FRAMEWORK

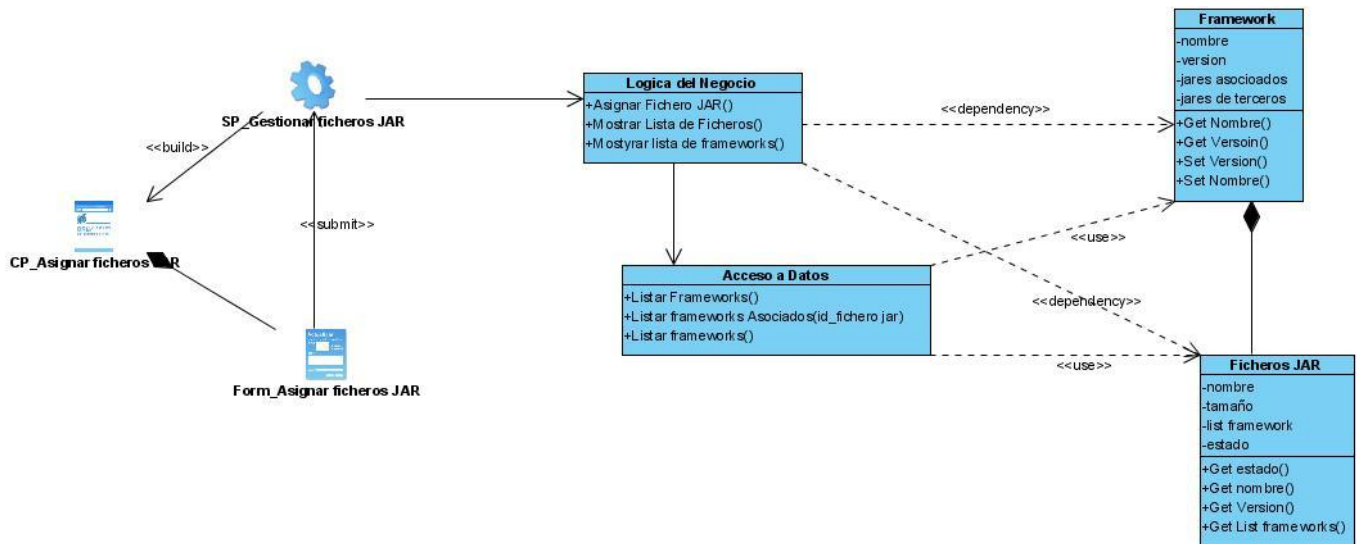


Figura 30: Diagrama de clases del diseño CU _ Asignar Fichero JAR a un Framework

3.3.2.1.3 DIAGRAMA DE CLASES DEL DISEÑO CU_GESTIONAR FRAMEWORKS

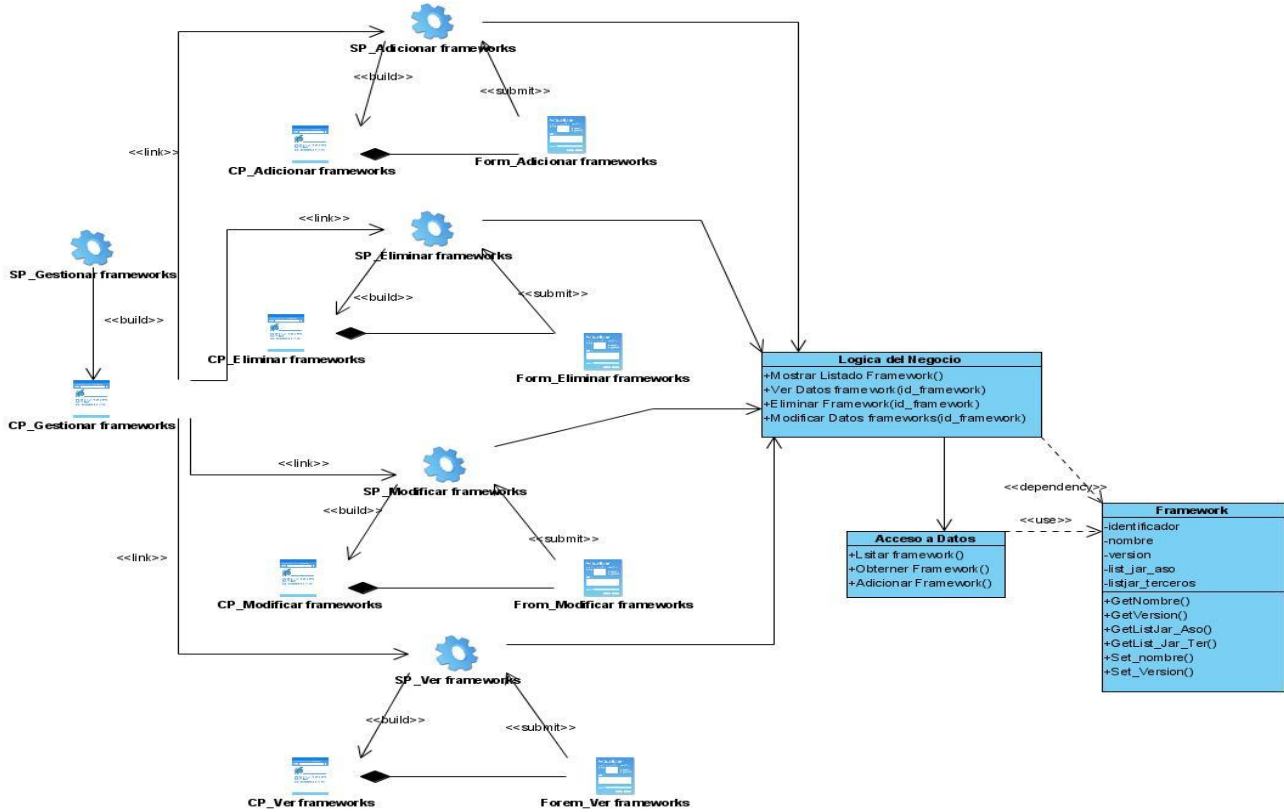


Figura 31: Diagrama de clases del diseño CU _ Gestionar Frameworks

3.3.2.1.4 DIAGRAMA DE CLASES DEL DISEÑO CU_GESTIONAR FICHEROS JAR

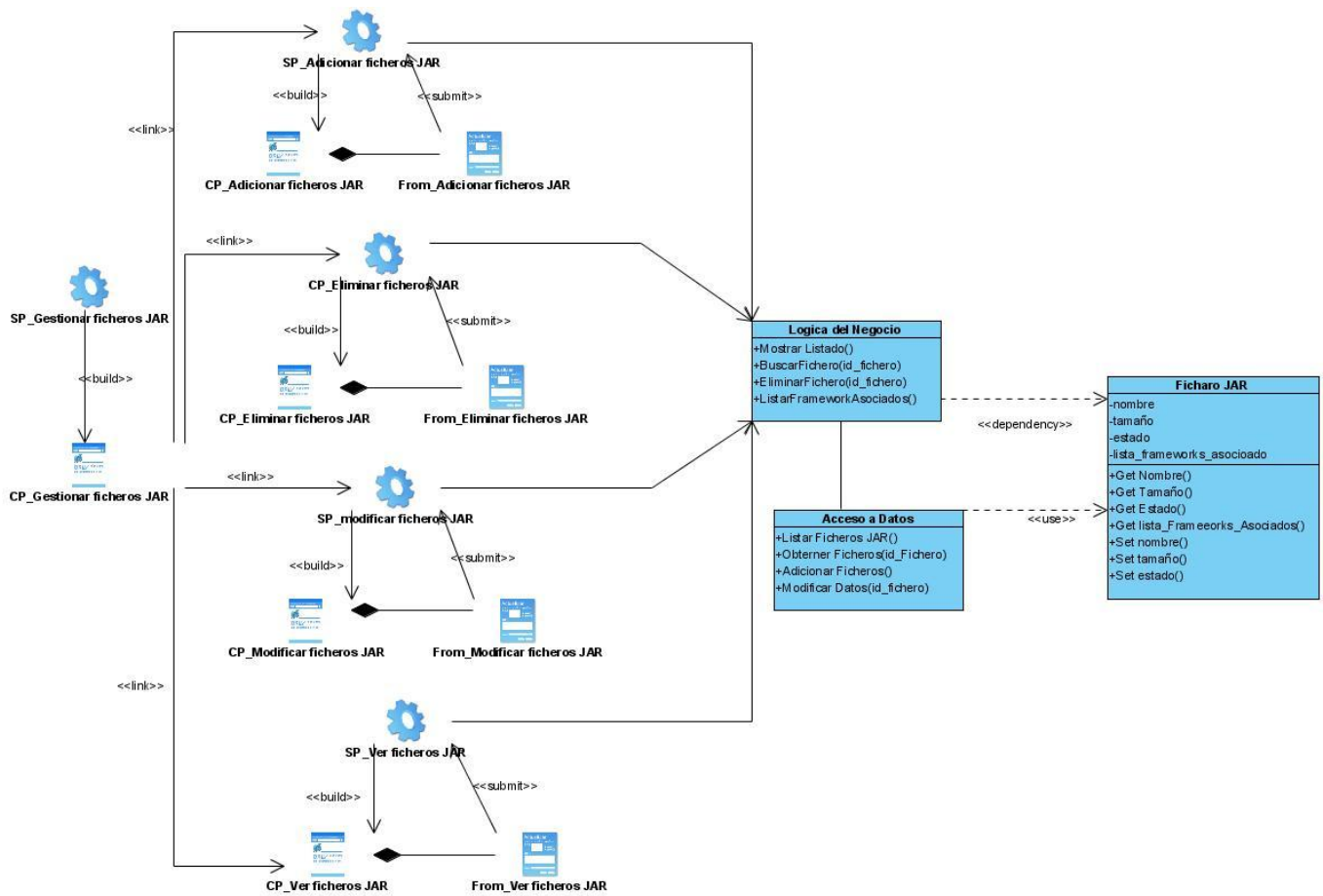


Figura 32: Diagrama de clases del diseño CU _ Gestionar Ficheros JAR



3. 3.2.1.5 DIAGRAMA DE CLASES DEL DISEÑO CU_CONSULTAR FRAMEWORKS

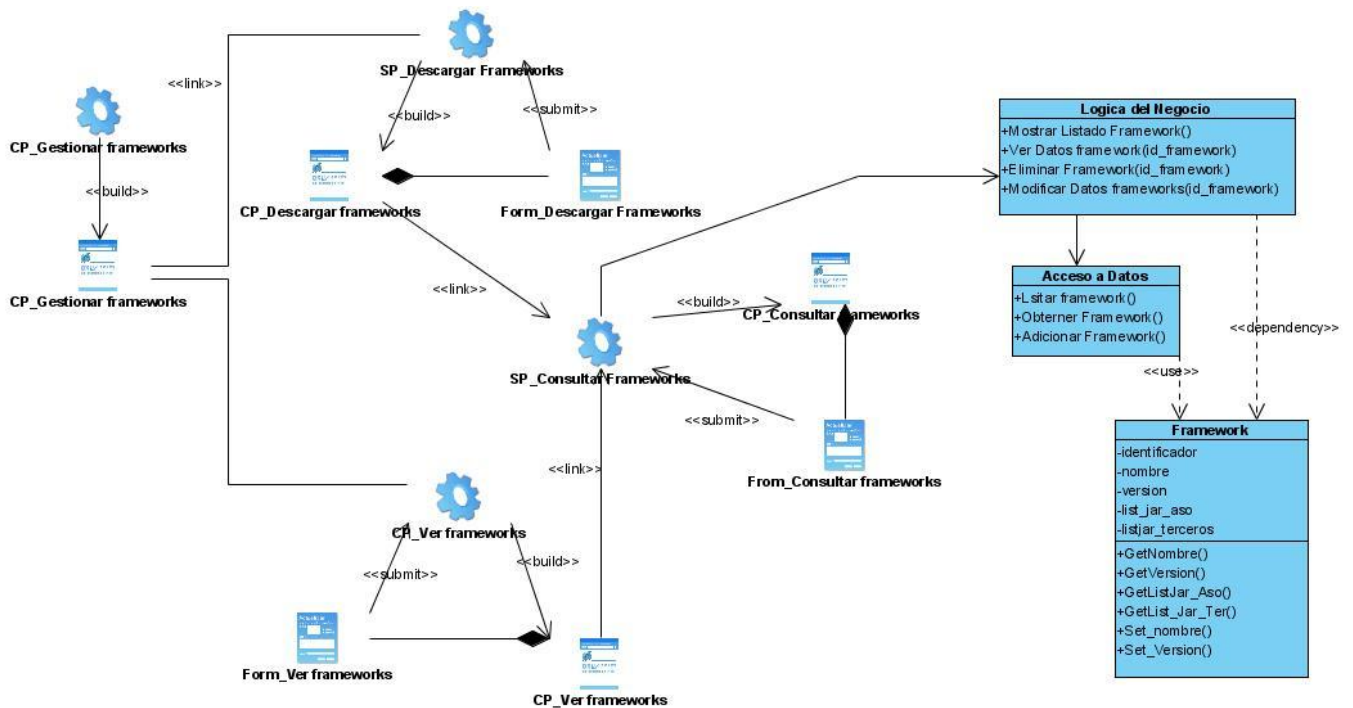


Figura 33: Diagrama de clases del diseño CU _ Consultar Frameworks

3. 3.2.1.3 DIAGRAMA DE CLASES DEL DISEÑO CU_DESCARGAR FRAMEWORKS

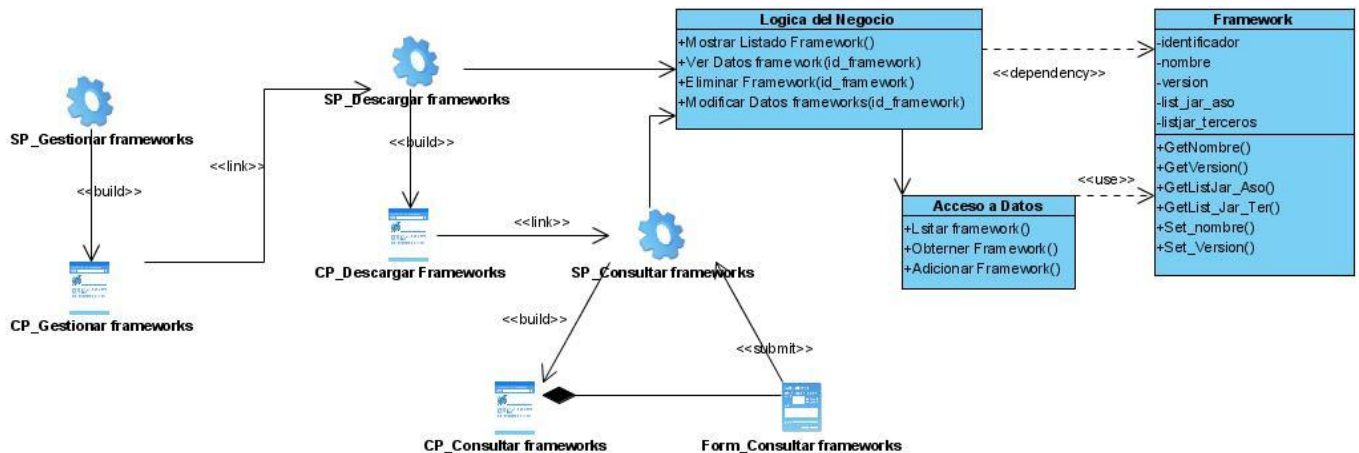


Figura 34: Diagrama de clases del diseño CU_Descargar Frameworks



3.3.2.1.3 DIAGRAMA DE CLASES DEL DISEÑO CU_CONSULTAR FICHEROS JAR

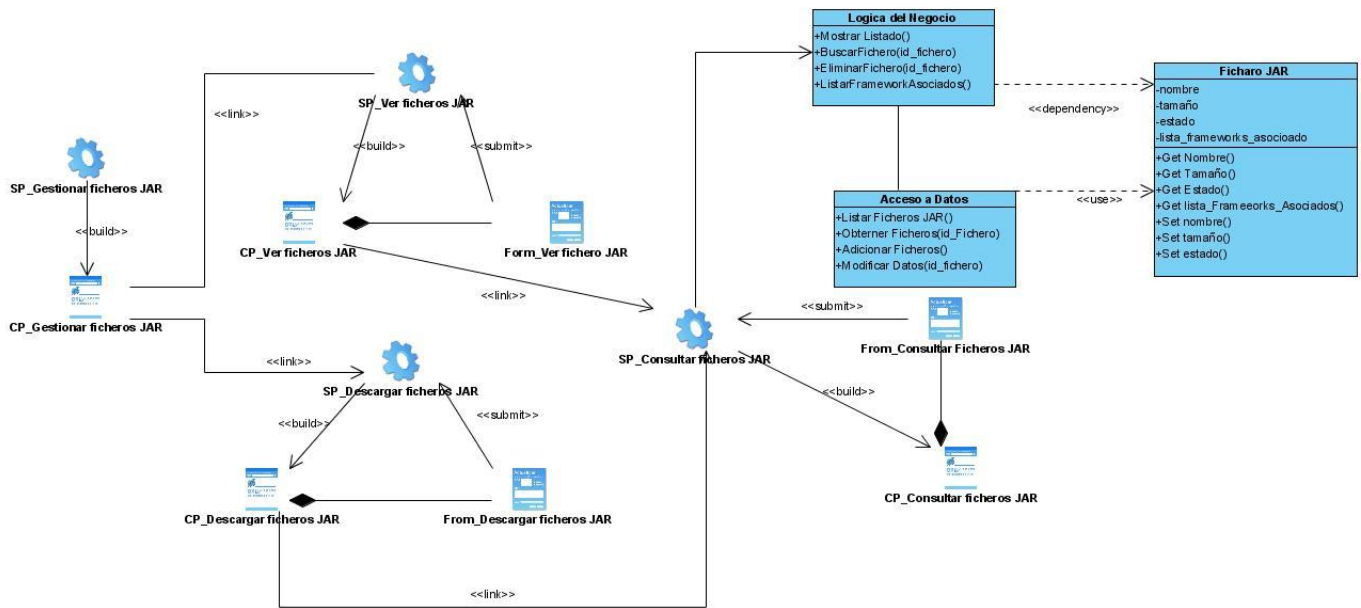


Figura 35: Diagrama de clases del diseño CU _ Consultar Ficheros JAR

3.3.2.1.3 DIAGRAMA DE CLASES DEL DISEÑO CU_DESCARGAR FICHEROS JAR

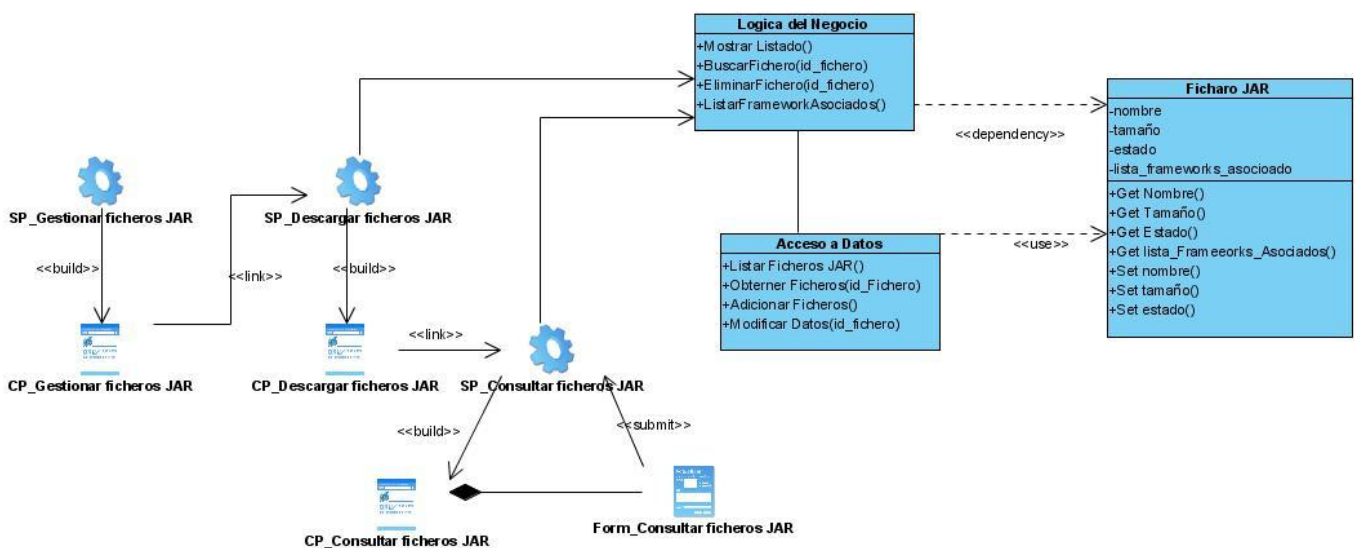


Figura 36: Diagrama de clases del diseño CU_Descargar Ficheros JAR



3.4 DISEÑO DE LA BASE DE DATOS

La explosión de nuevas tecnologías que empezó con la introducción de las computadoras y la llegada de la Internet, ha brindado opciones y herramientas que son explotadas con gran intensidad en la actualidad. Una de ellas es la utilización de instrumentos de información en la generación de las bases de datos, la cual es de gran importancia para cualquier entidad, ya sea grande o pequeña, o para cualquier institución en sentido general donde se maneje determinada cantidad de información.

El Glosario IEEE de Ingeniería del Software (IEEE 1990) define el término “base de datos” de la siguiente forma: “Una colección de datos interrelacionados almacenados conjuntamente en uno o más ficheros de computadora.” [13]

En fin una base de datos: es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta.

En el actual desarrollo, la base de datos que se representa estaría vinculada a los datos tanto de los ficheros JAR como de los frameworks, conjuntamente con los usuarios que sean administradores de la aplicación resultante, que aún siendo sencilla o poco compleja no pierde la importancia por la cual fue creada ya que mantiene la organización, el almacenamiento y control de acceso de la información que maneja.

En esta sección se muestra el diagrama de clases persistentes y el modelo entidad relación así como se describen las tablas de la base de datos, que dan soporte al contenido manejado por el sistema.

