

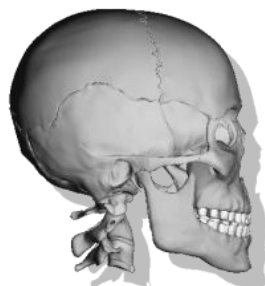
# Universidad de las Ciencias Informáticas

## Facultad 5



### **Título: Sistema para la visualización de imágenes y modelos 3D médicos en la WEB con fines educativos**

Trabajo de Diploma para optar por el título de  
Ingeniero en Ciencias Informáticas



**Autor(es):** Leandro Raúl Ramírez Batista

Reinier Talancón Díaz

**Tutor(es):** Ing. Leonardo Rafael Fernández Ruiz

Ing. Osvaldo Pereira Barzaga

Junio 2010

## DECLARACIÓN DE AUTORÍA

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

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

Firma del autor: Leandro Raúl Ramírez Batista

---

Firma del autor: Reinier Talancón Díaz

---

Firma del tutor: Leonardo Rafael Fernández Ruiz

---

Firma del tutor: Osvaldo Pereira Barzaga

## DATOS DE CONTACTO

**Tutor:** Ing. Leonardo Rafael Fernández Ruiz.

**Edad:** 26 años.

**Ciudadanía:** cubano.

**Institución:** Universidad de las Ciencias Informáticas (UCI).

**Título:** Ingeniero en Ciencias Informáticas.

**Categoría Docente:** Profesor Adiestrado.

**E-mail:** lfernandez@uci.cu.

Graduado de la UCI, experiencia en representación 3D e interacción en la web, pertenece a la Línea de Software Empotrado del CEDIN en la Universidad de las Ciencias Informáticas.

**Tutor:** Ing. Osvaldo Pereira Barzaga.

**Edad:** 26 años.

**Ciudadanía:** cubano.

**Institución:** Universidad de las Ciencias Informáticas (UCI).

**Título:** Ingeniero en Ciencias Informáticas.

**Categoría Docente:** Profesor Adiestrado

**E-mail:** opereira@uci.cu

Graduado de la UCI, con dos años de experiencia en el tema de la Gráfica Computacional y es Jefe de la Línea de Visualización Científica en la Universidad de las Ciencias Informáticas.

## AGRADECIMIENTOS

*Leandro:*

*Le agradezco a mi madre que tanto ha hecho por mí sin ninguna condición ni objeción, siempre apoyándome y dándome fuerzas para seguir estudiando. A mi padre por enseñarme desde pequeño los métodos más básicos para forjarme como un profesional y por ayudarme en todo momento. A ustedes muchas gracias por todo lo que han hecho por mí y por darme tanto amor y cariño. Quisiera agradecerle a mi abuela por ayudarme siempre y por confiar en mí y a mi abuelo por su apoyo incondicional. También quiero agradecerle a mi hermana por compartir conmigo todos los momentos que estuvimos lejos de nuestros seres queridos y por darme su apoyo.*

*En general quiero agradecerles a todos mis amigos que me ayudaron en todo momento, a nuestros tutores y en especial a mi compañero de tesis Talancón.*

## AGRADECIMIENTOS

*Reinier:*

*Desde lo más profundo de mi corazón le agradezco a mi mamá, la vida me privó de tu presencia y apoyo en estos momentos tan importantes, pero quiero que sepas mimi que te agradezco tu entrega, tu dedicación, todo lo que he alcanzado y soy, es gracias a ti, todo lo que hago y haré siempre será inspirado en tu persona. Le agradezco a mi padre por enseñarme a superar los obstáculos de la vida y no perderme por un camino equivocado. Quiero agradecerle a mis dos tías Idania y Zeida por brindarme el apoyo dentro del dolor, la confusión y la incertidumbre que se apoderaron de mi ser al perder a mi madre, gracias a ustedes hoy puedo ser un profesional, gracias por convertirse en mi confianza, en las portadoras del amor y la comprensión que solo una madre es capaz de brindar. Le doy las gracias a mi hermanita Diana por confiar en mí, por brindarme su amor en todo momento, eres una de las razones por las que hoy sigo adelante, también agradecerle a mis primos considerados para mí como hermanos, Adrianny, Gianni y Yanet. Un especial agradecimiento a mi pareja Misleydis por su entrega y apoyo, te has ganado un lugar especial dentro de mi vida. De forma general le agradezco a todos aquellos que me tendieron la mano y me ayudaron a seguir adelante, a mis compañeros de universidad, de aula, a mis tutores y agradecerle sinceramente a Leandro ya que sin su ayuda, conocimientos y amistad hubiera sido imposible realizar este trabajo, fue una experiencia muy grata haber trabajado a su lado.*

## DEDICATORIA

*Leandro:*

*Le dedico esta tesis a mis padres, a mi hermana y a mis abuelos.*

*Reinier:*

*Le dedico este trabajo a mis padres, a mi hermana y a mis tías.*

## RESUMEN

El desarrollo de las Tecnologías Informáticas de las Comunicaciones (TIC) ha posibilitado la utilización de software educativos para la enseñanza de la anatomía del cuerpo humano, estos se han convertido en una herramienta muy valiosa para lograr un proceso de aprendizaje eficaz y su gran mayoría se basan en aplicaciones multimedia. Con el desarrollo de la realidad virtual y la WEB se abre un nuevo camino para la creación de entornos distribuidos de bajo consumo de hardware involucrando modelos tridimensionales, los cuales contribuyen a un mejor acceso y distribución de dicha información en los centros de enseñanza.

Con el objetivo de brindar las facilidades antes expuestas es necesario desarrollar e implementar un sistema educativo distribuido de visualización de modelos tridimensionales e imágenes médicas que brinde la información necesaria para posibilitar la interacción y aprendizaje de la anatomía del cuerpo humano con vistas a mejorar los software educativos en nuestros centros de enseñanza. Posibilitando así el desarrollo del aprendizaje en los estudiantes que estudian medicina, específicamente en la rama de la anatomía.

En el presente documento se efectúa un detallado análisis y diseño del sistema, auxiliándose de la metodología RUP y el Lenguaje de Modelado Unificado empleando la herramienta Rational Rose Enterprise Edition.

**Palabras clave: WEB, educativo, tridimensional, modelo, médicas, visualización.**

## ÍNDICE

<b>Declaración de Autoría</b> .....	I
<b>Datos de Contacto</b> .....	II
<b>Agradecimientos</b> .....	III
<b>Dedicatoria</b> .....	V
<b>Resumen</b> .....	VI
<b>Introducción</b> .....	1
<b>Capítulo 1: Fundamentación Teórica</b> .....	4
1.1 La realidad virtual.....	4
1.1.1 La realidad virtual y la educación .....	4
1.1.2 La realidad virtual en la medicina .....	6
1.1.3 Aplicación de la realidad virtual en el estudio de la anatomía .....	6
1.1.4 Uso de las TIC y la realidad virtual en la educación médica cubana.....	7
1.2 Trabajos relacionados hasta el momento de anatomía 3D.....	7
1.2.1. Visible Body .....	8
1.2.2. Visión Médica Virtual .....	9
1.3 Tendencias y tecnologías actuales de desarrollo WEB.....	9
1.3.1 Definición de aplicación WEB.....	9
1.3.2 Ventajas de las aplicaciones WEB .....	10
1.3.3 Estándares de programación WEB .....	10
1.3.4 Lenguajes de programación WEB .....	12
1.3.5 Técnicas de Programación WEB.....	13
1.3.6 Tecnologías de programación WEB .....	14
1.3.7 Sistemas de Gestión de Base de Datos .....	14
1.3.8 PDO .....	15
1.3.9 Servidores WEB .....	16
1.3.10 Framework.....	16



---

1.3.11	Arquitectura de Software .....	17
1.3.12	Metodología para el proceso de desarrollo de software .....	20
1.4	Principales estándares y tecnologías para el desarrollo 3D sobre la WEB .....	21
1.4.1	Tecnologías de visualización 3D en la WEB .....	22
1.4.2	Estándares para la visualización 3D en la WEB.....	25
1.5	Herramientas de desarrollo.....	29
1.5.1	Rational Rose .....	29
1.5.3	Vivaty Studio.....	30
1.5.4	X3D-Edit .....	30
1.6	Selección de las tecnologías, lenguajes y herramientas a utilizar .....	30
	Consideraciones finales del capítulo .....	33
<b>Capítulo 2:</b>	<b>Características del Sistema .....</b>	<b>34</b>
2.1.	Modelo de Dominio .....	34
2.2.	Captura de Requisitos .....	35
2.2.1.	Requisitos funcionales.....	35
2.2.2.	Requisitos no funcionales.....	37
2.3.1.	Descripción de los actores del sistema .....	40
2.3.2.	Descripción de los Casos de Usos del Sistema .....	42
	Consideraciones finales del capítulo .....	44
<b>Capítulo 3:</b>	<b>Análisis y Diseño del Sistema .....</b>	<b>45</b>
3.1.	Análisis.....	45
3.2.	Diseño.....	45
3.2.2.	Patrones de diseño.....	48
3.2.3.	Patrones de Diseño de Arquitecturas de Aplicaciones Empresariales .....	49
3.2.4.	Diagrama de clases del diseño.....	50
3.2.5.	Diagramas de secuencia .....	50
3.2.6.	Diseño de la Base de Datos .....	51
	Consideraciones finales del capítulo .....	54

---

<b>Capítulo 4: Implementación</b> .....	55
4.1. Modelo de Implementación .....	55
4.1.1. Diagrama de Componentes .....	55
4.1.2. Modelo de Despliegue .....	58
4.1.3. Descripción del método “Visualización de modelos 3D” .....	59
Consideraciones finales del capítulo .....	60
<b>Conclusiones</b> .....	61
<b>Recomendaciones</b> .....	62
<b>Bibliografía</b> .....	63
<b>Anexos</b> .....	66
<b>Glosario de Términos</b> .....	126

## INTRODUCCIÓN

La aparición de Internet como medio de comunicación ha supuesto que el acceso a la información sea sencillo y rápido. La mayor parte de esta información reside en las conocidas páginas WEB que suelen presentar textos e imágenes en dos dimensiones. El mundo real es tridimensional por lo que al reducir el "entorno WEB" a sólo dos dimensiones se está perdiendo información, de ahí la conveniencia de la integración de una tercera dimensión que permita por ejemplo: recorrer las instalaciones de un museo o navegar a través de la anatomía del cuerpo humano.

Con el paso de los años, estudiantes y profesores de todos los niveles han afrontado un problema común: algunas de las áreas de la educación son difíciles de asimilar y de enseñar. En busca de solución a este problema se ha incrementado el interés en una importante rama de la computación, la realidad virtual. Esta nueva tendencia tiene importantes aplicaciones en la educación pues hay indicios de que estimula de manera considerable el proceso de aprendizaje a través del llamado efecto de "inmersión" que genera el ordenador. Gracias a él, los estudiantes pueden interactuar completamente con un ambiente artificial utilizando los sentidos del oído, la vista y el tacto. Una característica importante de los escenarios virtuales es que brinda a sus usuarios la posibilidad de interactuar como participantes activos. La realidad virtual tiene un potencial enorme en la esfera educativa como una gran herramienta motivacional.

En estos tiempos se puede observar el vuelco que ha dado el sistema de educación de las ciencias médicas en cuanto a la utilización de las tecnologías de la información. Cuba no se queda atrás porque se han mejorado las condiciones y los medios tradicionales de enseñanza de muchos centros educacionales de ciencias gracias a la informatización y establecimiento de equipos informáticos que son capaces de ofrecer al docente diversas herramientas que contribuyen a su formación profesional, como es el caso de las multimedias, enciclopedias y bibliografía digital. Todo esto ha posibilitado el aprendizaje de forma más factible y amena de materias como la anatomía, la cual tradicionalmente se estudia con el empleo de cadáveres y libros de texto. No obstante, estas no son las únicas herramientas que podrían contribuir a la asimilación de dichos conocimientos, ya que el empleo de modelos tridimensionales digitales de fácil acceso constituiría un recurso eficaz en esta esfera del aprendizaje. Sin embargo no se está haciendo un uso pleno de estas tecnologías en dichos centros. Siendo, entonces, de gran utilidad el desarrollo de un medio capaz de facilitar el acceso a modelos anatómicos tridimensionales, con los

cuales el estudiante pueda interactuar visualizándolo desde distintos puntos de vista, teniendo en cuenta las bajas prestaciones de las máquinas existentes en dichos centros, que imposibilitan la instalación de software de mayor capacidad de procesamiento y mayor requerimiento de hardware.

Es la situación problemática antes expuesta la que genera el siguiente **problema científico**:

- ❖ ¿Cómo facilitar el acceso, visualización e interacción con modelos tridimensionales que faciliten el aprendizaje de la anatomía del cuerpo humano?

A partir de este problema se define como **objeto de investigación** del trabajo:

- ❖ Visualización 3D en la WEB.

Dentro de esta gama de información se propone como **campo de acción**:

- ❖ Visualización tridimensional de modelos anatómicos sobre la WEB.

Adoptándose como **objetivo general**:

- ❖ Desarrollar una aplicación WEB para la visualización de imágenes y modelos 3D anatómicos.

La investigación tiene como **idea a defender**:

- ❖ Se logrará un mayor acceso y disponibilidad a imágenes y modelos 3D anatómicos que facilitarán el aprendizaje de la anatomía del cuerpo humano, mediante la implementación de un sistema distribuido sobre plataforma WEB.

Para alcanzar nuestro objetivo se trazan las siguientes **tareas investigativas**:

- ❖ Estudio de las tendencias actuales del desarrollo 3D en la WEB para seleccionar la más adecuada para nuestra aplicación.
- ❖ Análisis de los principales sitios de visualización médica existentes para poder definir el diseño y las funcionalidades principales que debe brindar nuestra aplicación WEB.
- ❖ Selección de las tecnologías WEB más adecuadas para la implementación de la aplicación.
- ❖ Estudio del estándar extensible X3D para la confección de los modelos médicos 3D.
- ❖ Selección de la metodología y lenguaje de modelado para el correcto desarrollo de la aplicación WEB.
- ❖ Determinar las herramientas a utilizar para el desarrollo del sistema.
- ❖ Diseño e implementación de la aplicación WEB.

**Métodos teóricos:**

## ❖ Analítico – sintético

La utilización de este método teórico nos posibilitará el análisis de los elementos más importantes que se relacionan con el objeto de estudio en este caso la visualización de modelos e imágenes médicas en la WEB. Nos dará la facilidad de obtener y procesar la información de lo más importante y significativo relacionado con la utilización de estas técnicas de visualización 3D en plataforma WEB.

## ❖ Histórico - Lógico

Mediante la utilización de este método podemos estudiar la evolución que ha tenido hasta actualidad todo lo referente a la visualización tridimensional sobre plataforma WEB.

## ❖ Modelación

Mediante este método podremos representar toda la información obtenida hasta el momento mediante diagramas, ya sea de clases o de diseño, los cuales nos permitirán reflejar las relaciones y cualidades de la aplicación WEB que queremos desarrollar.

**Métodos empíricos:**

## ❖ Observación

Este método será aplicado en distintos momentos del proceso de investigación con el objetivo de evaluar el impacto de las distintas técnicas de visualización 3D en la WEB en el rendimiento de la computadora, con lo cual se podrá seleccionar en cada momento la más adecuada para emplear en la aplicación.

## ❖ Entrevista

Es una conversación planificada para obtener información, mediante este método se consultará a diferentes profesores del Polo de Realidad Virtual de la facultad 5 los cuales han trabajado en el tema.

## **CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA**

En el presente capítulo se describen las tendencias existentes para el desarrollo de aplicaciones WEB con entornos virtuales y las ventajas que brinda en campos como la educación y la medicina. Además se expone un estudio sobre las características, técnicas y tecnologías a utilizar para el desarrollo del trabajo propuesto. También se hace un análisis de la metodología a utilizar, el servidor WEB, la base de datos apropiada y los lenguajes de programación que se van a emplear para el desarrollo del sistema y darle así solución al problema científico de esta investigación.

### **1.1 La realidad virtual**

Una sencilla definición de Realidad Virtual (RV) puede ser “*simulación generada por un computador en un ambiente de tres dimensiones, en donde el usuario puede visualizar y manipular el contenido de dicho ambiente*” [1]. Podemos decir que el término “realidad virtual”, o VR (por sus siglas en inglés Virtual Reality), es muy común en nuestro entorno y en nuestro tiempo. Es decir, cuando hablamos de realidad virtual estamos hablando de algo que no es real pero simula una apariencia de realidad que permite al usuario tener la sensación de estar inmerso en ella, utilizando para ello dispositivos, que engañan nuestros sentidos.

La RV brinda una amplia gama de funcionalidades en diferentes ramas del desarrollo humano. De forma general podemos definir que la RV puede ser de dos tipos: inmersiva y no inmersiva. Los métodos inmersivos de RV con frecuencia se ligan a un ambiente tridimensional creado por computadora el cual se manipula a través de cascos, guantes u otros dispositivos que capturan la posición y rotación de diferentes partes del cuerpo humano. La RV no inmersiva utiliza medios como el que actualmente nos ofrece Internet en el cual podemos interactuar en tiempo real con diferentes personas u objetos en espacios y ambientes que en realidad no existen sin la necesidad de dispositivos adicionales a la computadora.

#### **1.1.1 La realidad virtual y la educación**

La RV ha develado un nuevo camino en muchos aspectos que nos conciernen hoy en día. Por ejemplo los métodos tradicionales de educación que han sido invariables en varias décadas basados en el uso de libros de textos, con el amplio y continuo desarrollo de la humanidad se han cuestionados y analizado

debido a que la sociedad está informatizada y la tecnología de la realidad virtual es ampliamente señalada como un desarrollo tecnológico importante que puede apoyar al proceso de enseñanza.

La RV estimula de manera considerable el proceso de aprendizaje a través del llamado efecto de “inmersión” que produce la computadora. Además que la misma captura la completa atención y reacción de los estudiantes incrementando la curva de aprendizaje [2]. Las actividades educativas que pueden ser soportadas por estas capacidades conducen a la opinión actual de que los estudiantes podrán alcanzar un mejor dominio, retención y generalización de los nuevos conocimientos. Gracias al alto grado de interactividad que ofrece el uso de gráficos tridimensionales la RV puede considerarse como un potente recurso didáctico del cual los profesores pueden valerse para lograr un alto grado de motivación y una mejor atención de los estudiantes.

Dentro del universo de la informática de consumo, la RV puede ser considerada como una forma perfeccionada de multimedia y como tal, combinada con otros tipos de herramientas pedagógicas, ofrece algunas ventajas objetivas en el proceso de enseñanza. Una de sus originales capacidades para mejorar el aprendizaje de los escolares es la posibilidad de permitir la visualización de conceptos abstractos, observar eventos a escalas atómicas o planetarias, o visitar ambientes e interactuar con eventos que la distancia, el tiempo o los factores de seguridad los hacen completamente inalcanzables en condiciones reales[3].

Beneficios de la realidad virtual en la educación [4]:

- La realidad virtual facilita la manipulación y análisis de modelos complejos y grandes, ya que se pueden analizar desde cualquier ángulo y punto de vista.
- La realidad virtual puede mejorar la comprensión de información compleja y abstracta, ya que realiza reificación (mostrar de manera concreta conceptos abstractos).
- Los ambientes virtuales en línea (colaborativos) conforman un espacio de trabajo adecuado para que interactúen social y pedagógicamente alumnos, maestros e investigadores.
- Con la realidad virtual, los estudiantes pueden realizar actividades y aprender ciertos conceptos científicos más fácil y rápidamente.

Podemos apreciar que la combinación de la RV a las redes de telecomunicaciones ofrece la posibilidad de crear espacios y entornos para aprender fuera de instalaciones educativas, constituyendo una

renovación de los tradicionales métodos de educación a distancia como el correo, la radio y la televisión [5].

### **1.1.2 La realidad virtual en la medicina**

La medicina es una ciencia muy amplia a la que se le dedican incontables horas de estudio; por su parte la RV es una vía para darle solución a diversas limitantes. La visualización de modelos por ejemplo es un pilar fundamental para la resolución de muchos problemas en los pacientes porque el doctor puede alcanzar ángulos y puntos de vistas que difícilmente se pueden lograr en la realidad y puede lograr así un diagnóstico de una patología más exacta. Muchas son las ventajas que brinda la RV pero una fundamental es en la enseñanza porque brinda de forma dinámica y amena los contenidos que se estudian de forma tradicional en un aula o con un libro. Pero debido a los altos precios de la aplicación de la RV en la enseñanza, esta tecnología no estaba al alcance del personal docente, aunque la aparición de tecnologías para la visualización 3D en la WEB soluciona en cierta medida este inconveniente, haciéndola asequible a cualquier persona que posea simplemente un ordenador y un navegador de Internet. Obviamente, sólo con estos dispositivos se pierde el sentido del tacto al carecer de guantes, pero la sensación de inmersión en un mundo virtual se puede lograr.

De forma general podemos decir que para la confección de un sistema virtual en medicina se utiliza la visualización de geometría ya que permite al usuario recorrer la reconstrucción 3D de las estructuras anatómicas y brinda la posibilidad de usar métodos de estereoscopia para percibir la sensación de profundidad en el modelo y la manipulación de geometría brinda un complemento al sistema de visualización en el cual el usuario puede interactuar con el modelo para simplemente desplazar, rotar, acercar o alejar la estructura [6].

### **1.1.3 Aplicación de la realidad virtual en el estudio de la anatomía**

La RV es una nueva vía para la representación de órganos y modelos anatómicos muy importante porque permite que los usuarios o estudiantes aprecien de una forma directa una aproximación de la realidad. Las descripciones a través de fotos o palabras complicadas se resumen a la visualización del órgano en 3D sin perder ningún tipo de información que antes se exponía, sino se gana en un sentido tangible respecto a la descripción física del mismo. Con la utilización de esta tecnología los conceptos se arraigan más a los alumnos logrando que las clases de anatomía puedan ser más amenas e instructivas.



La mayoría de los cursos de anatomía se apoyan en la utilización de libros y de cuerpos para la disección donde los libros cada vez son más caros y los cadáveres son más difíciles de encontrar y almacenar, en cambio los modelos anatómicos digitales están en disposición de ir facilitando a los futuros médicos la posibilidad de acceso e interacción con estos recursos.

#### **1.1.4 Uso de las TIC y la realidad virtual en la educación médica cubana**

El proceso de enseñanza de medicina en nuestro país ha tenido que asumir fuertes compromisos y metas en la esfera pedagógica a causa de la formación de grandes masas de profesionales de la salud, existiendo difíciles condiciones, limitación y escasez de materiales dentro de los que podemos citar: imágenes de calidad, modelos anatómicos, libros de texto actualizados entre otros. Los recursos informáticos existentes actualmente son utilizados para consulta auxiliar, los cuales almacenan bibliografía complementaria como es el caso de documentos y alguna que otra multimedia, atlas, videos e imágenes.

Hasta ahora nuestro país ha avanzado muy poco en el campo de la WEB y la RV embebida, solo se han llegado a desarrollar sitios de intercambio de información de forma virtual como “La Universidad Virtual de Salud” que es una institución académica que cuenta con la participación activa y creadora de las instituciones académicas, docente-asistenciales e investigativas del “Sistema Nacional de Salud Cubano”, para desarrollar sistemas de programas de educación en red [9]. También se pueden encontrar diferentes sitios o plataformas WEB que no utilizan la RV pero posibilitan el aprendizaje de la anatomía del cuerpo humano a través de textos, atlas e imágenes.

#### **1.2 Trabajos relacionados hasta el momento de anatomía 3D**

Actualmente con el desarrollo de los ambientes distribuidos y las diversas tecnologías de visualización tridimensional sobre plataforma WEB han surgido un nuevo campo para la representación y extensión de materias como es el caso de la anatomía humana, logrando de esta manera un mayor acceso y disponibilidad a conceptos y representaciones de diferentes secciones del complejo cuerpo humano, lográndose con ello proveer un entorno de interacción simultánea el cual puede ser explotado por innumerables personas posibilitándoles incrementar sus conocimientos de una manera amena y sencilla.

### 1.2.1. Visible Body

Es una aplicación WEB en 3D con la que se puede estudiar la anatomía humana interactivamente. Esta aplicación permite ver cada uno de los aparatos, sistemas y órganos del cuerpo humano de manera individual o conjunta y desde cualquier punto de vista. También se pueden hacer transparentes algunos sistemas o simplemente pinchando un botón se ocultan. Además se puede acercar o alejar el modelo y girarlo en cualquier dirección del espacio con unos sencillos controles.

Visible Body permite:

- Buscar y localizar estructuras anatómicas por nombre.
- Esconder, rotar, explorar y ver a través de las partes de la anatomía humana.
- Mover en el espacio el modelo tridimensional.
- Aumentar y disminuir el zoom.
- Hacer clic en los sistemas o estructuras para hacerlos transparentes o esconderlos.
- Hacer clic en las estructuras anatómicas para revelar los nombres.

Algunas de las características de esta aplicación son:

- Requiere Procesador Pentium 4 o superior, 512 MB de memoria RAM.
- Solamente es para plataformas propietarias Windows y Macintosh.
- Exige que se realicen suscripciones para poder usar su contenido mediante el pago de las mismas o sea es privado [7].

El manejo, aunque la WEB está en inglés, es muy intuitivo y la calidad del detalle de los modelos 3D es sorprendente.

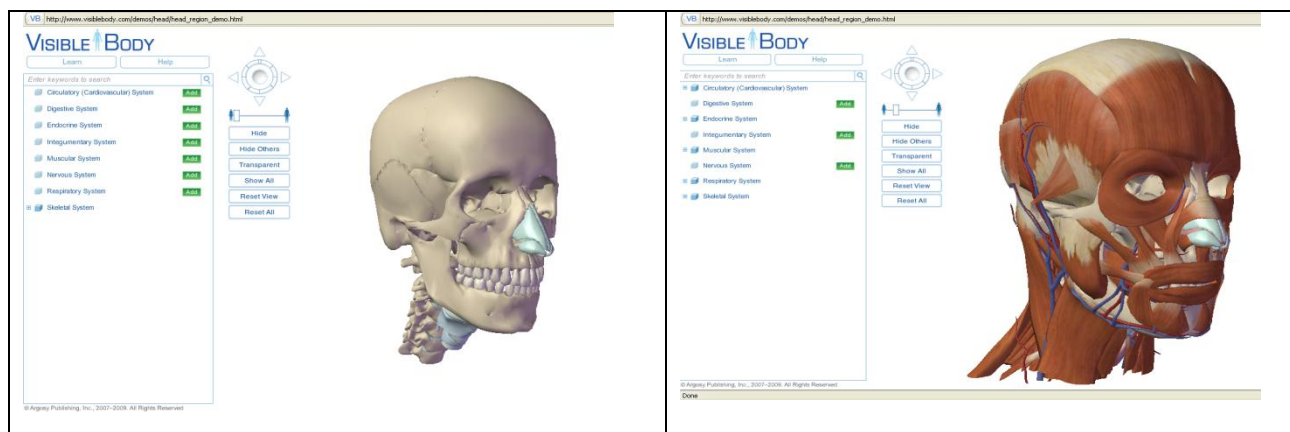


Fig. 1. Visible Body.

### 1.2.2. Visión Médica Virtual

Es una compañía de reciente creación y su objetivo es generar un software, que permita visualizar on-line un Atlas 3D de Anatomía Humana, interactivo y de alto rigor científico-médico. Cualquier Universidad de Medicina o Escuela de Ciencias de la Salud puede acceder a los modelos 3D del Atlas Anatómico, aunque para esto, es necesario ponerse en contacto con 3DVisiomedicavirtual y establecer las condiciones de uso y privacidad de los contenidos de la WEB. Existe la posibilidad de acuerdos de acceso restringidos a grupos y se puede acceder mediante el pago de una cuota anual [8].



Fig. 2. Visión Médica Virtual

## 1.3 Tendencias y tecnologías actuales de desarrollo WEB

### 1.3.1 Definición de aplicación WEB

Una aplicación WEB no es un sistema en tiempo real, por lo cual el usuario es el encargado de realizar el almacenamiento de datos y la intervención del mismo es primordial para la actualización y el perfecto funcionamiento de la misma. Es decir, es una aplicación de software que se codifica en un lenguaje soportado por los navegadores WEB en la que se confía la ejecución al navegador que está diseñado para automatizar procesos cuyo resultado se entrega a través de Internet o una Intranet [10]. Las

aplicaciones WEB son populares debido a la facilidad para actualizar y mantenerlas sin tener que distribuir e instalar software a miles de usuarios.

En la actualidad y debido al auge de este tipo de aplicaciones, se reconocen diferentes tipos de categorías en dependencia del tipo de negocio que modelen:

- Informacionales: periódicos, catálogos, manuales, libros electrónicos, etc.
- Interactivas: Formularios de registros, presentación de información personalizada, etc.
- Transaccionales: tienda electrónica, bancos online, etc.
- Workflow: planificación online, monitoreo, gerencia de inventario, etc.
- Comunitarias: Chat, mercados, subastas online, etc.
- Portales: tiendas electrónicas, etc.

### 1.3.2 Ventajas de las aplicaciones WEB

Los medios distribuidos han tenido un alto auge en los últimos años colocándose en primer lugar ante las pesadas aplicaciones de escritorio que requieren muchas veces de instalaciones previas y dispositivos auxiliares para la perfecta visualización de la misma. Las páginas Web brindan la posibilidad del libre acceso de navegación y de interacción, sin tener que instalar ni copiar nada a tu máquina [11].

Ventajas:

- No necesita ninguna configuración especial o cambios necesarios en la computadora del usuario.
- Posee costos más bajos que cualquier aplicación de escritorio.
- Los datos se encuentran centralizados garantizando la integridad y la copia de seguridad.
- Las actualizaciones pueden hacerse de forma rápida y fácilmente.
- Permite que la información sea accesible a un amplio público desde cualquier lugar del mundo.
- Posee una disponibilidad permanente.

### 1.3.3 Estándares de programación WEB

#### ❖ HTML

HTML es la sigla de **HiperText Markup Language** (Lenguaje de Marcas de Hipertexto) es el conjunto de símbolos de marcado o códigos insertados en un archivo destinado a mostrar en una página del World Wide WEB. Se utiliza comúnmente para establecer la estructura y contenido de un sitio WEB, tanto de texto, objetos e imágenes. Los archivos desarrollados en HTML usan la extensión (.htm o .html) [12]. El

lenguaje de HTML funciona por medio de “etiquetas” que describen la apariencia o función del texto enmarcado. Este lenguaje puede llegar a incluir un script o código que tenga incidencia en el comportamiento del navegador WEB de elección. El HTML Dinámico o **DHTML** (Dinámico HTML) es el grupo de técnicas que permite construir sitios WEB interactivos combinado el HTML estático, JavaScript, CCS y DOM.

#### ❖ XML

XML es la sigla de **Extensible Markup Language** (Lenguaje de Marcas extensible) es un perfil de aplicación o forma restringida de (SGML), el lenguaje de marcas estándar generalizado Language [ISO 8879]. Es un metalenguaje extensible de etiquetas, es decir, lenguajes para definir lenguajes de estructuración de texto basados en marcas explícitas o tags, que son entendibles por una persona y que pueden ser interpretadas por un computador.

Los documentos XML se componen de unidades de almacenamiento llamadas entidades, que contienen datos. Este marcado codifica una descripción del diseño de almacenamiento del documento y la estructura lógica. XML proporciona un mecanismo para imponer restricciones sobre el diseño de almacenamiento y estructura lógica [13]. Además soporta una amplia variedad de aplicaciones que se pueden generar de una forma rápida y su mantenimiento es mucho más sencillo. XML facilita la integración desde fuentes de datos heterogéneas, por ejemplo: páginas WEB, bases de datos y propone un estándar para el intercambio de información estructurada entre diferentes plataformas.

#### ❖ CSS

CSS es la sigla de **Cascading Style Sheets** (Hojas de Estilo en Cascada), es la forma recomendada para controlar la presentación de la capa en un documento WEB. La principal ventaja de CSS en HTML es la presentación del marcado del estilo puede ser totalmente independiente del contenido. Mediante la separación de la capa de presentación, CSS permite que todos los estilos se mantengan en un número limitado de hojas permitiendo guardar el estilo de 10000 archivos HTML aproximadamente y de esta manera el ahorro global en el ancho de banda es medible. Con el uso del CCS se logra un control centralizado de la presentación de un sitio WEB completo, agilizando en el mismo la actualización de sus páginas [14].

### 1.3.4 Lenguajes de programación WEB

#### ❖ JavaScript

Es un lenguaje interpretado que está del lado del cliente porque es el navegador el que soporta la carga de procesamiento. Se utiliza integrado en un navegador WEB permitiendo el desarrollo de interfaces de usuario mejoradas y páginas WEB dinámicas. Este lenguaje es utilizado para acceder a objetos en las aplicaciones y gracias a su compatibilidad con la mayoría de los navegadores modernos es el más utilizado por parte de los clientes. JavaScript posee una sintaxis similar a la del lenguaje Java y el lenguaje C, aunque no es orientado a objeto está destinado al desarrollo de aplicaciones WEB como complemento del HTML. También puede realizar efectos especiales sobre páginas WEB para crear contenidos dinámicos y elementos de la página que tengan movimiento. Además permite que se ejecuten instrucciones como respuesta a las acciones del usuario, con lo que se crean páginas interactivas [15].

Ventajas:

- Los programas JavaScript tienden a ser pequeños y compactos, no requieren mucha memoria ni tiempo adicional de transmisión.
- Funciona en cualquier navegador que soporte JavaScript independientemente de la plataforma o el sistema operativo.
- Es un lenguaje de scripting seguro y fiable.
- El código JavaScript es gestionado en el navegador.

Desventajas:

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

#### ❖ PHP

PHP es la sigla de **Hypertext Preprocessor** (Procesador de Hipertextos) es generalmente muy usado por su flexibilidad. Es un lenguaje de código abierto de scripting (escritura) especialmente adecuado para el desarrollo de WEB.

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas WEB dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser

compilado para ejecutarse. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Además permite la conexión a diferentes tipos de servidores de base de datos tales como MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird, SQLite logrando aplicaciones WEB robustas y sus archivos cuentan con la extensión (.php) [16].

Ventajas:

- Es identificado como un lenguaje muy rápido y manuable.
- Soporta de cierta forma la orientación a objeto. Clases y herencia.
- Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- Capacidad de conexión con la mayoría de los manejadores de base de datos.
- Puede expandir su capacidad de potencial utilizando módulos.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos y su código está disponible bajo la licencia GPL.
- Posee una biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

Desventajas:

- Se necesita instalar un servidor WEB para la ejecución del código.
- Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto, puede ser más ineficiente a medida que las solicitudes aumenten de número.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.

### 1.3.5 Técnicas de Programación WEB

#### ❖ Ajax

Ajax es el acrónimo de **Asynchronous JavaScript And XML** (JavaScript y XML asíncronos). Ajax no es una tecnología porque es la unión de un conjunto de tecnologías para el desarrollo WEB que se ejecutan en el cliente, es decir, en el navegador de los usuarios y mantiene comunicación asíncrona con el servidor en segundo plano.

Ajax tiene una presentación basada en estándares usando XHTML y CSS. Además realiza una interacción dinámica usando el DOM y realiza el intercambio y manipulación de datos usando XML. Esta tecnología posee recuperación de datos asíncrona usando XMLHttpRequest. Los datos adicionales que carga del servidor no sobrecargan la página ni afecta su comportamiento porque no interfiere con la

visualización de la misma. También Ajax se ha extendido a la manipulación de entornos 3D en la WEB con Ajax3D [17].

### 1.3.6 Tecnologías de programación WEB

#### ❖ DOM

DOM es la sigla de **Document Object Model** (Modelo de Objetos del Documento) es una representación del modelo de un documento y porque permite verlo de una manera diferente describiendo el contenido del mismo en un conjunto de objetos que el lenguaje JavaScript puede actuar sobre ellos. Acorde al W3C, DOM es una interfaz de programación de aplicaciones (API) para documentos HTML ya define la estructura lógica de los mismos y el modo en que se accede y manipula. DOM permite un acceso a la estructura de una página HTML mediante el mapeo de los elementos de esta en un árbol de nodos. Cada elemento se convierte en un nodo y cada porción de texto en un nodo de texto [18].

### 1.3.7 Sistemas de Gestión de Base de Datos

Un sistema de gestión de bases de datos con siglas en inglés (DBMS) es un paquete de software diseñado para almacenar y administrar base de datos. Estos permiten el acceso directo a la información y son capaces de manipularla ya que funcionan como una interfaz entre el usuario, la base de datos y las aplicaciones que la utilizan. Estos brindan un independiente y eficiente acceso a los datos reduciendo el tiempo de ejecución e integridad y seguridad de los mismos [19].

Además, podemos definir como una base de datos la colección ordenada de datos, que normalmente se almacena en uno o más archivos asociados. Estos están estructurados en forma de tablas, donde las referencias cruzadas entre las tablas son posibles. Algunos ejemplos de sistemas de bases de datos relacionales: MySQL, PostgreSQL, Post Oracle, el servidor de Microsoft SQL y DB2 de IBM.

#### ❖ MySQL

MySQL es un sistema de base de datos relacional, multihilo, multiusuario que posee una arquitectura cliente servidor. Soporta como lenguaje de base de datos como su nombre indica SQL (Structured Query Language). SQL es un lenguaje normalizado para consultar y actualizar los datos. MySQL brinda una búsqueda de texto completo siendo muy eficaz para la pesquisa de la información y además brinda la replicación del contenido de la base de datos permitiendo aumentar la protección contra el fallo del sistema. También posee un sistema de privilegios y contraseñas que es muy flexible y seguro que



permite la verificación basada en el host. En los últimos años este sistema con más de 6 millones de instalaciones se convirtió en propietario y hoy en día está patrocinado por una empresa privada que posee el copyright de la mayor parte del código [20].

#### ❖ PostgreSQL

PostgreSQL es el más avanzado sistema de base de datos relacionales de código abierto con modelo de arquitectura de comunicación cliente servidor [21]. Cuenta con más de 15 años de desarrollo activo y funciona en todos los sistemas operativos existentes hoy en día. También posee una arquitectura que se ha ganado una sólida reputación de confiabilidad e integridad de datos. Tiene interfaces de programación nativa de C / C + +, Java, NET, Perl, Python, Ruby, Tcl, ODBC, entre otros, contando con una documentación de carácter excepcional. PostgreSQL cuenta con sofisticadas funciones como la versión multicontrol de concurrencia y posee punto de tiempo de recuperación, replicación asincrónica, transacciones anidadas, un planificador de consultas sofisticadas y posee un registro para la tolerancia a fallos. Soporta un tamaño ilimitado de almacenamiento de datos y tiene soporte completo para subconsultas.

Posee un avanzado sistema que reúne una amplia gama de diferentes algoritmos de clasificación y búsqueda, incluido el B-tree, árbol B +, R-árbol y muchos otros. También proporciona una interfaz que permite tanto la creación de tipos de datos personalizados, así como métodos de consulta extensible [22]. La utilización de PostgreSQL puede traer muchas ventajas por ejemplo el costo para su uso es nulo porque su código fuente está disponible bajo la licencia del código abierto (BSD) que brinda la posibilidad de usarlo y modificarlo a conveniencia del usuario convirtiéndolo en una opción muy factible para su utilización.

#### 1.3.8 PDO

Es las siglas de **PHP Data Objects** (PHP Objetos de Datos) es una extensión nativa de PHP5, consiste en una interfaz uniforme para acceder a varias bases de datos. PDO provee una capa de abstracción de acceso a bases de datos (BD), que permite al desarrollador abstraerse de la BD de una aplicación. Así, si en alguna fase de desarrollo del proyecto se necesita cambiar de MySQL a PostgreSQL o viceversa, esto no afectaría la lógica de la aplicación porque hace el código portable a otras plataformas y motores de BD [23].

### 1.3.9 Servidores WEB

Los servidores WEB son programas que envían contenidos tales como: páginas WEB, hipertextos, formularios, imágenes y botones a través del protocolo HTTP. Estos manejan la entrega de los componentes de las páginas WEB como respuesta a peticiones de los navegadores de los clientes pues se ejecutan continuamente en un ordenador y están manteniéndose siempre a la espera de peticiones por parte de un cliente o navegador WEB.

#### ❖ Servidor Apache

Apache es el servidor WEB más completo de código libre siendo altamente configurable con un diseño modular. También funciona muy bien con Perl, PHP y otros lenguajes de scripting. Su robustez lo ha colocado como el servidor más popular en Internet desde abril de 1996 y en una encuesta en noviembre del 2005 Netcraft encontró que más del 70% de los sitios WEB en Internet lo están utilizando [24].

Apache tiene la característica de ser multiplataforma incluyendo módulos que se cargan de forma dinámica. Además tiene soporte para base de datos y para SSL, logrando que las transacciones sean seguras y confiables.

### 1.3.10 Framework

Un framework WEB es un conjunto de componentes que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas WEB. Son diseñados con la intención de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel de proveer un sistema funcional. Un framework es un producto que sirve como base para la programación avanzada de aplicaciones, que aporta una serie de funciones o códigos para realizar tareas habituales. Son unas librerías de código que contienen procesos o rutinas ya listas para usar. Los programadores lo utilizan para no tener que desarrollar ellos mismos las tareas más básicas, puesto que en el propio framework ya hay implementaciones que están probadas, funcionan y son reutilizables [25].

Ventajas:

- El programador no necesita plantearse una estructura global de la aplicación, sino que el framework le proporciona un esqueleto que hay que "rellenar".

- Facilita la colaboración. Todos sabemos lo difícil que resulta el entendimiento del código de otra persona por lo que todo lo que sea definir y estandarizar va a ahorrar tiempo y trabajo a los desarrollos colaborativos.
- Es más fácil encontrar herramientas (utilidades, librerías) adaptadas al framework concreto para facilitar el desarrollo.

#### ❖ **JQuery framework**

JQuery es un framework para el lenguaje JavaScript. Este implementa una serie de clases (de programación orientada a objetos) que nos permiten programar sin preocuparnos del navegador con el que nos está visitando el usuario pues funciona de forma exacta en todas las plataformas más habituales [26].

Ventajas:

- Ofrece una infraestructura que da una mayor facilidad en la creación de aplicaciones complejas del lado del cliente.
- Obtendremos ayuda en la creación de interfaces de usuario, efectos dinámicos y aplicaciones que hacen uso de Ajax.
- Su obtención es de manera gratuita, ya que el framework tiene licencia para uso en cualquier tipo de plataforma.
- El archivo del framework ocupa unos 56 KB, lo que es bastante razonable y no retrasará mucho la carga de nuestra página.
- Producto serio, estable, bien documentado y con un gran equipo de desarrolladores a cargo de la mejora y actualización del framework.

#### **1.3.11 Arquitectura de Software**

La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos, el ambiente y los principios que orientan su diseño y evolución. Una arquitectura bien definida y robusta valida que la aplicación tenga las mejores condiciones ya que es la columna vertebral de la aplicación [27].

#### ❖ **Arquitectura Cliente-Servidor**

En la actualidad para el desarrollo de sistemas distribuidos como es el caso de las aplicaciones WEB, la arquitectura empleada es la Cliente-Servidor que define dos tipos de entidades diferenciadas que se

ocupan de acciones diferentes: clientes y servidores. Esta establece un modelo de comunicación o de interacción sustentado en el concepto de servicio, materializado a través de un diálogo de petición-respuesta. El cliente inicia el diálogo a través del envío de peticiones al servidor, el cual presta el servicio y responde a las peticiones recibidas. Además se especifica el modo en que se sincronizan los procesos donde el cliente es la parte activa del cual demanda servicios a los servidores y este asume que las peticiones deberán tener respuesta ya que está diseñado para soportar la interacción con el usuario final, en cambio, el servidor constituye la parte pasiva el cual espera las peticiones de los clientes y procesa las mismas para enviar las respuestas ya que su diseño está orientado a maximizar la eficiencia. La ubicación del cliente es independiente a la ubicación del servidor y generalmente los clientes se encuentran en ordenadores de bajas prestaciones u ordenadores personales y los servidores en máquinas de mayor poder de procesamiento. Para la comunicación de los procesos en la red se utiliza componente de sistema denominado middleware que da unidad y abstrae las peculiaridades de las plataformas (hardware y SO) de clientes y servidor, este a su vez da soporte al envío/recepción de mensajes [27].

#### Ventajas

- Localización transparente.
- Recursos compartidos.
- Escalabilidad.
  - ✓ Horizontal: > n<sup>o</sup> estaciones.
  - ✓ Vertical: migración a otras plataformas
- Interoperabilidad entre distintos hardware y software.

