



Universidad Nacional de Catamarca

Facultad de Tecnología y Ciencias Aplicadas

LICENCIATURA EN SISTEMAS DE INFORMACIÓN

**“APLICACIÓN DE LA METODOLOGÍA UWE PARA EL MODELADO Y
DESARROLLO DE UN PROTOTIPO DE SISTEMA DE INFORMACIÓN
ADMINISTRATIVA PARA LA DIRECCIÓN DE RECONOCIMIENTOS MÉDICOS DE
LA PROVINCIA DE CATAMARCA.”**

ALUMNO:

IBAÑEZ RAUL ADRIAN MU: 2139

DIRECTOR:

DR. AHUMADA CESAR HERNAN

MARZO DE 2017

ACTO QUE DEDICO A:

DIOS

Por darme la fortaleza y sabiduría para lograr una meta más.

MIS PADRES

Por apoyarme en todo momento y alentarme a seguir adelante.

MI ESPOSA

Por estar siempre a mi lado brindándome su apoyo y confianza.

AGRADECIMIENTOS

A MI DIRECTOR

El Dr. Hernán Ahumada por el tiempo dedicado, sus consejos, sus revisiones y su buena disposición y responsabilidad para guiarme a lo largo de este trabajo final.

OTROS

A la Lic. Carola Flores por su valiosa orientación y recomendaciones en la elección del tema de este trabajo final y a la Lic. Cecilia Gallardo por ayudarme oportunamente a salir adelante en las distintas encrucijadas durante el desarrollo del mismo.

RESUMEN

La Dirección Provincial de Reconocimientos Médicos es una institución pública, cuya función social es controlar, diagnosticar y determinar la vinculación que hubiere entre la generación de los derechos sociales a favor de los agentes públicos, en función de sus diversos estados de dolencia con la concurrencia a sus lugares de trabajos.

Actualmente, la problemática que presenta esta institución, es la de no poder agilizar y optimizar el proceso administrativo, ya que el Software que utiliza, carece de funcionalidades requeridas por los empleados para llevar a cabo de manera eficiente y eficaz sus tareas, y así brindar un mejor servicio a la sociedad.

En consecuencia, este trabajo está pensado en base a esa necesidad. El diseño y desarrollo de un Prototipo de Sistema de Información Administrativa basado en Web, propone mediante la aplicación de la metodología UWE, suplir las necesidades informáticas de esta institución. Es importante destacar, por una lado, el empleo de un enfoque de Ingeniería Web para garantizar las funcionalidades requeridas en gestión administrativa y, por otro lado, las ventajas de Internet y la nuevas tecnologías para la Web en el desarrollo del Prototipo, ya que permitirá una fácil implementación; una forma sencilla de centralizar las áreas de trabajo y una mejor calidad de atención.

Palabras claves: Metodología UWE; Ingeniería Web; Aplicación Web; Desarrollo de Software.

ÍNDICE

RESUMEN	4
CAPÍTULO I: INTRODUCCIÓN GENERAL	7
1.1 PLANTEAMIENTO DEL PROBLEMA	7
1.1.1 Problema Principal	8
1.1.2 Causas del Problema	8
1.2 JUSTIFICACIÓN	9
1.3 CARACTERÍSTICAS DEL PROYECTO	9
1.4 ALCANCE	9
1.5 OBJETIVOS	10
1.5.1 Objetivos Generales	10
1.5.2 Objetivos Específicos	10
1.6 IMPACTO	11
CAPÍTULO II: MARCO TEÓRICO	12
2.1 INTRODUCCIÓN	12
2.2 APLICACIÓN O SISTEMA BASADO EN WEB	12
2.3 MODELADO DE APLICACIONES WEB	13
2.4 METODOLOGÍAS DE INGENIERÍA WEB	14
2.5 Lenguaje Unificado De Modelado (UML)	17
2.6 METODOLOGÍA UWE: INGENIERÍA WEB BASADA EN UML	18
2.6.1 Características de UWE	19
2.6.2 Modelos de la Metodología UWE	19
2.7 EJEMPLO DE MODELADO DE ANÁLISIS Y DISEÑO EN UWE UML	21
2.7.1 Análisis de Requisitos	21
2.7.2 Modelo de Contenidos	22
2.7.3 Modelo de Navegación	22
2.7.4 Modelo de Presentación	24
2.7.5 Modelo de Procesos	26
2.7.5.1 Modelo de Estructura de Proceso	27
2.7.5.2 Modelo de Flujo del Proceso	28
2.8 IDENTIFICACIÓN DEL FRAMEWORK.	29
2.8.1 Características requeridas del Framework	30
2.8.2 Evaluación de los Framework PHP	30
2.8.3 Patrón de Diseño Modelo-Vista-Controlador (MVC)	31
2.8.4 Arquitectura MVC empleada en el Prototipo SIARMe	32
CAPÍTULO III: ANÁLISIS DE REQUISITOS DEL PROTOTIPO SIARME	34
3.1 INTRODUCCIÓN	34
3.2 ESPECIFICACIÓN DE REQUISITOS	34
3.2.1 Propósito	34
3.2.2 Ámbito del Sistema	34
3.2.3 Personal Involucrado	35

3.2.4	Definiciones, Acrónimos y Abreviaturas	35
3.2.5	Referencias	36
3.2.6	Descripción General	36
3.2.6.1	Perspectiva del producto	37
3.2.6.2	Características de los Usuarios:	38
3.2.7	Funciones del Producto	38
3.2.7.1	Restricciones	39
3.2.7.2	Requisitos No funcionales	39
3.2.7.3	Requisitos Funcionales	39
3.2.7.4	Requisitos que Representan Reglas de Negocio	40
3.3	ANÁLISIS DE REQUISITOS EMPLEANDO LA METODOLOGÍA UWE	40
3.3.1	Modelo de Requisitos del Prototipo SARMe	41
3.4	DESCRIPCIÓN TEXTUAL DE CASOS DE USO	44
3.4.1	Diagrama de Casos de uso del Paquete Gestión de Usuario	46
3.4.2	Diagrama Casos de uso del Paquete Gestión de Expediente	48
3.4.3	Diagrama de Casos de Uso del Paquete Gestión de Documento	53
CAPÍTULO IV: DISEÑO EMPLEANDO LA METODOLOGÍA U.W.E.		58
4.1	INTRODUCCIÓN	58
4.2	MODELO DE CONTENIDO	58
4.3	MODELO NAVEGACIONAL	59
4.4	MODELO DE PRESENTACIÓN	61
4.4.1	Diseño Base del Prototipo SIARMe	62
4.4.2	Diseño de la Zona Pública o Portada	62
4.4.3	Diseño de la Zona Restringida o Intranet	64
4.4.3.1	Diagrama de Presentación Paquete Gestión de Expediente	64
4.4.3.2	Diagrama de Presentación del Paquete Gestión de Documento	65
4.4.3.3	Diagrama de Presentación del Paquete Gestión de Usuario	67
4.5	MODELO DE PROCESOS	68
4.5.1	Diagramas de Procesos del Paquete Gestión de Usuario	69
4.5.2	Diagramas de Proceso del Paquete Gestión de Expediente	71
4.5.3	Diagramas de Proceso del Paquete Gestión de Documento	75
CAPÍTULO V: DESARROLLO DEL PROTOTIPO SIARME.		77
5.1	INTRODUCCIÓN	77
5.2	DESARROLLO DE LA ZONA PÚBLICA O PORTADA	77
5.3	DESARROLLO DE LA ZONA RESTRINGIDA O INTRANET	79
5.4	DESARROLLO DEL MÓDULO DE ADMINISTRACIÓN	84
CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES		86
6.1	CONCLUSIONES	86
6.2	RECOMENDACIONES	86
REFERENCIA BIBLIOGRÁFICA		87
ANEXO I: Instalación del Prototipo SIARMe		89

CAPÍTULO I: INTRODUCCIÓN GENERAL

Desde los inicios de la década del 40, cuando se desarrollaron las primeras computadoras digitales, la programación de estas máquinas era considerada un arte que no disponía de métodos sistemáticos para la realización de la programación, por lo que se llevaba a cabo, sin ninguna planificación, evolución y perspectivas de ingeniería. Con el transcurso de los años, y la evolución de estas máquinas, la programación fue cada vez más compleja. [1]

Recién a finales de la década de los 60 se realizó la primera conferencia elaborada por la OTAN (Organización del Tratado del Atlántico Norte), donde por primera vez se trataron las dificultades o errores ocurridos en la planificación, estimación de los costos, productividad y calidad de un software, conocidos como “la crisis del software”, es decir, se necesitaban nuevas técnicas y métodos para controlar la complejidad inherente a los grandes sistemas. Para dar solución a estos problemas se creó una nueva rama de ingeniería, la “Ingeniería de Software”(disciplina o área de la informática que ofrece métodos y técnicas para desarrollar y mantener software de calidad)[2], cuyo propósito fue ayudar a identificar y corregir mediante principios y metodologías los procesos de desarrollo y mantenimiento de sistemas de software. Luego, a principios de la década de los 80, esta disciplina ya había surgido como una genuina profesión, para estar al lado de las ciencias de la computación y la ingeniería tradicional (ingeniería civil, ingeniería electrónica, etc). A su vez, se realizaron muchos progresos en los procesos de diseño y metodologías de desarrollo del software tales como la programación estructurada, programación orientada a objetos, programación orientada a los aspectos, el surgimiento de las herramientas CASE (Ingeniería de Software Asistida por Computadora), el lenguaje de programación ADA, la documentación, el lenguaje de modelado UML, etc [3]. La incorporación de estos nuevos avances de métodos formales, hacía suponer que la industria del software sería predecible como lo eran otras ramas de la ingeniería.

Sin embargo, el auge de Internet con las nuevas tecnologías de comunicación y el surgimiento la World Wide Web (WWW, o simplemente Web) dieron origen a un nuevo tipo de software las “aplicaciones Web” que poseían aspectos y características diferentes del software tradicional debido a la conjunción de hipertexto y multimedia.

Recién a mediados de la década del 90, el reconocimiento por parte de la comunidad de Ingeniería de Software de que los métodos tradicionales no eran apropiados para este nuevo tipo de Software, trajo aparejado el surgimiento de la Ingeniería Web, como un nuevo paradigma.

En la actualidad la web está sufriendo grandes cambios, que ha obligado a expertos en esta nueva disciplina a proponer extensiones y adecuaciones a los métodos y modelos tradicionales para el modelado conceptual de aplicaciones Web.

1.1 PLANTEAMIENTO DEL PROBLEMA

El presente trabajo pretende aplicar un enfoque disciplinado y sistemático en base a la metodología UWE (Ingeniería Web basada en UML) para el modelado y desarrollo de un Prototipo de Sistema Basado en Web de alta calidad para dar soporte a la información

requerida en la gestión administrativa de expedientes en la Dirección Provincial Servicio de Reconocimientos Médicos de la Provincia de Catamarca.

La Dirección Provincial de Reconocimientos Médicos (DPSRM) de Catamarca se compone actualmente de tres Departamentos: **Ausentismo**, **Despacho** y **Autoseguro**. Las funciones que se llevan a cabo en cada una de ellos son:

El Departamento Ausentismo: cuenta con un sistema informático, mediante el cual se registran **solicitudes de licencias por estado sanitario** (Art. 23, Art. 25 y Art. 38 según Dcto. Acdo. 1875/94) [4] y, además, se calculan estadísticas de las mismas.

El Departamento Despacho: Se encarga de llevar el registro y la clasificación de **expedientes** y, además, se realizan trámites por **excepciones de licencias mediante expediente** (Art. 23, Art. 25, Art. 32 y Art.38 según Dcto. Acdo. 1875/94) [4].

El Departamento Autoseguro [5]: Se encarga de los trámites de licencias por **accidentes de trabajo por expediente** (Art. 28 - Dcto. Acdo. 1875/94), entre otros trámites con relación a dicha licencia.

1.1.1 Problema Principal

Los procesos administrativos en la gestión de expedientes, dependen en gran medida de programas ofimáticos [6] debido a que los empleados necesitan de estos para crear los documentos o llevar un registro de la información generada en los trámites por expedientes que se realizan en la institución, para dar solución y brindar respuestas concretas e inmediatas a los agentes interesados. Por lo que estos programas forman una parte integral en la realización y concreción de las tareas administrativas. Pero debido a las funcionalidades que ofrecen estos programas, no logran satisfacer las necesidades administrativas, razón por la cual traen aparejadas un sinnúmero de problemas para los mencionados propósitos, en los Departamentos Autoseguro y Despacho de la DPSRM.

1.1.2 Causas del Problema

Tanto el Departamento Autoseguro como el Departamento Despacho cuentan con sus propias aplicaciones desarrolladas en Microsoft Access [7], pero presentan los siguientes problemas:

- No poseen una base de datos común, para centralizar la información, esto obliga a los empleados a ingresar los mismos datos en ambas aplicaciones, duplicando de manera innecesaria la información (datos de expedientes, datos de agentes públicos, datos de organismos, etc).
- La interfaz gráfica que poseen, no es amigable al usuario ya que para comprender la información y las opciones que se presentan es necesario el asesoramiento de un usuario experimentado.
- En la validación de datos de los formularios, no se les informa a los usuarios, cuáles son los datos incorrectos introducidos.

- Existe redundancia en los datos, por ejemplo los datos de un mismo expediente pueden estar registrado varias veces.
- Dificultades para tener acceso a los datos, pocas opciones para realizar reportes, es decir, pueden acceder a ellos solo empleados con experiencia.
- El control de acceso para el resguardo de la información sensible no es suficiente, por ejemplo, los informes médicos. Esto se debe a que el control es mediante una única contraseña para acceder a la información.

1.2 JUSTIFICACIÓN

La propuesta de un Prototipo de Sistema de Información Administrativa basado en Web para la DPSRM se fundamenta, por un lado, en la utilización de un moderno enfoque metodológico de ingeniería web con el fin de aplicar sólidos principios de ingeniería, y además enfoques disciplinados y sistemáticos para obtener una descripción completa de las funcionalidades que permitirán el adecuado soporte de la información, en este caso particular, en la gestión administrativa de expedientes requeridas por parte de:

- Los empleados, para llevar el registro de la información administrativas, necesaria en la gestión de expedientes.
- Los Médicos, para llevar un registro de la información de los dictámenes y licencias especiales otorgadas mediante expedientes.
- Los agentes públicos, para poder conocer el estado de sus expedientes vía internet.

Por otro lado, las características de las nuevas aplicaciones o sistemas basados en Web, incorporan múltiples ventajas frente al software tradicional, ya que permiten de forma sencilla centralizar todas las áreas de trabajo, debido a la inmediatez de acceso a través de los navegadores web en un equipo cliente.

1.3 CARACTERÍSTICAS DEL PROYECTO

Es un proyecto de investigación aplicada de Ingeniería Web, sistemática y secuencial (análisis de los requisitos, diseño y codificación), a fin de obtener un Prototipo de Sistema de Información Administrativa basado en Web que posibilite en su implementación, optimizar los procesos administrativos en la DPSRM.

Para tal fin, serán aplicados los conocimientos adquiridos referentes a Ingeniería Web, programación y bases de datos. Además se utilizará la metodología de Ingeniería Web Basada en UML (UWE) [8] y herramientas CASE como: MagicDraw 1.51 y MagicUWE 1.2 plugin, ArgoUML y ArgoUWE plugin.

1.4 ALCANCE

Aplicación de la metodología de Ingeniería Web para el modelado y desarrollo de un prototipo de Sistema basado en Web (SIARMe) para la DPSRM, que permita contemplar la

totalidad de las funcionalidades requeridas por los empleados, necesarias en la realización de las tareas administrativas.

Obtener un Prototipo funcional que permita centralizar la información administrativa generada en la gestión de expedientes, abarcando las siguientes funcionalidades:

Gestionar Usuarios: Administrar los usuarios del sistema, asegurando el resguardo e integridad de la información y el acceso a la misma según el tipo de usuario.

Gestionar Expedientes: Los expedientes deben ser registrados en el sistema para tener su correspondiente registro electrónico y poder realizar el seguimiento automático de éste o saber en qué área se encuentra, o bien, su estado.

Gestionar Información de Documentos: Un vez creado el registro electrónico de un expediente, los usuarios podrán registrar la información de documentos administrativos.

Cabe aclarar que si bien, el Prototipo del Sistema propuesto no alcanzará las tareas que se realizan en el departamento ausentismo, sí tomará del sistema informático que este posee, la estructura y configuración de la base de datos relacional en MySQL, para ser implementado, agregando las tablas que sean necesarias, sin afectar la funcionalidad de dicho sistema.

1.5 OBJETIVOS

1.5.1 Objetivos Generales

- Modelar y desarrollar un Prototipo de Sistema basado en Web para la gestión administrativa de expedientes de la DPSRM de Catamarca, empleando la metodología de Ingeniería Web basada en UML (UWE).

1.5.2 Objetivos Específicos

- Especificar las funcionalidades del Prototipo de Sistema de Información Administrativa basado en Web mediante la metodología UWE y las instrucciones dadas por el estándar IEEE ANSI/IEEE 830 para obtener una descripción completa de las funcionalidades requeridas en la gestión administrativa de los Departamentos Autoseguro y Despacho.
- Modelar el Prototipo, en base a las funcionalidades requeridas, empleando la metodología UWE para poder visualizar los diferentes puntos de vistas del prototipo, requeridos en el desarrollo o codificación.
- Obtener una codificación del Prototipo organizada que permita su escalabilidad, empleando técnicas y herramientas de programación y de acuerdo a las funcionalidades y modelado.

1.6 IMPACTO

- Despertar el interés sobre las ventajas de implementar Sistemas Basados en Web en ámbito de la Administración Pública.
- Lograr que el Prototipo SIARMe responda a las demandas administrativas y posibilite agilizar y optimizar los procesos administrativos.
- Posibilitar una mejor calidad de atención y eficacia laboral de cada empleado.
- Ofrecer un prototipo que permita su escalabilidad ante nuevos requerimientos, con una modularidad fácil de adaptar, mejorar, probar y utilizar.
- Desarrollar experiencias que puedan resultar en destreza profesional.

CAPÍTULO II: MARCO TEÓRICO

2.1 INTRODUCCIÓN

El avance de Internet y las nuevas tecnologías de comunicación, marcaron una nueva tendencia en el mercado del software. Esta nueva tendencia conjuga hipertexto y multimedia, por un lado, y la popularización de la Web, por otro. Ambos generaron desafíos que los métodos tradicionales no eran capaces, aparentemente, de resolver. Al respecto, Roger Pressman indica que el uso de soluciones de Ingeniería de Software, conduciría a una “web enmarañada” (que entraña un cúmulo de aplicaciones basadas en web mal desarrolladas) y que tienen muy altas probabilidades de fracaso. Por lo que sugiere que para lograr mayor éxito en el desarrollo y la aplicación de sistemas basados en Web complejos y a gran escala existe una apremiante necesidad de enfoques disciplinados y nuevos métodos y herramientas con qué desarrollar, desplegar y evaluar las aplicaciones o sistemas basados en Web [2]. Se trata de un enfoque integral y proactivo para el desarrollo de aplicaciones web diversas y complejas, ya que su objetivo es mantener bajo control el desarrollo de sistemas basados en la Web, y así, minimizar los riesgos y mejorar la capacidad de mantenimiento y calidad.

Al respecto, Murugesan, Deshpande, Hansen, & Ginige, (1999) [9], afirman que: Ingeniería Web es “la aplicación de principios científicos, de ingeniería y de gestión y de enfoques disciplinados y sistemáticos para el desarrollo, implementación y mantenimiento exitoso de sistemas y aplicaciones basadas en la web, de alta calidad”. Sus características son: confluencias de disciplinas (Sistemas de Información, Ingeniería Software y Diseño Gráfico que requieren de equipos multidisciplinares y polivalentes)[9], Ciclos de vida y tiempo de desarrollo muy cortos, cambio continuo (brindar soluciones que permitan la flexibilidad y adaptación en los cambios del proyecto.), Requisitos fuertes de Seguridad, Rendimiento y Usabilidad[10].

A continuación, en este capítulo se desarrollarán los siguientes temas:

- Aplicación o Sistema basado en Web y Modelado de Aplicaciones Web en las secciones 2.2 y 2.3;
- Metodologías de Ingeniería Web en las secciones 2.4 a 2.6;
- Ejemplo de Modelado de Análisis y Diseño en UWE en la sección 2.7;
- Identificación del Framework de Desarrollo en la sección 2.8;

2.2 APLICACIÓN O SISTEMA BASADO EN WEB

Una aplicación Web es un sistema informático en el que los usuarios ingresan a un servidor por medio de internet o una intranet mediante un navegador. Una Aplicación o Sistema Web podría definirse entonces como “una aplicación cliente / servidor que utiliza un navegador Web como su programa cliente, y lleva a cabo un servicio interactivo mediante la conexión con servidores Web a través de Internet o Intranet” [11]. Estas aplicaciones generan dinámicamente una serie de páginas en un formato estándar, como HTML o XHTML, que soportan los navegadores web como cliente ligero, sin que se deba distribuir e instalar software adicionales para cada usuario. La alta popularidad de este tipo de

aplicaciones se debe a diversas bondades que estas presentan, por ejemplo: actualización inmediata, multiplataforma (varias tecnologías incluyendo PHP, Java, Flash, ASP y Ajax permiten un desarrollo efectivo de programas soportando todos los sistemas operativos principales), fácil mantenimiento, portables, y sobre todo, un ambiente colaborativo, ya que son capaces de soportar grupos de usuarios que realizan tareas de manera cooperativa, y permiten que distintas personas accedan a una información común.

2.3 MODELADO DE APLICACIONES WEB

El modelado de una aplicación web tiene como objetivo proporcionar una especificación del sistema que será construido con un grado de detalle lo suficientemente efectivo como para poder llevar a cabo la implementación. Los modelos representan los aspectos relevantes del sistema en una manera simplificada y comprensible.

Durante el desarrollo de Aplicaciones Web, los modelos también constituyen un sólido punto de partida, teniendo en cuenta su complejidad, los aspectos estáticos y dinámicos de los niveles de contenido, hipertexto y presentación de una aplicación web a través de tres dimensiones: niveles, aspectos y fases [12].

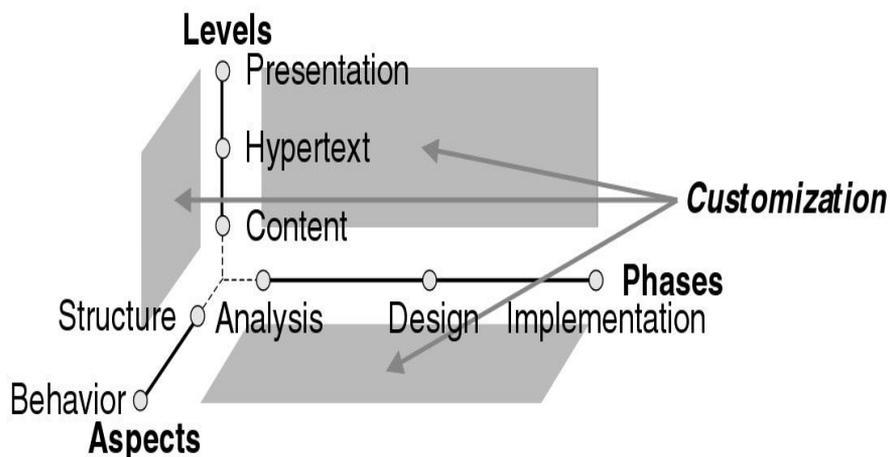


Figura [1]: Requerimientos de modelado de Aplicaciones Web.

Los modelos se construyen a lo largo de las diferentes "fases" de la ingeniería de requisitos, análisis, diseño e implementación correspondientes al proceso de desarrollo como se observa en la figura [1]. A su vez, estos modelos se utilizan para representar diferentes vistas de una misma aplicación web correspondiente a los distintos "niveles" que se detectan: contenido, hipertexto y presentación. El modelo de contenido se utiliza para especificar los conceptos que son relevantes al dominio de aplicación y la relación entre ellos. El modelo de hipertexto representa las rutas de navegación del sistema Web, se modela por separado del modelo de contenido, aunque deriva de éste. El modelo de presentación se enfoca en la creación de una estructura de presentación uniforme para las páginas que componen la Aplicación Web. En la dimensión de "aspectos", se modelan la estructura y el comportamiento para cada uno de los tres niveles de contenido, hipertexto y presentación, de acuerdo a los principios orientados a objeto. La relevancia de los modelos de estructura y comportamiento depende del tipo de Aplicación Web a ser implementada.

Las aplicaciones web que principalmente manejan información estática requieren menos modelado del comportamiento en comparación con las aplicaciones web sumamente interactivas, como por ejemplo aplicaciones de comercio electrónico.

La dimensión de “personalización”, se refiere a un sistema web personalizado y dependiente del contexto que proporciona al usuario información más apropiada, vínculos o páginas dependientes de las características contextuales o del usuario. Esta dimensión influye sobre las demás dimensiones de modelado web.

2.4 METODOLOGÍAS DE INGENIERÍA WEB.

Los métodos de la Ingeniería Web definen las etapas y actividades necesarias para efectuar la construcción completa de una aplicación Web. El principio subyacente en todos ellos es que una aplicación Web debe desarrollarse partiendo de una descripción precisa en forma de un esquema conceptual que se transforma a una representación software, mediante un conjunto de correspondencias entre las abstracciones conceptuales que constituyen su esquema conceptual y los componentes de software.

Los métodos de modelado siguen diferentes enfoques. En menor o mayor medida y a veces con diferentes nombres o sub-fases, la mayoría de los métodos coinciden en las siguientes etapas (Diseño Conceptual, Diseño Navegacional, Diseño de la presentación o diseño de Interfaz, e Implementación). Estos se detallan a continuación [12], [13]:

- **Métodos orientados a datos**, el principal objetivo es el modelado de aplicaciones web basadas en bases de datos. Los Métodos orientados a datos pueden ser RMM (Relationship Management Methodology) y WebML (Web Modeling Language)[14].
- **Métodos orientados a hipertexto**, surgieron en el campo de los sistemas de hipertexto. Ejemplos de métodos orientados a hipertexto pueden ser HDM (Hypertext Design Model) o WSDM (Web Site Design Method).
- **Métodos orientados a objetos**, que están basados en OMT o UML. UML es la anotación preferida. Estos métodos están orientados al diseño hipermedia. Ejemplos de métodos orientados a objetos pueden ser OOHDM (Object-Oriented Hypermedia Design Model) [15] o UWE (UML-based Web Engineering)[16].
- **Métodos orientados a software**, que afrontan las aplicaciones web desde la perspectiva que sigue la ingeniería del software tradicional. Ejemplos de métodos orientados a software son WAE (Web Application Extensión) y WAE2.

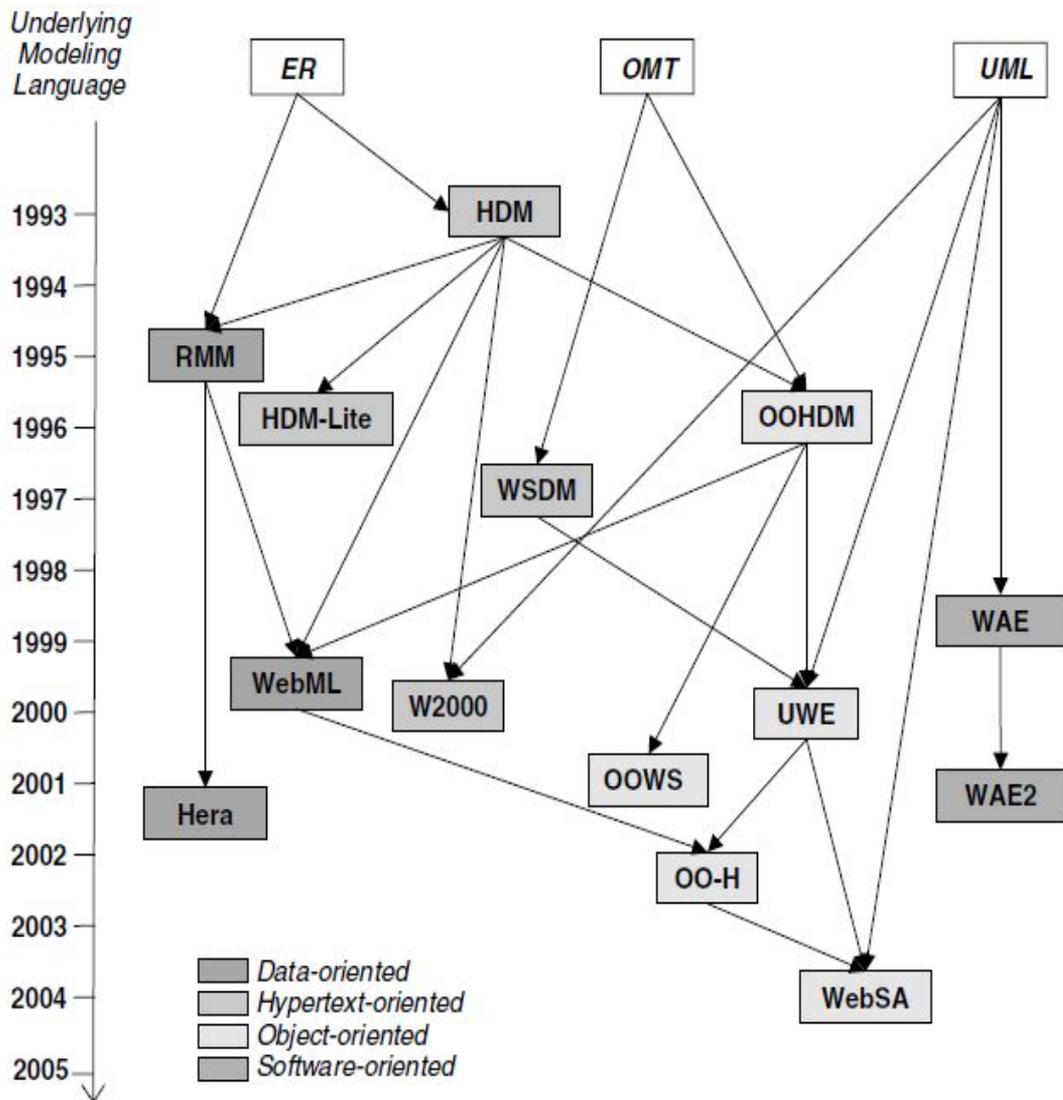


Figura [2]: Desarrollo histórico de Métodos de Modelado de Aplicaciones Web [12].

Los métodos disponibles para el modelado de aplicaciones web, están basados, generalmente, en métodos tradicionales, como el modelo entidad-relación (ER) o el modelado orientado a objetos, como UML. Los métodos actuales se basan en la fortaleza de los métodos anteriores.

En la figura [2], se presenta una descripción de la historia de los métodos de modelado de aplicaciones web. Se observa que HDM-lite es el sucesor de HDM. Este método fue diseñado para centrarse en la automatización del proceso de desarrollo y con generación automática de aplicaciones web. Otra mejora de HDM es W2000 en la que el hipertexto y las perspectivas están centradas en el usuario. RMM es un método basado en el modelo ER y define un proceso gradual para el perfeccionamiento de los modelos sucesivos. Hera también está basado en los modelos ER. WebML es un lenguaje de modelado de aplicaciones web fácil de entender y bastante maduro. WebML utiliza la herramienta WebRatio. OOHDM enfatiza el concepto de navegación, y por eso es recomendado para construir aplicaciones web que ofrezcan diferentes opciones de navegación. WSDM se centra en un enfoque metódico orientado a las necesidades de los

usuarios. UWE es un enfoque que ofrece una notación basada en UML y un meta-modelo de control basado en el modelo de consistencia. OO-H es uno de los métodos más recientes, que combina los beneficios de WebML, OOHDM y UWE. Utiliza la herramienta VisualWADE que apoya el modelo de generación automática de código. OOWS es un enfoque orientado a objetos basado en UML y con una notación propia. WAE2 es un enfoque que se centra en UML y que tiene en cuenta la distribución lógica de la aplicación. Por último WebSA es un enfoque basado en arquitectura.