3.4.1 DIAGRAMA DE CLASES PERSISTENTES

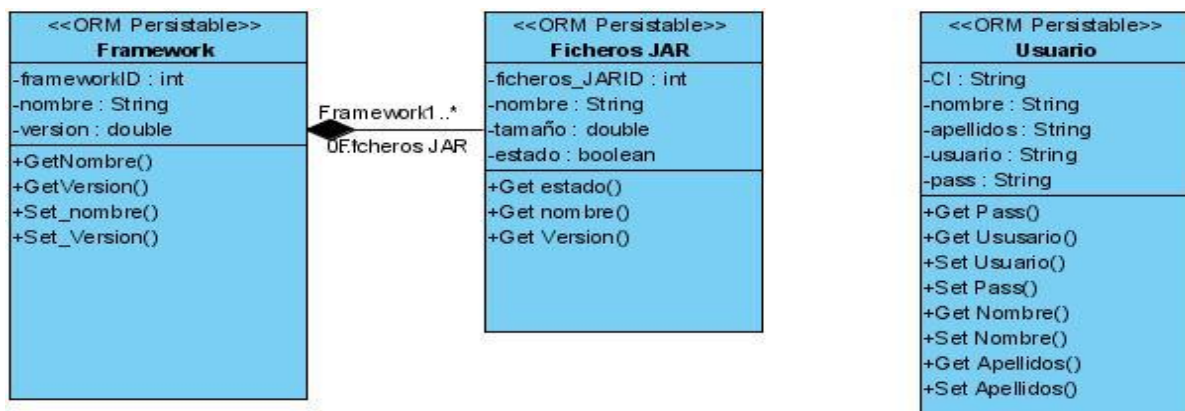


Figura 37: Diagrama de clases persistentes



3.4.2 MODELO ENTIDAD RELACIÓN

A partir del diagrama de clases persistentes se obtuvo el siguiente Modelo Entidad Relación.

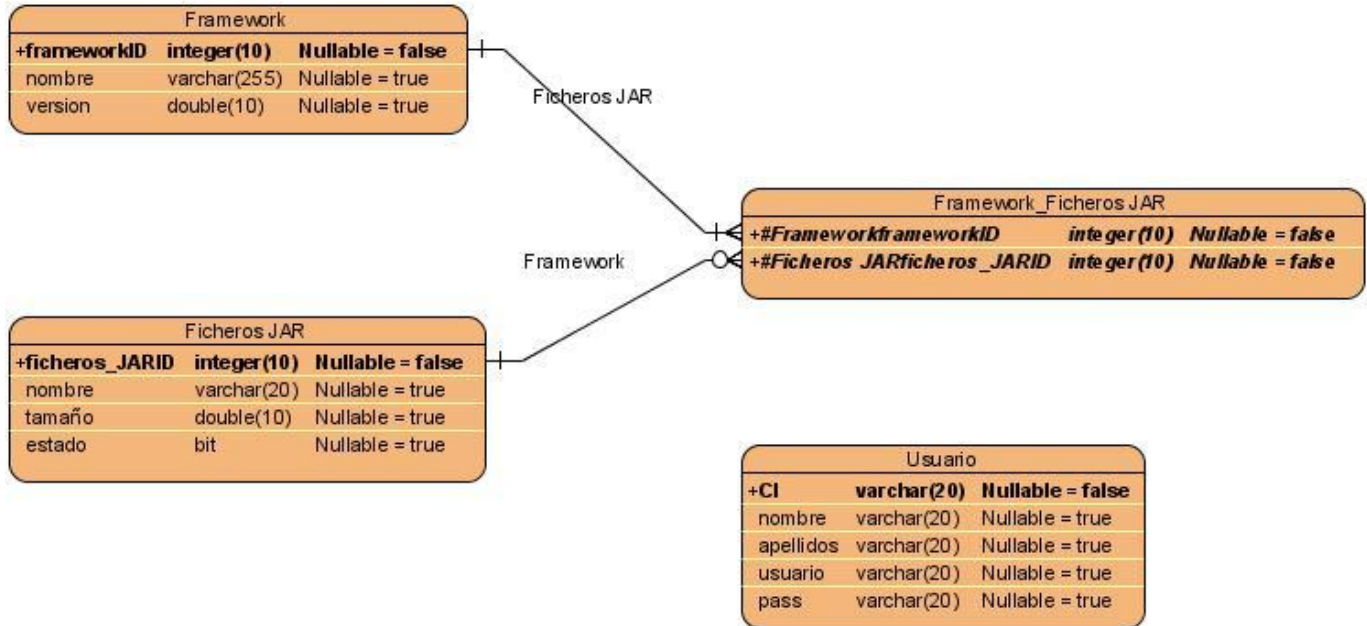


Figura 38: Modelo Entidad Relación

3.4.3 DESCRIPCIÓN DE LAS TABLAS

Nombre: tb_Frameworks		
Descripción: En esta tabla se almacenan los datos correspondientes a los frameworks.		
Atributo	Tipo	Descripción
frameworkID	Int	Número que identifica al framework.
nombre	Varchar	Nombre del framework.
versión	Double	Versión del framework.

Tabla 20: Descripción de la tabla Framework



Nombre: tb_Ficheros JAR		
Descripción: En esta tabla se almacenan los datos correspondientes a los frameworks.		
Atributo	Tipo	Descripción
Fichero_JARID	Int	Número que identifica al fichero.
nombre	Varchar	Nombre del fichero.
tamaño	Double	Tamaño del fichero expresado en MB.
estado	Boolean	Estado del fichero (Asignado / No asignado).

Tabla 21: Descripción de la tabla Fichero JAR

Nombre: tb_Usuarios		
Descripción: En esta tabla se almacenan los datos correspondientes a los frameworks.		
Atributo	Tipo	Descripción
CI	Varchar	Carné de Identidad que identifica al usuario.
nombre	Varchar	Nombre del usuario.
apellidos	Varchar	Apellidos del usuario.
usuario	Varchar	Usuario para ingresar al sistema.
pass	Varchar	Contraseña para acceder al sistema.

Tabla 22: Descripción de la tabla Usuario



3.5 CONCLUSIONES

En este capítulo se incorporaron los términos referentes al flujo de trabajo de Análisis y Diseño definido por la metodología que durante todo el desarrollo se ha utilizado. Se introdujeron los diferentes diagramas que el mismo sugiere para su correcta elaboración, como son, los diagramas de clases del análisis y los diagramas de clases del diseño, ambos confeccionados para cada caso de uso presente en el capítulo anterior. Además se modelaron los diagramas de interacción tanto de colaboración como de secuencia, centrando su construcción para cada escenario de cada caso de uso, permitiendo así, evidenciar el camino básico a seguir en la implementación. Se agrega igualmente el diseño de la base de datos con sus tablas, atributos, funciones, que conjuntamente con el Diagrama de Clases Persistentes dan lugar al Modelo Entidad Relación, patentizando así la descripción de las diferentes tablas que existen, en las cuales se centra el principio funcional de almacenamiento y control, así como la administración de la información que pertenece a la base de datos definida.



CAPÍTULO 4: IMPLEMENTACIÓN

4.1 INTRODUCCIÓN

En el presente capítulo se pormenoriza y exterioriza el modelo de implementación propuesto. Este se describe mediante los diagramas de componentes y despliegue, los cuales darán cumplimiento al objetivo fundamental de este capítulo. Además se evidencia la transformación de los elementos del diseño en elementos de implementación. También se describe cómo los elementos del modelo del diseño se implementan en términos de componentes, lo que muestra la representación lógica de los mismos y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue.

4.2 DIAGRAMA DE DESPLIEGUE

El diagrama de despliegue representa las relaciones físicas correspondientes a software - hardware que tendrá el sistema final, o sea, las relaciones existentes entre los diferentes nodos de que componen a un sistema, además incluye las relaciones que pueden existir entre los diferentes componentes de un nodo. Este modelo responde directamente a la confección de la vista de despliegue, la que representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación.

Es posible afirmar que un nodo es un recurso de ejecución que puede ser una computadora, un servidor y una impresora, los que se pueden clasificar en dispositivos o procesadores.

En la aplicación que se desarrolla el diagrama de despliegue constará con tres nodos los que se especifican en: un servidor Web donde se montará la aplicación, el cual estará conectado al servidor de base de datos PostgreSQL mediante el protocolo TCP/IP para realizar consultas y las actualizaciones necesarias de la información almacenada, a su vez, se vincularía a través del protocolo http con un tercer nodo que representaría las computadoras mediante las cuales los usuarios, ya sean usuarios generales o administradores, acceden a la aplicación.

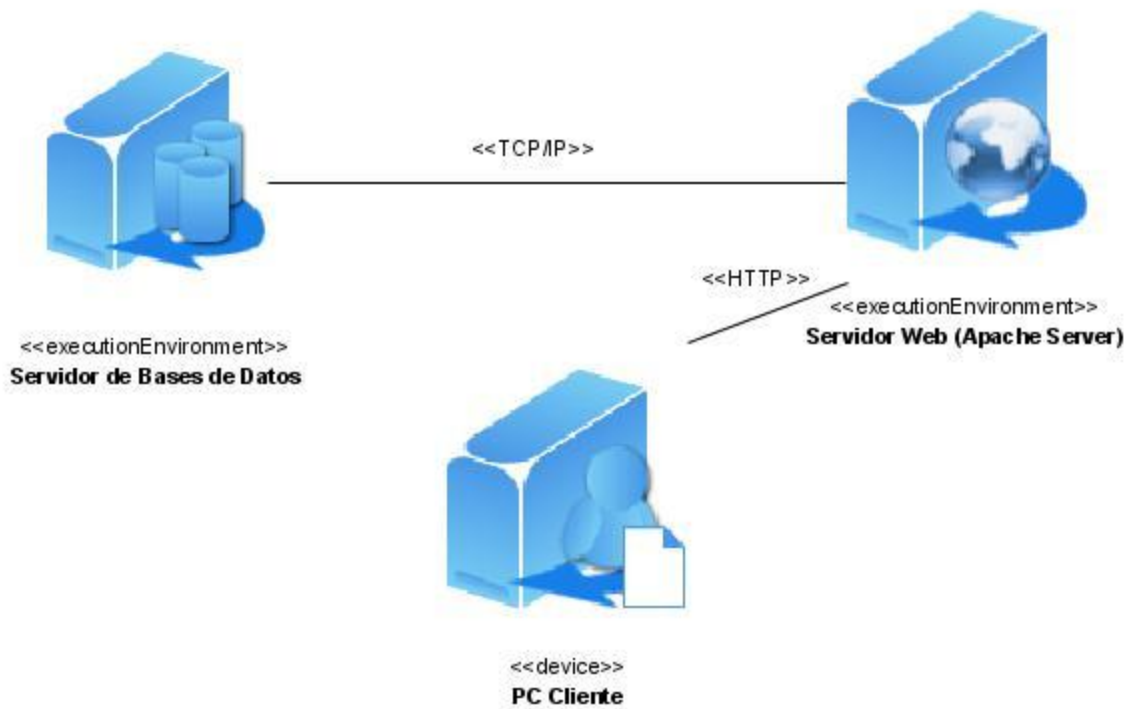


Figura 39: Diagrama de Despliegue

4.3 DIAGRAMA DE COMPONENTES

El diagrama de componentes concibe un conjunto de elementos tales como, componentes, paquetes, subsistemas de implementación y sus relaciones. Este diagrama se utiliza para representar la vista estática de cualquier sistema. Facilita el entendimiento del modelo de implementación, ya que en él se presentan los componentes lógicos de la aplicación así como las relaciones de dependencias que existen entre ellos ya sean componentes de código fuente, librerías, binarios o ejecutables.

