Universidad de las Ciencias Informáticas

Facultad 10

Título: Análisis y diseño de un software para la gestión de licencias.

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

Autor: Dayan Font Hernández

Tutores: Lic. Marlen García Parrondo

Ing. Reynier Pérez Mira

Ing. Kiosmy Almenares Herrera

Ciudad de La Habana, julio 2008

Declaración de autoría

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

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

Autor

_____

Dayan Font Hernández

Tutores

_____ _____

Lic. Marlen García Parrondo Ing. Reynier Pérez Mira

_____

Ing. Kiosmy Almenares Herrera

Agradecimientos

Agradezco a todos los que de una forma u otra colaboraron en el desarrollo de este trabajo, a mis amigos por darme la oportunidad de poder llamarlos así, se que recordarlos sin tenerlos me hará daño. A mi familia, a mi Odalis por ser para mí calma y paciencia, a mi hermana por quererme como ni ella sabe que me quiere, a mamá por darme siempre su vida y a papá estar siempre, por dejarme sin palabras para expresar cuanto lo quiero. A todos gracias.

Dedicatoria

Este trabajo lo dedico a los que desde siempre no esperaron menos de mí, en especial a mis padres por poner toda su fe en mí aún cuando pensé que no me quedaban fuerzas.

Resumen

En el presente trabajo se expone el análisis y diseño de un sistema para la gestión de licencias de software en la Universidad de las Ciencias Informáticas. El motivo del presente, esta dado por la necesidad de mantener el control de las licencias que son usadas por los proyectos productivos en la institución. Como solución se propone un sistema web, para ello se hace un previo análisis investigativo de las principales tendencias actuales desde el punto de vista tecnológico y científico. En un segundo momento de la investigación se analiza a fondo la manera en que ocurren los procesos relacionados con la solicitud de licencias de software en la institución, se identifican en donde radican las ineficiencias y posteriormente se propone un sistema capaz de erradicarlas. Por último, se exhibe la modelación del sistema mediante los diagramas correspondientes a esta fase del desarrollo del software y se muestra una visión comprensible de las características que cumplirá el sistema propuesto. Como resultado se obtiene la propuesta del sistema que cumple todos los requisitos definidos por los clientes.

Palabras Claves: Gestión, Licencia de Software, Modelación, Sistema.

Índice

Introducción

Los servicios de información y documentación accesibles a través de internet, específicamente mediante servidores web, aumentan a una alta velocidad. La evolución de la web desde hace más de una década ha ido produciendo la sustitución de páginas y documentos estáticos por documentos generados dinámicamente, dependiendo la interacción del usuario con la lógica de procesos y flujos de trabajo definida por los creadores del servicio y a la disponibilidad de repositorios de información cada vez mayores. Evidentemente, se ha ido pasando progresivamente de un concepto de publicación de páginas web, muy simple en un principio, a esquemas más complejos y diferenciados, fundamentados en procedimientos y técnicas basados en la gestión de información. La creciente complejidad de los servicios y de los sistemas que los soportan ha hecho necesaria la formulación de un corpus teórico y práctico en el que se combinen las técnicas clásicas de gestión de información en las organizaciones con las características propias del medio digital.

Los sistemas de información presentan gran dependencia de quienes publican la información, de quienes la autorizan y procesan, por esta razón el contenido tarda en llegar al usuario, lógicamente la información tiene que pasar por manos de cada uno de los especialistas y es en este proceso donde se pierde un tiempo valioso para el usuario final, además de que muchas veces lo publicado se mantiene mucho tiempo después de su fecha de publicación, por tanto se torna completamente obsoleto para los usuarios que precisan de un constante dinamismo de la información requerida , por lo que afecta el desarrollo de su trabajo , y consecuentemente el de la entidad a la cual pertenece .Es por ello que las empresas desechan los sistemas de información que dependen de una pagina web estática y comienzan a utilizar sistemas de gestión de contenidos , que producen un flujo de información constante permitiendo a personas sin avanzados conocimientos en informática publicar, cambiar o eliminar contenidos del sitio , siempre sin violar los controles de calidad exigidos .

Los sistemas de gestión de contenidos tiene sus orígenes a mediados de la década de los 90, surgen a raíz de los problemas presentados por los sistemas de información mencionados anteriormente y específicamente por la necesidad del avance tecnológico, una de las principales empresas en desarrollar este tipo de software fue la Illustra Information Tecnology, que utiliza una base de datos, haciendo más asequible y provechoso el trabajo del usuario. Aunque se debe aclarar que cuando toman verdaderamente un importante auge los sistemas de gestión es a principios del año 2000.

En Cuba una de las entidades pioneras en el uso de este recurso es Infomed, institución que trabaja e interactúa bajo la política de los sistemas de gestión, aunque ya existen en el país diferentes

centros que utilizan estos tipos de software, hasta el momento no existe ningún software que específicamente sea para la gestión de licencias de software, o sea, no se conoce de un archivo de licencias de software donde los usuarios puedan solicitar servicios e intercambiar con la aplicación.

Este tipo de servicios en especifico es de vital importancia en una institución como la Universidad de las Ciencias Informáticas, en donde se vincula el estudio con la producción de software y en la cual se han creado algunos de gran nivel a escala internacional, en estos casos es un aspecto muy importante tener conocimiento del tipo de licencias que requiera el software que se produzca, además este proceso de selección, por así decirlo, se ha de llevar a cabo de manera rápida y eficaz.

En la actualidad es de vital importancia tener conocimiento de las licencias de software que se utilizan, ya sea libre o privativo, debido a que se han producido importantes desarrollos de software de código abierto y de la divulgación del mismo a través de Internet. Muchos de estos programas han adquirido un gran uso e importancia, entre otras cosas gracias a las ventajas y beneficios que otorgan las licencias por medio de las cuales son distribuidos. Es una realidad irrefutable que existe cierto desconocimiento con respecto al tema de las licencias.

Teniendo en cuanta la situación antes planteada nos enfrentamos a la siguiente interrogante ¿cómo lograr, mediante un sistema informático, una rápida gestión de los procesos de solicitud de licencias en la Universidad de las Ciencias Informática, así como lograr el control local y centralización de estos procesos?

Partiendo de la interrogante anterior se ha definido como **Objeto de Estudio** la gestión de Contenidos.

Se ha decidido como **Campo de Acción:** La gestión de Licencias de software.

Para lograr definir una línea base en la investigación se ha marcado como **objetivo general** diseñar un sistema de gestión de contenidos para el control de licencias en la Universidad de las Ciencias Informáticas.

Para dar cumplimiento al objetivo general que se propuso con anterioridad se han trazado los siguientes **objetivos específicos**:

- Levantar los requisitos que va a cumplir la aplicación.
- Describir los procesos que se van a implementar en la aplicación.

- Determinar los ciclos de desarrollo.
- Modelar conceptualmente las clases implicadas en la aplicación.
- Desarrollar los diagramas que describen el diseño web de la aplicación.
- Describir las clases del diseño.

Para que se lleve a cabo un exitoso cumplimiento de los objetivos planteados se proponen las siguientes **tareas de investigación:**

- Verificar la existencia de soluciones o aplicaciones similares.
- Selección de una metodología de desarrollo que se adecue al desarrollo del sistema.
- Análisis y diseño de la aplicación.

Con el objetivo de desarrollar estas tareas se trabajará con algunos **métodos de investigación,** estos son:

Los teóricos:

Histórico y lógico

Este se empleó para analizar la manera en la que ha evolucionado el desarrollo de los proyectos desde el inicio de los procesos automatizados y como ha ido en aumento la dependencia de estos en las diferentes áreas de trabajo, lo que lógicamente permite la culminación de los trabajos en menor tiempo y con mayor calidad.

Inductivo y deductivo

Se empló para estudiar algunos casos aislados de proyectos que tuvieron problemas con la entrega del producto. Se encontró que la gestión de licencias jugaba un papel importante en esta situación y este era un problema que está vigente en casi todos los proyectos que se desarrollan y por lo tanto es un problema que golpea a nuestra universidad donde se desarrollan numerosos software.

Los empíricos:

Método de entrevista:

Este método se utilizó para lograr un mayor acercamiento con los clientes con el objetivo de tener mayor dominio del negocio en cuestión, así como para obtener información relacionada con el tema de alguno de los especialistas de la facultad.

Para un mayor entendimiento de este trabajo y con el objetivo de lograr mejor claridad en las ideas, a continuación se muestra la estructura capitular de la tesis.

Capítulo 1: “Fundamentación Teórica”, en este capítulo se mencionan los aspectos teóricos utilizados para el desarrollo de la investigación, que sirven de apoyo para la mejor comprensión del presente trabajo. Se hace referencia a los diferentes conceptos que cimentan la investigación, qué es una licencia, tipos de licencias, qué es la gestión de la información y junto con ello la gestión de contenidos. Se hace un análisis de las diferentes herramientas existentes para el desarrollo de la aplicación, así como la metodología de desarrollo más efectiva para realizar el trabajo.

Capítulo 2: “Propuesta del sistema”, se describen las características del sistema dando a conocer las herramientas con las que se pretende dar solución al problema de investigación planteado, además se realiza un modelo de negocio en el cual se plantean los principales conceptos de la aplicación. Se determina la captura de requisitos, así como los principales casos de uso de los cuales se ofrece una descripción textual.

Capítulo 3: “Análisis y Diseño del sistema”, se describe el sistema a través de las fases de Análisis y Diseño, que propone la metodología RUP para su desarrollo.

Capítulo I: Fundamentación teórica

Es necesario para obtener una correcta calidad del producto, estudiar con detenimiento cuáles son las tecnologías que en la actualidad se usan a nivel internacional y nacional para realizar software, una vez hecho este estudio, se intentó escoger las mas adecuadas para su posterior usanza.

A continuación además de aclarar términos que se consideran relevantes para comprender con claridad esta investigación, se analizan cuales son las tecnologías actuales que clasifican para desarrollar el sistema a desarrollar en este trabajo, se exponen las ventajas y desventajas que presenten, teniendo en cuenta que de una correcta elección dependerá la futura calidad del producto.

Por otra parte es imprescindible tener dominio del contenido alrededor del cual girará el sistema que se desee desarrollar, haciendo un estudio previo de este. Por tal motivo es que en el presente trabajo de forma abreviada pero clara se definen aspectos importantes relacionados con las licencias.

1.1 Soluciones informáticas existentes.

Actualmente existen en la Universidad de las Ciencias Informáticas sistemas para automatizar procesos relacionados con el que hacer diario de esta, pero no se cuenta con una aplicación capaz de agrupar toda la información asociada a los proceso de solicitud de licencias por parte de los proyectos productivos. En el ámbito internacional tampoco se tiene referencia de alguna institución que utilices un aplicación con tal fin. Por lo que la presente investigación es posiblemente el primer paso que se de para dicho propósito.

1.2 ¿Qué es una Licencia de software?

Licencia: contrato entre el desarrollador de un software sometido a propiedad intelectual y a derechos de autor y el usuario, en el cual se definen con precisión los derechos y deberes de ambas partes. Es el desarrollador, o aquel a quien éste haya cedido los derechos de explotación, quien elige la licencia según la cual distribuye el software[1].

[1] **Labrador, Ramón M. Gómez.** Tipos de Licencias de Software. 2005. http://www.informatica.us.es/~ramon/articulos/LicenciasSoftware.pdf.

1.2.1 Tipos de Licencias

Dentro de las licencias de software existen dos grupos las licencias para software libre y las licencias para software no libre. En el primer de los casos los consumidores de los productos distribuidos bajos esas condiciones tienen acceso al código fuente de estos teniendo la posibilidad de hacerle cambios adaptándolo a sus necesidades, así como redistribuir el producto una vez modificado. Para el segundo grupo es importante destacar que estas licencias también se conocen con el nombre de software propietario. En ellas los propietarios establecen los derechos de uso, distribución, redistribución, copia, modificación, cesión y en general cualquier otra consideración que se estime necesaria. Este tipo de licencias, por lo general, no permiten que el software sea modificado, desensamblado, copiado o distribuido de forma ilegal (piratería de software), regula el número de copias que pueden ser instaladas e incluso los fines concretos para los cuales puede ser utilizado. La mayoría de estas licencias limitan fuertemente la responsabilidad derivada de fallos en el programa. Los fabricantes de programas sometidos a este tipo de licencias por lo general ofrecen servicios de soporte técnico y actualizaciones durante el tiempo de vida del producto.

Si se hace un análisis de lo antes planteado se puede concluir el importante rol que juegan las licencias en la producción de software, así como la responsabilidad que implica hacer la selección de estas para un proyecto en específico. Por lo tanto se considera que se ha de hacer un estudio de los sistemas que se especializan en brindar este tipo de servicios.

1.3 Sistemas de Información.

Un sistema de información es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio.[2]

A partir de este reconocimiento los sistemas de información comienzan a formar parte de las estrategias de las organizaciones, por lo que tienen una alta prioridad y son la base de la eficiencia y eficacia de las organizaciones que así los consideran.

Una organización puede adquirir nuevas máquinas computadoras, instalar nuevos productos de telecomunicaciones, elaborar una página web, realizar comercio electrónico, pero ello no implica, que exista en su organización un sistema de información.

[2] **Peralta, Manuel**. Sistema de Información. [En línea] 1997. http://www.monografias.com/trabajos7/sisinf/sisinf.shtml#esi.

Junto a la información, los otros componentes básicos que constituyen a un sistema de información son: los usuarios, los equipos informáticos (software, hardware y tecnologías de almacenamiento de la información y de las telecomunicaciones) y los especialistas.

Para comprender los Sistemas de Información hay que conocer que existen necesidades en las organizaciones y comunidades que deben ser satisfechas, hay que dominar las complejidades de cómo se maneja la información y cuáles son las potencialidades de los medios que se emplean para organizarla y recuperarla.

El manejo de datos e información constituye uno de los aspectos más importantes para cualquier organización, abarcando diferentes actividades como la recolección, almacenamiento, recuperación, difusión hacia lugares y personas indicadas, así como el uso que de ellos se hace para varias actividades dentro de una organización.

Por lo general los Sistemas de Información son grandes y complejos debido a la diversidad de sus componentes, procesos y las relaciones que existen entre ellos, caracterizándose por la capacidad que tienen de adaptarse al medio.

Al mencionar los procesos que ocurren en los Sistemas de Información, se consideran aquellos procesos que están asociados a los recursos de información y a cómo estos aumentan la probabilidad de que los usuarios en un ambiente dado encuentren mensajes útiles en las salidas de estos sistemas.

Las facilidades que pudiera llegar a dar este tipo de servicio informático dependen en gran medida de la manera en que se conciba la aplicación, por lo que se considera de carácter obligatorio llevar a cabo un estudio de las tendencia tecnológicas actuales, para que de esta manera se pueda hacer una selección lo mas aceptada posible de las herramientas y otros elementos influyentes.

1.4 Tendencias y Tecnologías Actuales.

En el presente epígrafe se hará un análisis y estudio de las tecnologías actuales con el objetivo de contar con los elementos necesarios para dar una propuesta lo mas eficiente posible de las tecnologías destinadas a ser usadas en el desarrollo de la aplicación.

1.4.1 ¿Qué es un sitio Web?

Un sitio Web es un conjunto de archivos electrónicos y páginas Web referentes a un tema en particular, que incluye una página inicial de bienvenida, generalmente denominada home page, con un nombre de dominio y dirección en Internet específicos. Empleados por las instituciones públicas y privadas, organizaciones e individuos para comunicarse con el mundo entero[3].

Ventajas principales de los sitios Web:

- Son sistemas multiplataforma, ya que se puede migrar de sistema y cambiar el hardware libremente y no afecta el funcionamiento de las aplicaciones del servidor.
- Se necesita solo una computadora con un navegador Web para el uso de estos ya que no se requiere de altas complicaciones hardware/software.
- El usuario no se da cuenta de las actualizaciones del software ya que estas se realizan automáticamente.
- Proporciona facilidad en el trabajo a distancia ya que el acceso es a través de la red.

Desventajas principales de los sitios Web:

- Difícil implementación de la seguridad.
- Es necesaria una conexión rápida y permanente.
- Existen limitaciones en los documentos estándar HTML.

Este tipo de aplicación puede ser desarrollada en diversos lenguajes de los cuales se hará un estudio con el fin de lograr explotar al máximo las bondades de las aplicaciones Web.

1.4.2 Lenguajes de programación Web.

Un lenguaje de programación es un lenguaje que se utiliza para controlar el comportamiento de una computadora. Es un grupo de reglas sintácticas y semánticas que definen su estructura y significado. Existen muchos lenguajes de programación, y cada uno tiene características que definen los campos en los que pueden ser implementados. A continuación se verán algunos de ellos, específicamente aquellos que se utilizan para implementar aplicaciones Web.

[3] **Anónimo.** Milenium. 2008. http://www.informaticamilenium.com.mx/Paginas/espanol/sitioweb.htm.

Personal Home Page (PHP)

PHP (Personal Home Page) es el acrónimo recursivo de PHP: Hypertext Preprocessor.

PHP es un lenguaje de programación usado generalmente para la creación de contenidos para sitios Web.

Este lenguaje se acopla al HTML (páginas web) para definir procedimientos que ha de realizar el servidor de Web, por ejemplo procesar un formulario, enviar o extraer datos de una base de datos (acoplándose también con un lenguaje de tipo SQL), enviar una u otra página Web según determinadas condiciones prefijadas por el programador, etc.

Últimamente también se usa para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráfica usando la biblioteca GTK+.

PHP fue originalmente diseñado en Perl, seguidos por la escritura de un grupo de CGI binarios escritos en el lenguaje C por programador Danés – Canadiense Rasmus Lerdorf en el año 1994, pero como PHP está desarrollado en política de código abierto muchos desarrolladores han hecho sus contribuciones, siendo la PHP 5.2.3 la versión más reciente. Esta versión incluye PDO (Objetos de Información de PHP o PHP Data Objects) y mejoras utilizando las ventajas que provee el nuevo Zend Engine 2.

PHP es gratuito e independiente de la plataforma, pues existe un módulo de PHP para casi cualquier servidor Web. Esto hace que cualquier sistema pueda ser compatible con el lenguaje, lo que significa una ventaja importante, ya que permite portar el sitio desarrollado en PHP de un sistema a otro sin prácticamente ningún trabajo. Además, es seguro, con gran librería de funciones y mucha documentación.

Es un lenguaje multiplataforma que puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo Linux, muchas variantes Unix (incluido HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS.

La característica más potente de PHP es su soporte para una gran cantidad de bases de datos que se utilizan actualmente, destacando su conectividad con MySQL.

Agrupa en su última versión 5.2.3 una serie de ventajas tales como:

- Soporte sólido para Programación Orientada a Objetos con PHP Data Objects.

- Es un software totalmente libre.
- Mejoras de rendimiento.
- Mejor soporte para MySQL con extensión de rewrite completa.
- Iteradores de datos.
- Excepciones de errores.

PERL

PERL, Lenguaje Práctico para la Extracción e Informe, es un lenguaje especializado en el procesamiento de textos, particularmente, extraer y validar las respuestas a cuestionarios incluidos en páginas Web.

Perl es un lenguaje interpretado que tiene muchas utilidades, pero está orientado principalmente a la búsqueda, extracción y formateo de ficheros de tipo texto. También es muy usado para el manejo y gestión de procesos (estado de procesos, conteo y extracción de parámetros característicos, etc.). Aunque desarrollado originalmente en un entorno UNIX, actualmente hay versiones para casi todos los sistemas operativos: Windows, Mc OS, de manera que es un lenguaje multiplataforma. Es gratuito, libre y uno de los lenguajes más utilizados en la programación de CGI script para el intercambio de información entre aplicaciones externas y servicios de información. Tiene la desventaja de ser un lenguaje que consume muchos recursos de la máquina, lo que significa que no es muy ligero. Si no se escribe con cuidado puede llegar a ser un código ilegible y no se pueden compilar programas con Perl.

ASP.

ASP (Active Server Pages - Página Activa en el Servidor) es una tecnología del lado servidor desarrollado por Microsoft para páginas Web generadas dinámicamente.

La tecnología ASP está estrechamente relacionada con el modelo tecnológico de su fabricante. Intenta ser solución para un modelo de programación rápida ya que programar en ASP es como programar en VisualBasic, pero con muchas limitaciones.

ASP no es en sí mismo un lenguaje de programación, si no más bien un marco sobre el que se construyen aplicaciones basadas en Internet, apoyándose para ello en el lenguaje HTML, en lenguajes de script conocidos (generalmente VBScript, pero también JavaScript–Jscript para Microsoft-, Perl, y otros.), en motores de bases de datos y en el lenguaje de consulta SQL. ASP ha pasado por cuatro iteraciones mayores, ASP 1.0 (distribuido con IIS 3.0), ASP 2.0 (distribuido con IIS 4.0), ASP 3.0 (distribuido con IIS 5.0) y ASP.NET (parte de la plataforma .NET de Microsoft). Las versiones PRE-.NET se denominan actualmente (desde 2002) como ASP clásico. (Anónimo, 2004)

Evidentemente una de las desventajas es que su empleo se realiza solo sobre plataformas funcionando bajo sistema Windows NT.