#### ❖ Patrón Arquitectura en Capas

Dentro de los patrones de arquitecturas para aplicaciones distribuidas tenemos:

- Arquitecturas de dos niveles.
- Arquitecturas de tres niveles.
- Arquitecturas de n niveles.

La arquitectura de tres niveles se divide en tres capas distintas:

- Capa de presentación (interfaz de usuario) interacciona con el usuario, presenta los datos y recibe las entradas.

- Capa de aplicación/negocio (lógica de aplicación) responsable de las tareas propias de la aplicación concreta implementa la lógica de la aplicación y aplica las reglas de negocio sobre los datos y las entradas de usuario.
- Capa de datos (almacenamiento y acceso a datos) responsable de la gestión y almacenamiento permanente de los datos.

Cada tipo de sistema cliente-servidor distribuye esas capas de modo distinto entre los componentes cliente y servidor.

### ❖ Patrón Modelo Vista Controlador

Patrón de diseño de arquitectura de software muy empleado actualmente. Este sugiere la separación del software en tres estratos (Modelo, Vista y Controlador) [28]:

**Modelo:** Es la representación de la información que maneja la aplicación. El modelo en sí son los datos puros que puestos en contexto del sistema proveen de información al usuario o a la aplicación misma. (Información almacenada en una base de datos o en XML junto con las reglas de negocio que transforman esta información).

**Vista:** Es la representación del modelo en forma gráfica disponible para la interacción con el usuario. En el caso de una aplicación WEB, la “Vista” es una página HTML con contenido dinámico sobre el cual el usuario puede realizar operaciones. (La página HTML).

**Controlador:** Es la capa encargada de manejar y responder las solicitudes del usuario, procesando la información necesaria y modificando el Modelo en caso de ser necesario. (Código que obtiene datos dinámicamente y genera el contenido HTML).

#### Ventajas

- La separación del Modelo de la Vista, es decir, separar los datos de la representación visual de los mismos.
- Facilita agregar nuevos tipos de datos según sea requerido por la aplicación ya que son independientes del funcionamiento de las otras capas.
- Crea independencia de funcionamiento.
- Facilita el mantenimiento en caso de errores.
- Permite el escalamiento de la aplicación en caso de ser requerido.

### 1.3.12 Metodología para el proceso de desarrollo de software

Se puede decir que un proceso de desarrollo de software es el conjunto de actividades destinadas a guiar los esfuerzos de las personas implicadas en un proyecto de desarrollo, a modo de plantilla explica los pasos necesarios para terminar el proyecto.

#### ❖ RUP

Es las siglas de **Rational Unified Process** (Proceso Unificado Rational) cuyo fin es entregar un producto de software. Rup es la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Tiene una forma disciplinada de asignar tareas y responsabilidades, por lo que se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y por ser guiado por los casos de uso [29].

Ventajas:

- Desarrollo iterativo del software.
- Administración de requerimientos.
- Uso de arquitecturas basadas en componentes.
- Modelamiento visual del software.
- Verificación de la calidad del software.
- Control de cambios.

RUP es un proceso de desarrollo de software el cual utiliza el lenguaje unificado de modelado (UML) para especificar, visualizar y documentar los diferentes aspectos relativos a un sistema de software bajo desarrollo.

Unified (unificado):

- El aporte de muchos métodos y notaciones
- Independiente de implementaciones, plataformas y lenguajes.

Modeling (modelado):

- Los modelos son utilizados en todas las ingenierías.

Language (lenguaje):

- Si hay gente, requieren comunicarse. Si se tienen que comunicar, se tienen que entender y para entenderse necesitan un lenguaje común.

Características

- Lenguaje de representación visual

- Divide cada proyecto en un número de diagramas que representan las distintas vistas del proyecto y juntos representan la arquitectura del mismo.
- Permite describir un sistema en diferentes niveles de abstracción.

Ventajas:

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

#### **1.4 Principales estándares y tecnologías para el desarrollo 3D sobre la WEB**

Las tecnologías de visualización tridimensional (3D) en la WEB han demostrado ser una gran herramienta para la enseñanza de la Medicina. La visualización en 3D permite a los estudiantes apreciar espacialmente representaciones tan precisas o didácticas como sean necesarias de objetos tangibles o intangibles de escalas tanto macro como microscópicas de naturaleza estable o volátiles [30].

Ventajas:

- Son rápidas de crear.
- Requieren plug-ins de pequeño tamaño o no lo requieren (Si están basados en Java).
- Compatibles con varias plataformas: Mac, Windows, Linux.
- Aspecto espectacular
- Posibilidad de añadir efectos de render: reflexiones y sombras.
- Fáciles de integrar en páginas WEB y en otras aplicaciones.
- Tiempos de descarga reducidos.
- Basados en estándares.
- Existen compañías importantes que los apoyan y los desarrollan: Adobe, Metastream.

Estas tecnologías pueden agruparse en:

- Tecnologías basadas en XML: Utilizan dos ficheros, uno contiene la geometría y otro las transformaciones y las interacciones posibles del usuario con el mundo virtual. Estas requieren un plug-ins para su visualización.
- Tecnologías basadas en Java3D. No es necesario plug-ins aunque son más lentas.

- Tecnologías propias desarrolladas por las firmas comerciales. Requieren plug-ins.

### 1.4.1 Tecnologías de visualización 3D en la WEB

#### ❖ Java3D

Java3D es un API (Application Programming Interface) orientado a objetos para el lenguaje Java, que permite la confección de aplicaciones tridimensionales las cuales posibilitan:

- Construir objetos 3D.
- Visualizarlos.
- Controlar su comportamiento.

Es una interfaz que encapsula programación de gráficos usando un verdadero concepto orientado a objetos. Aquí, un escenario se construye utilizando un grafo de escena que es una representación de los objetos que tienen que ser mostrados. En este grafo se describen la geometría de los objetos tridimensionales y sus propiedades (colores, texturas, movimientos, etc.), su ubicación dentro de la escena, orientación, fuentes de luz, lugar en donde está situado el observador, etc. [31]

Ventajas:

- Al ser una extensión de la API de Java, es independiente de la plataforma.
- Es un API de código abierto.
- Interfaz de alto nivel.
- La visualización se basa en las API:
  - ✓ OpenGL
  - ✓ DirectX
- La escena se construye creando un grafo.
- La aplicación 3D puede ser un applet.

Desventajas:

- Una máquina virtual de Java es pesada en comparación con el uso de aplicaciones compiladas nativamente, lo que hace menos deseables en ambientes con altos requerimientos en las prioridades del procesador y la asignación de recursos, tales como gráficos 3D y la animación.
- El API oculta detalles de cómo se visualiza la escena.
- Los componentes de Java3D son pesados.
- No es tan rápido como una aplicación en código nativo en OpenGL o DirectX.



### ❖ **WEBGL**

WEBGL es una de las más recientes tecnologías de visualización 3D sobre plataforma WEB, está diseñada para presentar gráficos en tres dimensiones en las páginas WEB que son acelerados por hardware sin la necesidad de plug-ins para el navegador. El sistema WEBGL basa su tecnología combinando JavaScript, OpenGL y la etiqueta canvas, incorporada a HTML5. WEBGL busca dotar de gráficos 3D a los navegadores sin necesidad de extensiones. Además, debido a que está plenamente integrado en el navegador, una aplicación WEBGL puede aprovechar la infraestructura de JavaScript y DOM, fundamental para cualquier documento HTML [32].

Ventajas:

- Carga de imágenes: una aplicación WEBGL puede utilizar las facilidades de carga de imágenes del navegador directamente.
- Gestión de eventos: una aplicación de WEBGL puede establecer una llamada a función en cualquier evento de JavaScript.
- Composición de contenido WEB: WEBGL utiliza el estándar canvas.
- Gestión de memoria automática.

### ❖ **Unity3D**

Es una potentísima herramienta de creación de entornos 3D, aplicaciones interactivas, aplicaciones WEB y juegos en 3D. Está basado en JavaScript y se debe descargar el plug-ins para poder ver el contenido en cualquier browser [33].

Ventajas:

- No afecta el rendimiento de la máquina independientemente de las características de la misma, o sea, no sobrecarga la computadora.
- Se puede programar en C#, JavaScript, AJAX, .NET y se puede comunicar mediante sockets al flash player por lo que se puede ligar con tecnologías Adobe.
- Para el desarrollo de proyectos se puede usar el editor de Unity, aunque también cualquier otro editor para JavaScript, llámese Dreamweaver, Eclipse, etc, ya que las librerías Unity se integran a un proyecto JavaScript.
- Cuenta con un editor exclusivo para 3D con el cual se puede testear sin necesidad de compilar, siendo de gran ventaja porque permite ver cómo se comporta el entorno.

Desventajas:

- La documentación de las clases no es muy clara y muchas veces hay que perder mucho tiempo experimentando para entender el comportamiento de una clase en particular.
- Se puede bajar la versión Trial para desarrollar, lo demás es bajo licencia.

#### ❖ **Google O3D**

Google O3D es una API de código libre basada en JavaScript para crear mundos 3D, aún en etapa temprana de desarrollo, sin embargo ya dispone de un plug-ins que muestra gráficos 3D. La API de O3D maximiza el rendimiento de la GPU para la programación del lenguaje de sombreado de forma directa, siendo una ventaja sobre el software de puro renderizado.

Google O3D funciona a tres niveles [34]:

- Sistema: proporciona un plug-ins para el navegador que añade capacidades gráficas dentro del motor de renderizado.
- Contenido: aparte de trabajar con texto, imágenes o vídeo, O3D proporciona nuevas herramientas para crear contenido en 3 dimensiones.
- Código: O3D extiende el código disponible para los programadores en JavaScript con un API para gráficos 3D, que usa el sistema de eventos y procedimientos típico de JavaScript.

#### ❖ **RIA**

RIA es la sigla de **Rich Internet Applications** (aplicaciones enriquecidas en internet) y se ejecutan desde un navegador como si fuesen una página WEB sin precisar de una instalación de software cliente en cada máquina en la que vayan a ser utilizadas. Son aplicaciones WEB diseñadas para ofrecer las mismas características y funciones normalmente asociadas con las aplicaciones de desktop. Mediante su uso se puede lograr la visualización tridimensional sobre plataforma WEB. Las posibilidades principales para desarrollar RIA actualmente son: Adobe Flash, Microsoft Silverlight, Ajax, Java y Adobe Director [35].

##### ➤ **Flash 3D**

Dentro de estas tecnologías podemos encontrar **PaperVision3D** que es un motor de renderizado 3D en tiempo real, hecho en ActionScript 3, de código abierto, que posee las funcionalidades básicas de la computación gráfica tridimensional, creando una ilusión 3D en el motor de renderizado que posee el Flash Player [33].

Desventajas:

- No posee soporte de hardware.
- No incorpora sombras.
- Modelos sencillos con pocos polígonos.

➤ **Away3D**

Es un framework de código abierto basado en ActionScript, motor de renderizado 3D para el desarrollo de aplicaciones flash [\[33\]](#).

Desventajas:

- No es fácil importar modelos al proyecto y el comportamiento de estos es errático.
- El manejo de la cámara no es tan certero como PaperVision3D.
- No maneja tantos polígonos como X3D.

➤ **Sandy3D**

Librería open-source desarrollada en ActionScript para la visualización 3D en el entorno de flash [\[36\]](#).

Desventajas:

- De manera general los modelos construidos tienen poca definición debido a que poseen pocos polígonos.

### 1.4.2 Estándares para la visualización 3D en la WEB

#### ❖ VRML

VRML es la sigla de **Virtual Reality Modeling Language** (Lenguaje modelado de realidad virtual) que es un lenguaje de especificaciones, tridimensional e interactivo, orientado a la modelación y la visualización de objetos, situaciones y mundos virtuales en Red. Provee un conjunto básico de primitivas para el modelaje geométrico tridimensional y tiene la capacidad de dar comportamiento a los objetos y asignar diferentes animaciones que pueden ser activadas por eventos generados por diferentes usuarios. VRML es un lenguaje jerárquico de marcas que usa Nodos, Eventos y Campos para modelar realidades virtuales tanto estáticas, como dinámicas [\[31\]](#).

Los modelos construidos con VRML se definen en uno o más archivos VRML, que son archivos de texto con una sintaxis normalizada. Los bloques de construcción de un modelo VRML se llaman nodos VRML, y cada nodo se especifica mediante una sintaxis estándar y describe, por ejemplo, una forma tridimensional, una fuente de luz, el camino para una animación, la posición de una fuente de sonido, y así sucesivamente. Los nodos se organizan en lo que se denomina un grafo de escena, que es una estructura jerárquica de uso general para la construcción y gestión de contenidos complejos de tres dimensiones.

Hay dos versiones de este estándar, VRML 1.0 y VRML 2.0.

VRML 1.0 es un lenguaje para la descripción de mundos virtuales estáticos, que cumple tres requisitos fundamentales: es independiente de la plataforma donde se ejecute el visualizador, tiene capacidad para trabajar de un modo eficiente con conexiones lentas, y es extensible, es decir, susceptible de ser ampliado fácilmente [37].

La versión VRML 2.0 es mucho más compleja que su predecesora, y en ella destacan los siguientes aspectos:

- Posibilidad de especificar comportamientos para los objetos, ya sea usando el propio lenguaje VRML o mediante scripts en lenguajes externos (JavaScript, Java, Visual Basic, etc.), los cuales no están limitados por la especificación.
- Posibilidad de interacción con el usuario mediante la definición de una serie de sensores de posición, de contacto y de colisión. La información registrada por estos sensores es enviada a los diferentes objetos que componen el mundo virtual y en función de los valores recibidos, cada objeto virtual actuará en consecuencia.
- Finalmente, el lenguaje de descripción de escenas tridimensionales ha sido ampliado significativamente, posibilitando efectos de fondo, sonidos tridimensionales y niebla.

Ventajas:

- Facilidad de elaboración de objetos 3D con movimientos y detección de colisiones.
- La sintaxis del lenguaje VRML es relativamente simple y flexible posibilitando que sea fácil de aprender para los programadores. Además de su sencilla integración con otros idiomas, JavaScript y Java contribuye a su flexibilidad y extensibilidad.
- Accesibilidad: el bajo costo y la disponibilidad del software (navegador de VRML o plug-ins) favorecen el interés de la comunidad de Internet en esta tecnología [37].

### ❖ X3D

X3D es la sigla de (Extensible Tridimensional) que es un estándar abierto XML que tiene un formato de archivo 3D que permite la creación y transmisión de datos 3D entre distintas aplicaciones y especialmente aplicaciones en red. Sus principales características son:

Está integrado en XML y esto representa un paso fundamental a la hora de conseguir una correcta integración en [\[31\]](#):

- Servicios WEB.
- Redes Distribuidas.
- Sistemas multiplataforma y transferencia de archivos y datos entre aplicaciones.

Es **Modular** (tiene componentes): esto permite la creación de un núcleo 3D más ligero ajustado a las necesidades de los desarrolladores.

Es **Extensible**: permite añadir componentes para ampliar las funcionalidades según las necesidades del mercado.

Es **Perfilado**: se pueden escoger distintos grupos de extensiones apropiadas según las necesidades específicas de la aplicación.

Es **Compatible** con VRML: se mantiene el desarrollo, el contenido y la base de VRML97.

Para la visualización de escenas X3D se emplean los X3D browsers que son aplicaciones de software que pueden leer o procesar escenas X3D, no solo permiten mostrar a los objetos desde distintos puntos de vista sino que también permiten la animación de los mismos y la interacción con el usuario. Estos se suelen aplicar como plug-ins que funcionan como una parte integrada del lenguaje HTML que interpreta el navegador WEB.

#### Perfiles de X3D:

- **Intercambio**: es el perfil básico para la comunicación entre aplicaciones. Este perfil da soporte a las geometrías, las texturas, las iluminaciones básicas y las animaciones.
- **Interactivo**: es el perfil encargado de aportar las capacidades básicas de interacción con el escenario virtual mediante la incorporación de varios nodos sensoriales. Además aporta más capacidades de iluminación.
- **Inmersivo**: habilita todas las capacidades de gráficos 3D e interacción con el entorno, además incluye el soporte para audio, colisiones, niebla y scripting.

- **Completo:** incluye todos los nodos definidos incluyendo los componentes Nurbs, H-Anim y GeoSpatial.

Arquitectura del estándar X3D [38]:

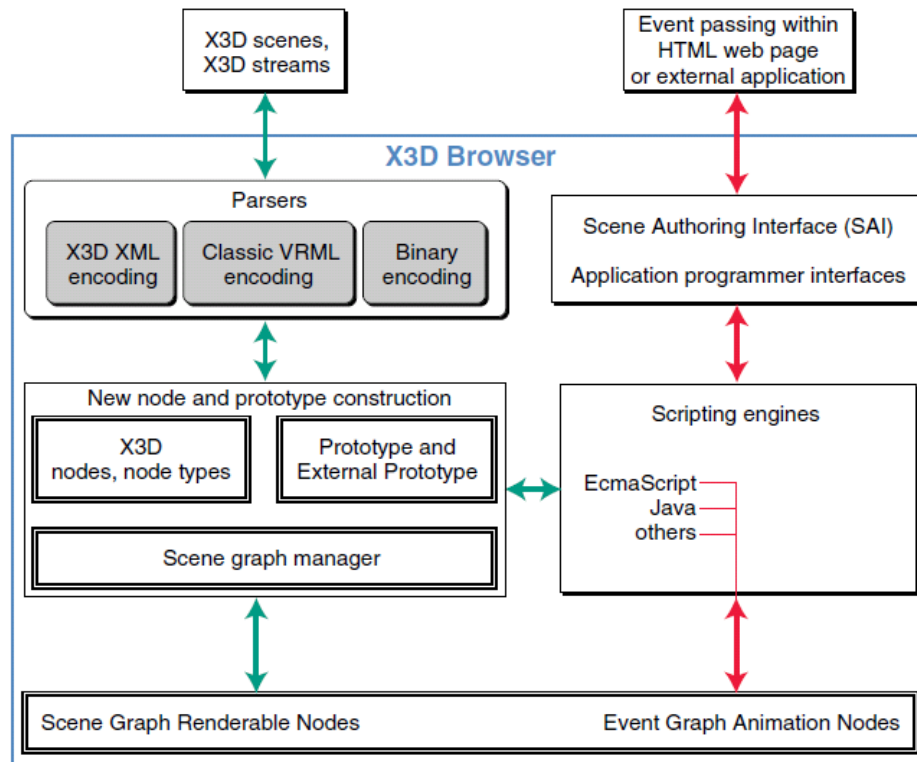


Fig.3. Arquitectura del estándar del X3D.

La manipulación y la modificación de los objetos definidos en la escena X3D, como se puede observar en la figura anterior se logra mediante peticiones realizadas por la página WEB donde el estándar se comunica con el usuario mediante un API como es el caso del SAI (Escena de Acceso de Interfaz). Aquí podemos señalar el uso de Ajax3d, el cual utiliza SAI para acceder a la escena 3D en tiempo real y XMLHttpRequest para almacenar y recuperar datos de la aplicación 3D, además de emplear el DOM para manipular el contenido de páginas WEB en respuesta a cambios que se ejecuten. Ajax3D se basa en la integración de dos modelos de programación: el DOM y el SAI. El DOM define un modelo para los programas de interacción con un documento WEB y el SAI permite a los programadores controlar las escenas X3D. El código JavaScript que se ejecuta en una página WEB puede interactuar con ambos de forma simultánea.

Ventajas de X3D:

- Fácil de aprender.
- Sencillo y ligero.
- Es extensible.
- Es abierto.
- Menos consumo de recursos.
- Mayor rapidez en la interacción con los elementos de la escena.
- Mayor rapidez en la transmisión de los datos.
- X3D tiene una única interfaz de programación de aplicaciones unificada (API).
- X3D da soporte para múltiples codificaciones de archivos: VRML97, XML y binario comprimido.
- X3D emplea una arquitectura modular para dar una mayor extensibilidad y flexibilidad.
- X3D se agrupa por componentes que pueden ser usados por las implementaciones para una plataforma definida o un mercado concreto.
- X3D también incluye el concepto de perfiles, una colección predefinida de componentes empleados generalmente en ciertas aplicaciones, plataformas, o en escenarios, por ejemplo el intercambio geométrico entre herramientas de diseño.
- X3D permite tener un soporte particular para cada necesidad. El mecanismo de componentes X3D también permite a las empresas implementar sus propias extensiones de acuerdo con un riguroso grupo de reglas.

## 1.5 Herramientas de desarrollo

### 1.5.1 Rational Rose

Las herramientas CASE (**Computer Aided Software Engineering**, Ingeniería de Software Asistida por Ordenador), son aplicaciones informáticas cuyo objetivo es aumentar la productividad en el proceso de desarrollo de software disminuyendo el coste de las mismas en términos de tiempo y dinero. Rational Rose es una herramienta CASE para el modelado visual mediante UML de sistemas de software, permitiendo de esta forma especificar, analizar y diseñar sistemas antes de codificarlos. Posibilita la generación de código a partir de los modelos. La ingeniería de código (directa e inversa) es posible para ANSI C++, Visual C++, Visual Basic 6, Java, J2EE/EJB, CORBA, Ada 83, Ada 95, Bases de datos: DB2, Oracle, SQL 92, SQL Server, Sybase, Aplicaciones WEB [39].

### 1.5.2 Aptana Studio

Aptana Studio es un IDE de desarrollo para aplicaciones de la WEB 2.0, gratuito, código libre, con soporte Ajax, PHP, Ruby on Rails, Adobe Air, iPhone, etc. Con Aptana se facilita en desarrollo integrado de Ajax con las tecnologías emergentes. Aptana está basado en el conocido entorno de desarrollo Eclipse (IDE), también Open Source. Pero mientras que Eclipse está focalizado en el desarrollo para Java, Aptana Studio es una distribución focalizada en el desarrollo WEB, con soporte HTML, CSS y Javascript, así como opcionalmente a otras tecnologías mencionadas como PHP, Adobe Air o Ruby on Rails [40].

### 1.5.3 Vivaty Studio

Es una herramienta de diseño libre. Vivaty, actúa muy parecido a otras herramientas de modelado 3D que permite al usuario importar una serie de tipos de formatos libres y de manipular una serie de variables. Posibilita la modificación rápida y fácil de los tipos de archivo VRML y X3D, así como la exportación a estos formatos. Está optimizado para crear contenido 3D para la entrega en un navegador WEB usando estándares libres de Internet [41].

### 1.5.4 X3D-Edit

Es una herramienta que está principalmente orientada a la programación utilizando el estándar X3D y dentro de sus prestaciones está la corrección de sintaxis que posibilita corregir los errores básicos en el código. Brinda la posibilidad de crear escenas virtuales organizando todos los nodos X3D que están presentes y permite acceder a sus atributos. Además, dispone de ayudas ajustables al contexto facilitando así la programación de la escena virtual [42].

## 1.6 Selección de las tecnologías, lenguajes y herramientas a utilizar

Después de analizar algunas de las tecnologías, lenguajes, estándares y herramientas más importantes para el desarrollo de aplicaciones WEB en el mundo, hemos seleccionado las más adecuadas para lograr que los objetivos trazados se cumplan satisfactoriamente. A continuación se mencionan las seleccionadas: como lenguaje del lado del servidor utilizaremos PHP, para la interacción de los modelos 3D en la WEB utilizaremos X3D, como framework se eligió JQuery, como servidor WEB se escogió Apache, como gestor de base de datos se pueden utilizar PostgreSQL o MySQL gracias al empleo del PDO y como entornos de desarrollo integrado (IDE), Aptana Studio para el trabajo con HTML, JavaScript,



PHP y Ajax; y X3D-Edit como editor para X3D. Por último se seleccionó RUP como metodología de desarrollo y el patrón de arquitectura Modelo Vista Controlador.

Seleccionamos PHP por:

- Ser un lenguaje flexible multiplataforma que brinda la posibilidad de programar orientado a objeto.
- Ser libre sin ningún tipo de restricciones para su uso y por acoplarse de una forma muy dinámica a MYSQL y a PostgreSQL.
- Poseer una amplia biblioteca de funciones con manejo de excepciones.

Seleccionamos X3D por:

- Ser ligero, sencillo y fácil de aprender.
- Ser libre para su utilización y distribución.
- Existir gran cantidad de herramientas de diseño que exportan modelos en este formato.
- Consumir menos recursos que otras tecnologías de visualización 3D.
- Una mayor rapidez en la interacción con los elementos de la escena.

Seleccionamos JQuery por:

- Ofrecer una amplia serie de funcionalidades implementadas en JavaScript.
- Ofrecer una amplia ayuda en la creación de interfaces de usuario y efectos dinámicos.
- Ser un framework de código abierto y libre para su uso.
- No sobrecargar las páginas WEB debido a su pequeño tamaño.
- Ser un producto estable y bien documentado.

Seleccionamos Apache por:

- Ser un servidor WEB que se integra con varios lenguajes scriping.
- Ser un servidor de código libre sin restricciones para su utilización.
- Poseer una arquitectura robusta y sólida que brinda seguridad a las aplicaciones WEB.
- Poseer soporte para las bases de datos.

Utilizamos PDO por:

- Por ser una interfaz uniforme para acceder a varias bases de datos.
- Por proveer una capa de abstracción de acceso a la base de datos.

- Por permitir una mayor portabilidad a la aplicación.

Seleccionamos Aptana Studio por:

- Ser libre para su utilización y distribución.
- Poseer librerías de funciones en JavaScript populares y en Ajax/JavaScript para utilizar en los proyectos.
- Poseer ejemplos ya creados para empezar a conocer las posibilidades de desarrollo rápidamente.
- Tener ayudas visuales para la escritura de scripts en diversos lenguajes, como coloreado y auto escritura del código, de ayudas contextuales referencia a medida que se escribe, etc.

Seleccionamos X3D-Edit por:

- Ser un editor de archivo de gráficos en 3D para X3D.
- Permitir desarrollar escenas virtuales interactivas con el estándar extensible X3D.
- Suministrar un mayor nivel de comprensión facilitando trabajo.
- Presentar ayudas para la programación de la escena.
- Permitir trabajar con los nodos X3D y sus atributos, de una forma fácil.

Seleccionamos RUP por:

- Proveer un entorno de proceso de desarrollo configurable.
- Estar dirigido por casos de uso nos da la posibilidad de centrarnos en las funcionalidades específicas y más importantes del producto a desarrollar.
- Presentar una estructura estable y sólida que ocupa todas las fases de vida de nuestro producto.
- Posibilitar llevar a cabo el desarrollo de nuestro proyecto de forma organizada y con la mayor calidad posible.

Seleccionamos el patrón de arquitectura Modelo Vista Controlador por:

- Brindar la posibilidad de desarrollar la aplicación forma organizada.
- Facilitar agregar nuevos tipos de datos según sea requerido por nuestra aplicación, pues son independientes del funcionamiento de las otras capas.
- Facilitar el mantenimiento de la de la aplicación.

Seleccionamos Vivaty Studio por:

- Ser una herramienta de diseño libre.
- Estar especializado en el desarrollo de entornos a partir del estándar X3D.
- Facilitar la manipulación y aprendizaje.
- Posibilitar la importación de múltiples formatos de mallas.
- Tener la capacidad de exportación a X3D.

### **Consideraciones finales del capítulo**

El desarrollo de este capítulo nos permitió seleccionar las tecnologías y recursos más apropiados para la realización del sistema: un software que permita una correcta visualización de modelos anatómicos tridimensionales que admita la interacción con los mismos (selección, rotación, escalamiento, etc.), que posibilite el acceso por parte de varias estaciones de trabajo a estos recursos y funcionalidades requiriendo poco consumo de hardware. A partir de las características antes mencionadas es que se decide implementar el sistema sobre plataforma WEB, con ello se elige como tecnología para la visualización tridimensional el estándar X3D ya que es multiplataforma, libre e integrado al lenguaje XML, agilizando enormemente la transmisión y visualización a través de la red, además de brindar una correcta y detallada visualización de modelos complejos con gran cantidad de polígonos y poseer herramientas que facilitan el diseño y exportación de modelos tridimensionales a este formato. También el uso de este estándar nos posibilitará vincularlo a diversos lenguajes y tecnologías de desarrollo WEB como JavaScript y Ajax con lo que podremos visualizar e interactuar con la escena 3D logrando con ello una mayor rapidez y rendimiento de la aplicación.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

En este capítulo se realizará la descripción de la solución propuesta para dar respuesta a la situación problemática planteada. Como los procesos de negocios no están visibles claramente se opta por la realización de un modelo de dominio, donde se identifican los conceptos y sus relaciones con el objetivo de lograr un mejor entendimiento del contexto en que se desarrolla el sistema. También se especificarán los requisitos funcionales y no funcionales de la aplicación, además se muestran los diagramas de casos de uso del sistema y se realizará la descripción de cada uno de los casos de usos que guiarán el proceso de desarrollo de software para brindar una solución eficiente de acuerdo con los requerimientos planteados.

### 2.1. Modelo de Dominio

Cuando no se pueden identificar claramente los procesos de negocio se elabora un modelo de dominio el cual mediante la representación de un conjunto de conceptos y las relaciones que se establecen entre ellos ayudarán tanto a usuarios como a desarrolladores a usar un vocabulario común que les permita entender el contexto en que se emplaza el sistema a desarrollar. Además con este modelo se podrá posteriormente identificar algunas de las clases necesarias para la implementación de la aplicación.

Para identificar los conceptos que se emplearán en el modelo de dominio se realiza un glosario de términos que responde a los nombres de estos conceptos [43]:

- ❖ **Estudiante:** Persona que cursa en un centro docente de medicina y accede a sus recursos bibliográficos.
- ❖ **Asignatura de anatomía:** Materia docente que estudia la estructura del cuerpo humano.
- ❖ **Computadora:** Es un equipo electrónico con capacidad de procesamiento de información que permite la interacción directa entre el estudiante y datos digitales educativos.
- ❖ **Laboratorio docente:** Instalación formada por un conjunto de computadoras al que pueden asistir los estudiantes para su preparación docente.
- ❖ **Imagen anatómica:** Representación visual de un objeto relacionado a la anatomía humana mediante técnicas de diseño como pintura y fotografía.
- ❖ **Concepto médico:** Conjunto de palabras que ofrecen una información relacionada a la medicina.
- ❖ **Documento:** Objeto o archivo que posee información relacionada a la anatomía humana.

- ❖ **Centro Docente:** Institución organizacional que agrupa un conjunto de medios, recursos y personas con el objetivo de su formación profesional.
- ❖ **Bibliografía complementaria:** Conjunto de recursos de contenido científico que sirven de apoyo al proceso de autoaprendizaje del estudiante.

**Modelo de dominio**

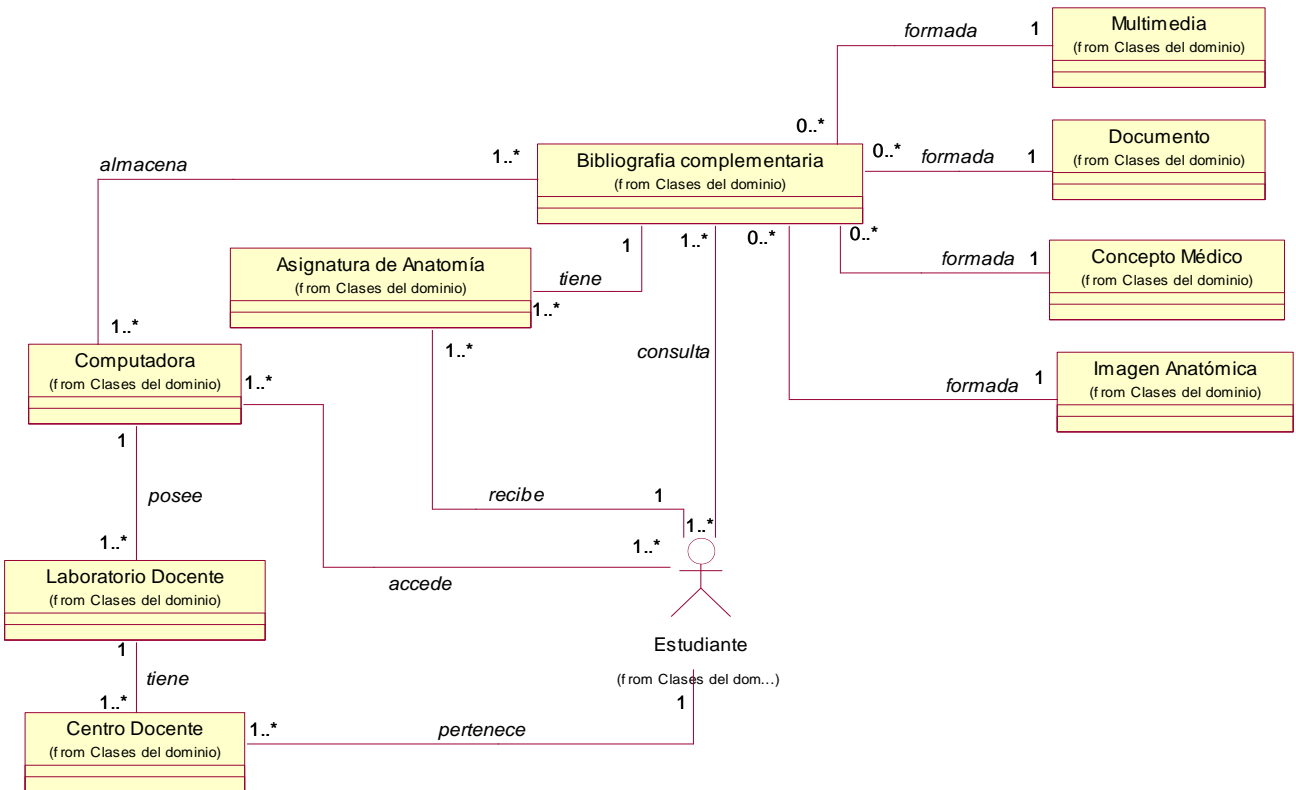


Fig.4. Modelo del dominio.

**2.2. Captura de Requisitos**

A continuación se podrán observar el conjunto de condiciones o capacidades que el sistema debe alcanzar para cumplir con el objetivo trazado.

**2.2.1. Requisitos funcionales**

- RF1. Autenticar usuario.
- RF2. Registrar usuario.

- RF3. Mostrar sistemas del cuerpo
  - RF3.1 Mostrar imágenes de los sistemas del cuerpo humano.
  - RF3.2 Mostrar imágenes de los sistemas del cuerpo en la cabeza.
- RF4. Mostrar imágenes médicas.
  - RF4.1 Mostrar imágenes anatomía esquelética.
  - RF4.2 Mostrar imágenes de anatomía radiológica.
  - RF4.3 Mostrar imágenes del corte transversal del cuerpo humano.
- RF5. Mostrar modelos médicos 3D.
  - RF5.1 Mostrar modelo tridimensional de los huesos.
  - RF5.2 Mostrar modelo tridimensional del órgano.
- RF6. Insertar comentario en el foro.
- RF7. Gestionar tema.
  - RF7.1 Insertar tema.
  - RF7.2 Eliminar tema.
  - RF7.3 Publicar tema.
  - RF7.4 Despublicar tema.
- RF8. Gestionar comentario
  - RF8.1 Eliminar comentario.
  - RF8.2 Publicar comentario.
  - RF8.3 Despublicar comentario.
- RF9. Gestionar modelo 3D.
  - RF9.1 Insertar modelo 3D.
  - RF9.2 Publicar modelo 3D.
  - RF9.3 Despublicar modelo 3D.
  - RF9.4 Eliminar modelo 3D.
  - RF9.5 Cargar modelo 3D externo.
- RF10. Gestionar imagen médica.
  - RF10.1 Insertar imagen médica.
  - RF10.2 Publicar imagen médica.
  - RF10.3 Despublicar imagen médica.
  - RF10.4 Eliminar imagen médica.
- RF11. Gestionar documento.

RF11.1 Insertar documento.

RF11.2 Publicar documento.

RF11.3 Despublicar documento.

RF11.4 Eliminar documento.

RF12. Gestionar concepto médico.

RF12.1 Insertar concepto médico.

RF12.2 Publicar concepto médico.

RF12.3 Despublicar concepto médico.

RF12.4 Eliminar concepto médico.

RF13. Gestionar usuario.

RF13.1 Eliminar usuario.

RF13.2 Habilitar usuario.

RF13.3 Inhabilitar usuario.

RF14. Buscar recurso.

RF15. Buscar usuario.

### **2.2.2. Requisitos no funcionales**

A continuación se reflejan las cualidades o propiedades que el producto debe tener para lograr características que lo hagan rápido, confiable, atractivo, usable y seguro, etc. Aquí se evidencian las restricciones del entorno, de la implementación, el rendimiento del sistema, así como las dependencias de plataformas entre otros aspectos.

#### **❖ Usabilidad**

- Debe ser de fácil manejo o manipulación aún para personas que no tengan muchos conocimientos de trabajo en computadoras y ambientes WEB.

#### **❖ Seguridad**

##### **➤ Confidencialidad**

- Garantizar que la información solo sea editada o modificada por las personas que tienen permisos para realizar esta actividad.
- Se debe tener un mecanismo de seguridad basado en el modelo (Autenticación, Autorización y Auditoría).
- Mostrar mensajes de error en caso de que el usuario intente loguearse con datos incorrectos.

➤ **Integridad**

- Prevenir posibles fallos y recuperarse ante la afectación de uno de estos.
- Para lograr la recuperación ante fallos deberá existir un mecanismo de recuperación de los datos, como salva automáticas de la base de datos o copias de respaldo que puedan contribuir posteriormente al restablecimiento ante pérdida total de información.
- Protección contra acciones no autorizadas que puedan afectar la integridad de los datos.

➤ **Disponibilidad**

- Se le garantizara el acceso a los usuarios autorizados a la información correspondiente de la manera más rápida posible, o sea, la seguridad implantada en el sistema no afectará el tiempo de respuesta a las solicitudes de los usuarios.
- Los tiempos de respuesta deben ser rápidos en menos de 4 segundos.
- Se debe garantizar la rápida visualización de imágenes y modelos tridimensionales que no sobrecargue el procesamiento de computadoras sin muchas prestaciones.

❖ **Soporte**

- El sistema debe ser de fácil mantenimiento.
- El sistema debe ser capaz de asimilar la incorporación de nuevos recursos que enriquezcan su contenido.

❖ **Interfaz**

- La aplicación deberá poseer una interfaz amigable y de fácil comprensión que permita un rápido entendimiento por parte de los usuarios.
- La interfaz debe permitir el fácil manejo para todo tipo de usuarios.
- Las funcionalidades deberán estar visibles en todo momento para facilitar la navegación a través de la aplicación WEB.

❖ **Requerimientos de Software**

➤ **Requisitos óptimos del software del servidor**

- Sistema operativo Windows.
- Para la implantación del sistema se requiere un servidor WEB Apache versión 2.0 o superior.



- Servidor de base de datos PostgreSQL 8.2.x o MySQL 5.x.
- PHP 5 o superior. Recomendado PHP 5.2.x.

➤ **Requisitos óptimos del software del cliente**

- Navegador WEB con capacidad de interpretación de JavaScript (Mozilla Firefox).
- Instalación del plug-ins de visualización tridimensional en la WEB Vivaty Player.

❖ **Requerimientos de Hardware**

➤ **Requisitos mínimos de la maquina servidora**

- Procesador de tipo Pentium IV con velocidad de 500 MHz.
- Memoria RAM de 512 MB.
- Capacidad mínima de disco duro de 20 GB.
- Periféricos tarjeta red.

➤ **Requisitos mínimos de la estación de trabajo o cliente**

- Procesador Pentium IV velocidad de 500 MHz
- Memoria RAM de 512 MB.
- Capacidad mínima de disco duro de 5GB.

❖ **Portabilidad**

- El sistema debe de fácil instalación sobre plataforma Windows.

### 2.3. Modelo de Casos de Uso del Sistema

En esta sección se muestran los posibles actores del sistema que van a interactuar con lo casos de usos definidos a partir de los requisitos funcionales antes expuestos. Lo cual posibilitará un mejor entendimiento de las funcionalidades que se requieren del sistema y darán un resultado de valor para los actores identificados.

2.3.1. Descripción de los actores del sistema

Actor	Descripción
Administrador	Es la persona que tiene permisos y privilegios de edición o modificación de los recursos con que cuenta el sistema.
Estudiante	Es la persona que tiene acceso a las funcionalidades públicas y educativas, este puede interactuar con los recursos y contenido del sistema.
Usuario Sistema	Es la persona que accede al sistema mediante el proceso de autenticación.

Relación entre los actores del sistema

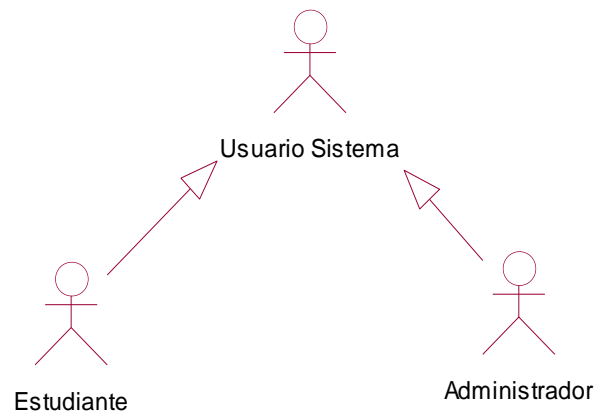


Fig.5. Relación entre los actores del sistema.