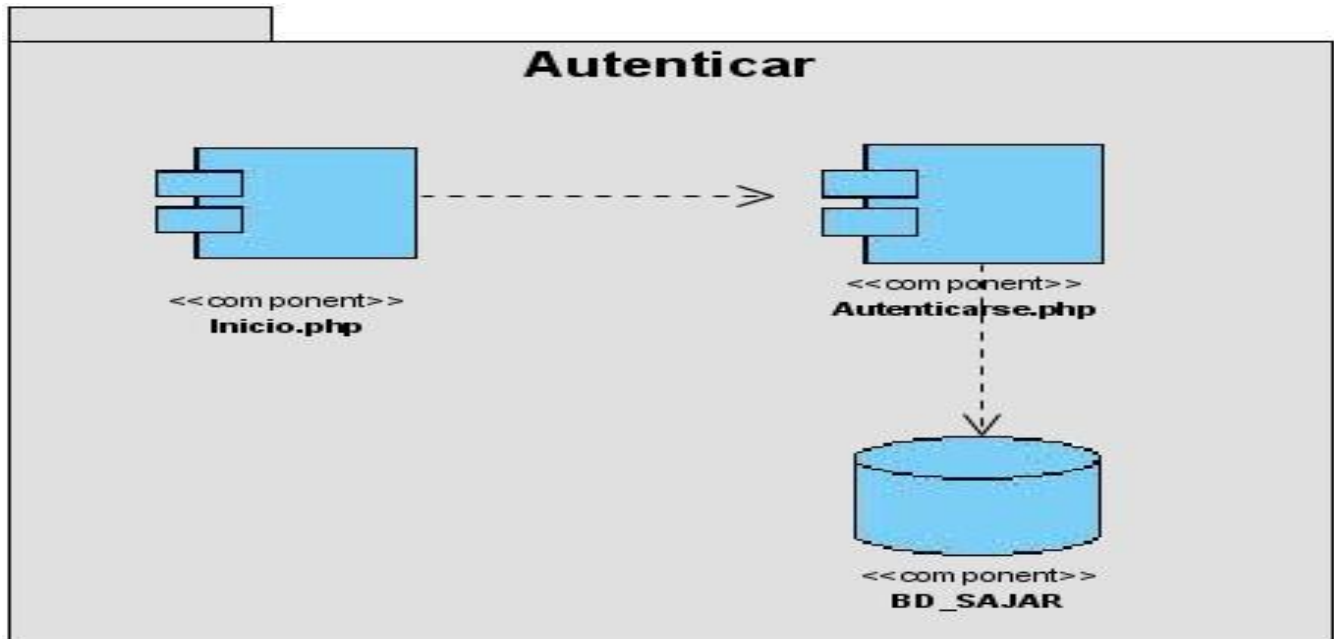


Figura 40: Diagrama de componentes Autenticarse

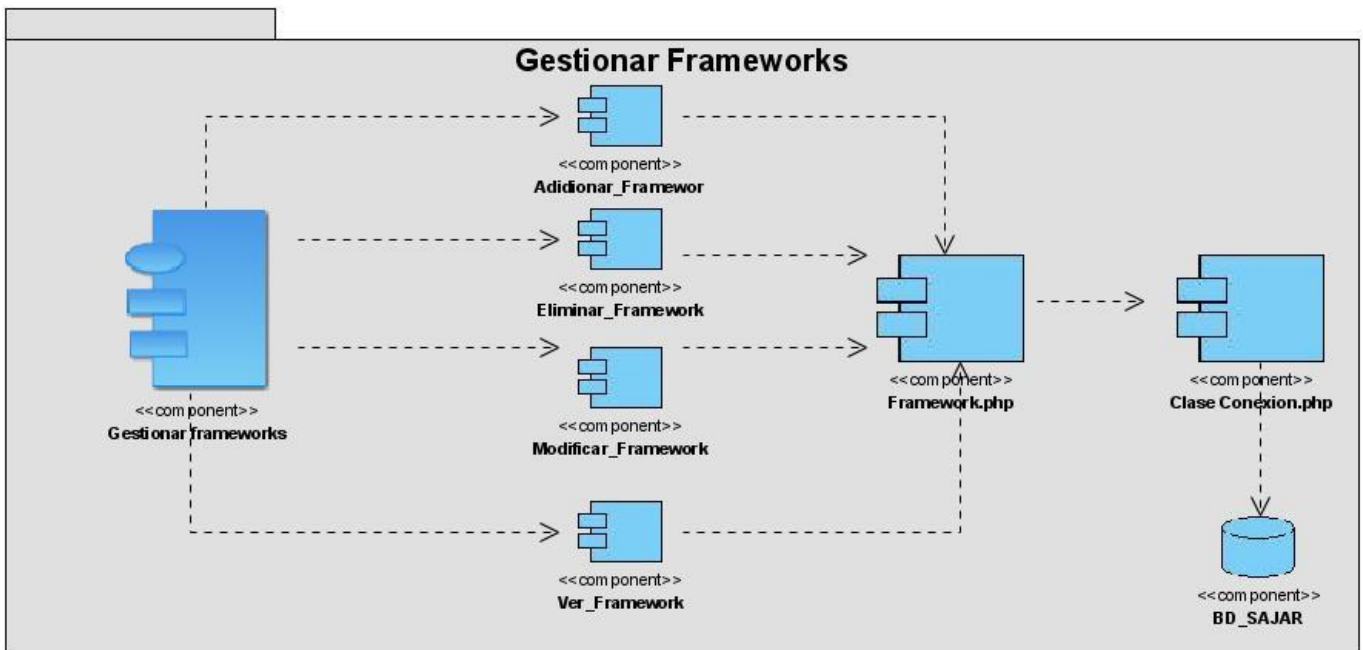


Figura 41: Diagrama de componentes Paquete: Gestionar Frameworks

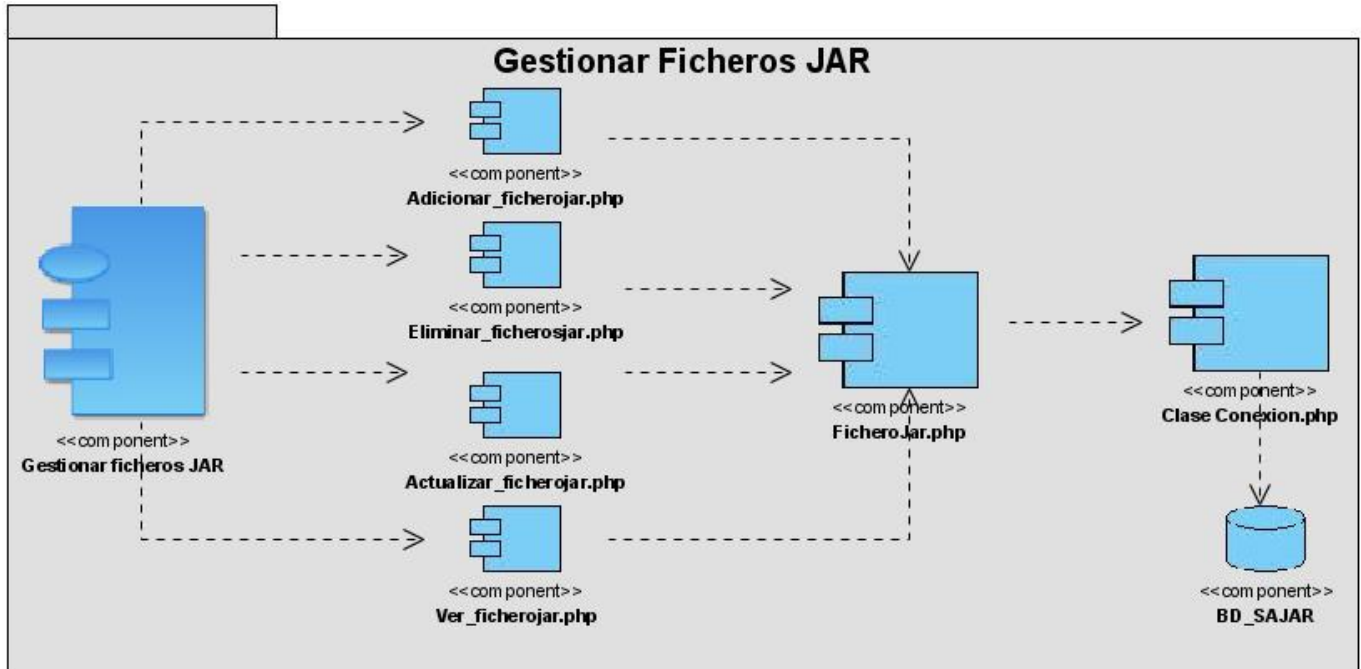


Figura 42: Diagrama de componentes Paquete: Gestionar Ficheros JAR

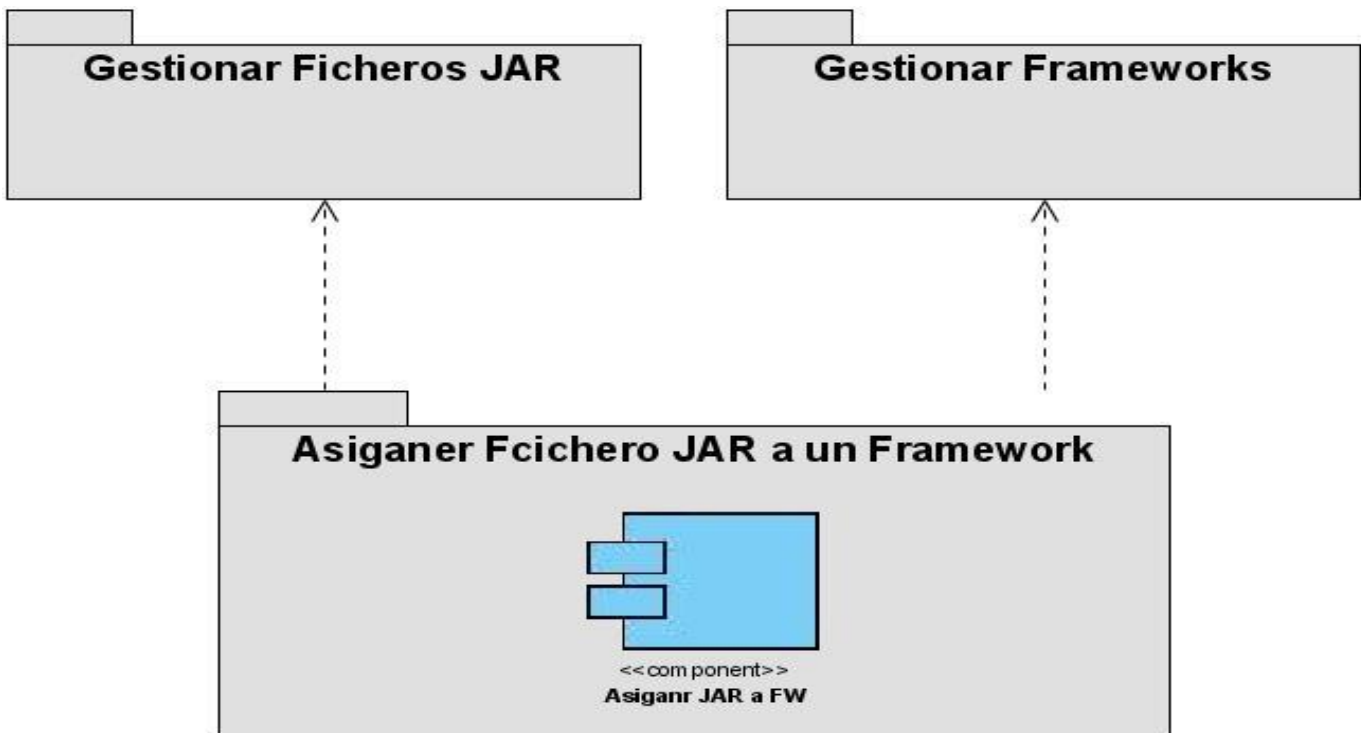


Figura 43: Diagrama de componentes Paquete: Asignar Fichero JAR a un Framework

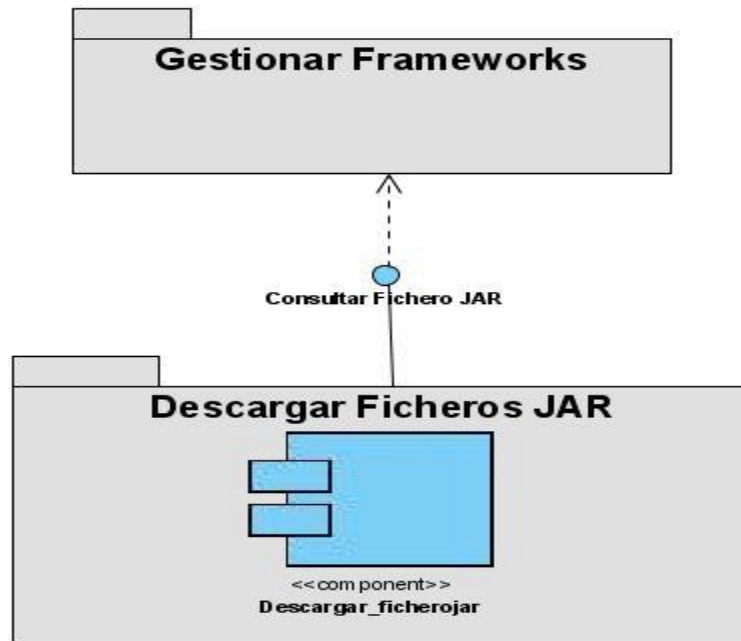


Figura 44: Diagrama de componentes Paquete: Descargar Framework

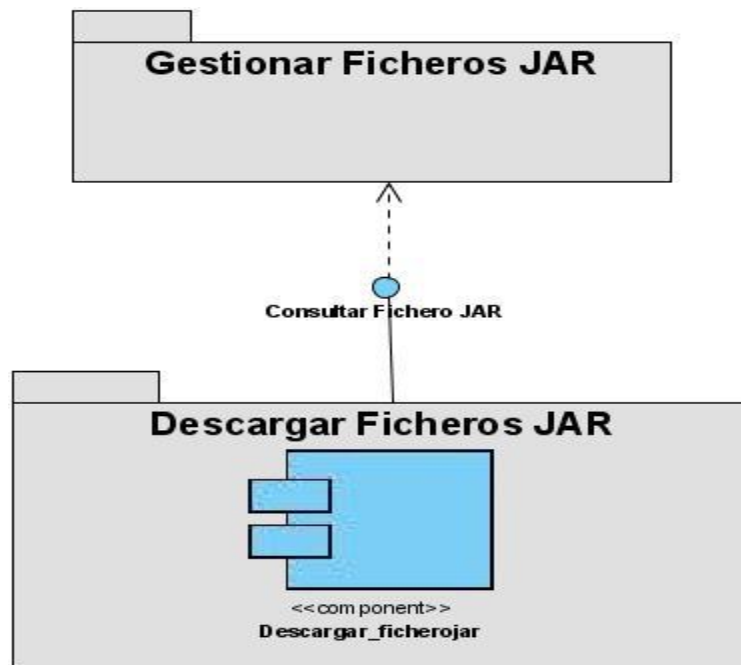


Figure 45: Diagrama de componentes Paquete: Descargar Ficheros JAR



4.4 CONCLUSIONES

Para poder detallar descriptivamente la solución propuesta, se realizaron en este capítulo los diagramas de despliegue y componentes, quedando confeccionada la vista estática del sistema. Todo ello da muestra tanto de la distribución física como la representación lógica de los elementos del diseño, comportándose estos últimos como componentes de un modelo que llevan implícito las relaciones de dependencias que existe entre ellos, los paquetes y los subsistemas.



CONCLUSIONES

Una vez realizada la investigación y concluido el desarrollo de la Aplicación Web para la gestión de la información de los archivos de java, tanto frameworks como ficheros JAR y cumplidos los objetivos planteados así como las tareas de investigación, se puede proyectar los resultados alcanzados:

- ✓ Se realizó el estudio detallado de los procesos llevados a cabo por los proyectos productivos de la Universidad que utilizan JAVA como lenguaje de programación.
- ✓ Se puntualizaron los diferentes frameworks utilizados en los proyectos pertenecientes al polo de gestión de software y que por ende utilizan java como lenguaje de programación y se seleccionaron los frameworks transitados de forma general.
- ✓ Se compendiaron las herramientas y tecnologías necesarias a utilizar para un buen modelado y desarrollo de la aplicación Web.
- ✓ Se elaboró el documento correspondiente al análisis, diseño e implementación de la aplicación Web.
- ✓ Se desarrolló un portal Web que garantiza la gestión de la información de los archivos de JAVA.



RECOMENDACIONES

Aún con el cumplimiento de los objetivos y de los requisitos fundamentales de la aplicación, se necesita incorporar un número de recomendaciones que se consideran fundamentales para la utilización de la misma debido al próspero, necesario y fundamental avance tecnológico, que engloban a la Informática como ciencia, conjuntamente con su plano ingenieril:

- ✓ Mantener actualizada la aplicación con las versiones de framework y ficheros JAR pertenecientes a JAVA como lenguaje de programación.
- ✓ Utilizar la información que se gestiona y rehacer la aplicación trabajando no solo con la información sino sobre ella.
- ✓ Desplegar la aplicación en el polo de software de gestión vinculando a todos los proyectos productivos que utilizan JAVA como lenguaje de programación.
- ✓ Incorporar un servicio de envío de boletines y foro debate.
- ✓ Trabajar para mejorar el diseño de la aplicación creada.



REFERENCIAS BIBLIOGRÁFICAS

1. MARCUS EDUARDO MARKIEWICZ, C. J. P. D. L. [Consultado el: 6 de Marzo de 2009]. Disponible en: <http://www.acm.org/crossroads/espanol/xrds7-4/frameworks.html>
2. [Consultado el: 2 de Febrero de 2009]. Disponible en: <http://www.docjar.com/>.
3. [Consultado el: 2 de Febrero de 2009]. Disponible en: <http://www.docjar.com/>.
4. WEB 2.0 DEVELOPMENT [Consultado el: 5 de Febrero de 2009]. Disponible en: <http://web2development.blogspot.com/2007/05/patron-mvc.html>.
5. VALDÉS, D. P. MAESTROS DEL WEB 2 de Noviembre de 2007, [Consultado el: 5 de Febrero de 2009]. Disponible en: <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programación-para-la-web>.
6. PHP [Consultado el: 4 de Febrero de 2009]. Disponible en: <http://www.php.net/>.
7. [Consultado el: 5 de Febrero de 2009]. Disponible en: <http://nereida.deioc.ull.es/~cleon/doctorado/doc05/0405/pspdf/LenguajesMarca.pdf>.