ASP.NET

Microsoft ASP.NET es una tecnología libre que permite a los programadores crear aplicaciones web dinámicas. ASP.NET puede utilizarse para crear cualquier cosa, desde pequeñas aplicaciones personales, hasta grandes sitos de empresas. Pude ejecutarse de formas distintas las cuales son:

Aplicaciones cliente/servidor: Estas aplicaciones están típicamente en formato de ejecutables compilados. Estos pueden integrar toda la riqueza de una interfaz de usuario, tal es el caso de las aplicaciones de desempeño y productividad, pero no se reúne la lógica de negocio como un recurso que se pueda reutilizar. Además acostumbran ser menos gestionables y escalables que las demás aplicaciones.

Aplicaciones que utilizan el navegador: Dichas aplicaciones están caracterizadas por contar con una interfaz de web rica y muy útil. La interfaz gráfica integra varias tecnologías, las cuales son el HTML, XHTML, scripting, etc; siempre y cuando el navegador que se esté utilizando soporte estas tecnologías.

JSP.

Java Server Pages (JSP – Páginas de Servidor Java) es la tecnología para generar páginas Web de forma dinámica en el servidor, desarrollado por Sun Microsystems basado en script que utilizan una variante del lenguaje Java. La tecnología JSP es una tecnología Java que permite a los programadores generar dinámicamente HTML, XML o algún otro tipo de página Web. Esta tecnología permite al código Java y a algunas acciones predefinidas ser embebidas en el contenido estático. En las JSP se escribe el texto que va a ser devuelto en la salida

(normalmente en código HTML) incluyendo código Java dentro de él para poder modificar o generar contenido dinámicamente.

Java Script.

JavaScript es un lenguaje de programación interpretado del lado del cliente porque es el navegador el que soporta la carga de procesamiento. Únicamente depende del cliente y no del sistema operativo, sólo necesita un browser capaz de interpretarlo .Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado. Cualquier persona puede desarrollar aplicaciones escritas en JavaScript del mismo modo que realiza páginas HTML. Por otro lado, JavaScript no es un lenguaje válido para desarrollar aplicaciones concurrentes o de acceso compartido. Está diseñado para controlar la apariencia y manipular los eventos dentro de la ventana del navegador Web. Este se integra directamente en páginas HTML (como programas propiamente dichos, combinando funciones y sentencias, e introduciendo manejadores de eventos JavaScript en etiquetas HTML) y la ventaja que presenta sobre el HTML es que permite crear páginas Web más dinámicas, lo que las hace más atractivas para el usuario.

HTML

El **HTML** (lenguaje de marcas hipertextuales), es un lenguaje de marcado diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web. Se ha convertido en uno de los formatos más populares que existen para la construcción de documentos y también de los más fáciles de aprender. El lenguaje HTML puede ser creado y editado con cualquier editor de textos básico, como puede ser el Bloc de Notas de Windows, o cualquier otro editor que admita texto sin formato. Uno de los principales problemas es el tiempo que se tarda en escribir un código eficiente HTML a mano. Además los elementos estándar HTML tienen también algunas limitaciones. Dependiendo del navegador que se use, pueden aparecer diferentes para varios usuarios. Esto puede causar alguna confusión si los usuarios se mueven de una estación de trabajo a otra diferente. HTML no es propiamente un lenguaje de programación como PHP, sino un sistema de etiquetas. No presenta ningún compilador, por lo tanto algún error de sintaxis que se presente éste no lo detectará y se visualizará en la forma como éste lo entienda.

A modo de resumen podemos decir que los lenguajes son solo conocimiento que puede ser adquirido de forma libre. Estos no servirían de nada de no ser por los IDE a los que se hace referencia en el siguiente epígrafe.

1.4.3 Entornos de Programación

Los entornos integrados de desarrollo o IDE (acrónimo de Integrated Development Environment), son aplicaciones compuestas por un conjunto de herramientas útiles para un programador. Estos pueden ser exclusivos para un lenguaje de programación o bien para varios. Estos suelen estar formados por un editor de código, un compilador, un depurador y un constructor de interfaz grafica (GUI).

Zend Studio.

Un programa orientado a desarrollar aplicaciones web, en lenguaje PHP. El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código.

El programa entero está escrito en Java, lo que a veces supone que no funcione tan rápido como otras aplicaciones de uso diario. Sin embargo, esto ha permitido a Zend lanzar con relativa facilidad y rapidez versiones del producto para Windows, Linux y MacOS, aunque el desarrollo de las versiones de este último sistema se retrase un poco más.

La parte del programa que nos permite escribir los scripts es bastante útil para la programación en PHP. La interfaz está compuesta por varias partes, en las que encontramos un explorador de archivos, una ventana de depuración, los menús y otra para mostrar el código de las páginas.

Lo más destacable es que contiene una ayuda contextual con todas las librerías de funciones del lenguaje que asiste en todo momento ofreciendo nombres de las funciones y parámetros que deben recibir. Aunque esta ayuda contextual no solo se queda en las funciones definidas en el lenguaje, sino que también reporta ayudas con las funciones que vayamos creando nosotros, incluso en páginas que tengamos incluidas con la función include().

Otras ayudas que ofrece a la hora de escribir son las típicas en editores avanzados, como permitir editar varios archivos, y moverse fácilmente entre ellos, marcar a qué elementos corresponden los inicios y cierres de las etiquetas, paréntesis o llaves, moverse al principio o al final de una función, identificación automática del código, etc.

Zend Studio dispone de una herramienta muy interesante de debug o depuración. Gracias a ella podemos ejecutar páginas y conocer en todo momento el contenido de las variables de la aplicación y las variables del entorno como las cookies, las recibidas por formulario o en la

sesión. Podemos colocar puntos de parada de los scripts y realizar las acciones típicas de depuración. Además de la ventana para visualizar el contenido de las variables, dispone de otras donde muestra la salida del script según se va generando, y otra donde se pueden ver las alertas y errores. Las posibilidades se completan con distintos tipos de depuración, en local, en remoto o a partir de una URL.

Quanta.

Quanta es un editor web desarrollado para el escritorio KDE. Soporta una multitud de lenguajes como HTML, JavaScript, CSS, PHP, SQL, XML, ColdFusion, Perl, DTML, Zope o C++, lo que hace que hoy por hoy sea uno de los editores más utilizados en el desarrollo de páginas web.

Este editor es un híbrido entre un editor web de texto y uno WYSIWYG ya que nos permite tanto hacer la página viendo el resultado final como directamente desde código.

Quanta ofrece multitud de funcionalidades, entre ellas, coloreado de sintaxis para todos los lenguajes soportados, cliente FTP integrado para colgar nuestra página de una manera fácil y rápida, cajas de diálogos contextuales, donde podremos elegir las etiquetas HTML que queremos utilizar, excelente navegador de directorios integrado, con el que accederemos a nuestros ficheros fácilmente, completo panel de previsualización en el que podremos ver nuestro resultado final, validador HTML integrado en la propia aplicación, soporte de extensiones/plugins para añadir funcionalidades extra, y gran integración con el escritorio KDE, al estar desarrollado para él, lo que nos permitirá trabajar más a gusto con la aplicación si este es el escritorio que utilizamos.

Eclipse

Eclipse es un entorno de desarrollo integrado de código abierto independiente de una plataforma para desarrollar opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Además es una plataforma de programación, desarrollo y compilación de aplicaciones Java. También posee una atractiva interfaz que lo hace fácil y agradable de usar. El usuario puede si lo desea añadir varios plugins que le permitirán usar Eclipse para otros lenguajes de programación como php. Entre las características con que cuanta están:

- Editor de texto
- Resaltado de sintaxis

- Compilación en tiempo real
- Pruebas unitarias con JUnit
- Control de versiones con Concurrent Versions System (CVS)
- Integración con Ant
- Asistentes (wizards): para creación de proyectos, clases, tests, etc.
- Refactorización

Módulos para Eclipse

Los módulos (plugins) son precisamente una de las ventajas del IDE Eclipse, de algunos de los propuestos para el desarrollo de nuestra aplicación hablaremos a continuación.

PDT

PDT (PHP Development Tools), este proyecto ha tenido una gran respuesta entre los desarrolladores de PHP y que ha sido descargado más de 300.000 veces, dato este que nos da una idea de la aceptación que ha tenido a novel internacional.

Entre las características en la versión actual (1.0) se encuentran:

- Editor sensible al contexto, el cual provee de resaltamiento de código, asistente de código y autocompletado de código.
- Integración con el modelo del proyecto Eclipse, que permite para inspeccionar el uso de las vistas del contorno del fichero y del proyecto, así como la nueva vista PHP Explorer.
- Soporte para el debug incremental del código de PHP
- Extensos frameworks y APIs que permiten a los desarrolladores e ISVs (vendedores de software independientes) fácilmente extender PDT para crear nuevas e interesantes herramientas orientadas al desarrollo de PHP.

Mylyn

Mylyn es un grupo centrado en la interfaz para Eclipse que reduce la sobrecarga de información y realiza varias tareas fáciles. Para ello, hacer tarea de primera clase de Eclipse, y

la integración de los ricos y la edición fuera de línea para los repositorios, como Bugzilla, Trac, y Jira. Una vez que sus tareas se integran, Mylyn supervisa su actividad para identificar la información pertinente para la tarea-a-mano, y utiliza este contexto para centrar la interfaz de usuario de Eclipse en información interesante, ocultar la poco interesantes, y de forma automática encontrar lo que está relacionado. Esto sitúa a la información que necesita para realizar su trabajo al alcance de su mano y mejora la productividad mediante la reducción de búsqueda, desplazamiento y la navegación. Mylyn también facilita la multitarea, planificación, la reutilización de los esfuerzos anteriores, y el intercambio de conocimientos.

Aptana

Aptana es una robusta y avanzada interface de desarrollo Web, enfocado a java script para el desarrollo de aplicaciones dinámicas. Entre sus principales características se tienen:

- Asistente en código Java script, HTML y CSS; incluyendo sus propias funciones en Java script.
- Vista instantánea de la estructura del código Java Script, HTML y CSS.
- Soporte para código PHP y ASP.
- Notificación en el código de errores y precauciones.
- Soporte Multiplataforma.
- Código abierto.

Estos IDE se encargan de la lógica del negocio, es decir son los encargados de controlar los datos, los cuales han de ser almacenados de alguna manera, la estructura por así decirlo en la que se guardan, son las bases de datos, a las cuales se accede mediante los gestores de bases de datos de los cuales se hará un estudio de los mas utilizados actualmente.

1.4.4 Sistemas Gestores de Bases de Datos. (SGBD)

En la actualidad la información es uno de los aspectos más importantes para el conocimiento, y la mayor parte de la información mundial se encuentra almacenada en bases de datos, las cuales poseen características que hacen que se pueda acceder a ellas de forma inmediata para adquirir la información necesaria.

Los Sistemas de Gestión de Bases de Datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

Los SGBD tienen a su vez un propósito general que es el de manejar de manera clara, sencilla y ordenada un conjunto de información; además, poseen un objetivo definido que es el de brindar a los usuarios una herramienta que les permita de forma general, manejar los datos de una base de datos sin conocer la forma en que se almacenan o se accede a los mismos.

Los SGBD poseen determinados objetivos específicos dentro de los que se pueden señalar:

- Independencia.
- Consistencia.
- Seguridad.
- Integridad.
- Tiempo de respuesta

Existen numerosos programas que se usan para diseñar, crear, definir y utilizar los ficheros y formularios de una base de datos. Entre ellos se pueden encontrar algunos como: SQL Server, MySQL, ACCESS entre otros como los que a continuación se describen.

MySQL

MySQL es un sistema de gestión de base de datos, multi-hilo y multiusuario con más de seis millones de instalaciones. Es una implementación Cliente-Servidor que consta de un servidor y diferentes clientes además de ser un sistema de administración de Base de Datos. MySQL es muy rápido, confiable y fácil de usar, es multiplataforma, multiusuario y permite elaborar consultas con el robusto SQL, además no tiene valor monetario, es un software cuya licencia es libre.

MySQL es muy utilizado en aplicaciones Web. Su popularidad como aplicación Web está muy ligada al lenguaje de programación Web PHP, que a menudo aparece en combinación con MySQL. En aplicaciones Web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones.

MySQL es un software de “Fuente Abierta” .Significa que cualquier usuario puede usarlo e incluso modificarlo. Puede además bajar el código fuente de MySQL y usarlo sin pagar. Entre las características disponibles en las últimas versiones y que además son implementadas únicamente por MySQL se puede destacar:

Múltiples motores de almacenamiento permitiendo al usuario escoger la que sea más adecuada para cada tabla de la base de datos.

Agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo.

A modo de resumen se muestran a continuación las principales características de este gestor de bases de datos son las siguientes:

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

PostgreSql

Es un potente sistema que maneja bases de datos, diseñado para administrar enormes cantidades de datos, es probadamente confiable, mantiene la integridad de los datos. Corre en los sistemas operativos mas utilizados en el mundo como Linux, Windows y en distintas versiones de Unix. Posee tipos de datos como INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, TIMESTAMP y algunos más. Puede almacenar objetos de gran tamaño incluyendo imágenes, videos y sonido, además tiene una amplia documentación. Mantiene una eficiente consulta de datos y resuelve múltiples problemas empresariales en el mundo actual. PostgreSql es escalable en cuanto a la cantidad de datos que maneja y al

número de usuarios concurrentes, puede manejar hasta más de 4 Terabytes de datos. Ha sido reconocido por las industrias y premiado en varias ocasiones. Algunas de sus características más importantes son:

- Escritura adelantada de registros, evitando que se pierdan datos si hay alguna falla de energía, fallos del Sistema Operativo, y fallas de hardware.
- Juegos de caracteres internacionales, codificación de caracteres multi-byte y Unicode;
- Características para la integridad de los datos como: claves primarias, llaves foráneas con capacidad de actualizar en cascada o restringir la acción en caso que existan hijos, restricción check, restricción de unicidad y restricción not null; todas las restricciones se pueden postergar hasta el momento de terminar la transacción.
- Multi-Version Concurrency Control, que sirve para lograr un control de concurrencia tan eficiente que en la gran mayoría de los casos no se requiere de bloqueos.
- Un sofisticado optimizador de consultas, que es capaz de resolver consultas complejas.
- Herencia de tablas (aunque la orientación a objetos no esta completa esta característica se puede utilizar para lograr el particionamiento de tablas en varios discos mediante una técnica llamada "restricciones excluyentes").

PostgreSql puede ser modificado y distribuido en la forma en que se quiera, puesto que su código fuente está disponible bajo la licencia BSD, licencia de código abierto.

SQL Server

Microsoft SQL Server es un sistema de gestión de bases de datos relacionales basada en el lenguaje SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. Es un servidor de bases de datos pensado para gestionar tantos clientes simultáneos como admita la potencia del hardware del equipo en el que esté instalado.

Posee una gran ventaja que es la de poder manejar bases de datos con millones de registros y lo hace funcionando con miles de conexiones simultáneas a los datos.

Entre sus desventajas se encuentran que está diseñado solamente para Windows o sistemas operativos de Microsoft por lo que se considera que no es multiplataforma, y de ahí su mayor

desventaja. Otras desventajas son los altos precios de las licencias y el gran consumo de recursos como la memoria RAM que necesita para ejecutarse correctamente.

Oracle

Es un Sistema de Gestión de Bases de Datos Relacionales (SGBDR) que dispone de potentes herramientas para la gestión y seguridad de los datos como son el motor de la base de datos Oracle y la herramienta Oracle Forms que permite diseñar pantallas de introducción y consulta de datos. Se basa en la tecnología cliente/servidor. Es un gestor de base de datos muy usado y algunas de sus características son:

- Seguridad en el acceso a los datos mediante la gestión de privilegios.
- Copias de seguridad. Oracle proporciona mecanismos para realizar copias de seguridad de los datos y su recuperación.
- Conectividad. Se puede acceder a datos de Oracle desde software de otro fabricante.

Es un producto vendido a nivel internacional, aunque la gran potencia que tiene y su elevado precio hacen que sólo se vea en empresas muy grandes y multinacionales.

1.5 Estilos arquitectónicos.

¿Qué es un estilo arquitectónico?

Cuando se habla del término de estilo arquitectónico en la arquitectura de software no queda más remedio que establecer una comparación entre estilos arquitectónicos de la arquitectura civil, por ejemplo: estilos eclécticos, barrocos, clásicos, etc. Los estilos de la arquitectura de software definen un conjunto de concepciones arquitectónicas comunes que identifican un momento en el desarrollo de la arquitectura.

Algunos de los principales estilos arquitectónicos que se usan en la actualidad están divididos por Clases de Estilos las que engloban una serie de estilos arquitectónicos específicos. Entre estos estilos numeran:

- Estilos de Flujo de datos
- Estilos centrados en datos

- Estilos de Llamada y Retorno
- Estilo de Código Móvil
- Estilos Peer – To – Peer

Cada uno de estos estilos contiene una serie de especializaciones, las que a continuación se hablan pertenecen al grupo de las de llamada y retorno.

1.5.1 Modelo – Vista – Controlador (MVC)

Se utiliza principalmente cuando es necesario modularizar la interfaz de usuario, las reglas de negocio y el control de eventos. Fue diseñado para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos. Sus características principales son que el Modelo, las Vistas y los Controladores se tratan como entidades separadas que tendrán funcionalidades propias como se podrá ver a continuación.

Modelo: Administra el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista y responde a instrucciones de cambiar el estado habitualmente desde el controlador). Mantiene el conocimiento del sistema. No depende de ninguna vista y de ningún controlador.

Vista: Maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa con el Modelo a través de una referencia al propio Modelo.

Controlador: Proporciona significado a las ordenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo.

Ventajas:

- Se puede mostrar distintas variantes de interfaz gráfica simultáneamente.
- La interfaz tiende a cambiar más rápido que las reglas del negocio. Agregar nuevos tipos de vista no afecta el modelo.

- Evita poner el código indebido en la capa impropia. Facilita despliegue en caso de modificaciones en el modelo de datos.

Desventajas:

- Puede aumentar un poco la complejidad de la solución. Como está guiado por eventos puede ser algo más difícil de depurar.
- Si hay demasiados cambios en el modelo la vista puede provocar un constante refrescamiento de las vistas, a menos que se prevea programáticamente.

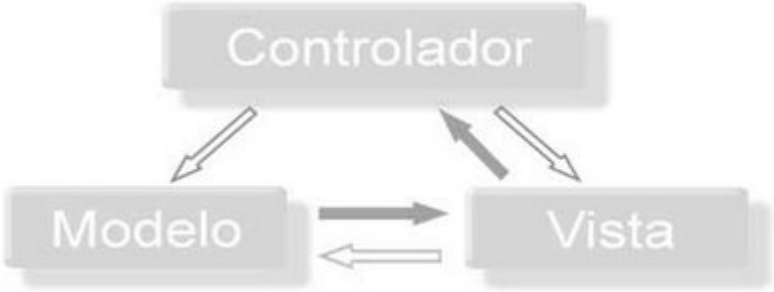


Figura 1.1 Vista de la arquitectura MVC

1.5.2 Arquitectura en Capas

Resulta útil cuando se pueden identificar distintas clases de servicios que pueden ser articulados jerárquicamente. Los componentes de cada capa consisten en conjuntos de clases. Las interacciones entre las capas ocurren generalmente por invocación de métodos, por definición los niveles de abajo no deben poder utilizar funcionalidad ofrecida por los de niveles superiores.

Ventajas

- Modularidad del sistema.
- Facilita la localización de errores.
- Mejora soporte del sistema

Desventajas

- Puede ser difícil definir que componentes ubicar en cada una de las capas.

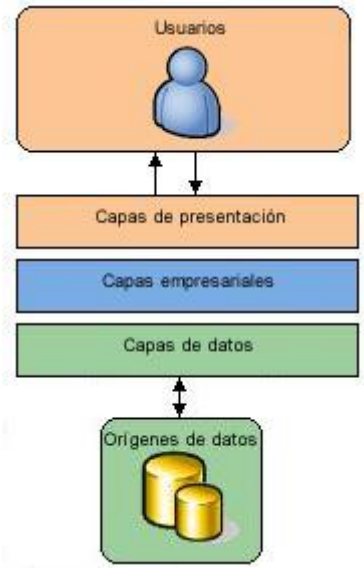


Figura 1.2 Vista de la Arquitectura en tres capas.

Estos patrones rigen muchas de las aplicaciones actuales, ejemplo del uso de estos se puede apreciar en los Framework, de los que se hará un análisis de los mas usados hoy en día con el objetivo de seleccionar uno para la futura implementación del sistema.