### Diagrama de Casos de Uso por paquetes

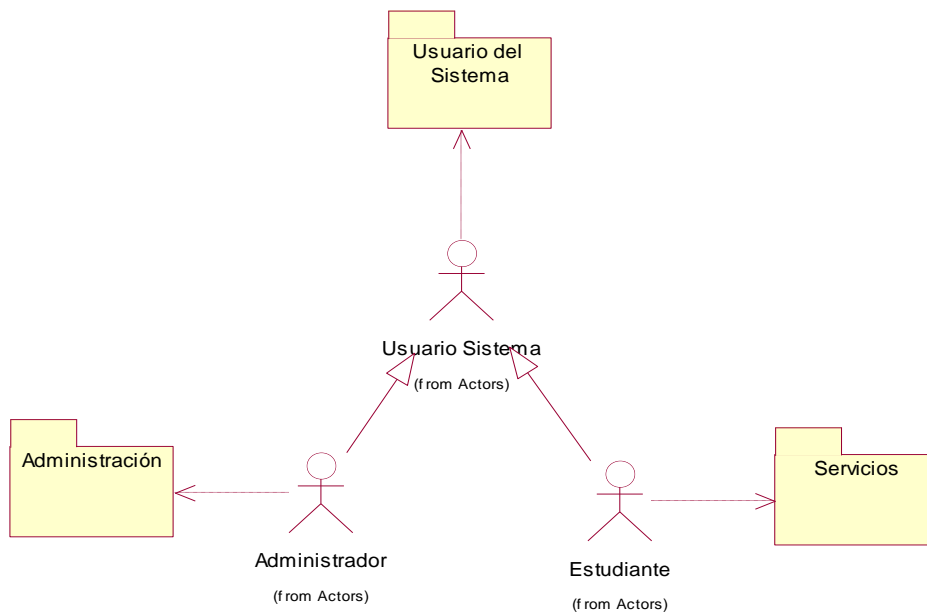


Fig.6. Diagrama de Casos de Uso por paquetes

### Diagrama de Casos de Uso del paquete Usuario del Sistema

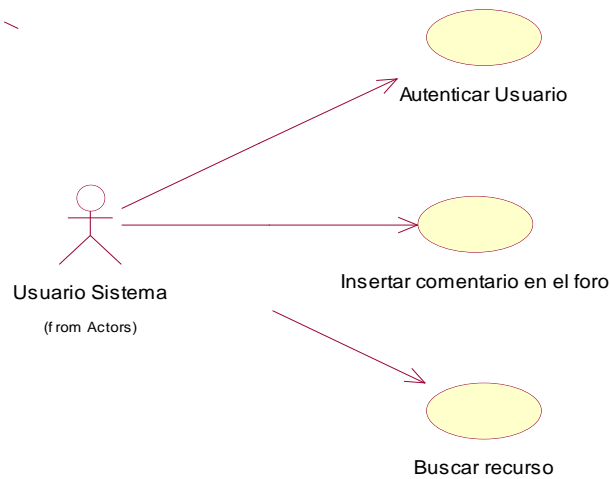
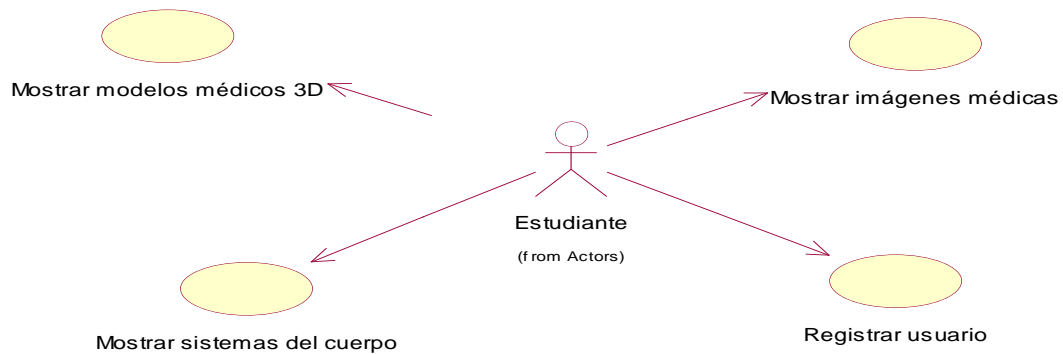


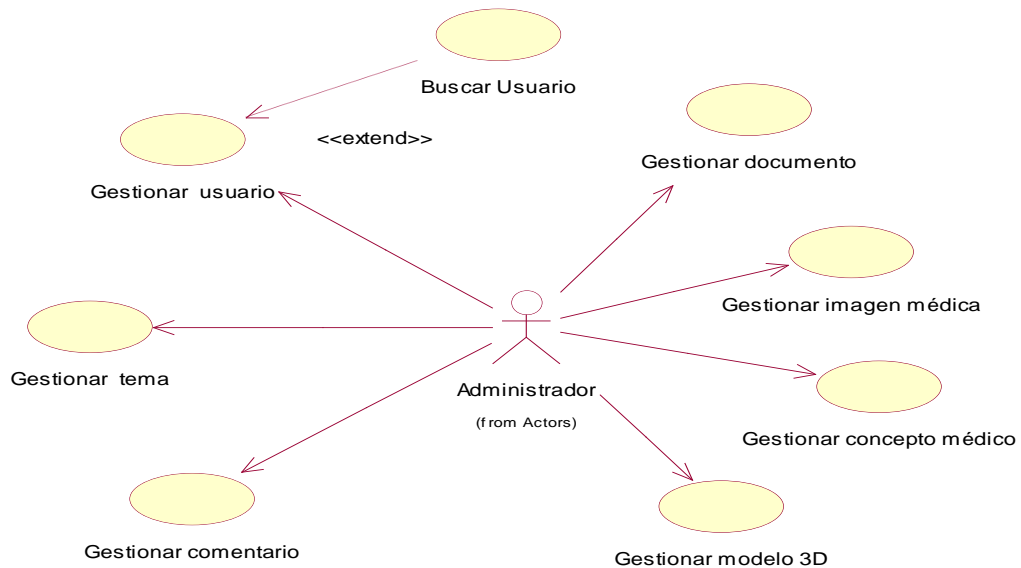
Fig.7. Diagrama de Casos de Uso del paquete Usuario del Sistema

**Diagrama de Casos de Uso del paquete Servicios**



**Fig.8.** Diagrama de Casos de Uso del paquete Servicios.

**Diagrama de Casos de Uso del paquete de Administración**



**Fig.9.** Diagrama de Casos de Uso del paquete Servicios de Administración.

**2.3.2. Descripción de los Casos de Usos del Sistema**

Con la descripción de los casos de usos del sistema se brinda una mayor información y entendimiento de la aplicación a implementar. También se reflejan las acciones de los actores y las respuestas del sistema. Aquí se muestra una de las descripciones de uno de los Casos de Uso más importantes. Para ver las restantes descripciones de los casos de usos del sistema remitirse al **Anexo I**.

**Caso de Uso “Mostrar modelos médicos 3D”**

Caso de Uso:	Mostrar modelos médicos 3D
Actores:	Estudiante
Resumen:	El caso de uso se inicia cuando el estudiante accede a la interfaz sección 3D donde podrá observar las partes del esqueleto humano en tres dimensiones y podrá seleccionar e interactuar con la parte específica que desea analizar. Además podrá observar la representación en tres dimensiones de un órgano determinado y al seleccionarlo puede ver la información del mismo. El caso de uso termina cuando se muestra el modelo 3D y su descripción.
Precondiciones:	El estudiante tiene que estar autenticado.
Referencias	RF5, RF5.1, RF5.2
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1- Accede a la sección 3D.	2- El sistema ejecuta alguna de las siguientes acciones. a) Para interactuar con los modelos tridimensionales de los huesos ir a la sección “Mostrar modelo tridimensional de los huesos” b) Para interactuar con los modelos tridimensionales de los órganos ir a la sección “Mostrar modelo tridimensional del órgano” c) Para observar la descripción de un hueso y órgano determinado ir a la sección “Mostrar descripción de los modelos”
Sección: “Mostrar modelo tridimensional de los huesos”	
Acción del Actor	Respuesta del Sistema
1- Accede a la sección 3D.	2- Muestra un esqueleto en 2D.
3- Selecciona en la figura del esqueleto en 2D la parte del esqueleto que desea visualizar en 3D.	4- Sombrea en color rojo la parte seleccionada en la imagen y la muestra en 3D.

<i>Prototipo de Interfaz</i>	
<b>Sección: “Mostrar modelo tridimensional del órgano”</b>	
Acción del Actor	Respuesta del Sistema
1- Accede a la sección 3D.	2- Muestra una lista de imágenes de los órganos del cuerpo.
3- Selecciona el órgano deseado.	4- Muestra el modelo del órgano en 3D.
<i>Prototipo de Interfaz</i>	
<b>Sección: “Mostrar descripción de los modelos”</b>	
Acción del Actor	Respuesta del Sistema
1- Selecciona en el modelo 3D la parte específica de la cual quiere ver la descripción.	2- Sombrea la parte seleccionada en el modelo y muestra la información relacionada a la misma.
<i>Prototipo de Interfaz</i>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Poscondiciones	1- Se visualiza el modelo 3D de la parte del esqueleto seleccionada. 2- Se visualiza el modelo del órgano determinado. 3- Se visualiza la descripción de una parte del modelo deseado.

### Consideraciones finales del capítulo

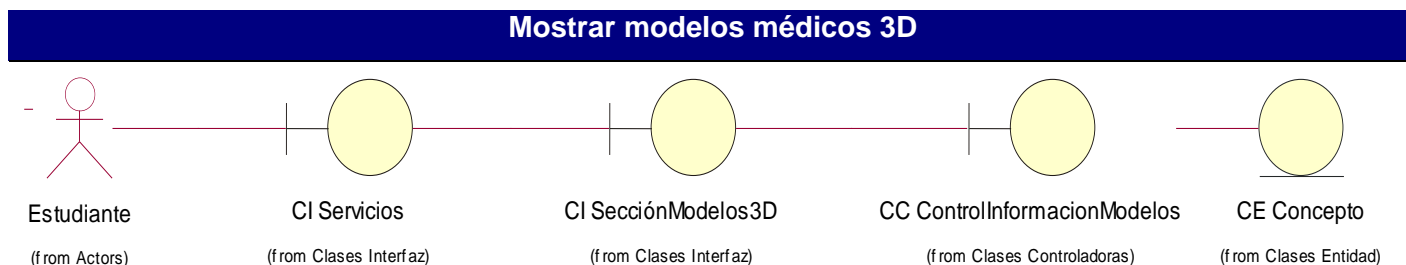
En el capítulo se detalló la lógica del negocio y el funcionamiento de cada uno de sus procesos estableciendo los roles de los diferentes actores que participan en el sistema. Además se detallaron los casos de usos del sistema para brindar una mayor información y entendimiento de la aplicación para pasar a la fase de análisis y diseño.

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

Este capítulo es de gran importancia porque describe la puesta en práctica de la construcción de la solución propuesta y para la realización del mismo se desarrolla el análisis del sistema, mediante la elaboración del diagrama de clases del análisis de la aplicación. También se encuentra el diseño del sistema que parte del análisis anterior, definiéndose los diagramas de clases de diseño y los diagramas de interacción siguiendo un flujo de proceso.

### 3.1. Análisis

Con el análisis se obtiene una mejor visión del sistema a implementar ya que este está centrado en los requisitos funcionales que se definieron en el capítulo anterior. Además representa la relación entre los actores del sistema y las clases de análisis, las cuales están clasificadas en: clase de interfaz, clase controladora y clase entidad. Para ver los restantes diagramas de clases del análisis remitirse a **Anexo II**.



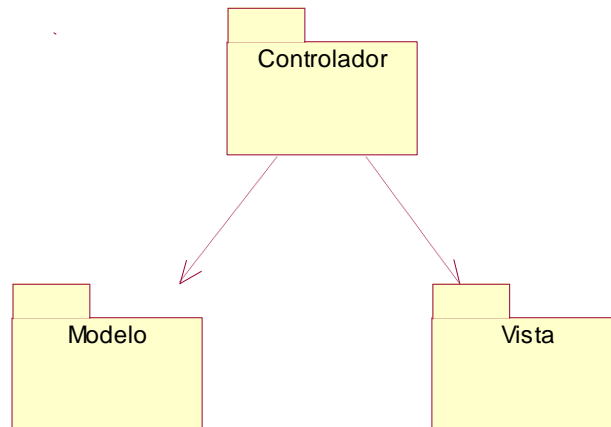
**Fig.10.** Diagramas de clases de análisis.

### 3.2. Diseño

El diseño es el refinamiento de los modelos de análisis donde se tienen en cuenta principalmente los requisitos no funcionales para lograr más tarde una solución sólida y factible. Esto es lo que ayuda a mantener una arquitectura estable y sólida para poder crear un plano del modelo de implementación.

#### 3.2.1. Patrón de Arquitectura (Modelo Vista Controlador)

Como se explicó en el capítulo I se utilizó el patrón Modelo Vista Controladora (MVC) por las ventajas que se expusieron anteriormente y al aplicarlo se logró que la aplicación facilitara la reusabilidad y la comprensión durante y después del desarrollo del sistema ya que es posible modificar cualquiera de estos tres componentes sin que esto afecte el funcionamiento del resto.



**Fig.11.** Patrón Modelo Vista Controladora.

- El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La vista transforma el modelo en una interfaz que permite al usuario interactuar con ella.
- El controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.



**Fig. 12.** Flujo de datos

La entrada de los datos se realizó a través de la Vista y en esta se ubican todas las interfaces que representan al modelo en un formato adecuado para interactuar con el usuario. En este proceso de entrada de datos participa el Controlador el cual toma decisiones, responde a eventos (acciones del usuario) y a su vez invoca cambios en el modelo que pueden también modificar la vista. El procesamiento de datos se realiza en el Modelo que es la representación específica del dominio de la información sobre la cual funciona la aplicación, ocurriendo posteriormente la salida de los datos a través de la Vista.

❖ **Aplicación del patrón en el desarrollo del sistema**

A continuación se representa el código que comúnmente se utiliza para mostrar la información almacenada en una base de datos sin la utilización del patrón:



```

1  <?php
2  // Conectar con la base de datos y seleccionarla
3  $conexion = mysql_connect('localhost', 'mi_usuario', 'mi_contrasena');
4  mysql_select_db('nombre_base_datos', $conexion);
5
6  // Ejecutar la consulta SQL
7  $resultado = mysql_query('SELECT campo1, campo2 FROM tabla', $conexion);
8  ?>
9  <html>
10 <head>
11   <title>Listado</title>
12 </head>
13 <body>
14   <h1>Listado</h1>
15   <table>
16     <tr><th>Campo1</th><th>Campo2</th></tr>
17 <?php
18 // Mostrar los resultados con HTML
19 while ($fila = mysql_fetch_array($resultado, MYSQL_ASSOC))
20 {
21   echo "\t<tr>\n";
22   printf("\t\t<td> %s </td>\n", $fila['campo1']);
23   printf("\t\t<td> %s </td>\n", $fila['campo2']);
24   echo "\t</tr>\n";
25 }
26 ?>
27   </table>
28 </body>
29 </html>
30 <?php
31 // Cerrar la conexion
32 mysql_close($conexion);
33 ?>

```

Fig.12. Mostrar Información

Como se puede observar todo el código está mezclado, el php para realizar la conexión y la consulta a la base de datos, así como el html para mostrar los datos recibidos de la consulta. De esta manera es mucho más difícil mantener y actualizar una aplicación, en respuesta a esta limitante es que se aplica el patrón MVC y a continuación pueden ver un fragmento de código de la aplicación evidenciando lo planteado.

Vista.php	Controlador.php	Modelo.php
<pre> &lt;?php echo "&lt;table&gt;";   echo "&lt;tr&gt;";     echo "&lt;td&gt;";       campo1     echo "&lt;/td&gt;";   echo "&lt;td&gt;";     campo2   echo "&lt;/td&gt;"; echo "&lt;tr&gt;";  while(\$fila = \$datos-&gt;fetch()) {   echo "&lt;tr&gt;";   echo "&lt;td&gt;";     \$fila[0];   echo "&lt;/td&gt;";   echo "&lt;td&gt;";     \$fila[1];   echo "&lt;/td&gt;";   echo "&lt;/tr&gt;"; } echo "&lt;/table&gt;"; ?&gt; </pre>	<pre> &lt;?php include(Modelo.php)  \$peticion = solicita_valor;  \$objetoModelo;  if(\$peticion == X) {   \$objetoModelo-&gt;insertar(); } if(\$peticion == Y) {   \$datos = \$objetoModelo-&gt;listar(); }  require(Vista.php); ?&gt; </pre>	<pre> &lt;?php include(Coneccion.php); class AccesoDatos {   function __construct()   {   }   function insertar(\$atributos)   {     consulta SQL;   }   function listar()   {     consulta SQL;     return \$datos;   }   function eliminar(\$id)   {     consulta SQL;   } } ?&gt; </pre>

Fig.13. Mostrar Información utilizando el patrón MVC

### 3.2.2. Patrones de diseño

Los patrones de diseño son esqueletos para dar soluciones a problemas comunes que ocurren constantemente en el proceso de desarrollo de software. Estos a su vez permiten la reusabilidad, la no reiteración de búsqueda de soluciones a problemas ya solucionados, formalizan un vocabulario común y facilitan la comprensión y aprendizaje a personas ajenas al sistema. Con la utilización de los mismos los diseños serán mucho más flexibles, modulares y reutilizables.

#### ❖ Patrón de Creación (Singleton)

El patrón Singleton en la aplicación asegura que sólo exista una instancia de una clase dentro del sistema proporcionando un punto de acceso común, también conocido como "Clase de instancia única" o

“Instancia única”. La utilización de este patrón nos posibilitará tener solamente un punto de acceso a la clase que se encarga de establecer la conexión a la base de datos.

### 3.2.3. Patrones de Diseño de Arquitecturas de Aplicaciones Empresariales

Se entiende por Enterprise, sistemas cliente /servidor de tres o más capas. Los patrones usados para el desarrollo de este tipo de aplicaciones permiten seleccionar la tecnología a utilizar en cada caso, según el tamaño del sistema a desarrollar, el tipo de plataforma sobre el cual debe funcionar, la complejidad del negocio que el sistema resuelve, etc. Posibilitan mantener la separación de la lógica del negocio, la presentación y los datos. Han habido muchas formas de plantear una solución para este tipo de sistemas, y básicamente todo sistema Enterprise tiene una estructura cliente / servidor distribuido en capas verticales. Estas capas consisten generalmente en algunas de las siguientes: una capa cliente, una capa de aplicación o WEB server, una capa de acceso a la capa de negocio, una capa de modelo de negocio, una capa de persistencia y una base de datos [44].

#### ❖ Patrón de Lógica de Dominio (Table Module)

Por cada tabla tendremos una instancia del negocio. El patrón Table Module organiza la lógica del dominio como una clase por tabla en la base de datos, y una sola instancia de una clase contiene varios procedimientos que interactúan con la base de datos. Table Module está diseñado para trabajar con Record Set<sup>1</sup>. La capa de persistencia podría ser un conjunto de Gateways que hacen consultas a la base de datos y devuelvan los resultados en Record Set, que no son más que una representación en memoria de los resultados que pudo haber arrojado una consulta de SQL.

#### ❖ DAO

DAO es las siglas de **Data Access Object** (Objeto de Acceso a Datos) y actúa como un adaptador entre los componentes del sistema y la base de datos proporcionando una interfaz común entre la aplicación y los dispositivos de almacenamiento de información. La ventaja de usar objetos de acceso a datos es que cualquier objeto de negocio (aquel que contiene detalles específicos de operación o aplicación) no requiere conocimiento directo del destino final de la información que manipula [45].

---

<sup>1</sup>Record Set: Es un elemento esencial de un representado por una estructura en memoria, se ve exactamente como el resultado de una consulta a la base de datos, los que pueden ser generados y manipulados por otras partes del sistema.

### 3.2.4. Diagrama de clases del diseño

Los diagramas de clases del diseño son una representación más concreta que los diagramas de clases del análisis. Representan la parte estática del sistema, las clases y sus relaciones. Para ver los diagramas restantes remitirse al **Anexo III**.

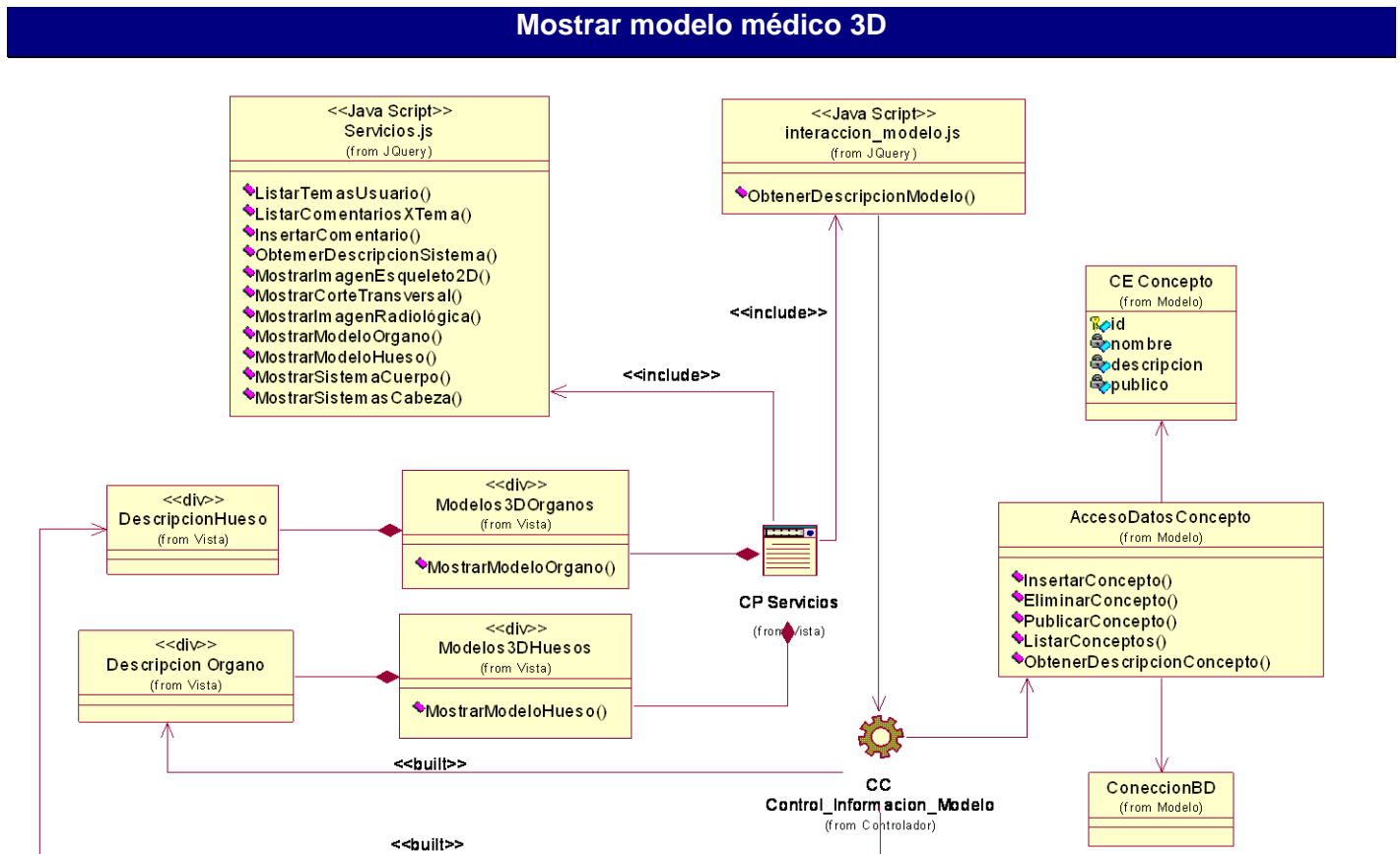


Fig.14. Diagramas de clases del diseño.

### 3.2.5. Diagramas de secuencia

Los diagramas de secuencia representan gráficamente las interacciones del actor y las operaciones generadas que da origen. En estos se muestran como los objetos se comunican entre ellos con el objetivo de cumplir con los requerimientos planteados. Para ver los restantes diagramas de secuencia remitirse al **Anexo IV**.

**Mostrar modelo médico 3D**

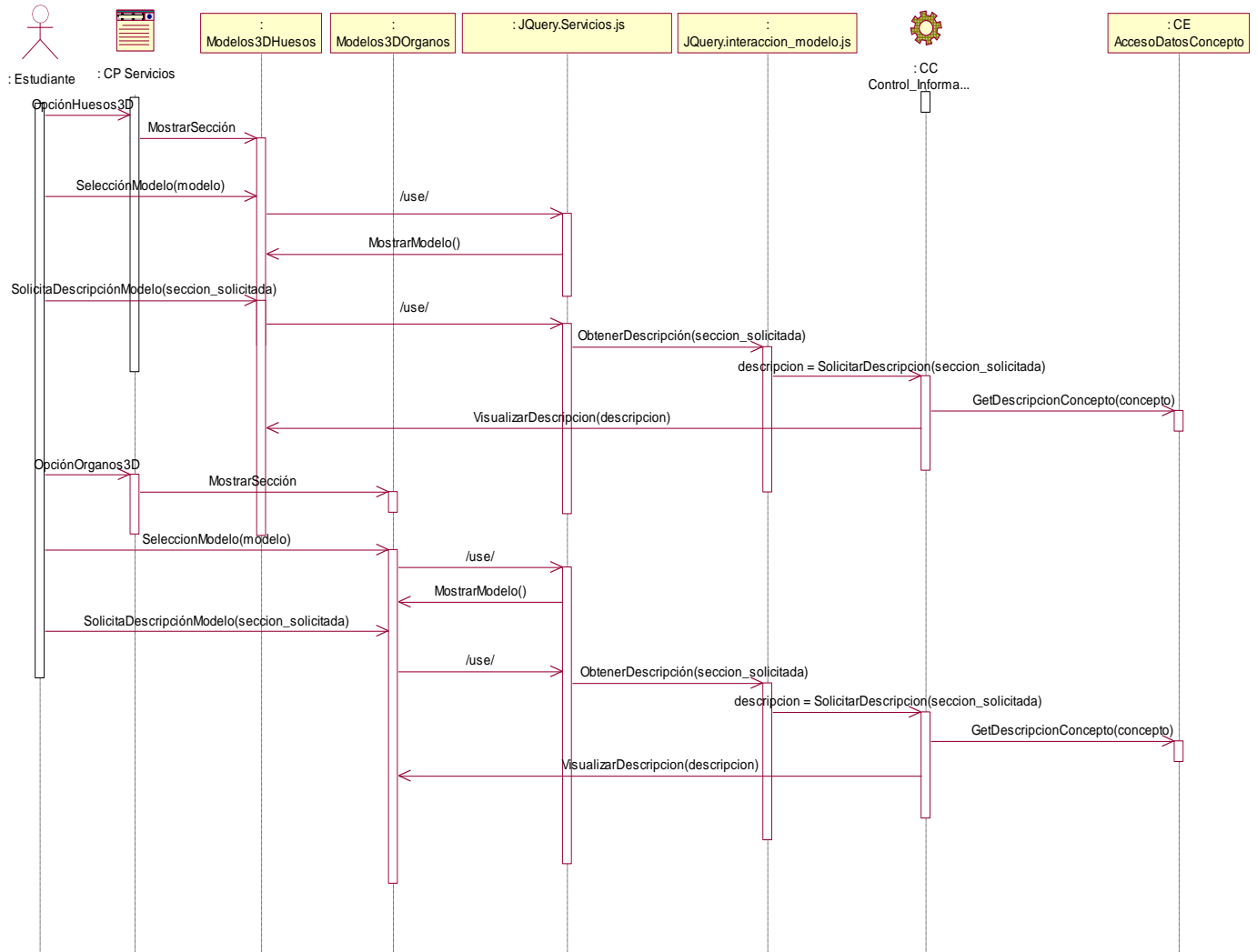


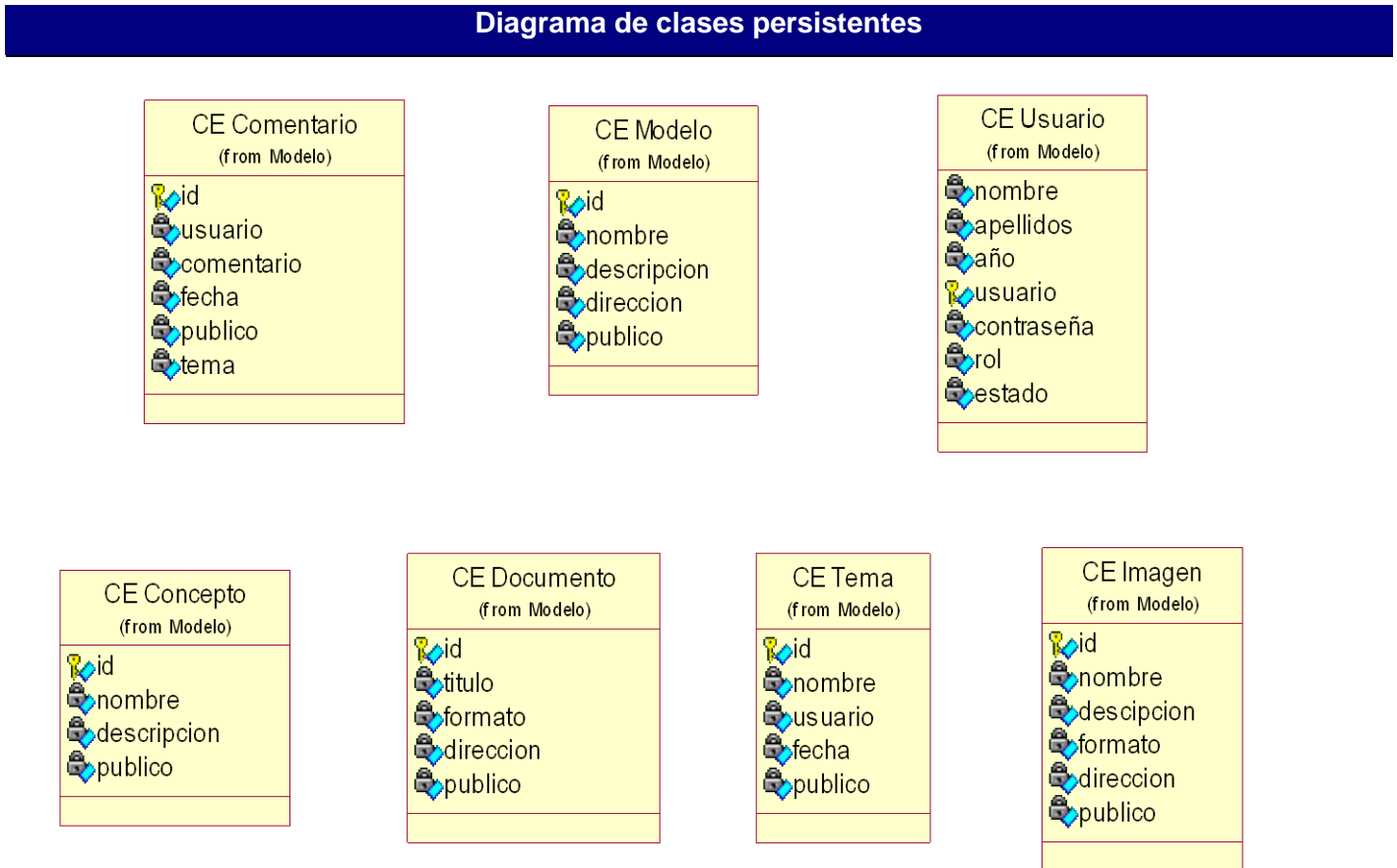
Fig.15. Diagrama de secuencia.

### 3.2.6. Diseño de la Base de Datos

Con un correcto diseño de la base de datos se garantiza obtener un acceso exacto a la información sin ningún tipo de problema. La definición de su estructura permite almacenar los datos, reconocer el contenido, recuperar información y siempre tiene que responder a las necesidades que se desean desarrollar.

❖ **Diagrama de clases persistentes**

A partir de la utilización del patrón anteriormente mencionado Table Module es que se definen las clases persistentes que a continuación se representan:

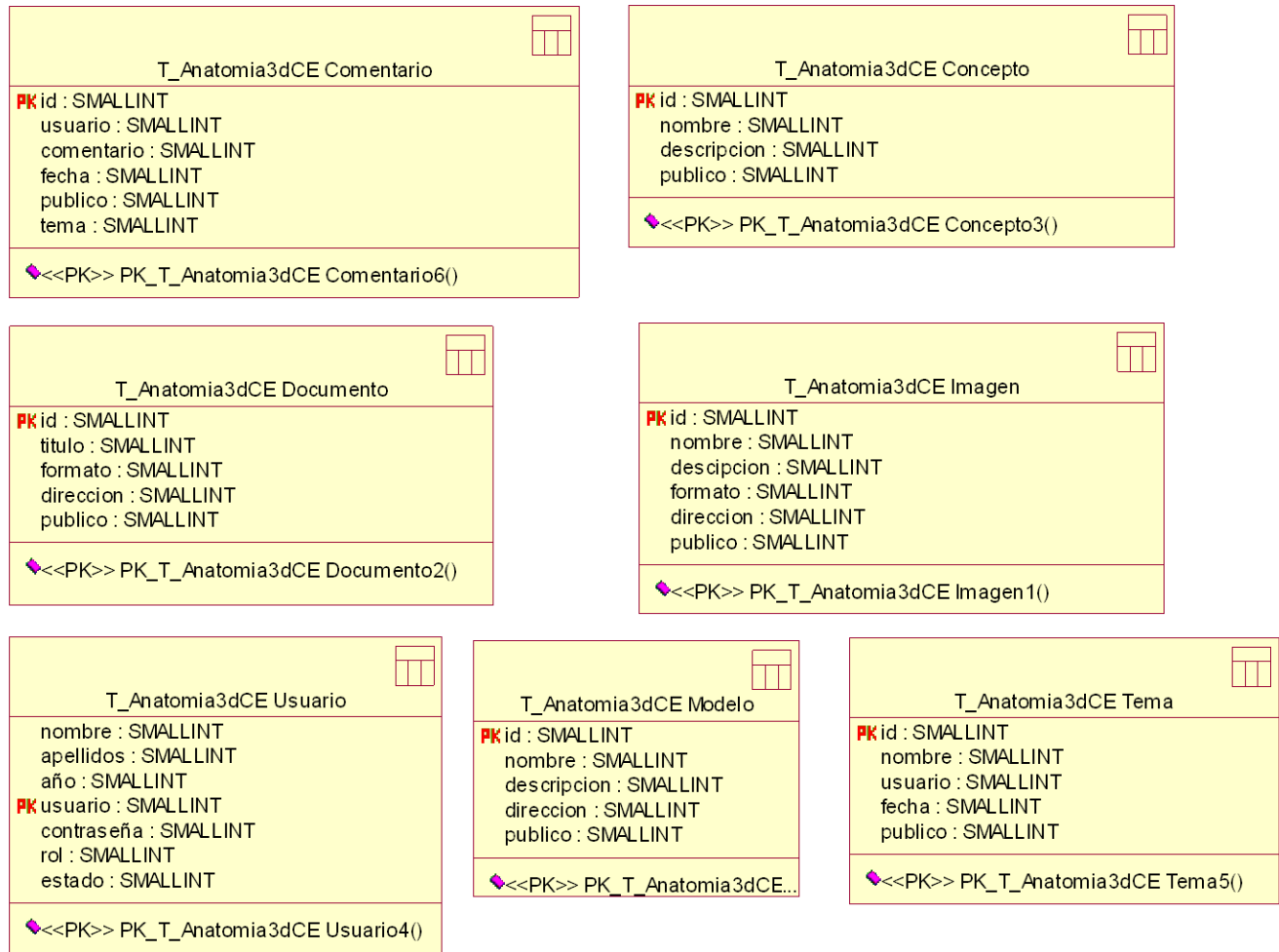


**Fig.16.** Diagramas de clases persistentes.

❖ **Modelo de datos**

A continuación se presentan el modelo de datos generado a partir del diagrama de clases persistentes. Destacar que no existen relaciones entre las clases ya que la lógica de negocio es simple, por lo que se utilizan los patrones antes mencionados que permiten la representación de una clase por tabla en la base de datos que sirve como punto de acceso a los datos de la misma.

**Modelo de datos**



**Fig.17.** Modelo de datos.

❖ **Patrón de acceso a datos**

A continuación se puede observar como se refleja la utilización del Patrón de Acceso a Datos (Table Data Gateway), donde existe una clase de acceso para cada tabla física existente en la base de datos.

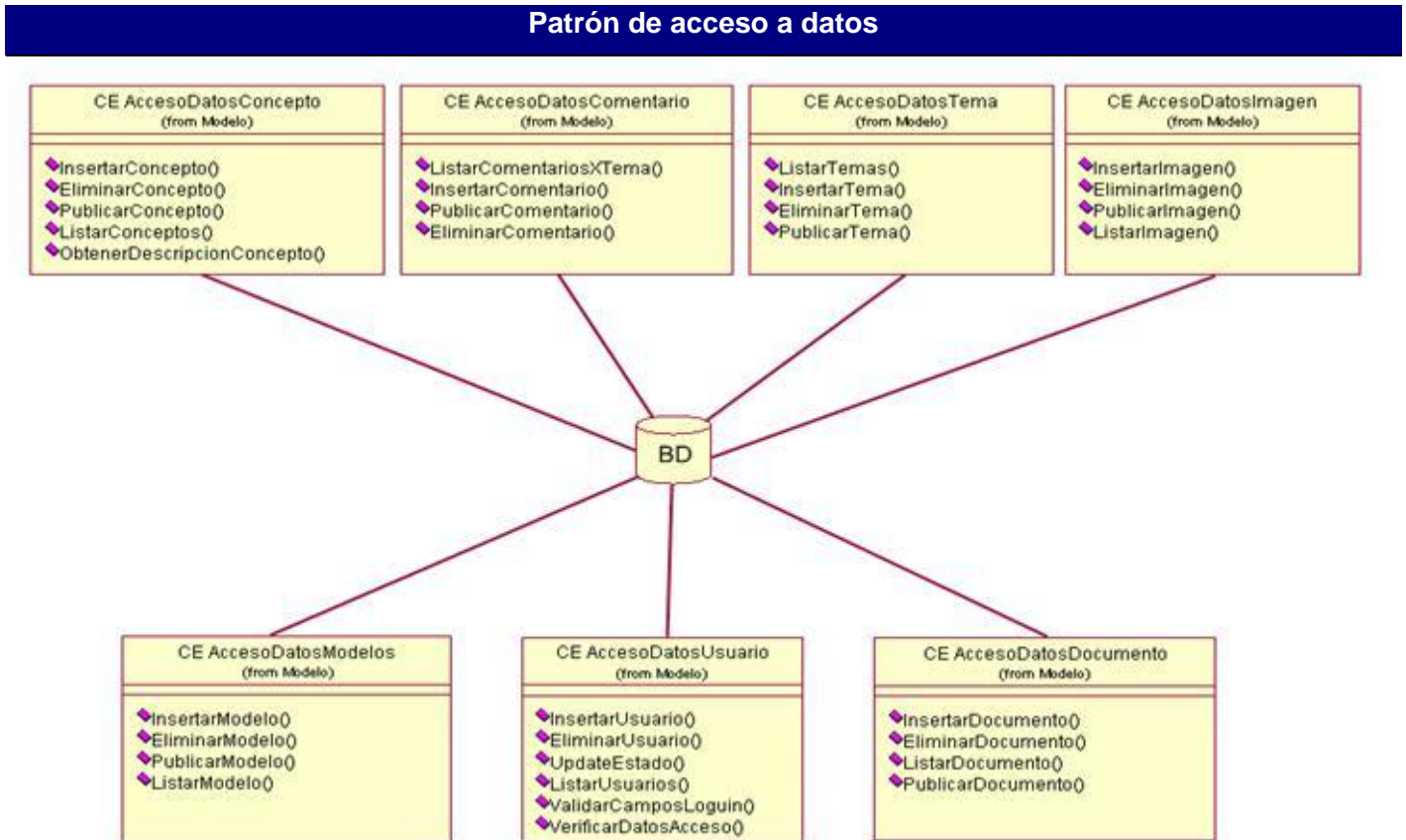


Fig.18. Patrón de Acceso a Datos.

### Consideraciones finales del capítulo

Con la elaboración del capítulo se mostraron los resultados de la etapa del análisis y diseño del sistema para abrir paso a la implementación de la aplicación. Además se describió cada caso de uso y su representación en cada etapa. Los elementos y diagramas obtenidos brindan una imagen más clara de las páginas, subsistemas, algoritmos y demás elementos que influyen en el beneficio de una mejor programación.



## **CAPÍTULO 4: IMPLEMENTACIÓN**

En este capítulo se representarán un conjunto de diagramas que reflejan los componentes por los que está constituido el sistema, así como la distribución física de los mismos facilitando la comprensión de la estructura de la aplicación. Además se plasma cómo se realiza todo el proceso de interacción con la escena 3D, o sea, como se logra a partir de los modelos tridimensionales utilizados gestionar la información almacenada en la base de datos que describen a los mismos, y cómo es posible cambiar su aspecto para informar de su selección.

### **4.1. Modelo de Implementación**

El Modelo de Implementación describe cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue. Estos componentes representan la composición física de la implementación del sistema.

#### **4.1.1. Diagrama de Componentes**

El diagrama de componentes es usado para estructurar el modelo de implementación y mostrar las relaciones entre los elementos de implementación. Este diagrama muestra la estructura de alto nivel del modelo de implementación.

El diagrama de componentes se realizó utilizando la estructura ofrecida por el patrón modelo vista controlador. De esta manera, se posibilitó ubicar adecuadamente los componentes dentro de la correspondiente vista, ya que aquí es donde se permite visualizar la información obtenida y se logra gracias a los componentes ubicados dentro del paquete del modelo a través de los componentes situados en el paquete control.

**Diagrama de componentes**

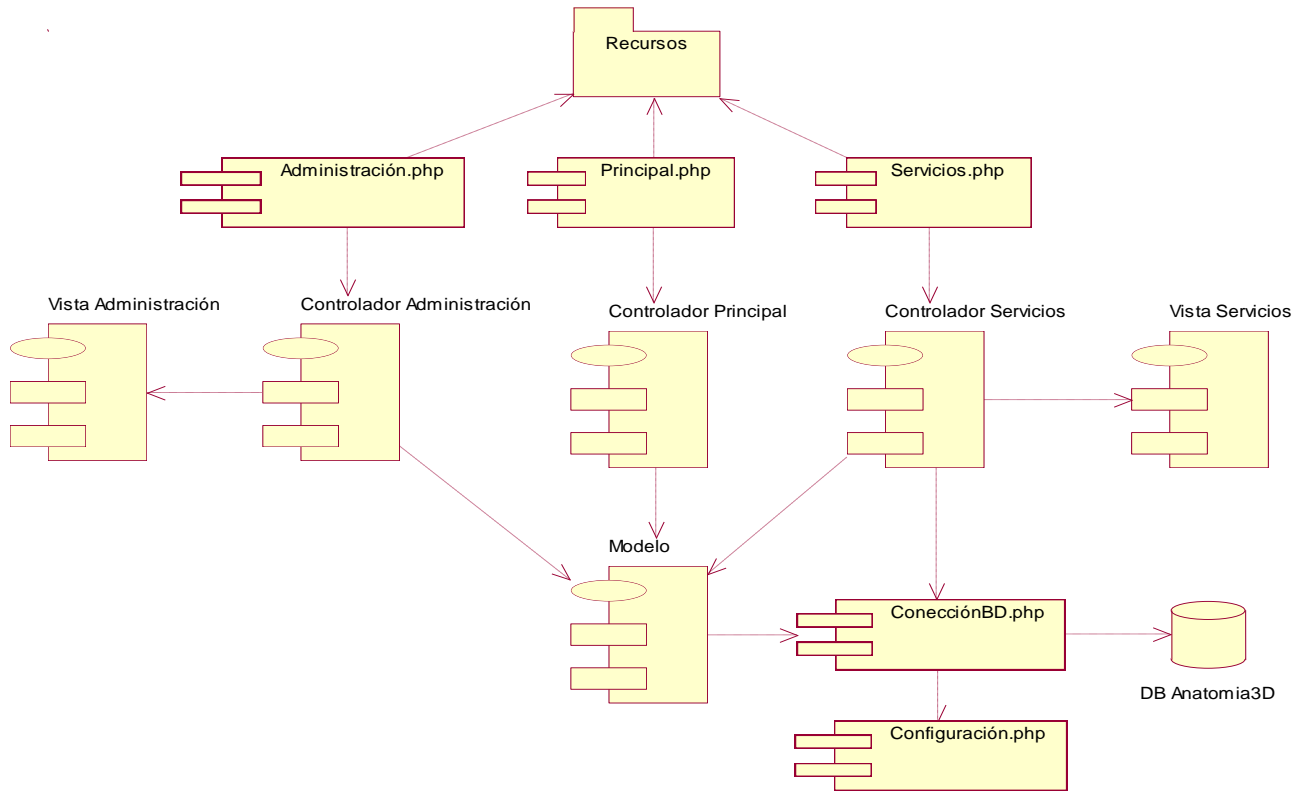


Fig.19. Diagrama de componentes.

**Diagrama de componentes del paquete Recursos**

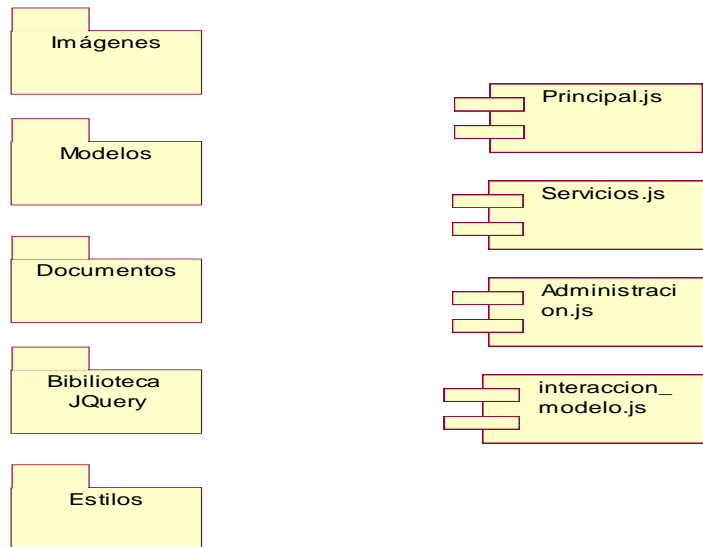


Fig.20. Diagrama de componentes del paquete Recursos.

**Diagrama de componentes del paquete Modelo**

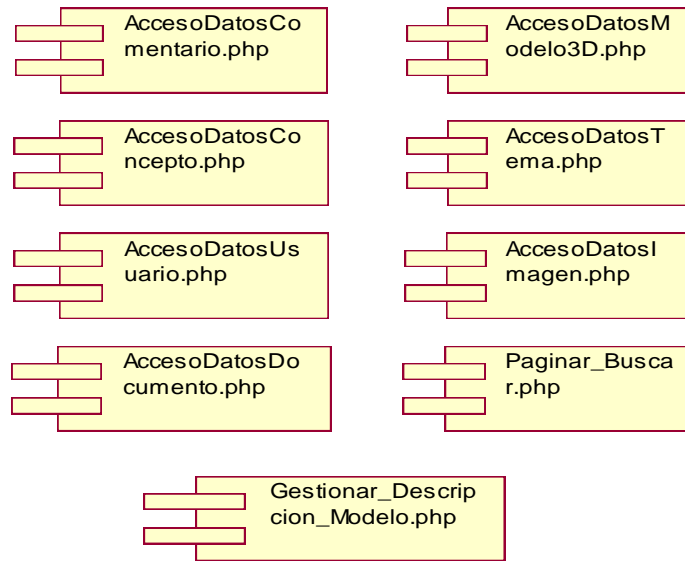


Fig.21. Diagrama de componentes del paquete Modelo.