8. MARK BAKER, M. I., SHINICHI MATSUI, PETER STARK, TED WUGOFSKI, TOSHIHIKO YAMAKAMI. [Consultado el: 5 de Febrero de 2009]. Disponible en: <http://www.sidar.org/recur/desdi/traduc/es/xhtmll/xhtmll-basic.html>.
9. Desarrolladores Web 4 de Junio 2003 [Consultado el: 4 de Febrero de 2009]. Disponible en: <http://www.desarrolloweb.com/articulos/1178.php>.
10. PRESSMAN, R. S. Ingeniería del Software. Un enfoque práctico. 2005.
11. IVAR JACOBSON, G. B., JAMES RUMBAUGH El Lenguaje Unificado de Modelado. 2000.
12. Cyta.com [Consultado el: 4 de Febrero de 2009]. Disponible en: <http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm>.
13. SICILIA, M. Á. Conexión 23 septiembre 2008, [Consultado: 1 Abril de 2009]. Disponible en: <http://cnx.org/content/m17423/latest/>.



BIBLIOGRAFÍA

Ruz, Fidel Castro. 2006. Intervención en la Mesa Redonda: Un millón de Graduados en los Joven Club. 23 de Junio de 2006.

2009. 2X3. [En línea] 2009. [Citado el: 6 de Enero de 2009.]

<http://www.2x3.cu>

22. AVANTCAD BLOG, [En línea] Junio 2007, 22. [Citado: Enero 30, 2009.]

<http://avantcad.blogspot.com>

[Citado: Febrero 4, 2009.]

<http://www.mastermagazine.info/termino/4294.php>

ANALIA LANZILLOTA 2004.

2007. ASAYC, A. I. S.A. DE C.V., [En línea] 2007. [Citado el: 4 de Febrero de 2009.]

<http://www.asayc.com/administracion/cliente.htm>

2000. VISTA - BUENOS AIRES - ARGENTINA , 2000. [Citado el: 4 de Febrero de 2009.]

<http://www.netwareinformtaica.com.ar/index.asp>

2007. RODRIGO CORRAL, [En línea] 15 de Enero de 2007. [Citado el: 6 de Febreo de 2009.]

<http://geeks.ms/blogs/rcorral/archive/2007/01/15/iquest-que-metodolog-iacute-ade-desarrollo-elegir.aspx>

IVON TARAZONA [Citado el: 6 de Febrero de 2009.]

<http://alfa.facyt.uc.edu.ve/computacion/pensum/cs0347/download/exposiciones2005-2006/uml.pdf>

ORIANA GOMEZ [Citado el: 6 de Febrero de 2009.]

<http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm>

[Citado el: 7 de Febrero de 2009.]

<http://www.eafit.edu.co/nr/rdonlyres/e12d52f4-e868-4c79-98c2-b917eada7892/0/frameworks.pdf>



2007. BULBA. [En línea] BENVINGUT A LA PLANA WEB DEL DEPARTAMENT DE MATEMÀTIQUES I INFORMÀTICA, UNIVERSITAT DE LES ILLES BALEARS, 5 de Mayo de 2007. [Citado el: 5 de Febrero de 2009.]

<http://www.bulba.net>

2008. DOSIDEAS . [En línea] 2008. [Citado el: 6 de Febrero de 2009.]

<http://www.dosideas.com>

CREATIVE COMMON

2003. EJEMPLO DE DESARROLLO SOFTWARE UTILIZANDO LA METODOLOGÍA RUP .

[En línea] CÉSAR LÓPEZ RODRÍGUEZ , 28 de Julio de 2003. [Citado el: 6 de Febrero de 2009.]

<http://www.dsic.upv.es/asignaturas/facultad/lsi/ejemplorup>

EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE . [En línea] [Citado el: 6 de Febrero de 2009.]

<http://yaqui.mx.l.uabc.mx/~molquin/as/rup.htm>

1998. EL RINCÓN DE LINUX PARA HISPANOHABLANTES. [En línea] RAFAEL MARTINEZ, 1998.

[Citado el: 5 de Febrero de 2009.]

<http://www.linus-es.org/taxonomy/term/43/0>

2001. FUNCIONES DE LOS SISTEMAS DE GESTIÓN DE BASES DE DATOS . [En línea] MARÍA

MERCEDES MARQUÉS ANDRÉS , 12 de Febrero de 2001. [Citado el: 5 de Febrero de 2009.]

<http://www3.uji.es/~mmarques/f47/apun/node39.html>

HERRAMIENTAS WEB PARA LA ENSEÑANZA DE PROTOCOLOS DE COMUNICACIÓN. [En línea]

MASTERMAGAZIN. [Citado el: 4 de Febrero de 2009.]

<http://neo.lcc.uma.es/evirtual/cdd/tutorial/aplicacion/cliente-servidor.html>

2002. JAVA HISPANO . [En línea] ASOCIACIÓN JAVAHISPANO, 2002. [Citado el: 5 de Febrero de 2009.]



<http://www.javahispano.org/contenidos/es/mysqlvspostgresql>

2009. LA TECLA DEL ESCAPE . [En línea] 25 de Enero de 2009. [Citado el: 6 de Febrero de 2009.]

<http://latecladeescape.com/w0/ingeniería-del-software/metodologías-de-desarrollo-del-software.html>

2009. MARBLE STATION. [En línea] SERGI BLANCO CUARESMA, 2009. [Citado el: 6 de Febrero de 2009.]

<http://www.marblestation.com>

2008. MYSQL . [En línea] SUN MICROSYSTEMS, 2008. [Citado el: 5 de Febrero de 2009.]

<http://dev.mysql.com/doc/refman/5.0/es/features.html>

2002. POCESO DE DESARROLLO RUP, XP, FDD . [En línea] ALBERTO MOLPECERES, 15 de Diciembre de 2002. [Citado el: 6 de Febrero de 2009.]