1.6 Framework

Un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Entre otras cosas un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, un framework proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, un framework facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas. En la actualidad existen diferentes framework tales como Symfony,

CakePHP, entre otros. De las características particulares de cada una de estos se hará un estudio con el propósito de determinar cual es el más apropiado para la aplicación.

1.6.1 CakePHP

CakePHP es un framework de desarrollo rápido de aplicaciones de código abierto en PHP. Inspirado en Rails, un framework para la construcción de sitios web que utilizan una base de datos como fuente de recursos, posee una infraestructura que tiene como finalidad, permitir el desarrollo de aplicaciones web de manera ágil y estructurada, sin perder flexibilidad.

Entre las características más destacables de CakePHP se incluyen:

- Arquitectura basada en el patrón Modelo Vista Controlador (MVC) y orientada a objetos: define clases modelo, vista y controlador con funcionalidades básicas y de las cuales heredan todas las clases que se ajustan a este patrón y que son usadas en la aplicación construida con el framework.

- Una comunidad activa de usuarios: creada tras la publicación del framework en 2005 y que ha contribuido a mejorar el framework (a través de subproyectos específicos en CakeForge.Org) y difundir su uso.

- Compatible con PHP4 y PHP5: aunque en PHP4 se requiere especificar algunos parámetros de configuración adicionales en las clases a implementar.

1.6.2 Zend Framework

Zend Framework destaca el hecho de que no sólo busca facilitar la programación a través del patrón MVC, sino también automatizar tareas más específicas, como el acceso a base de datos, el filtrado de datos ingresados a la aplicación o la búsqueda en un sitio web ordenando resultados por relevancia.

Entre sus metas se encuentran:

- Proveer un repositorio de componentes de alta calidad y que cuenten con soporte activo.
- Proveer un sistema completo para el desarrollo de aplicaciones web elaboradas en PHP5.

- Facilitar el aprendizaje en el uso del framework sin tener que aprender un nuevo lenguaje de programación.
- Organizar la colaboración de la comunidad para una programación avanzada en PHP5.

1.6.3 Symfony

Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web[4].

Symfony está desarrollado completamente con PHP 5. Además es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle entre otros. Es multiplataforma pues se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows. A continuación se mencionan algunas de las principales características de Symfony.

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares)
- Independiente del sistema gestor de bases de datos
- Sencillo de usar en la mayoría de los casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos
- Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador sólo debe configurar aquello que no es convencional
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web
- Preparado para aplicaciones empresariales y adaptable a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo

[4] **Zaninotto, François.** Symfony, la guía definitiva. [En línea] http://www.librosweb.es/symfony/index.html.

- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros

1.7 Metodología de desarrollo de software.

La metodología de desarrollo de software es un marco de trabajo conceptual de la ingeniería de software que promueve iteraciones en el desarrollo a lo largo de todo el ciclo de vida del proyecto.

Existen muchos métodos de desarrollo; la mayoría minimiza riesgos desarrollando software en cortos lapsos de tiempo. El software desarrollado en una unidad de tiempo es llamado una iteración, la cual debe durar el tiempo que se defina según la complejidad de la misma. Cada una de las iteraciones en un proyecto de software entero: incluye planeación, análisis de requerimientos, diseño, codificación, revisión y documentación.

Una iteración no debe agregar demasiada funcionalidad para justificar el lanzamiento del producto al mercado, pero la meta es tener un demo (sin errores) al final de cada iteración y al final de cada una el equipo vuelve a evaluar las prioridades del proyecto.

Entre las más conocidas se encuentran XP (eXtreme Programming) y RUP (Rational Unified Process).

1.7.1 Rational Unified Process (RUP)

El Proceso Unificado de desarrollo de Software (RUP) es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, a través del lenguaje Unified Modeling Language (UML) y el trabajo de muchas metodologías utilizadas por los clientes. La versión que se ha estandarizado se creó en 1998 y se conoció en sus inicios como Proceso Unificado de Rational 5.0, de ahí las siglas con las que se identifica a este proceso de desarrollo.

RUP está conformado por fases y flujos de trabajo. Las fases son inicio, elaboración, construcción y transición, los flujos de trabajo están comprendidos en 2 grupos, 6 de ingeniería: modelación del negocio, requerimientos, análisis y diseño, implementación, prueba e instalación, 3 de soporte: administración del proyecto, administración de configuración y cambios y por último ambiente.

Características de RUP:

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

Centrado en la arquitectura: La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción. RUP se desarrolla mediante iteraciones, comenzando por los casos de uso relevantes desde el punto de vista de la arquitectura.

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

RUP establece actividades y criterios que conducen a un sistema desde su máximo nivel de abstracción (la idea en la cabeza del cliente), hasta su nivel más concreto (un programa ejecutándose en las instalaciones del cliente). UML ofrece la notación gráfica necesaria a RUP para representar los sucesivos modelos que se obtienen en este proceso.

RUP es una metodología para proyectos de largo alcance, pero es estructurable y puede ser modificado en dependencia de las necesidades del proyecto y de la experiencia del desarrollador en el uso de esta metodología, logrando que se ajuste también a proyectos con un corto tiempo de desarrollo. Es fácil de entender tanto por usuarios avanzados como por clientes con pocos conocimientos técnicos. Está basado en normas provenientes de estándares seguidos por el entorno de desarrollo. Tiene procesos mucho más controlados, con numerosas políticas/normas. Existe un contrato prefijado donde el cliente interactúa con el equipo de desarrollo mediante reuniones. La arquitectura del software es esencial y se expresa mediante modelos. Por todo lo expuesto anteriormente se usa RUP como metodología de desarrollo.

1.7.2 Extreme Programing (XP)

Es una de las metodologías de desarrollo de software más exitosas en la actualidad, utilizadas para proyectos de corto plazo, pequeño equipo y cuyo plazo de entrega era ayer. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

Características de XP, la metodología se basa en:

- Pruebas Unitarias: se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándonos en algo hacia el futuro, podamos hacer pruebas de las fallas que pudieran ocurrir. Es como si nos adelantáramos a obtener los posibles errores.

- Refabricación: se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.

- Programación en pares: una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

Es también importante destacar lo importante que son algunos factores a la hora de hacer uso de este tipo de metodología, estos son:

- La comunicación, entre los usuarios y los desarrolladores.

- La simplicidad, al desarrollar y codificar los módulos del sistema.

- La retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.

Estas metodologías son de gran ayuda en la organización de los proyectos, ayudan a agilizar y optimizar el trabajo de los desarrolladores. Para ello se divide el trabajo en cuatro faces dentro de las cuales ocurren diferentes flujos de trabajos algunos de los cuales pueden ser modelados de manera grafica con herramientas especializadas de las que haremos un estudio en el siguiente epígrafe.

1.8 Herramienta CASE de modelado de UML

Las Herramientas para el modelado visual son también un aspecto importante a tener en cuanta en el desarrollo del presente trabajo. Existen varias y cada una de ellas con una serie de características que le permiten destacarse o no en dependencia de la línea de trabajo en la que se quiera insertar este tipo de herramientas. Dentro del uso de estas se estable como lenguaje de modelado el UML el cual es el más utilizado y conocido en la actualidad, este no es mas que un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software.

1.8.1 Rational Rose

Es una herramienta propietaria para el modelado visual, que forma parte de un conjunto más amplio de herramientas que juntas cubren todo el ciclo de vida de desarrollo de software. Esta herramienta propone la utilización de cuatro tipos de modelo para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software.

Características:

- Mantiene la consistencia de los modelos del sistema software.
- Chequeo de la sintaxis UML.
- Genera la documentación automáticamente.
- Generación de código a partir de los modelos.
- Ingeniería inversa (crear modelo a partir de código).

1.8.2 Visual Paradigm

Es una herramienta CASE que da soporte al modelado visual con UML 2.0, que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Este software ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

Visual Paradigm ofrece:

- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa (versión profesional) e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de integrarse en los principales IDEs.
- Disponibilidad en múltiples plataformas.

1.9 Decisiones y soluciones técnicas.

En el presente epígrafe se mostrarán las decisiones tomadas respecto a las herramientas que se utilizarán para el futuro desarrollo del sistema y se establecerá una comparación en cada uno de los casos con el objetivo de dejar claro la razón de la decisión tomada.

1.9.1 Lenguaje de programación

Se utilizará PHP como lenguaje de programación para la implementación del sistema, debido a que es un lenguaje "open source" interpretado de alto nivel embebido en páginas HTML, y su código es ejecutado en el servidor. Este lenguaje nos permite no solo leer, sino también manipular datos desde diversas fuentes, incluso los datos que pueden ser ingresados por los usuarios desde formularios HTML.

El código PHP es mucho más legible que el de PERL, este último lenguaje consume muchos recursos de la máquina haciéndolo menos ligero.

PHP no es el único lenguaje de programación cliente/servidor Web que existe, entre las alternativas más conocidas esta ASP por parte de Microsoft, el cual corre bajo el servidor Web propietario de Microsoft: IIS, debido a esto es de pago, por lo que es muy poco usado y aunque es muy potente su metodología de programación es poco robusta y no logra superar a PHP como lenguaje de programación Web.

JSP por parte de Java, es la otra alternativa desarrollada por Sun Microsystems, este se suele usar más para scripts de línea de comando, mientras que PHP se usa más para páginas dinámicas. PHP a la hora de servir páginas Web es más eficaz que JSP, ya que se integra como módulo en Apache, y así se evita la carga del intérprete en memoria cada vez que se

ejecuta una página. PHP al ser una herencia positiva de Perl facilita al programador muchas cosas que en JSP son más difíciles de programar. Por ejemplo, el acceso a bases de datos es mucho más sencillo en PHP, ya que el API de MySQL, por ejemplo, viene integrado en el propio lenguaje.

1.9.2 Entorno de desarrollo

Como propuesta para el entorno de programación se ha designado el Eclipse, por todas las características mencionadas anteriormente. Además, por formar parte de la comunidad de software libre, cuanta con un gran número de desarrolladores que continuamente intervienen en su mejoramiento, por otra parte cuenta con un sistema de plugins, aspecto este que influye mucho en su flexibilidad, pues existen cientos de estos que pueden ser usados para programar en lenguajes como C++, Perl, PHP, XML, por solo mencionar algunos.

Con respecto a las conexiones que se establecen con las bases de datos es importante destacar que a diferencia de otros IDE como Zend Studio no presenta problemas para relacionarse con diferentes bases de datos como My SQL.

Quanta es un completo editor de código HTML, pero le falta mucho aun para llegar a alcanzar la calidad y flexibilidad que tiene Eclipse, además el trabajo con sus herramientas puede ser muy complicado en ocasiones.

1.9.3 Sistema gestor de base de datos

Se utilizará MySQL como gestor de base de datos, primeramente porque está desarrollada bajo la filosofía de código abierto.

A diferencia de PostgreSql su principal objetivo de diseño fue la velocidad, y consume muy pocos recursos, tanto de CPU como de memoria, mientras que PostgreSql es lento, pesado, y consume bastante recursos.

Una de las ventajas que tiene MySQL es que a partir de la versión 3.23.19 está siendo desarrollado bajo la licencia GPL, a diferencia de SQL Server que posee como desventaja altos costos de licencia y al no ser multiplataforma solo esta disponible en Sistemas Operativos de Microsoft.

MySQL, también tiene la ventaja de ser un SGBD robusto, que puede almacenar gran cantidad de datos, es rápido, seguro, estable, gratuito y soporta múltiples lenguajes de programación, con lo que puede conectarse a una base de datos de este tipo a través de cualquiera de ellos.

1.10 Conclusiones del capítulo

En el presente capítulo se abordaron una serie de conceptos que fundamentan todo lo referente a la parte teórica de la investigación, proporcionando a los lectores un mayor entendimiento del mismo. Además se han descrito una serie aspectos relacionados con la información y el control de esta dentro de un sistema web, así como los lenguajes de programación, las tecnologías, los sistemas de gestión de bases de datos y la metodología a utilizar para guiar el desarrollo de la aplicación. Todo esto unido ha permitido dar paso a hacer una propuesta del sistema que compete a esta investigación.

Capítulo II: Propuesta del sistema.

En este capítulo se hace una descripción de las características del sistema donde se da a conocer cada uno de los elementos que serán utilizados para desarrollar la aplicación, así como el flujo de trabajo con el que se determina la solución al problema de investigación planteado. Además se describe el Modelo de Negocio, definiendo conceptos que son muy útiles para entender el contexto del problema. Se enumeran los requisitos no funcionales y funcionales, estos últimos se estructuran mediante los Casos de Uso del sistema, de los cuales se ofrece una descripción textual, para brindar una mayor interpretación de los mismos.

2.1 Objetivos estratégicos de la organización y procesos de negocio que los soportan.

1. Solicitar la compra o actualización de una licencia.

2. Recuperar información de licencias existentes.

3. Solicitar la instalación de una licencia de software.

Actualmente para hacer el pedido de licencias para un software producido por la institución solo se esta efectúan los procesos de negocio comprar licencia y actualizar licencia.

2.2 Flujo actual del proceso del negocio.

La Universidad de las Ciencias Informáticas desde su inicio fue concebida para que el estudio y la producción se desarrollaran de manera paralela, es decir, lograr producir software de alta calidad que pudieran ser solicitados por empresas nacionales y extranjeras, para ello estos productos han de estar legalizados, por lo que se hace necesario la compra o solicitud de las licencias asociadas a la realización de los mismos, este proceso se resume en la solicitud de licencias que puede ser compra o actualización de licencia.

¿Cómo se realizan estos procesos?

CU Comprar Licencias: las peticiones de compra pueden venir de parte de las facultades o de la Infraestructura Productiva (IP), en cualquiera de los dos casos los encargados de hacer la solicitud de compra, de antemano deben verificar que la licencia no exista, esto se hace buscando de manera manual en la Plataforma Tecnológica de Servicios de la Dirección Técnica (DT), en caso de que se aseguren que no está entonces hacen la petición de la

licencia, para lo que debe completar una planilla especificada por la DT, esta será enviada mediante el correo con el formato establecido previamente por la dirección técnica.

CU Actualizar Licencias: las peticiones de actualización pueden venir de parte de las facultades o de la Infraestructura Productiva (IP), en cualquiera de los dos casos los encargados de hacer la solicitud de actualización, de antemano deben verificar que la licencia este disponible, esto se hace buscando de manera manual en la Plataforma Tecnológica de Servicios de la DT, en caso de que se aseguren que está entonces hacen la petición de la licencia, para lo que debe completar una planilla especificada por la DT, esta será enviada mediante el correo con el formato establecido previamente por la dirección técnica.

2.3 Análisis crítico de cómo se ejecutan actualmente esos procesos, las causas que originan la situación problémica y las consecuencias.

Una de las causas que origina la situación problémica es la carencia de una aplicación que gestione de manera rápida las licencias de software en la Universidad de las Ciencias Informáticas, esto se debe a que es una institución nueva, que aunque cuenta con los medios tecnológicos suficientes aun se encuentra en una etapa de perfeccionamiento; estas gestiones actualmente se hacen mediante correos electrónicos, método este que no es todo lo seguro como el proceso lo requiere, además de incurrir en perdidas de tiempo innecesarias, por otra parte se deja la responsabilidad de verificar las licencia al cliente dando la posibilidad de que se equivoque en el proceso de búsqueda, situación esta que pudiera atentar con el cumplimiento del plazo asignado para la entrega del software producido.

2.4 Objeto de automatización.

Se desea automatizar el proceso de gestión de licencias de la Universidad de las Ciencias Informáticas, así como el control y registro de estas.

2.5 Información que se maneja.

La información que se maneja es toda la referente a los datos específicos de las licencias adquiridas por la Universidad de las Ciencias Informáticas.

2.6 Propuesta del sistema.

Para darle solución al problema en cuestión se ha decidido utilizar una aplicación Web que garantice la disponibilidad de un catálogo de licencias de software así como toda la información necesaria relacionada en el proceso, esta decisión ha sido tomada pues es la vía mas rápida y eficiente para todo lo referente a la gestión de las licencias de software. Hasta el momento no se tiene referencia de ningún sistema que de solución a este problema tanto en el ámbito nacional como en el internacional.

Siguiendo la marcada intención de nuestro país de migrar hacia el software libre, esta aplicación será elaborada con herramientas libres, específicamente Eclipse usando elementos dentro de este tales como PDT, Mylyn y Aptana.

Según el estudio realizado a diferentes lenguajes de programación resultó ser el más indicado PHP el cual se ejecuta del lado del servidor, por sus potencialidades y las ventajas que presenta como software libre y reúne las mejores capacidades funcionales para la creación de la aplicación Web.

En cuanto al Sistema de Gestión de Base de Datos se escogió MySQL, herramienta libre que posee una gran estabilidad. Además de esto tiene la capacidad brindar altas velocidades de respuestas aspecto este considerado como critico para la solución del sistema.

Atendiendo a la idea de que en futuro se deseen hacer cambios de algún tipo en el sistema se decidió implementar el sistema propuesto basándose en una arquitectura MVC la cual esta representada en la forma en que esta concebido el framework Symfony que será el utilizado para enmarcar las soluciones del negocio, estos queda reflejado en la siguiente figura 2.1. En esta figura queda reflejada la manera en la que el controlador frontal, que además de ser el único, es el encargado del manejo de las peticiones del usuario permitiendo una mejor asignación de los permisos de usuarios, es la única vía de entrada al sistema, mientras que las acciones incluyen el código especifico del controlador de cada pagina. La vista se encarga de producir las páginas que se muestran como resultado de las acciones, esto se hace a través de las planillas las cuales son mostradas de forma decoradas mediante layouts. Todo este proceso no tiene sentido sin la presencia de datos los cuales son tomados de bases de datos, para la gestión de estos en Symfony existe el modelo, estas acciones se hacen mediante objetos, de esta forma nunca se accede de forma explícita a la base de datos. Este comportamiento permite un alto nivel de abstracción y permite una fácil portabilidad.

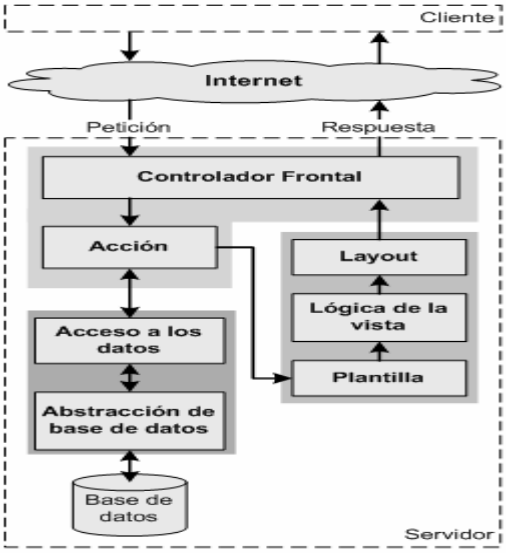


Figura 2.1 Vista de la arquitectura MVC para Symfony

Luego de un análisis se ha decidido utilizar RUP pues esta metodología traza una mejor y completa línea de trabajo. Es un proceso de desarrollo de Software que proporciona una guía en el orden de las actividades de un equipo, dirige las tareas individuales de los desarrolladores, especifica que productos deberían ser desarrollados y ofrece criterios para monitorear y medir los productos y actividades de un proyecto así como usar casos de uso en forma efectiva facilita tener una interacción continua y clara, evitando construir sistemas de información que no están acorde a las expectativas finales.

Para el modelado de la aplicación se ha decidido utilizar Visual Paradigm debido a que es una herramienta multiplataforma que presenta numerosas ventajas especialmente por ser libre.

Una ves hecha una descripción de diferentes elementos que forman parte del trabajo se esta realizando, se considera necesario abordar aspectos técnicos del mismo.

2.7 Descripción técnica

2.7.1 Modelo del negocio.

Un modelo del negocio (también llamado diseño del negocio) es el mecanismo por el cual un negocio trata de de generar ingresos y beneficios. Es un resumen de cómo una compañía planifica servir a sus clientes.

De ahí, que en el campo del software también resulte útiles la creación de modelos que organicen y presenten los detalles importantes de problemas reales que se vinculan con el sistema informático a construir. Estos modelos deben cumplir una serie de propiedades, entre ellas la de ser coherentes y relacionados. Uno de los modelos útiles previo al desarrollo de un software es el modelo del negocio.

Los objetivos de la modelación del negocio son:

- Comprender le estructura y la dinámica de la organización en la cual se va a implantar un sistema.
- Comprender los problemas actuales de la organización e identificar las mejoras potenciales.
- Asegurar que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común de la organización.
- Derivar los requerimientos del sistema que va a soportar la organización.

2.7.2 Definición del actor y trabajador del negocio.

A continuación se muestran los actores y trabajadores del negocio así como sus respectivas descripciones.