En la figura [3] se presenta una descripción más detallada de cada uno de los métodos des

	Modeling Method	Modeling Paradigm	Notation	Evolution	Requirements Modeling	Content Modeling	Hypertext Modeling	Presentation Modeling	Customization Modeling	Structure Modeling	Process / Approach	Tool Support	Generation	Strengths
HDM-IIIe	HT	ER + own notation	x	x	x	✓	✓	x	s	own	generation tools	auto	process for model transformation, automatic generation	
Hera	DB	ER + RMM + own notation	✓	x	✓	✓	✓	s + b	own	authoring & generation tool	semi	model-driven development		
OO-H	OO	UML + own notation	✓	✓	✓	✓	x	pers	s + b	own	modeling- & generation tool	auto	tool for automatic generation	
OOHDM	OO	UML + own notation	✓	✓	✓	✓	✓	pers	s + b	own	x	x	powerful concepts for contextual navigation, personalization	
OOWS	OO	UML + own notation	✓	✓	✓	✓	x	s + b	own	modeling- & generation tool	auto	advanced (commercial) tool for automatic generation		
RMM	DB	ER + own notation	x	x	✓	✓	✓	x	s	own	authoring tool	semi	hypertext modeling based on ER-model, predefined process	
UWE	OO	UML	✓	✓	✓	✓	✓	pers	s + b	RUP	extended UML tool & generation tools	semi	UML-based method, model-driven development, aspect-oriented customization	
W2000 (HDM)	HT	UML	✓	✓	x	✓	✓	pers	s	x	extended UML-tool	x	user-centric hypertext modeling	
WAE2 (WAE)	SW	UML	✓	✓	✓	x	✓	x	s + b	RUP	standard UML-tools	x	Implementation design, architectural design	
WebML	DB	ER, UML	✓	✓	✓	✓	x	pers	s + b	own	modeling- & generation tool	auto	well-elaborated notation, database integration, generation	
WS DM	HT	own notation	✓	x	✓	✓	x	x	s + b	own	x	x	user-centric approach for analysis	

✓	supported	pers	personalization	RUP	Rational Unified Process	DB	data-oriented
x	not supported	s	structure modeling	own	own process model / approach	HT	hypertext-oriented
		b	behavior modeling	auto	automatic generation	OO	object-oriented
				semi	semi-automatic generation	SW	software-oriented

critos.

Figura [3]: Métodos de Modelado de Aplicaciones Web [12].

Según el análisis comparativo de las características de las diferentes metodologías de ingeniería Web presentadas en la figura [3] se observa que no todos los métodos cubren los mismos aspectos. En algunos casos, hay métodos que no cubren todo el ciclo de vida, porque se especializan en una fase, sin brindar suficiente soporte al resto. Sin embargo, la elección de UWE como metodología para modelar la Aplicación Web propuesta en este trabajo, se debe a que es uno de los métodos más completos, ya que propone al menos un tipo de diagrama UML para la visualización de cada modelo cubriendo todos los aspectos de modelado [18].

2.5 Lenguaje Unificado De Modelado (UML).

UML (Unified Modeling Language) o Lenguaje Unificado de Modelado comenzó a gestarse en octubre de 1994, cuando Rumbaugh se unió a la compañía Rational fundada por Booch (dos reputados investigadores en el área de metodología del software). El objetivo de ambos era unificar dos métodos que habían desarrollado: el método Booch y el OMT (Object Modelling Tool) [19].

El primer borrador apareció en octubre de 1995. En esa misma época otro reputado investigador, Jacobson, se unió a Rational y se incluyeron ideas suyas. Estas tres personas son conocidas como los “tres amigos”. Además, este lenguaje se abrió a la colaboración de otras empresas para que aportaran sus ideas.

Todas estas colaboraciones condujeron a la definición de la primera versión de UML 1.1 como un estándar de OMG (Object Management Group) en 1997. UML ha madurado considerablemente desde su primera versión, en la actualidad, es un estándar internacional de sistemas software para el de diseño orientado a objetos. Permite visualizar, especificar, construir y documentar un sistema software. UML ofrece un estándar para describir el modelo del sistema incluyendo aspectos conceptuales y aspectos concretos como expresiones de lenguajes de programación.

UML es apropiado para modelar diversos sistemas:

- Sistemas de información de gestión en empresas.
- Aplicaciones distribuidas basadas en la Web.
- Sistemas empotrados de tiempo real.

UML 2.0 propone 13 tipos de diagramas y para comprenderlos de manera concreta, a veces es útil categorizarlos jerárquicamente como se observa en la figura[4] :

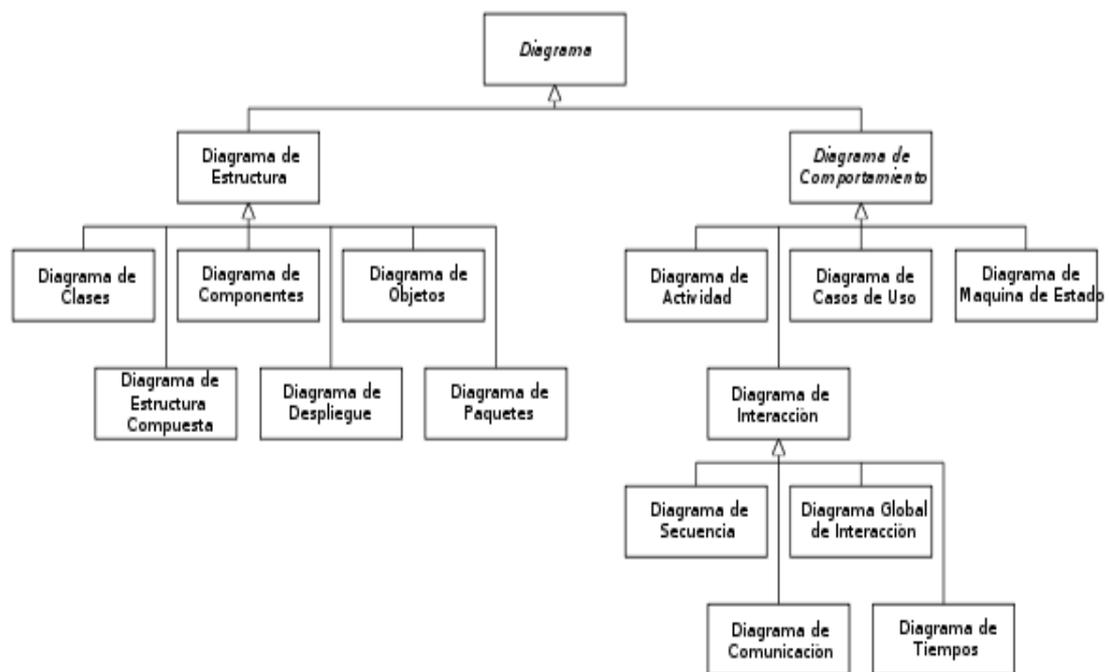


Figura [4]: Diagramas de UML

Los Diagramas de Estructura enfatizan en los elementos que deben existir en el sistema modelado: Diagrama de clases, Diagrama de componentes, Diagrama de objetos, Diagrama de estructura compuesta (UML 2.0), Diagrama de despliegue, Diagrama de paquetes.

Los Diagramas de Comportamiento enfatizan en lo que debe suceder en el sistema modelado: Diagrama de actividades, Diagrama de casos de uso, Diagrama de estados.

Los Diagramas de Interacción son un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado: Diagrama de secuencia, Diagrama de comunicación, Diagrama de tiempos (UML 2.0), Diagrama global de interacciones o Diagrama de vista de interacción (UML 2.0).

2.6 METODOLOGÍA UWE: INGENIERÍA WEB BASADA EN UML

UWE es una metodología de Ingeniería Web que sirve para modelar aplicaciones web, basada en UML y el Proceso Unificado de desarrollo de Software (RUP). Esta metodología surgió en el año de 1999 como un proyecto del Departamento de Programación e Ingeniería de Software de la Universidad Ludwig-Maximilians de Múnich, encabezado por Alexander Knapp, Gefei Zhang y Nora Koch, con la finalidad de estandarizar el proceso de análisis y diseño de sistemas Web; su objetivo es cubrir todo el ciclo de vida, centrándose en el modelado visual, en el diseño sistemático y la generación automática [18].

Este mismo grupo se encargó de la creación de las herramientas Open Source para el soporte de UWE como ArgoUWE (ArgoUWE: CASE Tool for Modelling Web Applications) [20] y MagicUWE (MagicUWE: UWE Plugin for MagicDraw) [21].

UWE está basado en estándares de la OMG como UML, Model Driven Architecture de OMG (MDA), Object Constraint Language (OCL) y eXtensible Markup Language (XML), asegurando su seguimiento mediante guías y especificaciones para el uso de tecnologías orientadas a objetos. La notación UWE es una extensión ligera de UML y proporciona lo que se llama el perfil UML para el dominio web.

2.6.1 Características de UWE

- **Uso de una notación estándar**, para todos los modelos (UML: Lenguaje de modelado unificado).
- **Definición de métodos:** Definición de los pasos para la construcción de los diferentes modelos.
- **Especificación de Restricciones:** Se recomienda el uso de restricciones escritas (OCL: Lenguaje de restricciones de objetos) para aumentar la exactitud de los modelos.

2.6.2 Modelos de la Metodología UWE

Esta metodología consta de una notación basada en UML y de modelos construidos en las diferentes etapas del proceso de desarrollo. Su objetivo es cubrir todo el ciclo de desarrollo de aplicaciones web proporcionando técnicas y notaciones, comenzando con modelos de requisitos, siguiendo con los modelos de diseño e incluyendo modelos de arquitecturas y aspectos [22]. La notación está basada en un perfil de UML donde se utilizan principalmente cinco tipos de diagramas: Diagrama de Casos de Uso, para capturar los requisitos del sistema; Diagrama de Contenido, para el modelo de dominio; Diagrama de Estructura de Navegación, para modelar los caminos de navegación de la aplicación que se está modelando; Diagrama de Presentación, para modelar la apariencia visual y, finalmente, el modelado de tareas o procesos [23], [24].

Las extensiones definidas en el perfil representan los aspectos estructurales de las distintas vistas (contenido, navegacional y presentación), aunque los autores se apoyan en diagramas de interacción y diagramas de estado UML para modelar la perspectiva de comportamiento del sistema web.

En general, UWE provee diferentes modelos que permite describir una aplicación Web desde varios puntos de vista abstractos, dichos modelos están relacionados tal como se ilustra en la figura [5]

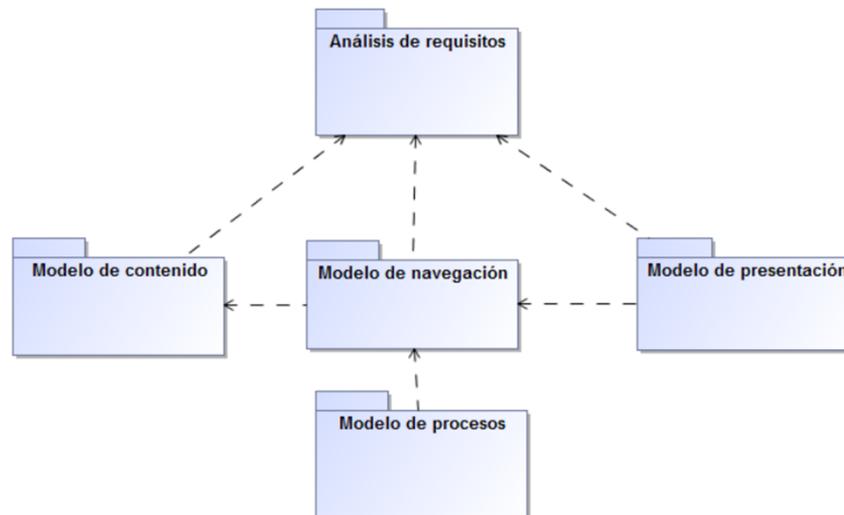


Figura [5]: Modelos de UWE

- **Análisis de Requisitos:** Plasma los requisitos funcionales de la aplicación Web mediante modelos de casos de uso.
- **Modelo de Contenido o Conceptual:** Materializado mediante un modelo de dominio, considerando los requisitos reflejados en los casos de uso; el diagrama de clases representan los conceptos con un gran porcentaje de detalle.
- **Modelo Navegacional:** Lo podemos subdividir en dos pasos:
 - *El espacio Navegacional:* Su objetivo es especificar qué objetos pueden ser visitados a través de la aplicación.
 - *La estructura de navegación:* Amplía, el modelo con un conjunto de estructuras de acceso necesarias para la navegación como Índices, consultas y visitas guiadas.
- **Modelo de Presentación:** Permite la especificación lógica de la aplicación Web. Sobre este modelo lógico se puede construir una representación física. Representa las vistas de la interfaz del usuario mediante modelos estándares de interacción UML. Dentro de este modelo se distinguen dos diferentes vistas.
 - *Estructura de Vista:* Muestra la estructura del espacio de presentación.
 - *Interfaz de Usuario* (UI por sus siglas en inglés de User Interface): Vista que presenta detalles acerca de los elementos de la interfaz de usuario dentro de las páginas.
- **Modelo de Procesos:** Se especifican las acciones que realiza cada clase de proceso, se distinguen dos diferentes vistas:
 - *Vista Estructural de Procesos:* mediante un modelo de estructura de procesos se describen las relaciones entre las diferentes clases de proceso.

o *Vista de Comportamiento o flujo de Proceso*: Mediante un modelo de flujo de procesos se especifican las actividades conectadas con cada proceso. Describe los comportamientos de una clase proceso, lo que ocurre en detalle dentro de cada una.

2.7 EJEMPLO DE MODELADO DE ANÁLISIS Y DISEÑO EN UWE UML.

El ejemplo planteado a continuación, a pesar de ser muy sencillo, servirá para explicar con mayor claridad lo propuesto por UWE, y debido a que el objetivo es mostrar los diferentes modelos de UWE, no se incluye la totalidad del modelo, sino que desarrolla al menos un ejemplo de cada diagrama. Todos los diagramas fueron realizados con la herramienta de modelado MagicDraw [21].

La aplicación web de ejemplo [25], es el de una libreta de direcciones, donde el usuario debe poder buscar contactos, sus direcciones y borrar contactos existentes. Además, los contactos pueden ser creados o modificados (estos cambios se pueden guardar o cancelar). Después de la creación exitosa de un contacto, este debe ser mostrado. En caso de error, debe mostrarse un mensaje. Además, se puede realizar búsquedas de contactos.

2.7.1 Análisis de Requisitos

Se realiza una visualización gráfica mediante el uso de diagramas de caso de uso de UML para modelar las funcionalidades requeridas como se observa en la figura[5]. Adicionalmente se puede agregar un listado de los casos de uso y una descripción de cada uno con el nivel de detalle que sea necesario para el proyecto

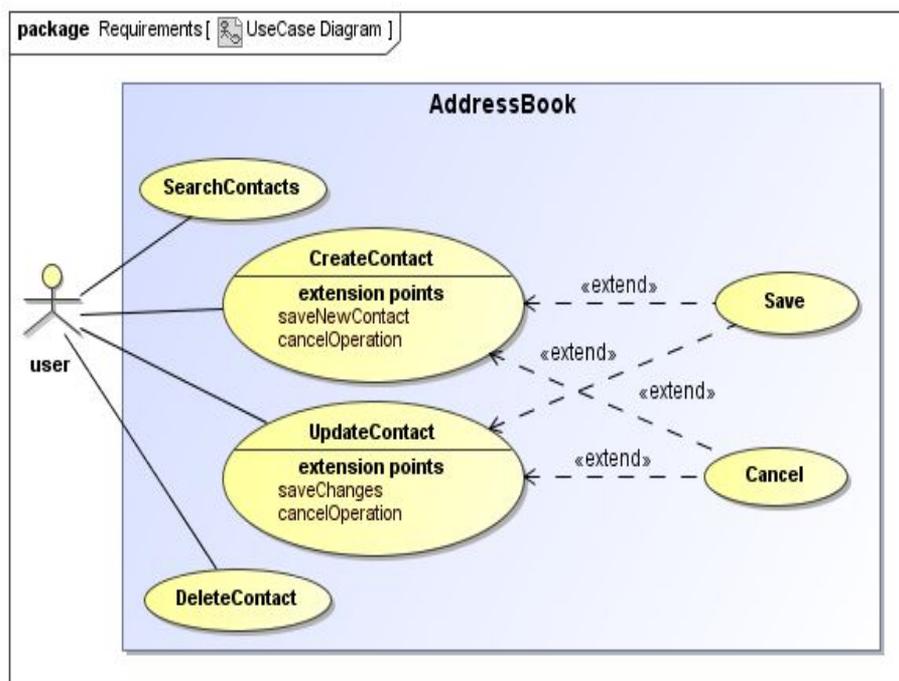


Figura [6]: Diagrama de Casos de Uso del ejemplo de UWE.

2.7.2 Modelo de Contenidos

Se utiliza el modelo de contenidos en base a un modelo de dominio de UML utilizando un diagrama de clases sin necesidad de aplicar semánticas adicionales, con el objetivo de representar los requerimientos de datos del sistema propuesto, modelando los datos persistentes como se observa. La figura [6] muestra el modelo de contenido de la libreta de direcciones de ejemplo, con las clases definidas para la libreta de direcciones, contacto, dirección y teléfono. El modelo de contenido es representado como un diagrama de clases UML.

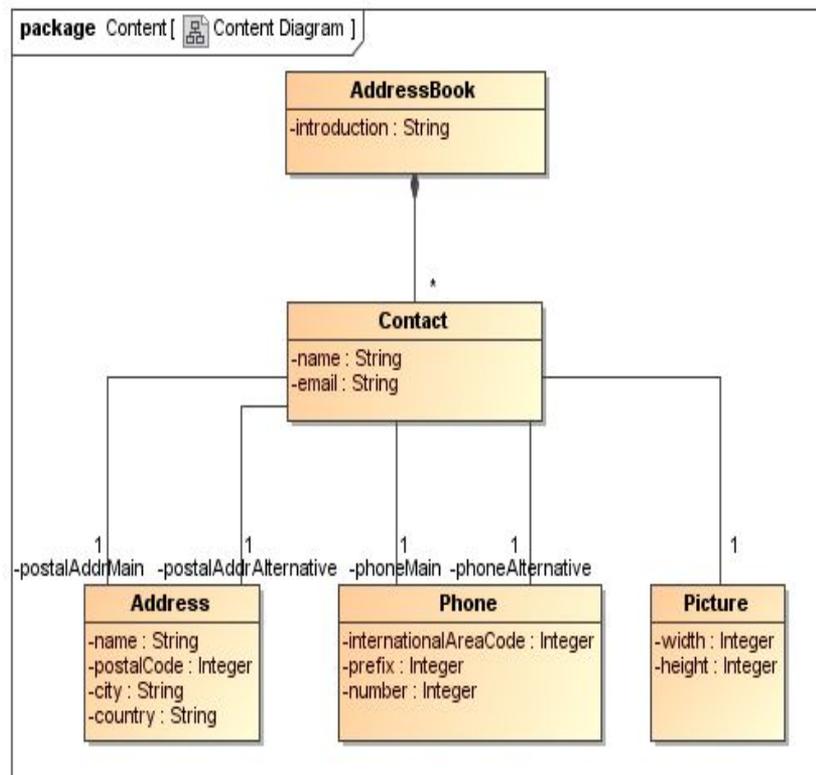


Figura [7]: Modelo de Contenidos del ejemplo de UWE UML.

2.7.3 Modelo de Navegación

El modelo de navegación tiene como objetivo representar nodos y vínculos de la estructura de hipertextos, diseñar los caminos de navegación, evitar desorientación y sobrecarga cognitiva. Este modelo es representado mediante el uso de diagramas de clase UML, utilizando semánticas adicionales para modelar elementos del contexto Web. Por una parte, el modelo de navegación representa la estructura de información estática de una aplicación Web accesible a un usuario, especificando las funcionalidades de navegación a través de los siguientes estereotipos.

Nombres de estereotipos y sus iconos

 navigationClass	 menu
 index	 query
 guidedTour	 processClass

- **<<navigationClass>>**: especifica un nodo de hipertexto visitado por un usuario a través de la navegación, y está relacionado a una clase de contenido.
- **<<navigationLink>>**: especifica un hipervínculo o link utilizado para acceder a un objeto de navegación objetivo desde el objeto de navegación origen.

Los nodos de estereotipo **<<navigationClass>>** cuentan con otros adicionales semánticos utilizados con el objetivo de modelar la estructura de navegación de forma más completa y clara:

- **<<index>>** : para indicar los vínculos de navegación que tienen una multiplicidad mayor a 1 en la asociación direccionada
- **<<menú>>**: para todas las clases de navegación que tienen más de una asociación de salida. Los menús son utilizados para estructurar los vínculos salientes de un nodo, generalmente asociados a una clase de navegación mediante composición, consistiendo en un grupo de vínculos a elementos heterogéneos como ser index, queries, instancias de clases de navegación u otros menús.
- **<< query>>**: Para la selección de instancias de una clase de navegación.
- **{isHome}**: Como una etiqueta en una determinada clase de navegación para indicar el nodo inicial de una aplicación.
- **{isLandmark}**: como una etiqueta en una determinada clase de navegación para indicar que el nodo es alcanzable desde cualquier lugar en la aplicación, y que por tanto todos los otros nodos contienen un vínculo a dicho nodo.

Por otra parte, se considera también el modelo de procesos que representará los aspectos dinámicos de una aplicación Web, especificando funcionalidades como transacciones y otras actividades. Es necesaria, entonces la definición de clases de proceso e integración de dichas clases de proceso en el modelo de navegación de nodos **<<navigationClass>>**, para lo cual se utilizan los siguientes agregados semánticos:

- **<<processClass>>**: representa un proceso a través del cual un usuario será guiado. En casos de procesos complejos es posible realizar un modelo de procesos adicional.
- **<<processLink>>** : es utilizado para modelar la asociación entre un nodo de estereotipo **<<navigationClass>>** y un nodo de estereotipo **<<processClass>>**.

En la Figura[8] se muestra el modelo de navegación del ejemplo UWE. Incluye dos clases de navegación **AddressBook** y **Contact**. La **dirección** y el **teléfono** no están incluidos ya que no son relevantes para la navegación. El usuario puede seleccionar los contactos mediante una búsqueda basada en un campo de entrada. El resultado de la búsqueda es o bien una lista de contactos que coincidan con la consulta o un mensaje que

indica ningún resultado. la clase de navegación **AddressBook** se identifica como punto de entrada de la aplicación web con la etiqueta **{isHome}**. La actualización de contenido, es decir, la modificación de los datos de contacto, al eliminar un contacto o añadir un nuevo contacto, se modela con nodos adicionales en el modelo de navegación - los llamados nodos de proceso, y se integra en el flujo de navegación por enlaces de proceso.

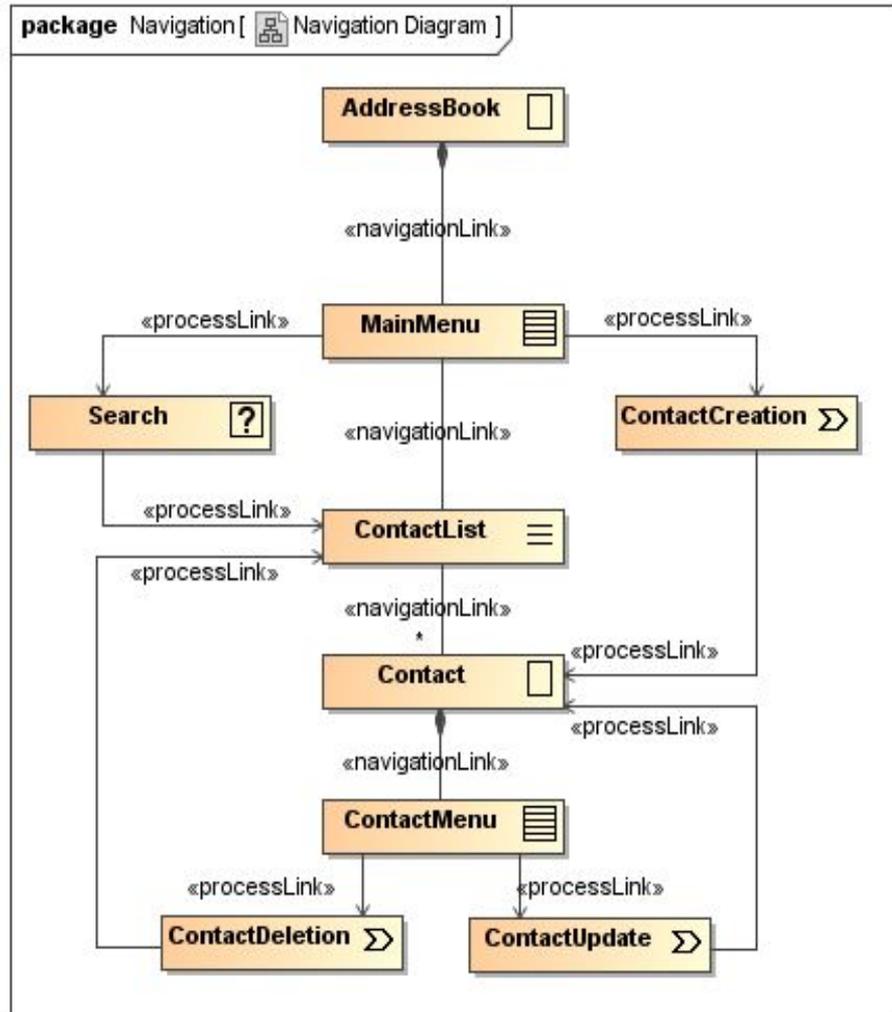


Figura [8]: Modelo de navegación del ejemplo de UWE

2.7.4 Modelo de Presentación

El modelo de presentación permite crear una vista de la estructura de la interfaz de usuario de la aplicación Web y de todos los elementos que permiten la navegación o ejecución de acciones dentro de la misma. Está muy relacionado con los elementos de las interfaces definidas en HTML. Estos elementos también están definidos como estereotipos de UML. Se distinguen dos vistas diferentes:

Estructura de Vista: Muestra la estructura del espacio de presentación. Podemos considerar los siguientes elementos semánticos agregados al modelo de clases:

- <<**presentationClass**>>: son estereotipos estructurales utilizados para representar páginas Web o porciones de páginas Web, contando con una composición jerárquica de elementos de interfaz de usuario
- <<**presentationGroup**>>: es otro elemento estructural de presentación, y este estereotipo contiene un grupo de elementos <<presentationClass>>.
- <<**presentationPage**>>: es el elemento de presentación de mayor jerarquía, indicando una ventana de un navegador y por tanto conteniendo diversos <<presentationClass>> y <<presentationGroup>>.

Interfaz de Usuario (UI por sus siglas en inglés de User Interface): presenta detalles acerca de los elementos de la interfaz de usuario dentro de las páginas. Se mencionan los estereotipos que sirven para agrupar elementos de interface de usuario: <<**anchor**>> <<**button**>> <<**text**>> <<**image**>> <<**textInput**>> <<**selection**>> <<**form**>>.

Nombres de estereotipos y sus iconos.

 presentationGroup	 presentationPage
 text	 textInput
 anchor	 fileUpload
 button	 image
 inputForm	 customComponent
 presentationAlternatives	 selection

Este diseño se guía por algunas reglas:

- Crear una clase de presentación por cada clase de navegación que se presente en el modelo de estructura de navegación.
- Crear una clase de presentación por cada clase de menú, índice, visita guiada o consulta.
- Como soporte de la navegación se debe crear una clase de presentación.
- Adicionar enlaces a las clases de presentación.
- Señalar qué elementos deberán presentarse juntamente al usuario en alguna ventana o sección.
- Adicionar si son necesarias expresiones OCL
- Crear escenarios mediante serie de bocetos, representando sucesión de vistas de interfaz de usuario
- Por medio de los diagramas de presentación se puede indicar qué clase de navegación y de procesos pertenecen a una página web.

En la figura[9] se muestra el modelo de presentación para el ejemplo, con un formulario contenedor, seleccionado con el fin de proporcionar una representación más intuitiva de las páginas. La página de la libreta de direcciones contiene una introducción, un

campo de búsqueda, un botón y una lista de contactos. Para cada contacto, se muestran los campos correspondientes de correo electrónico, los teléfonos y direcciones.

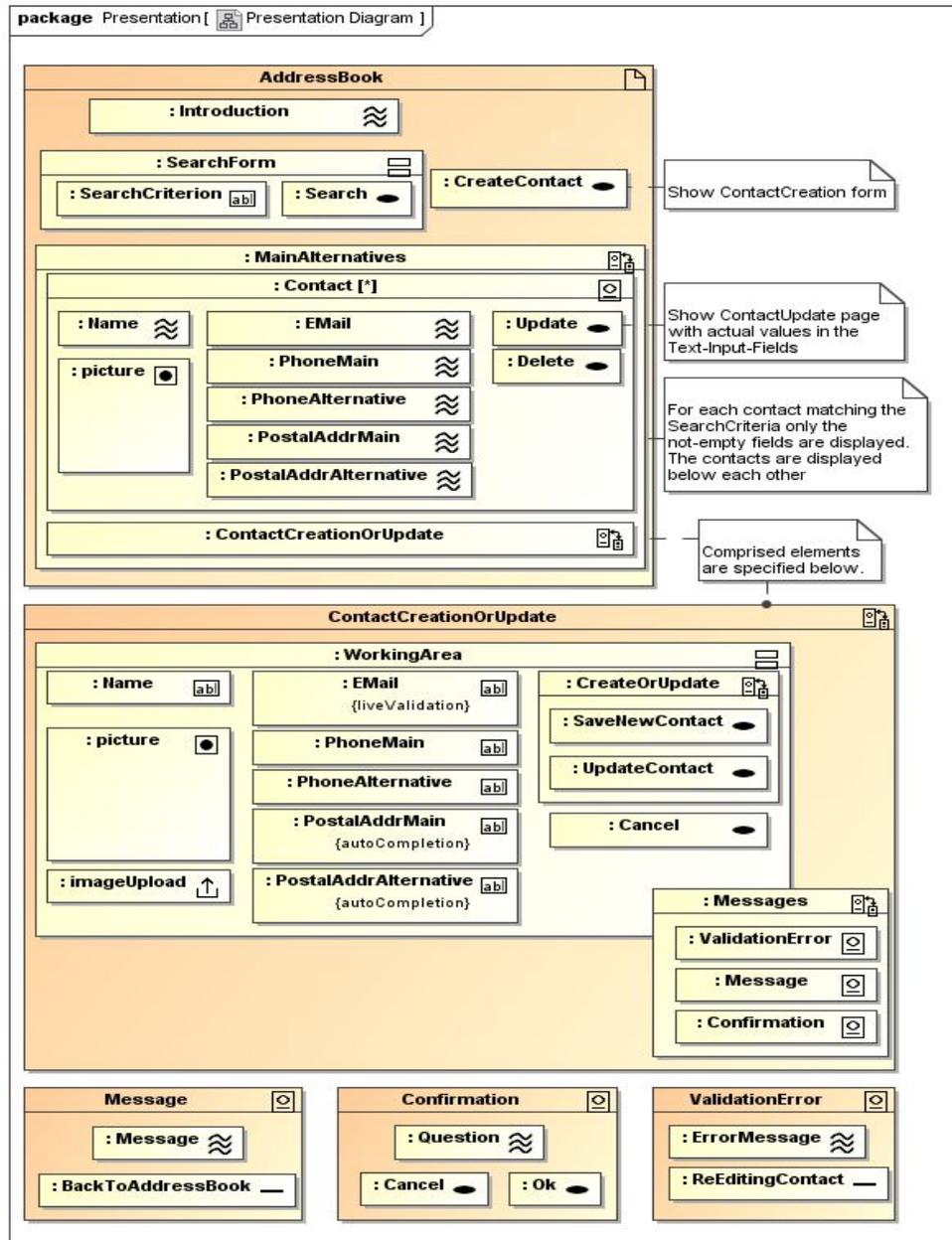


Figura [9]: Modelo de Presentación del ejemplo de UWE.