**Controlador Principal      Controlador Administración      Controlador Servicios**

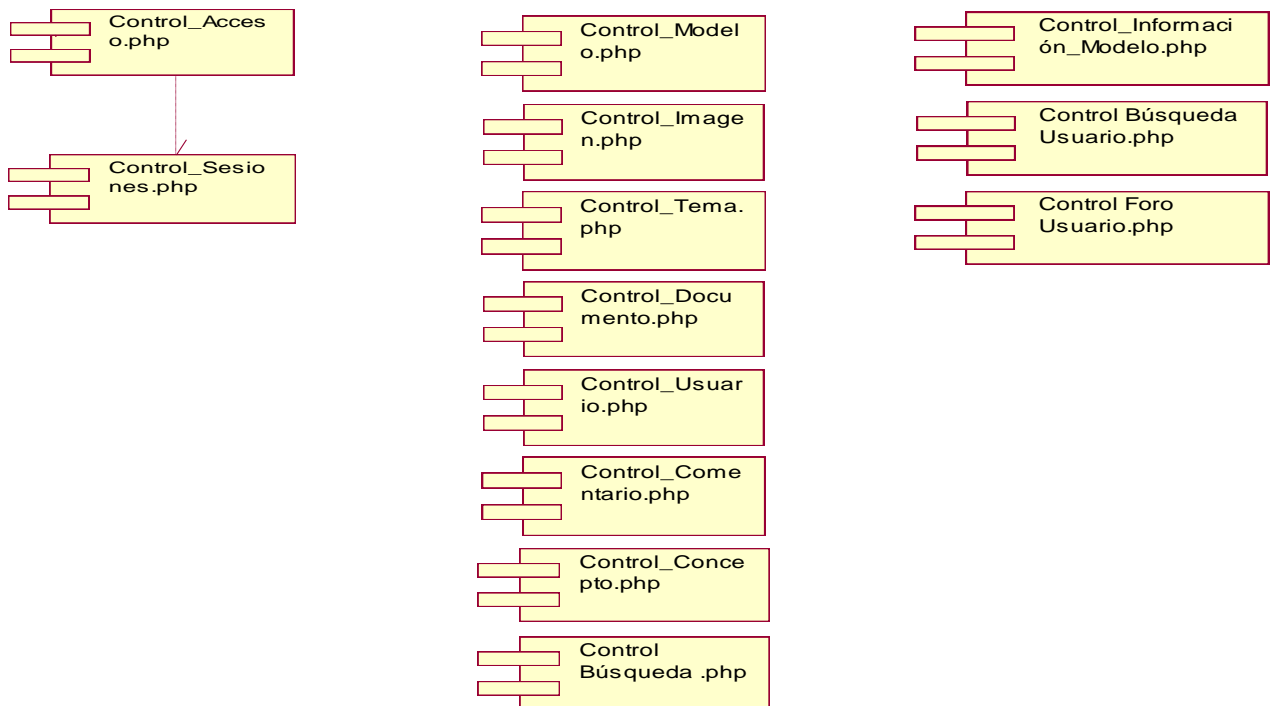


Fig.22. Diagrama de componentes de los paquetes Controladores.

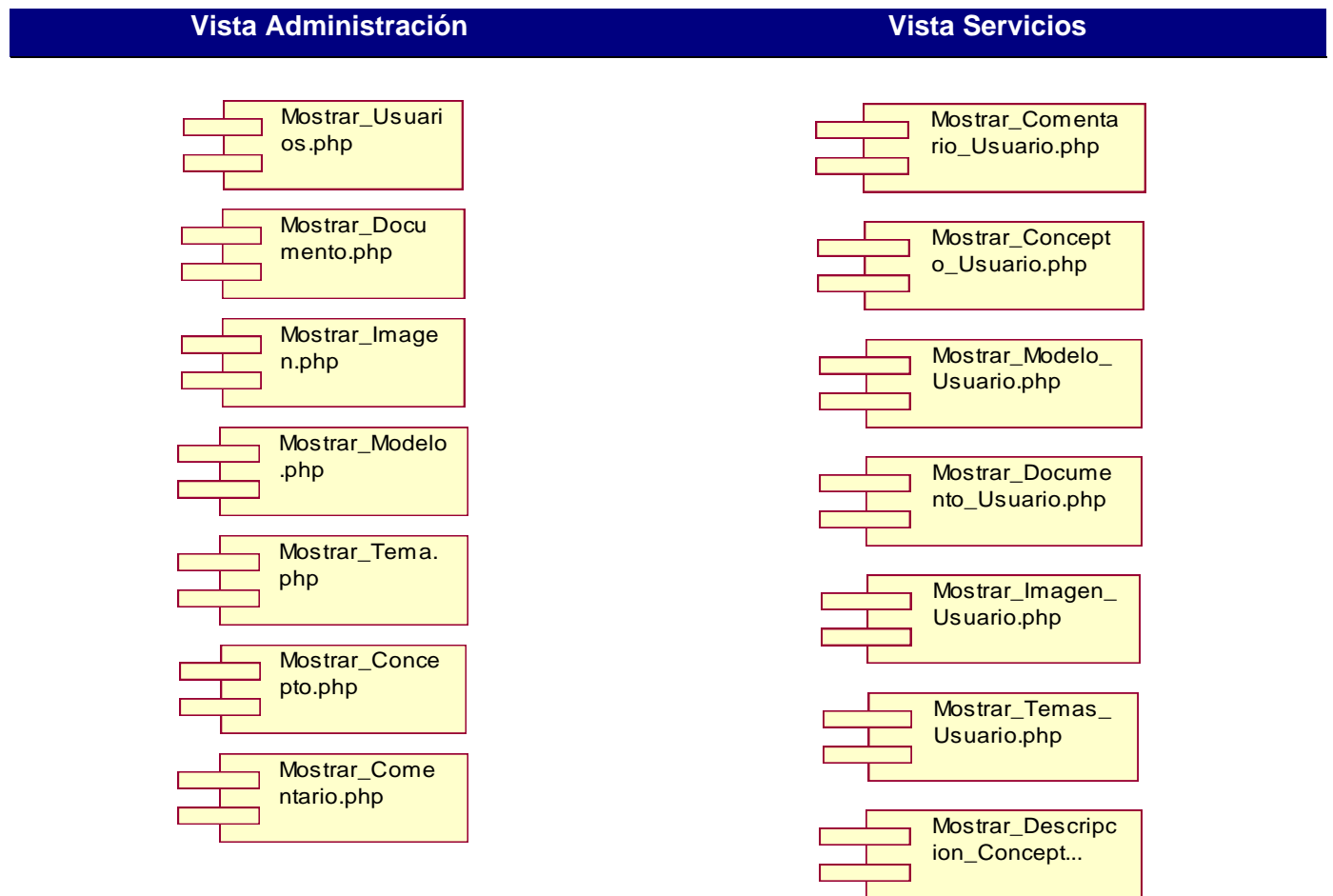


Fig.23. Diagrama de componentes del paquete Vista.

#### 4.1.2. Modelo de Despliegue

En el siguiente diagrama se muestra la configuración de hardware del sistema, así como los nodos físicos por los que está compuesto y sus respectivas conexiones.

El sistema está constituido siguiendo la arquitectura cliente servidor, donde se interactúa con el usuario a través de una estación de trabajo cliente desde donde se solicita una determinada petición al servidor. Del lado del servidor estará funcionando el servidor de base de datos MySQL o PostgreSQL así como el servidor WEB Apache. La comunicación con el nodo del cliente que es donde se van a mostrar las interfaces para interactuar con el sistema se va a realizar por medio del protocolo HTTP.

## Modelo de despliegue

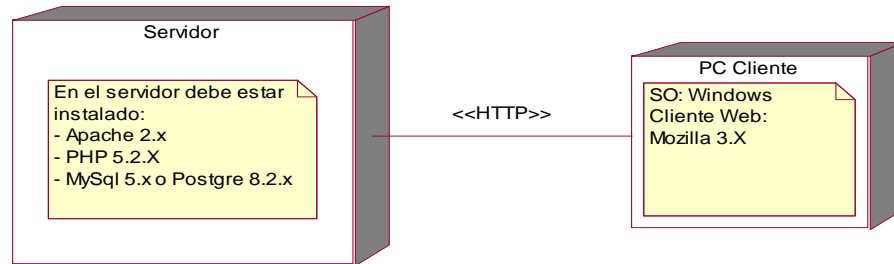


Fig.24. Modelo de Despliegue

### 4.1.3. Descripción del método “Visualización de modelos 3D”

A continuación se describen los métodos fundamentales para lograr la interacción entre el usuario y los modelos representados en la escena 3D. Esto se realiza en la sección 3D de la aplicación.

## Visualización de modelos 3D

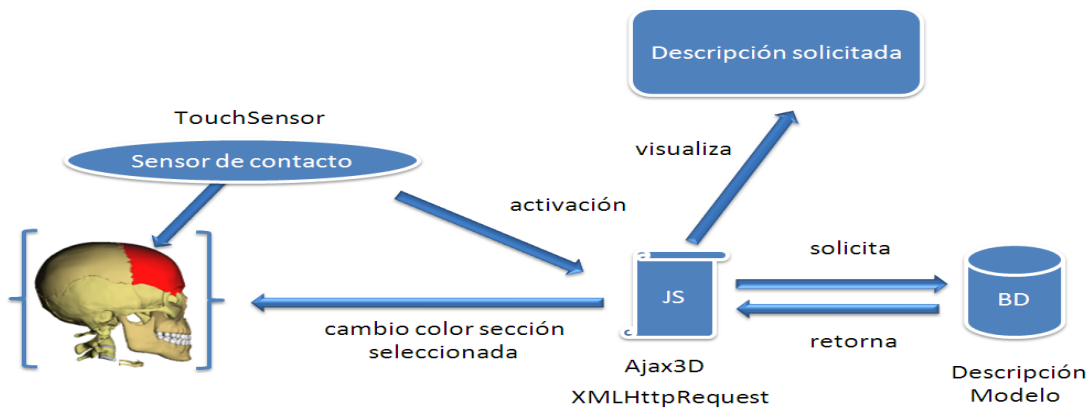


Fig.25. Visualización de modelos 3D

1. Se define un grupo dentro del modelo .x3d al que se le va a asignar un sensor de contacto para interactuar con el mouse.

```
<Group DEF='GrupoModelo'>
```

2. Se definen los objetos que van a estar contenidos dentro de ese grupo.

```
<Shape DEF='Modelo'>
```

3. Se define el material que va a tener el objeto.

```
<Appearance
  containerField='appearance'>
  <Material DEF='TexturaModelo'
    containerField='material'
    ambientIntensity='.2'
    shininess='.2'
    diffuseColor='1 0 0'>
</Appearance>
```

4. En caso de ya estar definida una textura dentro de la escena solo hay que especificar cual se le desea asignar al objeto.

```
<Material containerField='material' USE='TexturaModelo' />
```

5. Se le asigna al grupo un sensor de contacto.

```
<TouchSensor DEF='SensorModelo' containerField='children' />
```

6. Después de cargar el modelo X3D en la aplicación WEB por medio de la utilización del Ajax3D es que se realiza el cambio de color.

- Se carga el modelo en la página WEB.
- Se detecta cualquier interacción en el navegador.
- En dependencia de la zona del modelo donde se realizó el clic se pregunta qué sensor de contacto tiene asignado ese objeto, en correspondencia del que sea, se envía el nombre de la textura que se le definió al material dentro de la escena X3D para poder capturar así el nodo que lleva ese nombre.
- Después de haber capturado este objeto para así decirlo podemos acceder a sus propiedades y modificarlas. En este caso transformamos el **diffuseColor** que está compuesto por las componentes **R G B** y las modificamos a nuestra conveniencia para lograr el efecto de color en el modelo.

```
var material = null;
material = context.getNode("TexturaModelo");
material.diffuseColor.g = .1;
```

- De forma asíncrona se envía el nombre que tiene esa zona en específica del modelo para obtener la descripción que se encuentra en la base de datos.
- En caso que se seleccione otra parte del modelo que está visualizado en la escena se le restaura el color inicial que contaba el material de la zona anteriormente seleccionada y se realiza el mismo proceso antes descrito.

### Consideraciones finales del capítulo

En el presente capítulo se elaboraron diagramas del flujo de trabajo de implementación que permiten entender de una manera concisa la construcción de la aplicación. También se brindó de forma breve y comprensible la descripción del método fundamental de interacción entre el usuario y los modelos anatómicos 3D y los pasos necesarios para lograr obtener el resultado esperado.

## CONCLUSIONES

Con el desarrollo del sistema para la visualización de imágenes y modelos 3D médicos en la WEB se hizo un estudio de las tendencias actuales de la representación tridimensional sobre esta plataforma, lo que arrojó la selección y empleo del estándar extensible X3D lográndose la visualización de los modelos anatómicos. Partiendo del estudio de aplicaciones similares se definieron e implementaron las principales funcionalidades que debe brindar nuestro sistema para lograr la aceptación de los usuarios.

Seleccionando las técnicas y los lenguajes de programación adecuados posibilitamos que nuestra aplicación WEB garantice el acceso, visualización e interacción de los modelos representados en la escena 3D, contribuyendo de esta forma al aprendizaje de la anatomía del cuerpo humano. Con la realización de este trabajo, se facilita al sistema de enseñanza cubano un nuevo recurso académico. Constituyendo un gran apoyo para el personal docente de la facultad de medicina, en la explicación y representación de una manera más amena y eficiente de estructuras y conceptos de la anatomía del cuerpo humano, aumentando a su vez la disponibilidad y acceso a estos recursos.

Con la implementación de este sistema se avanza en el empleo y explotación de los recursos informáticos que nuestro gobierno revolucionario ha puesto a nuestra disposición. También se les brinda a las estudiantes una nueva forma de obtener los conocimientos de una manera interactiva adentrándolos en el mundo de la realidad virtual.

## **RECOMENDACIONES**

Se recomienda ampliar las funcionalidades de la aplicación incorporándoles evaluadores online para poder medir los avances de los estudiantes.

Se recomienda con un equipo de diseñadores de nuestra universidad diseñar todos los modelos de la aplicación para lograr una mayor calidad de visualización.

Se recomienda que la aplicación brinde la funcionalidad de visualizar las imágenes DICOM con los algoritmos que se desarrollaron en el proyecto Vismedic para que sirva como herramienta diagnóstica de estudios reales en pacientes.



---

## BIBLIOGRAFÍA

1. *Museo Virtual de Historia Precolombina de México para jóvenes y adultos con atraso educacional*. **Castañeda Roldán, Carolina and Armando López, Sergio**. 2000.
2. **Stephen Matsuba, Bernie Roehl**. *Using VRML*. s.l. : QUE Corporation, 1996.
3. **Levis, Diego**. Realidad Virtual y educación. [Online] 1997.  
[http://www.diegolevis.com.ar/secciones/Articulos/master\\_eduvirtual.pdf](http://www.diegolevis.com.ar/secciones/Articulos/master_eduvirtual.pdf).
4. **García Ruiz, Miguel Ángel**. *La tecnología de realidad virtual aplicada a la educación: el futuro ya está aquí*. Universidad de Colima : s.n., 2004. 6.
5. **Cardozo, Hugo Javier**. Realidad Virtual. UCA TAI : s.n., 2004.
6. **Heras Lara, Lizbeth and Rivera Ojeda, María Teresa**. *Gráficos 3D en la WEB*. Publicación Mensual. México : s.n., 2008. p. 74. Universidad Nacional Autónoma de México.
7. *Visible Body*. [Online] 2010. <http://www.visiblebody.com/>.
8. *Visión Médica Virtual*. [Online] 2010. <http://www.visionmedicavirtual.com/es>.
9. *Universidad Virtual de salud*. [Online] 2010. <http://www.uvs.sld.cu/>.
10. Aplicaciones WEB. [Online] <http://www.hooping.net/glossary/aplicaciones-WEB-146.aspx>.
11. Benefits Of WEB Based Applications. [Online] 2010.  
<http://www.dbnetsolutions.co.uk/Articles/BenefitsOfWEBBasedApplications.aspx>.
12. **Trubac, Andrew**. HTML. [Online] 2001.  
[http://searchsoa.techtarget.com/sDefinition/0,,sid26\\_gci212286,00.html](http://searchsoa.techtarget.com/sDefinition/0,,sid26_gci212286,00.html).
13. **Bray, Tim, et al**. Extensible Markup Language (XML) 1.0. [Online] 2008.  
<http://www.w3.org/TR/REC-xml/>.
14. **Ortiz Fernandez, Juan Alberto**. Análisis de la tendencia WEB 2.0 y su relación-cliente empresa. 2009.
15. **Bradenbaugb, Jerry**. *Aplicaciones JavaScript* . s.l. : Artes Gráficas Guemo, 2000. p. 532.
16. *The Advantages of php*. **Soto, Celeste**. 2006.
17. **James Garrett, Jesse**. Ajax: Un nuevo acercamiento a las aplicaciones WEB. [Online] junio 11, 2005. [http://www.willydev.net/Descargas/WillyDev\\_AJAX.pdf](http://www.willydev.net/Descargas/WillyDev_AJAX.pdf).
18. **González Suárez, Maikel**. Introducción a la manipulación del DOM mediante Javascript. [Online] febrero 27, 2007. [www.maestrosdelWEB.com/editorial/dom/](http://www.maestrosdelWEB.com/editorial/dom/).
19. **Domínguez, Danilo**. Introducción a las base de datos. [Online] 2007.
20. **Kofler, Michael**. *The Definitive Guide to MySQL*. 2004.
21. **Brookins, Andrew and Holloway, Michael**. *Practical PostgreSQ*. s.l. : Command Prompt Inc., 2002.

22. About PostgreSQL. [Online] 2007. <http://www.postgresql.org/about/>.
23. **Triana Morín, Arley**. PDO (PHP Data Objects). Capa de Abstracción de acceso a Bases de Datos (Primera parte). [Online] noviembre 7, 2009. <http://desarrolladorsenior.blogspot.com/2009/10/pdo-php-data-objects-es-una-extension.html>
24. **Martin Maldonado, Daniel**. Apache, el servidor WEB más reconocido,. [Online] abril 20, 2008. [www.aplicacionesempresariales.com/apache-el-servidor-WEB-mas-reconocido.html](http://www.aplicacionesempresariales.com/apache-el-servidor-WEB-mas-reconocido.html) .
25. **Sánchez, Jordi**. ¿Qué es un framework? [Online] 2006.
26. ¿Que es el jQuery? [Online] 2008. <http://www.javascriptya.com.ar/jquery/temarios/descripcion.php?cod=57&punto=1&inicio=0>.
27. **Reynoso, Billy**. Introducción a la Arquitectura de Software. 2005. ArquitecturaCliente/Servidor. [Online] diciembre 27, 2009. <http://ccia.ei.uvigo.es/docencia/SCS/0910/transparencias/Tema1.pdf>..
28. **Pavón, Juan**. *Estructura de las Aplicaciones Orientadas a Objetos El patrón Modelo-Vista-Controlador (MVC)*. 2009. p. 12.
29. **I., Jacobson, G., Booch and Addison Wesley, Rumbaugh J**. El Proceso Unificado de Desarrollo de Software. 2000.
30. **Muneta, Martínez, et al**. Últimas Tendencias en Gráficos WEB3D para Internet. 2002.
31. **Macías, Jiménez**. Escenarios virtuales WEB3D: Simulación con VRML, JAVA3D y X3D. 2004.
32. WEB 3D, flash, los frameworks más usados, sus pros y contras. [Online] agosto 21, 2009. <http://www.riahispano.org/?q=node/3>.
33. **Edwards, Lin**. Superior 3D Graphics for the WEB a Step Closer. septiembre 22, 2009.
34. Google O3D, API para la creación de aplicaciones interactivas 3D. [Online] junio 6, 2009. <http://www.desarrolloWEB.com/actualidad/google-o3d-api-aplicaciones-interactivas-3d-1749.html>.
35. **Olmedo Rodríguez, H., Escudero Mancebo, D. and Cardeñoso Payo, V**. 3D en las Rich Internet Applications: comparativa de opciones tecnológicas. [Online] 2008.
36. Sandy is a Flash 3D engine, available in 3 versions: AS2, AS3 and haXe. [Online] 2009. <http://www.flashsandy.org/>.
37. **Hilera, José R., Otón, Salvador and Martínez, Javier**. Aplicación de la realidad virtual en la enseñanza a través de internet. 2000.
38. **Brutzman, Don and Daly, Leonard**. *X3D: Extensible 3D Graphics for WEB authors*. San Francisco : Elsevier Inc, 2007. p. 471.

39. Rational Rose Enterprise. [Online] 2007.  
<http://www.rational.com.ar/herramientas/roseenterprise.html>.
40. **Angel Alvarez, Miguel.** Un IDE para el desarrollo de aplicaciones WEB, enfocado en Ajax y la WEB 2.0. [Online] diciembre 17, 2007. <http://www.desarrolloWEB.com/articulos/aptana-studio.html>.
41. **Bruce.** 3D Modeling : Simple Boolean Extraction in Vivaty Studio. [Online] abril 24, 2009.
42. **Cardona, Jordi R.** X3D-Edit Free Software For X3D Code Edition And Worldbuilding. mayo 13, 2008.
43. **Torossi, Gustavo.** El Proceso Unificado de Desarrollo de Software. 2005.
44. **Fowler, Martin, et al.** *Patterns of Enterprise Application Architecture*. s.l. : Addison Wesley, 2002. p. 560.
45. Core J2EE Patterns - Data Access Object. [Online] 2010.  
<http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html>.

## ANEXOS

### 1. Anexo I

#### Descripciones textuales de los casos de uso del sistema.

##### Caso de Uso “Autenticarse”

Caso de Uso:	Autenticarse	
Actores:	Usuario del sistema	
Resumen:	El caso de uso se inicia cuando el usuario del sistema intenta acceder a la aplicación y en dependencia del rol que este posea se le dará el acceso a determinadas funcionalidades y secciones del sistema. El caso de uso finaliza cuando el usuario accede al sistema.	
Precondiciones:	El sistema debe estar funcionando y activo de forma tal que ofrezca la interfaz para la autenticación, además de estar instalado correctamente para poder realizar la verificación y validación necesaria.	
Referencias	RF1	
Prioridad	Crítico	
<b>Flujo Normal de Eventos</b>		
	<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1- Accede a la aplicación.	2- Muestra la página inicial con la opción para autenticarse en el sistema.
	3- Introduce los datos solicitados. <ul style="list-style-type: none"> <li>✓ Usuario.</li> <li>✓ Contraseña.</li> </ul>	4- Verifica la existencia del usuario.
		5- Verifica que la contraseña introducida coincida con la almacenada en la base de datos.
		6- Carga el sistema con los menús, las opciones y funcionalidades a las que el usuario tiene acceso y permisos.
<i>Prototipo de Interfaz</i>		
<b>Flujos Alternos</b>		
	<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>

	4.1 No encuentra al usuario e informa de que el usuario introducido es incorrecto.
	5.1 No encuentra igualdad en las contraseñas y muestra un mensaje donde le dice que la misma es incorrecta.
<i>Prototipo de Interfaz</i>	
Poscondiciones	1- Se le permite el acceso al usuario. 2- Quedan habilitadas las opciones a las que el usuario tiene acceso.

### Caso de Uso “Registrarse”

Caso de Uso:	Registrarse
Actores:	Usuario del sistema
Resumen:	El caso de uso se inicia cuando el usuario del sistema accede por primera vez a la aplicación, el sistema le muestra la opción de registrarse como usuario de la aplicación, el sistema le muestra los campos requeridos. El caso de uso termina cuando se le muestra un mensaje de registro satisfactorio.
Precondiciones:	El usuario accede por primera vez a la aplicación, por lo que no se encuentra registrado como usuario de la misma.
Referencias	RF2
Prioridad	Crítico.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- Accede a la aplicación.	2- Muestra la página inicial con la opción para registrarse en el sistema.
3- Selecciona la opción de registrarse.	4- Muestra una interfaz con los campos necesarios para registrarse como usuario de la aplicación.
5- Introduce los datos requeridos. ✓ Nombre	6- Verifica la existencia del usuario

<ul style="list-style-type: none"> <li>✓ Apellidos</li> <li>✓ Año académico</li> <li>✓ Usuario</li> <li>✓ Contraseña</li> </ul>	
	7- Muestra un mensaje haciéndole saber al usuario que se ha registrado correctamente.
<i>Prototipo de Interfaz</i>	
<b>Flujos Alternos</b>	
Acción del Actor	Respuesta del Sistema
	6.1 Encuentra que el usuario ya está en el sistema e informa su existencia.
<i>Prototipo de Interfaz</i>	
Poscondiciones	1- Se le muestra al usuario un mensaje informándole de que el registro ha sido satisfactorio.

### Caso de Uso “Mostrar sistemas del cuerpo”

Caso de Uso:	Mostrar sistemas del cuerpo
Actores:	Estudiante
Resumen:	El caso de uso se inicia cuando el estudiante selecciona la opción mostrar sistemas del cuerpo, pudiendo seleccionar en esta interfaz los distintos sistemas del cuerpo humano (nervioso, muscular, urinario, etc.) además también puede ver algunos de los mismos específicamente en la cabeza. El caso de uso termina cuando se le muestra al estudiante el sistema deseado.
Precondiciones:	El estudiante tiene que estar autenticado.
Referencias	RF3, RF3.1, RF3.2
Prioridad	Crítico
<b>Flujo Normal de Eventos</b>	
Acción del Actor	Respuesta del Sistema
1- Accede a la sección de selección de los sistemas del cuerpo humano.	2- El sistema ejecuta alguna de las siguientes acciones. a) Para observar los sistemas en el cuerpo

	humano ir a la sección “Sistemas en el cuerpo humano” b) Para observar los sistemas en la cabeza humana ir a la sección “Sistemas en la cabeza”
<b>Sección: “Sistemas en el cuerpo humano”</b>	
Acción del Actor	Respuesta del Sistema
1- Selecciona uno de los sistemas <ul style="list-style-type: none"> <li>✓ Circulatorio</li> <li>✓ Respiratorio</li> <li>✓ Urinario</li> <li>✓ Nervioso</li> <li>✓ Muscular Anterior</li> <li>✓ Muscular Posterior</li> <li>✓ Digestivo</li> <li>✓ Esquelético</li> <li>✓ Inmune</li> <li>✓ Reproductor Masculino</li> <li>✓ Reproductor Femenino</li> </ul>	2- Muestra la figura el cuerpo humano con el sistema seleccionado conjuntamente con la descripción del mismo.
<i>Prototipo de Interfaz</i>	
<b>Sección: “Sistemas en la cabeza”</b>	
Acción del Actor	Respuesta del Sistema
1- Selecciona la vista de la cabeza deseada. <ul style="list-style-type: none"> <li>✓ Frontal</li> <li>✓ Lateral</li> </ul>	2- Muestra las opciones para seleccionar los sistemas.
3- Selecciona todos los sistemas que desee observar en la figura de la cabeza. <ul style="list-style-type: none"> <li>✓ Circulatorio</li> <li>✓ Muscular</li> <li>✓ Nervioso</li> </ul>	4- Muestra la figura de la cabeza humana con todos los sistemas seleccionados por el estudiante.

✓ Esquelético	
<i>Prototipo de Interfaz</i>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Poscondiciones	1- Se muestra la imagen del sistema en el cuerpo humano. 2- Se muestra la imagen del sistema en la cabeza humana.

### Caso de Uso “Mostrar imágenes médicas”

Caso de Uso:	Mostrar imágenes médicas
Actores:	Estudiante
Resumen:	El caso de uso se inicia cuando el estudiante accede a la interfaz donde puede seleccionar la parte de la anatomía esquelética que quiere observar, además también puede observar la imagen de radiológica correspondiente a la parte del cuerpo deseada y también puede observar la imagen del corte transversal del cuerpo humano en el nivel deseado. El caso de uso termina cuando se muestra la imagen seleccionada por el usuario.
Precondiciones:	El estudiante tiene que estar autenticado.
Referencias	RF4, RF4.1, RF4.2, RF4.3
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1- Accede a la interfaz.	2- El sistema ejecuta alguna de las siguientes acciones. a) Para observar la anatomía esquelética ir a la sección “Mostrar imágenes anatomía esquelética” b) Para observar imágenes de anatomía radiológica ir a la sección “Mostrar imágenes de anatomía radiológica” c) Para observar imágenes del corte



	transversal del cuerpo ir a la sección “Mostrar imágenes del corte transversal del cuerpo humano”
<b>Sección: “Mostrar imágenes anatomía esquelética”</b>	
Acción del Actor	Respuesta del Sistema
1- Accede a la interfaz	2- Muestra una lista de las distintas partes del esqueleto.
3- Selecciona la parte del esqueleto que desea observar.	3- Muestra la imagen de la parte seleccionada resaltada en color rojo.
<i>Prototipo de Interfaz</i>	
<b>Sección :“Mostrar imágenes de anatomía radiológica”</b>	
Acción del Actor	Respuesta del Sistema
1- Accede a la interfaz.	2- Muestra una lista de partes del cuerpo humano.
3- Selecciona la parte del cuerpo deseada.	4- Muestra la parte seleccionada y muestra en una ventana su correspondiente imagen radiológica.
<i>Prototipo de Interfaz</i>	
<b>Sección: “Mostrar imágenes del corte transversal del cuerpo humano”</b>	
Acción del Actor	Respuesta del Sistema
1- Accede a la interfaz.	2- Muestra una barra de desplazamiento que indica el nivel del corte actual del cuerpo humano.
3- Desplaza la barra que indica el nivel del cuerpo humano del cual quiere obtener el corte transversal.	4- En dependencia del nivel seleccionado muestra la imagen del corte transversal del cuerpo correspondiente.
<i>Prototipo de Interfaz</i>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema

Poscondiciones	<p>1- Muestra la parte del esqueleto seleccionada por el usuario resaltando su color dentro de la imagen</p> <p>2- Se muestra la parte u órgano unido a su imagen radiológica seleccionada por el estudiante.</p> <p>3- Es mostrado el corte transversal correspondiente al nivel seleccionado por el estudiante.</p>

### Caso de Uso “Insertar comentario en el foro”

Caso de Uso:	Insertar comentario en el foro
Actores:	Usuario Sistema
Resumen:	El caso de uso se inicia cuando el usuario del sistema accede a la sección del foro y desea insertar un comentario en un tema determinado. El caso de uso finaliza cuando se inserta en el foro el comentario.
Precondiciones:	El estudiante tiene que estar autenticado.
Referencias	RF6
Prioridad	Crítico
<b>Flujo Normal de Eventos</b>	
Acción del Actor	Respuesta del Sistema
1- Accede a la sección del foro.	2- Muestra el listado de temas disponibles.
3- Selecciona el tema deseado.	4- Se muestra una interfaz donde aparece el campo insertar comentario.
5- Introduce el comentario deseado.	6- Inserta el comentario en el tema en que se encuentra el usuario del sistema.
<i>Prototipo de Interfaz</i>	
<b>Flujos Alternos</b>	
Acción del Actor	Respuesta del Sistema
<i>Prototipo de Interfaz</i>	
Poscondiciones	Se inserta en un tema del foro el comentario deseado.

### Caso de Uso “Gestionar Tema”

Caso de Uso:	Gestionar Tema
--------------	----------------

Actores:	Administrador
Resumen:	El caso de uso se inicia cuando el administrador desea realizar alguna acción sobre un tema del foro del sistema. Estas acciones pueden ser adicionar tema, eliminar tema, publicar tema, despublicar tema. Si es adicionar tema se introduce el tema deseado, si es eliminar tema se elimina el tema seleccionado, si es publicar tema se publica el tema seleccionado y si es despublicar tema se despublica el mismo. El caso de uso termina cuando se guardan los cambios efectuados.
Precondiciones:	Autenticación del usuario como administrador
Referencias	RF7, RF7.1, RF7.2, RF7.3, RF7.4
Prioridad	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El administrador solicita administrar los datos de un tema del foro.	2- El sistema ejecuta alguna de las siguientes acciones. a) Para adicionar un tema ir a la sección "Adicionar tema" b) Para eliminar un tema ir a la sección "Eliminar tema" c) Para publicar un tema ir a la sección "Publicar tema" d) Para despublicar tema ir a la sección "Despublicar tema".
<b>Sección: "Adicionar tema"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- Selecciona la opción adicionar tema	2- Muestra un campo para introducir el nombre del tema a insertar.
3- Introduce el nombre del tema y acepta.	4- Se visualiza el tema insertado.
<i>Prototipo de Interfaz</i>	
<b>Sección: "Eliminar tema"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- Selecciona la opción eliminar tema de tema.	2- Muestra un mensaje indicando si está

	seguro de que desea eliminar ese tema.
3- Confirma que desea eliminar el tema	4- Elimina el tema y actualiza la interfaz
<i>Prototipo de Interfaz</i>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3.1 Decide que no va a eliminarlo y cierra la ventana de mensaje.	
<b>Sección: “Publicar tema”</b>	
Acción del Actor	Respuesta del Sistema
1- Selecciona la opción publicar tema de tema.	2- Muestra un mensaje indicando si está seguro que desea publicar ese tema.
3- Confirma que desea publicar el tema	4- Publica el tema y actualiza la interfaz
<i>Prototipo de Interfaz</i>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3.1 Decide que no va a publicar y cierra la ventana de mensaje.	
<b>Sección: “Despublicar tema”</b>	
Acción del Actor	Respuesta del Sistema
1- Selecciona la opción despublicar tema de tema.	2- Muestra un mensaje indicando si está seguro de que desea despublicar ese tema.
3- Confirma que desea despublicar el tema	4- Despublica el tema y actualiza la interfaz
<i>Prototipo de Interfaz</i>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3.1 Decide que no va a despublicar y cierra la ventana de mensaje.	
Poscondiciones	1- Se adiciona un tema. 2- Se elimina un tema del foro. 3- Se publica un tema.

	4- Se despublica un tema.
--	---------------------------

### Caso de Uso “Gestionar Comentario”

Caso de Uso:	Gestionar Comentario
Actores:	Administrador
Resumen:	El caso de uso se inicia cuando el administrador desea realizar alguna acción sobre un comentario del foro del sistema. Estas acciones pueden ser publicar comentario, despublicar comentario y eliminar comentario. Si es publicar comentario se publica el comentario seleccionado, si es despublicar comentario se despublica el mismo y si es eliminar comentario se elimina el mismo. El caso de uso termina cuando se guardan los cambios efectuados.
Precondiciones:	Autenticación del usuario como administrador
Referencias	RF8, RF8.1, RF8.2 , RF8.3
Prioridad	Crítico

#### Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1- El administrador solicita administrar los datos de un comentario del foro.	2- El sistema ejecuta alguna de las siguientes acciones. a) Para publicar un comentario ir a la sección “Publicar comentario” b) Para despublicar un comentario ir a la sección “Despublicar comentario”. c) Para eliminar un comentario ir a la sección “Eliminar comentario”.

#### Sección: “Publicar comentario”

Acción del Actor	Respuesta del Sistema
1- Selecciona la opción publicar comentario de comentario.	2- Muestra un mensaje indicando si está seguro que desea publicar ese comentario.
3- Confirma que desea publicar el comentario	4- Publica el comentario y actualiza la interfaz

#### Prototipo de Interfaz

#### Flujos Alternos

Acción del Actor	Respuesta del Sistema
3.1 Decide que no va a publicarlo y cierra la ventana de mensaje.	
<b>Sección : “Despublicar comentario”</b>	
Acción del Actor	Respuesta del Sistema
1- Selecciona la opción despublicar comentario de comentario.	2- Muestra un mensaje indicando si está seguro de que desea despublicar ese comentario.
3- Confirma que desea despublicar el comentario	4- Despublica el comentario y actualiza la interfaz
<i>Prototipo de Interfaz</i>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3.1 Decide que no va a despublicarlo y cierra la ventana de mensaje.	
<b>Sección :“Eliminar comentario”</b>	
Acción del Actor	Respuesta del Sistema
1- Selecciona la opción eliminar de comentario.	2- Muestra un mensaje indicando si está seguro de que desea eliminar ese comentario.
3- Confirma que desea eliminar el comentario	4- Elimina el comentario y actualiza la interfaz
<i>Prototipo de Interfaz</i>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3.1 Decide que no va a eliminarlo y cierra la ventana de mensaje.	
Poscondiciones	1- Se publica un comentario. 2- Se despublica un comentario. 3- Se elimina un comentario de un tema.

**Caso de Uso “Gestionar modelo 3D”**

Caso de Uso:	Gestionar modelo 3D	
Actores:	Administrador	
Resumen:	El caso de uso se inicia cuando el administrador desea realizar alguna acción sobre alguno de los modelos anatómicos del sistema. Estas acciones pueden ser adicionar, publicar, despublicar y eliminar modelo. Si es adicionar se introducen los datos necesarios, si es publicar se publica el modelo, si es despublicar se despublica el modelo, si es eliminar se elimina el modelo deseado y si es cargar un modelo externo se realiza la visualización del mismo. El caso de uso termina cuando se guardan los cambios efectuados o se visualiza el modelo externo que se carga.	
Precondiciones:	Autenticación del usuario como administrador	
Referencias	RF9, RF9.1, RF9.2, RF9.3, RF9.4, RF9.5	
Prioridad	Crítico	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1- El administrador solicita administrar los datos de un modelo específico.	2- El sistema ejecuta alguna de las siguientes acciones. a) Para adicionar un modelo ir a la sección “Adicionar modelo” b) Para publicar un modelo ir a la sección “Publicar modelo” c) Para despublicar un modelo ir a la sección “Despublicar modelo” d) Para eliminar un modelo ir a la sección “Eliminar modelo” e) Para cargar un modelo externo a la aplicación ir a la sección “Cargar modelo 3D externo”	
<b>Sección: “Adicionar modelo”</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1- Selecciona la opción adicionar modelo	2- Muestra un formulario para realizar la	

	operación solicitada.
3- Introduce los datos siguientes y acepta la acción. <ul style="list-style-type: none"> <li>✓ Nombre</li> <li>✓ Descripción</li> <li>✓ Dirección del archivo</li> </ul>	4- Muestra un mensaje informando de que el modelo ha sido adicionado satisfactoriamente y muestra el modelo en la interfaz.
<i>Prototipo de Interfaz</i>	
<b>Sección: “Publicar modelo”</b>	
Acción del Actor	Respuesta del Sistema
1- Selecciona la opción publicar del modelo que desee.	2- Muestra un mensaje indicando si está seguro que desea publicar ese modelo.
3- Confirma que desea publicar el modelo.	4- Publica el modelo y actualiza la interfaz.
<i>Prototipo de Interfaz</i>	
<b>Flujos Alternos</b>	
Acción del Actor	Respuesta del Sistema
3.1 Decide que no va a publicarlo y cierra la ventana de mensaje.	
<b>Sección: “Despublicar modelo”</b>	
Acción del Actor	Respuesta del Sistema
1- Selecciona la opción despublicar del modelo que desee.	2- Muestra un mensaje indicando si está seguro de que desea despublicar ese modelo.
3- Confirma que desea despublicar el modelo.	4- Despublica el modelo y actualiza la interfaz.
<i>Prototipo de Interfaz</i>	
<b>Flujos Alternos</b>	
Acción del Actor	Respuesta del Sistema
3.1 Decide que no va a despublicarlo y cierra la ventana de mensaje.	
<b>Sección: “Eliminar modelo”</b>	
Acción del Actor	Respuesta del Sistema
1- Selecciona la opción eliminar del modelo	2- Muestra un mensaje indicando si está



deseado.	seguro de que desea eliminar ese modelo.
3- Confirma que desea eliminar el modelo.	4- Elimina el modelo y actualiza la interfaz.
<i>Prototipo de Interfaz</i>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3.1 Decide que no va a eliminarlo y cierra la ventana de mensaje.	
<b>Sección: “Cargar modelo 3D externo”</b>	
Acción del Actor	Respuesta del Sistema
1- Accede a la interfaz visualizar modelo externo.	2- Se muestra un campo que permite la búsqueda y carga del modelo.
3- Busca y carga el modelo.	4- Visualiza el modelo en la interfaz.
<i>Prototipo de Interfaz</i>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Poscondiciones	1- Se adiciona un modelo. 2- Se publica un modelo. 3- Se despublica un modelo. 4- Se elimina un modelo. 5- Se previsualiza un modelo.

### Caso de Uso “Gestionar imagen médica”

Caso de Uso:	Gestionar imagen médica
Actores:	Administrador
Resumen:	El caso de uso se inicia cuando el administrador desea realizar alguna acción sobre alguna de las imágenes anatómicas del sistema. Estas acciones pueden ser adicionar, publicar, despublicar y eliminar imagen. Si es adicionar se introducen los datos necesarios, si es publicar se publica la imagen deseada, si es despublicar se despublica la imagen y si es eliminar se elimina la imagen seleccionada. El caso de uso termina

	cuando se guardan los cambios efectuados.
Precondiciones:	Autenticación del usuario como administrador
Referencias	RF10, RF10.1, RF10.2, RF10.3, RF10.4
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
El administrador solicita administrar los datos de una imagen específica.	El sistema ejecuta alguna de las siguientes acciones. a) Para adicionar una imagen ir a la sección “Adicionar imagen” b) Para publicar una imagen ir a la sección “Publicar imagen” c) Para despublicar una imagen ir a la sección “Despublicar imagen” d) Para eliminar una imagen ir a la sección “Eliminar imagen”
Sección: “Adicionar imagen”	
Acción del Actor	Respuesta del Sistema
1- Selecciona la opción adicionar imagen	2- Muestra un formulario para realizar la operación solicitada.
3- Introduce los datos siguientes y acepta la acción. <ul style="list-style-type: none"> <li>✓ Nombre</li> <li>✓ Descripción</li> <li>✓ Dirección del archivo</li> </ul>	4- Muestra un mensaje informando de que la imagen ha sido adicionada satisfactoriamente y muestra la imagen en la interfaz.
Prototipo de Interfaz	
Sección: “Publicar imagen”	
Acción del Actor	Respuesta del Sistema
1- Selecciona la opción publicar de la imagen que desee publicar.	2- Muestra un mensaje indicando si está seguro de que desea publicar esa imagen.
3- Confirma que desea publicar la imagen.	4- Publica la imagen y actualiza la interfaz.

<i>Prototipo de Interfaz</i>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3.1 Decide que no va a publicar y cierra la ventana de mensaje.	
<b>Sección: “Despublicar imagen”</b>	
Acción del Actor	Respuesta del Sistema
1- Selecciona la opción despublicar de la imagen que desee despublicar.	2- Muestra un mensaje indicando si está seguro que desea despublicar esa imagen.
3- Confirma que desea despublicar la imagen.	4- Despublica la imagen y actualiza la interfaz.
<i>Prototipo de Interfaz</i>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3.1 Decide que no va a despublicar y cierra la ventana de mensaje.	
<b>Sección: “Eliminar imagen”</b>	
Acción del Actor	Respuesta del Sistema
1- Selecciona la opción eliminar de la imagen deseada.	2- Muestra un mensaje indicando si está seguro de que desea eliminar esa imagen.
3- Confirma que desea eliminar la imagen.	4- Elimina la imagen y actualiza la interfaz.
<i>Prototipo de Interfaz</i>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3.1 Decide que no va a eliminarla y cierra la ventana de mensaje.	
Poscondiciones	1- Se adiciona una imagen. 2- Se publica una imagen. 3- Se despublica una imagen. 4- Se elimina una imagen.

**Caso de Uso “Gestionar documento”**

Caso de Uso:	Gestionar documento
Actores:	Administrador
Resumen:	El caso de uso se inicia cuando el administrador desea realizar alguna acción sobre alguno de los documentos bibliográficos del sistema. Estas acciones pueden ser adicionar, publicar, despublicar y eliminar documento. Si es adicionar se introducen los datos necesarios, si es publicar se publica el documento deseado, si es despublicar se despublica el documento y si es eliminar se elimina el documento deseado. El caso de uso termina cuando se guardan los cambios efectuados.
Precondiciones:	Autenticación del usuario como administrador
Referencias	RF11, RF11.1 , RF11.2, RF11.3, RF11.4
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
El administrador solicita administrar los datos de un documento específico.	El sistema ejecuta alguna de las siguientes acciones. a) Para adicionar un documento ir a la sección “Adicionar documento” b) Para publicar un documento ir a la sección “Publicar documento” c) Para despublicar un documento ir a la sección “Despublicar documento” d) Para eliminar un documento ir a la sección “Eliminar documento”
Sección: “Adicionar documento”	
Acción del Actor	Respuesta del Sistema
1- Selecciona la opción adicionar documento	2- Muestra un formulario para realizar la operación solicitada.
3- Introduce los datos siguientes y acepta la acción.	4- Verifica que el documento no esté en el sistema.