Descripción del actor del negocio:

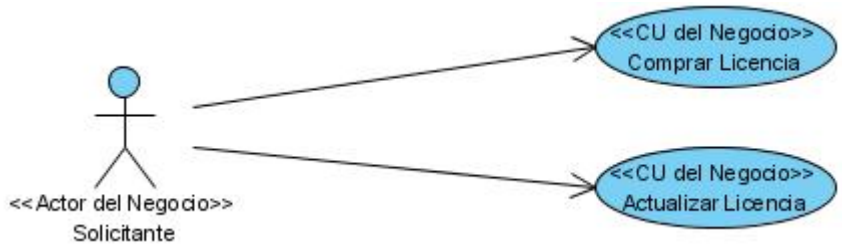
Nombre del actor	**Descripción**
Solicitante	Es la generalización de cada uno de los actores del negocio.
Encargado por Facultad	Es el Asesor de Arquitectura y Tecnologías de la Facultad o el Vicedecano de Producción lo cuales son los encargados de hacer los pedidos por la facultad.
Encargado por la IP	Es el especialista de la dirección el cual es el encargado de hacer los pedidos por la IP.

Descripción del trabajador del negocio:

Nombre del Trabajador	**Descripción**
Jefe de Grupo de Soporte Técnico (JGST)	Es la persona designada por la dirección de la IP para recibir los pedidos.

2.7.3 Diagrama de casos de de uso del negocio.

A continuación se muestra el diagrama de Caso de Uso del negocio con sus casos de usos asociados.

Diagrama de Caso de Uso del Negocio Gestionar Licencias.


2.7.4 Especificación de los casos de uso del negocio

Descripción del Caso de Uso Solicitar Autenticación:

Caso de Uso:	Comprar Licencia
Actores:	Solicitante
Trabajadores:	Jefe de Grupo de Soporte Técnico (JGST)
Resumen:	Este caso de uso se inicia cuando el Solicitante hace el pedido de compra de licencias.
Precondiciones:	---
Poscondiciones:	La solicitud queda registrada.
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Negocio**
1. El solicitante verifica que la licencia no exista.	

2. El solicitante confecciona la planilla de compra.	
3. El solicitante hace el pedido de la licencia.	3.1 El pedido queda registrado.
Flujos Alternos	
Acción del Actor	**Respuesta del Negocio**
Prioridad	Crítico

Descripción del Caso de Uso Solicitar Licencia:

Caso de Uso:	Actualizar Licencia
Actores:	Solicitante
Trabajadores:	Jefe de Grupo de Soporte Técnico (JGST)
Resumen:	Este caso de uso se inicia cuando el Líder de Proyecto hace el pedido de la actualización de licencias.
Precondiciones:	---
Poscondiciones	La solicitud queda registrada.
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Negocio**
1. El solicitante verifica que la licencia este disponible.	
2. El solicitante confecciona la planilla de actualización.	
3. El solicitante hace el pedido de la licencia.	3.1 El pedido queda registrado.
Flujos Alternos	
Acción del Actor	**Respuesta del Negocio**
Prioridad	Crítico

2.7.5 Diagrama de actividades.

Diagrama de Actividades Caso de Uso del Negocio Comprar Licencias.

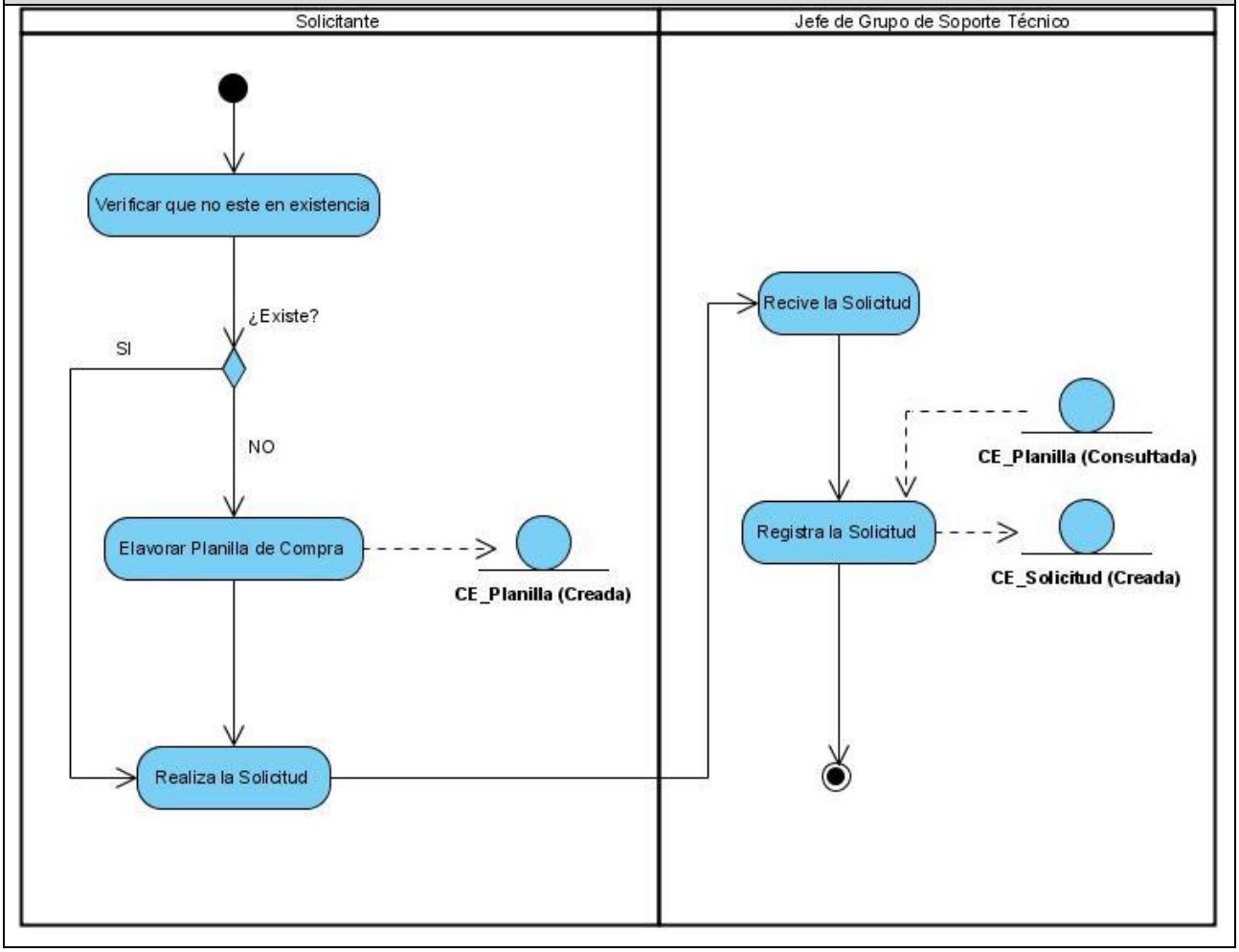
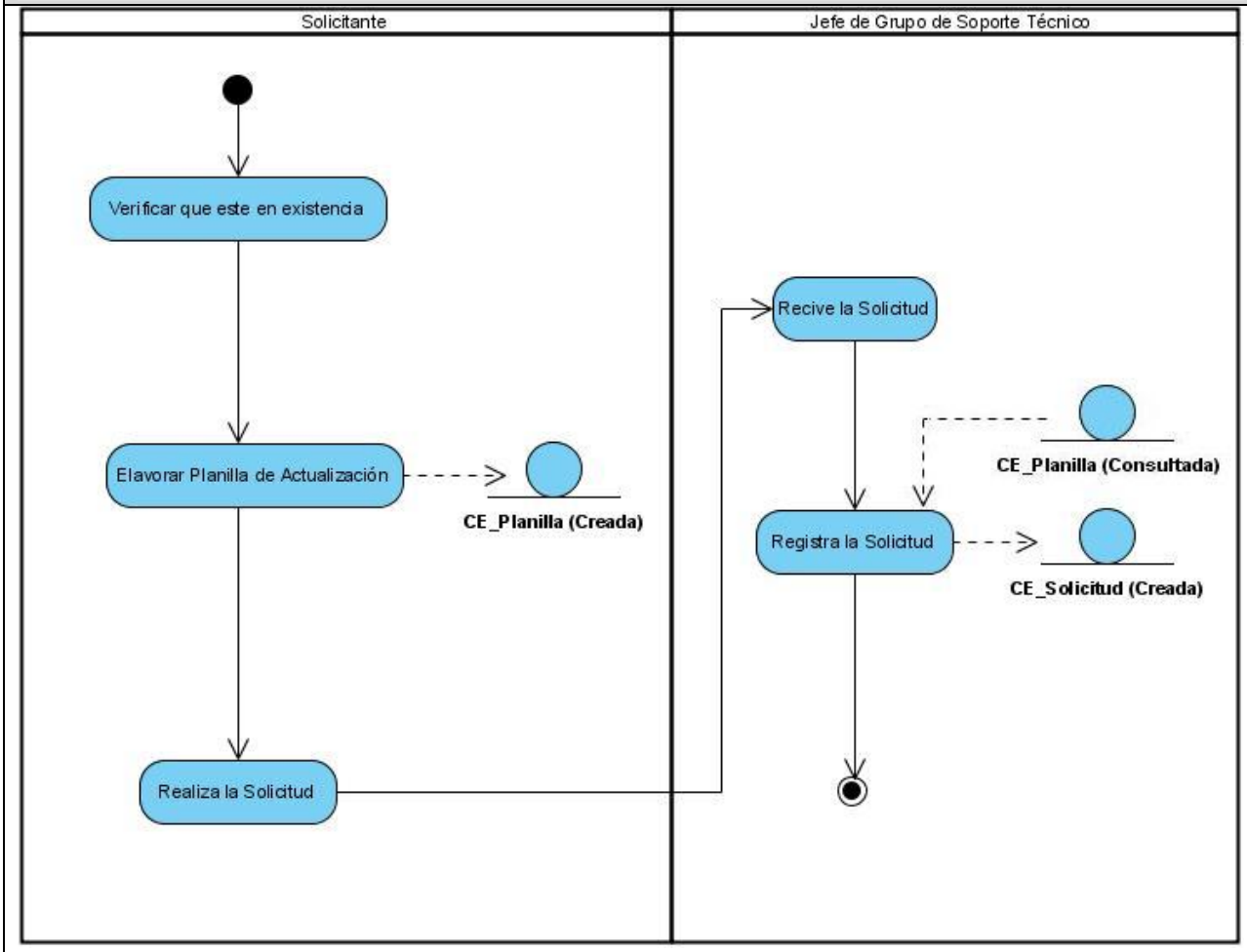


Diagrama de Actividades Caso de Uso del Negocio Actualizar Licencias.



2.7.6 Modelo de objetos del negocio.

El modelo de objeto es aquel diagrama que representa las entidades con que se relacionan los trabajadores del negocio. Una entidad del negocio no es más que algo físico que se utilice en el proceso de negocio y que sirva para obtener o actualizar información.

2.8 Relación de los requerimientos.

Con el apoyo de una entrevista, se efectuó el proceso de captura de los requisitos funcionales y no funcionales del Sistema para la gestión de licencias estos se muestran a continuación.

2.8.1 Requerimientos funcionales del sistema.

Los requerimientos funcionales definen las funciones que el sistema deberá realizar:

R1 Gestionar usuario

R1.1 Crear usuario

R1.2 Modificar usuario

R1.3 Eliminar usuario

R2 Autenticar usuario

R3 Editar perfil de usuario

R4 Solicitar licencia

R4.1 Solicitar préstamo o instalación de una licencia existente

R4.2 Solicitar compra de licencia no existente en el catalogo

R4.3 Solicitar actualización de licencia existente

R5 Gestionar licencia

R5.1 Adicionar licencia

R5.2 Actualizar licencia

R5.3 Eliminar licencia

R5.4 Listar licencias

R6 Buscar licencias

R6.1 Hacer búsqueda general

R6.2 Hacer búsqueda avanzada

R7 Confeccionar reporte

R7.1 Mostrar usuarios conectados

R7.2 Mostrar ranking de licencias mas descargadas

2.8.2 Requerimientos no funcionales del sistema.

Requerimientos de apariencia o interfaz externa:

El sistema esta concebido para que sea utilizado por personas con conocimientos de informática, no obstante, la interfaz a de ser amigable de fácil entendimiento, para facilitar el trabajo de los usuarios en la misma. Se ha decidido que la conexión entre servidor de base de datos y el servidor Web se a través del protocolo TCP/IP, entre la maquina cliente y el servidor Web mediante HTTPS. Las maquinas clientes pueden tener acopladas impresoras usando cable USB o paralelo.

Requerimientos de Usabilidad:

El sistema estará controlado por administradores con conocimientos de informática, los cuales, con el apoyo de materiales de ayuda que les serán proporcionados y teniendo acceso pleno al sistema, serán capaces de dar solución a cualquier situación que se pueda presentar.

Requerimientos de Rendimiento:

Para un funcionamiento óptimo de la aplicación se seguirán las diferentes técnicas de elaboración en la Web, que faciliten el rápido acceso a sus páginas. La eficiencia del producto estará determinada en gran medida por el aprovechamiento de los recursos que se disponen en el modelo Cliente/Servidor, y la velocidad de las consultas en la Base de Datos. La herramienta propuesta debe ser rápida y el tiempo de respuesta debe ser el mínimo posible, adecuado a la rapidez con que el cliente requiere la respuesta a su petición.

Requerimientos de Portabilidad:

La aplicación propuesta debe ser multiplataforma (podrá ser usada bajo cualquier sistema operativo).

Requerimientos de Seguridad:

Confiabilidad: la información manejada por el sistema debe estar protegida de acceso no autorizado y divulgación, dándole a casa usuario solo los privilegios indispensables para este.

Integridad: la información con la que se cuenta debe ser protegida, con el objetivo de mantenerla íntegra, es decir que esta no sea transformada.

Disponibilidad: La aplicación deberá estar disponible en todo momento para aquellas personas con acceso a la información y los mecanismos utilizados para lograr la seguridad no deben ser un obstáculo a los usuarios para obtener los datos deseados en un momento dado.

Requerimientos de Software:

En la computadora de los usuarios sólo se requiere un navegador Web, (bajo cualquier sistema operativo). Como servidor Web se debe tener el Apache en su versión 2.0 en adelante. Para su implementación se usará como herramienta de desarrollo Eclipse y como gestor de bases de datos MySQL.

Requerimientos de Hardware:

En el cliente se requiere de una máquina con 128 MB de RAM como mínimo, el servidor Web deberá tener 256 MB de RAM y 20 GB de capacidad de almacenamiento en disco como mínimo, y el servidor de base de datos 256 MB de RAM como mínimo, todas las máquinas implicadas en la funcionalidad de la aplicación deben estar conectadas a la red.

2.9 Modelo caso de uso del sistema

Se representan los requisitos funcionales del sistema mediante un diagrama de casos de uso utilizando UML como lenguaje de modelado visual, donde se definen los actores y los casos de uso del mismo.

2.9.1 Actores del sistema

Actores	**Justificación**
Usuario	Es la persona que interactúa con el sistema y que posee el permiso para ejecutar opciones específicas.
Solicitante	Es la persona que interactúa con el sistema, es el que solicita la licencia.

Administrador	Representa un usuario avanzado con privilegios plenos dentro del sistema, el cual será el encargado de llevar a cabo todo los procesos de gestión de licencias, usuarios y confeccionar reportes.

Tabla 2.9.1 Actores del sistema

2.9.2 Casos de Uso del Sistema

CU-1	Gestionar usuario
Actor	Administrador
Descripción	Permite al administrador gestionar todo lo referente a los usuarios ya sea crear, modificar o eliminar cuentas de usuarios.
Referencia	R1

Tabla 2.9.2 Caso de uso: Gestionar usuario

CU-2	Autenticar usuario
Actor	Solicitante
Descripción	Permite al usuario autenticarse y acceder al sistema.
Referencia	R2

Tabla 2.9.3 Caso de uso: Autenticar usuario

CU-3	Editar perfil de usuario
Actor	Solicitante
Descripción	Posibilita al usuario editar sus datos personales en el sistema.

Referencia	R3

Tabla 2.9.4 Caso de uso: Editar perfil de usuario

CU-4	Solicitar licencia
Actor	Solicitante
Descripción	Posibilita al usuario hacer solicitudes de los diferentes servicios relacionados con las licencias, ya sea préstamo o instalación, solicitar compra o actualización.
Referencia	R4

Tabla 2.9.5 Caso de uso: Solicitar licencia

CU-5	Gestionar licencia
Actor	Administrador
Descripción	Posibilita al administrador hacer operaciones con las licencias dentro del sistema tales como, adicionarlas, actualizarlas, eliminarlas y listarlas.
Referencia	R5

Tabla 2.9.6 Caso de uso: Gestionar licencia

CU-6	Buscar licencia
Actor	Solicitante
Descripción	Posibilita al solicitante realizar búsquedas de licencias, la búsqueda puede ser general o avanzada.
Referencia	R6

Tabla 2.9.7 Caso de uso: Buscar licencia

CU-7	Confeccionar reporte

Actor	Administrador
Descripción	Posibilita al administrador mostrar reportes cuantitativos de las licencias.
Referencia	R7

Tabla 2.9.8 Caso de uso: Confeccionar reporte

2.9.3 Diagrama de Casos de Uso del Sistema

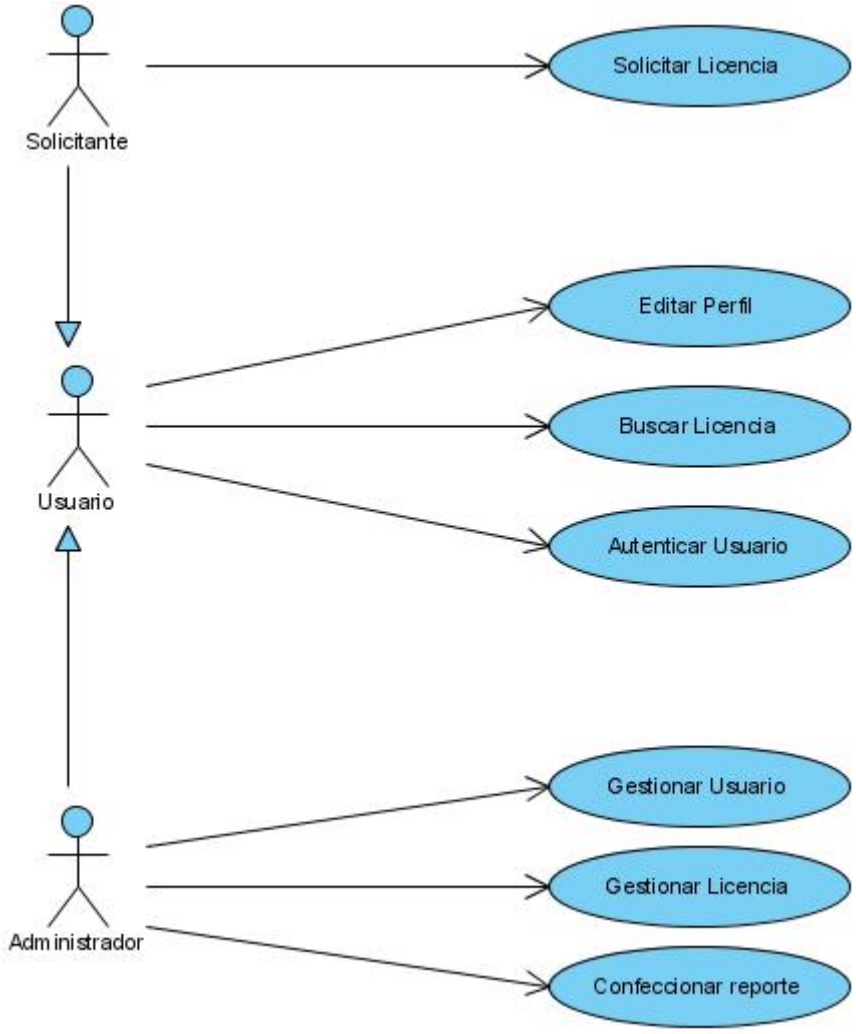


Figura 2.9.1 Diagrama de casos de uso del sistema

2.9.4 Descripción textual de los casos de uso

Caso de Uso	**Gestionar usuario**
Actores	Administrador
Propósito	Permitir a un administrador adicionar, modificar o eliminar un usuario.
Resumen	El caso de uso inicia cuando el administrador selecciona la opción de adicionar, modificar o eliminar un usuario, el sistema le solicita una serie de datos que debe introducir. Cuando el usuario envía estos datos el sistema los verifica y el caso de uso finaliza cuando se actualizan todos los cambios realizados.
Referencias	R1
Precondiciones	----
Poscondiciones	Se registran los cambios realizados por el administrador.
Sección Adicionar usuario	
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El administrador selecciona la opción adicionar usuario.	1.1 El sistema le solicita una serie de datos que debe introducir.
2. El administrador introduce los datos.	2.1 El sistema verifica los datos. 2.2 El sistema verifica si el usuario no existe. 2.3 El sistema adiciona el nuevo usuario a la base de datos.
Flujo Alternativo	
Acción del Actor	**Respuesta del Sistema**
	2.1 Si los datos no son válidos el sistema emite un mensaje de aviso. 2.2 Si el usuario existe el sistema envía

	un mensaje de aviso.
Sección Modificar usuario	
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El administrador busca el usuario a modificar.	1.1 El sistema le muestra el usuario.
2. El administrador modifica los datos del usuario.	2.1 El sistema verifica si los datos son válidos. 2.2 El sistema actualiza los cambios.
Flujo Alternativo	
Acción del Actor	**Respuesta del Sistema**
	1.1 Si el usuario no existe el sistema envía un mensaje de aviso. 2.1 Si los datos no son válidos el sistema emite un mensaje de aviso.
Sección Eliminar usuario	
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El administrador busca el usuario que desea eliminar.	1.1 El sistema verifica si el usuario existe. 1.2 El sistema muestra el usuario.
2. El administrador selecciona la opción eliminar usuario	2.1 El sistema elimina el usuario y se actualiza.
Flujo Alternativo	
Acción del Actor	**Respuesta del Sistema**
	1.1 Si el usuario no existe envía un mensaje de aviso.
Prioridad	Crítico

Tabla 2.9.9 Descripción textual del CU Gestionar usuario

Caso de Uso	**Autenticar usuario**
Actores	Usuario
Propósito	Permitir a un usuario autenticarse y acceder al sistema.
Resumen	El caso de uso inicia cuando el usuario decide acceder al sistema, este introduce los datos que le solicitan para acceder al mismo y finaliza cuando el sistema verifica los datos y le habilita la entrada.
Referencias	R2
Precondiciones	El usuario debe estar registrado.
Poscondiciones	Se habilitan las funcionalidades según lo privilegios.

Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El usuario selecciona la opción de autenticarse.	1.1 El sistema solicita usuario y contraseña del dominio (uci.cu) al usuario.
2. El usuario introduce los datos.	2.1 El sistema comprueba la validez de los datos del usuario. 2.2 El sistema verifica que el usuario este registrado. 2.3 El sistema muestra la interfaz de bienvenida y las opciones a las que tiene acceso según sus privilegios
Flujo Alternativo	
Acción del Actor	**Respuesta del Sistema**

	2.1 Si los datos no son válidos el sistema envía un mensaje de aviso. 2.3 Si el usuario no está registrado el sistema envía un mensaje de aviso.
Prioridad	Crítico

Tabla 2.9.10 Descripción textual del CU Autenticar usuario

Caso de Uso	**Editar perfil**
Actores	Usuario
Propósito	Permitir a un usuario editar su perfil.
Resumen	El caso de uso inicia cuando el usuario selecciona la opción de editar su perfil, este inserta los datos que el sistema le pide y finaliza cuando el sistema actualiza dichos datos.
Referencias	R3
Precondiciones	El usuario debe estar autenticado.
Poscondiciones	Los datos del usuario quedan actualizados.
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El usuario selecciona la opción de editar su perfil.	1.1 El sistema muestra los datos personales del usuario que puede insertar o modificar.
2. El usuario inserta los datos.	2.1 El sistema verifica los datos. 2.2 El sistema actualiza los datos.
Flujo Alternativo	
Acción del Actor	**Respuesta del Sistema**

	2.1 Si los datos no son válidos el sistema envía un mensaje de aviso.
Prioridad	Auxiliar

Tabla 2.9.11 Descripción textual del CU Editar perfil

Caso de Uso	**Solicitar licencia**
Actores	Solicitante
Propósito	Permite al solicitante hacer el pedido de una licencia.
Resumen	El caso de uso inicia cuando el solicitante selecciona la opción de instalar, actualizar o solicitar licencia no existente, el sistema le solicita una serie de datos que debe introducir. Cuando el usuario envía estos datos el sistema los verifica y el caso de uso finaliza cuando se actualizan todos los cambios realizados.
Referencias	R4
Precondiciones	El solicitante debe estar autenticado.
Poscondiciones	El pedido queda registrado.
Sección Solicitar préstamo o instalación de una licencia existente	
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El solicitante selecciona la opción de instalación de licencia.	1.1 El sistema muestra las licencias disponibles.
2. El solicitante elige las licencias deseadas.	2.1 El sistema agrega cada una de las licencias seleccionadas a una lista temporal de licencias.
3. El solicitante indica que ha concluido la selección.	3.1 El sistema registra las licencias solicitadas así como los datos del solicitante.3.2 El sistema registra el pedido.

	3.3 El sistema muestra las licencias especificadas.
Flujo Alternativo	
Acción del Actor	**Respuesta del Sistema**
Sección solicitar compra de licencia no existente en el catalogo	
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El solicitante selecciona la opción de solicitar una licencia no existente.	1.1 El sistema solicita los datos de la licencia.
2. El solicitante introduce los datos de la licencia.	2.1 El sistema hace una búsqueda para verificar que la licencia no exista. 2.2 El sistema registra el pedido. 2.3 El sistema muestra un mensaje.
Flujo Alternativo	
Acción del Actor	**Respuesta del Sistema**
	2.2 La licencia especificada ya existe. 2.3 El sistema muestra un mensaje.
Sección solicitar actualización de licencia existente	
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El solicitante selecciona la opción de actualizar una licencia.	1.1 El sistema solicita los datos.
2. El solicitante introduce los datos.	2.1 El sistema chequea que la actualización de la licencia exista. 2.2 El sistema registra el pedido. 2.3 Muestra el mensaje.
Flujo Alternativo	
Acción del Actor	**Respuesta del Sistema**

Prioridad	Critico

Tabla 2.9.12 Descripción textual del CU Solicitar licencia

Caso de Uso	**Gestionar licencia**
Actores	Administrador
Propósito	Permite al administrador ejecutar las acciones relacionadas con la gestión de una licencia.
Resumen	El caso de uso inicia cuando el administrador selecciona la opción de adicionar, actualizar o eliminar una licencia, el sistema le solicita una serie de datos que debe introducir. Una ves terminada esta acción el sistema lista la licencia en el catalogo terminando el caso de uso.
Referencias	R5
Precondiciones	El administrador debe estar autenticado.
Poscondiciones	El sistema se actualiza.

Sección Adicionar licencia	
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El administrador selecciona la opción adicionar licencia.	1.1 El sistema le solicita una serie de datos que debe introducir.
2. El administrador introduce los datos.	2.1 El sistema verifica los datos. 2.2 El sistema verifica si la licencia existe. 2.3 El sistema adiciona la nueva licencia al catálogo.
Flujo Alternativo	
Acción del Actor	**Respuesta del Sistema**
	2.1 Si los datos no son válidos el sistema

	emite un mensaje de aviso. 2.2 Si la licencia existe el sistema envía un mensaje de aviso.
Sección Actualizar licencia	
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El administrador busca la licencia a modificar.	1.1 El sistema le muestra la licencia.
2. El administrador modifica los datos de la licencia.	2.1 El sistema verifica si los datos son válidos. 2.2 El sistema actualiza los cambios.
Flujo Alternativo	
Acción del Actor	**Respuesta del Sistema**
	1.1 Si la licencia no existe el sistema envía un mensaje de aviso. 2.1 Si los datos no son válidos el sistema emite un mensaje de aviso.
Sección Eliminar licencia	
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El administrador busca la licencia que desea eliminar.	1.1 El sistema verifica si la licencia existe. 1.2 El sistema muestra la licencia.
2. El administrador selecciona la opción eliminar licencia.	2.1 El sistema elimina la licencia y se actualiza.
Flujo Alternativo	
Acción del Actor	**Respuesta del Sistema**
	1.1 Si la licencia no existe el sistema

	envía un mensaje de aviso.
Prioridad	Crítico

Tabla 2.9.13 Descripción textual del CU Gestionar licencia

Caso de Uso	**Buscar licencia**
Actores	Usuario
Propósito	Permitir a un usuario hacer la búsqueda de una licencia.
Resumen	El caso de uso inicia cuando el usuario selecciona la opción de buscar licencia, la búsqueda puede ser general o avanzada. El caso de uso culmina cuando el sistema muestra los resultados de la búsqueda.
Referencias	R6
Precondiciones	El usuario debe introducir al menos un parámetro para la búsqueda.
Poscondiciones	Los resultados de la búsqueda son mostrados.

Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El usuario selecciona la opción de buscar licencias.	1.1 El sistema muestra las dos formas de búsqueda existentes.
2. El usuario selecciona el tipo de búsqueda a realizar.	2.1 El sistema muestra el formulario de búsqueda seleccionado.
3. El usuario introduce al menos un dato de lo que desea buscar.	3.1 El sistema muestra la información que describe el usuario
Flujo Alternativo	
Acción del Actor	**Respuesta del Sistema**
	3.1 En caso de no existir ningún resultado el sistema envía un aviso. .
Prioridad	Auxiliar

Tabla 2.9.14 Descripción textual del CU Buscar licencia

Caso de Uso	**Confeccionar reporte**
Actores	Administrador
Propósito	Permitir al administrador mostrar los reportes del sistema
Resumen	El caso de uso inicia cuando el administrador escoge la opción confeccionar reporte, dentro del cual se puede mostrar los usuarios conectados y un ranking con las licencias mas solicitadas. Este finaliza cuando se han efectuado las acciones de control de reportes.
Referencias	R7
Precondiciones	El usuario debe estar autenticado.
Poscondiciones	Se muestra la información requerida.

Sección Mostrar usuarios conectados	
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El administrador escoge la opción mostrar usuarios conectados.	1.1 El sistema busca y muestra los datos de de los usuarios conectados (nombre del usuario, correo).
Flujo Alternativo	
Acción del Actor	**Respuesta del Sistema**

Sección Mostrar ranking de las licencias mas solicitadas	
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El administrador escoge la opción mostrar ranking de las licencias mas solicitadas.	1.1 El sistema hace una búsqueda en la lista de las licencias del catalogo mostrando en orden descendente las mas solicitadas.
Flujo Alternativo	

Acción del Actor	**Respuesta del Sistema**
Prioridad	Auxiliar

Tabla 2.9.14 Descripción textual del CU Confeccionar reporte

2.10 Conclusiones del capítulo.

En el presente capítulo se han descrito las características del sistema, precisando cómo se desarrollará y cada uno de los elementos que serán utilizados. Además se obtuvo un modelo del negocio, donde se muestra el comportamiento del sistema. Se precisaron los requisitos funcionales y no funcionales, así como los actores y los casos de uso correspondientes al sistema. También se ilustraron las relaciones entre los actores y casos de uso del sistema mediante un diagrama de casos de uso a partir del cual ya se tiene una mejor visión para realizar el análisis y el diseño del sistema.

Capítulo III: Análisis y diseño del sistema.

El presente capítulo se describe los elementos más importantes correspondientes a la etapa de análisis y diseño del sistema, entre los que se encuentran los diagramas de interacción del sistema por cada realización de casos de uso, así como el diagrama de clases del diseño. Es en esta fase se representan las clases que interactúan para realizar los distintos casos de uso y de esta manera dar cumplimiento a los requisitos funcionales. Se describe cómo se comunican las clases del sistema, y los usuarios que lo emplearán, además de otras restricciones inherentes al entorno de programación. Se incluye además en este capítulo el diagrama de las clases persistentes y el diagrama entidad relación de la base de datos.

3.1 Análisis del sistema

Durante esta etapa, se analizan los requisitos funcionales que se describen en el capítulo anterior, refinándolos y estructurándolos con el objetivo de conseguir una comprensión más precisa y una descripción más detallada que sea fácil de entender y que ayude a estructurar el sistema.

3.1.1 Clases del Análisis

Existen tres tipos de clases del análisis con sus funcionalidades propias, la primera es la clase interfaz la cual modelan la interacción entre el sistema y sus actores, le sigue la clase controladora que es la encargada de coordinar la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso y por ultimo la clase entidad que no es mas que información que posee larga vida y que es a menudo persistente.

3.1.2 Diagramas de Clases del Análisis

El Modelo de Clases del Análisis es un artefacto en el que se representan los conceptos del negocio. Representa las cosas del mundo real, no de la implementación automatizada de estas cosas. A continuación los diagramas correspondientes al análisis del sistema en cuestión:

Diagrama de clases de Análisis: Gestionar usuario

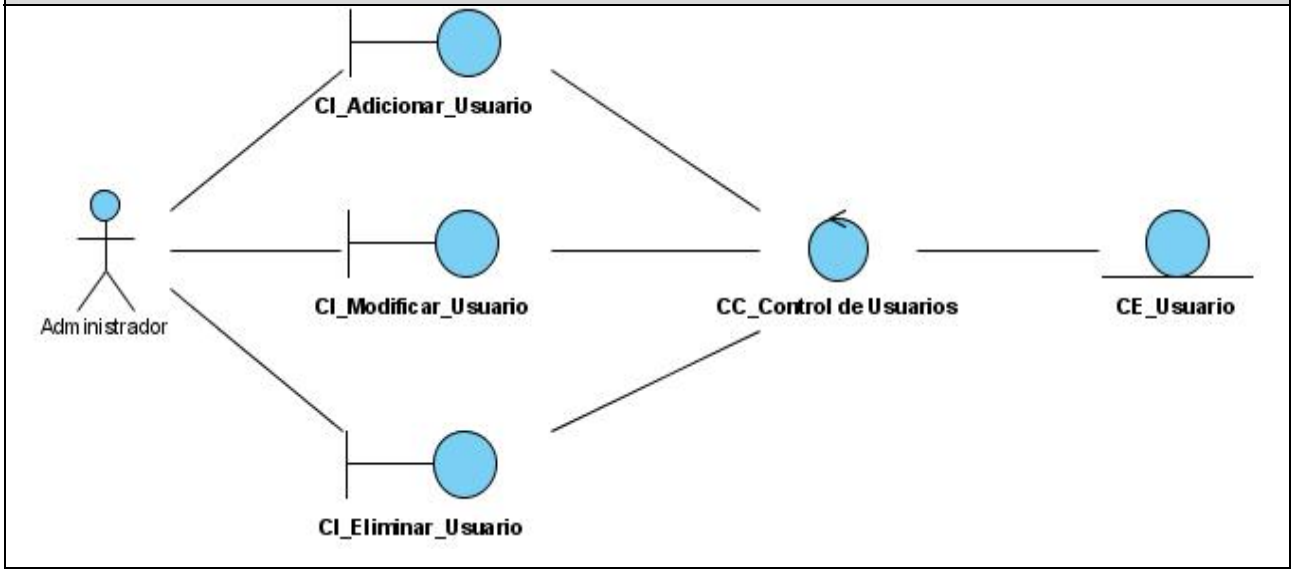


Figura 3.1 Diagrama de clases del análisis (CU Gestionar usuario)

Diagrama de clases de Análisis: Autenticar usuario

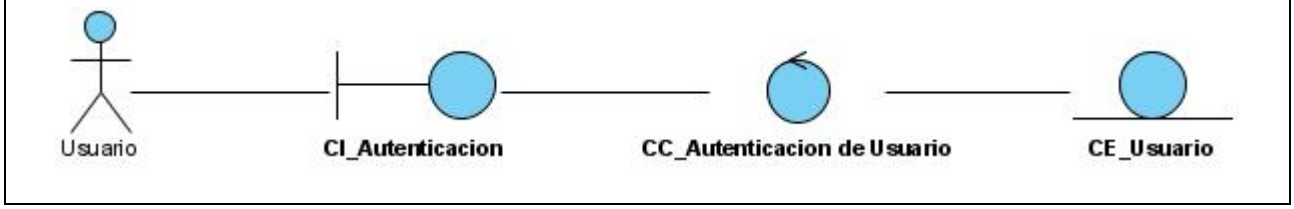


Figura 3.2 Diagrama de clases del análisis (CU Autenticar usuario)

Diagrama de clases de Análisis: Editar perfil de usuario

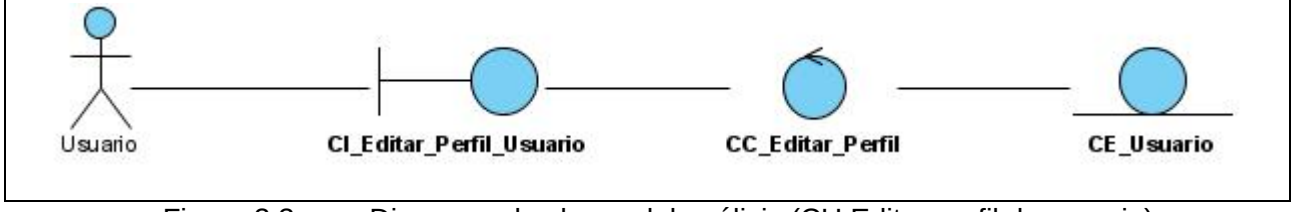


Figura 3.3 Diagrama de clases del análisis (CU Editar perfil de usuario)

Diagrama de clases de Análisis: Solicitar licencia

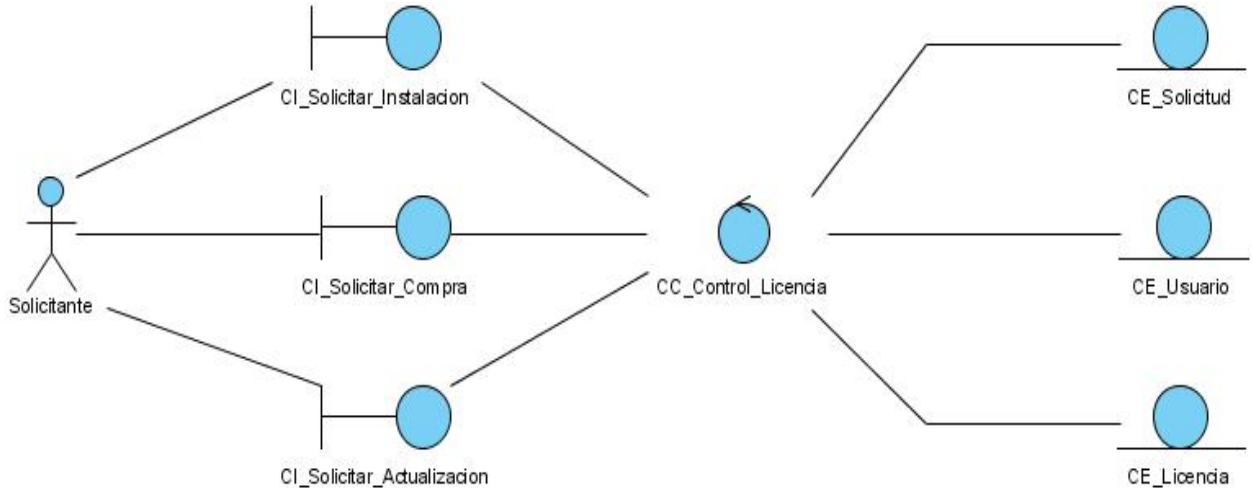


Figura 3.4 Diagrama de clases del análisis (CU Solicitar licencia)

Diagrama de clases de Análisis: Gestionar licencia

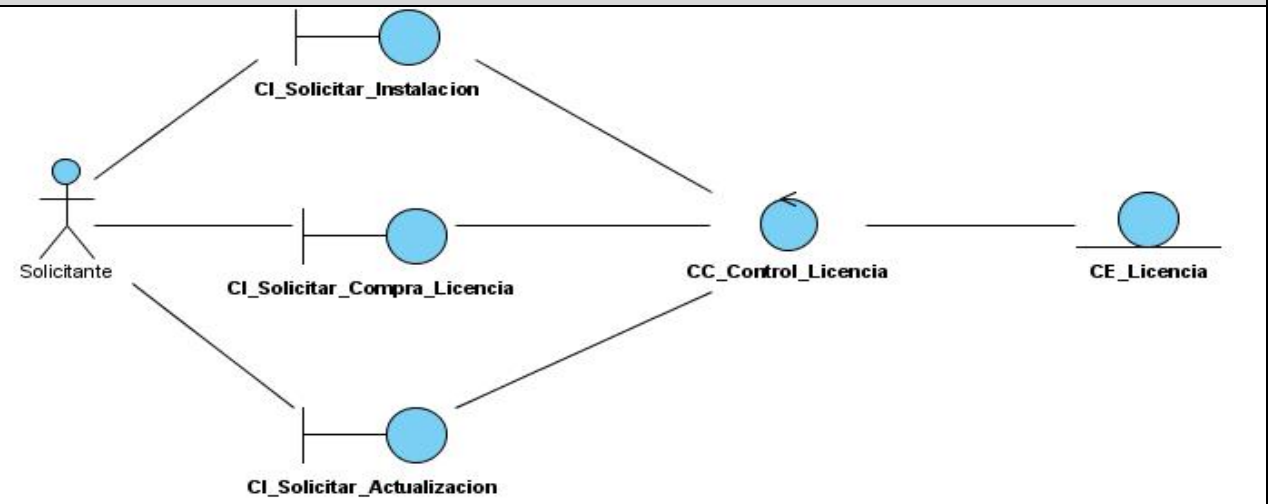


Figura 3.5 Diagrama de clases del análisis (CU Gestionar licencia)