<http://www.willydev.net/descargas/articulos/general/cualxpdfddrup.pdf>

2008. POSTGRESQL. [En línea] POSTGRESQL GLOBAL DEVELOPMENT GROUP , 2008. [Citado el: 5 de Febrero de 2009.]

<http://archives.postgresql.org/pgsql-es-ayuda/2006-05/msg00932.php>

2004. SISTEMAS DE GESTIÓN DE BASES DE DATOS . [En línea] INSTITUTO GEOGRÁFICO AGUSTÍN CODAZZI, Agosto de 2004. [Citado el: 5 de Febrero de 2009.]

<http://www.igac.gov.co:8080/igacweb/userfiles/file/ciaf/tutorialsig-2005-26-02/paginas/ctr-sistemasdegestiondedatos.htm>

2008. SLIDESHARE. [En línea] SLIDESHARE INC. ALL RIGHTS RESERVED, 2008.

<http://www.slideshare.net/decimo/arquitectura-3-capas>

2003. TIENDALINUX.COM . . [En línea] 31 de Mayo de 2003. [Citado el: 5 de Febrero de 2009.]

<http://soporte.tiendalinux.com/portal/portfolio/postgresql-ventajas.html>

TUTORIAL DE POSTGRESQL. [En línea] POSTGRES GLOBAL DEVELOPMENT GROUP. [Citado el: 5 de Febrero de 2009.]



<http://lucas.hispalinux.es/postgresql-es/web/navegable/tutorial/x1494.html>

2008. WEB DOCENTE DEPARTAMENTAL . [En línea] 2008. [Citado el: 5 de Febrero de 2009.]

<http://www.deciencias.net/disenoweb/elaborardw/páginas/introhtml.html>

libroweb.es [Consultado el: 5 de Marzo de 2009]. Disponible en:

http://www.librosweb.es/symfony/capitulo1/symfony_en_pocas_palabras.html.

JAVA [Consultado el: 5 de Marzo de 2009]. Disponible en:

<http://arechiga.50megs.com/tpoo2/2dept/informacionjava2.html>.

symfony.es [Consultado el: 5 de Marzo de 2009]. Disponible en:

<http://www.symfony.es/2009/01/29/comienza-el-curso-de-symfony-en-vitoria-gasteiz/>.

ALDANA, J. frogx.three [Consultado el: 5 de Marzo de 2009]. Disponible en:

<http://www.frogx3.com/2007/12/19/symfony-vs-zend-framework/>.

JAVIER BLANCO, J. L. I., GERARDO CUERO. Apache Web Server. instalación y Configuración [Consultado el: 4 de Marzo de 2009]. Disponible en:

<http://www ldc.usb.ve/~emilio/Portafolio/Exposiciones/Apache.ppt#261,6,Apache Web Server>.

MARAÑÓN, G. Á. [Consultado el: 5 de Marzo de 2009]. Disponible en:

<http://www.iec.csic.es/CRIPTONOMICON/java/quesjava.html>.

ROJAS, J. C. V. SISTEMAS DISTRIBUIDOS EN JAVA [Consultado el: 5 de Marzo de 2009]. Disponible en: http://www.unsj-cuim.edu.ar/portalezonda/congreso/papers/1999/Java_RMI.html.

WORDPRESS. About Karl Katzke [Consultado el: 5 de Marzo de 2009]. Disponible en:

<http://www.karlkatzke.com/about/>.



GLOSARIO DE TÉRMINOS

Acceder: Acceder a la información o datos en un sistema informático.

Artefacto: Es una de las múltiples formas de resultados tangibles producidos durante el desarrollo de software.

Copyright: Es un conjunto de normas y principios que regulan los derechos morales y patrimoniales que la ley concede a los autores por el solo hecho de la creación de una obra literaria, artística o científica, esté publicada o inédita.

CSS: Hojas de Estilo en Cascada (Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos.

DOM (Document Object Model): Es una interfaz de programación de aplicaciones que proporciona un conjunto estándar de objetos para representar documentos HTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos.

Ficheros JAR: contiene los ficheros de clases y los recursos auxiliares asociados con los applets y aplicaciones. Estos recursos auxiliares podrían incluir, por ejemplo, ficheros de imagen y sonido que sean utilizados por un applet.

Gestión: Implica un conjunto de trámites que se lleva a cabo para resolver un asunto o concretar un proyecto.

Gestión de información: Proceso que incluye operaciones como extracción, manipulación, tratamiento, depuración, conservación, acceso y/o colaboración de la información adquirida por una organización a través de diferentes fuentes y que gestiona el acceso y los derechos de los usuarios sobre la misma.

Hipertexto: Nombre que recibe el texto que en la pantalla de una computadora te conduce a otro texto relacionado.

Ingeniería de software: Disciplina o área de la informática que ofrece métodos y técnicas para desarrollar y mantener software de calidad.



Interfaz gráfica de usuario: Es el artefacto tecnológico de un sistema interactivo que posibilita, a través del uso y la representación del lenguaje visual, una interacción amigable con un sistema informático.

JavaScript: Lenguaje de script multiplataforma orientado a objetos, pequeño y ligero; no es útil como un lenguaje independiente, más bien está diseñado para una fácil incrustación en otros productos y aplicaciones, tales como los navegadores Web.

Librerías: conjunto de subprogramas utilizados para desarrollar software.

Modularidad: Término derivado de la programación y referido al hecho de que los distintos componentes de un programa deben ser lo más independientes entre sí, con lo cual cualquier falla puede atribuirse a uno de ellos en particular.

Multihilo: Característica que permite a una aplicación realizar varias tareas concurrentemente.

Multiplataforma: Término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas.

Multiusuario: Característica de un sistema operativo o programa que permite proveer servicio y procesamiento a múltiples usuarios simultáneamente (tanto en paralelismo real como simulado).

Página Web: Documento multimedia generalmente en formato HTML, que contiene enlaces hacia otros documentos. Se accede a este documento a través de un servidor web, gracias a una dirección única (URL) y puede ser visualizado desde un navegador.

Plataformas: Es una combinación de hardware y software usada para ejecutar aplicaciones; en su forma más simple consiste únicamente de un sistema operativo, una arquitectura, o una combinación de ambos.

Programación Orientada a Objetos (POO): Es un paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones y programas de computadora.

Sistema operativo: Software de sistema o conjunto de programas de computación destinados a realizar muchas tareas entre las que destaca la administración eficaz de sus recursos.

Sitio web: Conjunto de páginas Web, típicamente comunes a un dominio o subdominio en la World Wide Web en Internet.



Software: Conjunto de programas elaborados por el hombre, que controlan la actuación del computador, haciendo que éste siga en sus acciones una serie de esquemas lógicos predeterminados y pueda desempeñar tareas inteligentes. Suele sustituirse por expresiones tales como programas (informáticos) o aplicaciones (informáticas).

Software libre: Según la Free Software Foundation, el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.

Tecnología: Conjunto de habilidades que permiten construir objetos y máquinas para adaptar el medio y satisfacer nuestras necesidades.

XML(Extensible Markup Language): Lenguaje de marca considerado como un metalenguaje extensible de etiquetas