2.7.5 Modelo de Procesos

El modelo de proceso muestra el aspecto que tienen las acciones de nuestras clases de proceso. El Modelo de Proceso comprende:

- el **Modelo de Estructura del Proceso** que describe las relaciones entre las diferentes clases de proceso y

- el **Modelo de Flujo del Proceso** que especifica las actividades conectadas con cada «*processClass*».

2.7.5.1 Modelo de Estructura de Proceso

El modelo de estructura de procesos contiene un grupo de clases de procesos que describen a los procesos correspondientes, mencionados previamente en el modelo de navegación. Estos apuntan a mostrar mayormente la funcionalidad de los procesos necesarios de nuestro ejemplo, como puede observarse en la figura[10], se han agregado otras clases para expresar, que las tres operaciones requieren una confirmación (ver figura [9] diagrama de presentación) con una pregunta. Esto significa que si un usuario quiere borrar un contacto, un mensaje será mostrado, el cuál deberá ser confirmado con un *ok* para que el contacto sea borrado. **ContactCreation** and **ContactUpdate** funcionan en forma similar, ambos heredan de la clase abstracta **ContactProcessing**, asegurando que los campos de texto, que son atributos de **ContactDataInput** contienen valores válidos (por ejemplo podemos pensar en prohibir un nombre en blanco para prevenir entradas inservibles en la base de datos). Una vez que los datos han sido validados y no hay errores de validación (**ValidationError**) la página de confirmación es presentada al usuario.

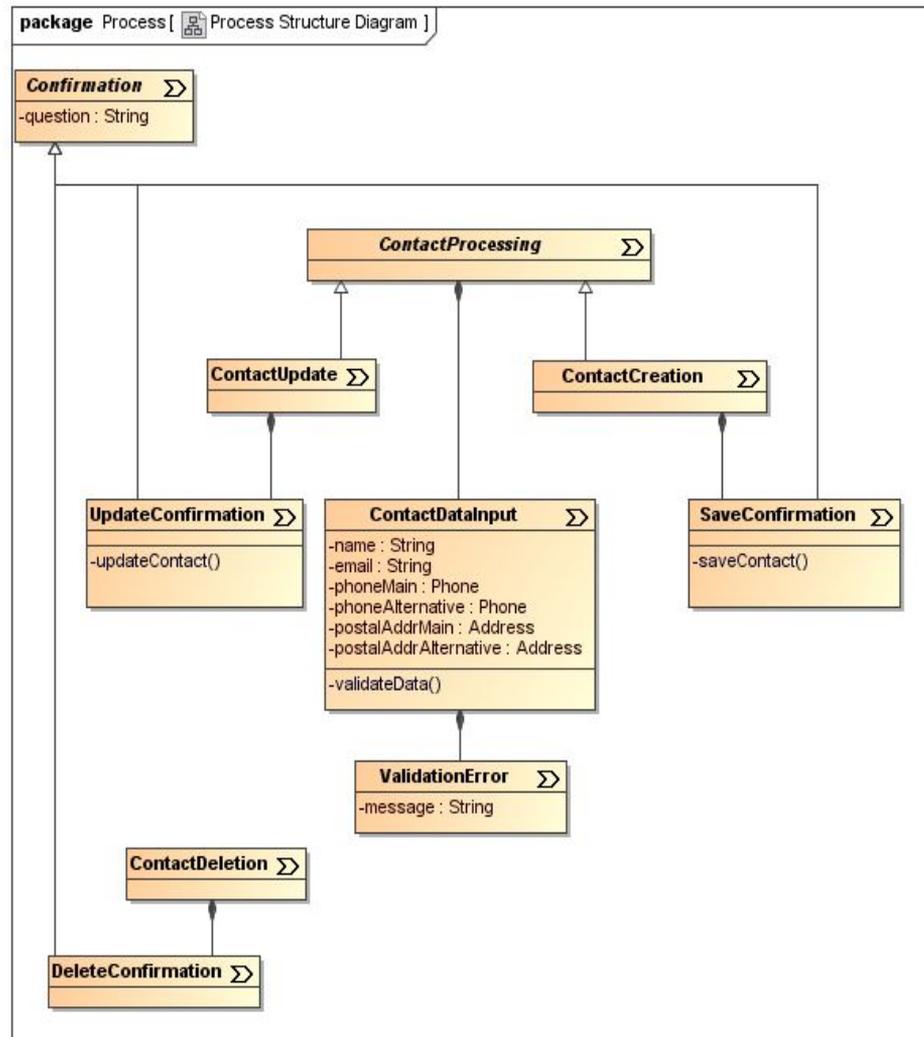


Figura [10]: Modelo de Estructura de Procesos del ejemplo de UWE

2.7.5.2 Modelo de Flujo del Proceso

Un flujo del proceso (flujo de trabajo) es representado como un diagrama de actividades, describiendo el comportamiento de una clase de proceso.

Nombres de estereotipos y sus iconos

 «user Action»

 «system Action»

El estereotipo «**user Action**» es usado para indicar interacciones de usuario con la página web iniciando un proceso o respondiendo a un requerimiento explícito de información. Por el contrario, «**system Action**» describe acciones, que son ejecutadas por el sistema.

En la figura [11] se muestra un ejemplo de lo que sucede en detalle, cuando el usuario navega a una clase de proceso (por ejemplo **ContactCreation** de nuestro ejemplo).

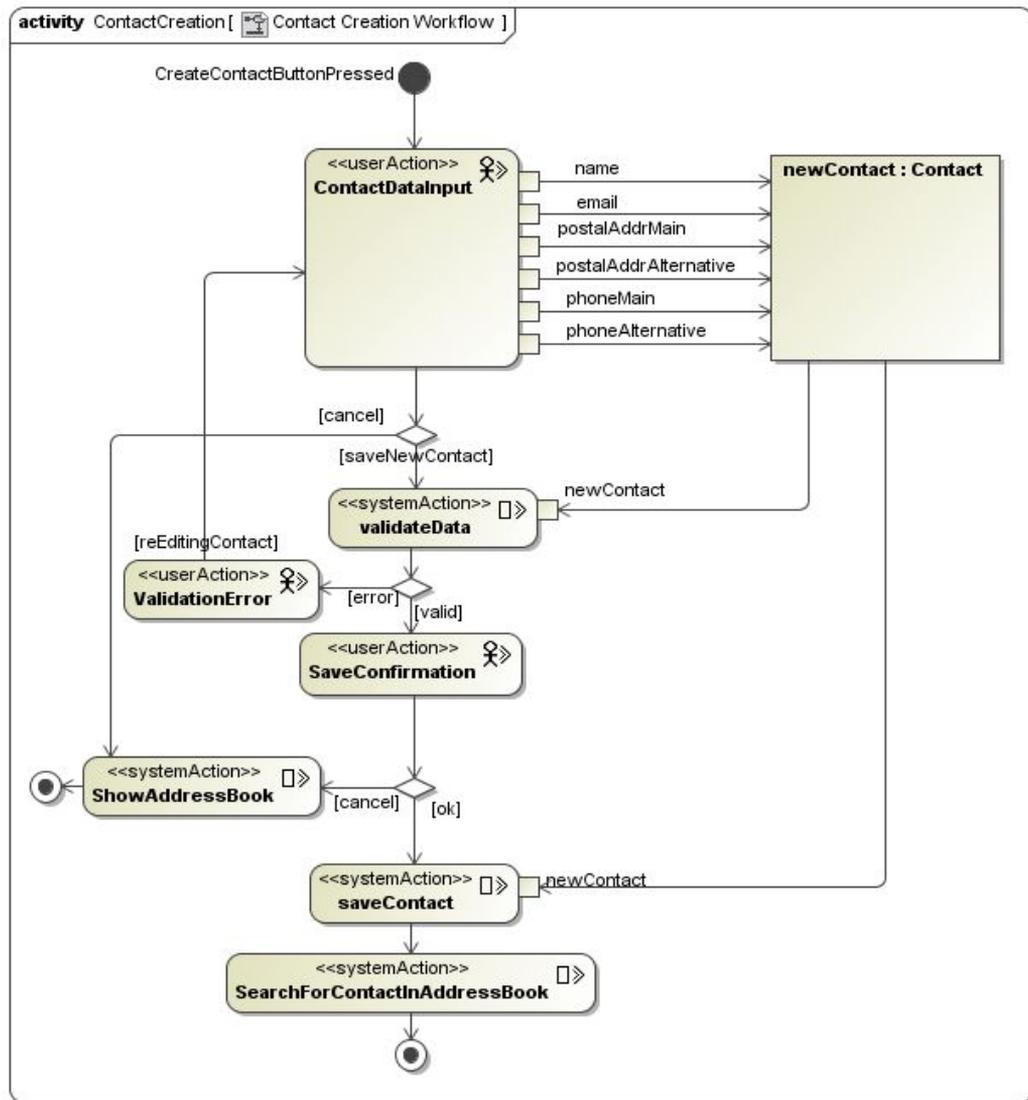


Figura [11]: Modelo de Flujo del Proceso ContactCreation del Ejemplo UWE.

2.8. IDENTIFICACIÓN DEL FRAMEWORK

Un Framework es una estructura de software, cuyos componentes son personalizables e intercambiables, para el desarrollo de una aplicación.

Su utilización en el desarrollo de SIARMe persigue los siguientes objetivos: acelerar el proceso de desarrollo; reutilizar el código de funcionalidades comunes en las aplicaciones Web, y proveer de buenas prácticas de desarrollo que propicien la escalabilidad.

El primer paso consiste en la selección de los framework PHP para el desarrollo de aplicaciones como **Zend Framework, Symfony, CakePHP y Kumbia**. Según Google Trends Lab [26], el promedio y la evolución en la tendencia de consultas referidas a frameworks PHP se muestran en la Figura [12]

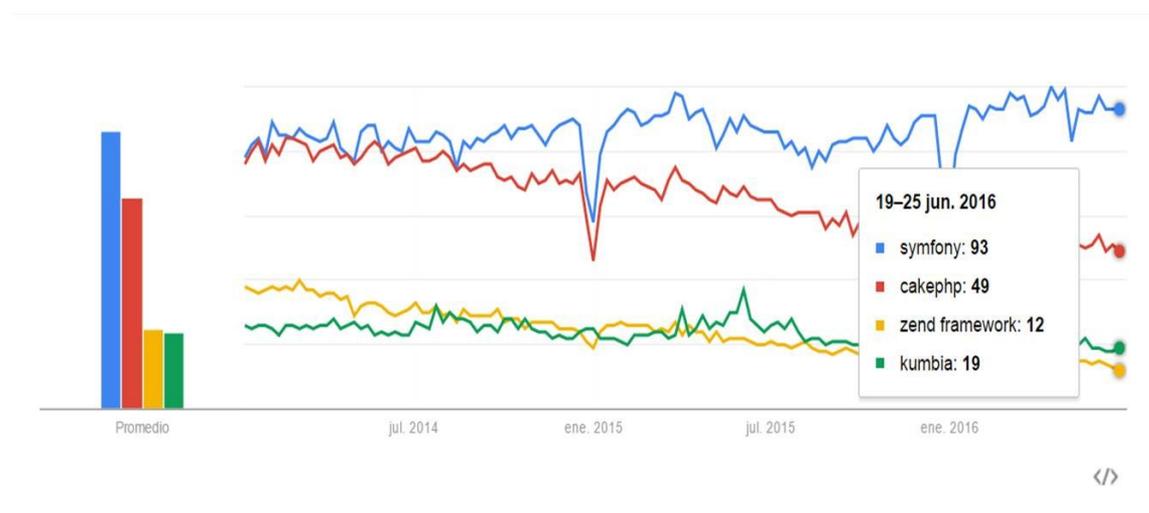


Figura [12]: Tendencia de Consultas de Framework PHP

Se observa que los Framework más consultados en los últimos meses son: Symfony [27], seguido de CakePHP [28] debido a las preferencias de los usuarios de la comunidad PHP por investigar sobre sus últimas características.

2.8.1 Características requeridas del Framework

1. Ser compatible con PHP5.
2. Poseer un patrón de Arquitectura de Software MVC.
3. Simplificación del acceso a distintos tipos de base de datos.
4. Poseer soporte del Modelo de Datos ORM (Objeto-Record Mapper).
5. Utilizar motor de plantillas.
6. Incluir caché de objetos o alguna otra forma de memoria caché.
7. Incorporar reglas de validación o filtros de entrada.
8. Poseer un módulo para el manejo de autenticación de usuarios.
9. Poseer otros módulos, como RSS, PDF, etc.
10. Documentación oficial en línea en español.
11. Implementación sencilla.
12. Entorno intuitivo y fácil aprendizaje.

2.8.2 Evaluación de los Framework PHP

A continuación se indicarán las características ponderadas en una escala del 1 al 5 como se observa en la Tabla [1], considerando los objetivos perseguidos y los framework mencionados que cumplen con dichas características. La fórmula matemática para determinar cuál es el más adecuado es: $x = \sum (\text{Peso}_i \times \text{Valor}_i)$ siendo el Peso_i la ponderación dada a las características evaluadas y el Valor_i si posee o no la característica.

Característica	1	2	3	4	5	6	7	8	9	10	11	12	Total
Peso _i	5	5	4	3	3	3	4	3	3	4	5	3	45
<i>CakePHP</i>	1	1	1	1	0	1	1	1	1	0	1	1	38
<i>Zend Framework</i>	1	1	1	1	0	1	1	0	1	1	1	0	36
<i>Symfony</i>	1	1	1	1	1	1	1	1	0	1	1	0	39
<i>Kumbia</i>	1	0	1	1	1	1	1	1	1	1	0	1	35

Tabla [1]: Evaluación de framework PHP

La máxima puntuación de Symfony no determina su implementación como framework, para ello se consideraron además, las siguientes razones [28]:

- Fácil de instalar y configurar en la mayoría de plataformas.
- Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador sólo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales y adaptable a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Fácil de extender, lo que permite su integración con las bibliotecas de otros fabricantes.
- Una potente línea de comandos que facilitan generación de código, lo cual contribuye a ahorrar tiempo de trabajo.
- Symfony utiliza una licencia MIT, con la que puedes hacer aplicaciones web comerciales, gratuitas y/o de software libre.
- Se puede controlar hasta el último acceso a la información e incluye por defecto protección contra ataques XSS y CSRF.

2.8.3 Patrón de Diseño Modelo-Vista-Controlador (MVC)

El patrón MVC (Model/View/Controller) fue descrito por primera vez en 1979 e introducido como parte de la versión Smalltalk-80 del lenguaje de programación Smalltalk.

Fue diseñado para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos. Sus características principales están dadas por el hecho de que, el Modelo, las Vistas y los Controladores se tratan como entidades separadas; esto hace que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas [29].

Esta arquitectura presenta varias ventajas:

- Separación clara entre los componentes de un programa; lo cual permite su implementación por separado.

- Interfaz de Programación de Aplicaciones API (Application Programming Interface) muy bien definida; cualquiera que use el API, podrá reemplazar el Modelo, la Vista o el Controlador, sin aparente dificultad.
- Conexión entre el Modelo y sus Vistas dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.

El **Modelo**: es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la **Vista** aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al **Modelo** a través del **Controlador**.

El **Controlador**: responde a eventos (usualmente acciones del usuario) e invoca peticiones al **Modelo** cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). Se podría decir que el **Controlador** hace de intermediario entre la **Vista** y el **Modelo**.

La **Vista**: presenta el **Modelo** (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho **Modelo** la información que debe representar como salida.

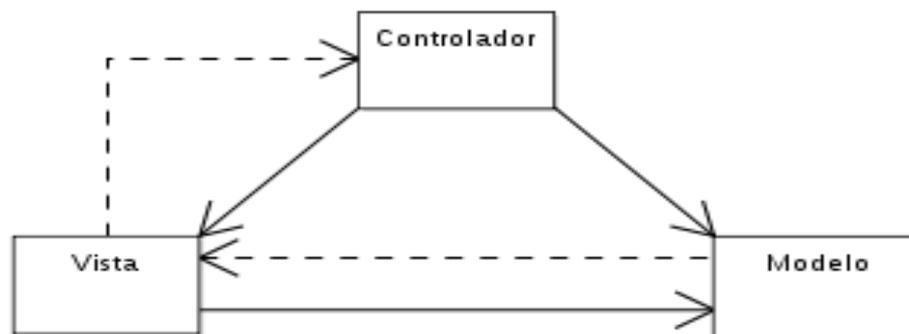


Figura [13]: Un diagrama sencillo que muestra la relación entre el modelo, la vista y el controlador.

2.8.4 Arquitectura MVC empleada en el Prototipo SIARME

Los objetivos de esta arquitectura de software son la reutilización del código y la mejora del mantenimiento del software. Así, para el desarrollo del prototipo SIARME se empleará el framework Symfony2 [28] basado en la arquitectura MVC con el fin de lograr los objetivos anteriormente mencionados.

En este framework cada parte del patrón MVC constituye un sistema de varios componentes, como Doctrine, Twig y Synfony separados en las tres capas respectivas de la arquitectura como se observa en la figura 14:

En la capa del **Modelo** se encuentra Doctrine que se encarga de la persistencia y lectura de información hacia y desde una base de datos. Symfony viene integrado con la biblioteca Doctrine ORM (Object-Relational mapping: mapeador de objetos-relacional escrito en PHP que proporciona una capa de persistencia para objetos PHP). Doctrine está

totalmente desconectado de Symfony y utilizarlo es opcional. Permite asociar objetos a una base de datos relacional (tal como MySQL, PostgreSQL o Microsoft SQL). También permite persistir datos en MongoDB utilizando la biblioteca ODM (object document mapper) de Doctrine.

En la capa de la **Vista** se encuentra TWIG que es un motor y lenguaje de plantillas para PHP muy rápido y eficiente. En Symfony se recomienda utilizar Twig para crear todas las plantillas de la aplicación. No obstante, se puede escribir las plantillas con código PHP normal y corriente. La sintaxis de Twig se ha diseñado para que las plantillas sean concisas, seguras, fáciles de leer y de escribir.

Finalmente, en la capa de **Controlador** se encuentra Symfony que es el puente entre las dos capas anteriores, donde básicamente su funcionamiento interno consiste en recibir las peticiones del usuario desde la **Vista**, y llevar a cabo todas las acciones necesarias para obtener los resultados desde el **Modelo** y enviarlos como respuesta a la **Vista** para que muestre los resultados al usuario.

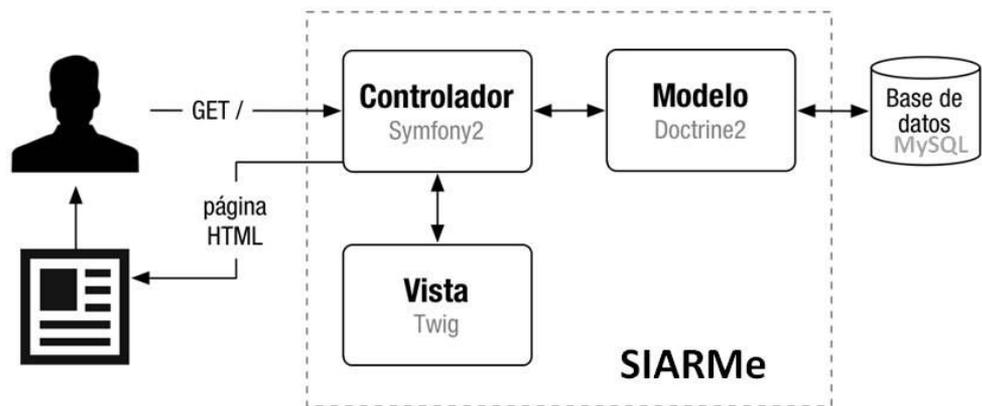


Figura [14]: Esquema simplificado de la arquitectura interna de SIARMe.

CAPÍTULO III: ANÁLISIS DE REQUISITOS DEL PROTOTIPO SIARME

3.1 INTRODUCCIÓN

El Análisis de Requisitos es el primer paso hacia el diseño formal del Prototipo de SIARMe. En este capítulo se especificarán los requisitos del sistema mediante los siguientes documentos:

- Especificaciones de Requisitos del Software ANSI/IEEE 830 en la sección 3.2.
- Análisis de Requisitos Empleando la Metodología UWE en la sección 3.3.
- Descripción Textual de Casos de Uso del del Sistema en la sección 3.4

3.2 ESPECIFICACIÓN DE REQUISITOS

Esta especificación se ha estructurado en base a las instrucciones dadas por el estándar IEEE Práctica Recomendada para Especificaciones de Requisitos del Software (ERS) ANSI/IEEE 830, 1998, donde las secciones omitidas se consideran no aplicables. Esta especificación se compone de tres módulos importantes donde se proporcionará una visión de las pautas del documento. El primero de ellos, es la descripción general, donde se especifica los recursos del documento ERS; mientras que el segundo módulo se centraliza en la descripción del sistema, determinando las funciones y los datos asociados sin entrar en excesivos detalles. Y por último, el tercer módulo es aquél en la que se definen detalladamente los requisitos que debe satisfacer el sistema.

3.2.1 Propósito

El propósito de la presente ERS es formalizar los requisitos que el Prototipo del Sistema basado en Web que debe cumplir en base a las necesidades de la institución previamente analizadas. Esta especificación servirá de guía para el diseño del prototipo, detallando los requisitos en base a la especificación de los casos de uso, junto a otros documentos adicionales según el modelado en UWE basado en UML.

3.2.2 Ámbito del Sistema

El Prototipo de Sistema de Información Administrativa Basado en WEB (SIARMe) comprenderá la centralización de la información y servirá de soporte en las tareas administrativas de gestión de expedientes de los Departamentos Autoseguro y Despacho que interactuará con el sistema informático del departamento Ausentismo de la DPSRM englobando las siguientes funcionalidades:

- Gestión de Usuarios.
- Gestionar Expedientes.
- Gestionar Información de Documentos Administrativos y Médicos.

3.2.3 Personal Involucrado

Se describen los roles del personal involucrado que brindarán información acerca de las tareas que se realizan según el área del departamento y la información que requiere para realizar su tarea.:

Roles Involucrados
Agente Público
Empleado Administrativo del Área Mesa de Entrada.
Empleado Administrativo del Área Administrativa.
Médico Auditor del Área Auditoría Médica

3.2.4 Definiciones, Acrónimos y Abreviaturas

Agente Público: se refiere a las personas que tienen una relación de dependencia laboral con el Estado Provincial (Escuelas, Hospitales, etc).

Mesa de Entradas y Salidas: Conformada por empleados administrativos, se encarga de llevar el registro de ingreso y salidas de expedientes y del archivo físico de estos. Además, informa a los agentes públicos interesados del estado de sus trámites.

Área Administrativa: Conformada por empleados administrativos, se encargada de gestionar el curso de expedientes entre las distintas áreas y crear los documentos administrativos pertinentes para la prosecución de trámites.

Auditoría Médica: conformada por un grupo de médicos, está encargada de realizar informes o dictámenes con el objetivo de evaluar el estado o la situación de salud en la que se encuentra el paciente en base al marco legal vigente. Luego, estos documentos se adjuntan los expedientes.

Expedientes: Son los trámites por expedientes iniciadas por los agentes públicos en sus respectivos organismos de trabajo caratulados en Mesa General de Entrada y Salida de la Administración Pública.

Documentos Administrativos: Son aquellos documentos generados en la DPSRM que se agregan a un expediente de los que se distinguen dos clases. Por un lado los documentos administrativos que son: proveídos o pases, informes y citaciones. Por otro lado, los documentos médicos, pueden ser Actas de Junta Médica o Dictámenes Médicos. En cuanto a los tipos de documentos se observa que poseen formatos específicos, cuyas características pueden ser diferentes de acuerdo al propósito y contenido. Estos formatos se definen y reglamentan legalmente en ámbito de la administración pública por lo que perduran en el tiempo. Esta particularidad de los documentos y la rutinariedad de las tareas administrativas hacen posible su automatización.

ERS	Especificación de Requisitos Software
Usuario	Persona que usará el sistema para gestionar procesos

Privilegios	Permiso especial que se le da a los usuarios para que realicen ciertas tareas.
RN	Reglas de Negocio
Apache	programa de servidor HTTP Web de código abierto.
SIARMe	Nombre del sistema.
IEEE	El Instituto de Ingeniería Eléctrica y Electrónica.
JavaScript	Lenguaje de programación interpretado.
MYSQL	Es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual GPL/Licencia comercial.
Hardware	Son todos los dispositivos físicos de la PC (Monitor, Mouse)
Software	Básicamente es el sistema operativo que carga los programas.
UWE	Metodología de desarrollo para aplicación web orientada a objetos.
PHP	Lenguaje de programación orientado a la web.
DQL	Es el lenguaje de consulta de Doctrine.
TWIG	Es un motor de creación de plantillas para utilizar con PHP.
YAML	es un lenguaje de marcado "ligero" que permite especificar estructuras (tales como arrays) con menos caracteres que XML, de forma sencilla.

3.2.5 Referencias

- IEEE Recommended Practice for Software Requirements Specification. ANSI/IEEE std. 830- 1998.
- UWE UML (UML-Based Web Engineering) - <http://uwe.pst.ifi.lmu.de/index.html>

14/01/2016

3.2.6 Descripción General

En esta sección se realiza una descripción de alto nivel de SIARMe, focalizándose en las funciones que realizará, así como aquellas restricciones generales que afectan al mismo.

- **Interfaces de usuario:** La interacción con el usuario será a través de un Explorador Web por lo que las aplicaciones del sistema mostrarán en algunas de las páginas formularios que los usuarios tendrán que completar, mediante cajas de texto o de listas y cuadros de opciones, etc. Existirán además hipervínculos a los diferentes elementos del menú principal desde todas las páginas. El diseño de las páginas debe ser amigable y sencillo para los usuarios, de tal manera que se simplifique su utilización.
- **Interfaces de hardware:** Para que el sistema funcione correctamente, el usuario debe disponer de un equipo de cualquier característica que cuente con un Explorador

Web y estar interconectado con el servidor Web y de base de datos a través de la intranet de la institución.

- **Interfaces de Software:** El servidor deberá contar con un motor de base de datos MySQL para la administración del sistema de base de datos. Se requiere igualmente de un servidor Web para alojar las páginas Web y sus recursos .
- **Operaciones:** Los usuario previamente cargados al sistema, utilizarán una página de autenticación de usuarios. Para ingresar al sistema, accederán con un nombre de usuario y una contraseña, de los que dependen los permisos para acceder a los módulos correspondientes ya sea del Dpto. Autoseguro o Despacho, visualizando únicamente la información que le concierne al área respectiva.

3.2.6.1 Perspectiva del producto

El Prototipo SIARMe interactuará con los productos de software que a continuación se mencionan:

- **Sistema Informático del Departamento Ausentismo:** Compartirá la base de datos y las tablas (Datos de Agentes Públicos y sus respectivos datos laborales, Organismos, Departamentos y Localidades) comunes en ambos sistemas.
- **Servidor de WEB Apache**
- **Base de datos,** Base de Datos Relacional en MySQL.

El Prototipo SIARMe permitirá por un lado, el rediseño de la arquitectura del sistema de información actual de la DPSRM, permitiendo optimizar el uso de los recursos tecnológicos disponibles en la institución por medio de las tecnologías de internet como se observa en la figura[15]. Por otro lado, permitirá centralizar la información administrativa en una base de datos común, ya que interactuará con el sistema informático del departamento **Ausentismo (SIA)**.

El SIA es una aplicación distribuida (cliente / servidor). Por una parte, del lado del cliente se ejecuta el software de escritorio que permite el registro de las licencias por razones de salud de los Agentes Públicos y, por otra parte, del lado del servidor se ejecuta el gestor de base de datos MySQL, en la que se almacena toda la información de los agentes públicos (datos personales, datos laborales, licencias, etc). Esta base de datos deberá ser reconfigurada para ajustarla al Prototipo SIARMe y compartir dicha información, sin que afecte la funcionalidad del sistema antes mencionado.

El SIARMe será un Prototipo de una Aplicación Web ya que estará alojado en un Servidor Web Apache que se encargará de procesar las **peticiones HTTP** realizadas por el cliente y devolver las consultas mediante **respuestas HTTP**. Del lado del cliente los usuarios podrán acceder a SIARMe a través de internet o una intranet mediante un navegador que se encargará mostrar las **respuestas HTTP** devuelta por el servidor.

Por último, debido a las ventajas que ofrecen los Sistemas Basados en Web frente al software tradicional, permitirá el acceso a la información vía **internet** por medio de un navegador web (Chrome, Firefox, IExplorer, etc), facilitando la movilidad y el acceso a la información a los empleados de los **departamentos Autoseguro y Despacho**, como así

también brindará información a los agentes públicos, quienes podrán realizar consultas respecto al estado de los trámites por expedientes.

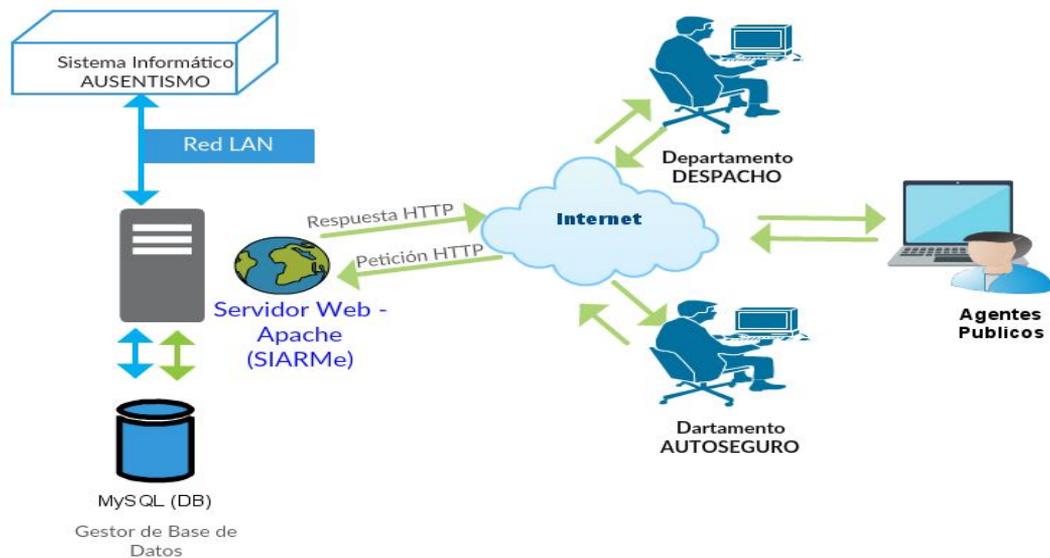


Figura [15]: Interacción de SIARMe con el Sistema Informático de Ausentismo.