Diagrama de clases de Análisis: Buscar licencia

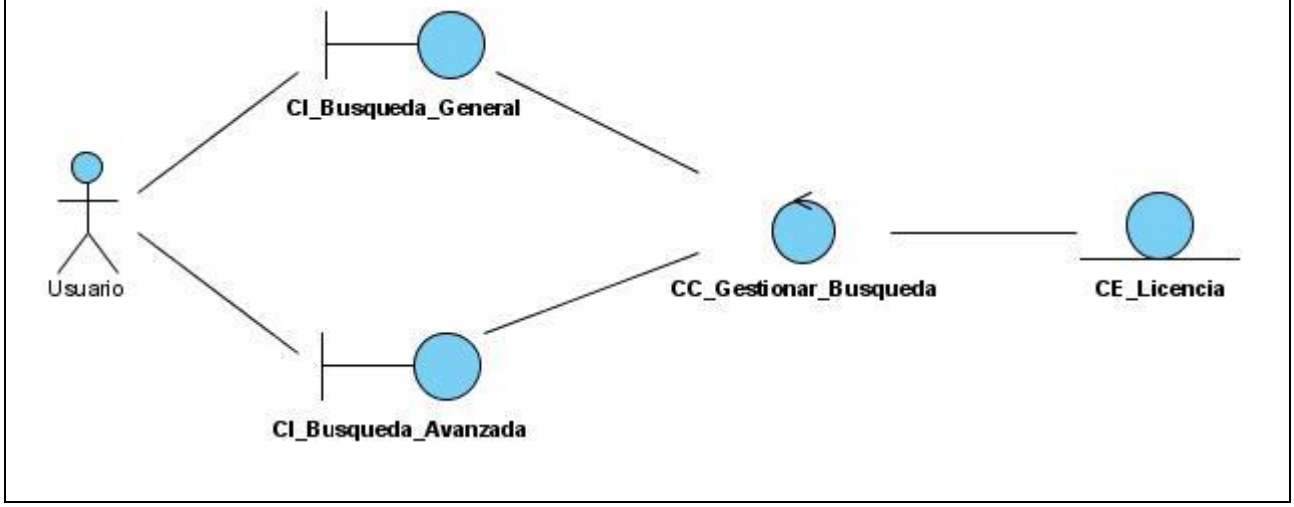


Figura 3.6 Diagrama de clases del análisis (CU Buscar licencia)

Diagrama de clases de Análisis: Confeccionar Reporte

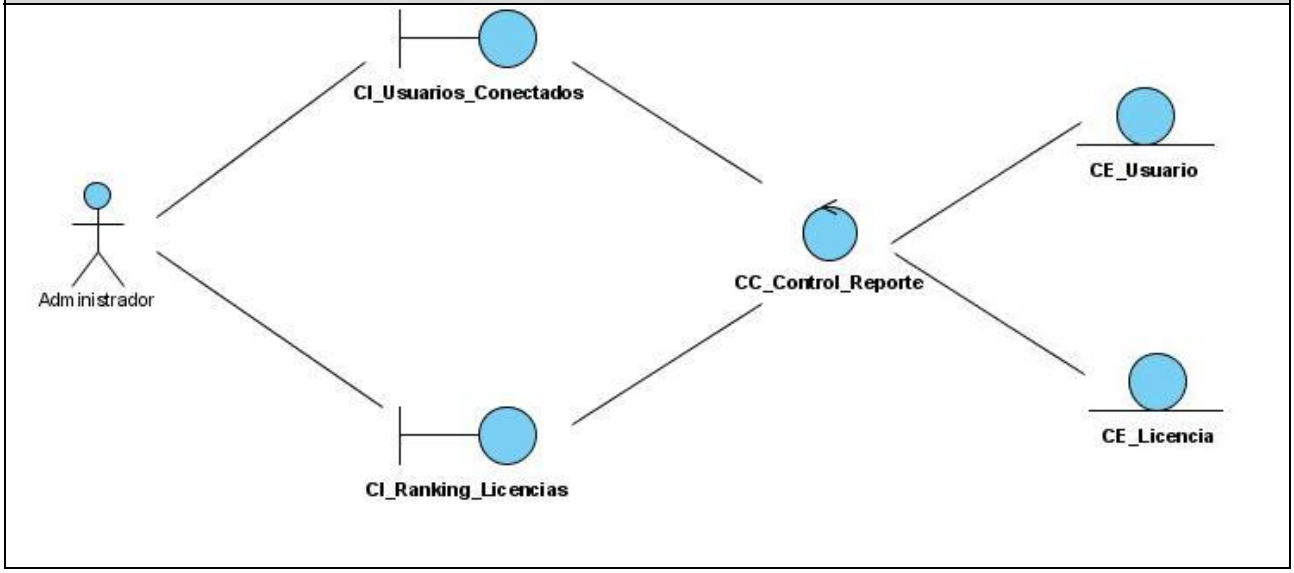


Figura 3.7 Diagrama de clases del análisis (CU Confeccionar Reporte)

3.1.3 Diagramas de Interacción

El lenguaje utilizado para ilustrar los diseños es, principalmente, los diagramas de interacción. UML incluye los diagramas de interacción para ilustrar el modo en el que los objetos interaccionan por medio de mensajes.

EL término diagrama de interacción es una generalización de dos tipos de diagramas UML más especializados; ambos pueden utilizarse para representar de forma similar interacciones de mensajes: Diagramas de colaboración y diagramas de secuencia.

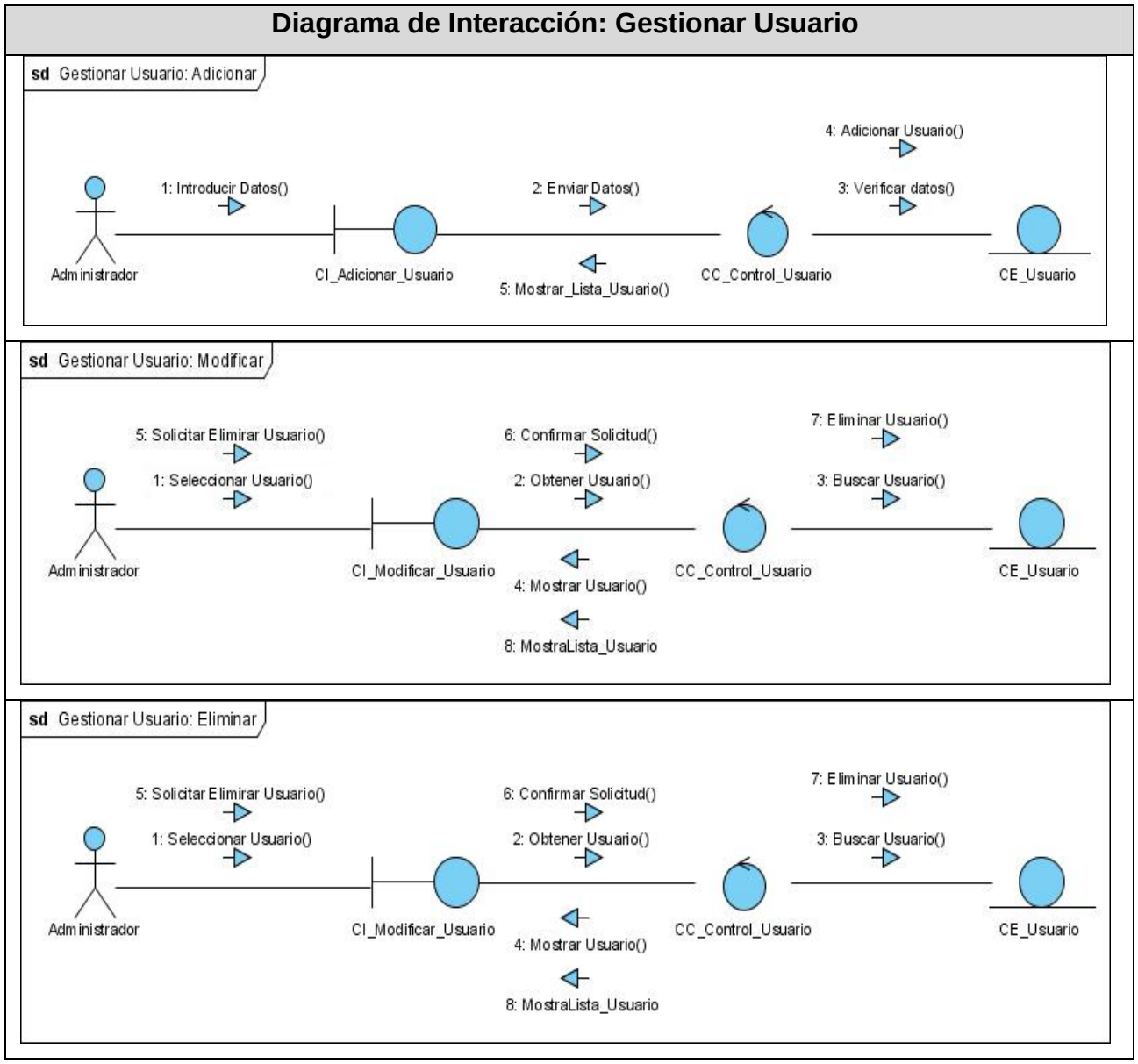


Figura 3.8 Diagrama de colaboración (CU Gestionar Usuario)

Diagrama de Interacción: Autenticar Usuario

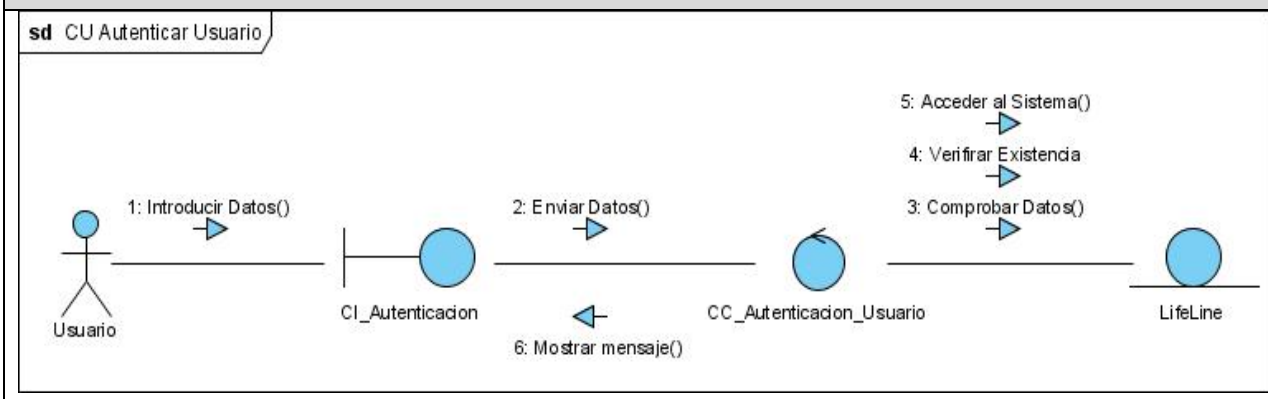


Figura 3.9 Diagrama de colaboración (CU Autenticar Usuario)

Diagrama de Interacción: Editar Perfil de Usuario

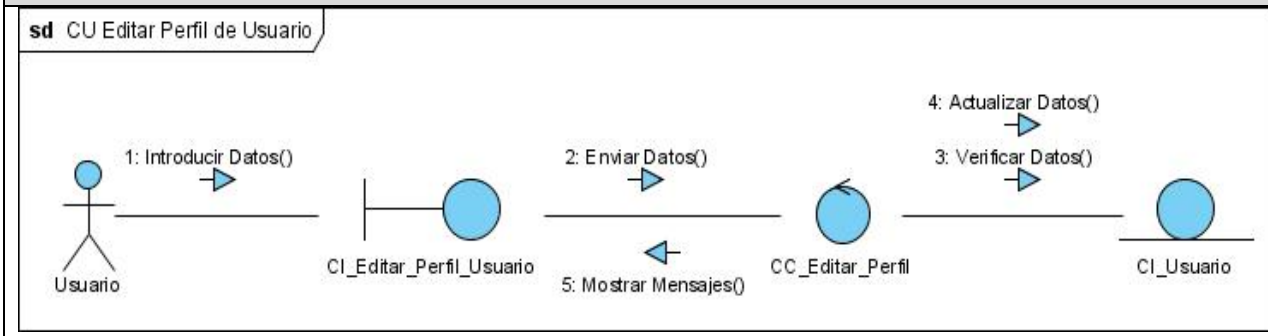


Figura 3.10 Diagrama de colaboración (CU Editar Perfil de Usuario)

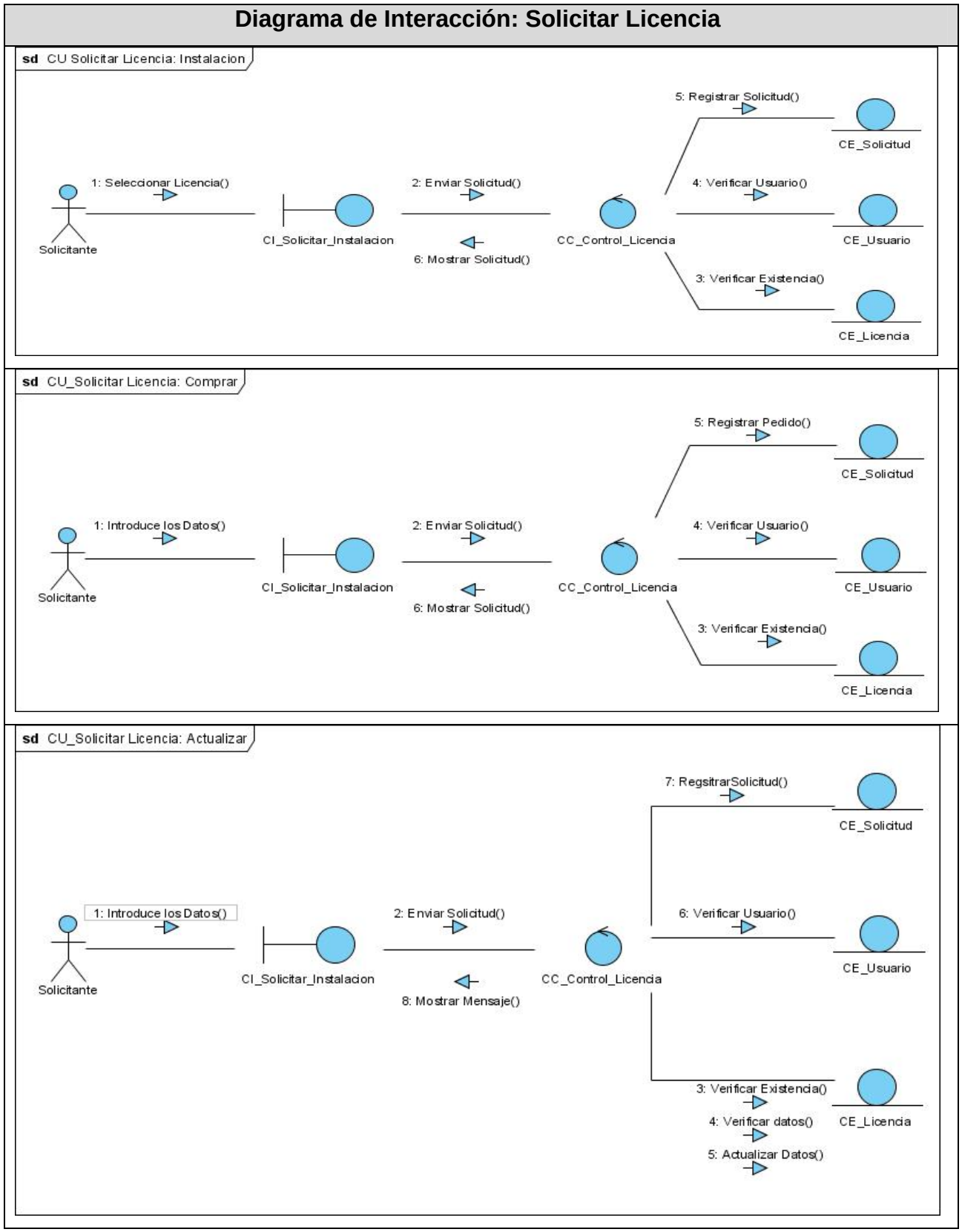


Figura 3.11 Diagrama de colaboración (CU Solicitar Licencia)

Diagrama de Interacción: Gestionar Licencias

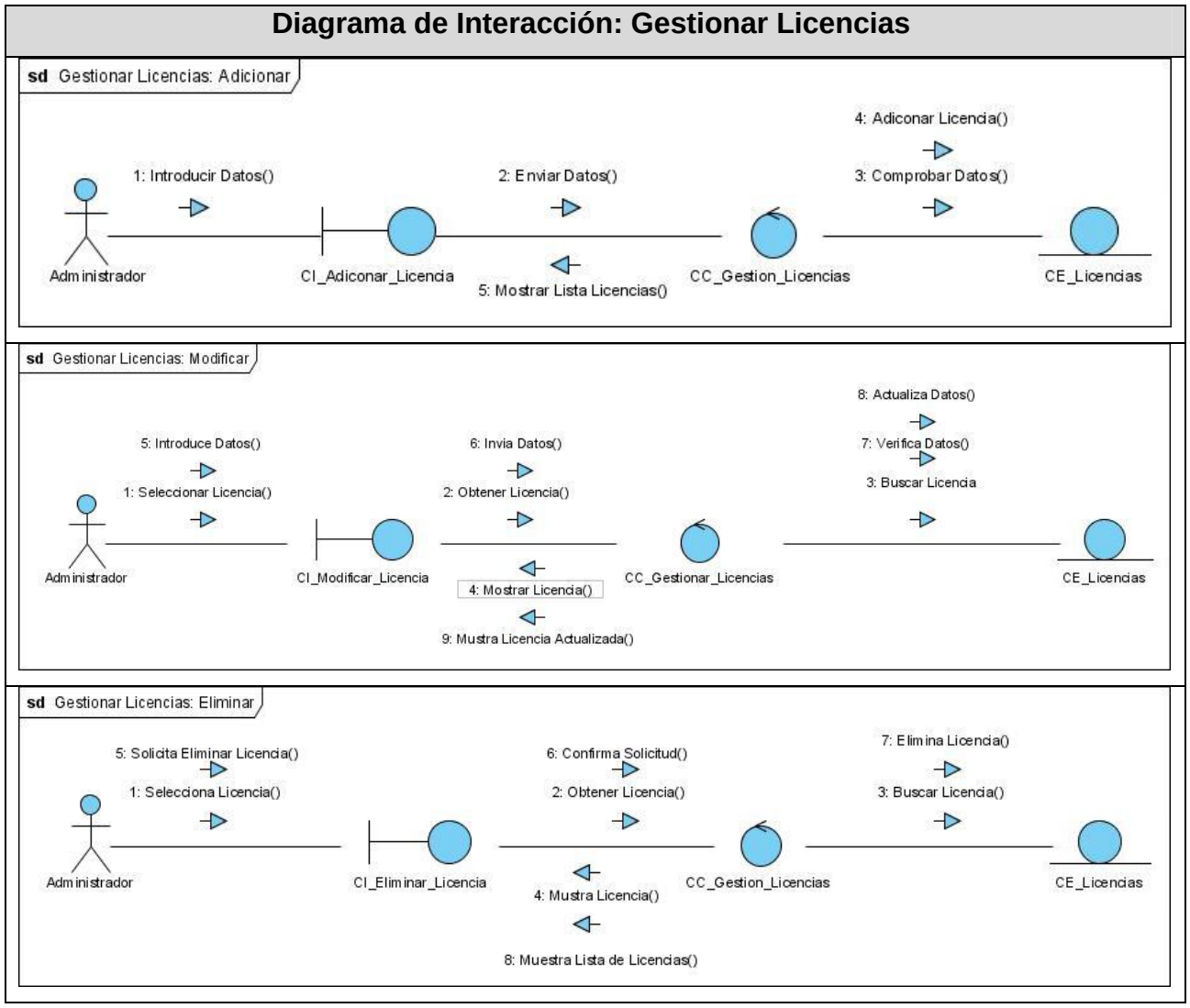


Diagrama de Interacción: Buscar Licencias

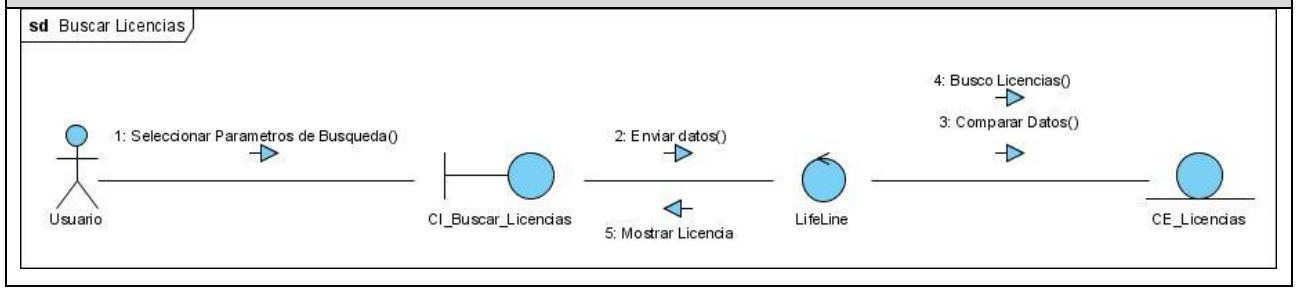
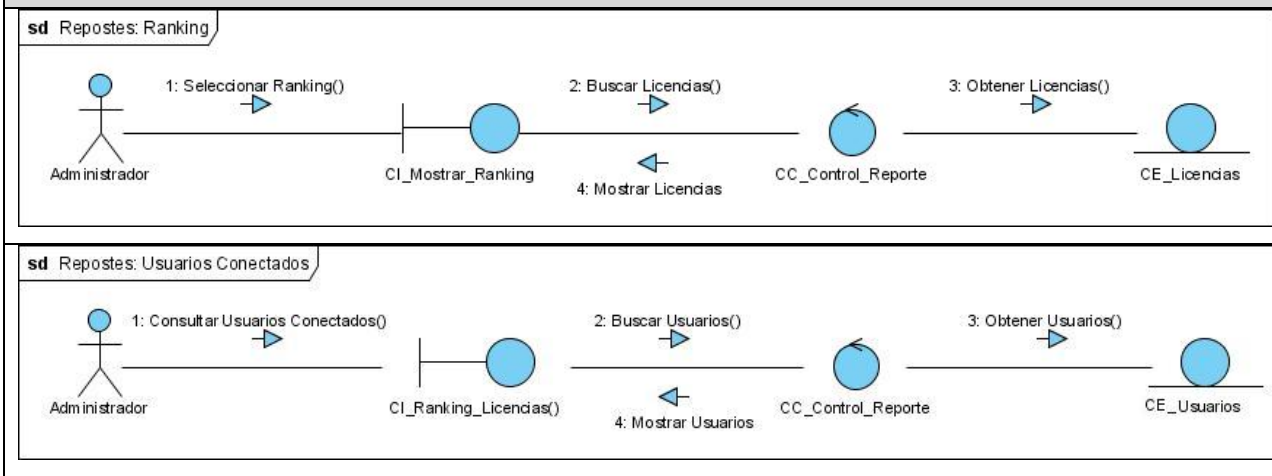