ANEXOS

ANEXO1: DIAGRAMA DE COLABORACIÓN CU_AUTENTICAR USUARIO

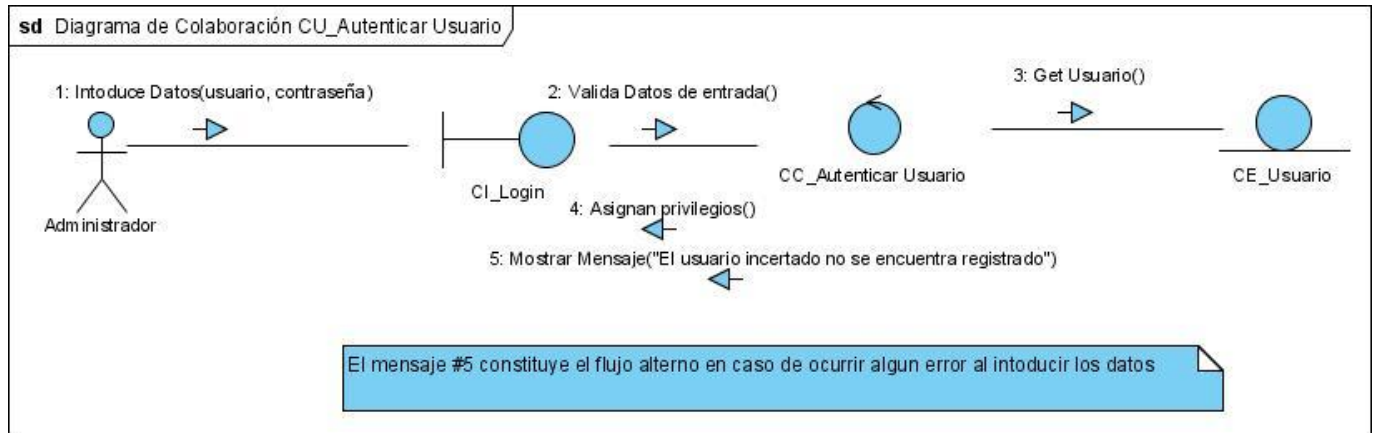


Figura 46: Diagrama de colaboración del CU _ Autenticar Usuario

ANEXO2: DIAGRAMA DE COLABORACIÓN CU_CONSULTAR FRAMEWORKS

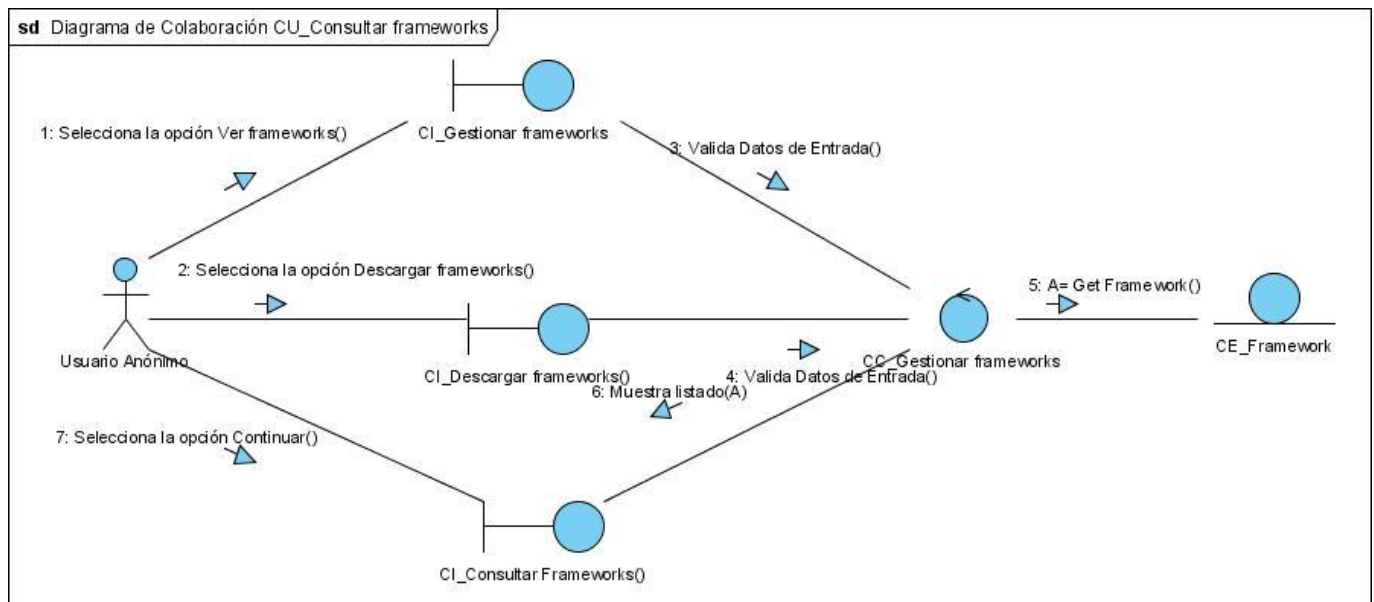


Figura 47: Diagrama de colaboración del CU _ Consultar Frameworks



ANEXO3: DIAGRAMA DE COLABORACIÓN CU_CONSULTAR FICHEROS JAR

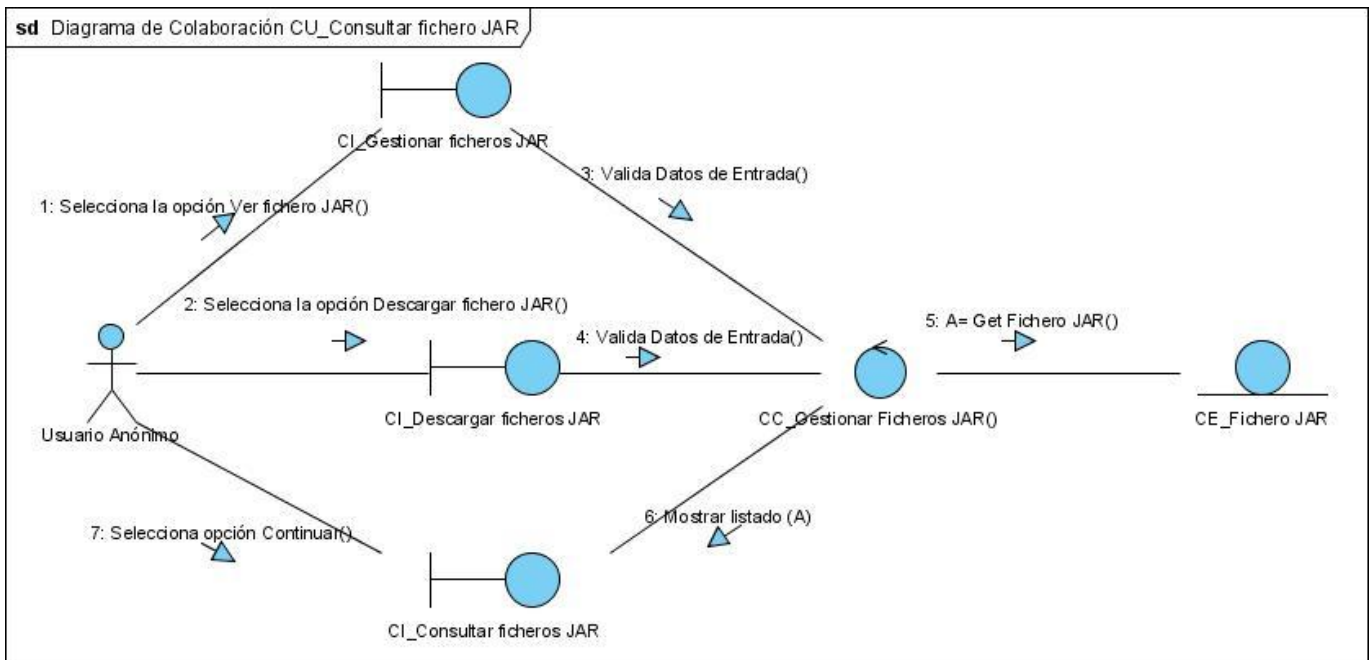


Figura 48: Diagrama de colaboración del CU _ Consultar Fichero JAR

ANEXO 4: DIAGRAMA DE COLABORACIÓN CU_DESCARGAR FRAMEWORKS

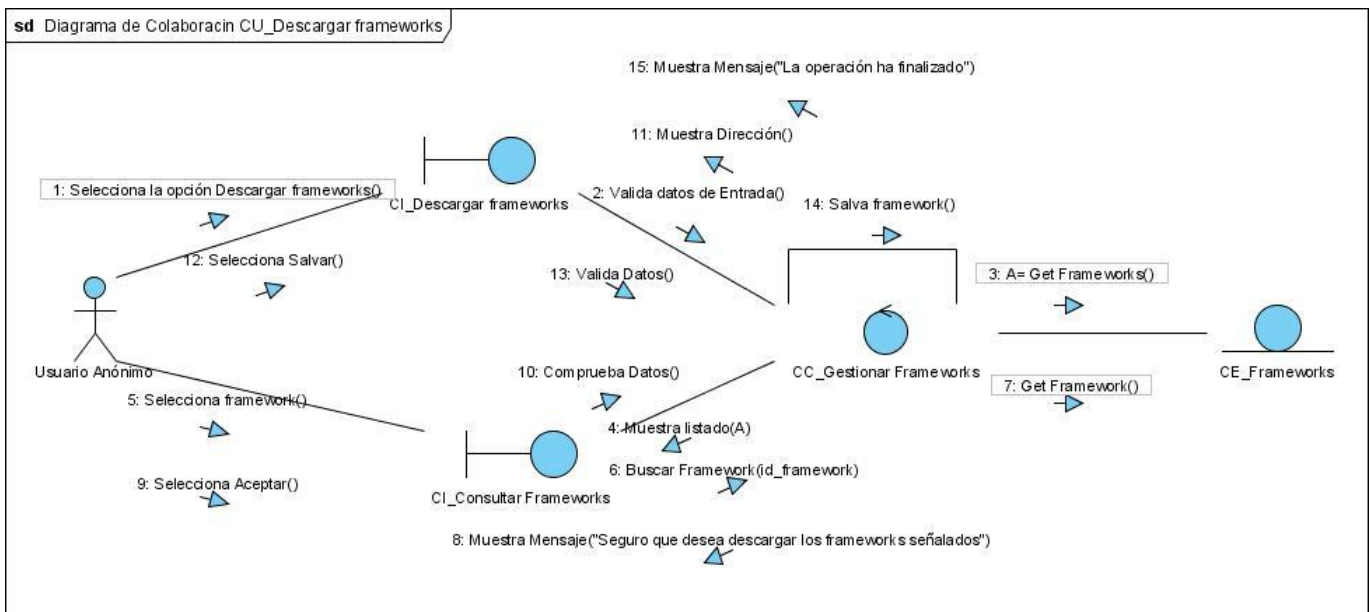


Figura 49: Diagrama de colaboración del CU_Descargar Frameworks



ANEXO 5: DIAGRAMA DE COLABORACIÓN CU_DESCARGAR FICHEROS JAR

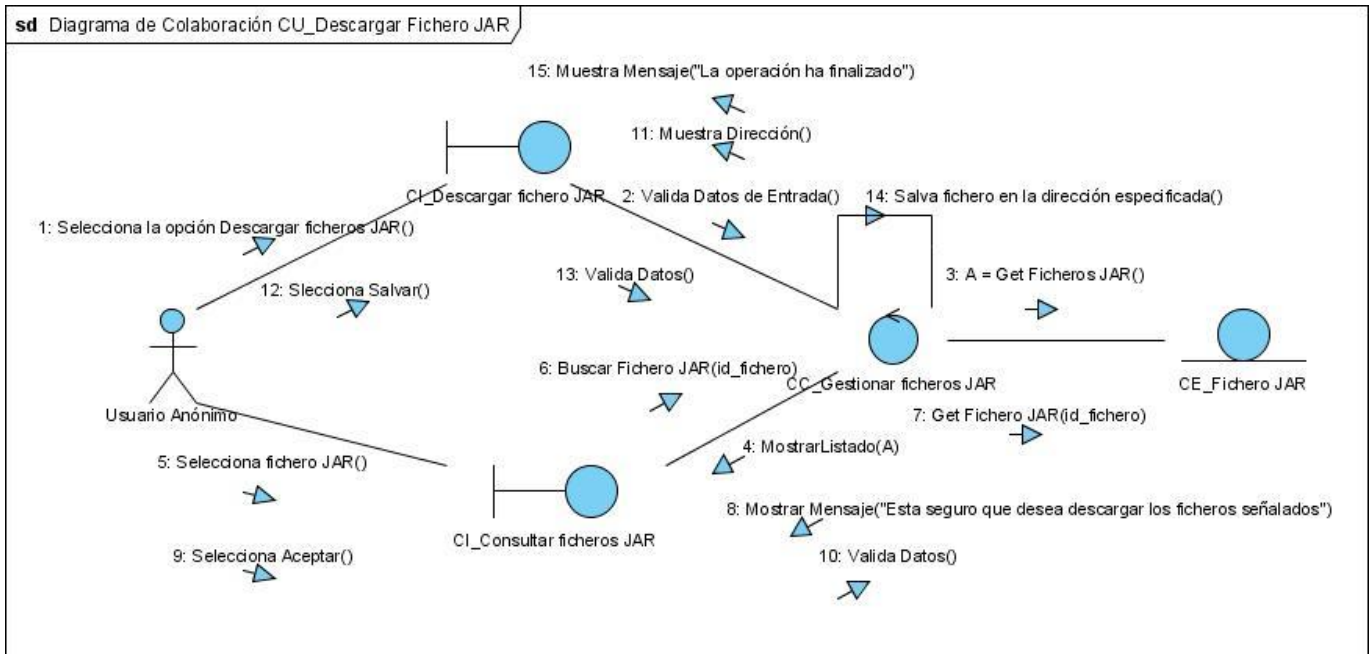


Figura 50: Diagrama de colaboración del CU_Descargar Ficheros JAR