✓ Título	
✓ Dirección del archivo	
	5- Muestra un mensaje informando que el documento ha sido adicionado satisfactoriamente y muestra el documento en la interfaz.
<i>Prototipo de Interfaz</i>	
<b>Sección: “Publicar documento”</b>	
Acción del Actor	Respuesta del Sistema
1- Selecciona la opción publicar del documento que desee publicar.	2- Muestra un mensaje indicando si está seguro que desea publicar ese documento.
3- Confirma que desea publicar el documento.	4- Publica el documento y actualiza la interfaz.
<i>Prototipo de Interfaz</i>	
<b>Flujos Alternos</b>	
Acción del Actor	Respuesta del Sistema
3.1 Decide que no va a publicar y cierra la ventana de mensaje.	
<b>Sección: “Despublicar documento”</b>	
Acción del Actor	Respuesta del Sistema
1- Selecciona la opción despublicar del documento que desee despublicar.	2- Muestra un mensaje indicando si está seguro que desea despublicar ese documento.
3- Confirma que desea despublicar el documento.	4- Despublica el documento y actualiza la interfaz.
<i>Prototipo de Interfaz</i>	
<b>Flujos Alternos</b>	
Acción del Actor	Respuesta del Sistema
3.1 Decide que no va a despublicar y cierra la ventana de mensaje.	
<b>Sección: “Eliminar documento”</b>	
Acción del Actor	Respuesta del Sistema

1- Selecciona la opción eliminar del documento deseado.	2- Muestra un mensaje indicando si está seguro que desea eliminar ese documento.
3- Confirma que desea eliminar el documento.	4- Elimina el documento y actualiza la interfaz.
<i>Prototipo de Interfaz</i>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3.1 Decide que no va a eliminarlo y cierra la ventana de mensaje.	
Poscondiciones	1- Se adiciona un documento. 2- Se publica un documento. 3- Se despublica un documento. 4- Se elimina un documento.

### Caso de Uso “Gestionar concepto médico”

Caso de Uso:	Gestionar concepto médico
Actores:	Administrador
Resumen:	El caso de uso se inicia cuando el administrador desea realizar alguna acción sobre alguno de los conceptos médicos del sistema. Estas acciones pueden ser adicionar, publicar, despublicar y eliminar concepto. Si es adicionar se introducen los datos necesarios, si es publicar se publica el concepto, si es despublicar se despublica el concepto y si es eliminar se elimina el concepto deseado. El caso de uso termina cuando se guardan los cambios efectuados.
Precondiciones:	Autenticación del usuario como administrador
Referencias	RF12, RF12.1, RF12.2, RF12.3, RF12.4
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1- El administrador solicita administrar los datos de un concepto específico.	2- El sistema ejecuta alguna de las siguientes acciones. a) Para adicionar un concepto ir a la sección “Adicionar concepto”

	<p>b) Para publicar un concepto ir a la sección “Publicar concepto”</p> <p>c) Para despublicar un concepto ir a la sección “Despublicar concepto”</p> <p>d) Para eliminar un concepto ir a la sección “Eliminar concepto”</p>
<b>Sección: “Adicionar concepto”</b>	
Acción del Actor	Respuesta del Sistema
1- Selecciona la opción adicionar concepto	2- Muestra un formulario para realizar la operación solicitada.
<p>3- Introduce los datos siguientes y acepta la acción.</p> <ul style="list-style-type: none"> <li>✓ Nombre</li> <li>✓ Descripción</li> </ul>	3- Verifica que el concepto no esté en el sistema.
	4- Muestra un mensaje informando que el concepto ha sido adicionado satisfactoriamente y muestra el concepto en la interfaz.
<i>Prototipo de Interfaz</i>	
Flujos Alternos	
Acción del Actor	Acción del Actor
	4.1 Encuentra que existe el concepto e informa mediante un mensaje su existencia.
<b>Sección: “Publicar concepto”</b>	
Acción del Actor	Respuesta del Sistema
1- Selecciona la opción publicar del concepto que desee publicar.	2- Muestra un mensaje indicando si está seguro que desea publicar ese concepto.
3- Confirma que desea publicar el concepto.	4- Publica el concepto y actualiza la interfaz.
<i>Prototipo de Interfaz</i>	

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3.1 Decide que no va a publicar y cierra la ventana de mensaje.	
Sección: “Despublicar concepto”	
Acción del Actor	Respuesta del Sistema
1- Selecciona la opción despublicar del concepto que desee despublicar.	2- Muestra un mensaje indicando si está seguro de que desea despublicar ese concepto.
3- Confirma que desea despublicar el documento.	4- Despublica el concepto y actualiza la interfaz.
Prototipo de Interfaz	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3.1 Decide que no va a despublicar y cierra la ventana de mensaje.	
Sección: “Eliminar concepto”	
Acción del Actor	Respuesta del Sistema
1- Selecciona la opción eliminar del concepto deseado.	2- Muestra un mensaje indicando si está seguro de que desea eliminar ese concepto.
3- Confirma que desea eliminar el concepto.	4- Elimina el concepto y actualiza la interfaz.
Prototipo de Interfaz	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3.1 Decide que no va a eliminarlo y cierra la ventana de mensaje.	
Poscondiciones	1- Se adiciona un concepto. 2- Se publica un concepto. 3- Se despublica un concepto. 4- Se elimina un concepto.



**Caso de Uso “Gestionar usuario”**

Caso de Uso:	Gestionar usuario
Actores:	Administrador
Resumen:	El caso de uso se inicia cuando el administrador accede a la interfaz de administración para realizar alguna operación sobre los usuarios existentes en el sistema. Las operaciones pueden ser de habilitación e inhabilitación de un usuario, o de eliminación cuando se elimina el mismo del sistema. El caso de uso termina cuando se guardan los cambios efectuados y se muestra el listado de usuarios actualizado.
Precondiciones:	Autenticación del usuario como administrador
Referencias	RF13, RF13.1, RF13.2, RF13.3
Prioridad	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El administrador solicita administrar los datos de un usuario específico.	2- El sistema ejecuta alguna de las siguientes acciones. a) Para habilitar un usuario ir a la sección “Habilitar usuario” b) Para inhabilitar un usuario ir a la sección “Inhabilitar usuario” c) Para eliminar un usuario ir a la sección “Eliminar usuario”.
<b>Sección: “Habilitar usuario”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- Selecciona la opción habilitar del usuario que desee habilitar.	2- Muestra un mensaje indicando si está seguro que desea habilitar ese usuario.
3- Confirma que desea habilitar el usuario.	4- Habilita el usuario y actualiza la interfaz.
<i>Prototipo de Interfaz</i>	
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
3.1 Decide que no va a habilitar el usuario y cierra la ventana de mensaje.	

<b>Sección: “Inhabilitar usuario”</b>	
Acción del Actor	Respuesta del Sistema
1- Selecciona la opción inhabilitar del usuario que desee inhabilitar.	2- Muestra un mensaje indicando si está seguro de que desea inhabilitar ese usuario.
3- Confirma que desea inhabilitar el usuario.	4- Inhabilita el usuario y actualiza la interfaz.
<i>Prototipo de Interfaz</i>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3.1 Decide que no va a inhabilitar el usuario y cierra la ventana de mensaje.	
<b>Sección : “Eliminar usuario”</b>	
Acción del Actor	Respuesta del Sistema
1- Selecciona la opción eliminar de un usuario	2- Muestra un mensaje indicando si está seguro que desea eliminar ese usuario.
3- Confirma que desea eliminar el usuario	4- Elimina el usuario y actualiza la interfaz
<i>Prototipo de Interfaz</i>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3.1 Decide que no va a eliminarlo y cierra la ventana de mensaje.	
Poscondiciones	1- Se habilita un usuario. 2- Se inhabilita un usuario. 3- Se elimina un usuario del sistema.

**Caso de Uso “Buscar recurso”**

Caso de Uso:	Buscar recurso
Actores:	Usuario del sistema
Resumen:	El caso de uso se inicia cuando el usuario del sistema selecciona la opción buscar recurso, se le muestra una interfaz en la cual puede realizar las búsquedas de imágenes, modelos, documentos y conceptos. El caso de uso termina cuando se retorna al usuario el resultado de su petición de

	búsqueda.
Precondiciones:	El usuario del sistema tiene que estar autenticado.
Referencias	RF14
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1- Selecciona la opción buscar.	2- Muestra una interfaz que permite escoger el recurso que se desee buscar.
3- Selecciona el recurso a buscar e introduce el criterio de búsqueda. <ul style="list-style-type: none"> <li>✓ Imagen – Modelo – Documento - Concepto</li> <li>✓ Nombre a buscar.</li> </ul>	4- Busca en la Base de Datos la existencia del recurso solicitado.
	5- Lista el conjunto de recursos con el nombre solicitado.
<i>Prototipo de Interfaz</i>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	4.1 No encuentra el recurso e informa su inexistencia.
<i>Prototipo de Interfaz</i>	
Poscondiciones	1- Se muestra al estudiante la existencia o inexistencia del recurso solicitado.

### Caso de Uso “Buscar usuario”

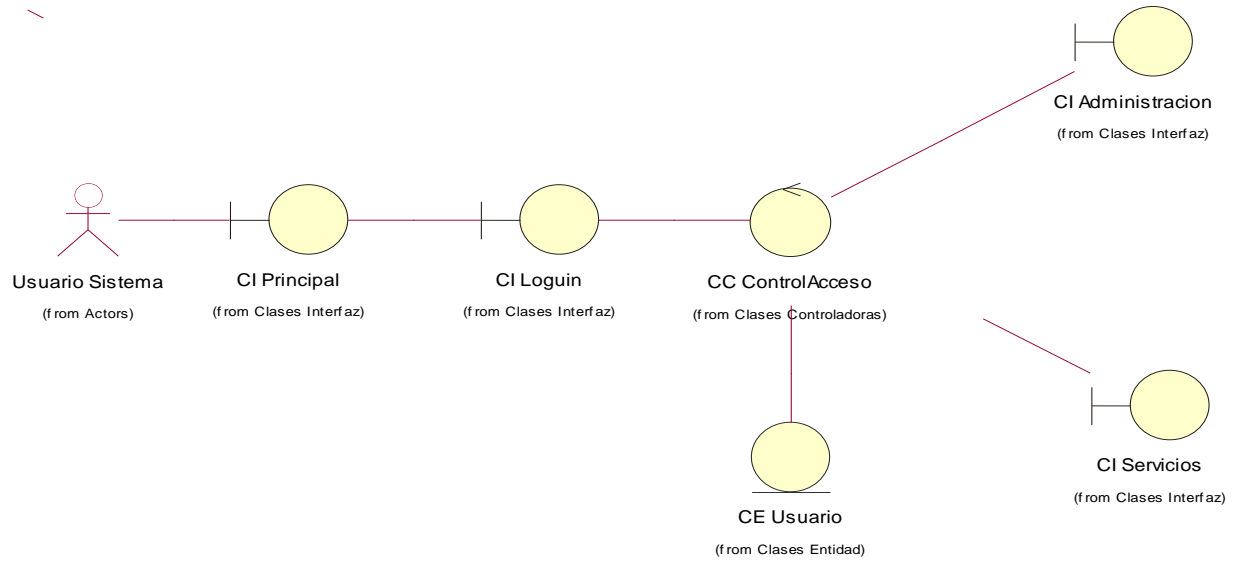
Caso de Uso:	Buscar usuario
Actores:	Administrador
Resumen:	El caso de uso se inicia cuando el administrador decide buscar un usuario en la interfaz de gestionar usuario. El caso de uso termina cuando se retorna al administrador el resultado de su petición de búsqueda.
Precondiciones:	El usuario tiene que estar autenticado como administrador.
Referencias	RF15

Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1- Introduce el criterio de búsqueda. ✓ Usuario a buscar.	2- Busca en la Base de Datos la existencia del usuario solicitado.
	3- Muestra el usuario solicitado.
<i>Prototipo de Interfaz</i>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	2.1 No encuentra el usuario e informa su inexistencia.
<i>Prototipo de Interfaz</i>	
Poscondiciones	1- Se muestra al administrador la existencia o inexistencia del usuario solicitado.

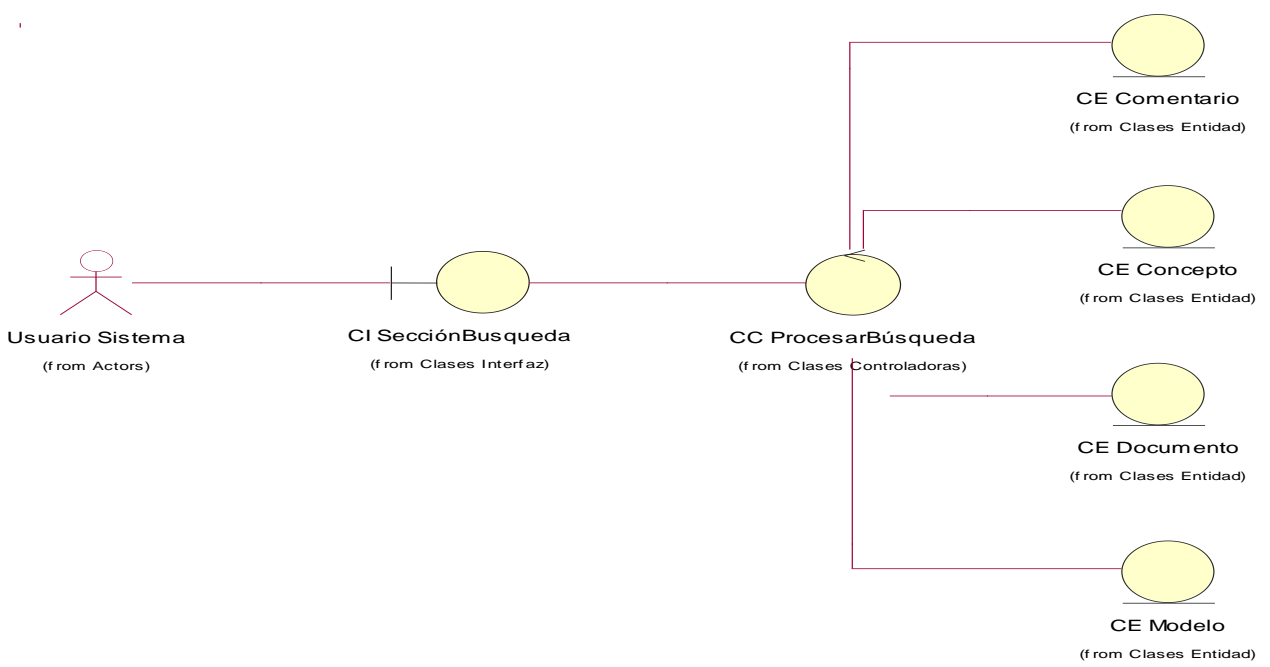
2. Anexo II

Diagrama de clases del diseño

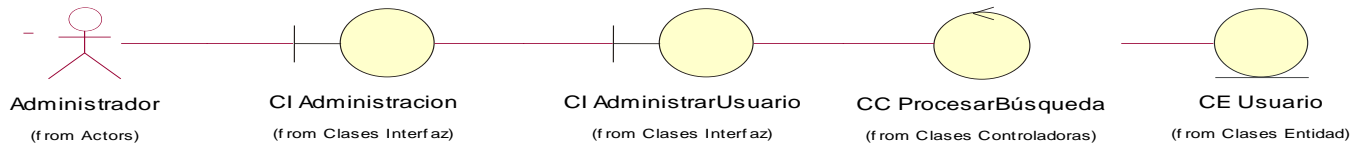
**Autenticar usuario**



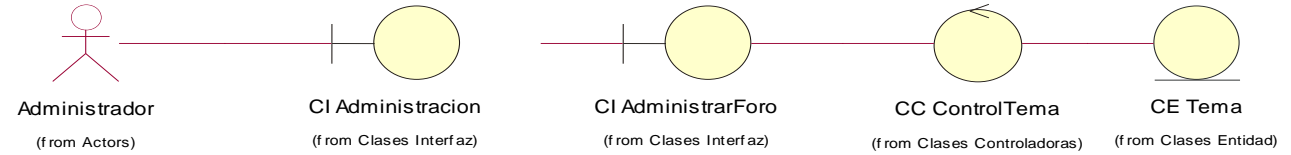
**Buscar recurso**



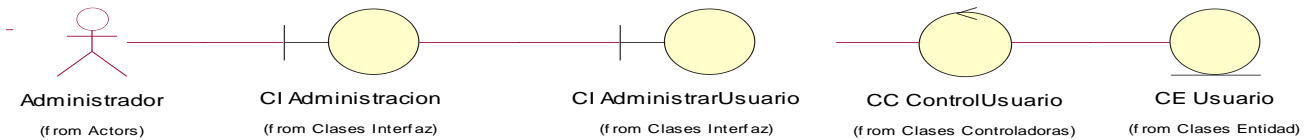
### Buscar usuario



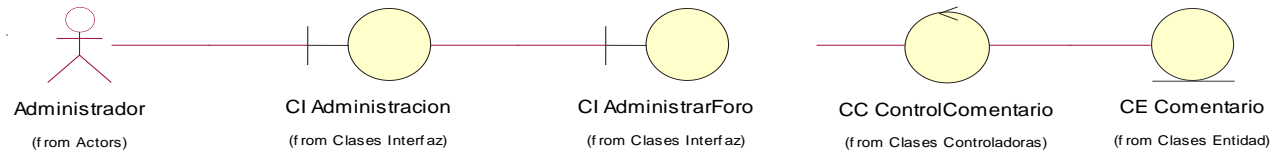
### Gestionar Tema



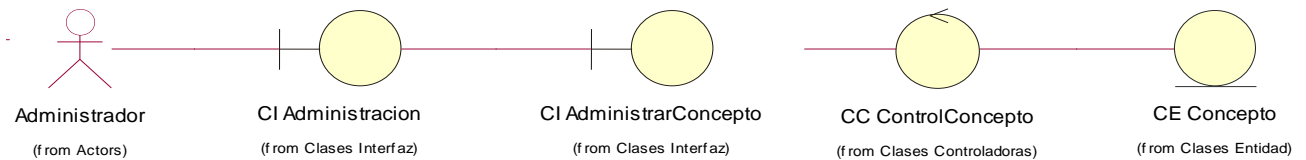
### Gestionar Usuario



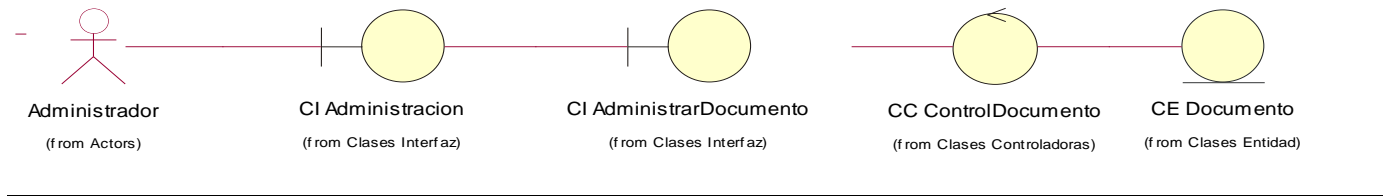
### Gestionar comentario



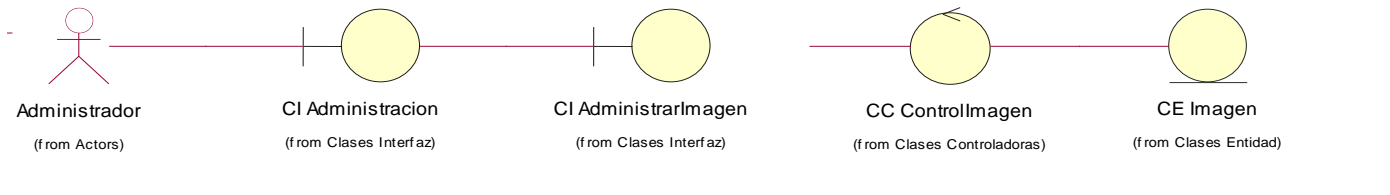
### Gestionar concepto médico



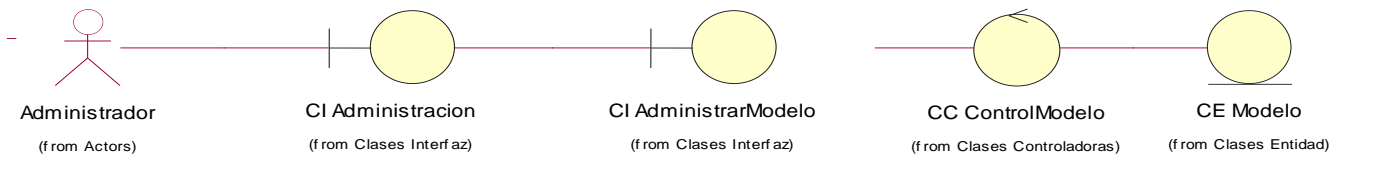
### Gestionar Documento



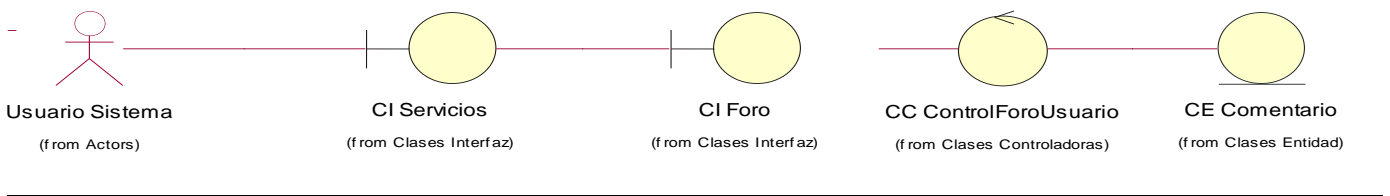
### Gestionar Imagen médica



### Gestionar Modelo 3D



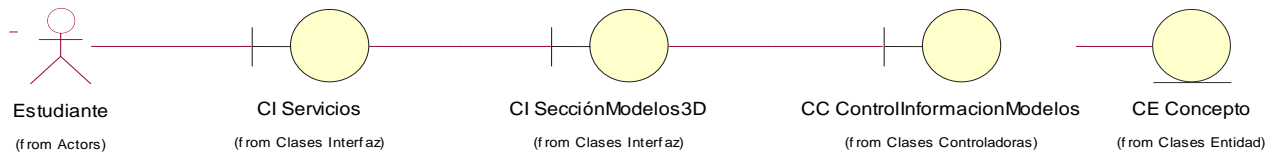
### Insertar comentario en el foro



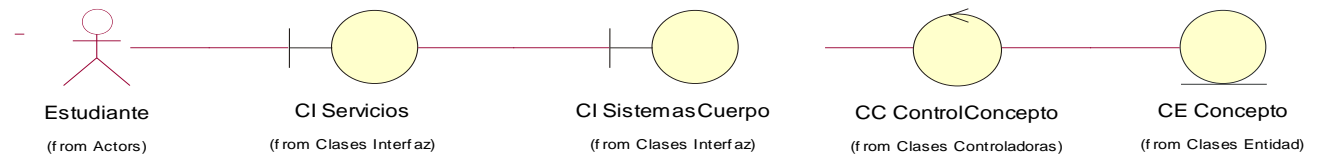
### Mostrar imágenes médicas



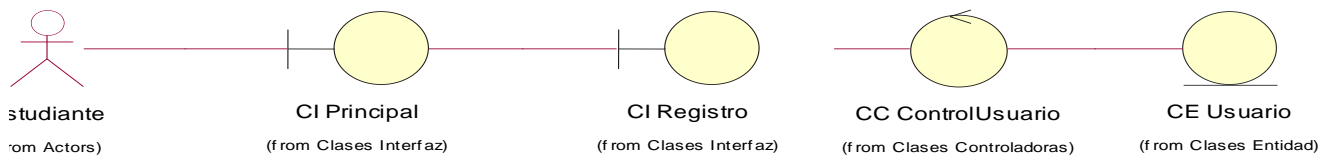
**Mostrar modelos médicos**



**Mostrar sistema del cuerpo**



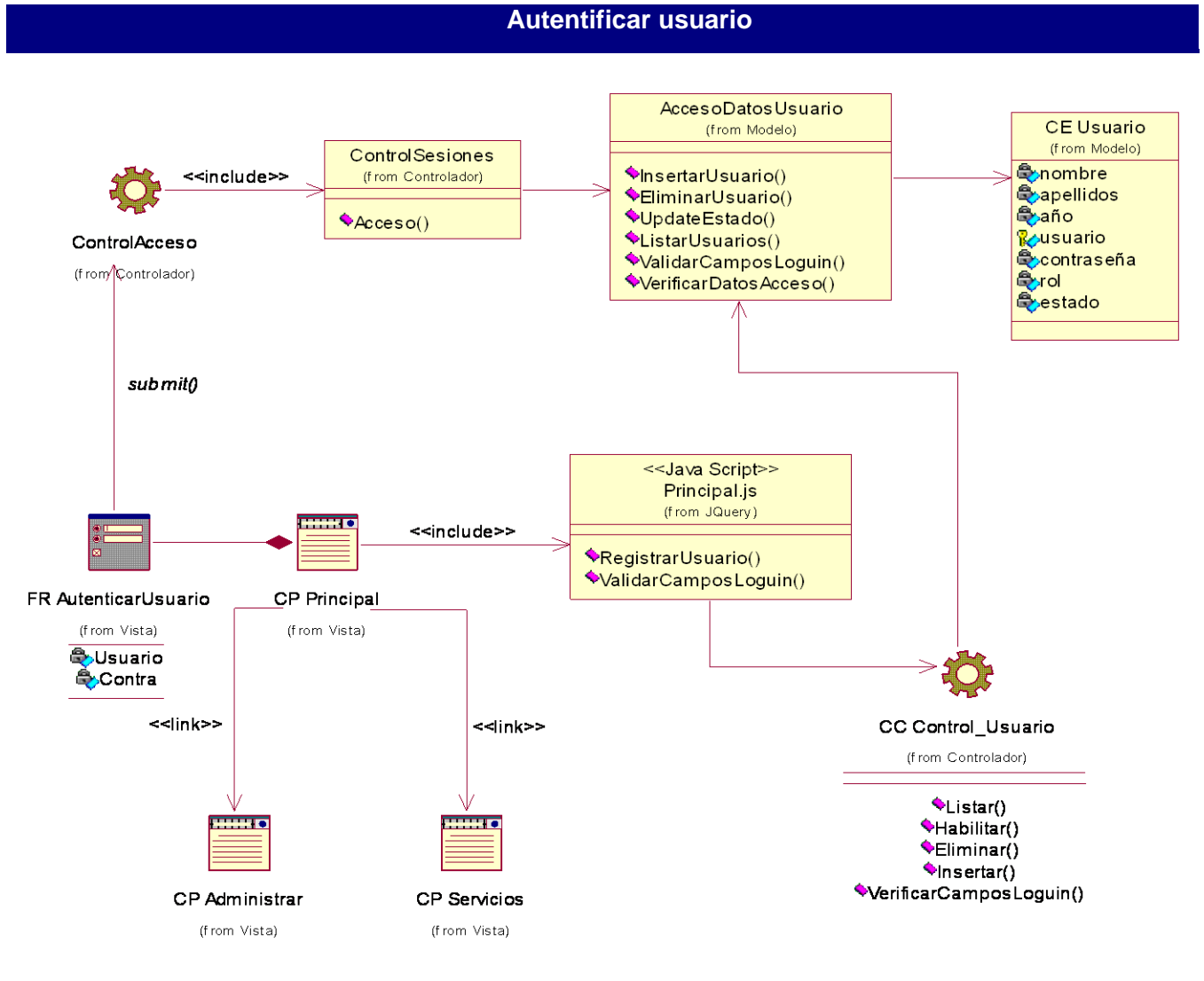
**Registrar usuario**



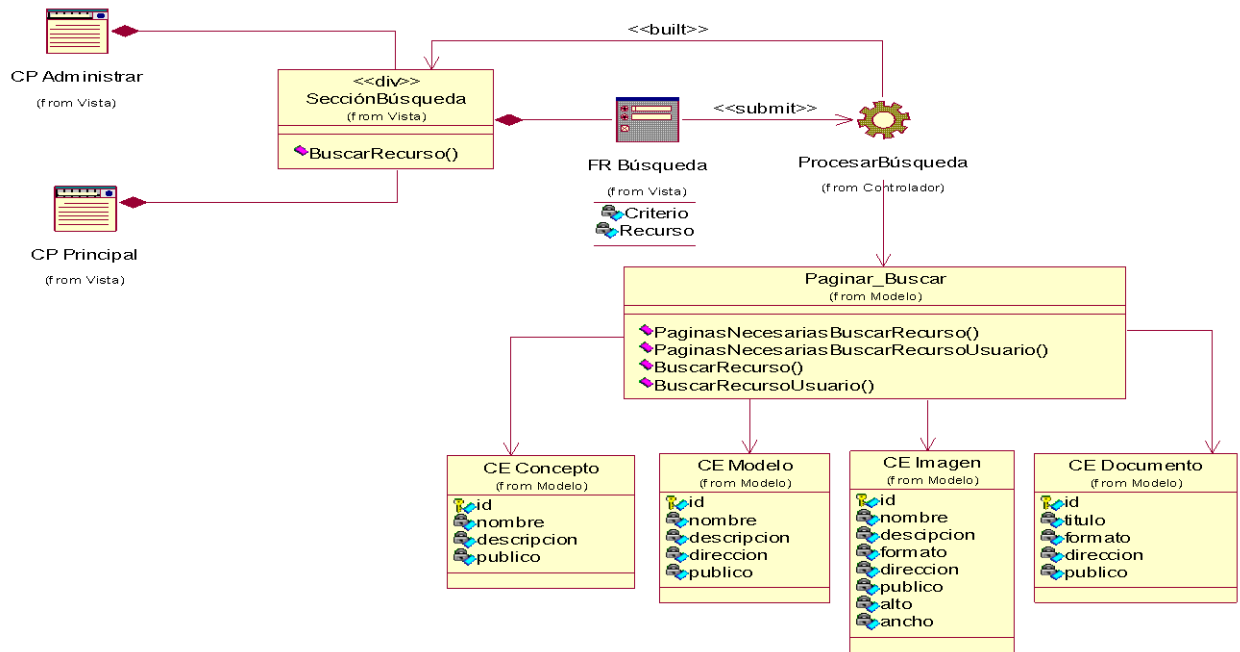


3. Anexo III

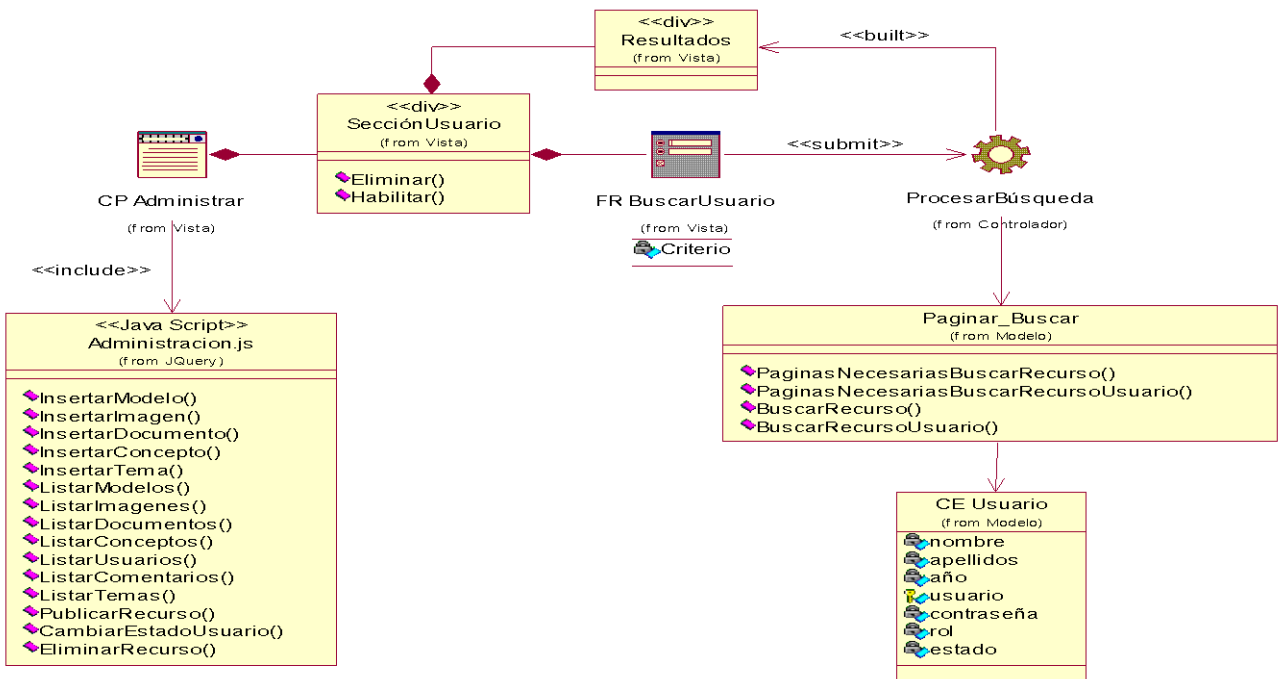
Diagrama de clases del diseño.