3.2.6.2 Características de los Usuarios

Los usuarios de SIARMe podrán ser tanto los empleados de la DPSRM (Administrativos, y/o Médicos), como los agentes públicos. Es por ello, que SIARMe deberá ofrecer una interfaz usable (fácil de entender presentando funciones y menús sencillos de usar), adaptable según el dispositivo que se utilice (Pc, portátiles, tabletas, etc) y segura (establecer los controles de seguridad de Autenticación y Autorización de usuarios según el rol de los mismos).

3.2.7 Funciones del Producto

El Prototipo SIARMe comprende las funcionalidades para el soporte de la información administrativa de los Departamentos Autoseguro y Despacho. Las funciones que realizará el producto se clasifican en:

Gestionar Usuarios: Los usuarios compartirán las interfaces de SIARMe, pero es necesario que el sistema posea reglas de Autenticación, Autorización y Registro de usuarios con propósito de presentar formularios y páginas con la información correspondiente a cada usuario de acuerdo con sus privilegios y según el área y departamento al que pertenezca.

Gestionar Expedientes: Los expedientes deben ser registrados en el sistema para tener su correspondiente registro electrónico y poder realizar el seguimiento automático de éste, para saber en qué área se encuentra, o bien, su estado.

Gestionar Información de Documentos: Una vez creado el registro electrónico de un expediente, los usuarios podrán registrar la información de documentos y generarlos automáticamente en formatos preestablecidos.

3.2.7.1 Restricciones

- SIARMe deberá proporcionar exclusivamente una interfaz Web, siempre que exista una conexión a través de internet o intranet.
- En la etapa de construcción de SIARMe se utilizará el framework Symfony y sus librerías en base a la arquitectura de software Modelo-Vista-Controlador.
- En cuanto a los lenguajes, se utilizarán para el modelado de interfaces Web: Html, CSS, JavaScript y para la lógica de la aplicación: PHP, TWIG, DQL, YAM.
- Interactuará con otro sistema informático por lo que utilizará la Base de Datos en MYSQL ya existente y a lo sumo agregará nuevas estructuras de almacenamiento.

3.2.7.2 Requisitos No funcionales

Renderizado de pantalla: Para ofrecer una presentación de los elementos en pantalla de forma rápida, segura y favorable a la vista del usuario, se utilizará el renderizado dinámico de todos los ítems que se presentan en ella.

Enlaces dinámicos: Mediante un menú de navegación el usuario podrá movilizarse por las opciones del sistema.

Optimización del hardware: Se realizará un diseño que permita ir hacia las bases de datos la menor cantidad de veces posible, ahorrando el tiempo de lectura y escritura en el disco duro y de procesador.

3.2.7.3 Requisitos Funcionales

Los requisitos funcionales identificados para SIARMe, definen el conjunto funciones que el sistema debe cumplir.

1. SIARMe debe establecer un modelo de seguridad específico de acuerdo a los roles que se detallarán a continuación:

Nº	Actores	Descripción
1.	Usuario no autenticado	Representa a un usuario que no se ha autenticado frente al sistema.
2.	Usuario autenticado	Representa a un usuario que se ha autenticado en el sistema.
3.	Empleado	Es un Usuario Autenticado . Representa al rol del usuario del <i>Área Mesa de Entrada</i> .
4.	Administrativo	Es un Usuario Autenticado . Representa al rol del usuario del <i>Área Administrativa</i> .
5.	Médico	Representa al rol del usuario del <i>Área Auditoría Médica</i> .

2. El actor **Empleado** requiere registrar los datos principales que figuran en la carátula de un expediente, consultar, eliminar o modificar la información de este.

3. El actor **Empleado** requiere registrar los datos de ingreso de un expediente (Trámite).
4. El actor **Usuario No Autenticado** requiere consultar el estado de expedientes y sus trámites.
5. El actor **Administrativo** requiere registrar la información de documentos administrativos, consultar, eliminar o modificar la información de este.
6. El actor **Médico** requiere registrar la información de documentos médicos, consultar, eliminar o modificar la información de este.

3.2.7.4 Requisitos que Representan Reglas de Negocio.

En esta sección se detallan las restricciones que se deben tomar en cuenta junto con los requisitos funcionales.

RN1: Solo el actor **Empleado** puede eliminar o modificar un expediente o un trámite.

RN2: No debe poder eliminarse un expediente que tenga al menos un documento asociado.

RN3: No debe poder eliminarse un trámite que contenga al menos un documento asociado.

RN4: Cuando un expediente tenga estado "iniciado", no se deben poder ingresar nuevos trámites.

RN5: Si el estado de un trámite es "concluido" no se deben poder crear, eliminar, ni modificar la información de los documentos asociados a este.

RN6: Cuando un el Actor **Médico** o **Administrativo** crea un documento, este debe estar disponible para el resto de los usuarios con el mismo rol pudiendo realizar las operaciones de consulta y modificación.

RN7: El sistema debe presentar al **Usuario Autenticado** una lista de expedientes según el rol (Empleado, Médico y Administrativo). Por Ejemplo parar en rol **Administrativo** y **Médico** deben listarse aquellos expediente con estado iniciado.

RN8: El usuario **Empleado** no puede consultar la información de Documentos.

3.3 ANÁLISIS DE REQUISITOS EMPLEANDO LA METODOLOGÍA UWE.

La Metodología UWE es un proceso del desarrollo para aplicaciones Web enfocado sobre el diseño sistemático, la personalización y la generación semiautomática de escenarios que guíen el proceso de desarrollo de una aplicación Web. UWE describe una metodología de diseño sistemática, basada en las técnicas de UML, la notación de UML y los mecanismos de extensión de UML.

En el marco de UWE es necesario la definición de un perfil UML (extensión) basado en estereotipos con este perfil se logra la asociación de una semántica distinta a los diagramas del UML puro, con el propósito de acoplar el UML a un dominio específico, en este caso, las aplicaciones Web [24].

Entonces, el Análisis de los Requisitos de una aplicación Web en UWE permite plasmar los requisitos funcionales de la aplicación mediante un modelo de casos de uso

UML empleando los elementos Actores y Caso de Uso. Un caso de uso describe la secuencia de interacciones (escenarios) que se producen entre el sistema y los actores (personas, otros sistemas, etc.) del mismo para realizar una determinada función o un objetivo específico [30]. En síntesis, un caso de uso es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso, es decir, una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.

En UWE se distinguen tres tipos de casos de uso estereotipados: navegación, procesos y casos de uso personalizados.

Los casos de uso de navegación <<browsing>> modelan el comportamiento del usuario al interactuar con la aplicación web. Los casos de uso de procesos <<processing>> describen si los datos persistentes de la aplicación son modificados; mientras que los casos de uso personalizados implican la personalización de un sistema web, la cual es desencadenada por el comportamiento del usuario.

Además, todos los elementos de U.M.L. para el modelado de diagramas de casos de uso están disponibles, tales como paquete, relación de generalización, los estereotipos de dependencias <<extend>> e <<include>> entre los casos de uso.

A continuación, se presentan los diagramas de Casos de Uso y sus descripciones para el Prototipo SIARMe empleando la Metodología de desarrollo U.W.E. complementándose con la Especificación de Requisitos de Software.

3.3.1 Modelo de Requisitos del Prototipo SARMe

En la figura[17] se pueden observar el diagrama de caso de uso que representa la funcionalidad completa de SIARMe y su interacción con los Actores. Esta representación se hace a través de las relaciones entre los actores (roles de usuario) y los casos de uso (funcionalidades) dentro del sistema y sus respectivos estereotipos.

El diagrama de casos de uso está conformado por los elementos actor (representado en forma de personaje) , caso de uso (representado en forma de elipse) y sus relaciones (representado en forma de líneas y/o flechas). Los actores se utilizan para modelar los roles usuarios que interactuarán, con SIARMe, que en este caso son (**Usuario No Autenticado, Usuario Autenticado, Empleado, Administrativo y Médico**). Los casos de uso se utilizan para visualizar gráficamente y/o modelar las diferentes funcionalidades que SIARMe tiene que proporcionar, como son: *Autenticar Usuario, Cerrar Sesión, Crear Expediente, Buscar Expediente, etc.* Las relaciones representan las conexiones entre los elementos del modelo, por ejemplo inclusión (<<include>>) y extensión(<<extend>>).

En cuanto a la relación de Actores, en figura[16] se observa la relación de “Herencia” que se emplea para modelar las interacciones entre el sistema y los usuarios según el rol asignado. Por ejemplo el Actor especializado **Empleado** hereda los casos de uso (*Cerrar Sesión, Actualizar Datos de Usuario, Listar Expedientes*) del actor general **Usuario Autenticado**. Los mismo ocurre con los Actores **Administrativo** y **Médico**. Sin embargo las los casos de usos *Crear Expediente, Iniciar Expediente* y *Concluir Expediente* solo están disponibles para el Actor (rol de usuario) **Empleado**.

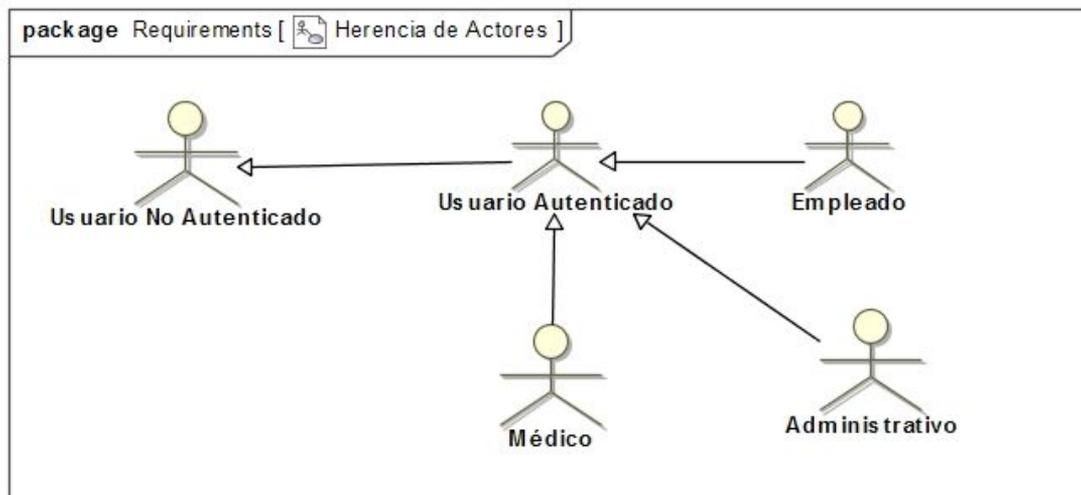


Figura [16]: Herencia de Actores.

En cuanto a la relación entre un actor y un caso de uso, en la figura[17] puede observarse la relación de “Asociación” que modela la participación del actor en un caso de uso determinado y se emplea para indicar las funcionalidades o acciones que estarán disponibles para un rol de usuario determinado.

En cuanto a la relación entre los casos de usos se observan a continuación las relaciones de <<include>> y <<extend>>. La relación <<include>> se emplea para indicar una funcionalidad común en más de un caso de uso, por ejemplo el caso de uso *Nuevo Trámite*, modela la función de registrar los datos sobre el ingreso de un trámite como ser lugar de origen, fecha de ingreso y tipo de trámite. Esta funcionalidad es común en los casos de usos *Iniciar Expediente* y *Crear Expediente* que la incluyan mediante la relación <<include>>. En el primer caso de uso *Iniciar expediente* modela la función de registrar un nuevo trámite para un expediente ya registrado y cambiar su estado a “iniciado” y en el segundo *Crear Expediente* en el cual, por un lado, se modela el la funcionalidad de registrar los datos de un nuevo expediente (letra, número, año, agente iniciador, etc) y por otro lado, en la relación <<include>> de registrar los datos del trámite en *Nuevo Trámite*).

En cuanto a la relación <<extend>> se emplea para añadir una funcionalidad a un caso de uso e indicar cómo se relacionan los Actores con cada caso de uso. De esta manera, se determinan las acciones que pueden realizar los distintos usuarios según el rol. Por ejemplo, el **Usuario Autenticado** se relaciona con el caso de uso *Listar Expediente* que a su vez es extendido por *Listar Trámite*. Pero como puede observarse, únicamente, el Actor **Empleado** se relaciona con los casos de uso *Iniciar Expediente* y *Concluir Expediente*. Es decir que estas funcionalidades solo estarán disponibles para el rol **Empleado**.

En cuanto a los estereotipos se distinguen <<processing>> y <<browsing>>. El estereotipo <<processing>> se emplea para indicar si un caso de uso cambia el estado de la aplicación o si los datos persistentes son modificados. Por ejemplo, los casos de uso etiquetados con <<processing>> *Cerrar Sesión* (modifica el estado de la aplicación al cambiar el estado de **Usuario Autenticado** a la de **Usuario No Autenticado**) y *Actualizar Datos de Usuario* que modela la funcionalidad de actualizar datos del usuario y guardarlos.

Contrariamente, el estereotipo <<browsing>> se emplea para indicar los casos de uso que modelan típicas interacciones de usuario con el sistema sin que los datos persistentes de la aplicación sean modificados. Por ejemplo *Buscar Expediente*, *Listar Expedientes*, *Listar Trámites*, *Listar Documentos*. Por un lado, *Buscar Expediente* modela la búsqueda de expedientes según los parámetros indicado por un usuario. Por otro lado, *Listar Expedientes* modela la presentación de los datos de expedientes mediante una lista, de la misma manera lo hacen *Listar Trámites* y *Listar Documentos*.

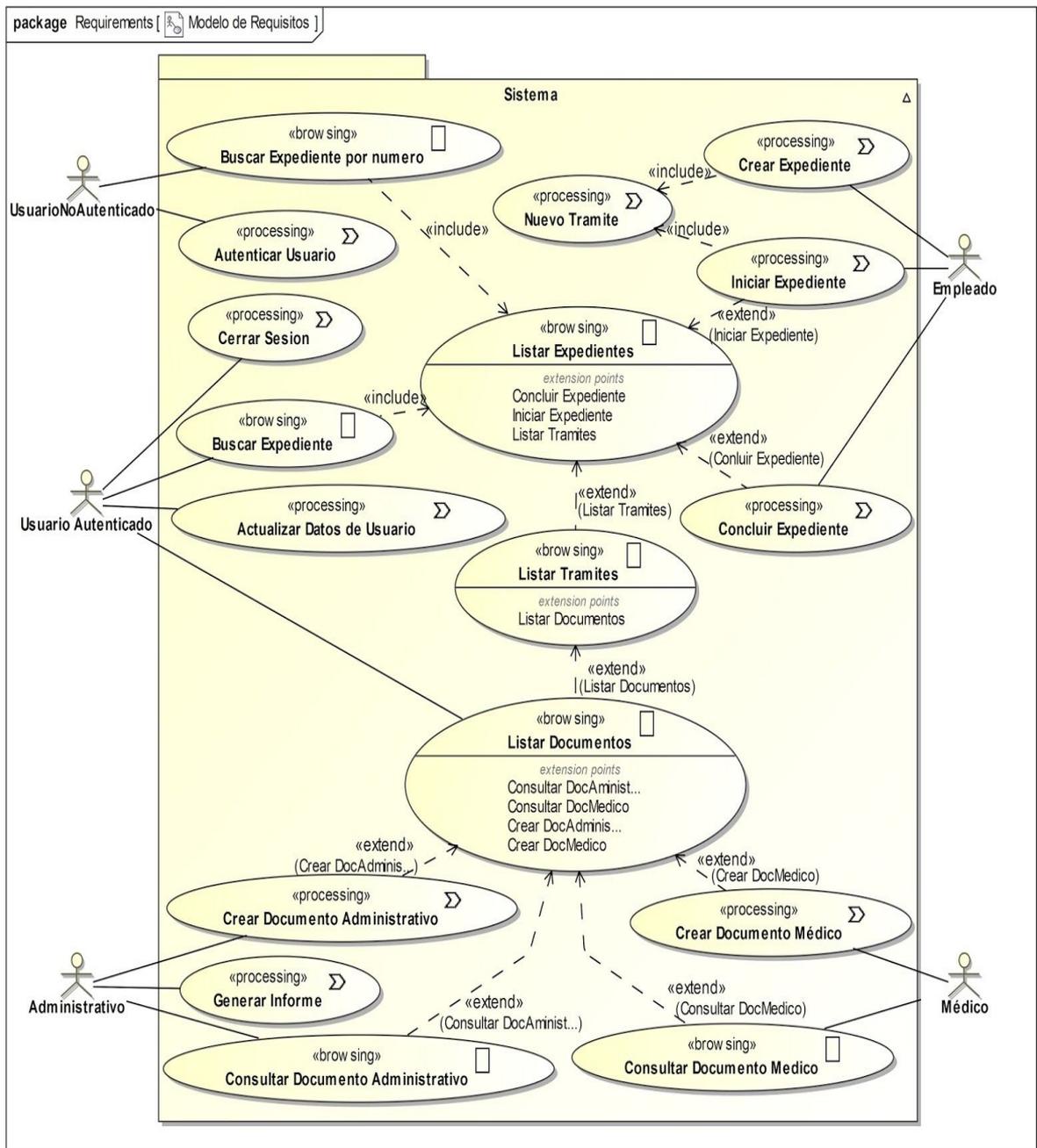


Figura [17]: Diagrama de Casos de uso de SIARMe

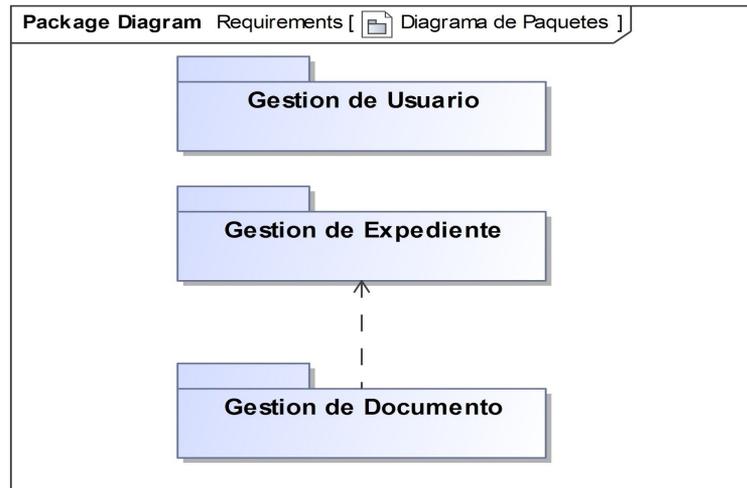
3.4 DESCRIPCIÓN TEXTUAL DE CASOS DE USO

El diagrama de caso de uso nos permite visualizar gráficamente la funcionalidad del Prototipo SIARMe mostrando qué/quién interactúa con el sistema (actores) y cuáles son sus objetivos al hacerlo (casos de uso). Cada caso de uso proporciona uno o más escenarios que indican cómo interactúa el sistema con el usuario para conseguir un objetivo específico.

La descripción textual de un caso de uso permite describir y documentar dicha interacción de cada escenarios mediante la descripción de un flujo de eventos de forma textual e informal, lo suficientemente claro para que lo entienda fácilmente el usuario. Este

flujo está compuesto por un escenario básico (llamado: flujo normal) y escenarios alternativos (subflujos o excepciones) en la interacción. Además se describen las condiciones de inicio y fin del mismo.

Para una mejor organización, los casos de uso se estructuran en paquetes de acuerdo a su funcionalidad y sus actores, los mismos se detallan a continuación:



Figura[18]: Diagrama de Paquetes del Prototipo SIARMe

- Paquete Gestión de Usuario.
 - CU1: Autenticar Usuario
 - CU2: Cerrar Sesión
 - CU3: Actualizar Datos de Usuario
- Paquete Gestión de Expediente.
 - CU5: Crear Expediente
 - CU5: Buscar Expediente por Número
 - CU6: Buscar Expediente
 - CU7: Listar Expedientes
 - CU8: Iniciar Expediente
 - CU9: Listar Trámites
 - CU10: Nuevo Trámite
 - CU11: Concluir Expediente
- Paquete Gestión de Documento.
 - CU12: Listar Documentos.
 - CU13 : Consultar Documento Administrativo.
 - CU14 : Consultar Documento Médico.
 - CU15: Crear Documento Administrativo.
 - CU16: Crear Documento Médico.

- CU17: Generar Informe.

3.4.1 Diagrama de Casos de uso del Paquete Gestión de Usuario

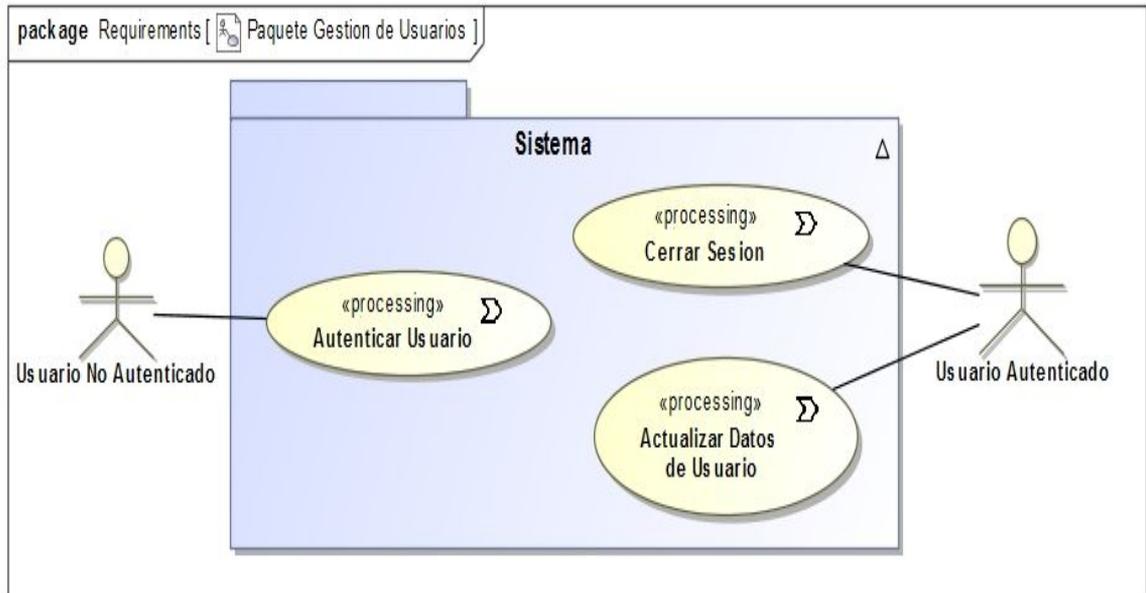


Figura [19]: Diagrama de Casos de Uso del Paquete Gestión de Usuario.

CU1: Autenticar Usuario
Descripción: Permite a los usuarios autenticarse e ingresar al sistema.
Actor: Usuario no autenticado.
Precondiciones: El Actor debe seleccionar la opción Autenticar Usuario .
Flujo Normal: <ol style="list-style-type: none"> 1. El sistema muestra el formulario de ingreso. 2. El actor ingresa el <i>nombre de usuario</i> y <i>contraseña</i> que lo identifica y selecciona ingresar. 3. El sistema valida y verifica la información ingresada. 4. El sistema determina rol y privilegios del actor. 5. El sistema presenta los datos y opciones en pantalla para el actor. 6. El caso de uso finaliza.
Flujo Alternativo: Escenario 5a: Datos inválidos. <ol style="list-style-type: none"> 1. El sistema muestra un mensaje de advertencia. 2. El sistema redirige al actor al paso 3 del flujo normal.
Poscondiciones: El usuario ha sido autenticado por el sistema.

CU2: Cerrar Sesión.
Descripción: Permite a los usuarios autenticados cerrar su sesión en el sistema.
Actor: Usuario Autenticado
Precondiciones: El Actor selecciona la opción Cerrar Sesión .
Flujo Normal: <ol style="list-style-type: none">1. El sistema cierra la sesión de usuario.2. El sistema muestra un mensaje al actor informando que la sesión ha sido cerrada.3. El caso de uso finaliza.
Flujo Alternativo:
Poscondiciones: La sesión de usuario ha sido cerrada por el sistema y debe autenticarse nuevamente.

CU3: Actualizar Datos de Usuario.
Descripción: Permite al actor modificar sus datos personales.
Actor: Usuario Autenticado
Precondiciones: El actor debe seleccionar la opción Actualizar Datos Personales .
Flujo Normal: <ol style="list-style-type: none">1. El sistema muestra y rellena el formulario de usuario con los campos de datos personales del actor editables: <i>Apellido y Nombre, DNI, Domicilio, Teléfono, Teléfono Móvil</i>.2. El actor ingresa los nuevos datos y los guarda.3. El sistema valida los datos y los almacena.4. El sistema muestra un mensaje de operación exitosa.5. El caso de uso finaliza.
Flujo Alternativo: Escenario 4a: Datos inválidos. <ol style="list-style-type: none">1. El sistema muestra un mensaje de advertencia.2. El sistema redirige al actor al paso 2 del flujo normal.
Poscondiciones: Flujo Normal Los datos del actor se modificaron en el sistema.

3.4.2 Diagrama Casos de uso del Paquete Gestión de Expediente

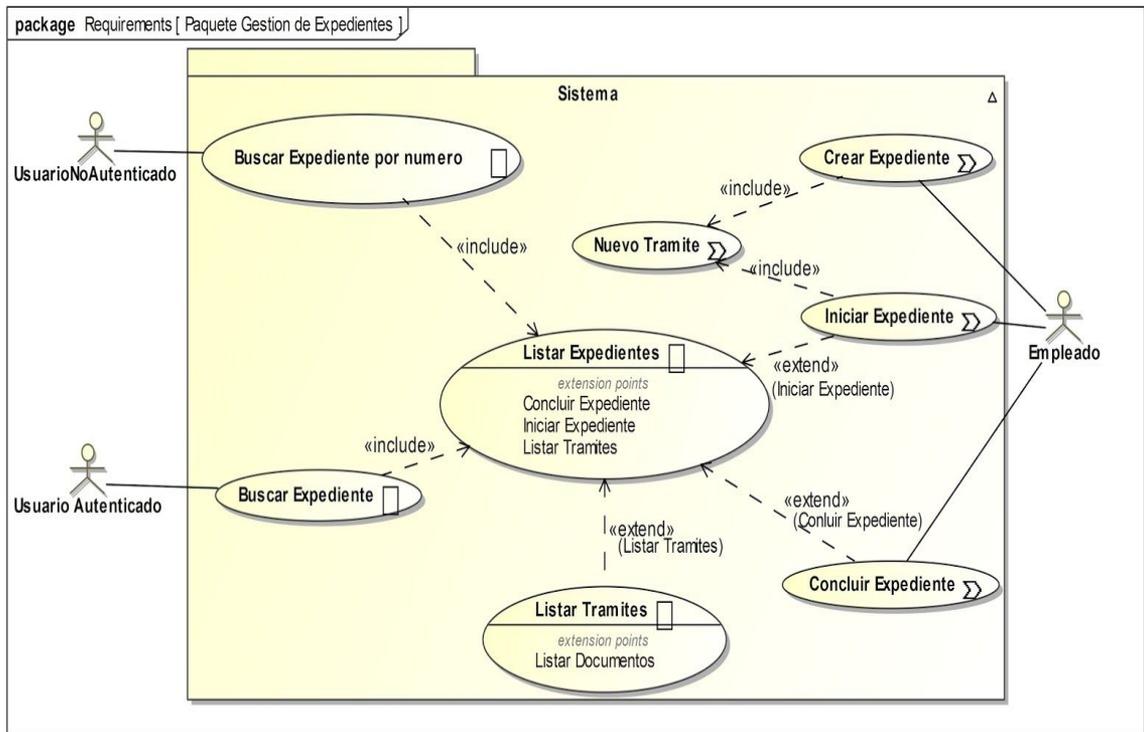


Figura [20]: Diagramas de Casos de Uso del Paquete Gestión de Expedientes

CU4: Crear Expediente
Descripción: Permite agregar un nuevo expediente al sistema.
Actor: Empleado
Precondiciones: El Actor debe seleccionar la opción de Crear Expediente .
<p>Flujo Normal:</p> <ol style="list-style-type: none"> 1. El sistema muestra un formulario de expediente con sus campos correspondientes. 2. El sistema invoca el caso de uso Nuevo Trámite (devuelve un subformulario para que el actor registre los datos de ingreso del expediente). 3. El actor introduce en el formulario de expediente: <i>Letra, Numero, Año, Agente Iniciador, Extracto, Clasificación, Departamento, observaciones</i>. Y en el subformulario: <i>Fecha de ingreso, Organismo de Origen, Tipo de Trámite</i>. y luego los guarda. 4. El sistema valida los datos y los almacena. 5. El sistema muestra el mensaje de operación exitosa. 6. El caso de uso finaliza.
<p>Flujo Alternativo:</p> <p>Escenario 5a: Datos inválidos.</p> <ol style="list-style-type: none"> 1. El sistema muestra un mensaje de advertencia. 2. El sistema redirige al actor al paso 4 del flujo normal.
<p>Poscondiciones:</p> <p>Flujo Normal: El nuevo expediente existe en el sistema.</p>

CU5: Buscar Expediente por Número

Descripción: Permite al actor buscar expedientes por *número y año*.

Actor: Usuario No Autenticado

Precondiciones: El sistema muestra al Actor el formulario de búsqueda con la opción **Buscar Expediente Número**.

Flujo Normal:

1. El Actor introduce el *número* de expediente a buscar.
2. El sistema realiza la búsqueda e invoca al caso de uso **Listar Expedientes** (retorna una lista de expedientes encontrados.)
3. El caso de uso finaliza.

Flujo Alternativo:

Escenario 2a: Búsqueda sin resultados.

1. El sistema muestra un mensaje al actor informando que no hay ningún expediente que coincida el *número* buscado.
2. El sistema redirige al actor al paso 1 del flujo normal.

Poscondiciones:

CU6: Buscar Expediente

Descripción: Permite al actor buscar expedientes por parámetros o una cadena texto.

Actor: Usuario Autenticado(Empleado, Administrativo, Médico)

Precondiciones: El sistema muestra al Actor los parámetros de búsqueda por *departamento y clasificación* y el formulario de búsqueda junto con la opción **Buscar**.

Flujo Normal:

1. El Actor selecciona un parámetro de búsqueda o introduce una cadena de texto y selecciona **Buscar**.
2. El sistema realiza la búsqueda seleccionada por el Actor e invoca al caso de uso **Listar Expedientes** (retorna una lista de expedientes encontrados.)
3. El caso de uso finaliza.

Flujo Alternativo:

Escenario 2a: Búsqueda sin resultados.

1. El sistema muestra un mensaje al actor informando que no hay ningún expediente que coincida con los parámetros o la cadena de texto de búsqueda.
2. El sistema redirige al actor al paso 1 del flujo normal.

Poscondiciones:

CU7: Listar Expedientes.

Descripción: Permite obtener una Lista de Expedientes obtenidos de la búsqueda seleccionada por el del Actor.