ANEXO 6: DIAGRAMA DE SECUENCIA CU_AUTENTICAR USUARIO

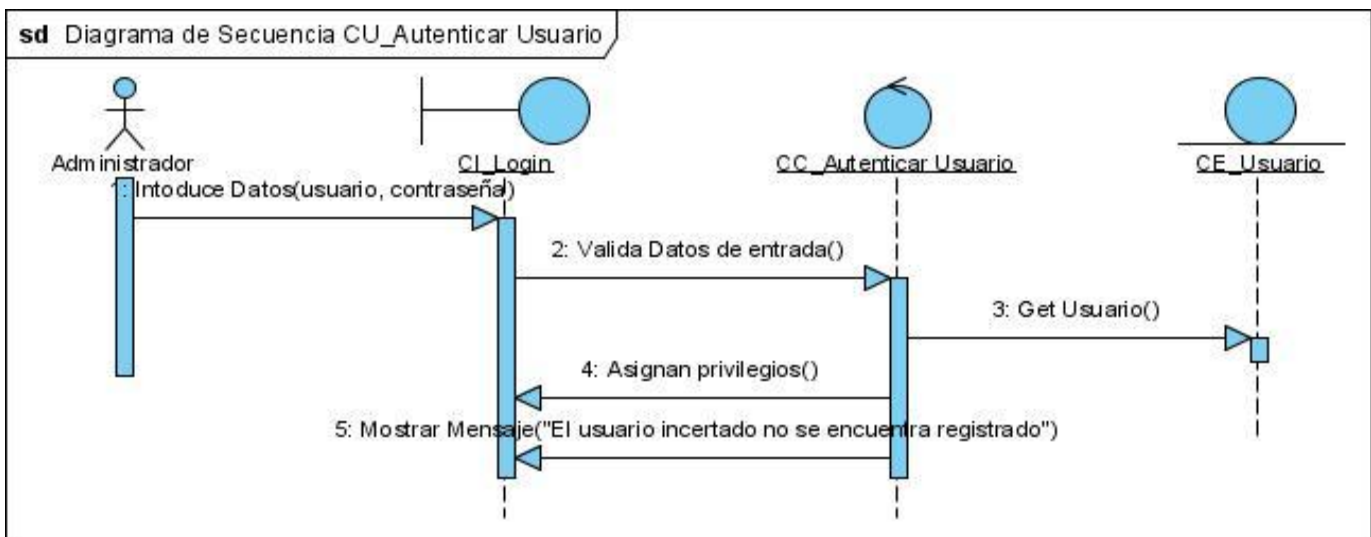


Figura 51: Diagrama de secuencia del CU_Autenticar Usuario



ANEXO 7: DIAGRAMA DE SECUENCIA CU_CONSULTAR FRAMEWORKS

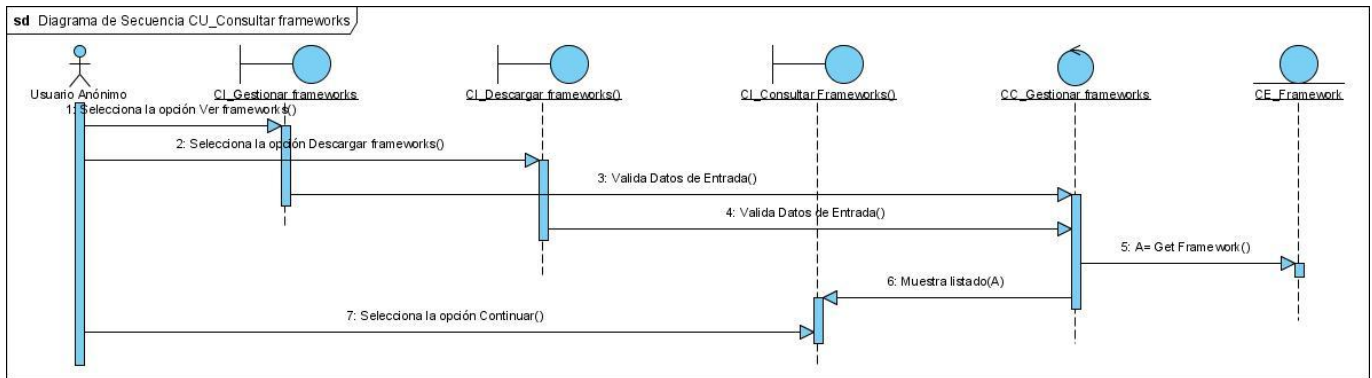


Figura 52: Diagrama de secuencia del CU _ Consultar Frameworks

ANEXO 8: DIAGRAMA DE SECUENCIA CU_CONSULTAR FICHEROS JAR

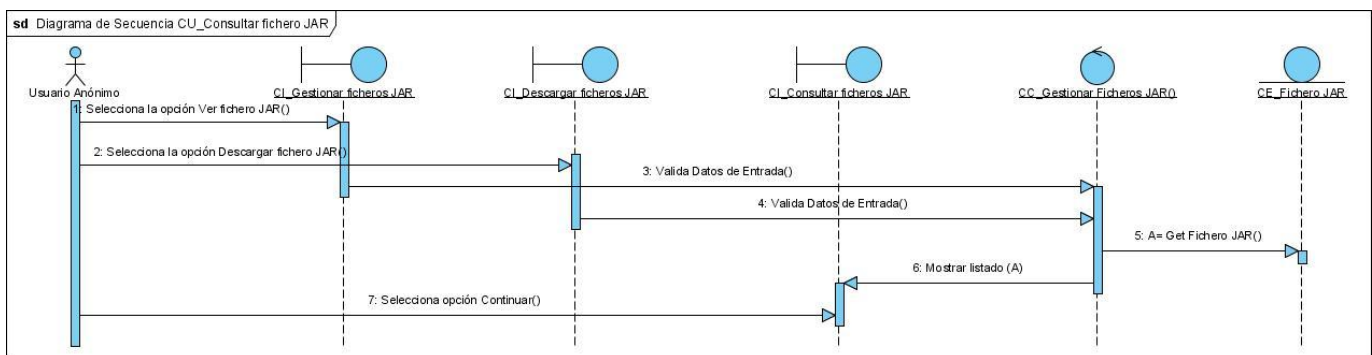


Figura 53: Diagrama de secuencia del CU _ Consultar Ficheros JAR



ANEXO 9: DIAGRAMA SE SECUENCIA CU_DESCARGAR FRAMEWORKS

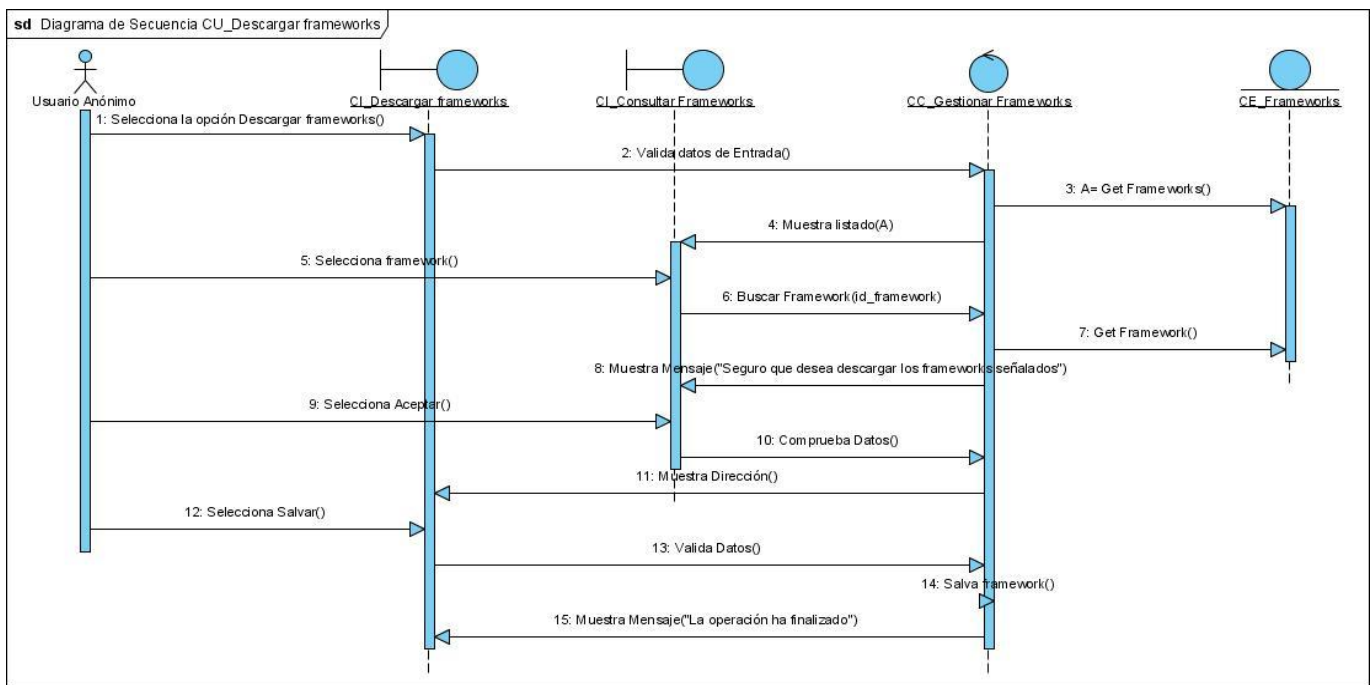


Figura 54: Diagrama de secuencia del CU_Descargar Frameworks



ANEXO 10: DIAGRAMA DE SECUENCIA CU_DESCARGAR FICHERO JAR

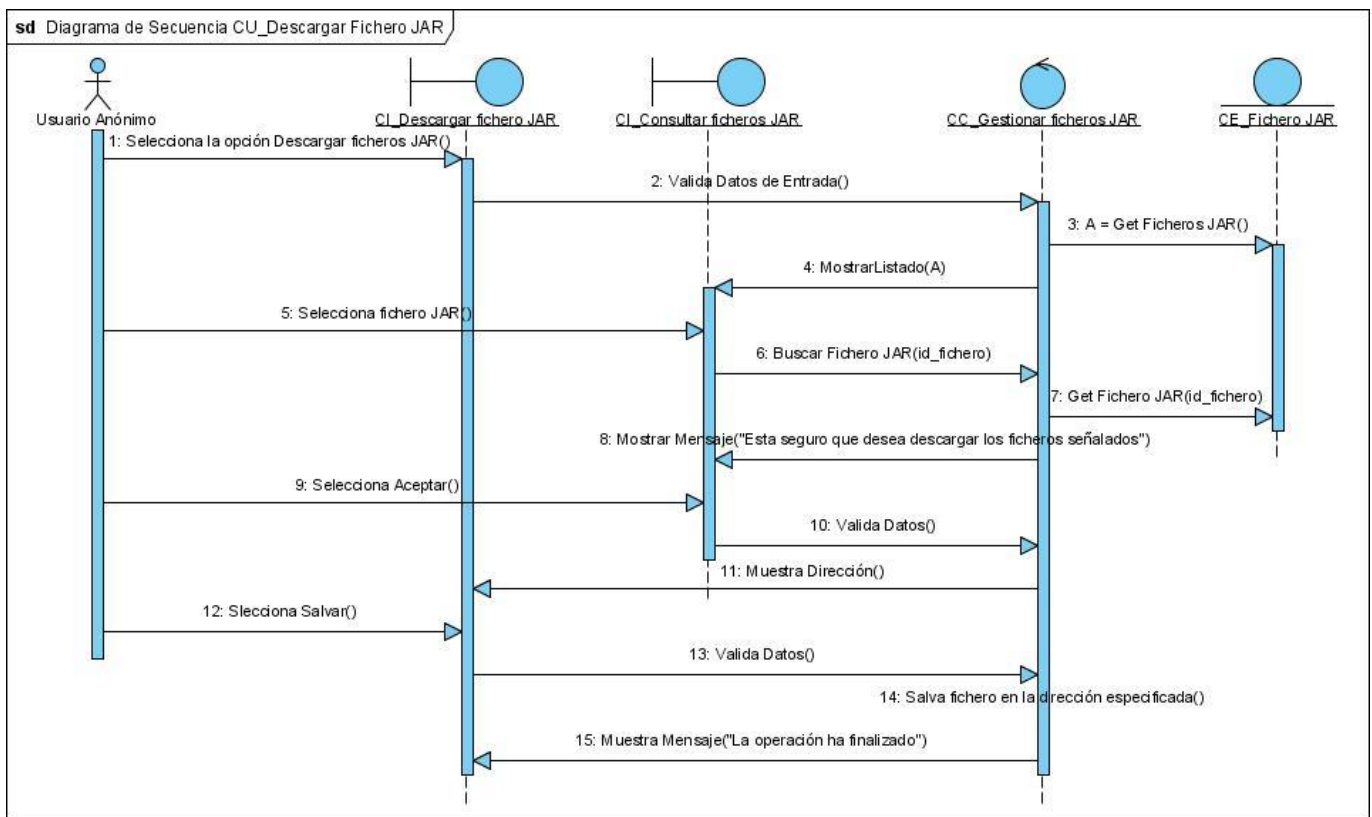


Figura 55: Diagrama de secuencia del CU_Descargar Fichero JAR