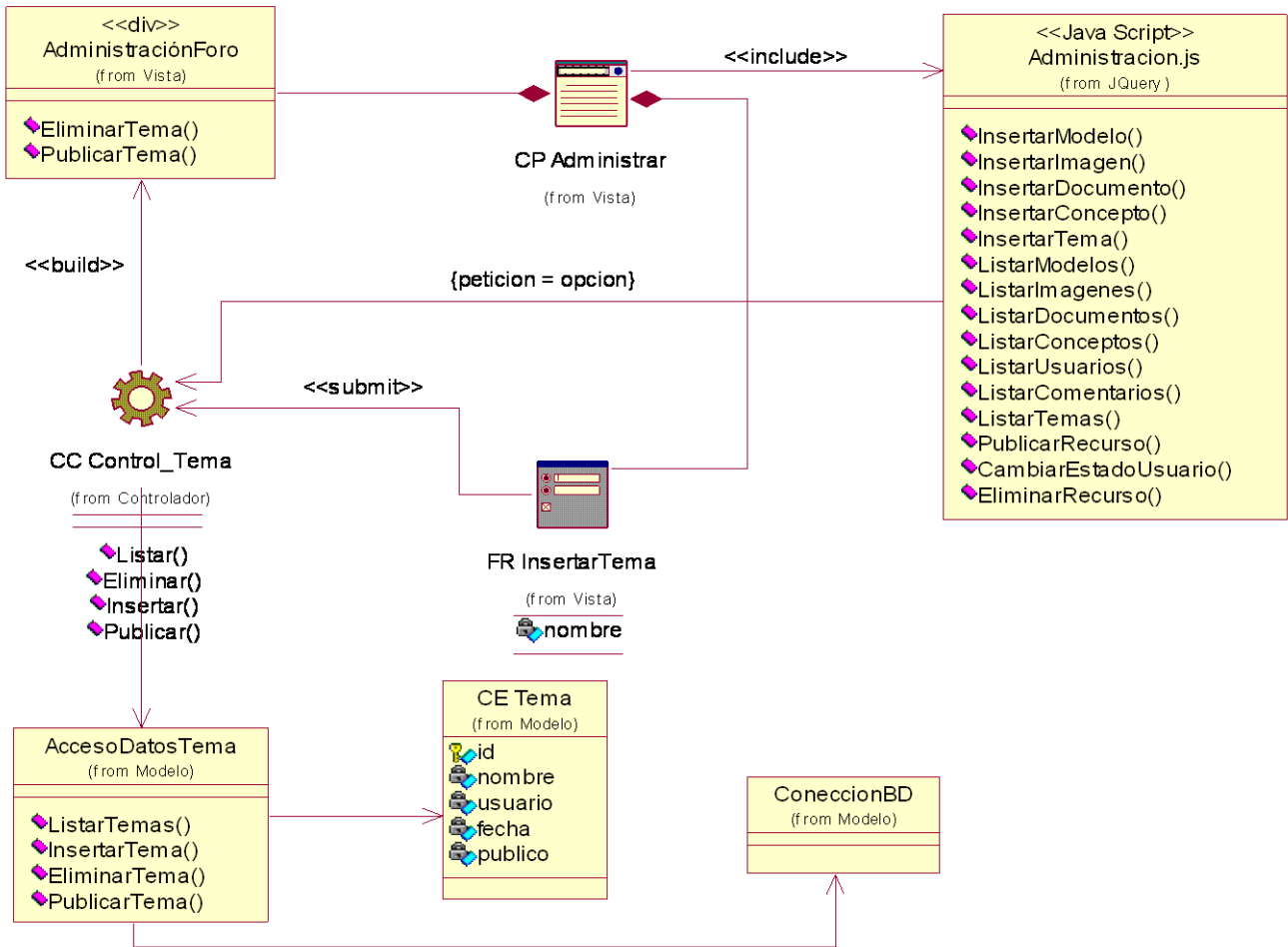
## Buscar recurso



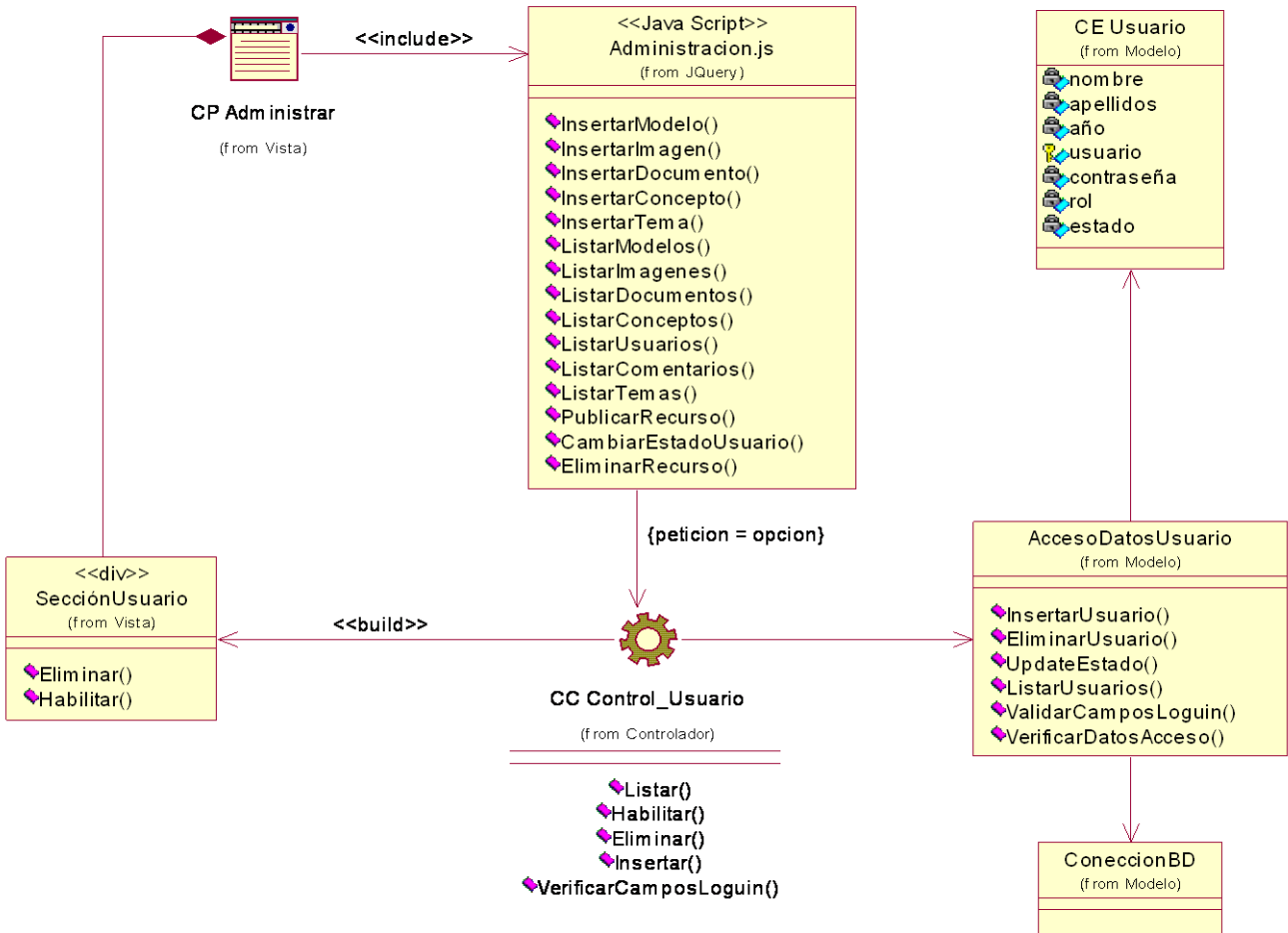
## Buscar usuario



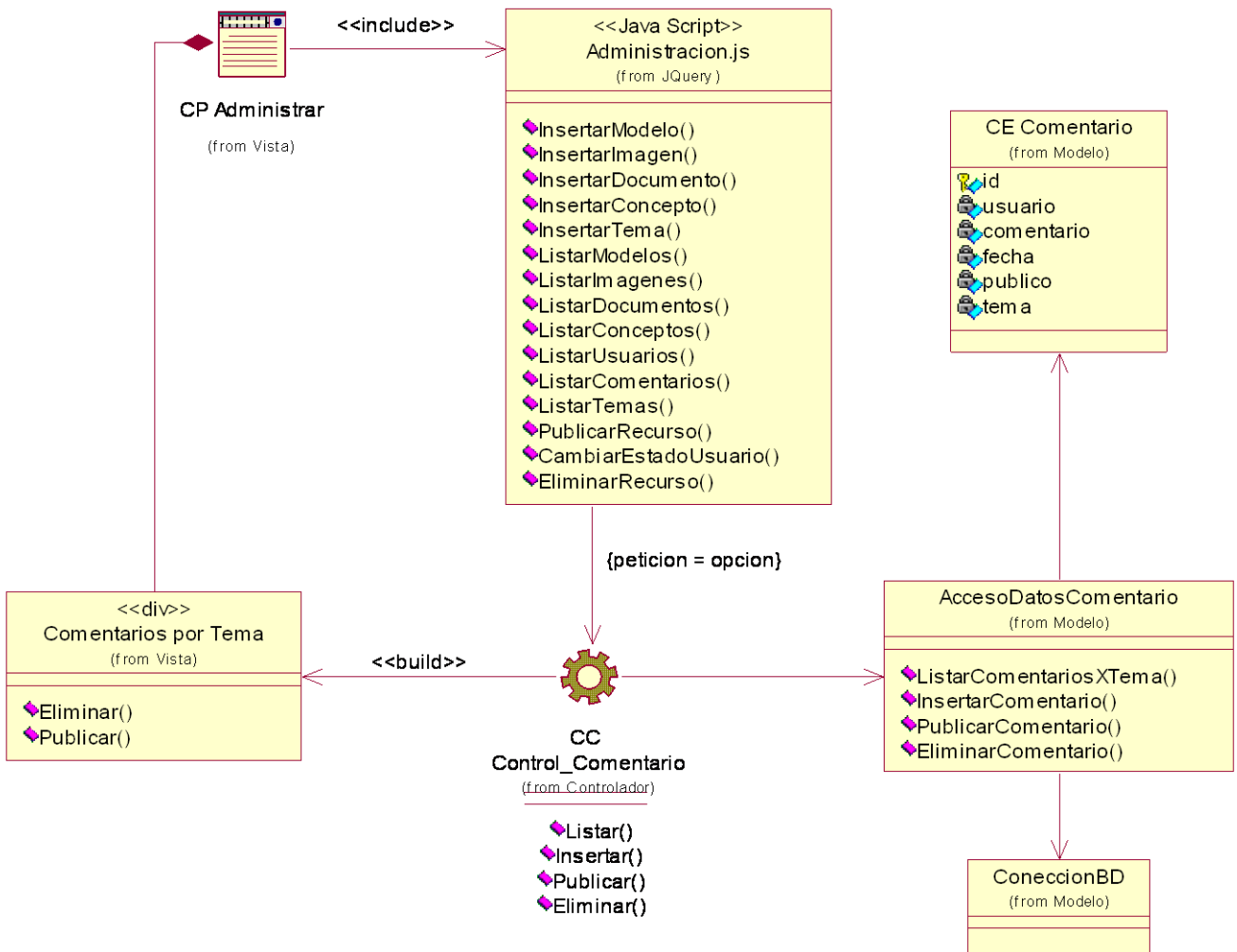
**Gestionar tema**



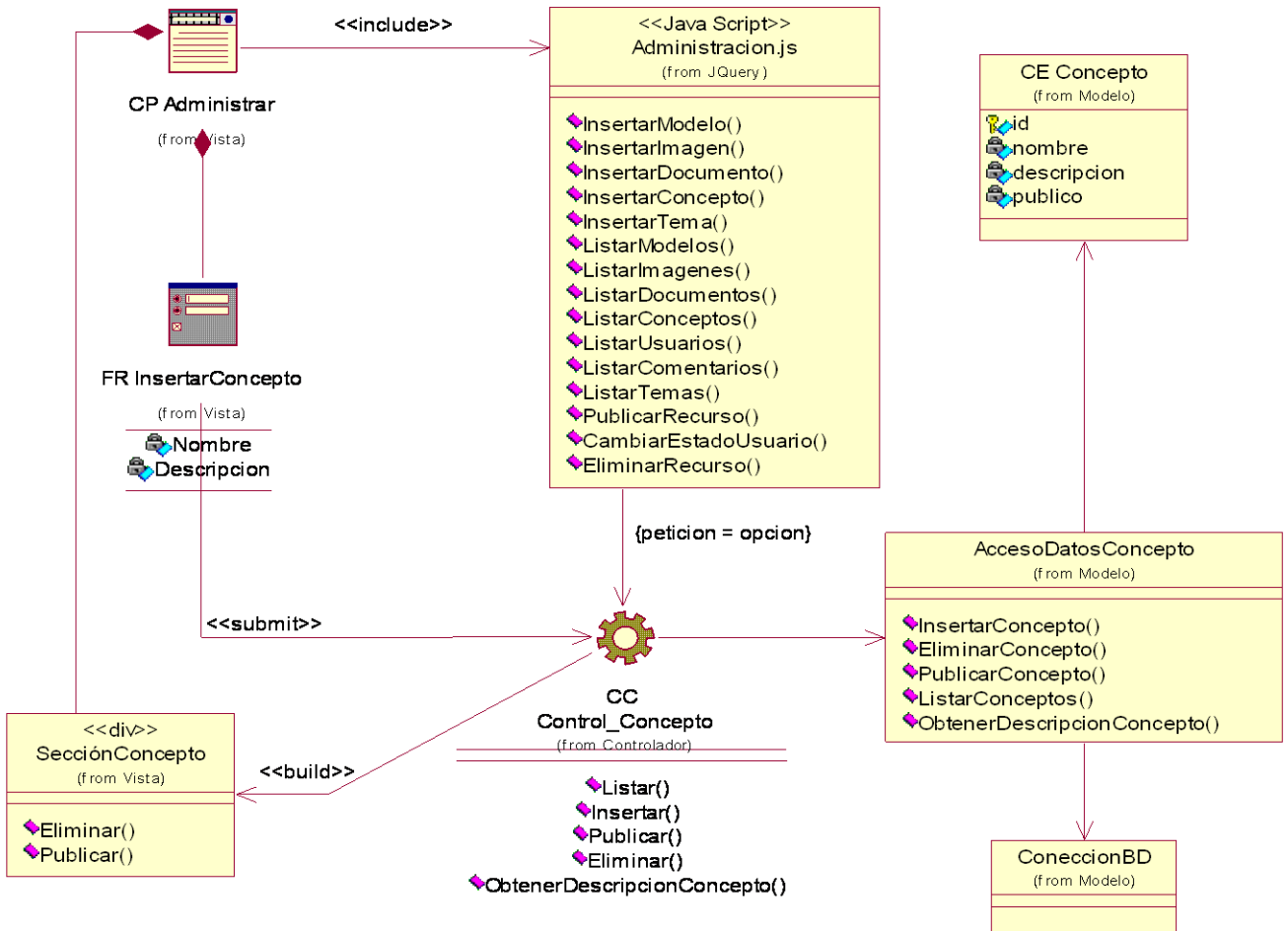
## Gestionar usuario



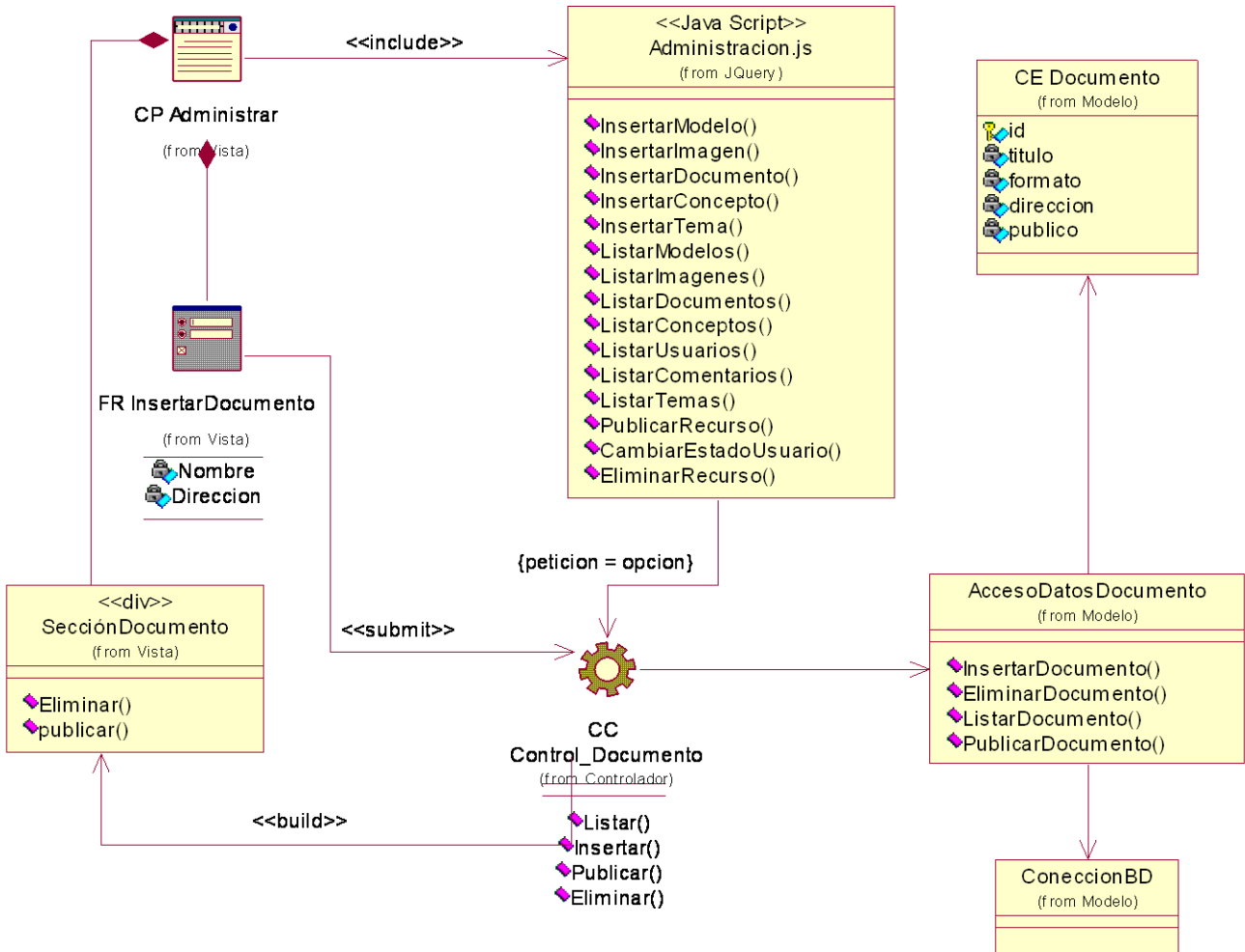
## Gestionar comentario



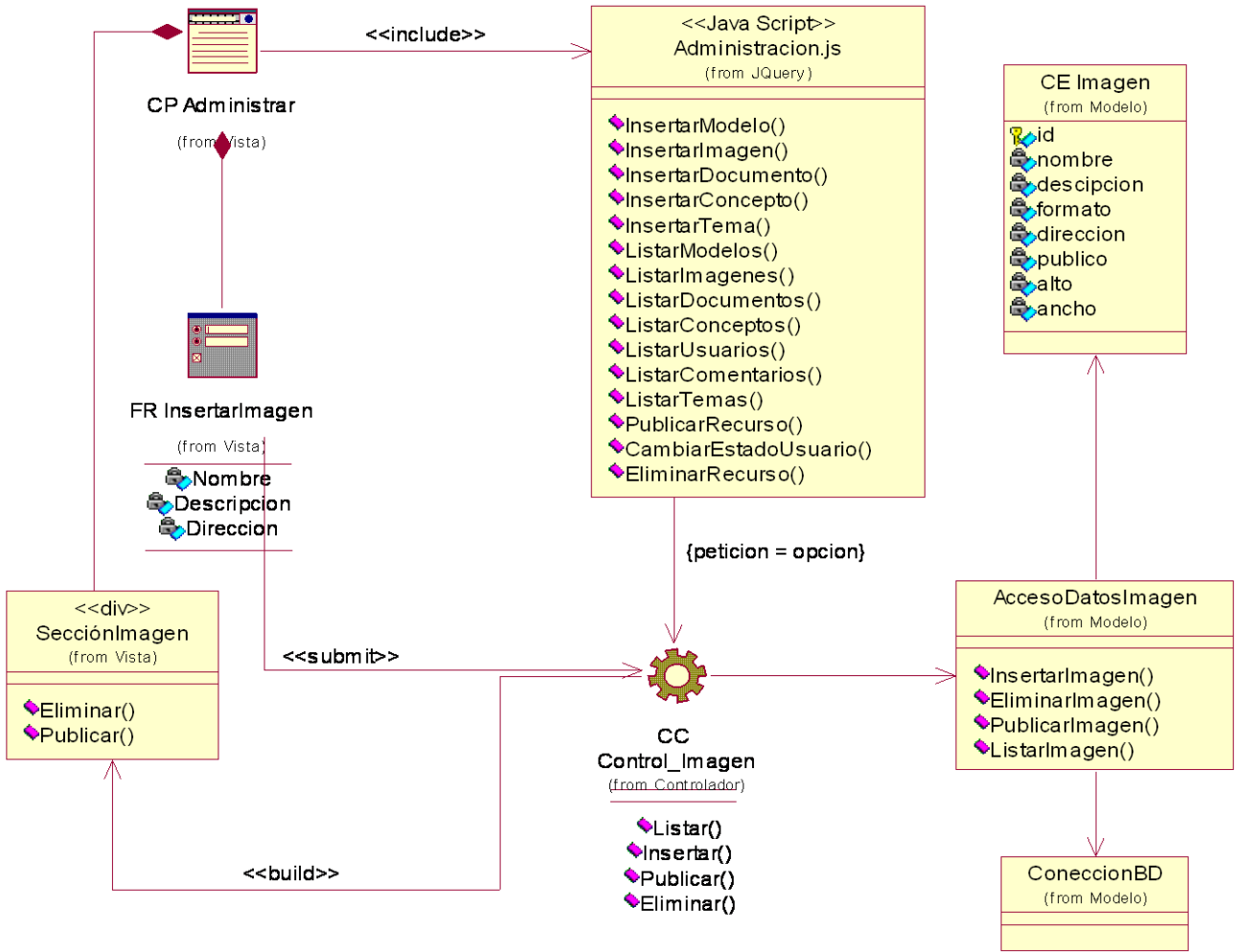
## Gestionar concepto médico



**Gestionar documento**

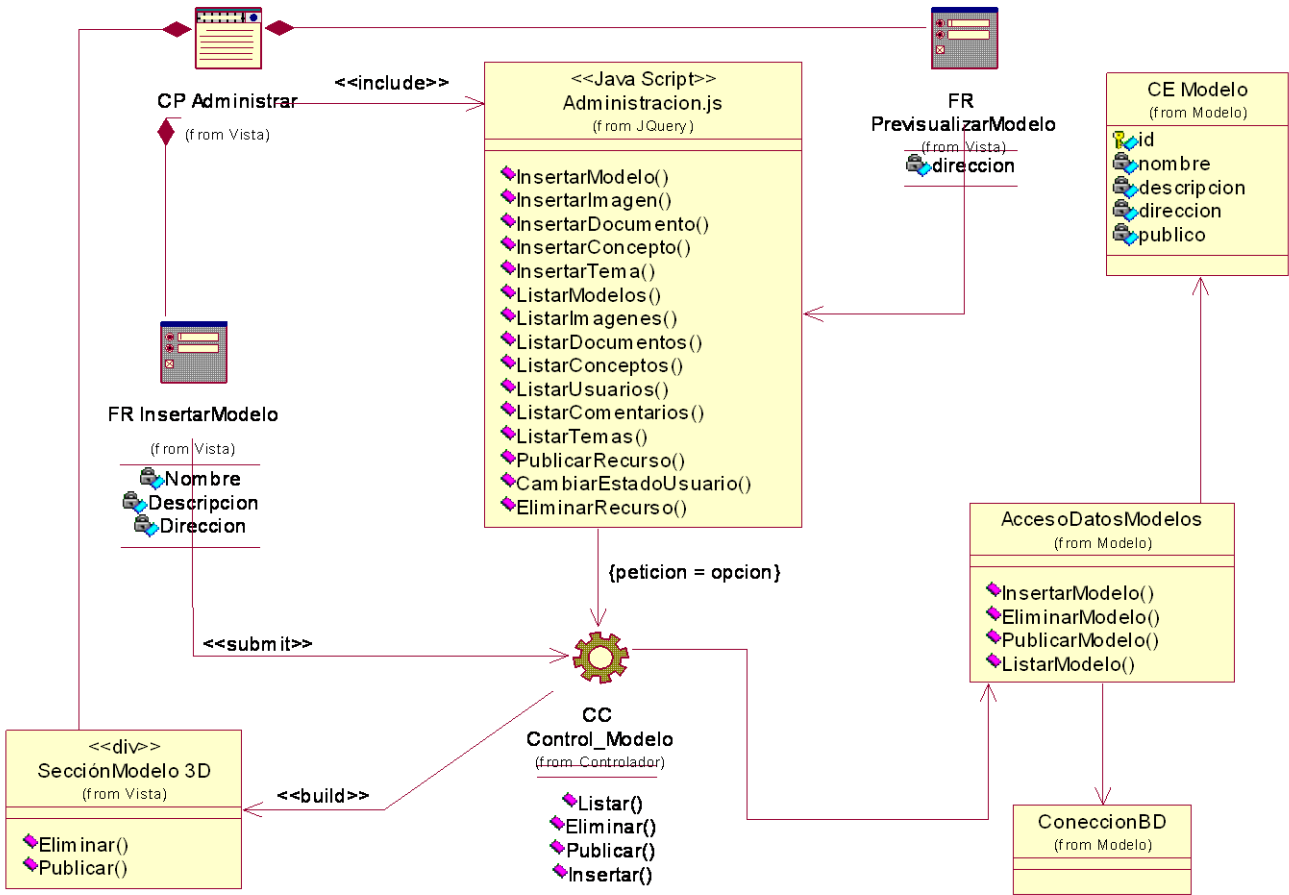


## Gestionar imagen médica

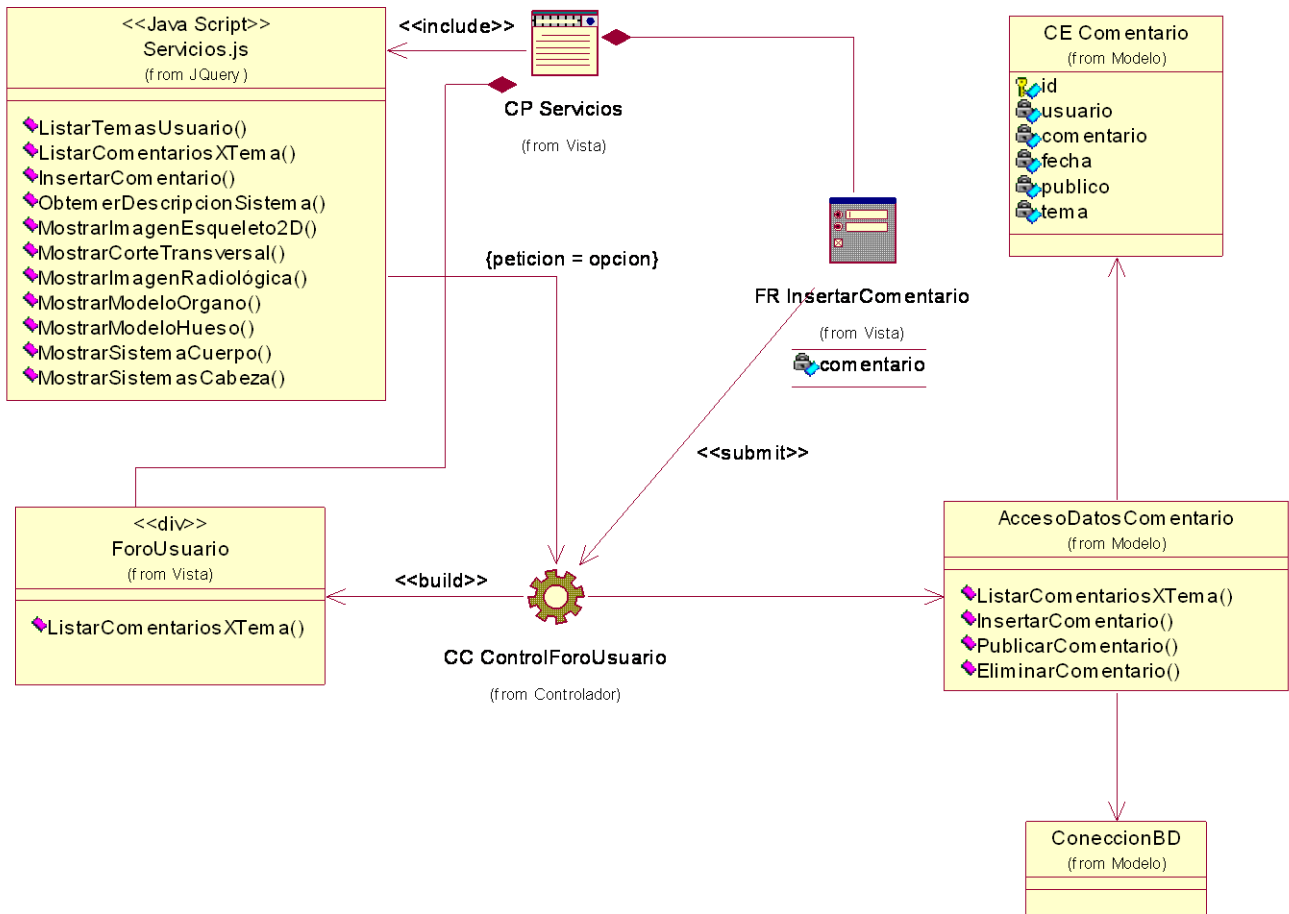




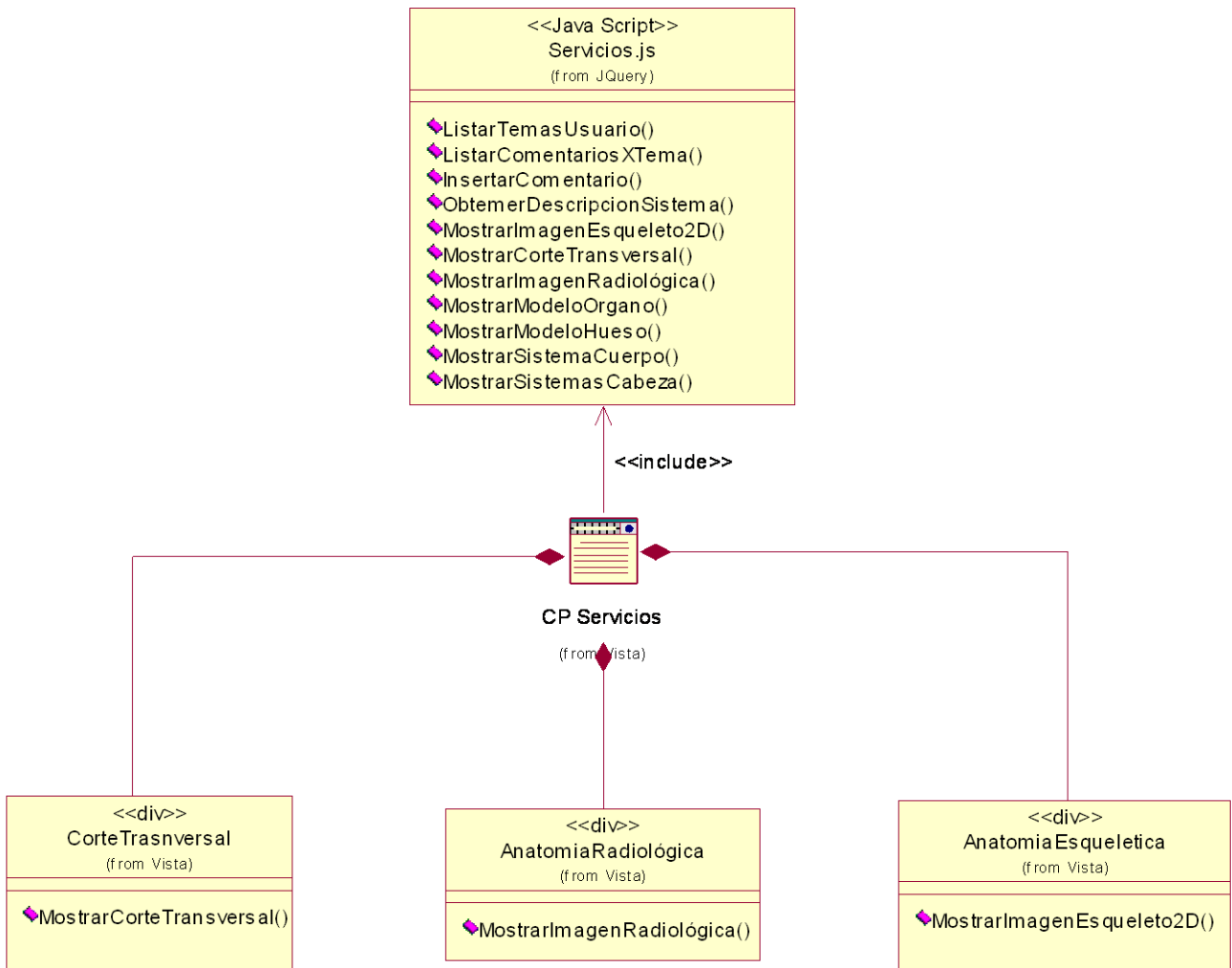
## Gestionar modelo 3D



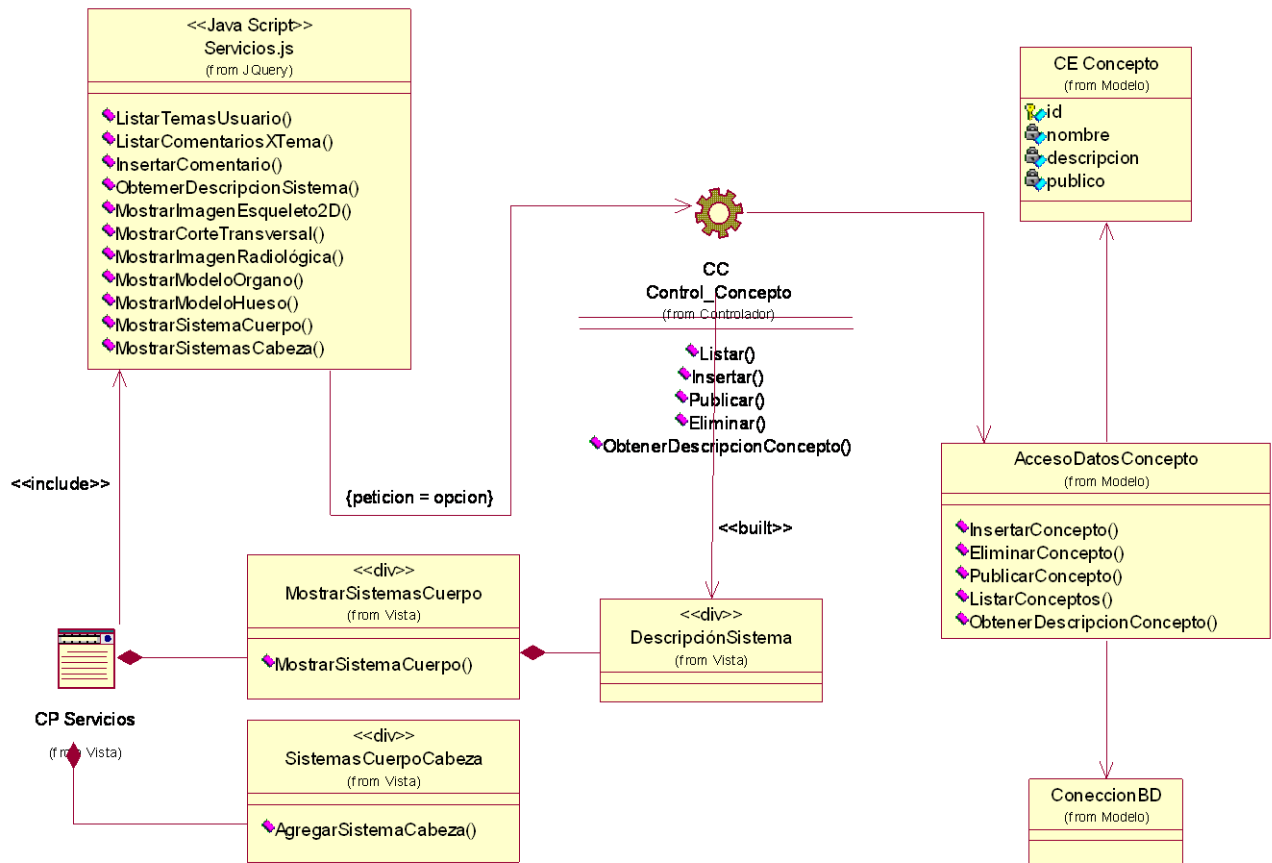
**Insertar comentario en el foro**



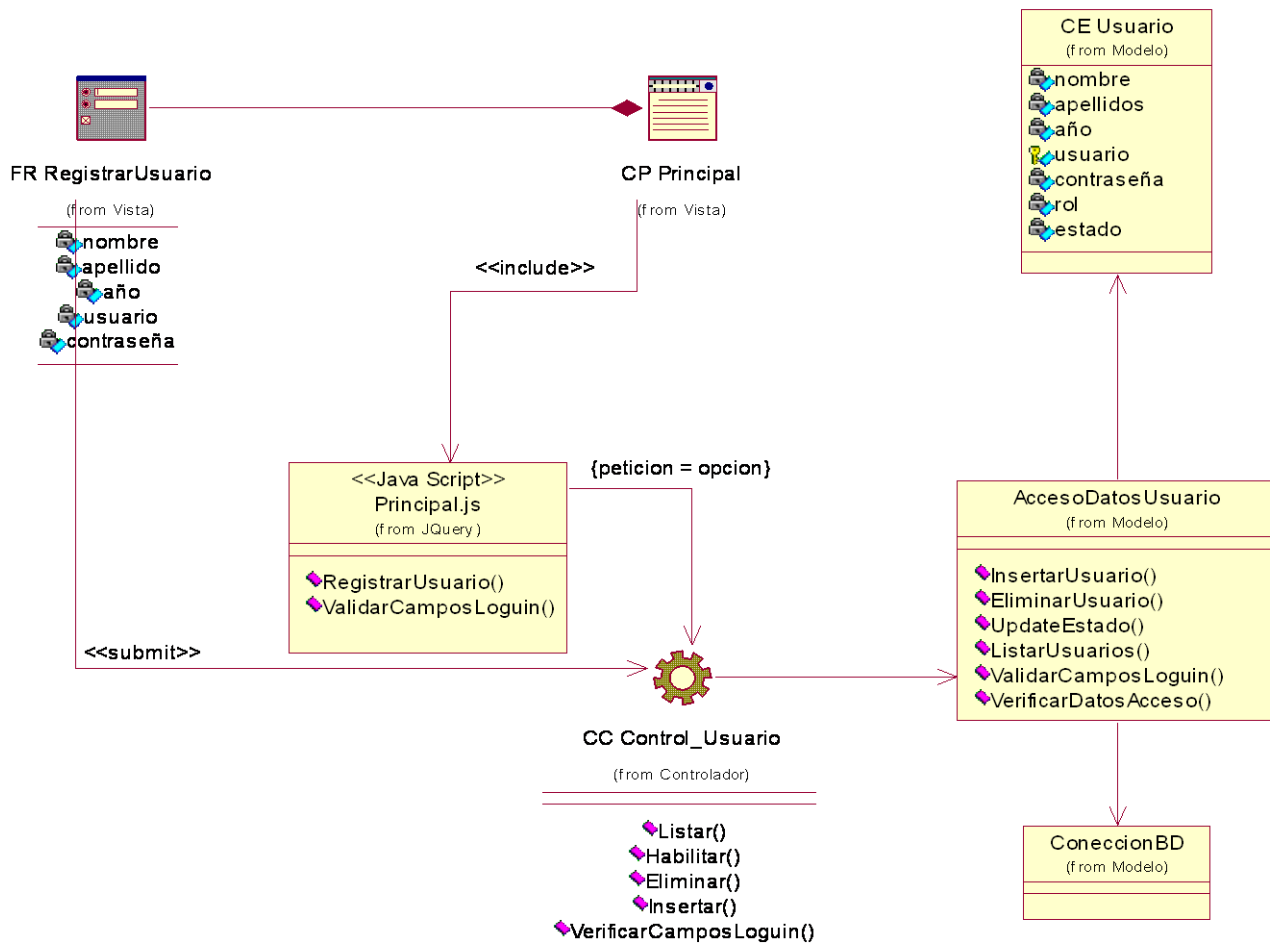
**Mostrar imágenes médicas**



**Mostrar sistema del cuerpo**



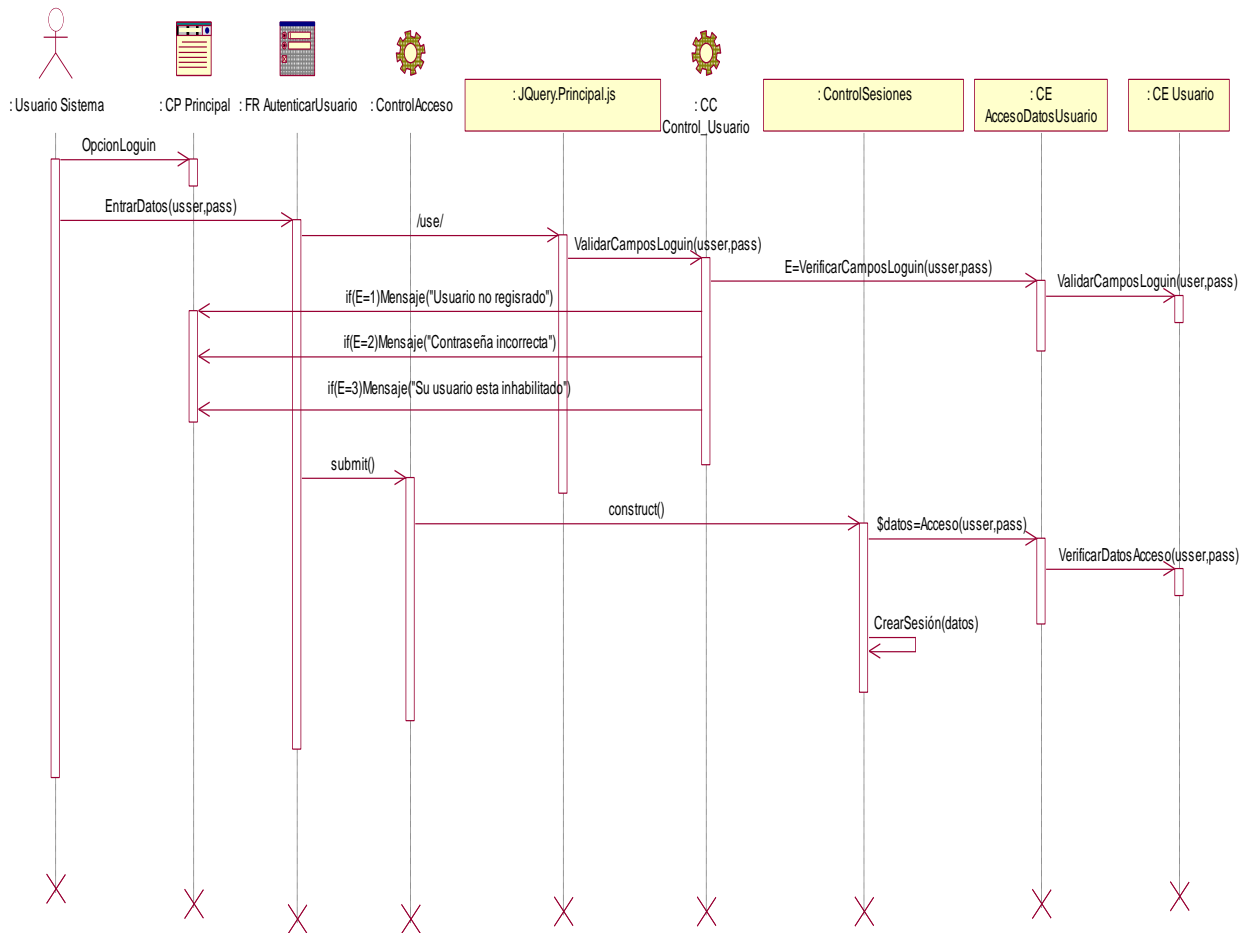
## Registrar usuario



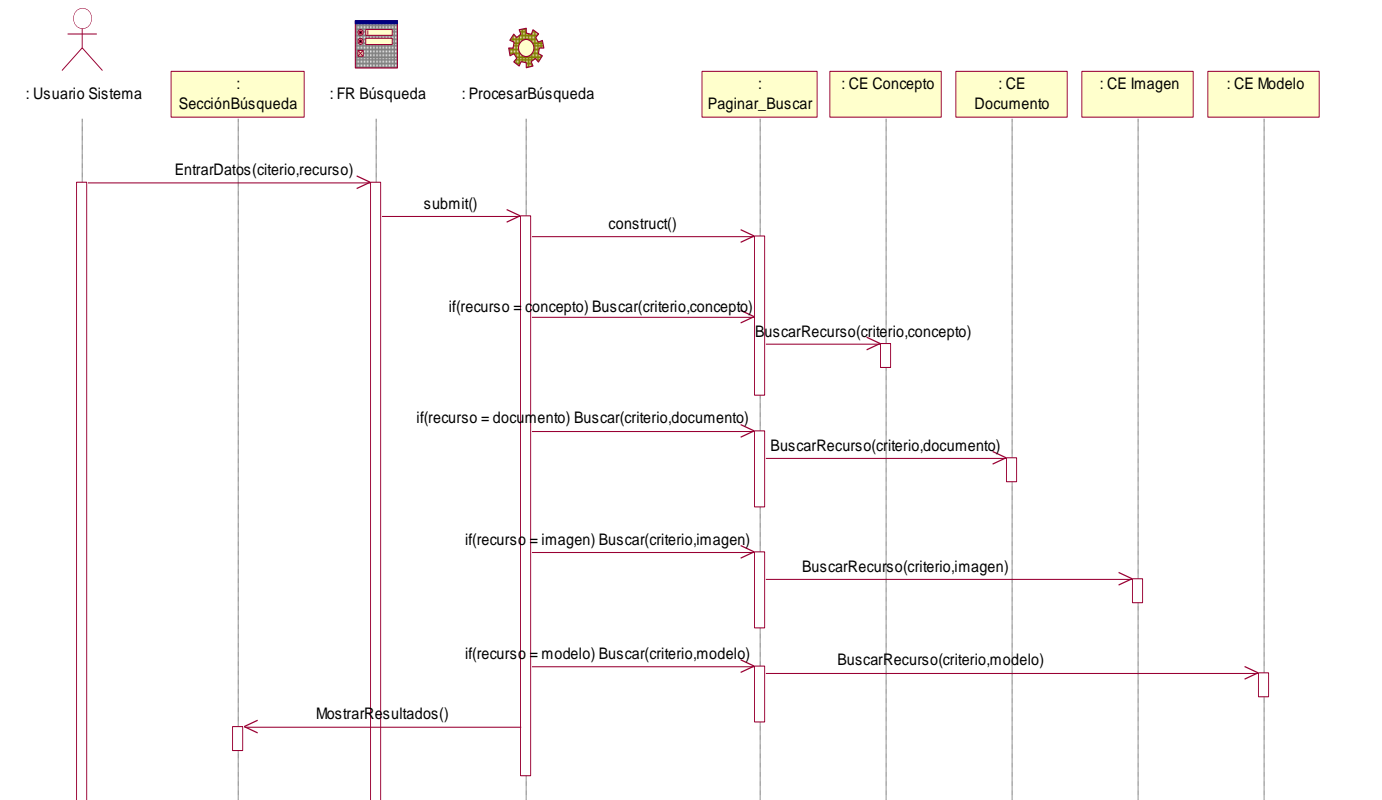
4. Anexo IV

Diagramas de secuencia

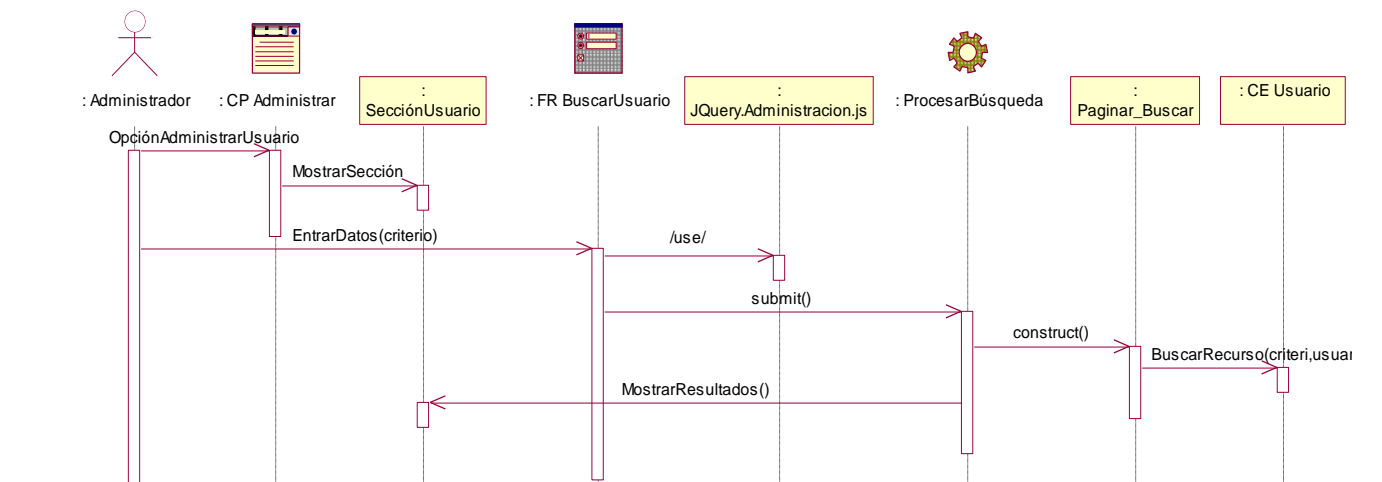
**Autenticar usuario**



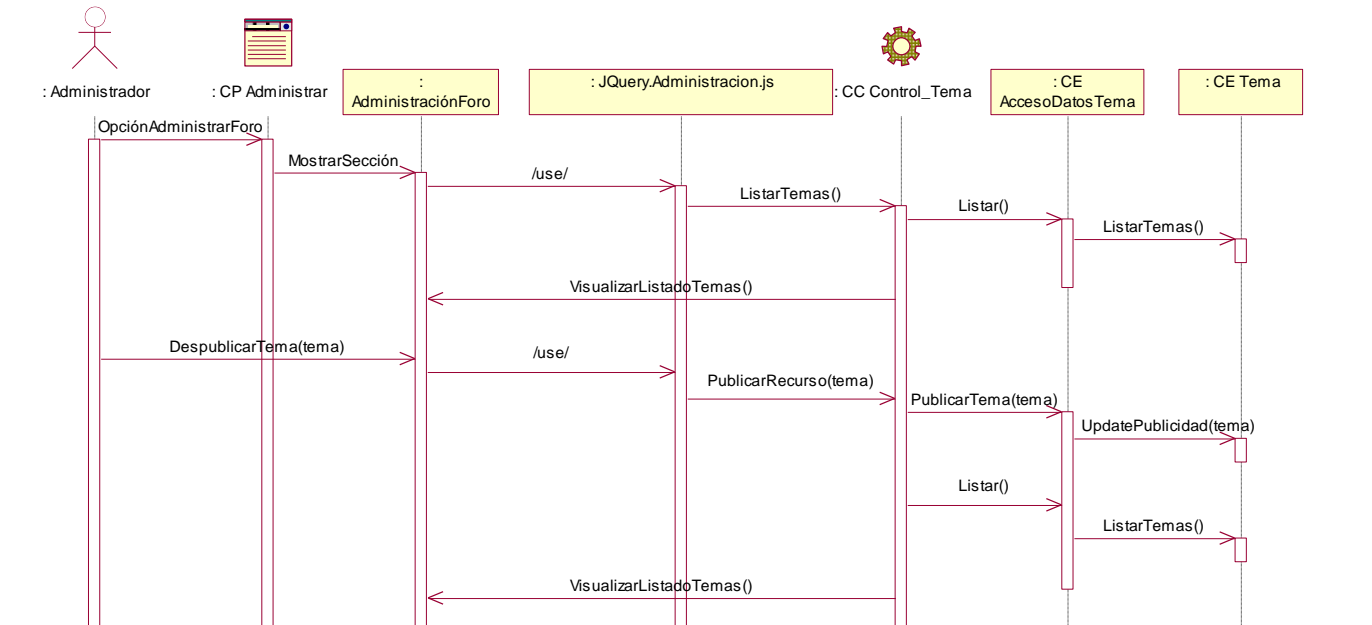
## Buscar Recurso



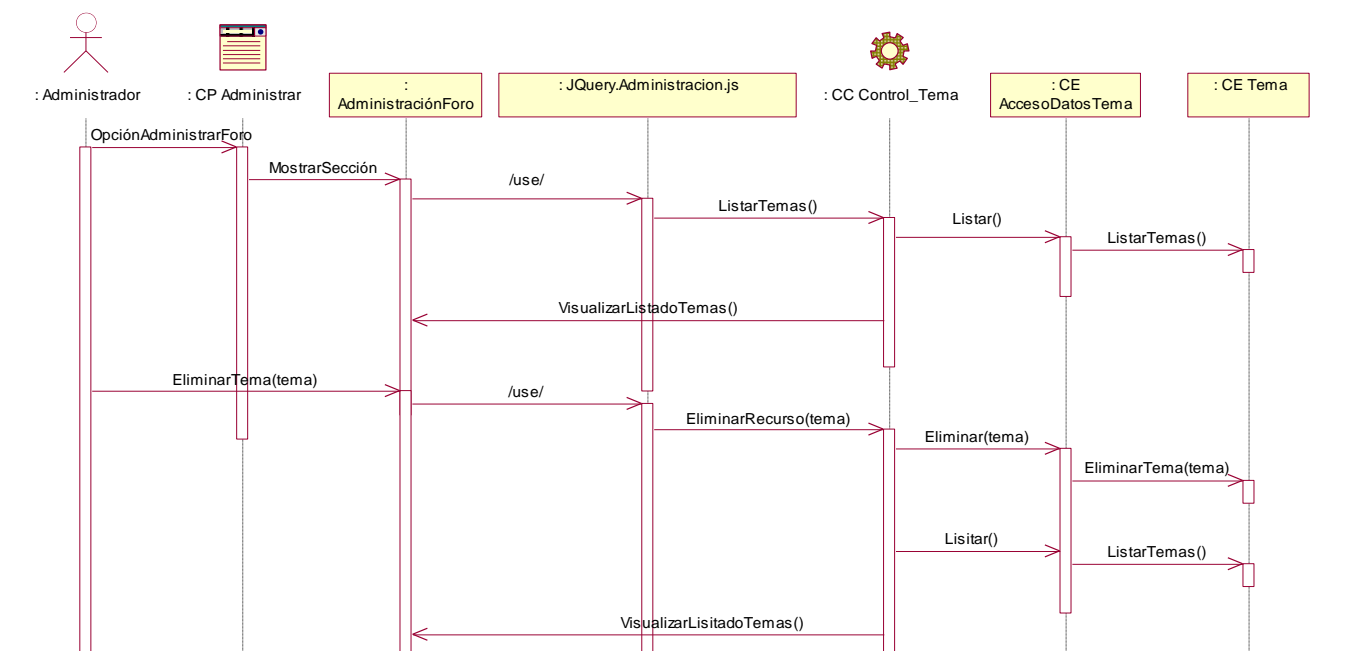
## Buscar Usuario



## Despublicar tema

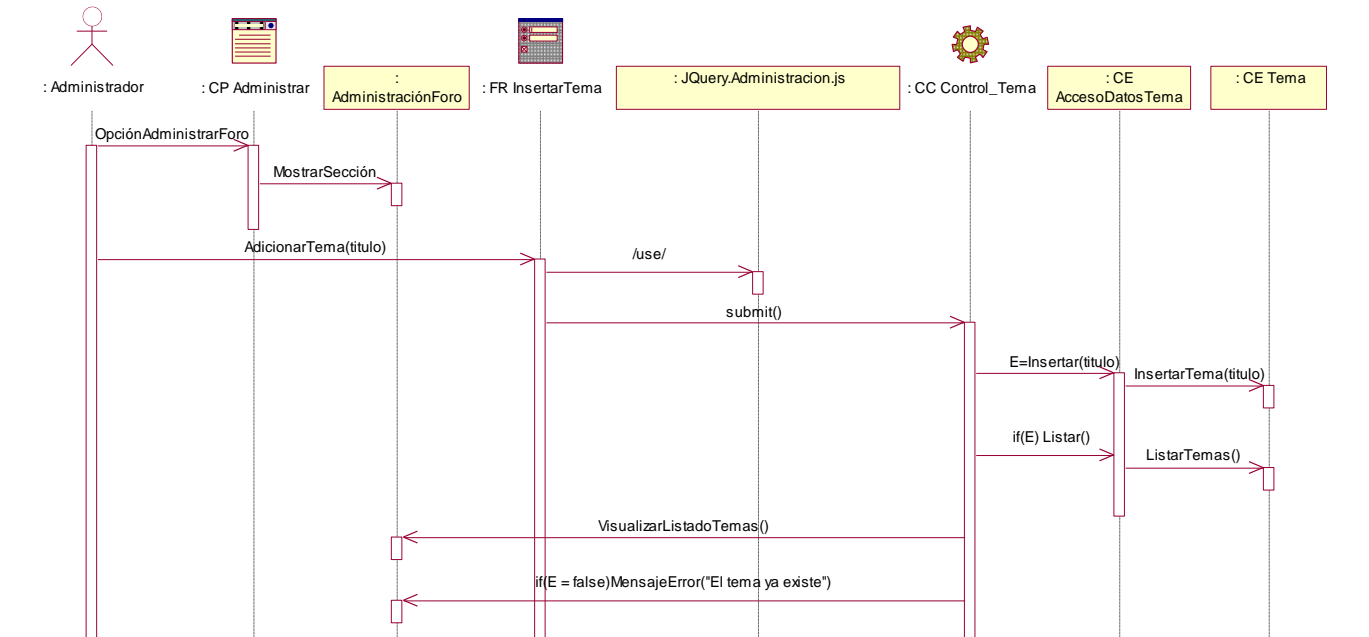


## Eliminar tema

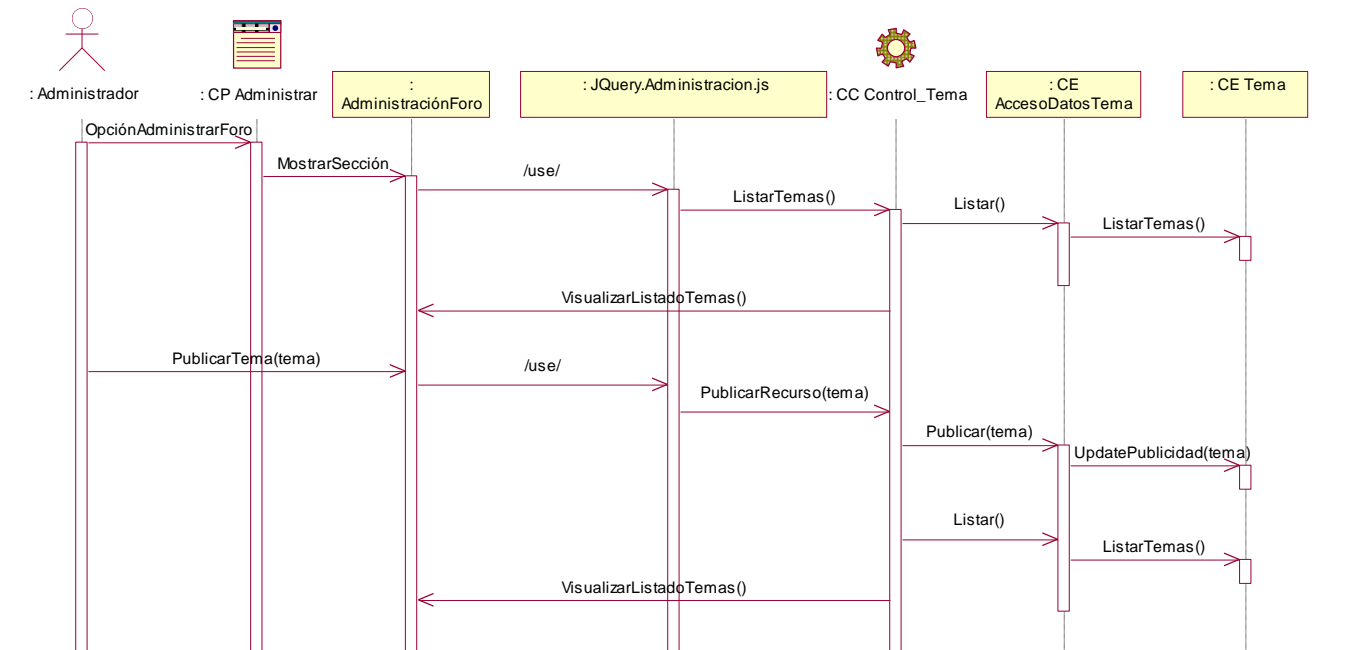




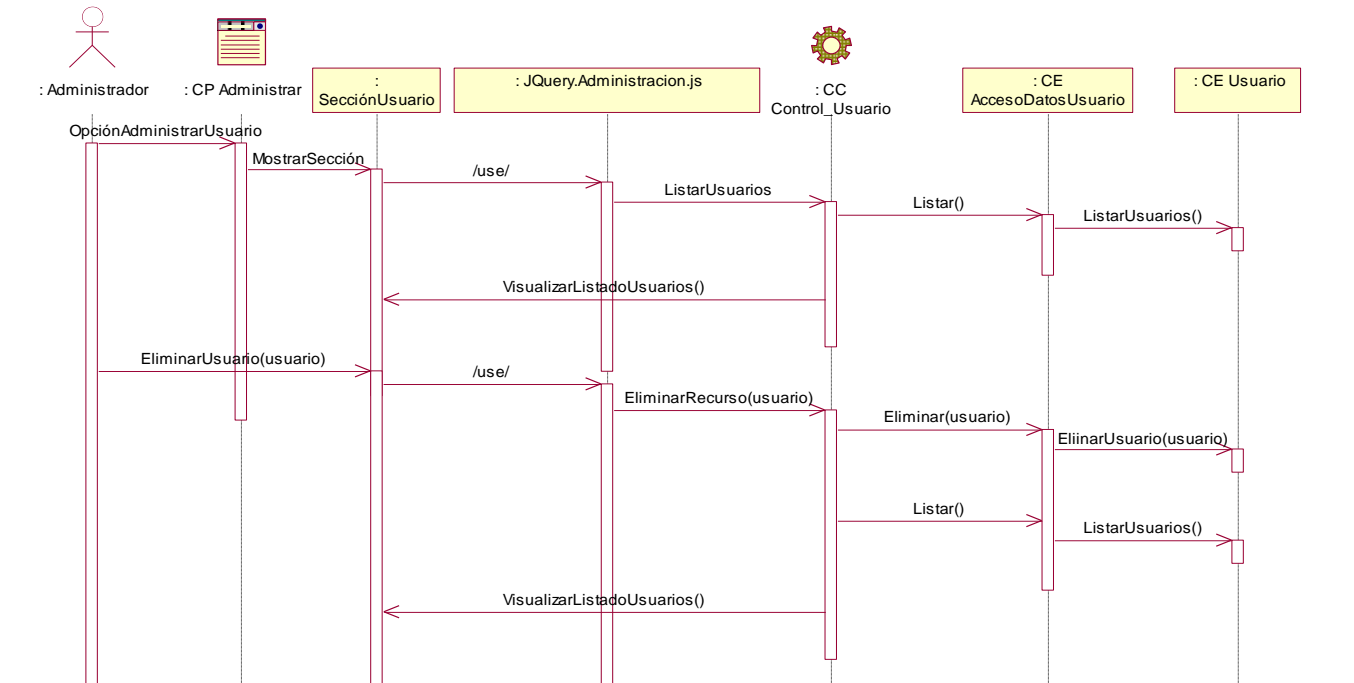
## Insertar tema



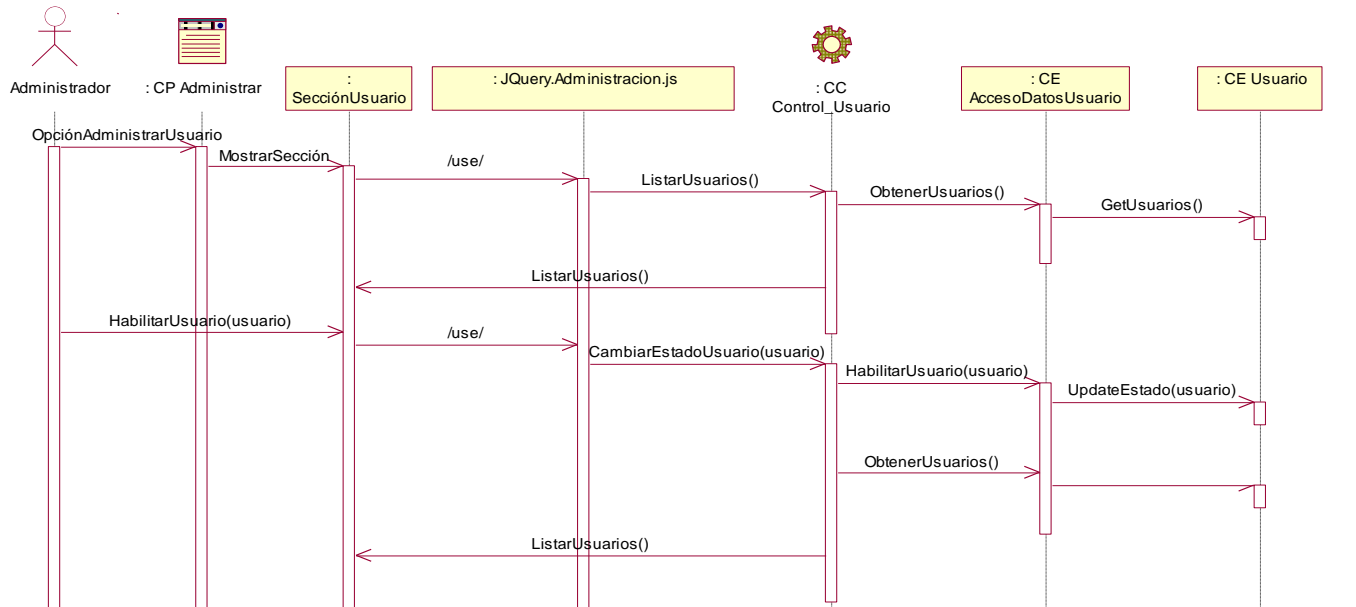
## Publicar tema



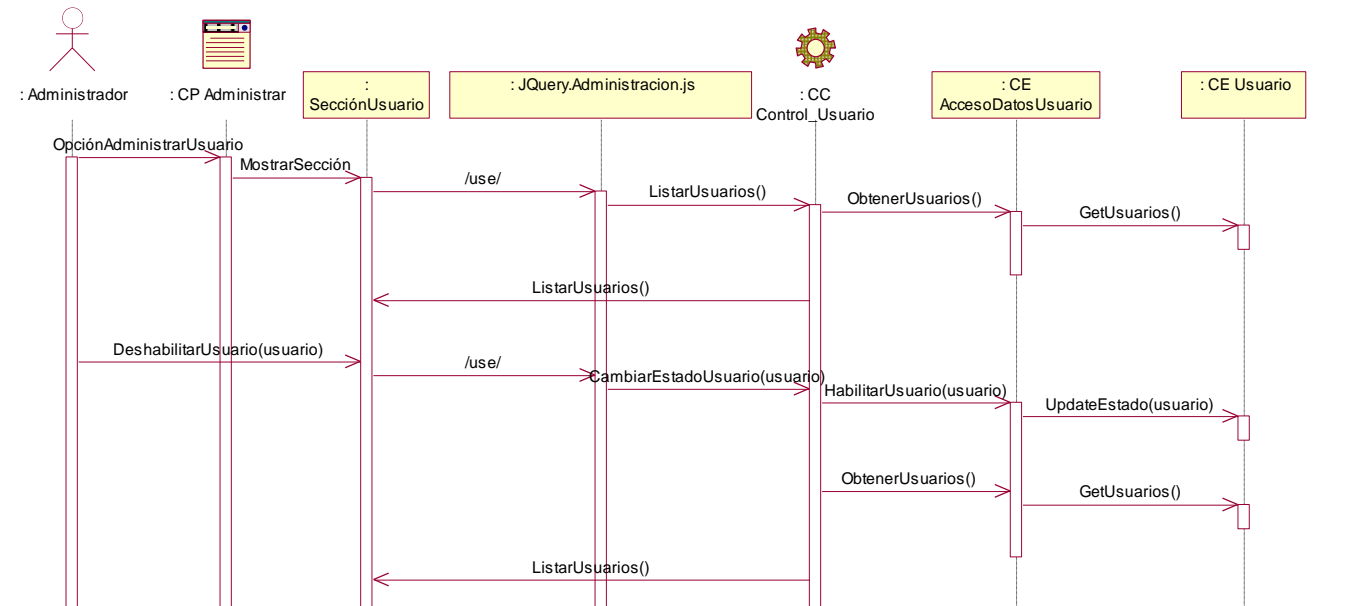
## Eliminar Usuario



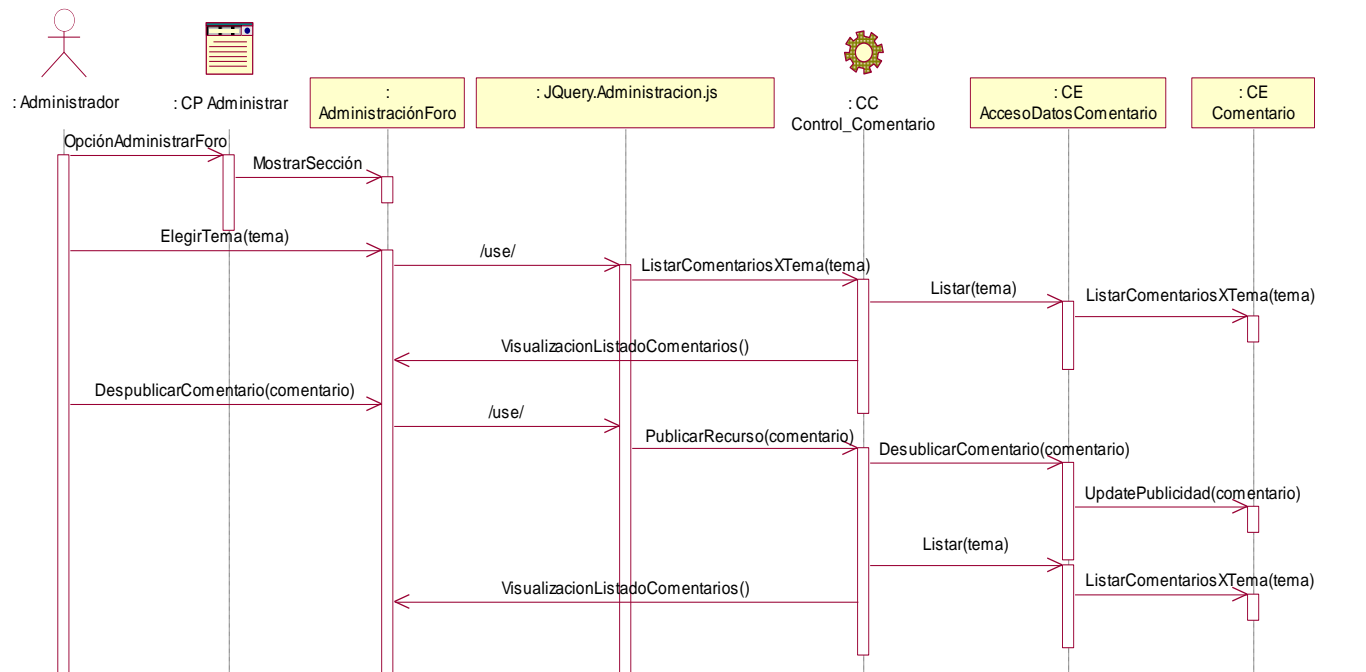
## Habilitar usuario



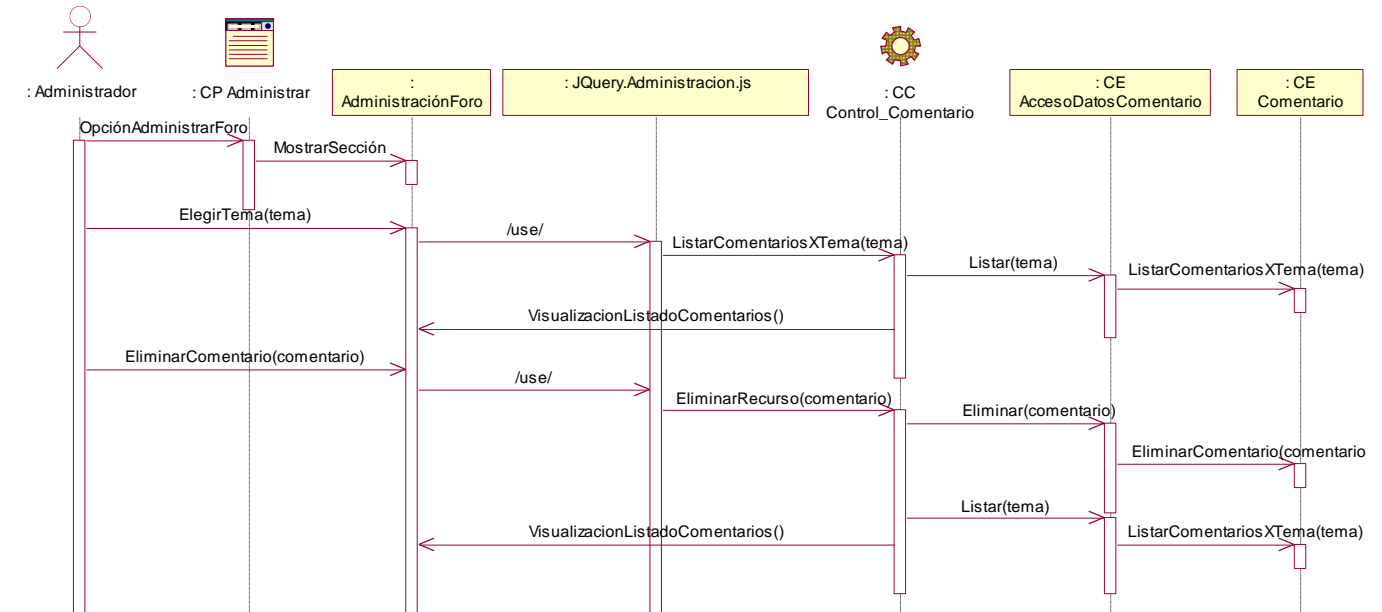
## Inhabilitar usuario



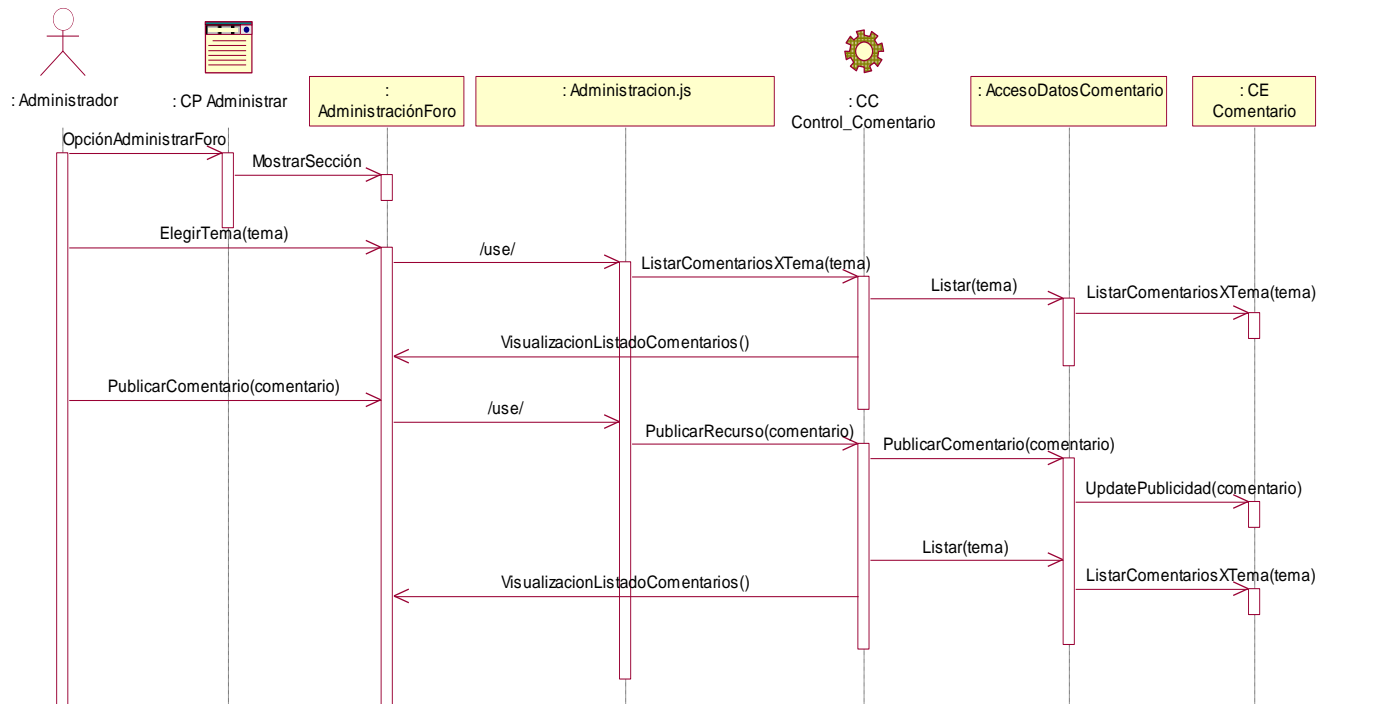
## Despublicar comentario



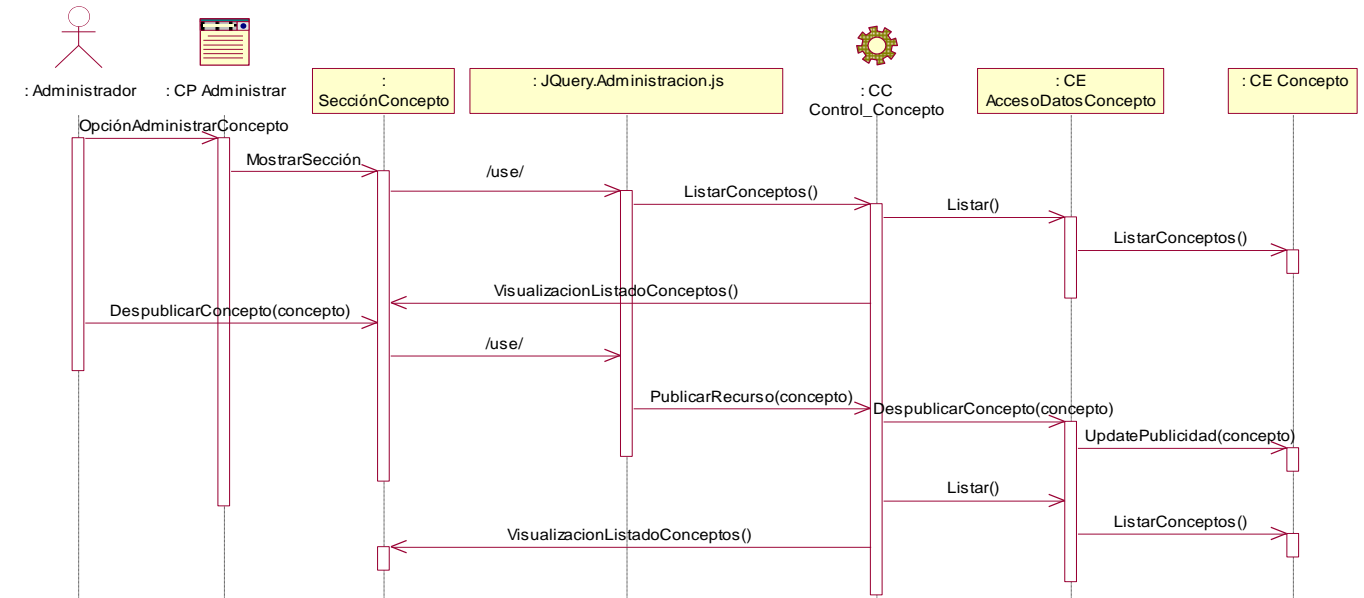
## Eliminar Comentario



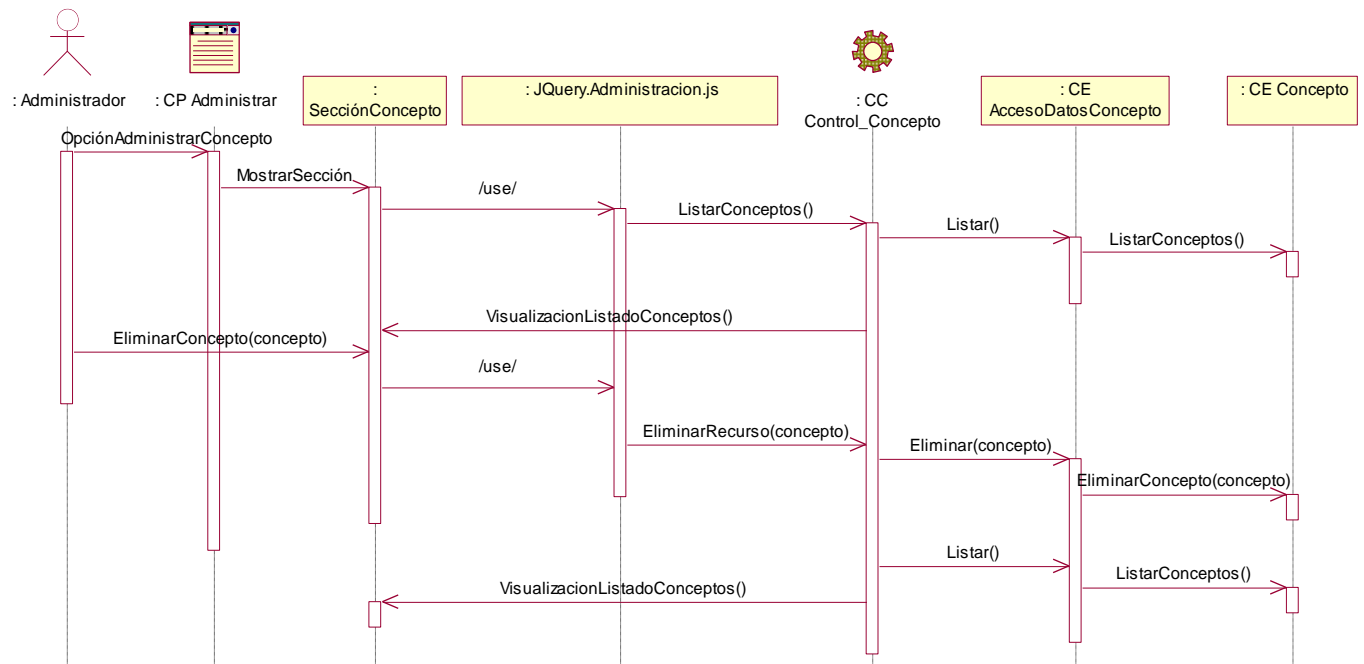
## Publicar comentario



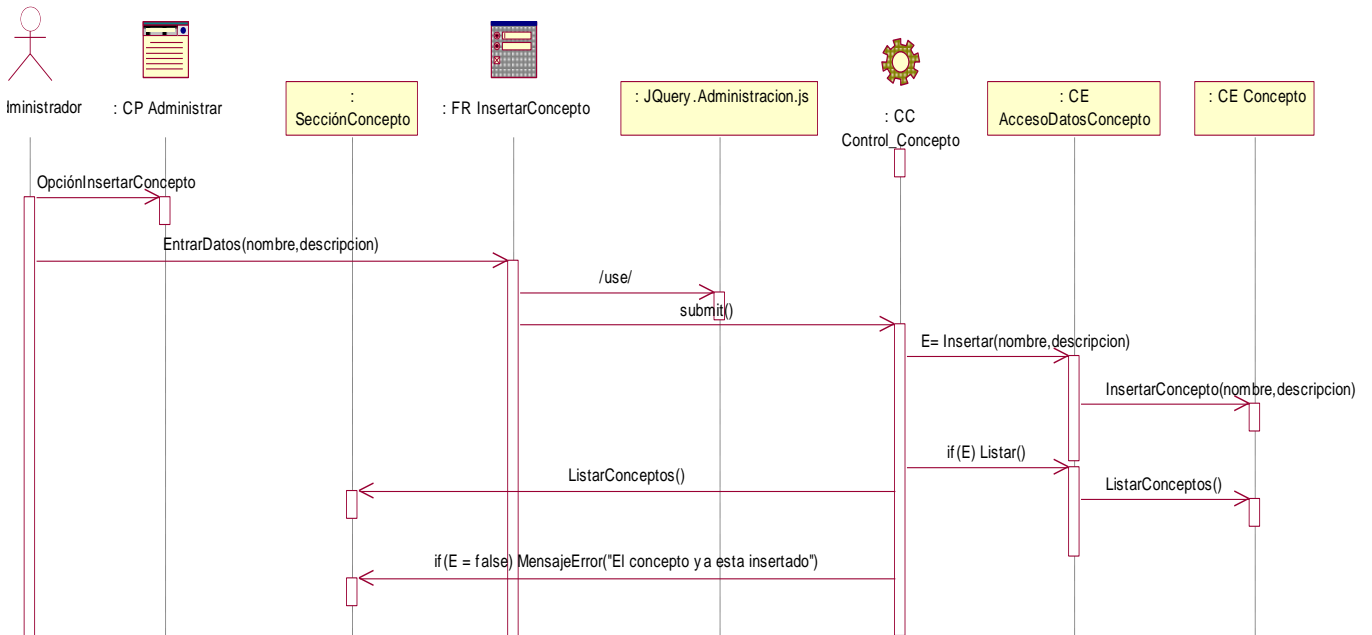
## Despublicar concepto



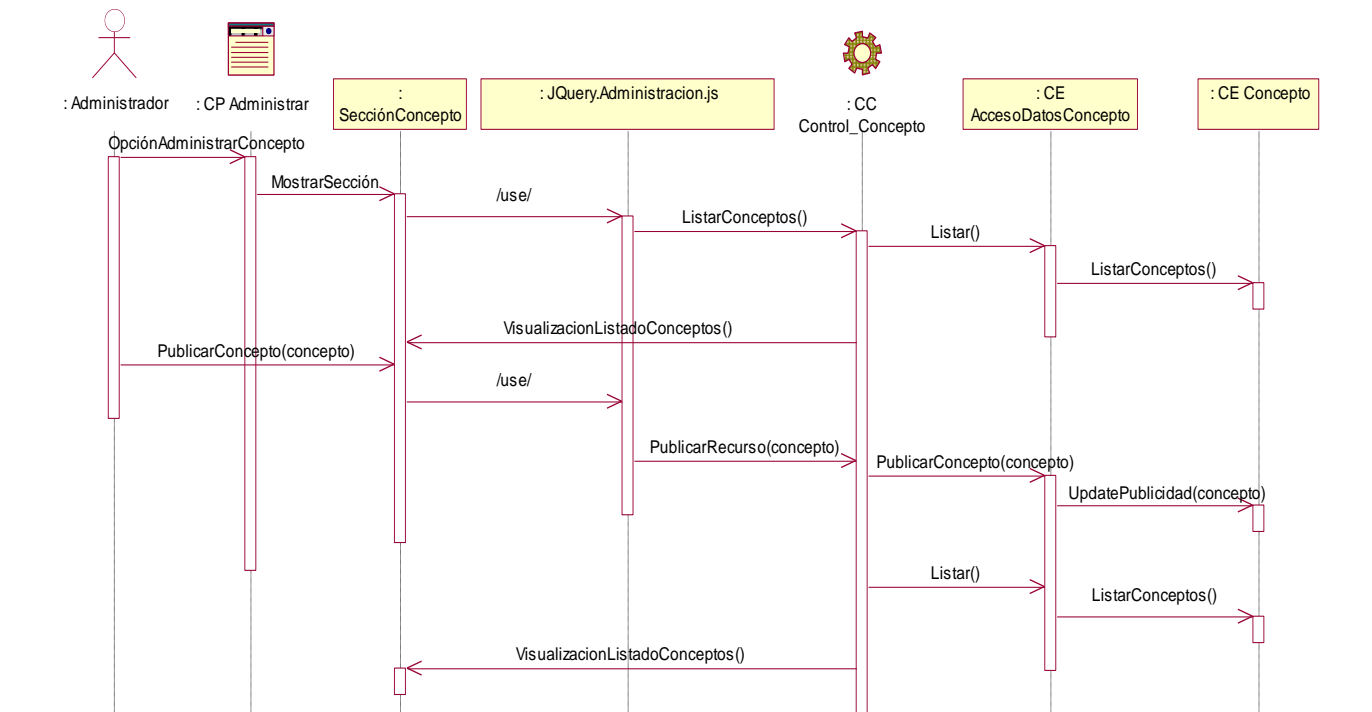
## Eliminar concepto



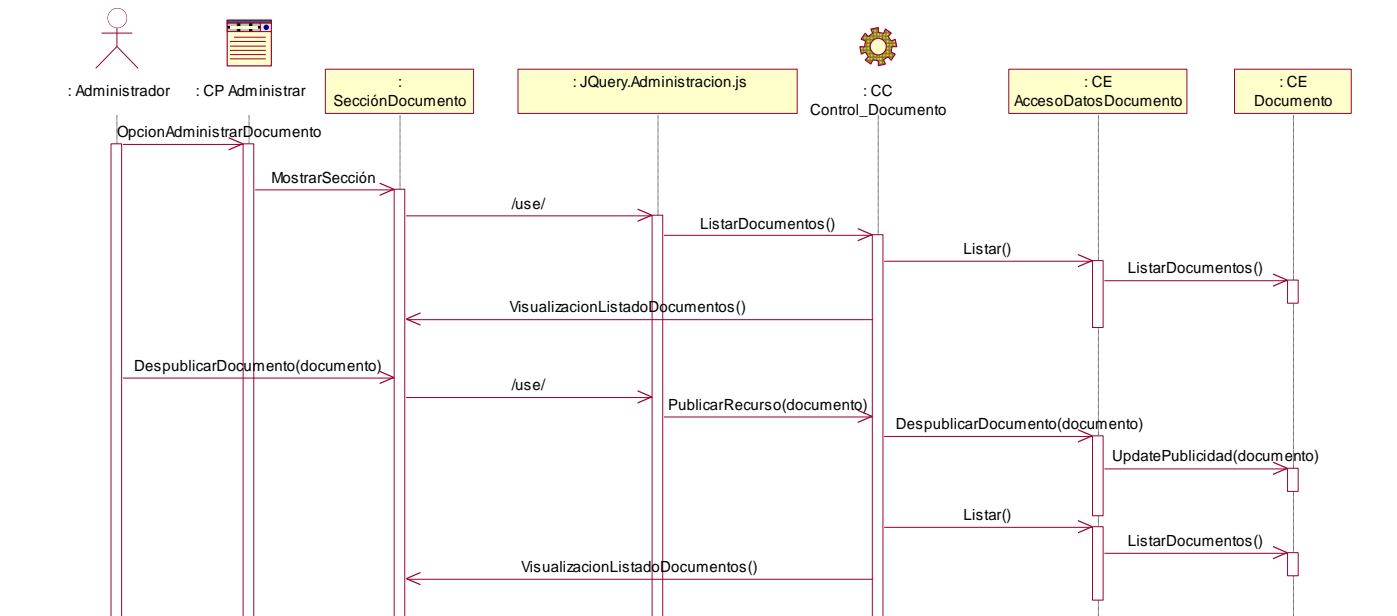
## Insertar concepto



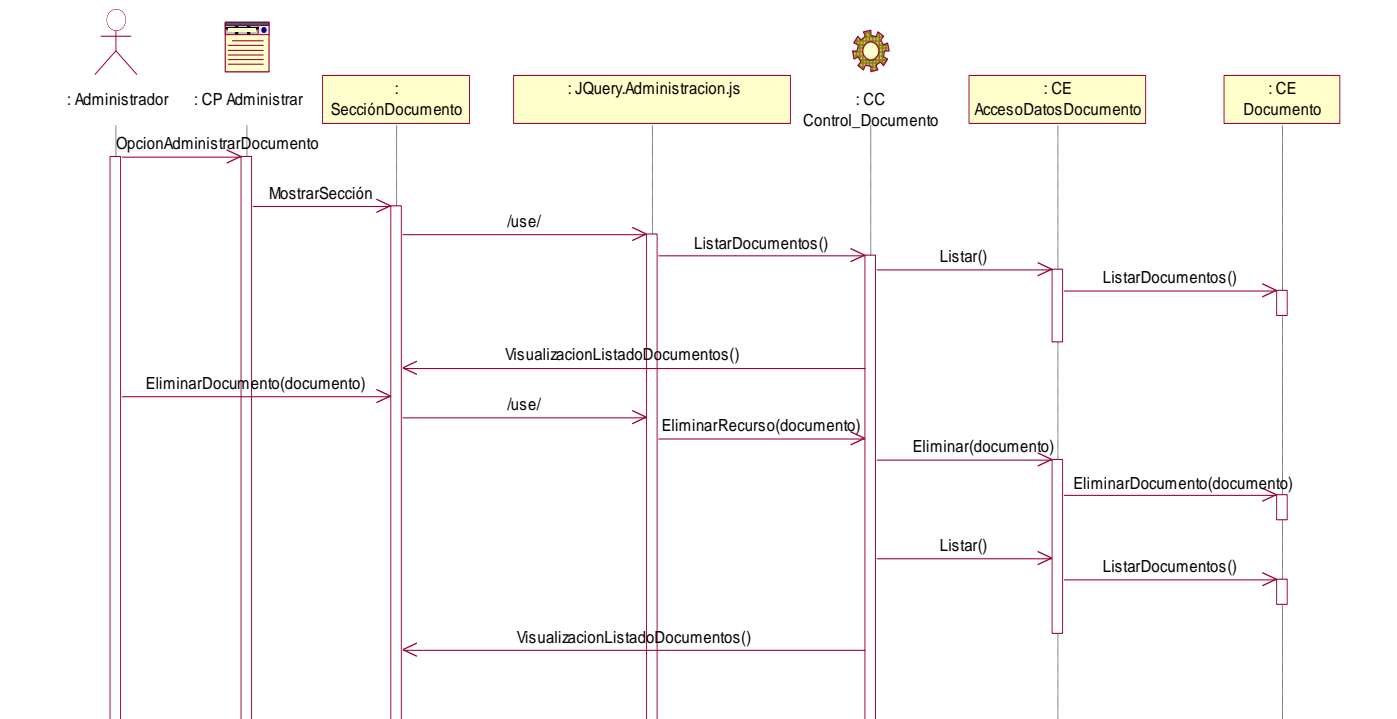
## Publicar concepto



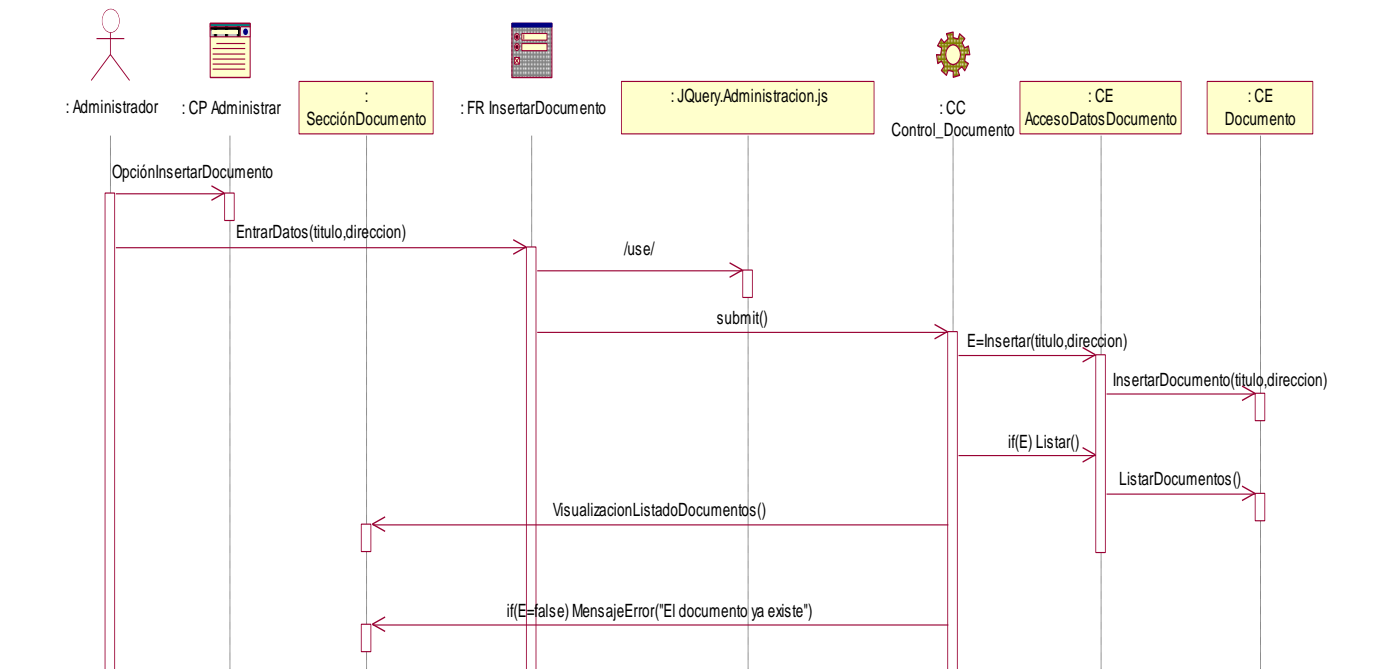
## Despublicar documento



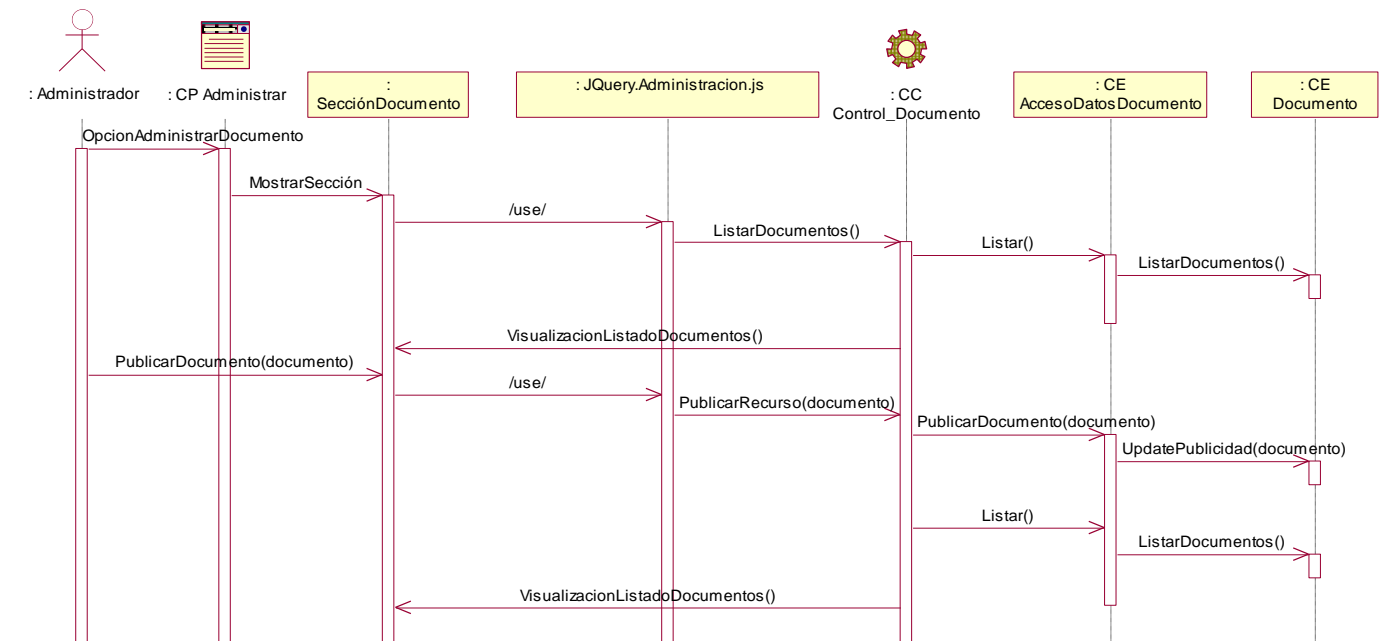
## Eliminar documento



## Insertar documento

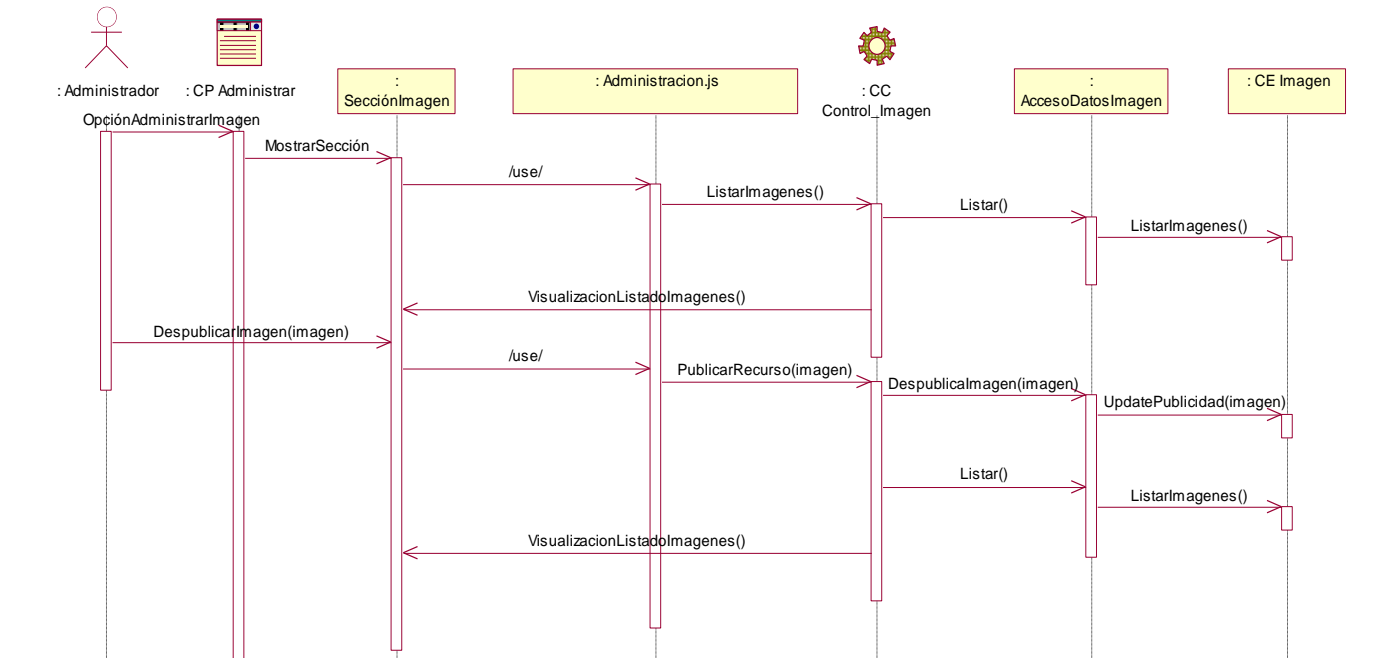


## Publicar documento

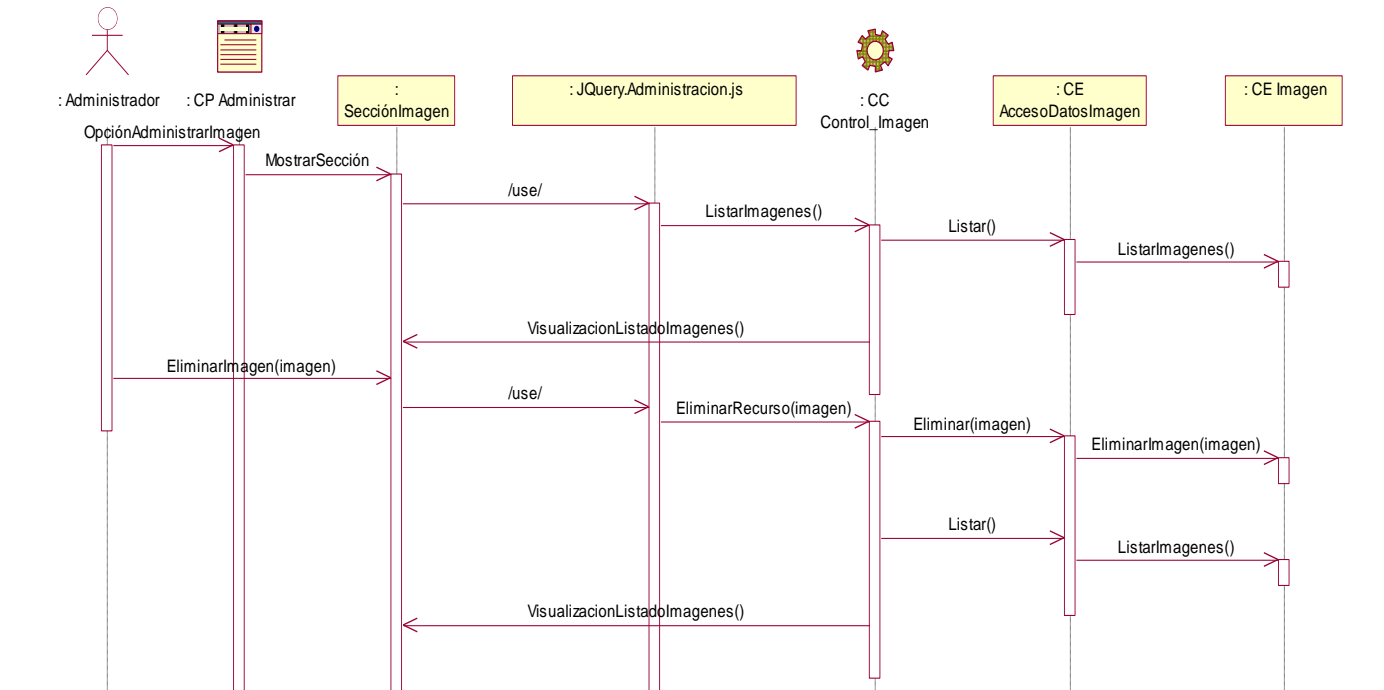




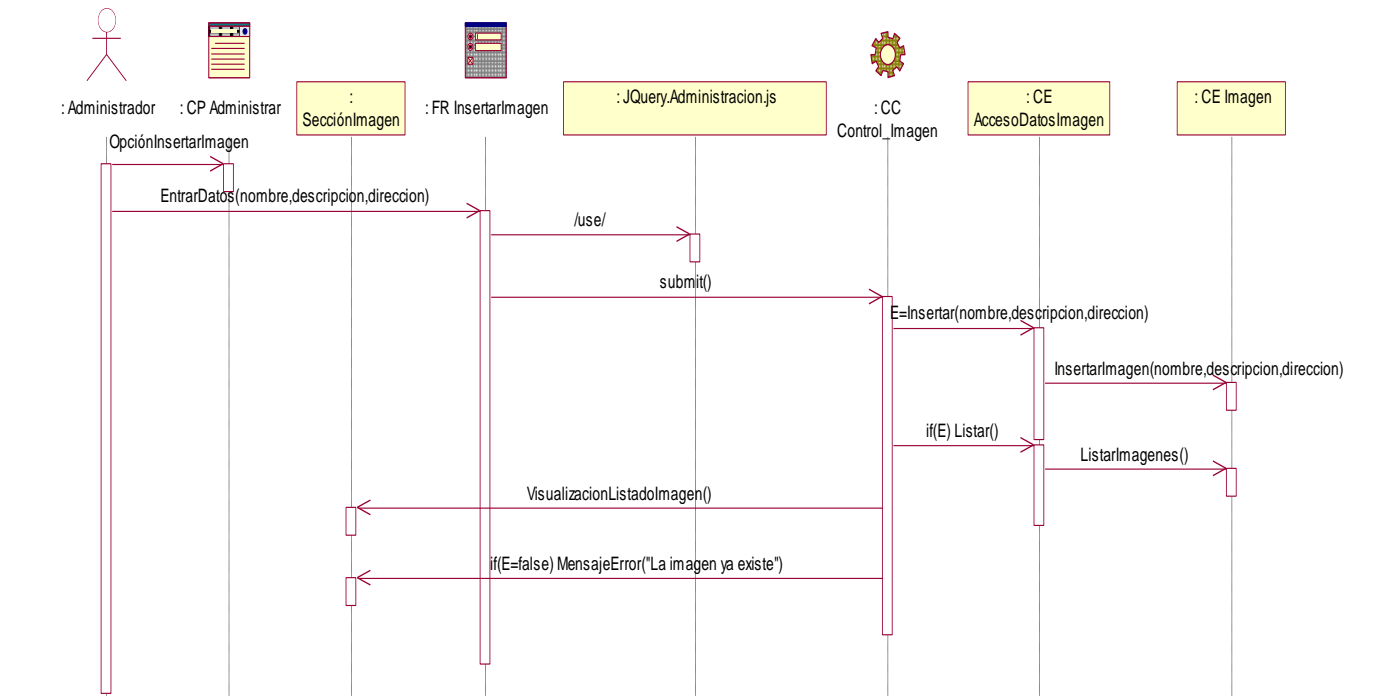
## Despublicar imagen



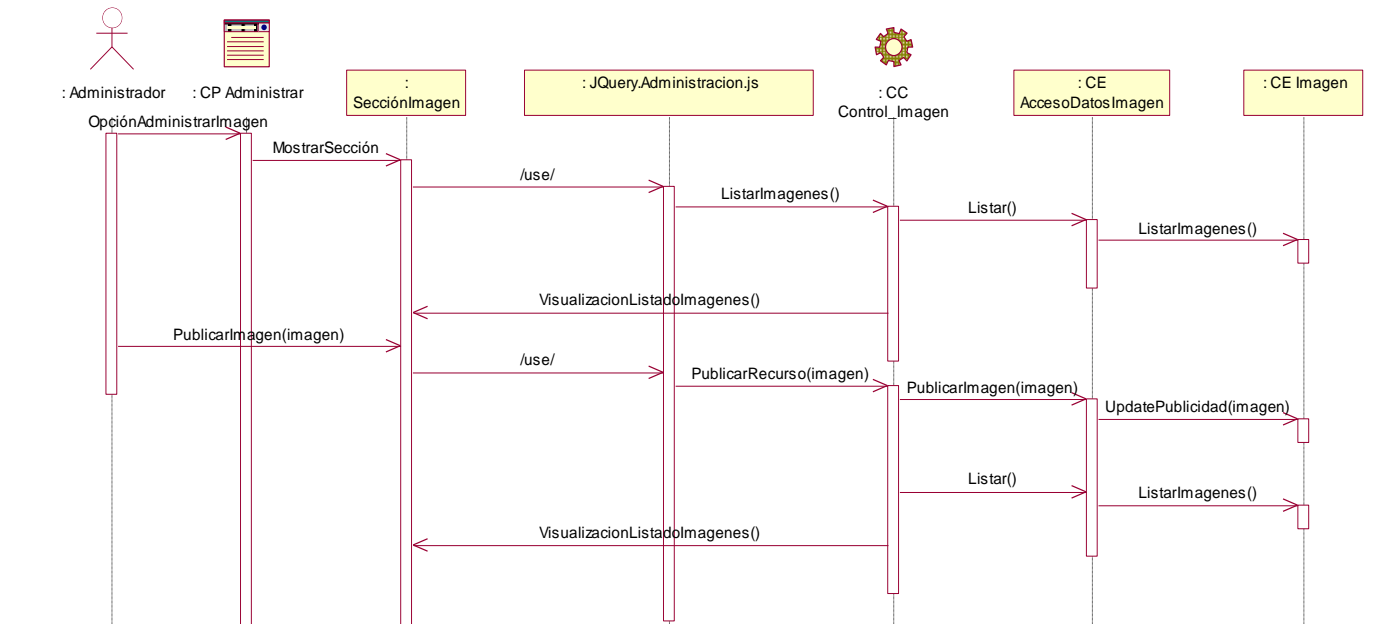
## Eliminar imagen



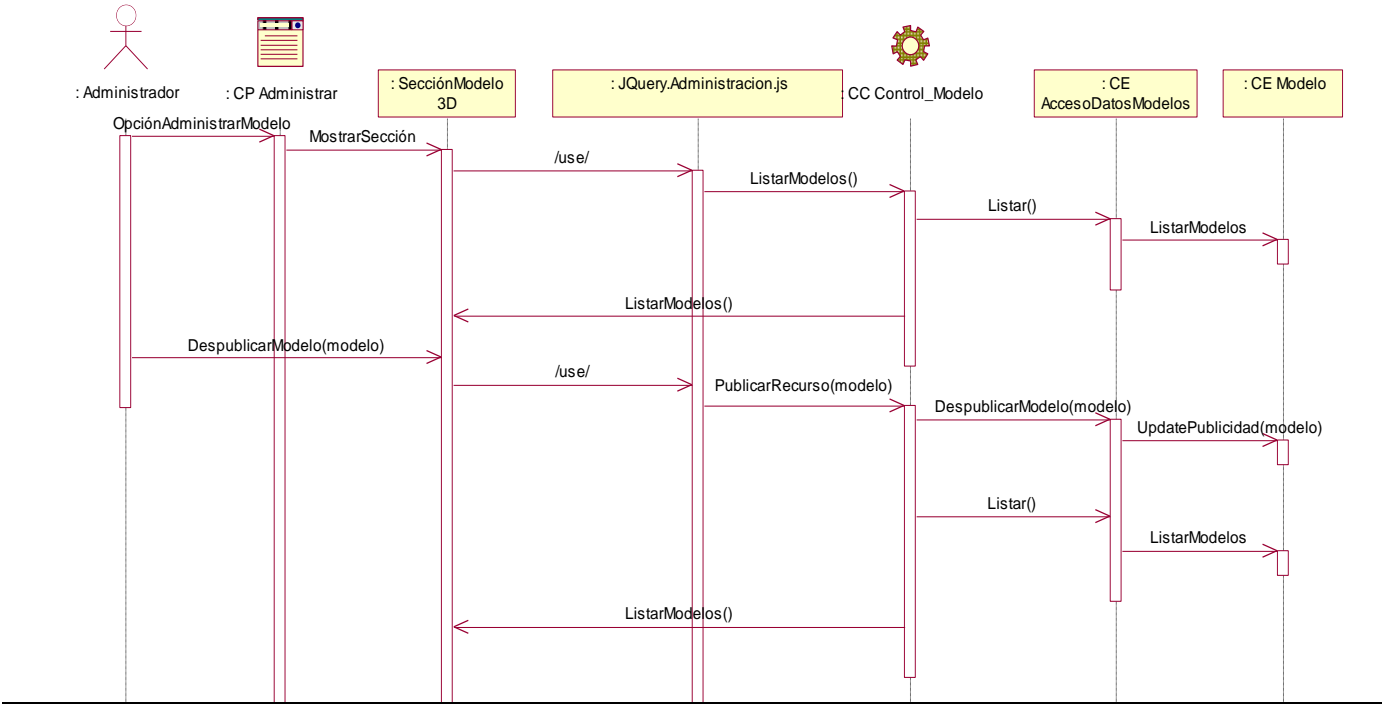
## Insertar imagen



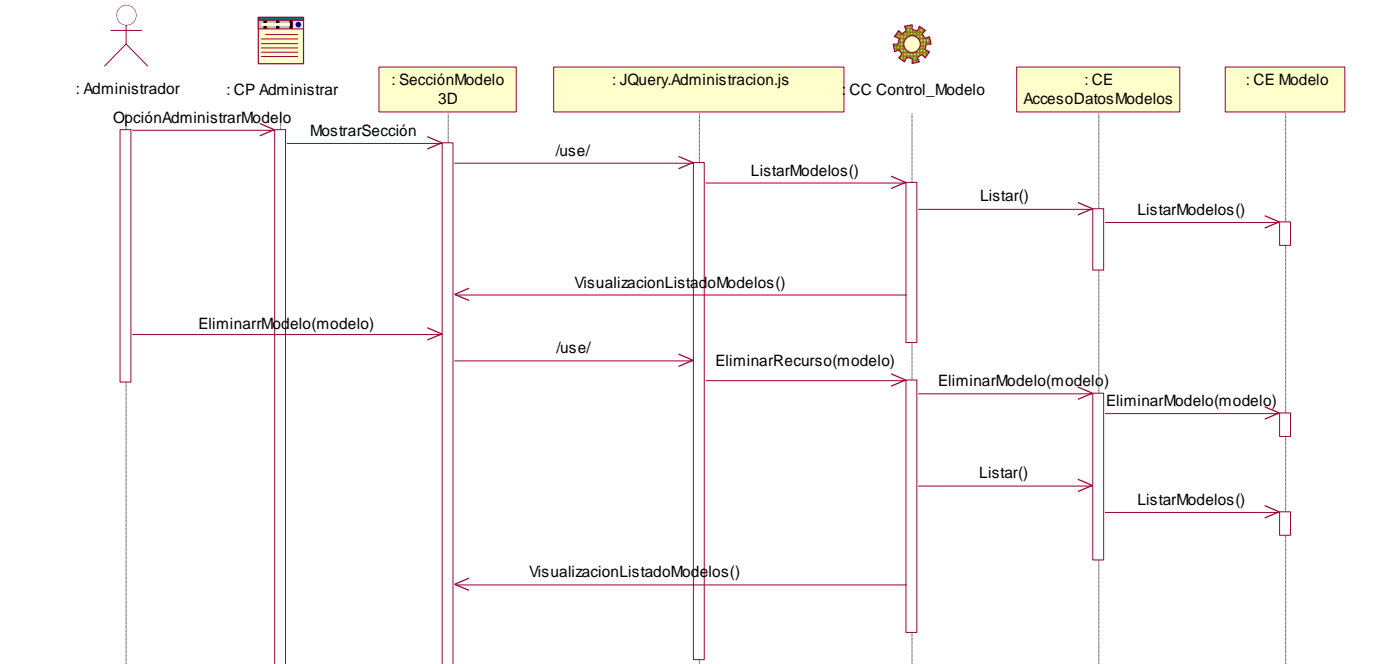
## Publicar imagen



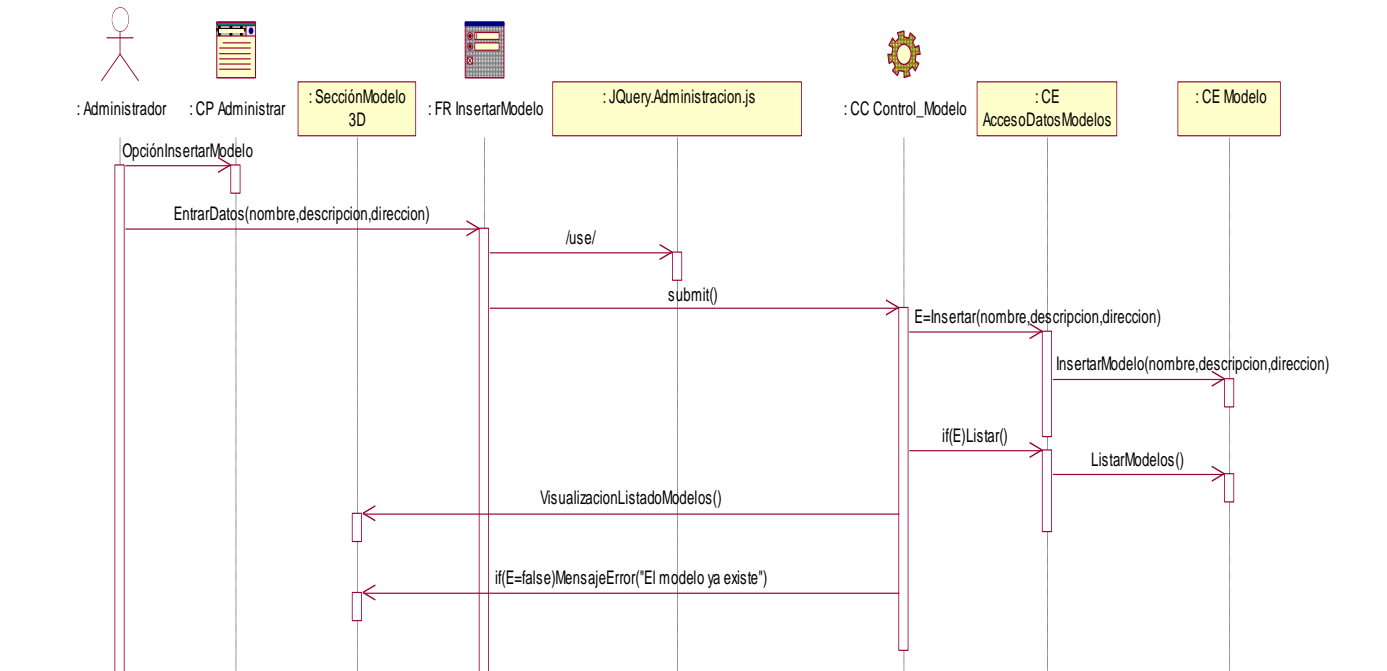
## Despublicar Modelo



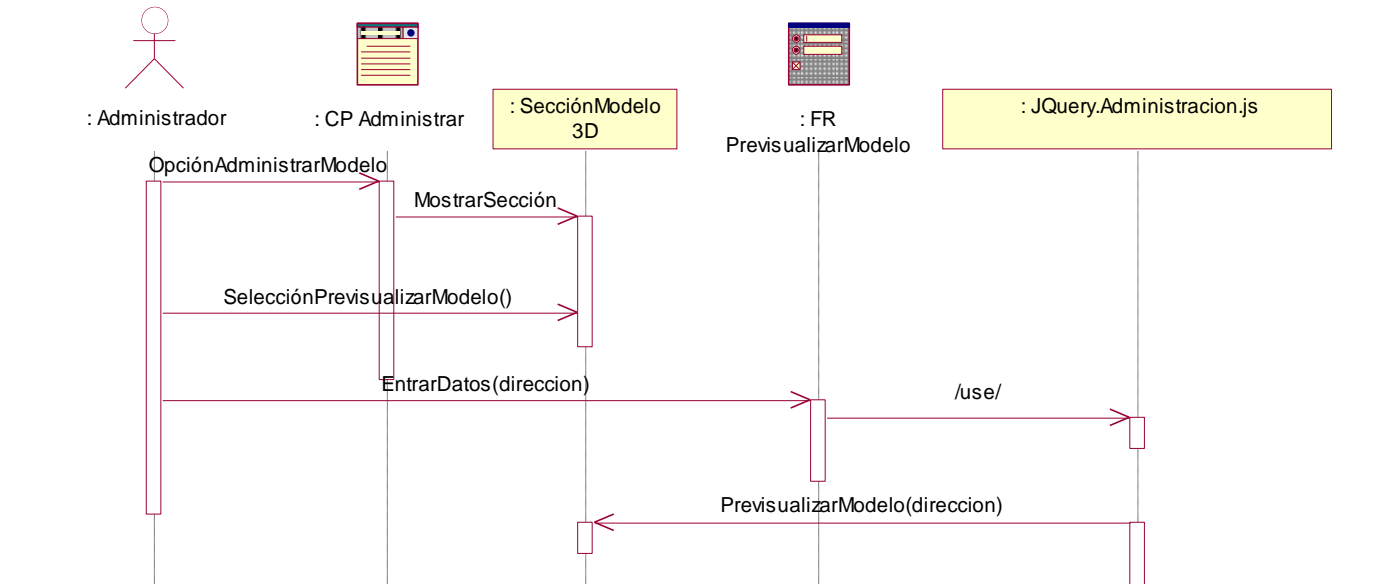
## Eliminar modelo



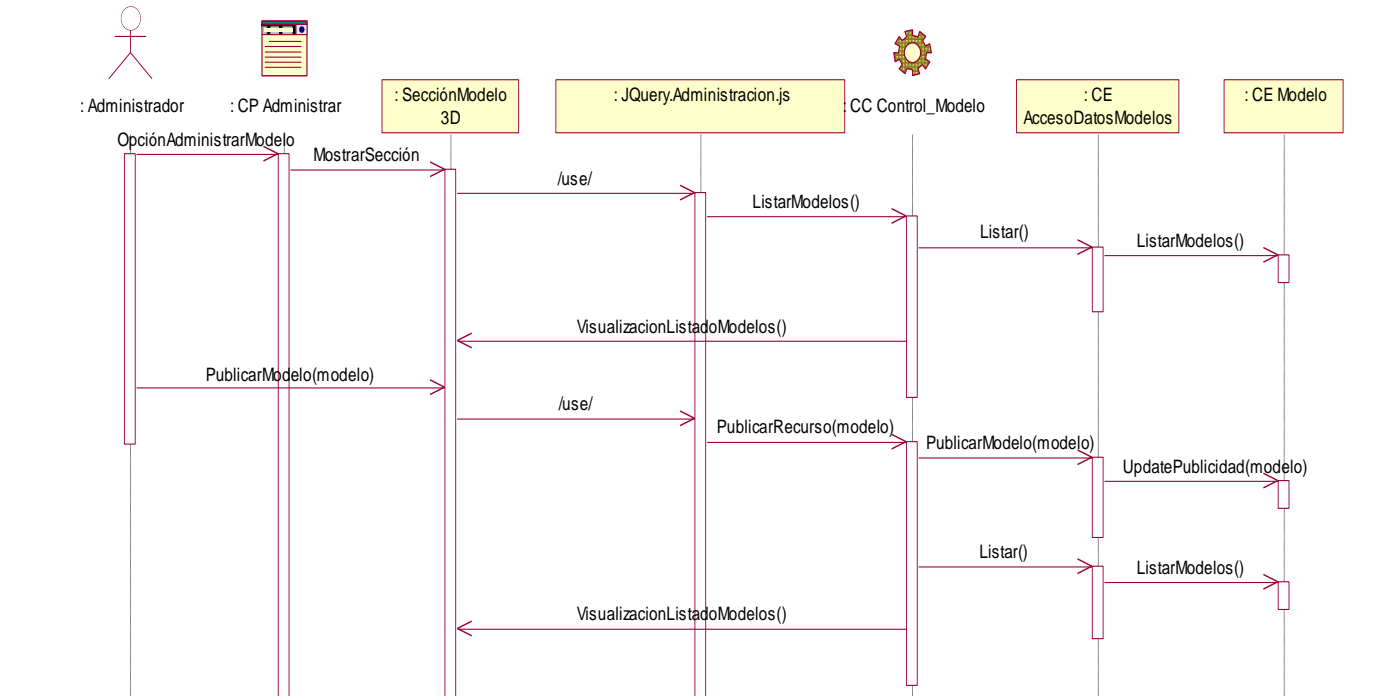
## Insertar modelo



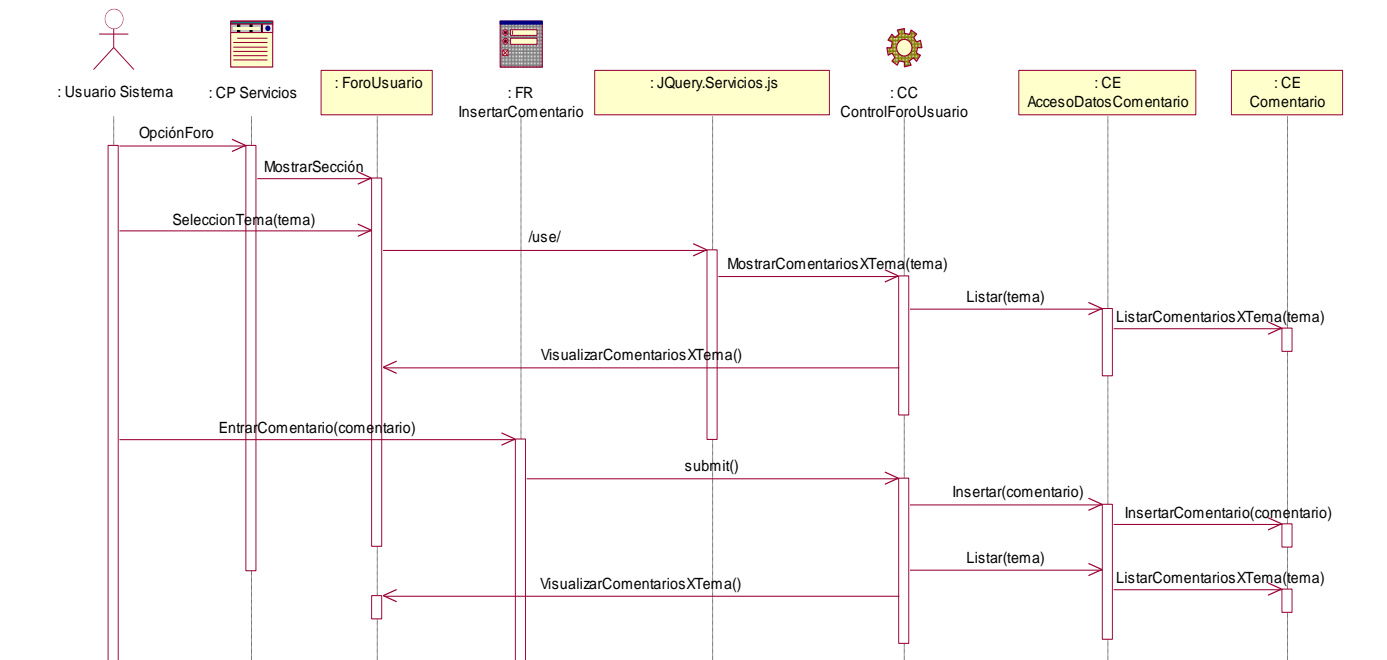
## Previsualizar Modelo



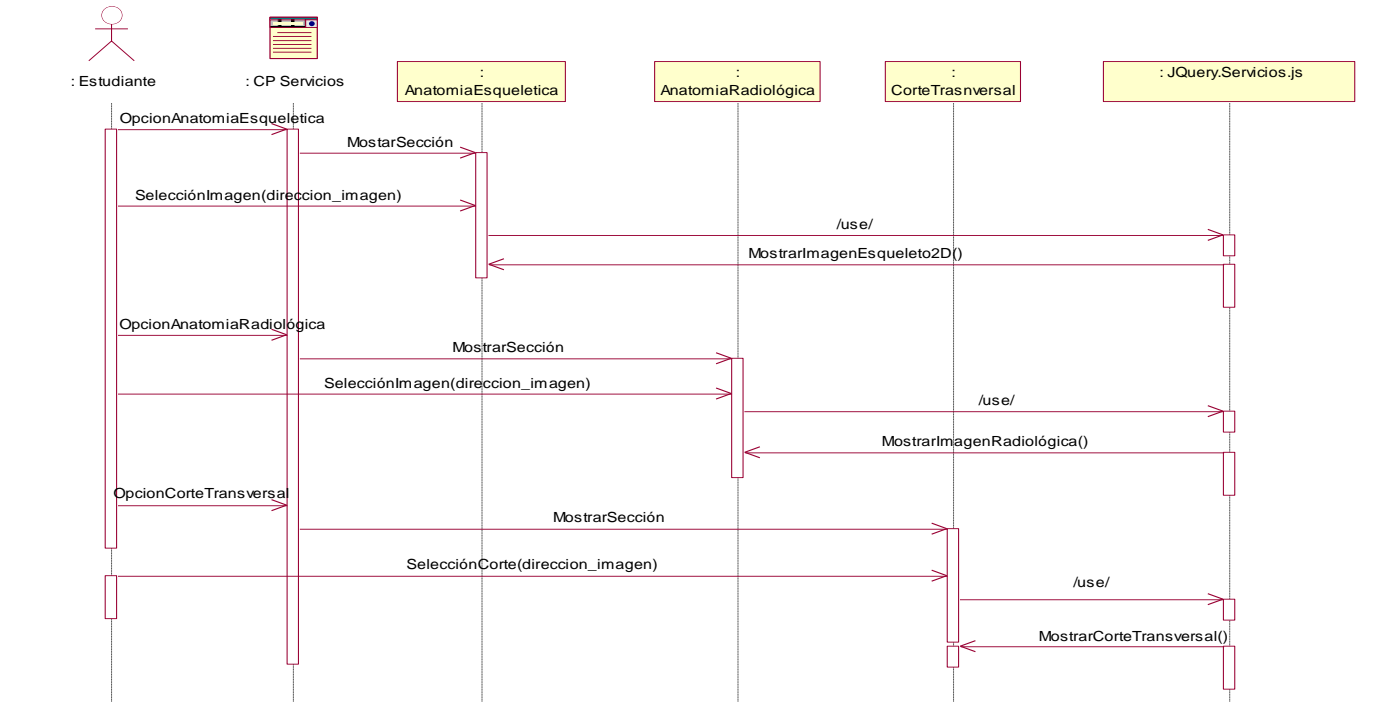