Actor: Usuario No Autenticado, Usuario Autenticado(Empleado, Administrativo, Médico)
Precondiciones: El sistema debe encontrar al menos un expediente que coincida con la búsqueda seleccionada por el Actor .
Flujo Normal: <ol style="list-style-type: none">1. El sistema muestra al Actor una lista de los expedientes encontrados junto con las opciones Listar Trámites (permite obtener una lista de trámites asociados al expediente), Consultar Expediente y si tiene privilegios se muestra las opciones Iniciar (RN2) o Concluir Expediente.2. El caso de uso finaliza.
Flujo Alternativo: <p>Escenario 1a: Listar Trámites.</p> <ol style="list-style-type: none">1. El actor selecciona la opción Listar Trámites.2. El sistema envía al actor al caso de uso Listar Trámites. <p>Escenario 1b: Consultar Expediente.</p> <ol style="list-style-type: none">1. El sistema muestra al actor los datos en detalle del expediente seleccionado y si tiene privilegios, muestra las opciones Eliminar(RN1) y Modificar Expediente .2. El caso de uso finaliza. <p>Consultar Expediente 1a: Eliminar Expediente.</p> <ol style="list-style-type: none">3. El actor selecciona eliminar expediente.4. El sistema solicita al actor confirmar la eliminación.5. El actor confirma la eliminación.6. El caso de uso finaliza. <p>Consultar Expediente 1b: Modificar Expediente.</p> <ol style="list-style-type: none">1. El actor selecciona modificar expediente.2. El sistema muestra al actor el formulario del expediente con sus datos.3. El actor introduce los nuevos datos y confirma la modificación.4. El sistema valida y almacena los nuevos datos.5. El sistema muestra al actor un mensaje de operación exitosa.6. El caso de uso continúa con el flujo normal. <p>Modificar Expediente 5a: Datos Inválidos.</p> <ol style="list-style-type: none">1. El sistema muestra un mensaje al actor informando que los datos son inválidos.2. El sistema redirige al actor al paso 3 del Escenario 3b. <p>Escenario 1c: Iniciar Expediente.</p> <ol style="list-style-type: none">1. El actor selecciona la opción Iniciar Expediente.2. El sistema envía al actor al caso de uso Iniciar Expediente. <p>Escenario 1d: Concluir Expediente.</p> <ol style="list-style-type: none">1. El actor selecciona la opción Concluir Expediente.2. El sistema envía al actor al caso de uso Concluir Expediente.
Poscondiciones: <p>Escenario 1a: El sistema continúa con el flujo normal del caso de uso Listar Trámites.</p> <p>Escenario 1b:</p> <p>Consultar Expediente 1a: El expediente se ha eliminado del sistema.</p> <p>Consultar Expediente 1b: Los datos del expediente se modificaron en el sistema.</p> <p>Escenario 1c: El sistema continúa con el flujo normal del caso de uso Iniciar Expediente.</p> <p>Escenario 1d: El sistema continúa con el flujo normal del caso de uso Concluir Expediente.</p>

CU8: Iniciar Expediente.

Descripción: Permite al actor registrar los datos del ingreso de un expediente.

Actor: Empleado

Precondiciones: El actor debe seleccionar la opción **Iniciar Expediente**. El estado del expediente debe ser concluido.

Flujo Normal:

1. El sistema invoca al caso de uso **Nuevo Trámite** (que devuelve un formulario que permite registrar los datos del ingreso del expediente).
2. El actor ingresa los datos de *organismo de origen, fecha de ingreso y tipo de trámite* y selecciona guardar.
3. El sistema valida y almacena los datos.
4. El sistema presenta un mensaje de operación exitosa.
5. El caso de uso finaliza.

Flujo Alternativo:

Escenario 4a: Datos inválidos

1. El sistema muestra al actor un mensaje de advertencia.
2. El sistema redirige al actor al paso 2 del flujo normal.

Poscondiciones:

Flujo Normal:

El ingreso del expediente está registrado y el estado es "iniciado" en el sistema.

CU9: Listar Trámites.

Descripción: Permite al actor obtener una lista de trámites asociados a un expediente.

Actor: Usuario Autenticado (Empleado, Administrativo, Médico)

Precondiciones: El actor debe seleccionar la opción **Listar trámites**. El expediente debe tener al menos un trámite asociado.

Flujo Normal:

1. El sistema muestra al actor una lista de trámites asociados al expediente junto con una opción para **Consultar Trámite**.
2. El caso de uso finaliza.

Flujo Alternativo:

Escenario 1a: Consultar Trámite.

1. El sistema muestra al actor los datos en detalle del trámite seleccionado y si cumple con **RN1** muestra las opciones **Eliminar** (si cumple con **RN3**) y **Modificar Trámite**.
2. El caso de uso finaliza.

Consultar Trámite 1a: Eliminar Trámite.

3. El actor selecciona eliminar Trámite.
4. El sistema solicita al actor confirmar la eliminación.
5. El actor confirma la eliminación.
6. El caso de uso finaliza.

<p>Consultar Trámite 1b: Modificar Trámite.</p> <ol style="list-style-type: none"> 1. El actor selecciona modificar Trámite. 2. El sistema muestra al actor el formulario del trámite con sus datos. 3. El actor introduce los nuevos datos y confirma la modificación. 4. El sistema valida y almacena los nuevos datos. 5. El sistema muestra al actor un mensaje de operación exitosa. 6. El caso de uso continúa con el flujo normal. <p>Modificar Trámite 5a: Datos Inválidos.</p> <ol style="list-style-type: none"> 3. El sistema muestra un mensaje al actor informando que los datos son inválidos. 4. El sistema redirige al actor al paso 3 del Escenario 3b.
<p>Poscondiciones: Escenario 1a: Consultar Trámite 1a: El Trámite se ha eliminado del sistema. Consultar Expediente 1b: Los datos del expediente se modificaron en el sistema. Escenario 1c: El sistema continúa con el flujo normal del caso de uso Iniciar Expediente. Escenario 1d: El sistema continúa con el flujo normal del caso de uso Concluir Expediente.</p>

<p>CU10: Nuevo Trámite.</p>
<p>Descripción: Permite obtener un formulario para que el actor registre los datos de ingreso de un expediente.</p>
<p>Precondiciones: El estado del expediente debe ser “concluido”.</p>
<p>Flujo Normal:</p> <ol style="list-style-type: none"> 3. El sistema muestra un formulario de Nuevo de Trámite con los campos (<i>organismo de origen, fecha de ingreso y tipo de trámite</i>). 4. El sistema retorna el formulario generado. 5. El caso de uso finaliza.
<p>Poscondiciones: El sistema continúa con el flujo normal del caso de uso que lo invocó.</p>

<p>CU11: Concluir Expediente.</p>
<p>Descripción: Permite al actor registrar los datos de entrega de un expediente en particular y cambiar su estado ha “concluido”.</p>
<p>Actor: Empleado</p>
<p>Precondiciones: El actor debe seleccionar la opción Concluir Expediente de un expediente en particular. El mismo debe estar en estado “iniciado”.</p>
<p>Flujo Normal:</p> <ol style="list-style-type: none"> 1. El sistema muestra un formulario para registrar los datos de entrega del expediente. 2. El actor ingresa los datos del <i>organismo de destino y la fecha de entrega</i> y los guarda. 3. El sistema valida y almacena los datos.

4.	El sistema muestra al actor un mensaje de operación exitosa.
5.	El caso de uso finaliza.
Flujo Alternativo:	
Escenario 2a: Datos inválidos.	
1.	El sistema muestra al actor un mensaje informando que los datos son inválidos.
2.	El sistema retorna al actor al paso 2 del flujo normal.
Poscondiciones:	
Flujo Normal:	
El trámite del expediente está concluido el sistema.	

3.4.3 Diagrama de Casos de Uso del Paquete Gestión de Documento

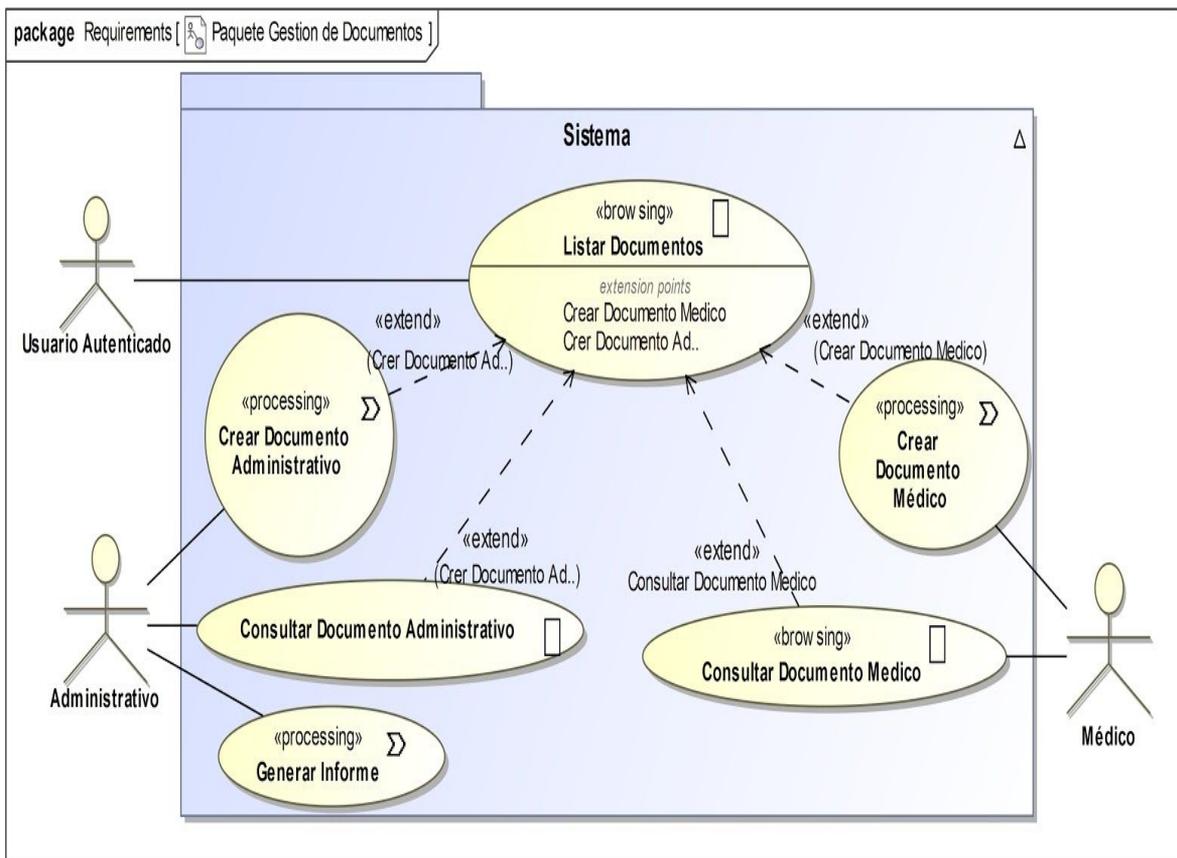


Figura [21]: Diagrama de Casos de Uso del Paquete Gestión Documento.

CU12: Listar Documentos.
Descripción: Permite al actor obtener una lista de documentos asociados a un trámite en particular.
Actor: Usuario Autenticado (Empleado, Administrativo, Médico)
Precondiciones: El actor debe seleccionar la opción Listar Documentos .
Flujo Normal:

<p>1. El sistema muestra al actor una lista de documentos asociados a un trámite con los campos <i>área, tipo de documento, número y fecha de creación</i>, y si el actor tiene privilegios y cumple con RN6 muestra las opciones Consultar Documento Administrativo o Médico y si cumple con RN5 Crear Documento Administrativo o Médico.</p> <p>2. El caso de uso finaliza.</p>
<p>Flujo Alternativo:</p> <p>Escenario 1a: Documento Médico.</p> <p>1. El actor selecciona la opción Consultar Documento Médico.</p> <p>2. El sistema envía al actor al caso de uso Consultar Documento Médico.</p> <p>Escenario 1b: Documento Administrativo.</p> <p>3. El actor selecciona la opción Consultar Documento Administrativo.</p> <p>4. El sistema envía al actor al caso de uso Consultar Documento Administrativo.</p> <p>Escenario 1c: Crear Documento Médico.</p> <p>5. El actor selecciona la opción Crear Documento Médico.</p> <p>6. El sistema envía al actor al caso de uso Crear Documento Médico.</p> <p>Escenario 1d: Crear Documento Administrativo.</p> <p>7. El actor selecciona la opción Crear Documento Administrativo.</p> <p>8. El sistema envía al actor al caso de uso Crear Documento Administrativo.</p>
<p>Poscondiciones:</p> <p>Escenario 1a / 1b: El sistema continúa en el flujo normal del casos de uso Consultar Documento Administrativo / Consultar Documento Médico.</p> <p>Escenario 1c / 1d: El sistema continúa en el flujo normal del casos de uso Crear Documento Administrativo / Crear Documento Médico.</p>

CU13 : Consultar Documento Administrativo.
Descripción: Permite al actor consultar un documento administrativo.
Actor: Administrativo
Precondiciones: El actor debe seleccionar la opción Consultar Documento . La clase de documento debe ser administrativo.
<p>Flujo Normal:</p> <p>1. El sistema identifica el tipo y formato del documento.</p> <p>2. El sistema muestra el documento junto con las opciones de Modificar y Eliminar.</p> <p>3. El caso de uso finaliza.</p>
<p>Flujo Alternativo:</p> <p>Escenario 2a: Eliminar</p> <p>1. El actor selecciona la opción Eliminar.</p> <p>2. El sistema solicita al actor que confirme la operación.</p> <p>3. El actor confirma la operación.</p> <p>4. El caso de uso finaliza.</p> <p>Escenario 2b: Modificar.</p> <p>1. El actor selecciona la opción Modificar.</p>

<ol style="list-style-type: none">2. El sistema muestra al actor el formulario correspondiente al documento con los datos a modificar.3. El actor introduce los nuevos datos y confirma la modificación.4. El sistema valida los datos y los almacena.5. El sistema muestra al actor un mensaje de operación exitosa.6. El caso de uso finaliza. <p>Modificar 5a: Datos Inválidos.</p> <ol style="list-style-type: none">1. El sistema muestra al actor un mensaje informando que los datos son inválidos.2. El sistema redirige al actor al paso 3 del Escenario 2b.
<p>Poscondiciones: Escenario 2a: El documento ha sido eliminado del sistema. Escenario 2b: La información del documento ha sido modificada en el sistema.</p>

CU14 : Consultar Documento Médico.
<p>Descripción: Permite al actor consultar un documento administrativo.</p>
<p>Actor: Médico</p>
<p>Precondiciones: El actor debe seleccionar la opción Consultar Documento Médico. La clase de documento debe ser Médico.</p>
<p>Flujo Normal:</p> <ol style="list-style-type: none">4. El sistema identifica el tipo y formato del documento.5. El sistema muestra el documento junto con las opciones de Modificar y Eliminar.6. El caso de uso finaliza.
<p>Flujo Alternativo: Escenario 2a: Eliminar</p> <ol style="list-style-type: none">5. El actor selecciona la opción Eliminar.6. El sistema solicita al actor que confirme la operación.7. El actor confirma la operación.8. El caso de uso finaliza. <p>Escenario 2b: Modificar.</p> <ol style="list-style-type: none">7. El actor selecciona la opción Modificar.8. El sistema muestra al actor el formulario correspondiente al documento con los datos a modificar.9. El actor introduce los nuevos datos y confirma la modificación.10. El sistema valida los datos y los almacena.11. El sistema muestra al actor un mensaje de operación exitosa.12. El caso de uso finaliza. <p>Modificar 5a: Datos Inválidos.</p> <ol style="list-style-type: none">3. El sistema muestra al actor un mensaje informando que los datos son inválidos.4. El sistema redirige al actor al paso 3 del Escenario 2b.
<p>Poscondiciones: Flujo Normal: La información del documento ha sido almacenada y asociada a un expediente.</p>

CU15: Crear Documento Administrativo.
Descripción: Permite al actor crear documentos administrativos.
Actor: Administrativo
Precondiciones: El actor debe seleccionar la opción Crear Documento Administrativo . El estado del expediente debe ser "iniciado".
Flujo Normal: <ol style="list-style-type: none">7. El sistema muestra al Actor una lista de documentos que puede seleccionar (<i>Proveído, Citación, Turno de Junta Médica, Constancia</i>)8. El actor selecciona una opción.9. El sistema muestra al Actor un formulario con los campos de datos requeridos para crear el tipo de documento.10. El actor introduce los datos en el formulario.11. El sistema valida los datos y los almacena.12. El sistema muestra un mensaje de operación exitosa.13. El sistema muestra al actor el documento en el formato correspondiente.14. El caso de uso finaliza.
Flujo Alternativo: Escenario 6a: Datos inválidos. <ol style="list-style-type: none">1. El Sistema muestra al actor un mensaje de advertencia.2. El sistema redirige al actor al paso 4 del Flujo Normal.
Poscondiciones: Flujo Normal: La información del documento ha sido almacenada y asociada a un expediente.

CU16: Crear Documento Médico.
Descripción: Permite al actor crear documentos administrativos.
Actor: Administrativo
Precondiciones: El actor debe seleccionar la opción Crear Documento Médico . El estado del expediente debe ser "iniciado".
Flujo Normal: <ol style="list-style-type: none">1. El sistema muestra al Actor una lista de documentos que puede seleccionar (<i>Acta Médica, Informe Médico</i>)2. El actor selecciona una opción.3. El sistema muestra al Actor un formulario con los campos de datos requeridos para crear el tipo de documento.4. El actor introduce los datos en el formulario.5. El sistema valida los datos y los almacena.6. El sistema muestra un mensaje de operación exitosa.7. El sistema muestra al actor el documento en el formato correspondiente.8. El caso de uso finaliza.
Flujo Alternativo:

Escenario 6a: Datos inválidos. 3. El Sistema muestra al actor un mensaje de advertencia. 4. El sistema redirige al actor al paso 4 del Flujo Normal .
Poscondiciones: Flujo Normal: La información del documento ha sido almacenada y asociada a un expediente.

CU17: Generar Informe.
Descripción: Permite al actor generar informes de trámites y licencias especiales de un determinado periodo.
Actor: Supervisor.
Precondiciones: El actor debe seleccionar la opción Generar Informe .
Flujo Normal: 1. El sistema muestra al actor un formulario con los parámetros a consultar. 2. El actor introduce los parámetros de consulta que son: <i>Tipo de Informe</i> (Informar de trámites por agente/ Informe de Licencias Especiales por Agente) y el periodo: <i>Fecha desde y Fecha Hasta</i> y luego confirma la consulta. 3. El sistema muestra al actor el informe solicitado. 4. El caso de uso finaliza.
Flujo Alternativo: Escenario 3a: Consulta sin resultados. 1. El sistema muestra un mensaje al actor informando que no hay ningún resultados para la consulta. 2. El sistema redirige al actor al paso 2 del flujo normal.
Poscondiciones:

CAPÍTULO IV: DISEÑO EMPLEANDO LA METODOLOGÍA U.W.E.

4.1 INTRODUCCIÓN

En base al Modelo de Requisitos del Prototipo obtenidos en el capítulo anterior se continúa con el diseño del Prototipo SIARMe mostrando los diferentes aspectos que contemplan los modelos de la Metodología de desarrollo U.W.E. basado en U.M.L.

Estos aspectos comprenden los siguientes modelos:

- Modelo de Contenido en la sección 4.1;
- Modelo de Navegacional en la sección 4.2;
- Modelo de Presentación en la sección 4.3;
- Modelo de Proceso en la sección 4.4.

4.2 MODELO DE CONTENIDO

En esta etapa se representa el dominio del problema con un diagrama de clases de UML sin necesidad de aplicar semánticas adicionales, considerando los requisitos plasmados en los casos de uso que permiten determinar las clases, métodos y atributos, con el objetivo de representar los requerimientos de datos del prototipo SIARMe, modelando los datos persistentes.

Este modelo proporcionando una especificación visual de la información en el dominio relevante para el prototipo. Particularmente la información de los usuarios es modelada por la clase "Usuario" donde se almacenan las propiedades que describen a los diferentes usuarios según un "Rol" asignado que se modela como una enumeración (**Empleado**, **Administrativo**, **Médico**). Un usuario pertenece a un departamento de la DPSRM por lo estos se modelan con la clase "**DepartamentoRm**". La información de "**Expediente**" y la de "**Documento**" se modelan con sus respectivas clases. Un departamento (Autoseguro o Despacho) se relaciona con muchos expedientes esta relación modela que en ambos departamentos se realizan trámites diferentes. Además, se observa que un documento se relaciona con un rol ya sea **Administrativo** o **Médico** según RN6.

Existen otras clases que modelan las tablas necesarias para el funcionamiento del Prototipo y pertenecen al sistema actual (SIA) con las cuales ambos sistemas interactúan. La clases que se describen a continuación representan las tablas que comparten ambos sistemas: La clase "**Agente**" modela la información de los agentes públicos y la clase "**Cargo**" sus lugares de trabajos. La Clase "**Localidad**" y "**Departamento**" modela la información de las localidades y los departamentos de la provincia. La clase "**Licencia**", "**Artículo**" y "**Enfermedad**" modelan las información de las licencias médicas de los agentes.

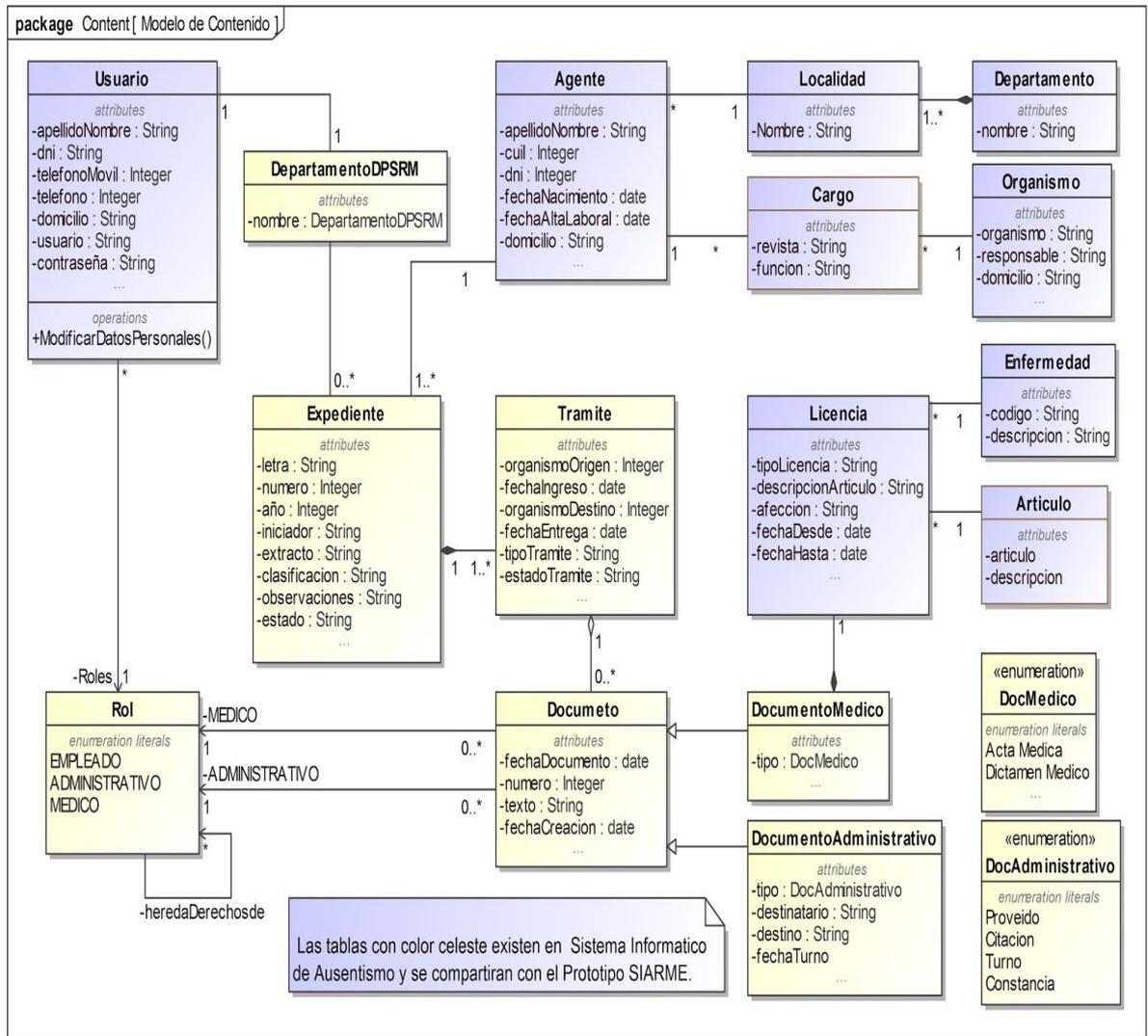


Figura [22]: Modelo de Contenido de SIARMe.

4.3 MODELO NAVEGACIONAL

En este modelo se indicará cómo las páginas web del Prototipo SIARMe están relacionadas internamente en base al análisis de requisitos y el modelo de contenido. Está basado en el diagrama de clases de U.M.L. usando elementos específicos de modelado para los conceptos orientados a la Web y algunos estereotipos de los elementos que podemos seleccionar para incurrir en la navegación como tours guiados, menús, links externos y páginas índices. Especificando las funcionalidades de navegación a través de los siguientes estereotipos.

Nombres de estereotipos y sus iconos

Elemento	Estereotipos UWE	Símbolo
Clase de navegación	<< navigationClass >>	
Índice	<< index >>	
Tours guiados	<< guidedTour >>	
Menú	<< menu >>	

Consulta	<< query>>	?
Clase de proceso	<<processClass>>	Σ

En la figura[23] se observa el Modelo de Navegación del Prototipo SIARMe en primer lugar, especifica qué objetos dentro del sistema pueden ser visitados por el usuario y, en segundo lugar, define cómo se alcanzan estos objetos en la aplicación. Particularmente, la clase navegación **Portada** □ se identifica como punto de entrada de la aplicación web con la etiqueta {**isHome**}, desde la que se accede a la clase de navegación **Inicio** □ que modela la página de inicio de la portada del prototipo. Un usuario, que no se ha autenticado frente al sistema, puede acceder a las distintas alternativas o clases de navegación mediante la clase **Menu UsuarioNoAutenticado** ≡. Los enlaces de navegación "navigationLink" y "processLink" muestran vínculos directos entre las clases de navegación y representan posibles pasos a seguir por el usuario por lo que brinda acceso a las clases de navegación **Buscar Expte. por Número** ? que se etiqueta con {**isLandmark**} para indicar que es una opción que siempre está disponible, **Inicio** □ e **Iniciar Sesión** Σ. Esta última clase modela el proceso de autenticación de usuario frente al sistema.

Una vez que el usuario se ha autenticado accede a la clase navegación **Intranet** □ que se identifica como punto de entrada a la zona restringida del prototipo con la etiqueta {**isHome**}. Un usuario autenticado puede acceder a las distintas alternativas de navegación comunes mediante la clase **Menu Usuario Autenticado** ≡. Las clases a las que brinda acceso pueden ser **Buscar Expediente** ? que se etiqueta con {**isLandmark**}, **Listar Expediente** ≡, **Modificar Datos Personales** Σ, **Cerrar Sesión** Σ. Además, según el rol del usuario se presentan las clases **Menú Empleado** ≡, **Menú Administrativo** ≡ y **Menú Médico** ≡. Cada una de estas clases ofrecen acceso a las distintas funcionalidades requeridas por los usuarios, por ejemplo: solo el usuario **Empleado** puede acceder a la clase de proceso **Crear Expediente** Σ y a las clases de navegación **Menú Expediente** ≡ (brinda las opciones de iniciar, concluir, modificar o eliminar un expediente en particular) y **Menú Trámite** ≡ (brinda las opciones de modificar o eliminar un trámite).

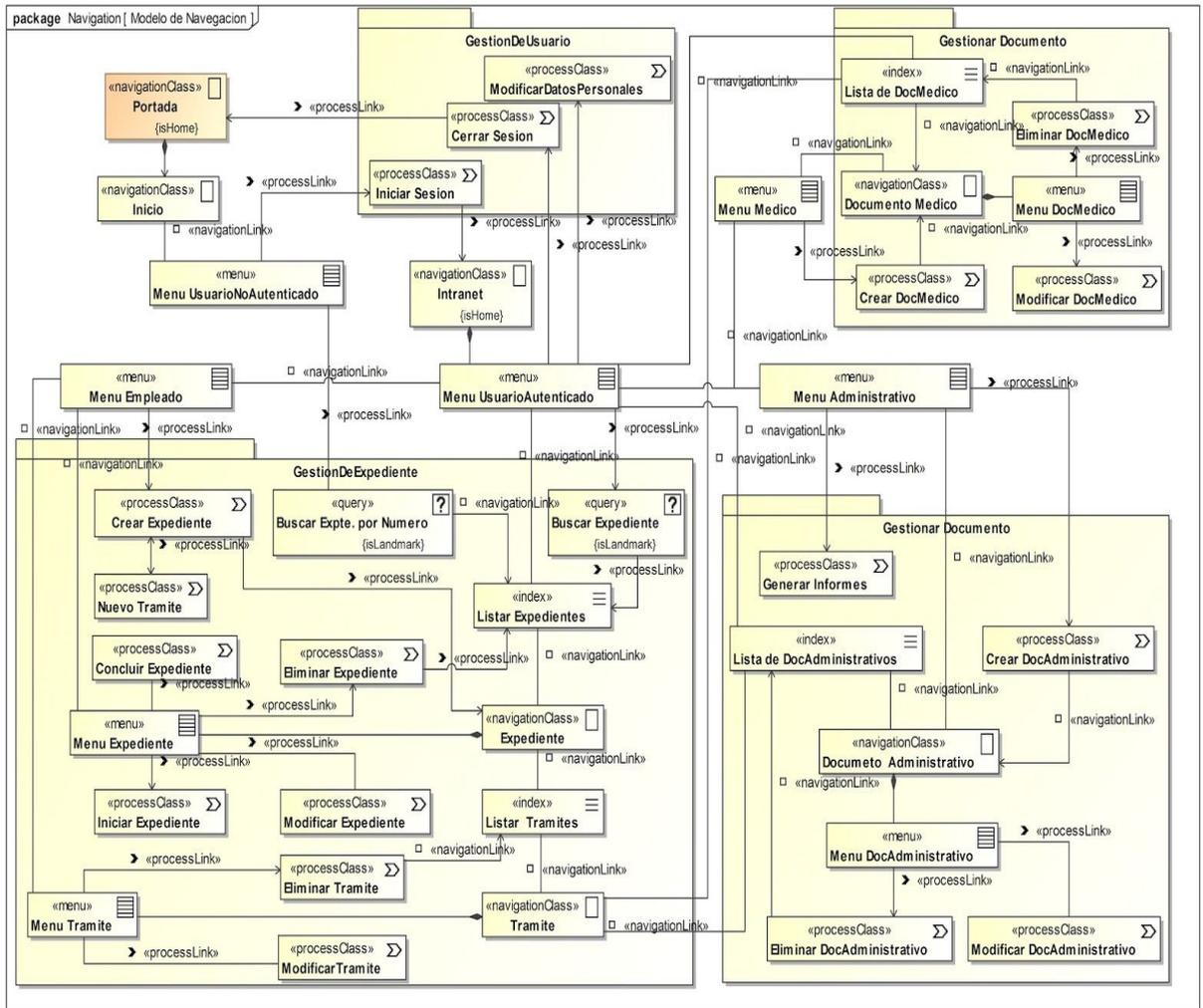


Figura [23]: Modelo de Navegación de SIARME

4.4 MODELO DE PRESENTACIÓN

El modelo de presentación ofrece una visión abstracta de la interfaz de usuario de una aplicación Web.