Diagrama de Interacción: Mostrar Reportes



3.2 Diagramas de Clases del Diseño Web.

El diagrama de diseño Web describe la realización física de los casos de uso, centrándose en como los requisitos funcionales y no funcionales junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema. Se representa de una forma sencilla como se lleva a cabo la colaboración y las responsabilidades de las distintas clases que forman el sistema. Se modelan las páginas, los enlaces entre estas, todo el código que irá creando las páginas, así como el contenido dinámico de las mismas, una vez que estén en el navegador del cliente.

Diagrama de Clases del Diseño Web: Adicionar Usuario

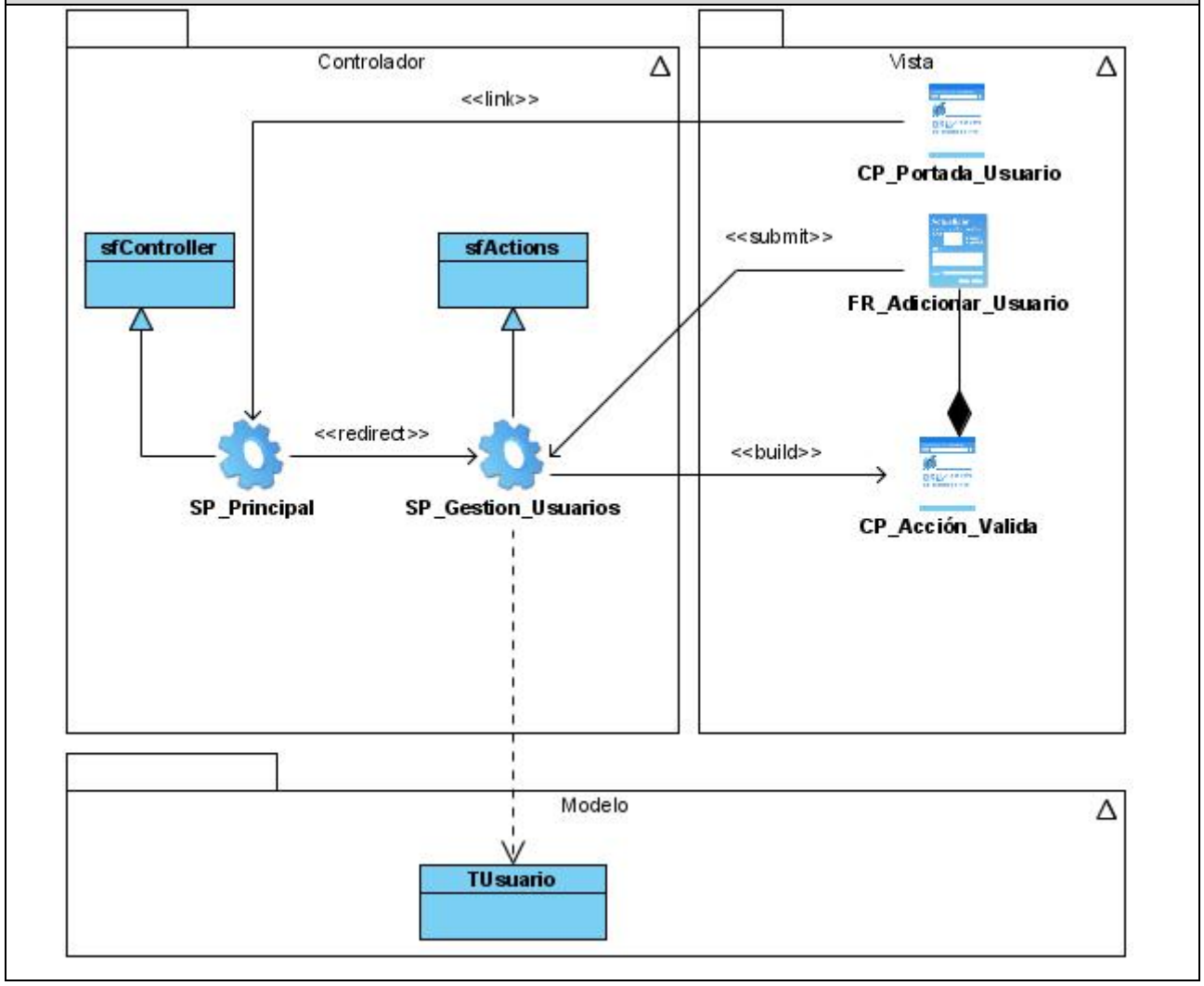


Figura 3.12 CU Gestionar Usuario: Sección Adicionar usuario.

Breve descripción del caso de uso:

Esta sección del caso de uso Gestionar usuario permite al administrador adicionar un nuevo usuario al sistema, actualizando la tabla TUsuario y muestra un lista de todos los usuarios existentes hasta el momento. Para comprender mejor la manera en que ocurren los procesos remítase al ANEXO I.

Diagrama de Clases del Diseño Web: Eliminar Usuario

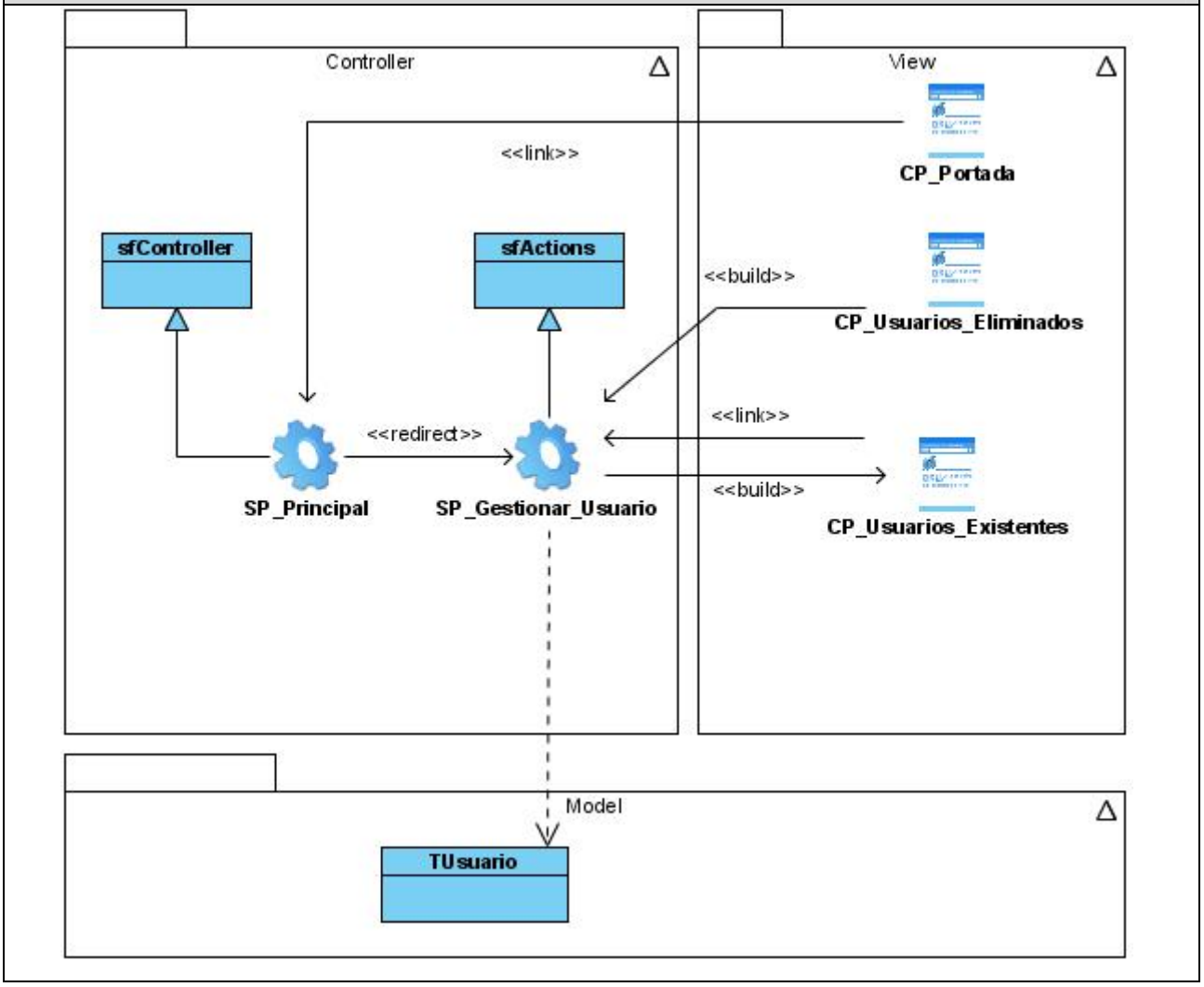


Figura 3.13 CU Gestionar Usuario: Sección Eliminar usuario.

Breve descripción del caso de uso:

Esta sección del caso de uso Gestionar usuario permite al administrador eliminar un usuario del sistema escogiéndolo de forma sencilla y queda actualizada la tabla TUsuario y muestra un lista de todos los usuarios eliminados en ese momento. Para comprender mejor la manera en que ocurren los procesos remítase al ANEXO II.

Diagrama de Clases del Diseño Web: Modificar Usuario

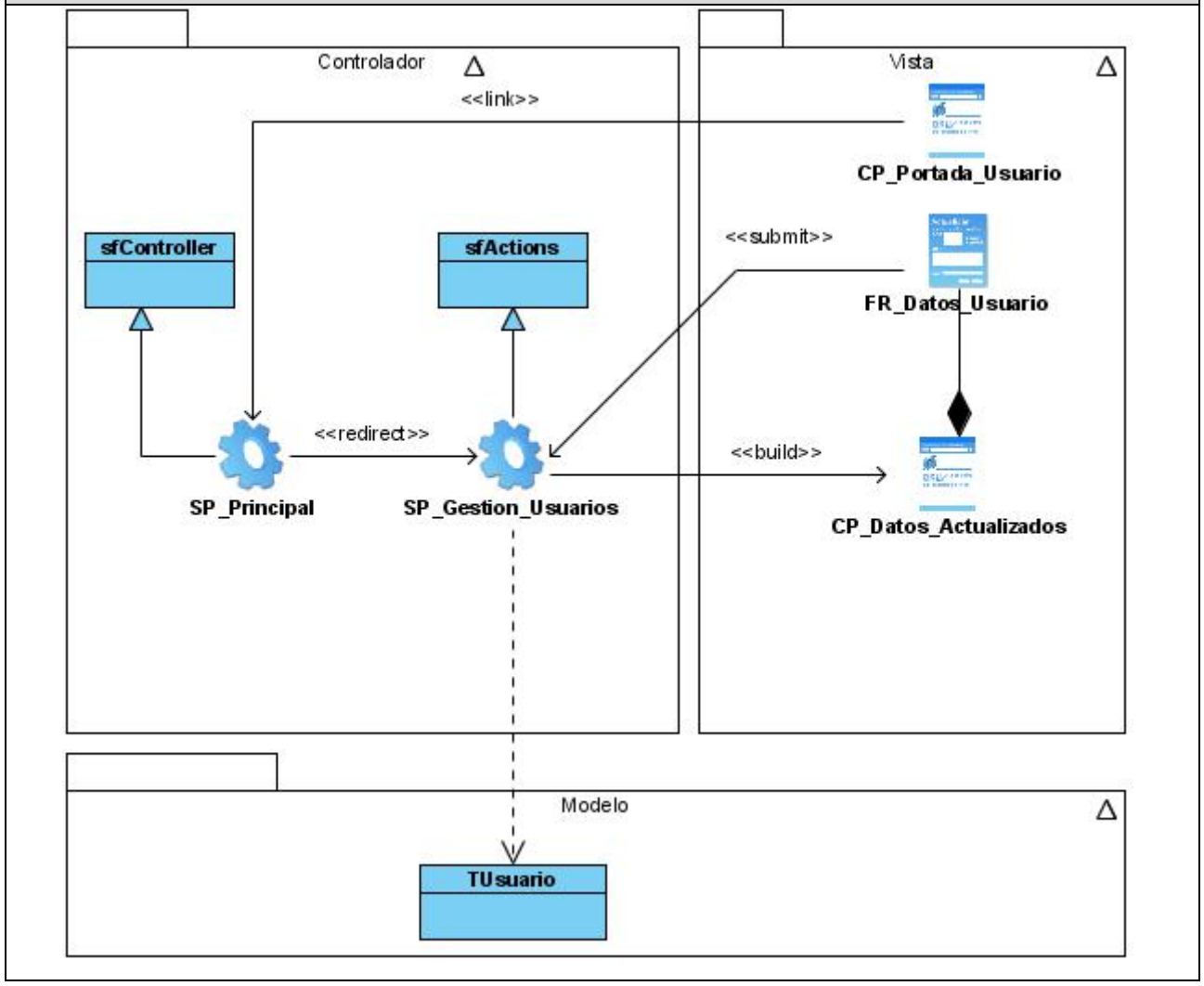


Figura 3.14 CU Gestionar Usuario: Sección Modificar usuario.

Breve descripción del caso de uso:

Esta sección del caso de uso Gestionar usuario permite al administrador actualizar los datos un usuario a través de un formulario y queda actualizada la tabla TUsuario y muestra los datos del usuario, ya modificados. Para comprender mejor la manera en que ocurren los procesos remítase al ANEXO III.

Diagrama de Clases del Diseño Web: Autenticar Usuario

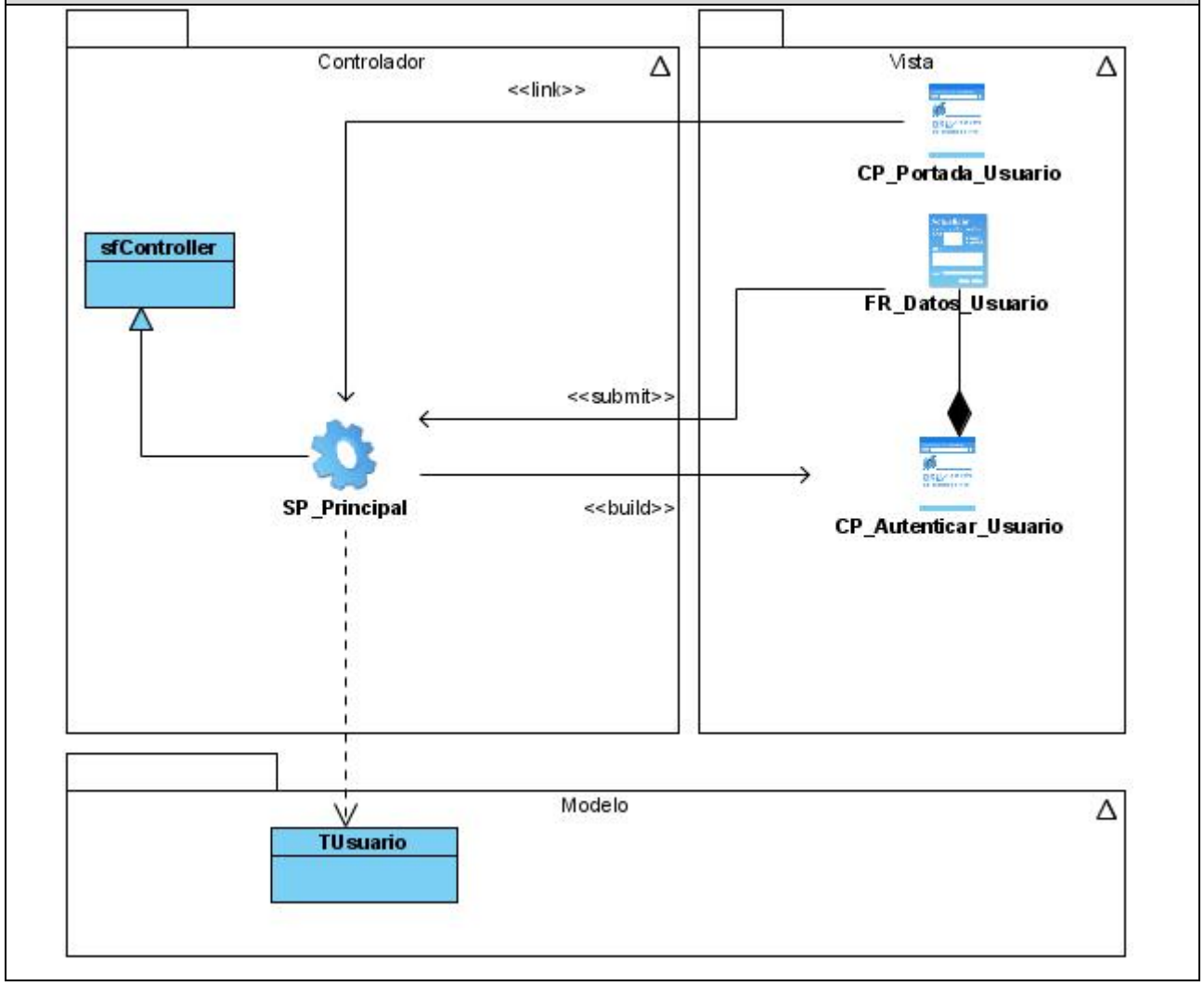


Figura 3.15 CU Autenticar Usuario

Breve descripción del caso de uso:

El caso de uso Autenticar usuario permite la autenticación de un usuario en el sistema verificando la valide de los datos introducidos a través de un formulario en la tabla TUsuario. Para comprender mejor la manera en que ocurren los procesos remítase al ANEXO IV.