## Publicar modelo



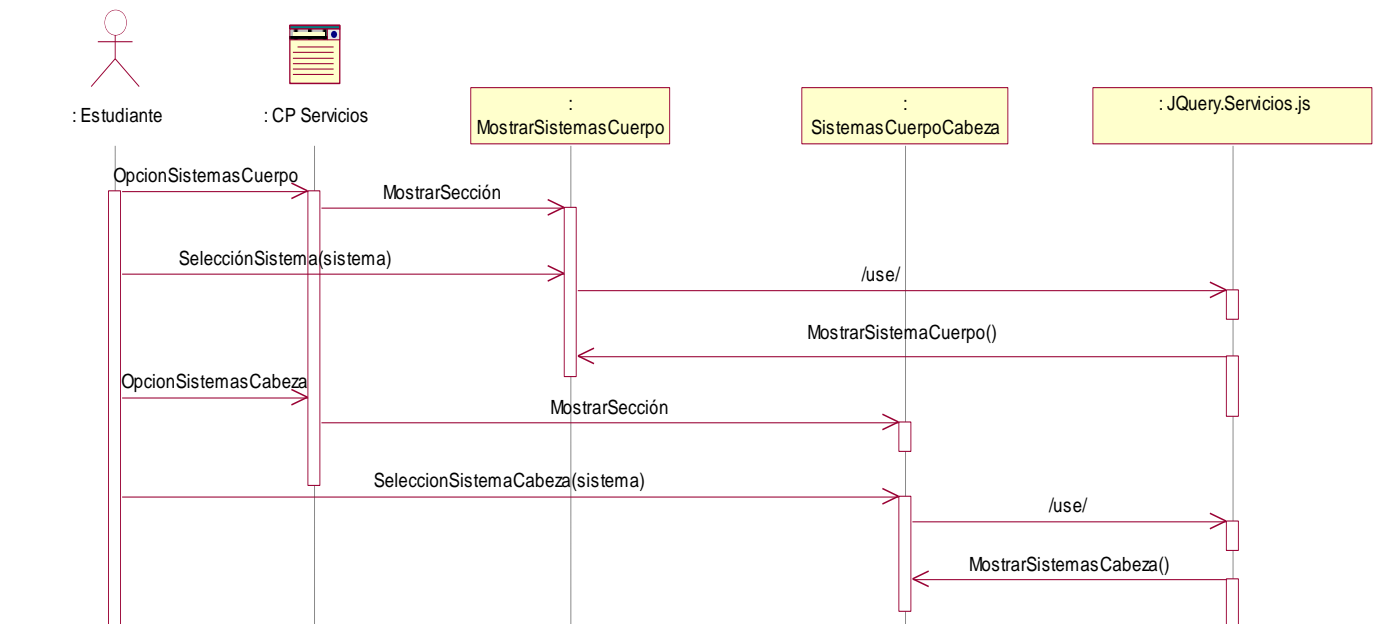
## Insertar comentario en el foro



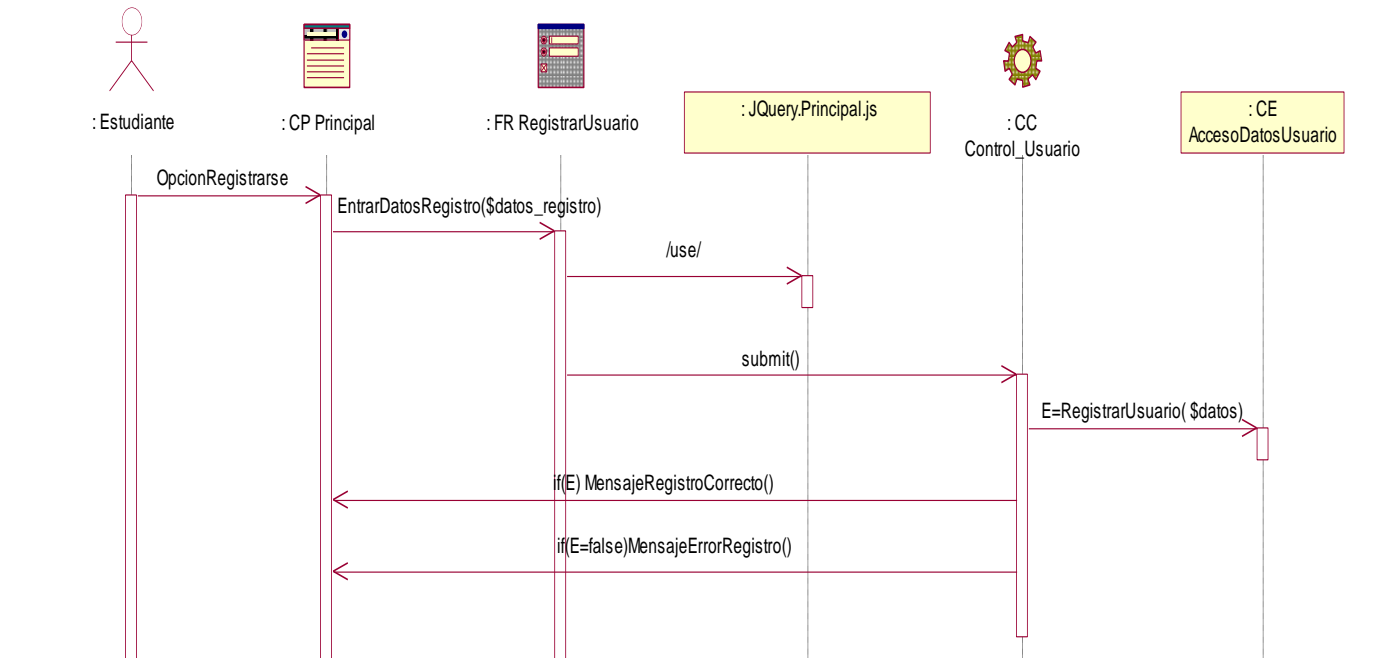
## Mostrar imágenes médicas



## Mostrar sistema del cuerpo



## Registrar usuario



## GLOSARIO DE TÉRMINOS

**Ajax:** Asynchronous JavaScript And XML no es una tecnología porque es la unión de un conjunto de tecnologías para el desarrollo WEB que se ejecutan en el cliente.

**Apache:** Es el servidor que se encargada de resolver las peticiones de los clientes de páginas Web.

**Aplicación Web:** Es una aplicación informática que los usuarios utilizan accediendo a un servidor Web a través de un navegador o browser. Estas son muy populares debido a la habilidad para actualizar y mantener la información manipulada sin distribuir e instalar el software en miles de potenciales clientes.

**Aptana Studio:** Es un IDE de desarrollo para aplicaciones de la WEB 2.0.

**Away3D:** Es un framework de código abierto basado en ActionScript, motor de renderizado 3D para el desarrollo de aplicaciones flash.

**BD:** Base de datos.

**CSS:** Cascading Style Sheets es la forma recomendada para controlar la presentación de la capa en un documento WEB.

**DICOM:** Digital Imaging and Communication in Medicine es el estándar reconocido mundialmente para el intercambio de imágenes médicas, pensado para el manejo, almacenamiento, impresión y transmisión de imágenes médicas.

**DOM:** Document Object Model, es una representación del modelo de un documento y su contenido porque permite verlo de una manera diferente describiendo el contenido del mismo en un conjunto de objetos que el lenguaje JavaScript puede actuar sobre ellos.

**Framework WEB:** Es un conjunto de componentes que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas WEB.

**Google O3D:** Es una API de código libre basada en JavaScript para crear mundos 3D.

**Hardware:** Componentes físicos de una computadora o de una red (a diferencia de los programas o elementos lógicos que los hacen funcionar).

**HTML:** Lenguaje de Marcas de Hipertexto, es un lenguaje para elaborar páginas Web.



**IDE:** Entorno integrado de desarrollo.

**Interfaz:** Es uno de los componentes más importantes de cualquier sistema computacional, funciona como el vínculo o comunicación entre el humano y la máquina.

**JavaScript:** Es un lenguaje interpretado que está del lado del cliente.

**Java 3D:** Es un API (Application Programming Interface) orientado a objetos para el lenguaje Java, que permite la confección de aplicaciones tridimensionales.

**JQuery:** Es un framework para el lenguaje JavaScript.

**Modelo 3D:** Es la unión de un conjunto de mallas, es como tal el resultado de todo el proceso de reconstrucción en la memoria.

**MySQL:** Es un sistema de base de datos relacional, multihilo, multiusuario que posee una arquitectura cliente servidor.