Se emplea un diagrama de presentación para definir cómo está estructurada una interfaz que interactuará con el usuario y los elementos que permiten la ejecución de acciones; estos elementos están definidos como estereotipos de UML como se detallan a continuación:

Nombres de estereotipos y sus iconos

Elemento	Estereotipos UWE	Símbolo
grupo de presentación	«presentationGroup»	
texto	«text»	
ancla	«anchor»	
botón	«button»	
formulario	«inputForm»	
página de presentación	«presentationPage»	

entrada de texto	«textInput»	
Subir archivo	«presentationAlternatives»	
imagen	«image»	
selección	«selection»	
interacción de grupo de presentacion	«iteratedPresentationGroup»	

4.4.1 Diseño Base del Prototipo SIARMe

A partir del Diseño Base se van a modelar las páginas con la información a la que el usuario puede acceder según su rol. Además, este diseño se utilizará en la herencia de plantillas durante la codificación de las páginas del Prototipo, empleando el Framework Symfony.

En la figura[24] se presenta la estructura principal sobre la que se construirán las distintas páginas.

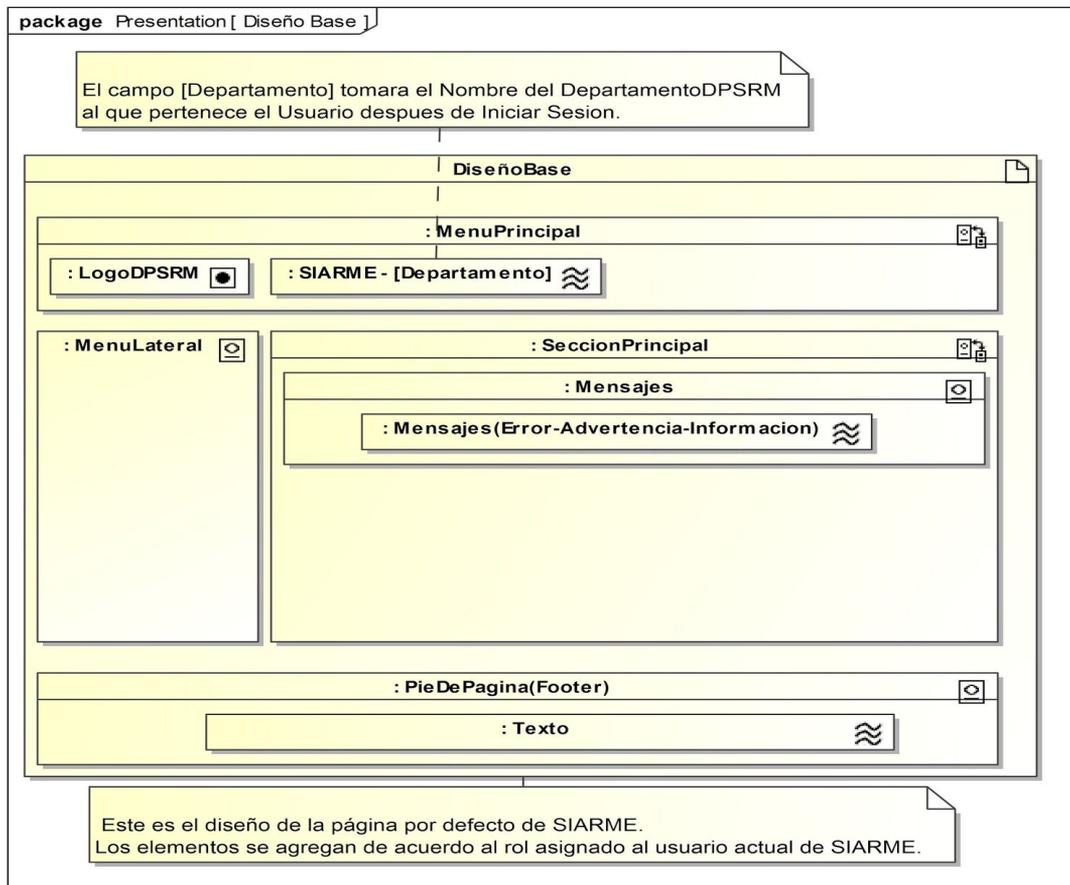


Figura [24]: Diagrama de Presentación - Diseño Base de SIARMe.

4.4.2 Diseño de la Zona Pública o Portada

A partir del Diseño Base se modela, a continuación, la interfaz de la Portada del Prototipo SIARMe como se observa en la figura[25].

La **PaginaDelInicio** se presenta cuando un usuario accede a la **Portada** de SIARMe ofreciéndole información general. Además, se observa el **MenuPrincipal**, donde el usuario puede acceder a las distintas opciones descritas en el modelo de navegación como por ejemplo “**Menú UsuarioNoAutenticado**” de la figura[23] con las opciones correspondientes al caso: **Inicio**, **BuscarExpedientePorNumero** e **IniciarSesion**.

En este diseño se tiene cuenta las herencia de plantillas por lo que además de las opciones descritas en el **MenuPrincipal**, también se presentan la imagen **LogoDPSM** y el texto **SIARMe** que se especifican en el **DiseñoBase**, figura[24]. En este sentido, se observa de qué manera se simplifica el mantenimiento y la reutilización de código en la herencia de plantillas, por ejemplo, cualquier cambio que se realice en el **DiseñoBase** afectará a todas las páginas.

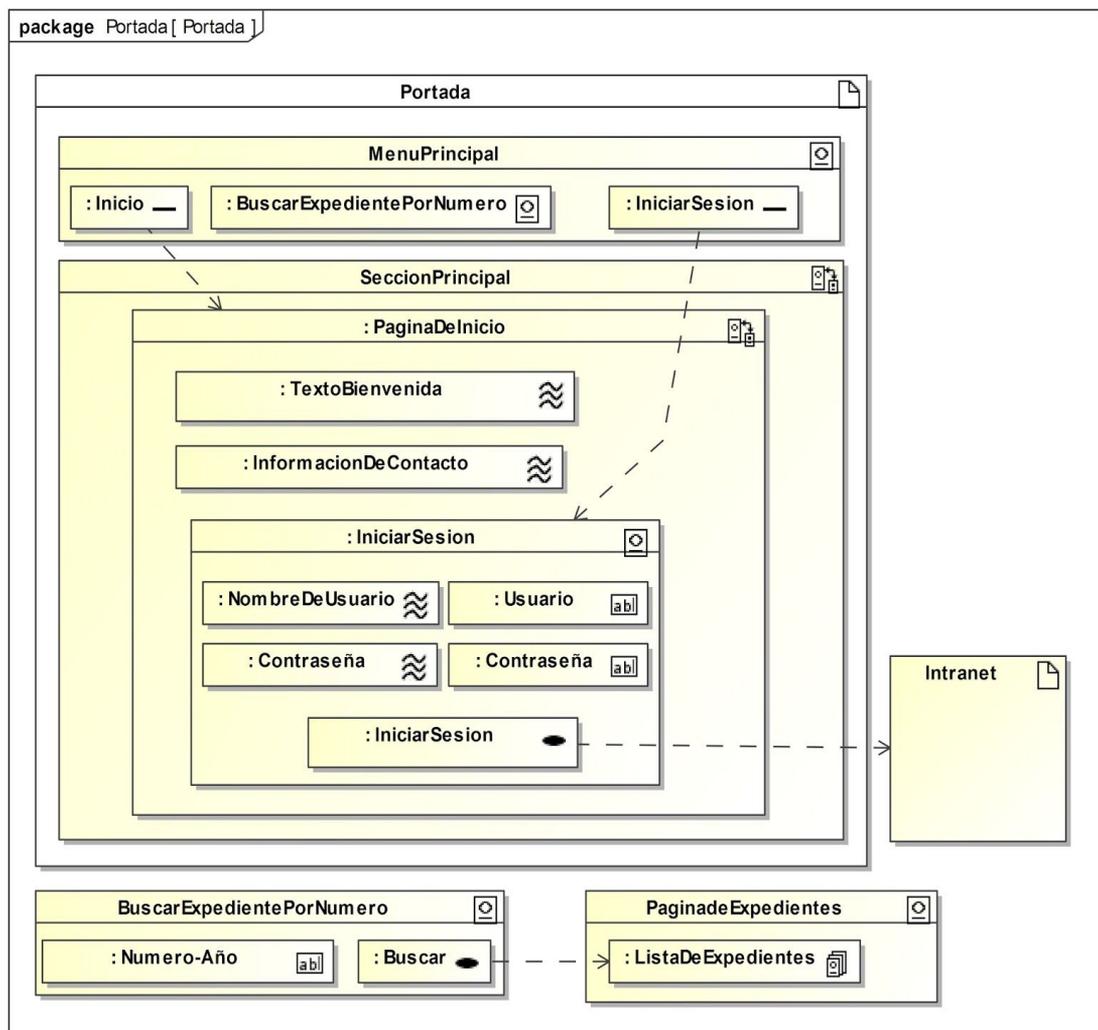


Figura [25]: Modelo de Presentación - Portada

4.4.3 Diseño de la Zona Restringida o Intranet

4.4.3.1 Diagrama de Presentación Paquete Gestión de Expediente

En la figura[26] se observan las opciones de acceso o restricciones según el rol del usuario. Por ejemplo una **ListaDeExpediente** es presentada cada vez que un usuario accede a **Intranet**, luego de su autenticación, pero únicamente al usuario con rol **Empleado** se le muestra el **menuEmpleado** que le permitirá **Iniciar** o **Concluir** un expediente (el diseño de las páginas para estas opciones se observan en la figura[27]). Además, tendrá acceso a otras opciones como, por ejemplo **Letra-Numero-Año**, que además de mostrar estos datos, permite al usuario ver en detalle los datos del expediente en **DetalleDeExpediente**, como se observa en la figura [27] y **ListarTramites** que permite a un usuario autenticado obtener una lista de los trámites asociados a un expediente específico. De la misma forma **ListarDocumentos** muestra una lista de documentos asociados a un trámite.

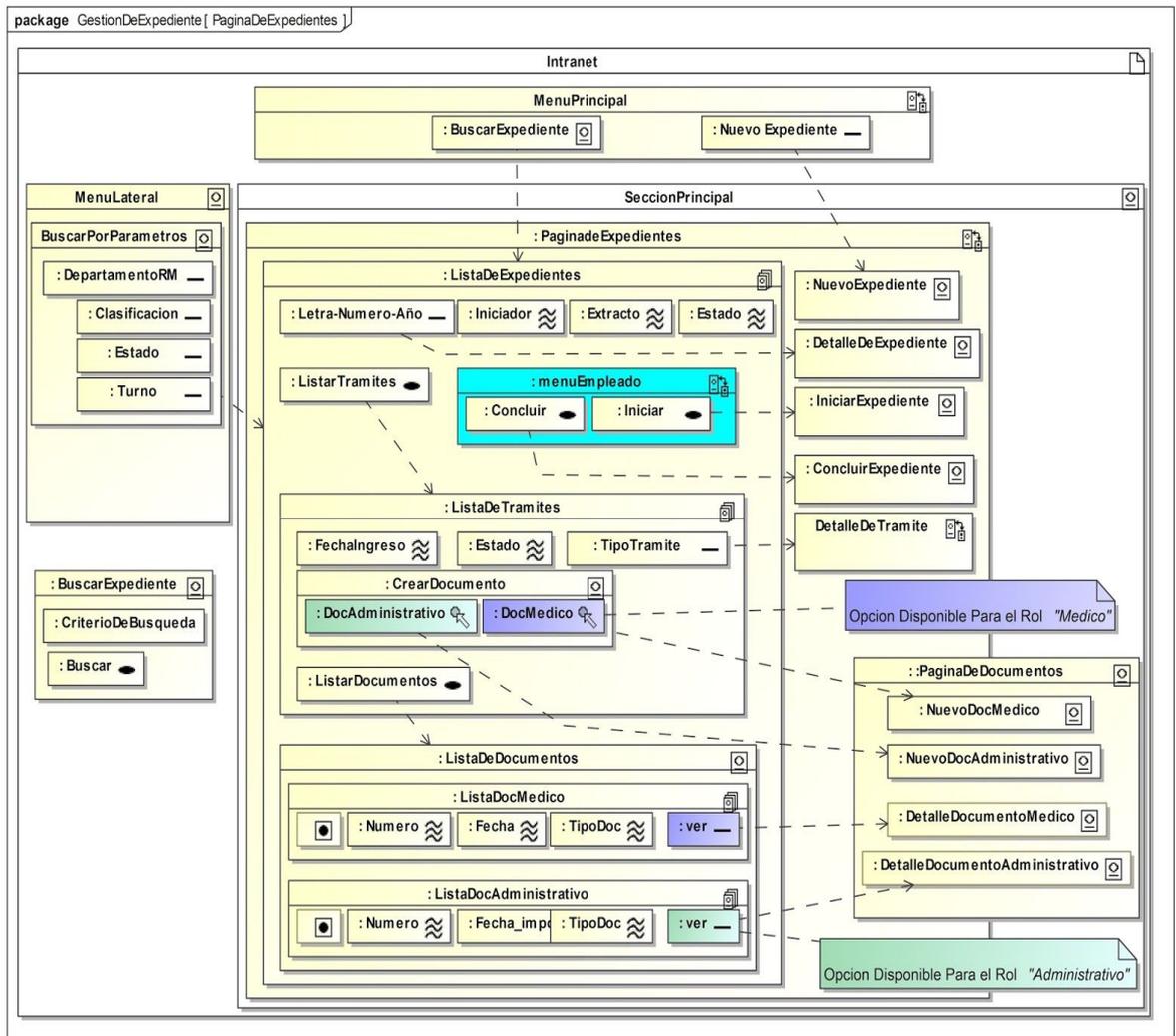


Figura [26]: Diagrama de Presentación- Paquete Gestión de Expediente.

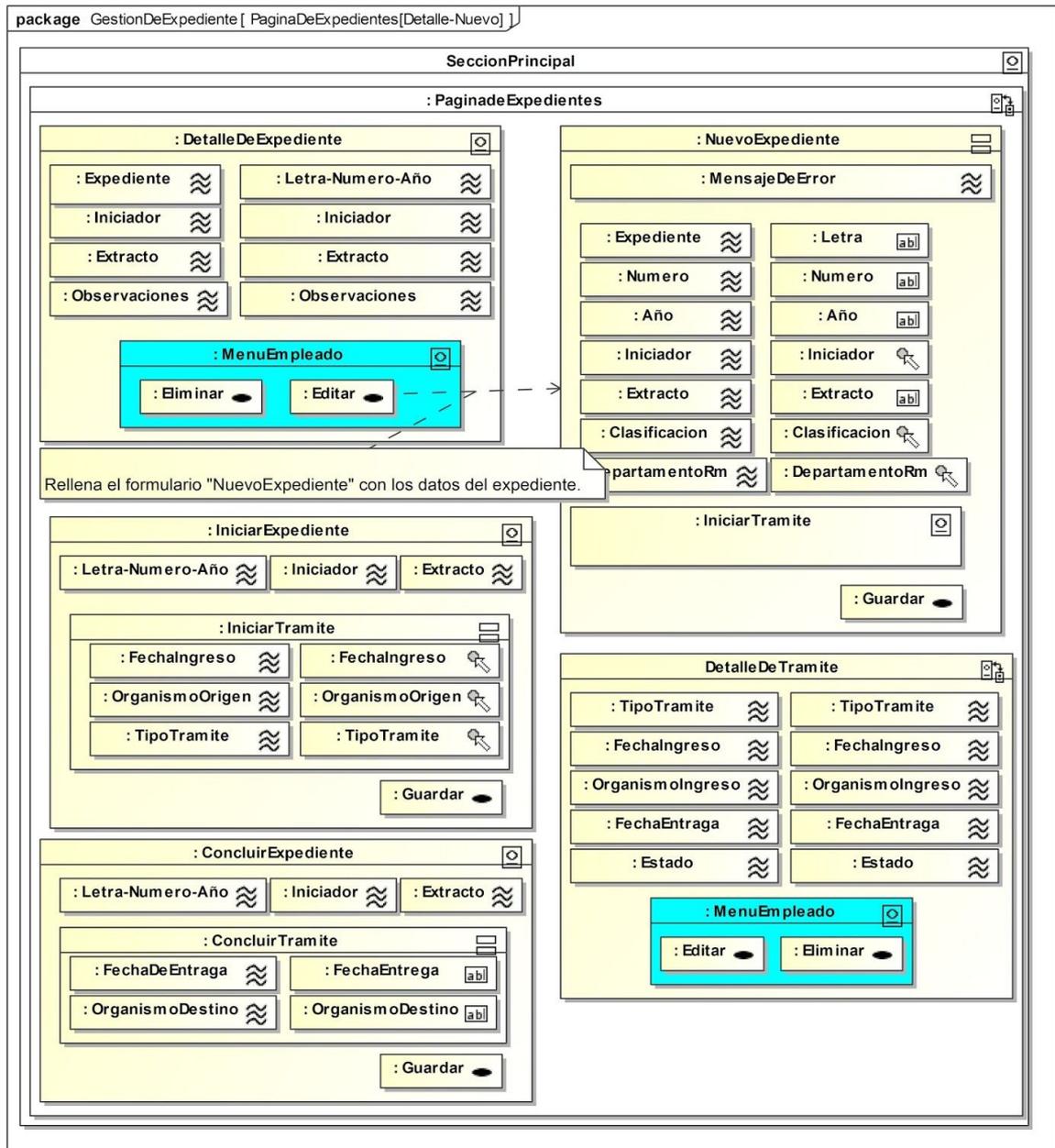


Figura [27]: Diagrama de Presentación- Paquete Gestión de Expediente.

4.4.3.2 Diagrama de Presentación del Paquete Gestión de Documento

En este diseño se presentan los diagramas de presentación de las páginas requeridas en la gestión de documentos disponibles para los roles **Administrativo** y **Médico**.

En el diagrama de la figura[32] se presenta un formato genérico mostrando la información relevante para los documentos administrativos o médicos. Por lo que, este formato, deberá adecuarse al tipo de documento en el desarrollo del prototipo.

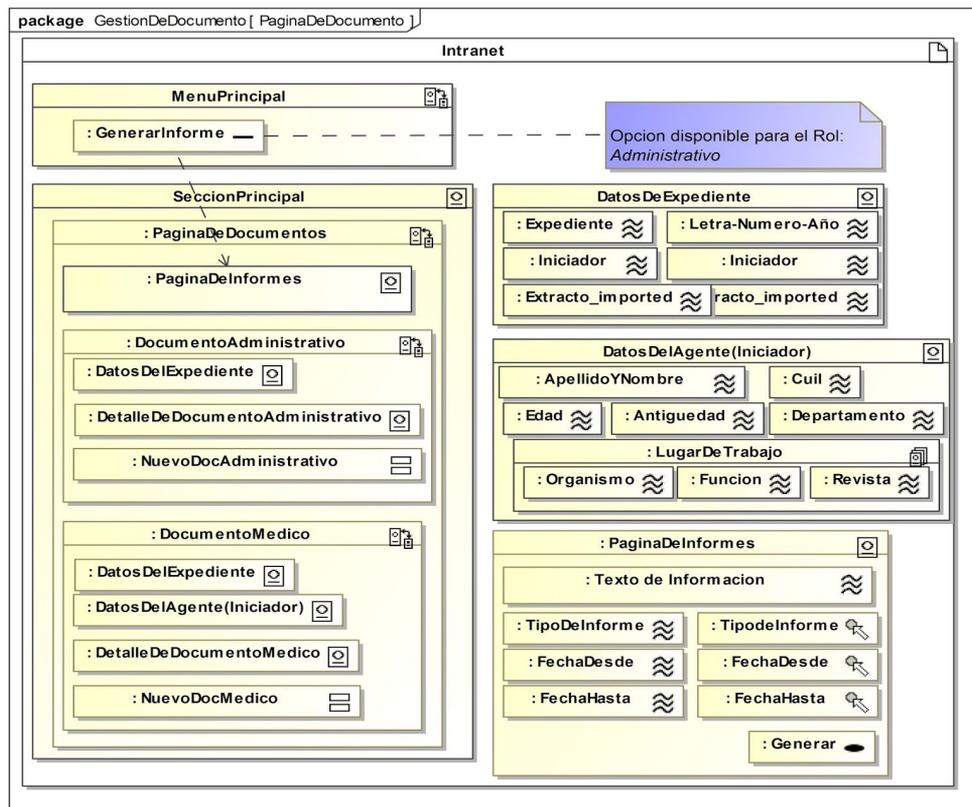


Figura [28]: Diagrama de Presentación- Paquete Gestión de Documento.

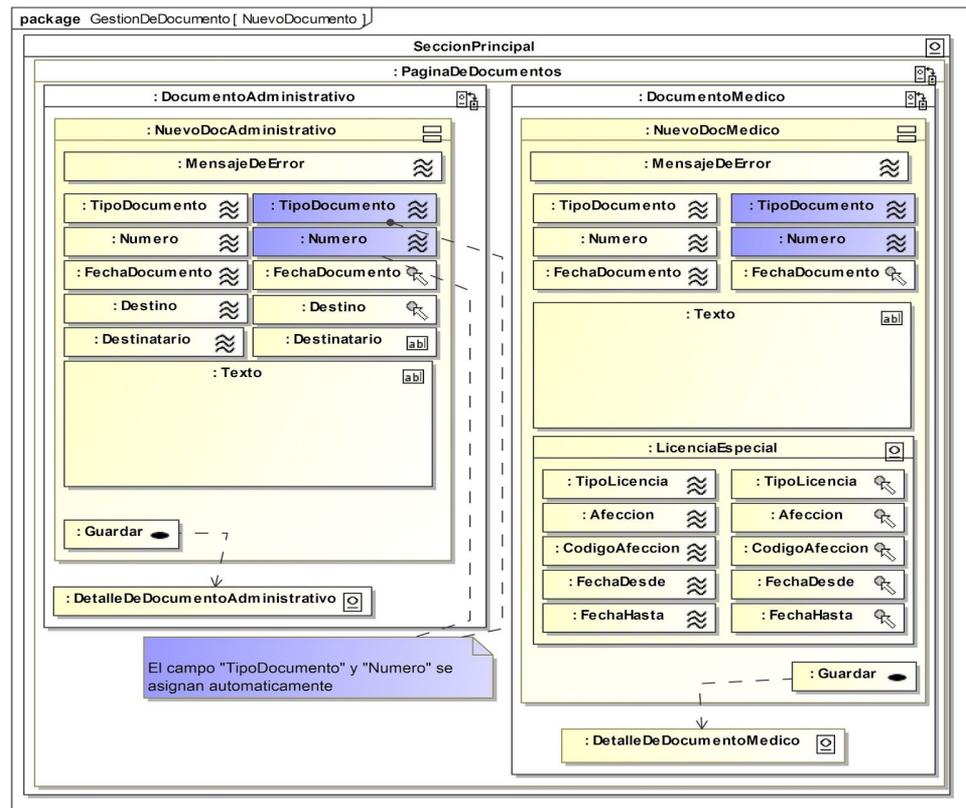
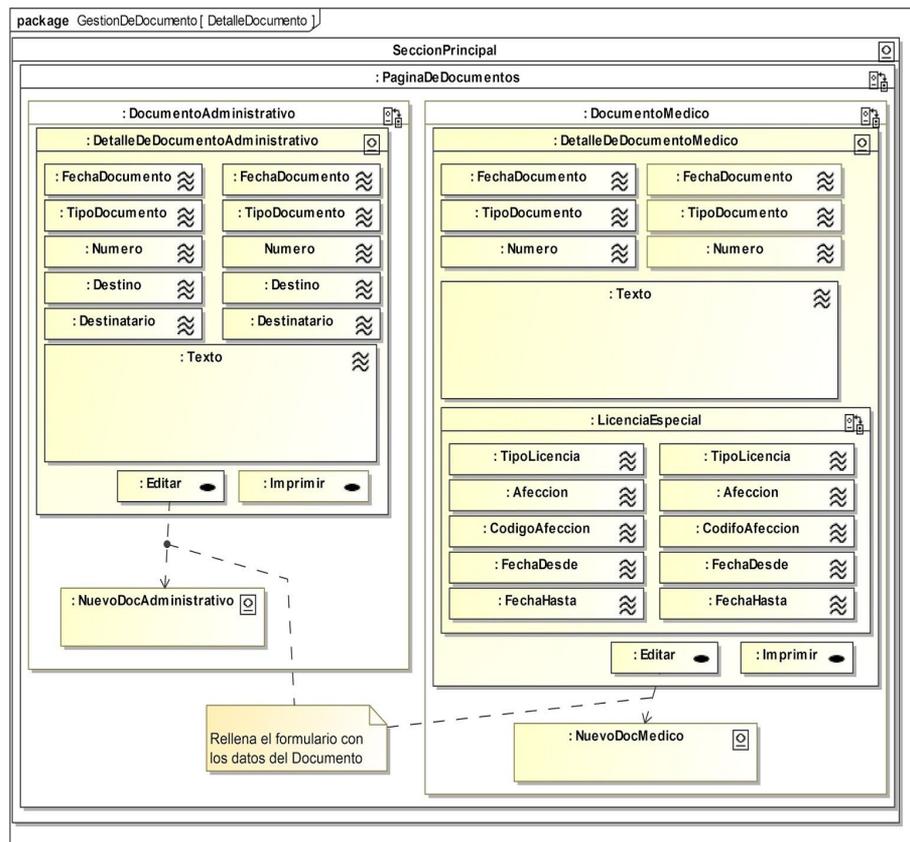


Figura [29]: Diagrama de Presentación del Paquete Gestión de Documento.



Figura[30]: Diagrama de Presentación del Paquete Gestión de Documento

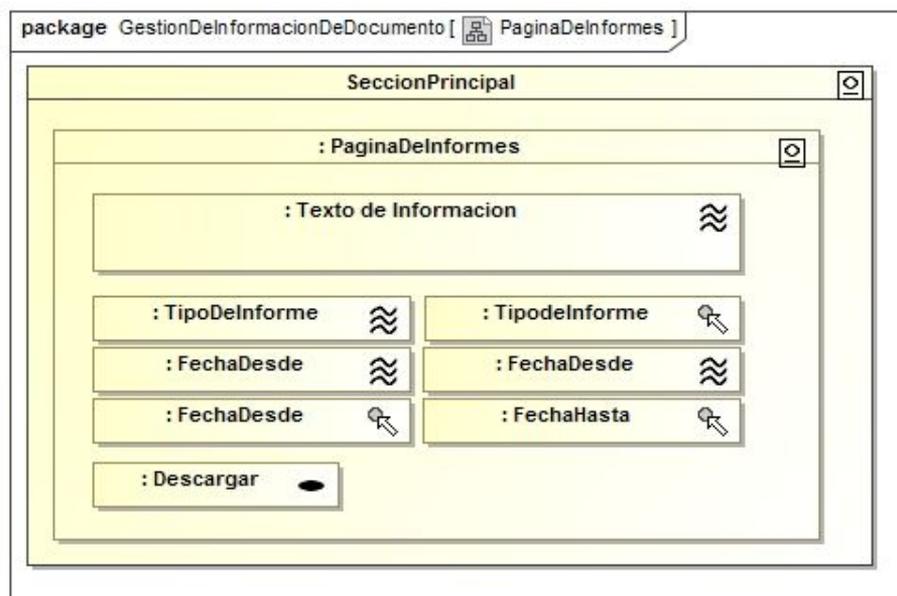


Figura [31]: Diagrama de Presentación del Paquete Gestión de Documento.

4.4.3.3 Diagrama de Presentación del Paquete Gestión de Usuario

En este diseño se observa un diagrama de presentación con las páginas requeridas en el paquete gestión de usuario. Las opciones correspondientes a éste estarán disponibles

a los usuarios autenticados, quienes podrán consultar sus datos personales y/o modificarlos; cambiar su contraseña y cerrar sesión.

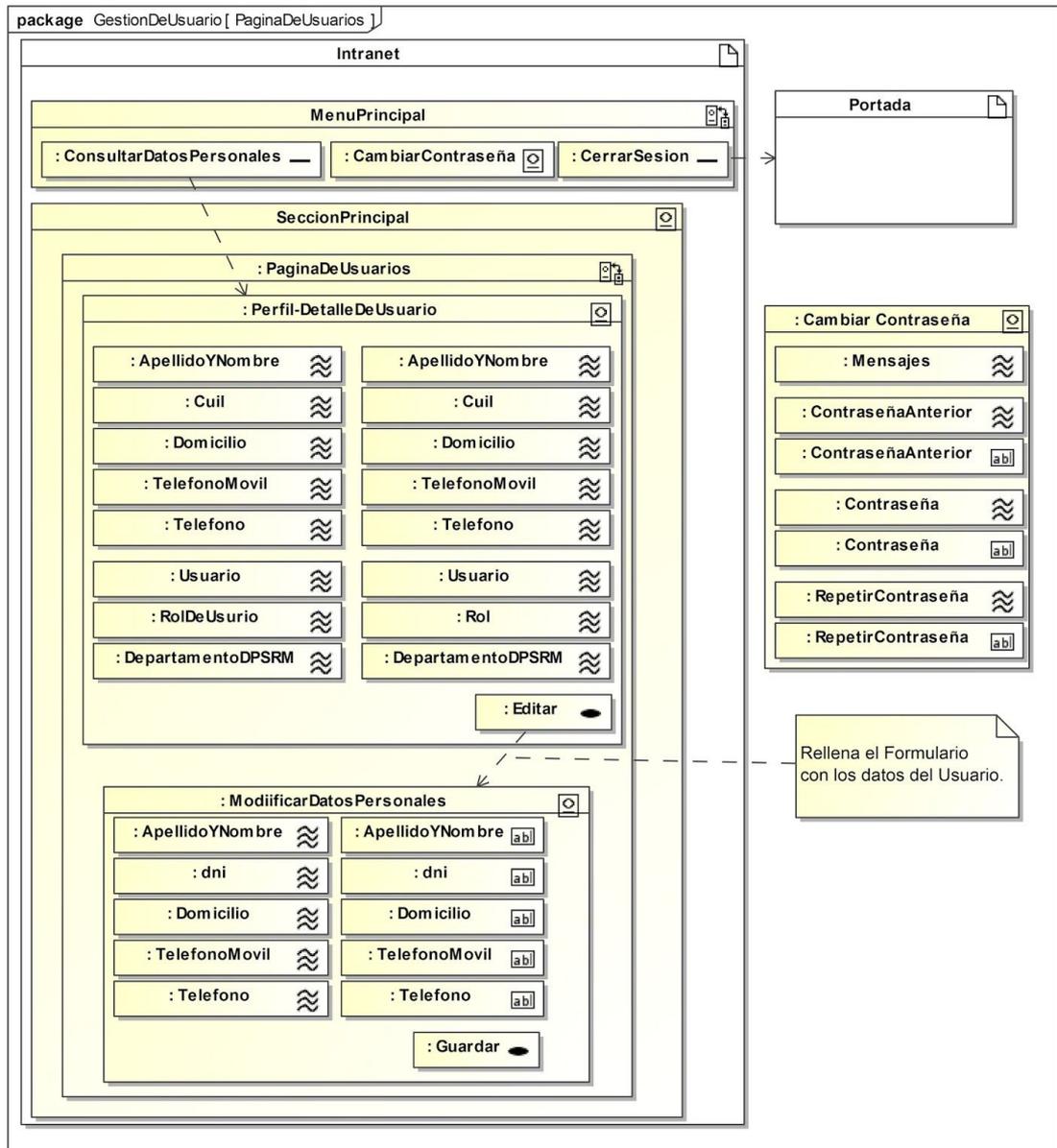


Figura [32]: Diagrama de Presentación del Paquete Gestión de Usuar.

4.5 MODELO DE PROCESOS

La estructura de navegación puede ser extendida mediante clases de procesos que representan la entrada y la salida de procesos de negocio. El modelo del proceso representa el aspecto que tienen las acciones de las clases de proceso etiquetadas con el estereotipo <<processClass>>. Este modelo comprende dos tipos de modelos:

- Modelo de estructura del proceso, que describe las relaciones entre las diferentes clases de proceso y es representado por un diagrama de clases donde se describen las relaciones entre las diferentes clases de proceso.

- Modelo de flujo del proceso, describe el flujo de control y de datos en cada clase de proceso, incluidos en el modelo de navegación, mediante diagramas de actividad UML. Los diagramas de actividades incluye actividades, actores responsables de estas actividades que pueden etiquetadas con <<userAction>>  cuando es un usuario o cuando es el sistema <<systemAction>>  y elementos de flujo de control. Ellos pueden ser enriquecidos con flujos de objetos que muestran objetos relevantes para la entrada o salida de esas actividades.

En cuanto al modelo de flujo de proceso se empleará únicamente para mostrar con más detalle aquellos procesos que se consideren relevantes.

4.5.1 Diagramas de Procesos del Paquete Gestión de Usuario

En la figura[33] se observa cómo se relacionan las clases de procesos que se describen en el Modelo de Navegación como por ejemplo: **IniciarSesion**  y **ModificarDatosDeUsuario** . Adicionalmente se agregan otras clases para representar los datos de entrada, como por ejemplo **IntroducirDatosDeUsuario**  de la clase abstracta **ProcesarUsuario** .

En la figura[34] se describe el comportamiento de la clase de proceso **ModificarDatosDeUsuario** , mostrando lo que sucede en detalle, cuando el usuario ejecuta esta acción (al presionar el botón **Editar**  en la Página **Perfil-DetalleDeUsuario**  Figura[32]).

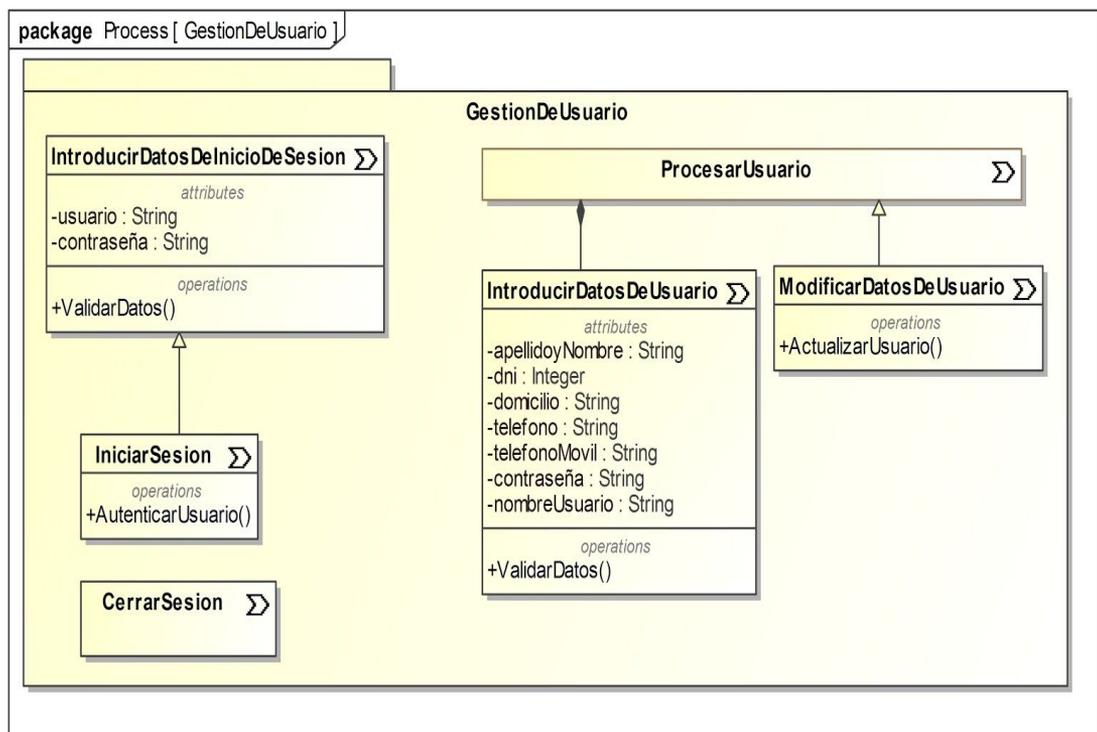


Figura [33]: Diagrama de Estructura de Procesos del Paquete Gestión de Usuarios.

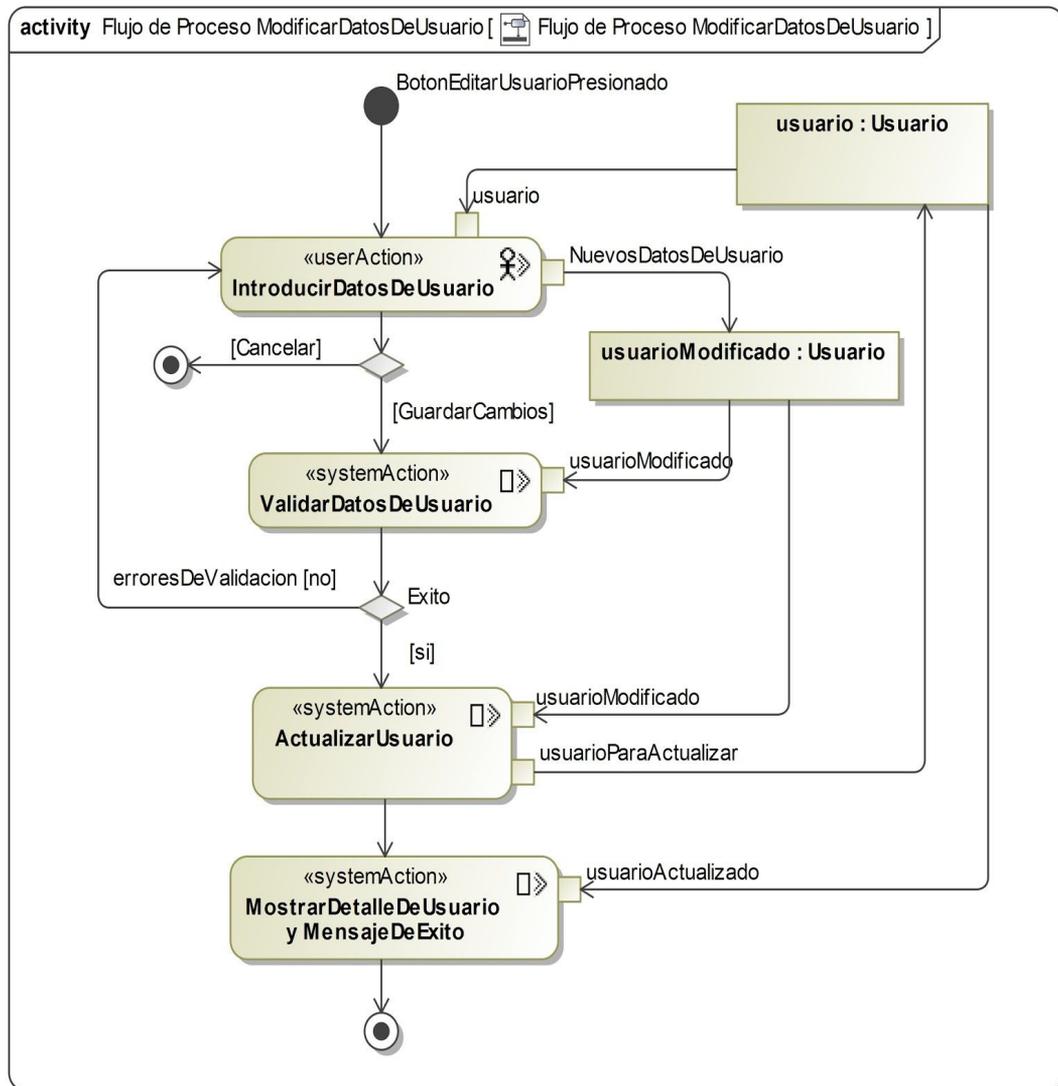


Figura [34]: Diagrama de Flujo Proceso de *ModificarDatosDeUsuario*.

En esta figura se observan las diferentes acciones que se llevan a cabo para que un usuario pueda modificar sus datos.

El proceso inicia cuando el usuario presiona el botón **Editar**. Un vez presionado, el usuario introduce los nuevos datos en **IntroducirDatosDeUsuario**. Este proceso requiere como entrada un objeto **usuario** que en este caso es el usuario activo por lo que se asume que previamente se obtuvo el objeto usuario de la sesión activa en **usuario:Usuario**. Un vez que se introdujeron los datos, el usuario presiona el botón guardar por lo que el sistema realiza la acción **ValidarDatosDeUsuario**, si no hay errores en la validación el sistema realizar la acción **ActualizarUsuario** y muestra por pantalla los datos actualizados con un mensaje de operación exitosa en **MostrarDetalleDeUsuario y MensajeDeExito**.

4.5.2 Diagramas de Proceso del Paquete Gestión de Expediente

En la figura[35] se observan dos clases de procesos abstractos *ProcesarExpediente* y *ProcesarTramite*. Estos permiten respectivamente representar los datos requeridos en los procesos, necesarios en la gestión de expedientes. Por ejemplo *NuevoExpediente* hereda la clase de proceso *IntroducirDatosDeExpediente* (permite obtener los datos para un nuevo expediente). Además, se compone de la clase de proceso *NuevoTramite* que hereda la clase de proceso *IntroducirDatosDeTramite* (permite obtener los datos para nuevo trámite). En la figura[37] se puede observar, con más detalle, el proceso de crear un nuevo expediente cuando un usuario presiona el botón *NuevoExpediente* de la figura[26].

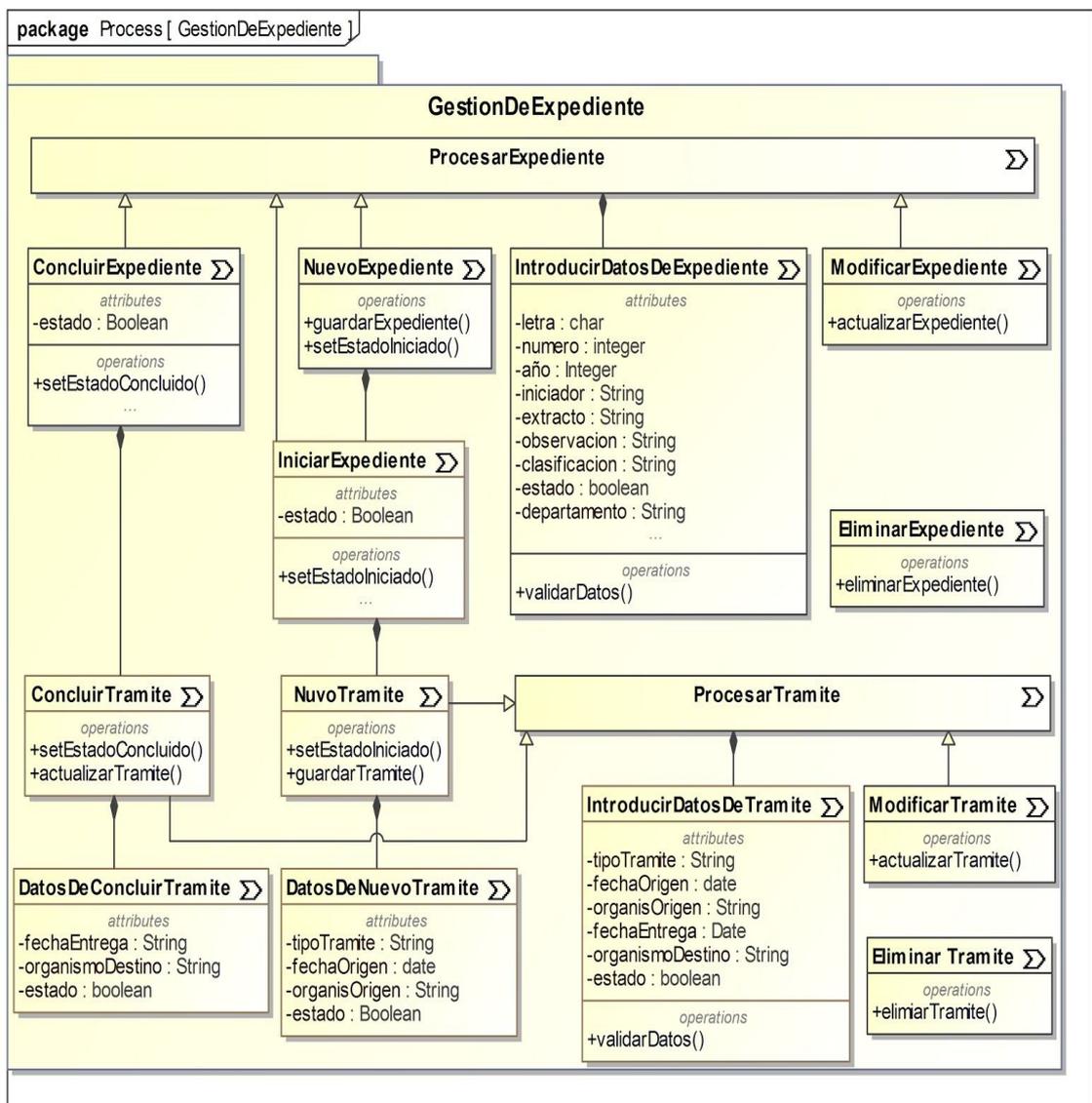


Figura [35]: Diagrama de Estructura de Procesos del Paquete Gestión de Expediente.

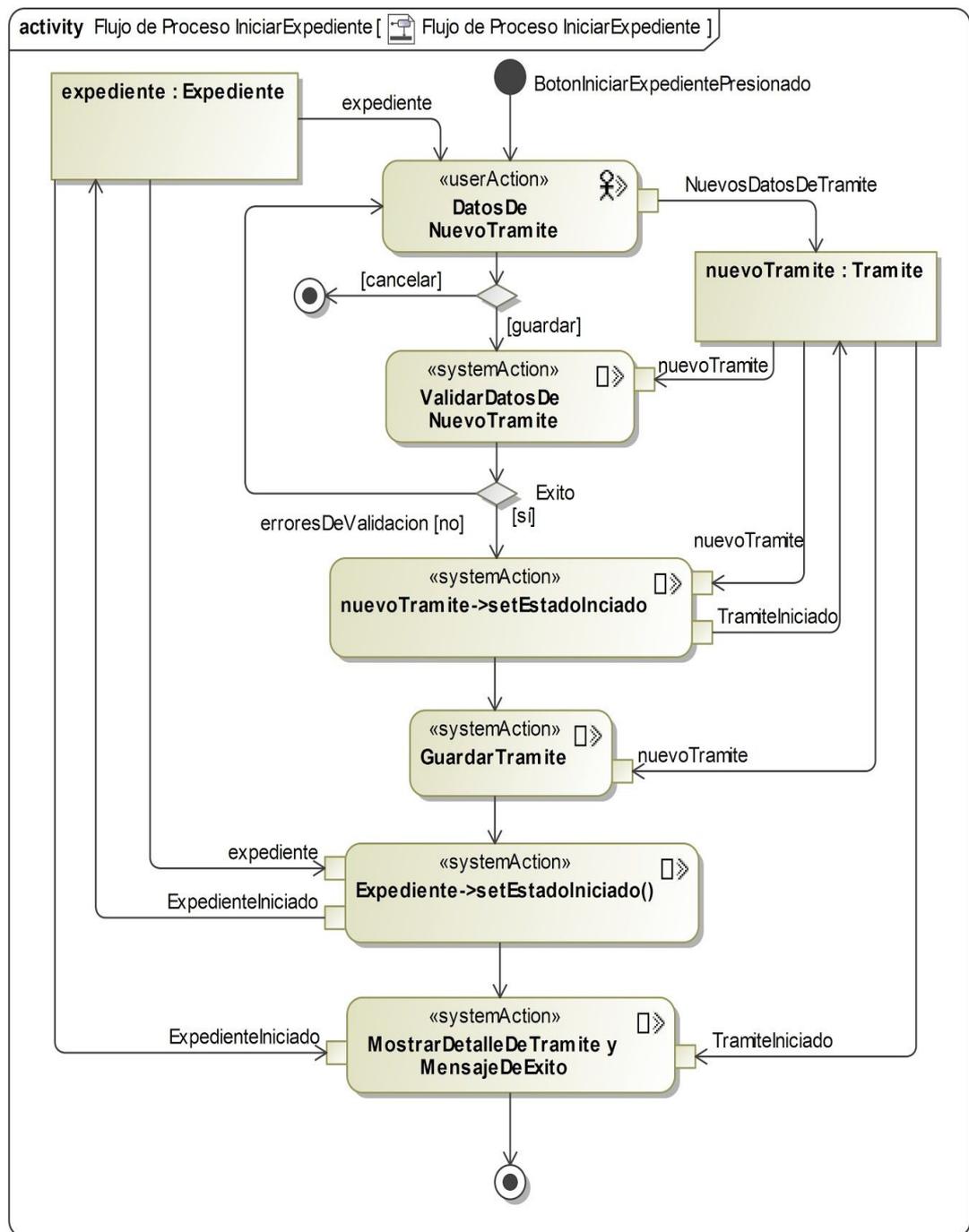


Figura [36]: Diagrama de Flujo Proceso de *IniciarExpediente*.

En esta figura se observan los procesos necesario para iniciar un expediente que implica la creación de un nuevo trámite para el expediente, cuando el usuario presiona el botón **Iniciar** de la figura[26] .

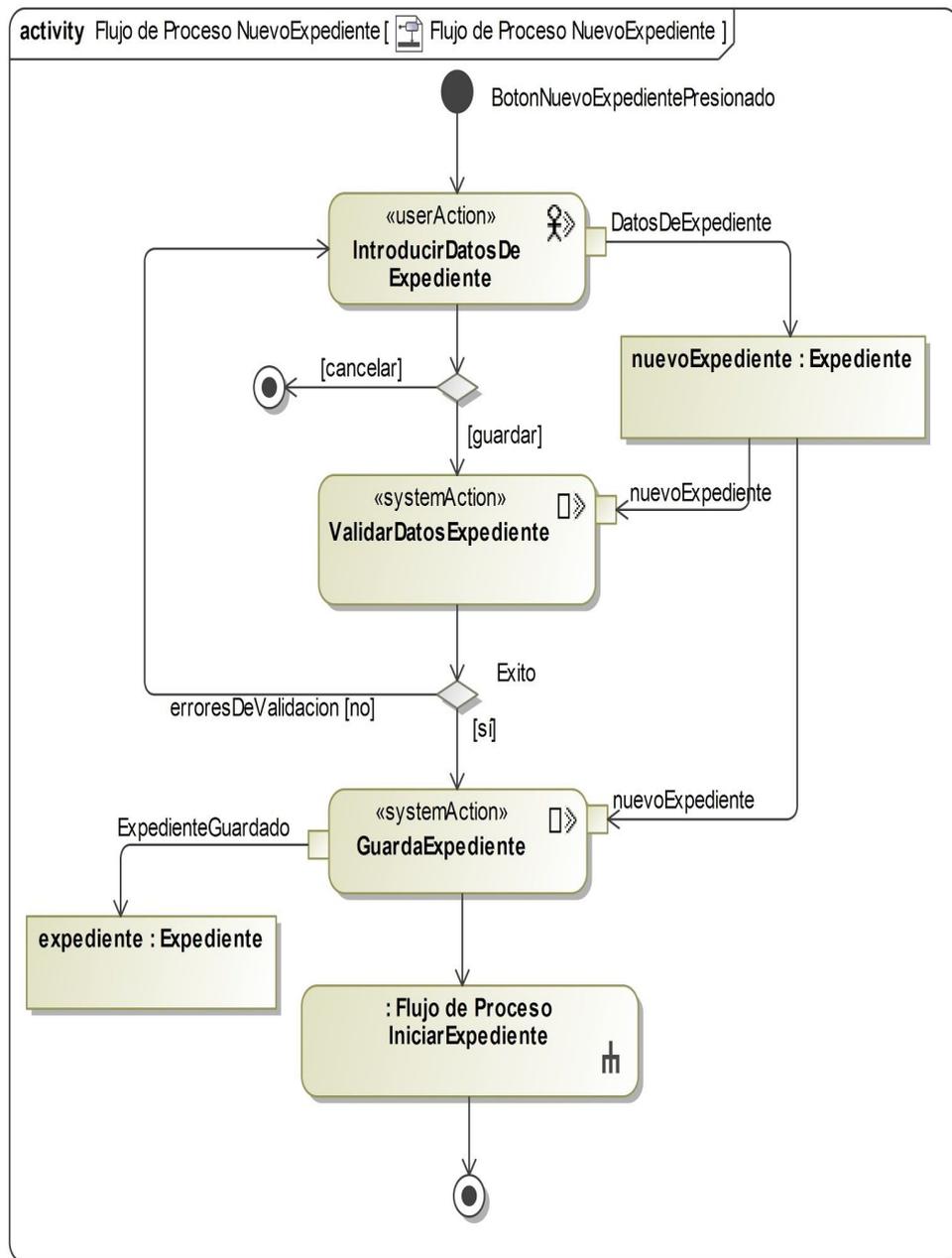


Figura [37]: Diagrama de Flujo Proceso de *NuevoExpediente*.

En esta figura se observa que una vez que se guarda un expediente en *GuardarExpediente* se recupera el expediente guardado en el objeto *expediente:Expediente* debido a que es necesario en el *flujo de proceso IniciarExpediente* como se observa en la figura[36].

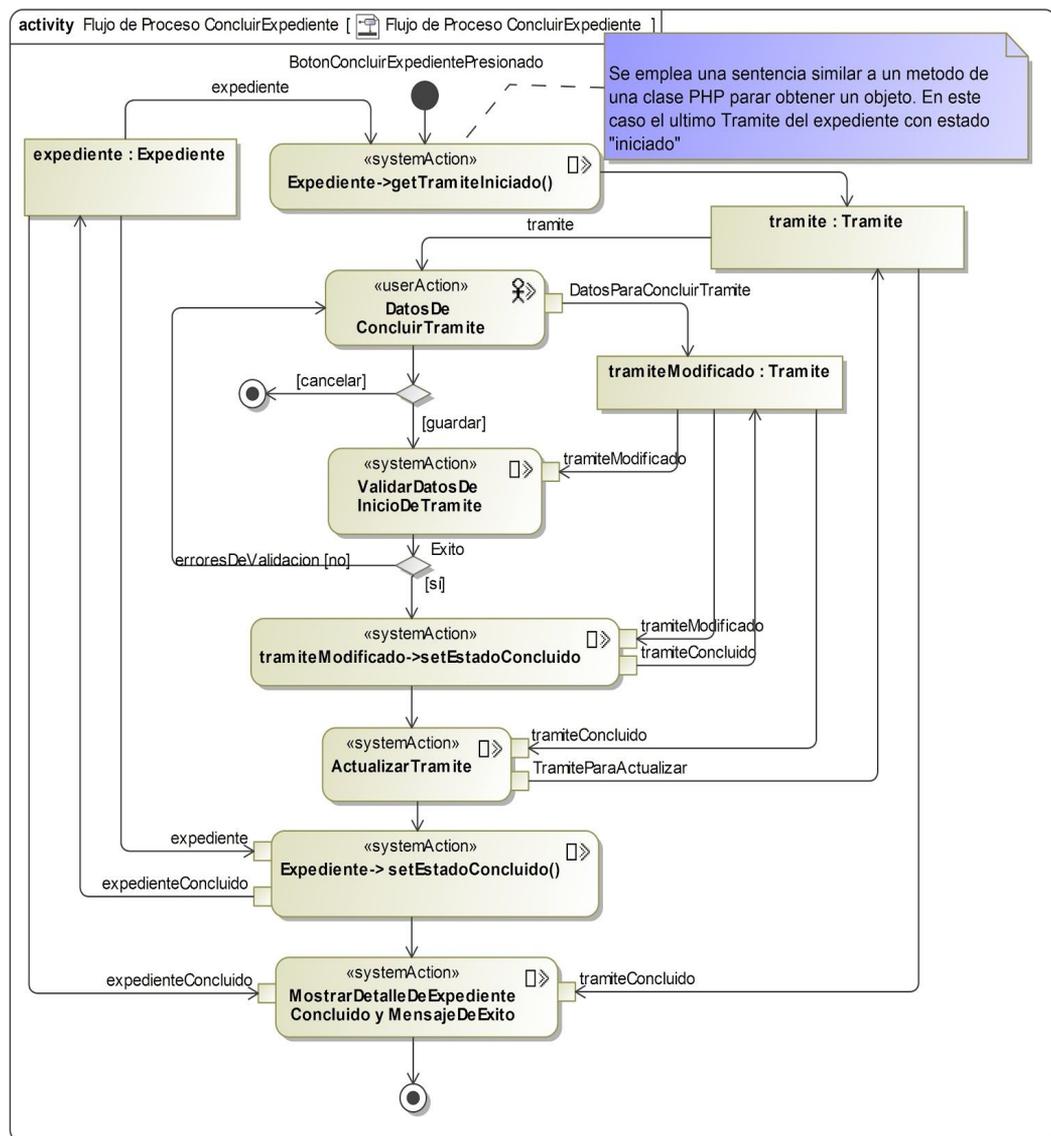


Figura [38]: Diagrama de Flujo Proceso de *ConcluirExpediente*.

En esta figura se observan los procesos necesario para concluir un expediente. En primer lugar se debe recuperar el objeto **tramite** activo, es decir, con estado “iniciado” por lo que el sistema ejecuta la acción **expediente->getTramiteIniciado()**.

4.5.3 Diagramas de Proceso del Paquete Gestión de Documento

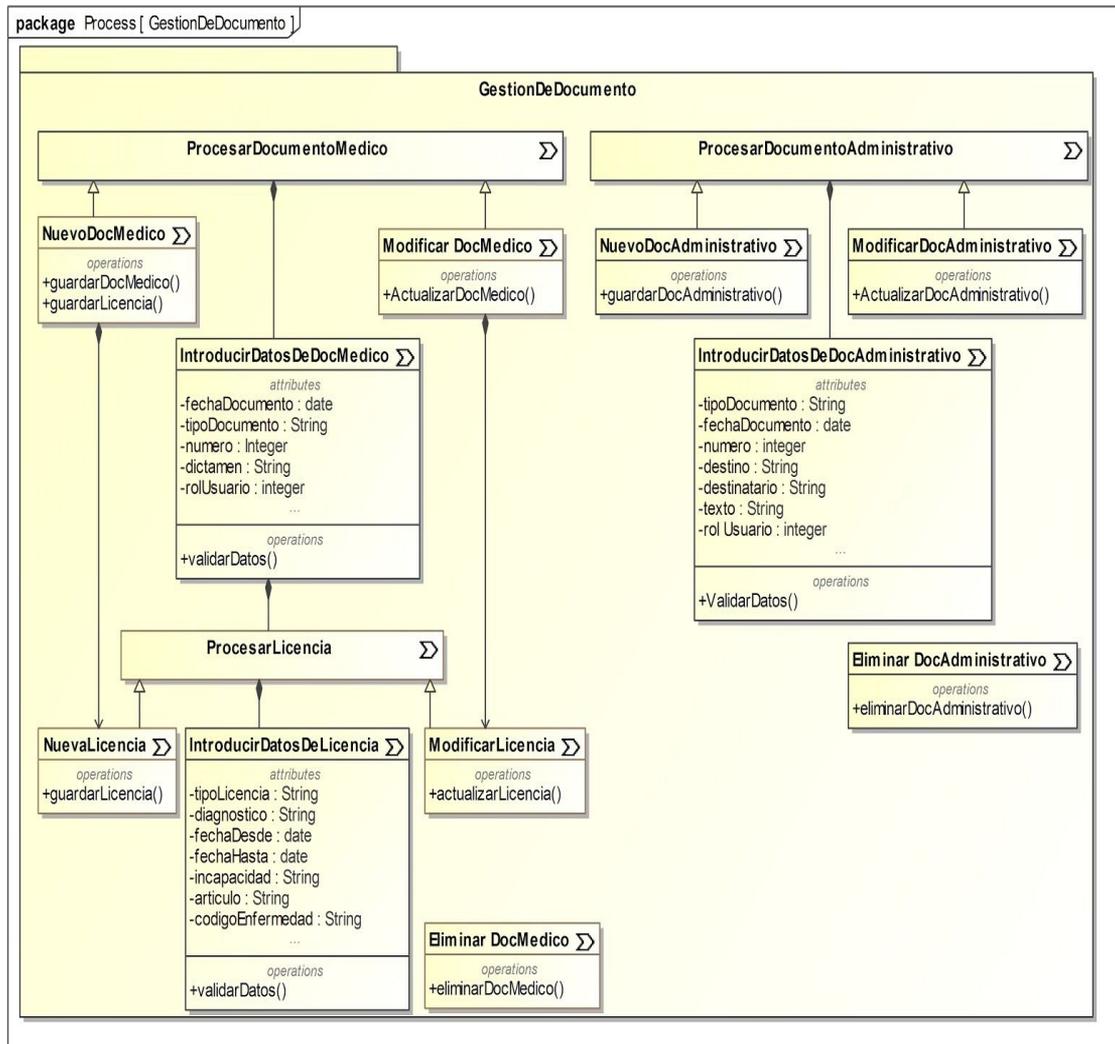


Figura [39]: Diagrama de Proceso del Paquete Gestión de Documento.