**PaperVision3D:** Es un motor de renderizado 3D en tiempo real, hecho en ActionScript 3.

**Patrón:** Es una solución a un problema de diseño no trivial que es efectiva (ya se resolvió el problema satisfactoriamente en ocasiones anteriores) y reusable (se puede aplicar a diferentes problemas de diseño en distintas circunstancias).

**PDO:** Librería de acceso a datos escrita en C originario de php5.

**PHP:** El lenguaje PHP (acrónimo de "PHP: Hypertext Preprocessor") es un lenguaje de script interpretado utilizado para la generación de páginas Web dinámicas, embebido en páginas HTML y ejecutado en el servidor.

**Plug-ins:** Aditamento para agregar a un equipo.

**PostgreSQL:** Es el más avanzado sistema de base datos relacional de código abierto con modelo de arquitectura de comunicación cliente servidor

**Rational Rose:** Es la herramienta CASE desarrollada por los autores de UML que cubre todo el ciclo de vida de un proyecto.

**Realidad Virtual:** Simulación generada por computadora de imágenes o ambientes tridimensionales interactivos con cierto grado de realismo físico o visual.

**RIA:** Rich Internet Applications se ejecutan desde un navegador como si fuesen una página WEB.

**RUP:** Proceso de desarrollo de software, metodología utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

**Sandy3D:** Librería open-source desarrollada en ActionScript para la visualización 3D en el entorno de flash.

**SGBD** Sistemas gestores de base de datos, tipo de software que se utilizan para servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

**UML:** “Unified Modelling Language”, lenguaje de modelado gráfico que permite especificar, construir, visualizar y documentar los artefactos de un sistema utilizando el enfoque orientado a objetos.

**Unity3D:** Es una potentísima herramienta de creación de entornos 3D, aplicaciones interactivas y aplicaciones WEB.

**Vivaty Studio:** Es una herramienta de diseño libre.

**VRML:** Virtual Reality Modeling Language es un lenguaje de especificaciones, tridimensional e interactivo, orientado a la modelación y la visualización de objetos, situaciones y mundos virtuales en Red.

**WEBGL:** Es una tecnología de visualización 3D que combina JavaScript, OpenGL y la etiqueta canvas, incorporada a HTML5.

**W3C:** Consorcio World Wide Web es una comunidad internacional y la misión es guiar la Web hacia su máximo potencial.

**XML:** Extensible Markup Language es un perfil de aplicación o forma restringida de (SGML).

**X3D:** Extensible Tridimensional es un estándar abierto XML que tiene un formato de archivo 3D que permite la creación y transmisión de datos 3D entre distintas aplicaciones.

**X3D-Edit:** Es una herramienta principalmente orienta a la programación utilizando el estándar. X3D.