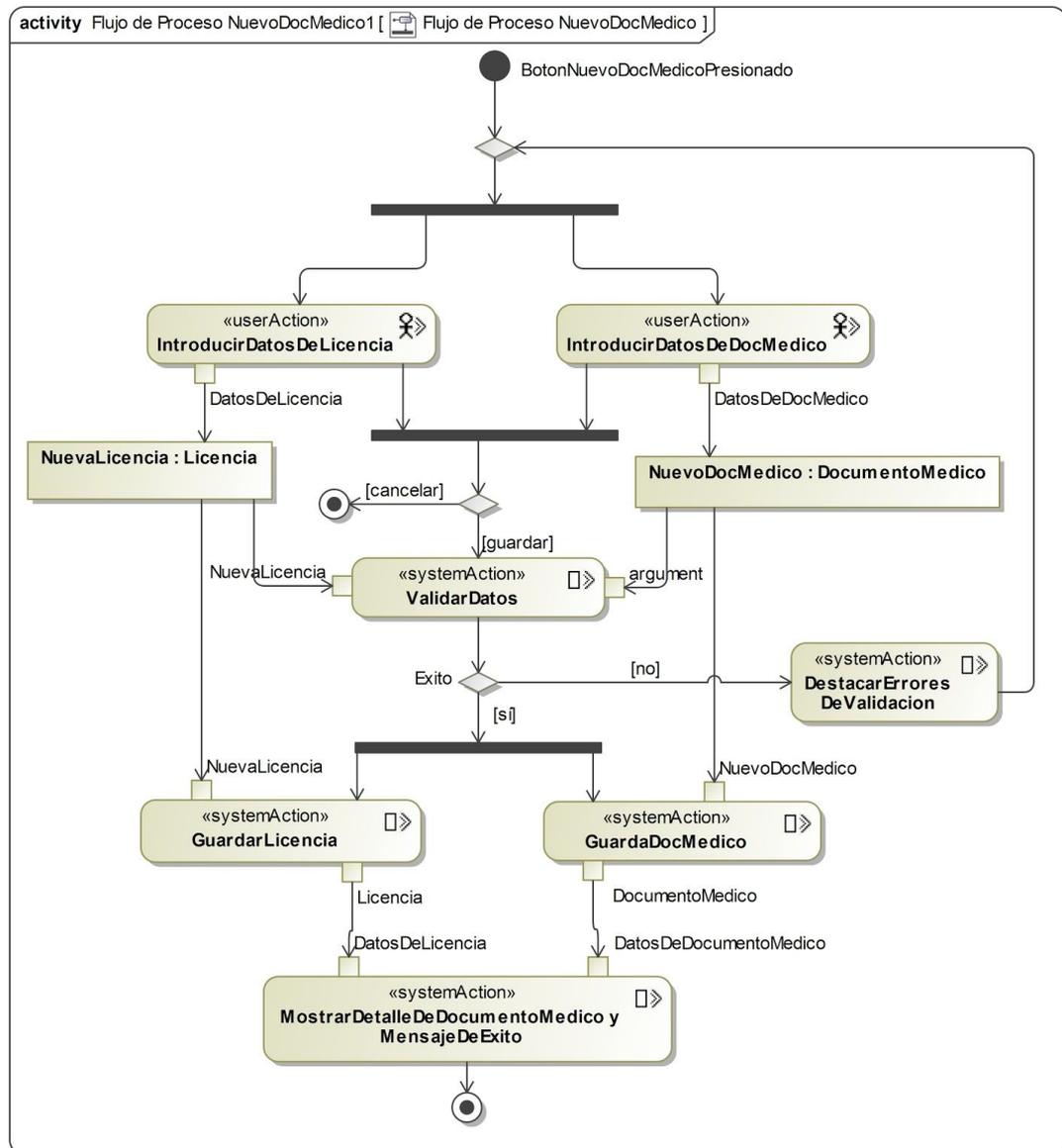


Figura [40]: Diagrama de Flujo Proceso de *NuevoDocMedico*.

En esta figura se observa que el usuario introduce simultáneamente los datos de los objetos *DocMedico* y *Licencia* debido a que estos datos se suministran en un único formulario como puede observarse en la figura[23].

CAPÍTULO V: DESARROLLO DEL PROTOTIPO SIARME

5.1 INTRODUCCIÓN

En este capítulo se presentarán las páginas web desarrolladas para el prototipo, separando el código en tres aplicaciones o módulos:

- Módulo de la Zona Pública o Portada en la sección 5.2.
- Módulo de la Zona Restringida o Intranet en la sección 5.3.
- Módulo de Administración en la sección 5.4.

5.2 DESARROLLO DE LA ZONA PÚBLICA O PORTADA

En la figura[41] se observa la página principal de la Zona Pública o Portada del prototipo, cuyo diseño se corresponde al Modelo de Presentación de la figura con las opciones disponibles para los usuarios no autenticados. Estas opciones se corresponden con las descritas en el Modelo de Presentación de la la figura[25] que permitió obtener múltiples ventajas.



Figura [41]: Página Principal o Portada.

Cuando un usuario accede desde otro dispositivo que no sea una computadora, por ejemplo, desde una tablet o teléfono móvil, las técnicas de diseño y desarrollo web empleadas en éstas, permiten que SIARMe se adapte a dichos entornos. La figura[42] muestra un ejemplo de cómo se vería la Portada en un dispositivo móvil.

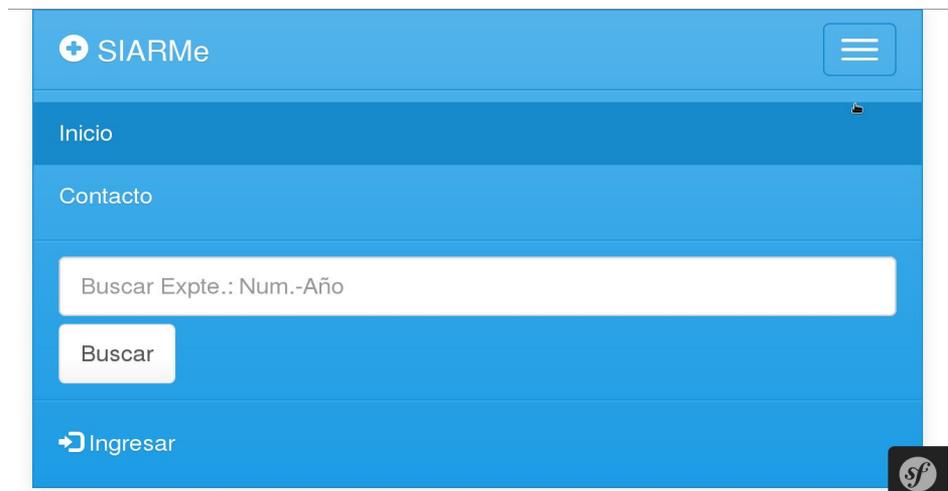


Figura [42]: Diseño Adaptativo del Prototipo SIARMe.

Cuando un usuario realiza la búsqueda de un expediente por número y año los resultados se muestran en la página de la figura[43] y esta funcionalidad se describe en el caso de uso CU5: *Buscar Expediente Por Numero* del Análisis de Requisitos (ver figura[20]).



Figura [43]: Página de Consulta de Expediente de la Portada.

La gestión de usuarios en las aplicaciones web engloban funcionalidades que suelen ser repetitivas y requieren de una compleja implementación. Sin embargo, el framework Symfony fue seleccionado exitosamente, ya que provee librerías que han permitido crearlas de forma sencilla, rápida y confiable con el fin de obtener un Prototipo de sistema seguro.

El sistema de seguridad de Symfony se basa, específicamente, en identificar primero al usuario (autenticación) para, luego comprobar, si ese usuario tiene acceso a ciertos recursos solicitados (autorización), o no. En la figura[44] se observa la página de autenticación cuando el usuario hace clic en la opción **Ingresar**. Esta funcionalidad se

describe en el caso de uso CU1: *Autenticar Usuario* del Análisis de Requisitos y su diseño se corresponde con el Modelo de Presentación la figura[25].

Figura [44]: Formulario de Autenticación de la Portada.

Cuando un usuario escribe los datos “Nombre de usuario” y “Contraseña” y luego clikea el botón **Ingresar**, el sistema realiza su autenticación. En caso de que la misma sea exitosa, se le presenta la página principal de la Zona Restringida o Intranet como se muestra en la figura[42]. Su diseño se corresponde con el modelo de Presentación de la Figura [26]. Esta funcionalidad corresponde al CU1: **Autenticar Usuario** (ver figura[20]).

5.3 DESARROLLO DE LA ZONA RESTRINGIDA O INTRANET

En esta figura[45] se observan las opciones para el usuario con rol **Empleado** (empleado administrativo del Área Mesa de Entrada). En esta página se distinguen las siguientes funcionalidades que corresponden a los casos de uso del Análisis de Requisitos:

- CU6: *Buscar Expediente* (comprende el formulario de búsqueda del menú principal y las opciones del menú lateral);
- CU5: *Crear Expediente* (del menú principal);
- CU7: *Listar Expedientes*: se observa la lista de expedientes donde cada registro muestra un botón **+ 4** CU9: *Listar Trámites* (para listar los trámites asociados al expediente que a su vez cada registro muestra un botón **+ 4** CU12: *Listar Documentos* (para listar los documentos asociados a un trámite pero no puede consultar la información de los documentos por Regla del Negocio RN8) y las opciones de Iniciar **Iniciar** CU8: *Iniciar Expediente* o Concluir **Concluir** CU11: *Concluir Expediente*.

Figura [46]: Página Principal del Rol Empleado - Intranet.

En la figura[47] se observan las opciones a las que podrá tener acceso el usuario con rol Administrativo del Departamento Autoseguro. Asimismo, sucede con el Departamento Despacho, por ejemplo, ambos podrán consultar y filtrar los expedientes con estado “INICIADO”, según regla de negocio RN7 de los expedientes, que se tramiten en cada departamento.

A continuación se distinguen algunas de las funcionalidades que corresponden a los casos de uso del Análisis de Requisitos:

- CU15: *Crear Documento Administrativo*: se presenta un menú  en el cual el usuario puede seleccionar de una lista desplegable los distintos tipos de documentos administrativo que desea crear. En la figura[48] se muestra un ejemplo del formulario para crear un documento Proveído.
- CU13: *Consultar Documento Administrativo*. Una vez que se crea un documento éste se agrega a la lista de documentos asociados al trámite, con la una imagen  que representa el documento y que el usuario puede clicar para ver en detalle, como se muestra en el ejemplo de la figura[49].
- CU17: *Generar Informe*: Se observa esta opción en el menú principal y permite generar informes de expedientes.

The screenshot shows the main administrative interface. At the top, there is a navigation bar with the text 'AUTOSEGURO: Área Administrativa', a search box, and buttons for 'Buscar', 'Generar Informe', and 'Salir'. On the left, there is a sidebar with a list of categories and their counts: Autoseguro (27), CAPRESCA (6), DOCENTES (3), POLICIA (5), SALUD (8), VARIOS (5), Turnos (0), and Citaciones (0). The main area displays a table of processes with columns: Tramites, Expediente, Agente, Extracto, and Estado. Two processes are visible: one with expediente 'D-2493-2013' and another with 'T-1146-2015'. The second process is expanded to show a detailed view of a 'Proveido' document. This view includes a table of documents with columns: Documentos, Trámite, Fecha Inicio, Estado, and Crear Documento. Two documents are listed: 'Informe Médico' (Nº: 43, 01-09-2016) and 'Proveido' (Nº: 6, 05-05-2016). A dropdown menu is open over the 'Proveido' document, showing options: 'Administrativo', 'Proveido', 'Turno', and 'Citación'. Below the table, there is a 'GUARDAR' button.

Figura [47]: Página Principal del Rol Administrativo - Intranet.

The screenshot shows a form titled 'PROVEIDO'. The form contains the following fields and elements: 'REFERENCIA: T-1146-2015', 'INICIADOR: GUTIÉRREZ JIMÉNEZ MARÍA CARMEN', and 'EXTRACTO: SOLICITA JUNTA MÉDICA'. Below this, there is a 'Fecha:' field with three dropdown menus showing '29', '11', and '2016'. A section titled 'Se remiten las presentes actuaciones A:' contains a text box with the value 'Dirección Provincial de Recursos Humanos'. Below that, a 'Motivo/Razon:' field contains the text 'Para continuidad de Trámite|...'. At the bottom right, there is a red 'GUARDAR' button.

Figura [48]: Formulario para crear un documento Proveído.



Poder Ejecutivo
Secretaría General De La Gobernacion
Subsecretaria De Recursos Humanos Gestion Pública
DIRECCION PROVINCIAL SERVICIO DE RECONOCIMIENTO

San Fernando Del Valle de Catamarca, Miércoles, 18 de Mayo de 2016

PROVEIDO N°: 08

REFERENCIA: T-1146-2015
INICIADOR: GUTIÉRREZ JIMÉNEZ MARÍA CARMEN
EXTRACTO: SOLICITA JUNTA MÉDICA

-----Remítanse las presentes actuaciones a: **Dirección Provincial de Recursos Humanos**. Nulla scelerisque blandit ligula eget hendrerit. Sed malesuada, enim sit amet ultricies semper, elit leo lacinia massa, in tempus nisl ipsum quis libero. Donec dapibus blandit odio, in auctor turpis commodo ut. Phasellus eu leo tellus, et accumsan libero. Pellentesque fringilla ipsum nec justo tempus elementum. Aliquam dapibus metus aliquam ante lacinia blandit. Donec ornare lacus vitae dolor imperdiet vitae ultricies nibh congue.

Editar Eliminar Imprimir sf

Figura [49]: Página de detalle de documento Proveído.

En la figura[50] se muestra la página para el usuario con rol Médico, donde se listan tanto los documentos médicos como administrativos, pero sólo se podrán consultar los documentos médicos (Regla del Negocio RN6).

Se distinguen dos funcionalidades que corresponden a los casos de uso CU16: *Crear Documento Médico* y CU14: *Consultar Documento Médico*, del Análisis de Requisitos.

Aplicación de la Metodología UWE para el Modelado y Desarrollo de un Prototipo de Sistema de Información Administrativa para la Dirección Servicio De Reconocimientos Médicos de la Provincia de Catamarca.

Tramites	Expediente	Agente	Extracto	Estado
+ 2	D-2493-2013	Díaz Alonso Lucía	Sitamet Aut Ligula Malesuada Pendisse Natoque Aliquam Lacinia	INICIADO
+ 1	T-1146-2015	Gutiérrez Jiménez María Carmen	Solicita Junta Médica	INICIADO
+ 4	I-2693-2013	Jiménez Medina Francisco	Lorem Aut Dum Tincidunt Aliquam	INICIADO
+ 1	P-736-2016	Medina Medina Laura	Sitamet Aut Ut Dum Scelerisque Blandit Malesuada Lacinia	INICIADO

Figura [50]: Página Principal del Rol Médico.

Finalmente, en la Figura[51], se muestra el menú donde el usuario podrá ingresar sus datos personales, su contraseña actual, o bien, modificarlos en el botón **Editar**; confirmando su elección. Esta funcionalidad de corresponde al caso de uso CU3: *Actualizar Datos de Usuario*.

Apellido y Nombre: Nuñez Blanco Sergio
 Usuario: medico
 Direccion: Plaza Adipiscing, N°:
 DNI: 24212731
 Telefono: 4446426
 Telefono movil: 383

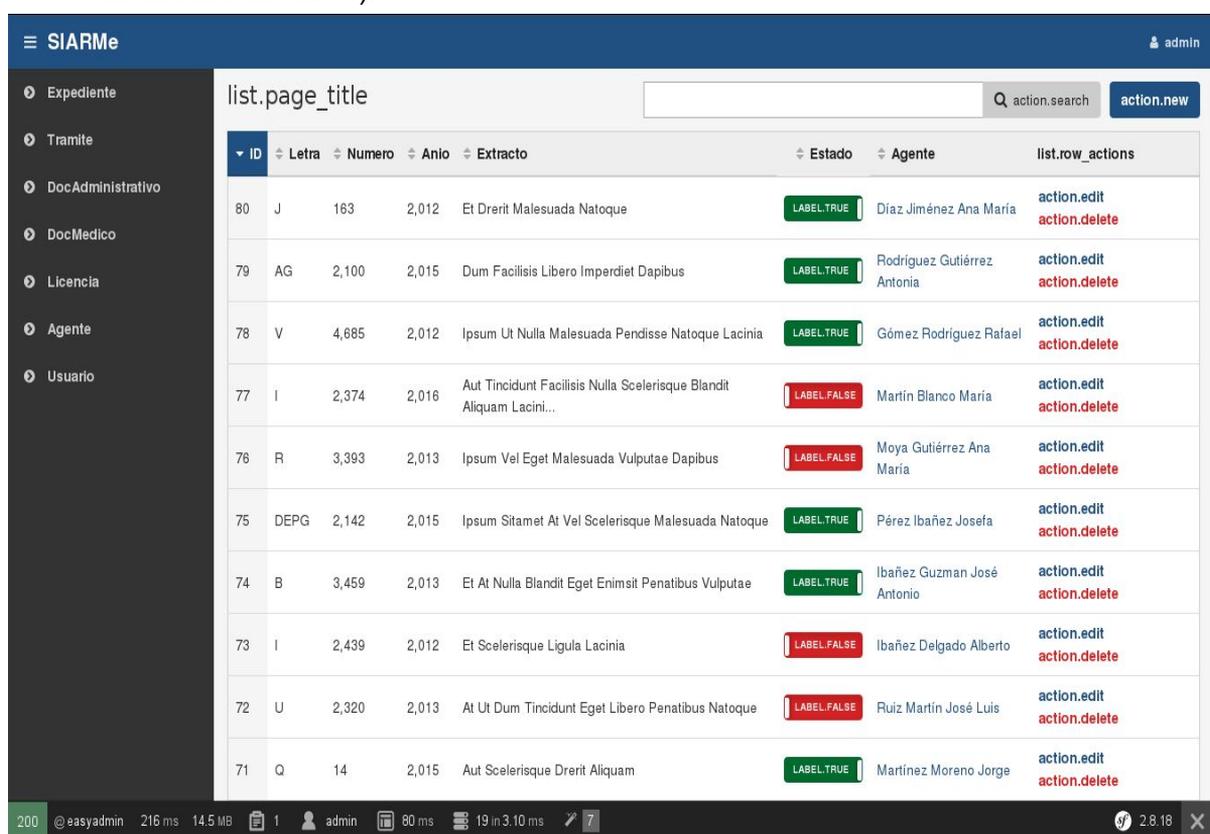
© 2016 - Direccion Provincial de Reconocimientos Médicos

Figura [51]: Página de Datos de Usuario.

5.4 DESARROLLO DEL MÓDULO DE ADMINISTRACIÓN

Este último módulo es la parte de administración del Prototipo, no se abordó su análisis y modelado debido a su sencillez en la implementación. Además, el framework Symfony permite la generación automática de este módulo con algunas configuraciones mínimas. Por una parte, se debe, a que está restringido a todos los usuarios, excepto los administradores del sistema. Por otro lado, la gestión de la información se divide en función de cada entidad (usuario, expediente, documento, etc.) y comprende cuatro operaciones básicas, referidas habitualmente con el acrónimo CRUD :

- Crear (create): crea una nueva entidad (equivalente a añadir una fila en la tabla de la base de datos).
- Ver (read): muestra todos los datos de una entidad. Se utiliza para ver el detalle de la información sin tener que entrar en el formulario que modifica sus datos.
- Actualizar (update): actualiza una o más propiedades de una entidad.
- Borrar (delete): borra una entidad completa (equivalente a borrar una fila en la tabla de la base de datos).



The screenshot shows the SIARMe administration interface. On the left is a sidebar with navigation items: Expediente, Tramite, DocAdministrativo, DocMedico, Licencia, Agente, and Usuario. The main content area displays a table titled 'list.page_title' with a search bar and an 'action.new' button. The table has columns for ID, Letra, Numero, Anio, Extracto, Estado, Agente, and list.row_actions. The data rows are as follows:

ID	Letra	Numero	Anio	Extracto	Estado	Agente	list.row_actions
80	J	163	2,012	Et Drerit Malesuada Natoque	LABEL.TRUE	Diaz Jiménez Ana María	action.edit action.delete
79	AG	2,100	2,015	Dum Facilisis Libero Imperdiet Dapibus	LABEL.TRUE	Rodriguez Gutiérrez Antonia	action.edit action.delete
78	V	4,685	2,012	Ipsum Ut Nulla Malesuada Pendisse Natoque Lacinia	LABEL.TRUE	Gómez Rodríguez Rafael	action.edit action.delete
77	I	2,374	2,016	Aut Tincidunt Facilisis Nulla Scelerisque Blandit Aliquam Lacini...	LABEL.FALSE	Martín Blanco María	action.edit action.delete
76	R	3,393	2,013	Ipsum Vel Eget Malesuada Vulputae Dapibus	LABEL.FALSE	Moya Gutiérrez Ana María	action.edit action.delete
75	DEPG	2,142	2,015	Ipsum Sitamet At Vel Scelerisque Malesuada Natoque	LABEL.TRUE	Pérez Ibañez Josefa	action.edit action.delete
74	B	3,459	2,013	Et At Nulla Blandit Eget Enimsit Penatibus Vulputae	LABEL.TRUE	Ibañez Guzman José Antonio	action.edit action.delete
73	I	2,439	2,012	Et Scelerisque Ligula Lacinia	LABEL.FALSE	Ibañez Delgado Alberto	action.edit action.delete
72	U	2,320	2,013	At Ut Dum Tincidunt Eget Libero Penatibus Natoque	LABEL.FALSE	Ruiz Martín José Luis	action.edit action.delete
71	Q	14	2,015	Aut Scelerisque Drerit Aliquam	LABEL.TRUE	Martínez Moreno Jorge	action.edit action.delete

Figura [51]: Página de Administración de SIARMe .

Para finalizar, con cada una de las figuras expuestas, se ha demostrado que el prototipo SIARMe, cumple con la totalidad de los requisitos funcionales del Análisis de Requisitos tratados en el Capítulo IV. Por otro lado, en cuanto a los modelos tratados en el Capítulo V sirvieron de guía para implementar la lógica de la aplicación y la interfaz de usuario de acuerdo con los roles de usuario como se ha demostrado en cada caso.

El uso de la metodología UWE permitió optimizar el desarrollo del Prototipo SIARMe al emplear un enfoque de Ingeniería Web, ya que proporcionó un proceso de diseño sistemático y la definición de un perfil UML propio que provee elementos necesarios para el modelado de los diferentes aspectos de una aplicación WEB.

De esta manera, el Prototipo resultante es un sistema funcional a las demandas administrativas, completo y efectivo.

CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

El Prototipo SIARMe, propuesto para la gestión de la información administrativa, en la DPSRM, permite la informatización e integración de los procedimientos administrativos de la institución, como así también, la gestión integrada y eficiente de todos los expedientes tramitados, evitando la dispersión y la redundancia de la información asociada a un expediente y optimizando el proceso de gestión del mismo. De esta manera, la información queda adherida al entorno de la web, facilitando su acceso y movilidad del personal dentro de la institución.

Para ello, el uso de la norma IEEE-830-1998 permitió recopilar la información que interviene en los procesos de administración teniendo como resultado los requisitos funcionales acorde a cada una de las necesidades.

Además, el Prototipo SIARMe fue desarrollado empleando el Framework Symfony versión 2.8, entre otras herramientas gratuitas y de código abierto como el entorno de desarrollo Eclipse, Servidor Web Apache; motor de base de datos MySQL por lo que se redujo el costo de desarrollo.

En cuanto a la aplicación del enfoque metodológico de Ingeniería Web en una situación problemática particular posibilita valorar positivamente la metodología seleccionada, ya que UWE ofrece una serie de modelos relevantes para tener una visión global del contexto de interoperabilidad y la correspondiente definición y especificación de la forma de procesamiento de datos, navegación, presentación y procesamiento de la información en una aplicación web.

Todo esto permitió optimizar ampliamente el desarrollo del Prototipo SIARMe acorde a las necesidades de la Institución.

6.2 RECOMENDACIONES

Los sitios Web, en su mayoría, deben ser creados de manera rápida, por lo cual es recomendable el uso de librerías probadas que faciliten el desarrollo de las aplicaciones web, como es el caso de Symfony, un framework robusto que incorpora librerías probadas en otros proyectos (Por ejemplo algunos sitios web conocidos que emplean este framework son Yahoo! Answers, Dailymotion, Yahoo! Bookmarks y delicious.com).

Por otro lado, se recomienda migrar el sistema del Departamento Ausentismo a un entorno web, añadiendo las funcionalidades de este sistema al Prototipo SIARMe. Esto beneficiaría en gran medida la integración y unificación de datos y tecnologías en la DPSRM. Así también, las tareas de mantenimiento del software.

REFERENCIA BIBLIOGRÁFICA

- [1]. Historia de la ingeniería del software - Wikipedia, 2016 https://es.wikipedia.org/wiki/Historia_de_la_ingenier%C3%ADa_del_software.
- [2]. Pressman, Roger. "Ingeniería del Software-Enfoque Práctico Mc Graw Hill 5ª." Edición Año, 2002, Capítulo 29.1.1, página 523.
- [3]. Sommerville, I. Ingeniería del software. 7ª edición, España: Pearson -Addison Wesley, 2005, capítulo 4.1.1, página 62.
- [4]. Dcto. Acdo. 1875/94 - Régimen de Licencias, Justificaciones y Franquicias.
- [5]. Dcto. Acdo. N° 2557 – Incorporación de Autoseguro Provincial a la Dirección
- [6]. Programas Ofimaticos <https://es.wikipedia.org/wiki/Ofim%C3%A1tica>
- [7]. Microsoft Acces Gestor de Bases de Datos https://es.wikipedia.org/wiki/Microsoft_Access
- [8]. Institute of Programming and Software Engineering, 2016, UWE – UML-based Web Engineering. Obtenido de <http://uwe.pst.ifi.lmu.de/index.html>,
- [9]. Murugesan, San, et al. "Web engineering: a new discipline for web-based system development." Proc. First Int'l Conf. Software Engineering (ICSE) Workshop on Web Engineering, Univ. of Western Sydney, Australia. 1999.
- [10]. Murugesan, San, and Athula Ginige. "Web engineering: Introduction and perspectives." Web engineering: principles and techniques , 2005.
- [11]. Shklar, Leon, and Richard Rosen. Web application architecture. John Wiley & Sons, 2009.
- [12]. Kappel, Gerti, et al. Web engineering. John Wiley, 2005.
- [13]. Schwinger, Wieland, and Nora Koch. "Modeling Web Applications.", 2006.
- [14]. S. Ceri, P. Fraternali, and A. Bongio, "Web Modeling Language (WebML): a modeling language for designing Web sites," Computer Networks, vol. 33, no. 1–6, pp. 137–157, Jun. 2000.
- [15]. D. Schwabe and G. Rossi, "The object-oriented hypermedia design model," Commun. ACM, vol. 38, no. 8, pp. 45–46, Aug. 1995.
- [16]. N. Koch, A. Knapp, G. Zhang, and Baumeister, H. UML-based web engineering. En Web Engineering: Modelling and Implementing Web Applications. Springer London, 2008. p. 157- 191.
- [17]. N. Koch. "Transformations Techniques in the Model-Driven Development Process of UWE". Proc. 2nd Wsh. Model-Driven Web Engineering (MDWE'06), Palo Alto, 2006.
- [18]. G. Booch, J. Rumbaugh, I. Jacobson; UML. El lenguaje de modelado unificado. Guía del usuario; Ed. Addison Wesley Iberoamericana, 1999.

- [19]. ArgoUWE - CASE Tool for Modelling Web Applications, 2016. Obtenido de <http://uwe.pst.ifi.lmu.de/toolargoUWE.html>
- [20]. MagicUWE: UWE Plugin for MagicDraw., 2016. Obtenido de <http://uwe.pst.ifi.lmu.de/toolMagicUWE.html>
- [21]. UWE – UML-based Web Engineering, 2016, obtenido de <http://uwe.pst.ifi.lmu.de>
- [22]. N. Koch, A. Kraus, and R. Hennicker. The authoring process of the uml-based web engineering approach. En First International Workshop on Web-Oriented Software Technology. 2001.
- [23]. M. Busch and M. A. G. de Dios. ActionUWE: Transformation of UWE to ActionGUI Models. Transformation, 2012, vol. 3, p. 2.
- [24]. Tutorial sobre UWE , 2016, Obtenido de <http://uwe.pst.ifi.lmu.de/teachingTutorialSpanish.html>
- [25]. Google Trends, 2016 <https://www.google.com/trends/explore#q=symfony%2C%20cakephp%2C%20zend%20framework%2C%20kumbia&date=1%2F2014%2030m&cmpt=q&tz=Etc%2FGMT%2B3>
- [26]. Symfony 2.4, el libro oficial, 2016, Obtenido de Wikipedia, 2016, http://librosweb.es/libro/symfony_2_x/
- [27]. Framework CakePHP, 2016, Obtenido de Wikipedia, 2016, <https://es.wikipedia.org/wiki/CakePHP>
- [28]. Framework Symfony, 2016, Obtenido de Wikipedia, 2016, <https://es.wikipedia.org/wiki/Symfony>
- [29]. Patron de Arquitectura Modelo-Vista-Controlador, Obtenido de Wikipedia, 2016, <https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%20controlador>
- [30]. Fowler M., Scott K. UML gota a gota; Ed. Addison Wesley México, 1999.

ANEXO I: Instalación del Prototipo SIARMe

En el Sitio oficial (<https://symfony.com>) se puede encontrar información que detalla cómo instalar Symfony, exitosamente.

A continuación, sólo se indican los principales pasos necesarios a seguir para una exitosa instalación del prototipo SIARMe..

En primer lugar, el usuario del sistema, debe tener Composer (<https://getcomposer.org/>) instalado globalmente.

En el caso de tener Linux o Mac OS X , deben ejecutarse los siguientes comandos:

```
$ curl -sS https://getcomposer.org/installer | php
$ sudo mv composer.phar /usr/local/bin/composer
```

Si se utiliza Windows, debe descargar el instalador ejecutable de Composer y seguir los pasos indicados por el instalador. En el proyecto se utilizó Git(<https://git-scm.com/>) y GitHub(<https://github.com>) para gestionar todo su código.

Una vez instalado Composer y Git se deben ejecutar los siguientes comandos.

```
# clona el código de la aplicación
$ cd proyectos/
$ git clone https://github.com/ira-adrian/SIARMe.git
```

El directorio /proyectos es donde se alojan los proyectos web. Por ejemplo en Linux en el directorio raíz de Apache (/var/www/proyectos).

```
# Instala las dependencias del proyecto (incluyendo Symfony)
$ cd SIARMe/
$ composer install
```

Probando la aplicación:

La forma más sencilla de probar la aplicación, es ejecutando el siguiente comando, que arranca el servidor web interno de PHP y hace que su aplicación se pueda ejecutar sin necesidad de usar Apache o Nginx:

```
$ php app/console server:run
[OK] Server running on http://127.0.0.1:8000
```

Una vez concluida, esta instancia, abrir el navegador y acceder a <http://localhost:8000> para probar la aplicación. Este comando requiere PHP 5.4 o superior.

Solución a los problemas comunes:

Si solamente se observa una página en blanco, es posible que se trate de un problema de permisos. Una solución rápida puede ser ejecutar el siguiente comando:

```
$ cd proyectos/SIARMe/
$ chmod -R 777 app/cache app/logs
```

Si no funciona esta solución, se puede encontrar más información en el sitio oficial de Symfony.

El Prototipo SIARMe usa por defecto una base de datos MySQL en la instalación de la aplicación. Aunque con una configuración mínima puede emplearse otra como SQLite o PostgreSQL esta información está disponible en el sitio oficial.

En este punto deben ejecutarse los siguientes comandos para crear la base de datos y rellenarla con datos de prueba:

```
$ php app/console doctrine:database:create
$ php app/console doctrine:schema:create
$ php app/console doctrine:fixtures:load
# si este último comando da error, ejecuta en su lugar:
$ php app/console doctrine:fixtures:load --append
# para gestionar activos CSS, JavaScript y archivos de imagen
$ php app/console assets:install --symlink
# para el entorno de Producción
$ php app/console assetic:dump --env=prod --no-debug
```

Accediendo al Prototipo SIARMe

Portada

URL:

- Entorno de desarrollo: http://localhost/web/app_dev.php/
- Entorno de producción: <http://localhost/web/app.php/>

Intranet

URL:

- Entorno de desarrollo: http://localhost/web/app_dev.php/intranet
- Entorno de producción: <http://localhost/web/app.php/intranet>
- Usuarios de Prueba:
 - Usuarios de prueba del Departamento AUTOSEGURO: {empleado, administrativo, medico}
 - Contraseña: es el nombre de usuario.
 - Usuarios de prueba del Departamento DESPACHO: {empleado1, administrativo1, medico1}
 - Contraseña: es el nombre de usuario.

Administración

URL:

- Entorno de desarrollo: http://localhost/web/app_dev.php/backend
- Entorno de producción: <http://localhost/web/app.php/backend>
- Usuario Administrador:
 - Nombre de usuario: admin
 - Contraseña: 1234