Diagrama de Clases del Diseño Web: Editar perfil de usuario

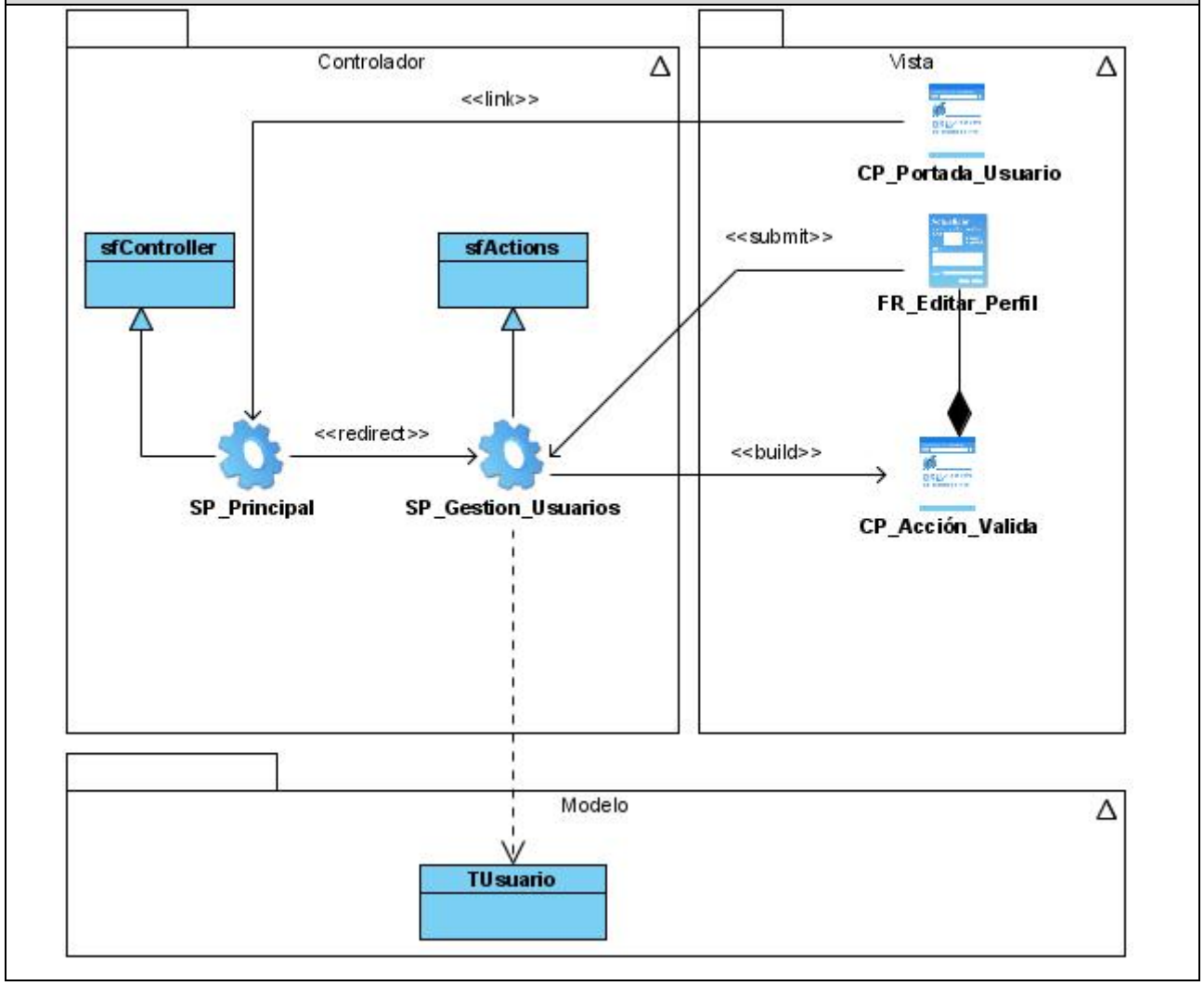


Figura 3.16 CU Editar perfil de usuario

Breve descripción del caso de uso:

El caso de uso Editar perfil de usuario permite modificar los datos propios de cada usuario en particular, estos son mostrados para que se vean los cambios realizados. Para comprender mejor la manera en que ocurren los procesos remítase al ANEXO V.

Diagrama de Clases del Diseño Web: Solicitar préstamo o instalación

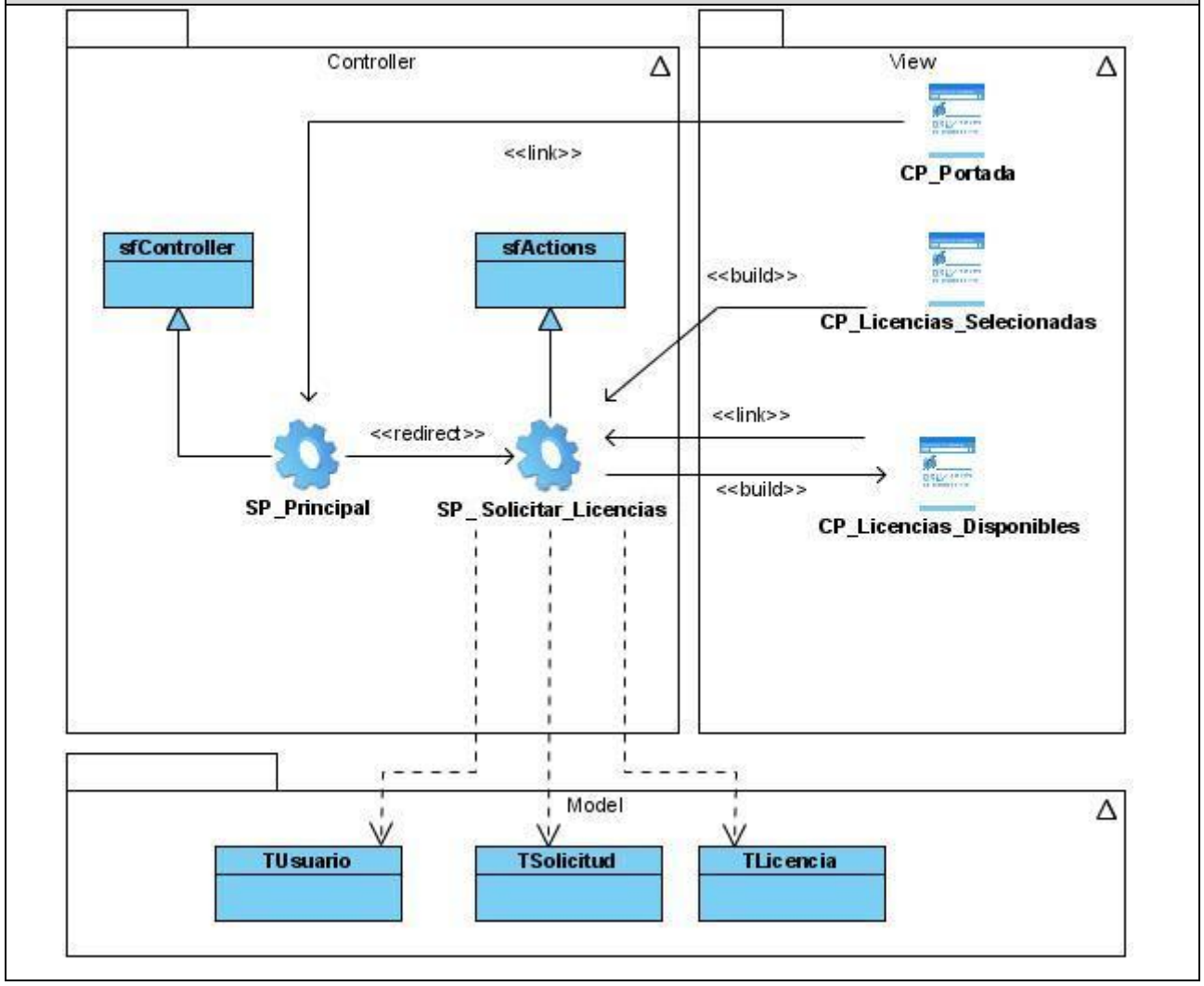


Figura 3.17 CU Solicitar licencia: Sección Solicitar préstamo o instalación.

Breve descripción del caso de uso:

Esta sección del caso de uso Solicitar licencia permite al solicitante pedir la instalación de una licencia presente en el catalogo, esto se hace con tan solo seleccionarla de la lista de que se muestra, la petición queda registrada en la tabla TSolicitud. Para comprender mejor la manera en que ocurren los procesos remítase al ANEXO VI.

Diagrama de Clases del Diseño Web: Solicitar compra

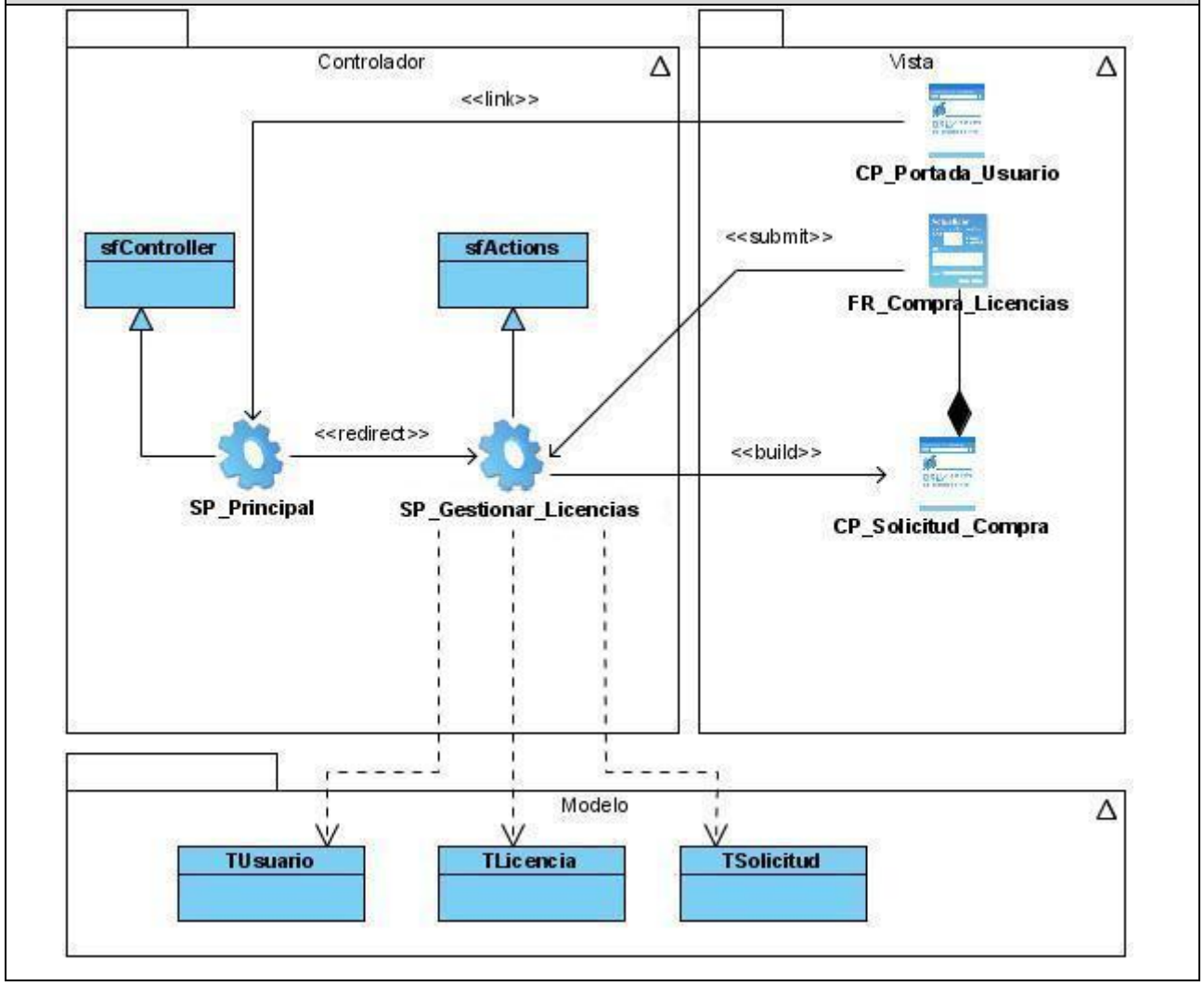


Figura 3.18 CU Solicitar licencia: Sección Solicitar compra.

Breve descripción del caso de uso:

Esta sección del caso de uso Solicitar licencia permite al solicitante hacer el pedido de compra de una licencia no existente en el catalogo, esto es posible a través de un formulario y queda registrada la solicitud en la tabla TSolicitud. Para comprender mejor la manera en que ocurren los procesos remítase al ANEXO VII.

Diagrama de Clases del Diseño Web: Solicitar actualización

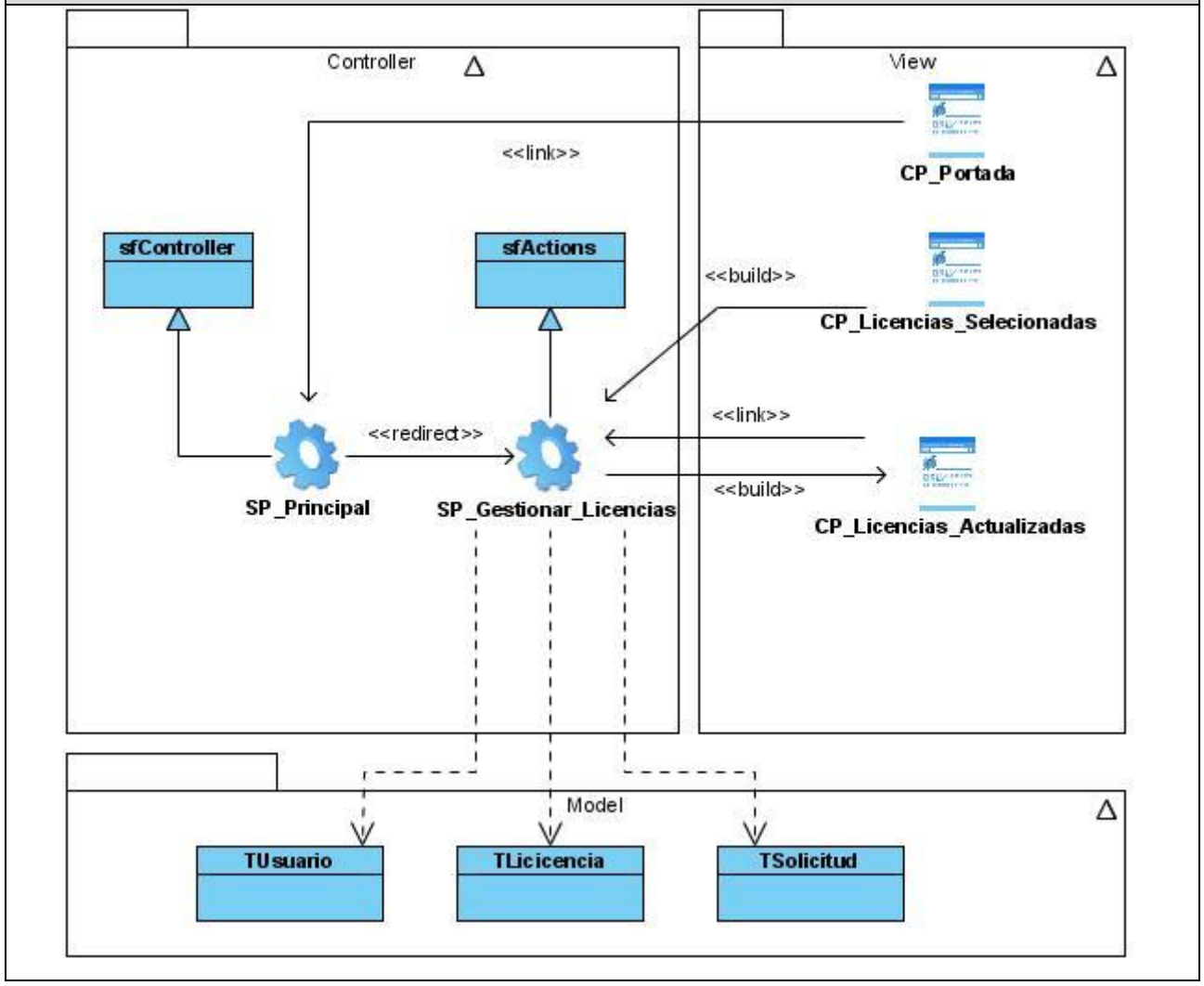


Figura 3.19 CU Solicitar licencia: Sección Solicitar actualización.

Breve descripción del caso de uso:

Esta sección del caso de uso Solicitar licencia permite al solicitante pedir la actualización de una licencia lista para actualizar, esto se hace con tan solo seleccionarla de la lista que se muestra en la pagina cliente que corresponde, la petición queda registrada en la tabla TSolicitud. Para comprender mejor la manera en que ocurren los procesos remítase al ANEXO VIII.

Diagrama de Clases del Diseño Web: Adicionar licencia

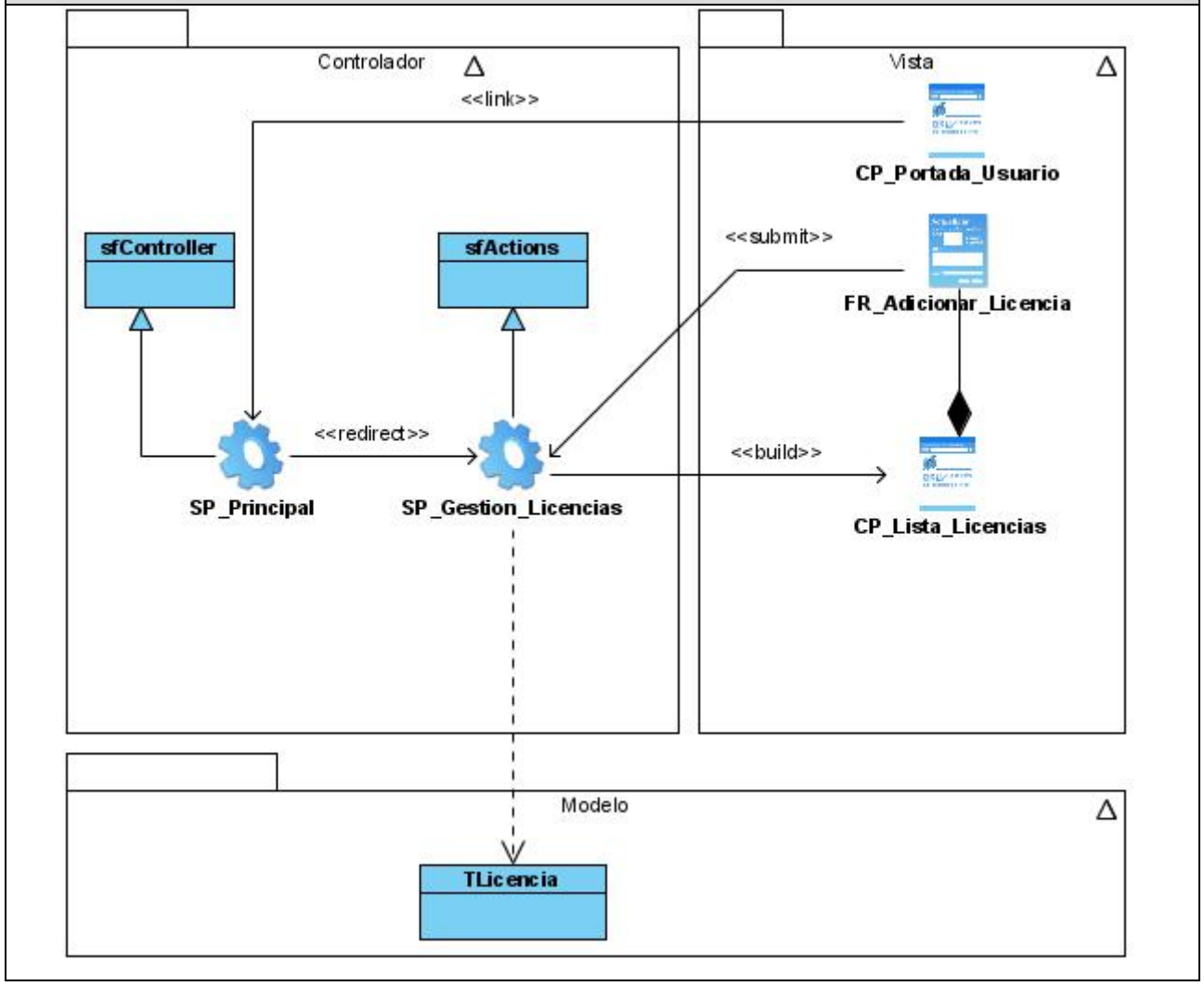


Figura 3.20 CU Gestionar Licencia: Sección Adicionar licencia.

Breve descripción del caso de uso:

Esta sección del caso de uso Gestionar licencia permite al administrador adicionar una nueva licencia al catalogo, actualizando la tabla TLicencia y muestra un lista de todas las licencias existentes hasta el momento. Para comprender mejor la manera en que ocurren los procesos remítase al ANEXO IX.

Diagrama de Clases del Diseño Web: Actualizar licencia

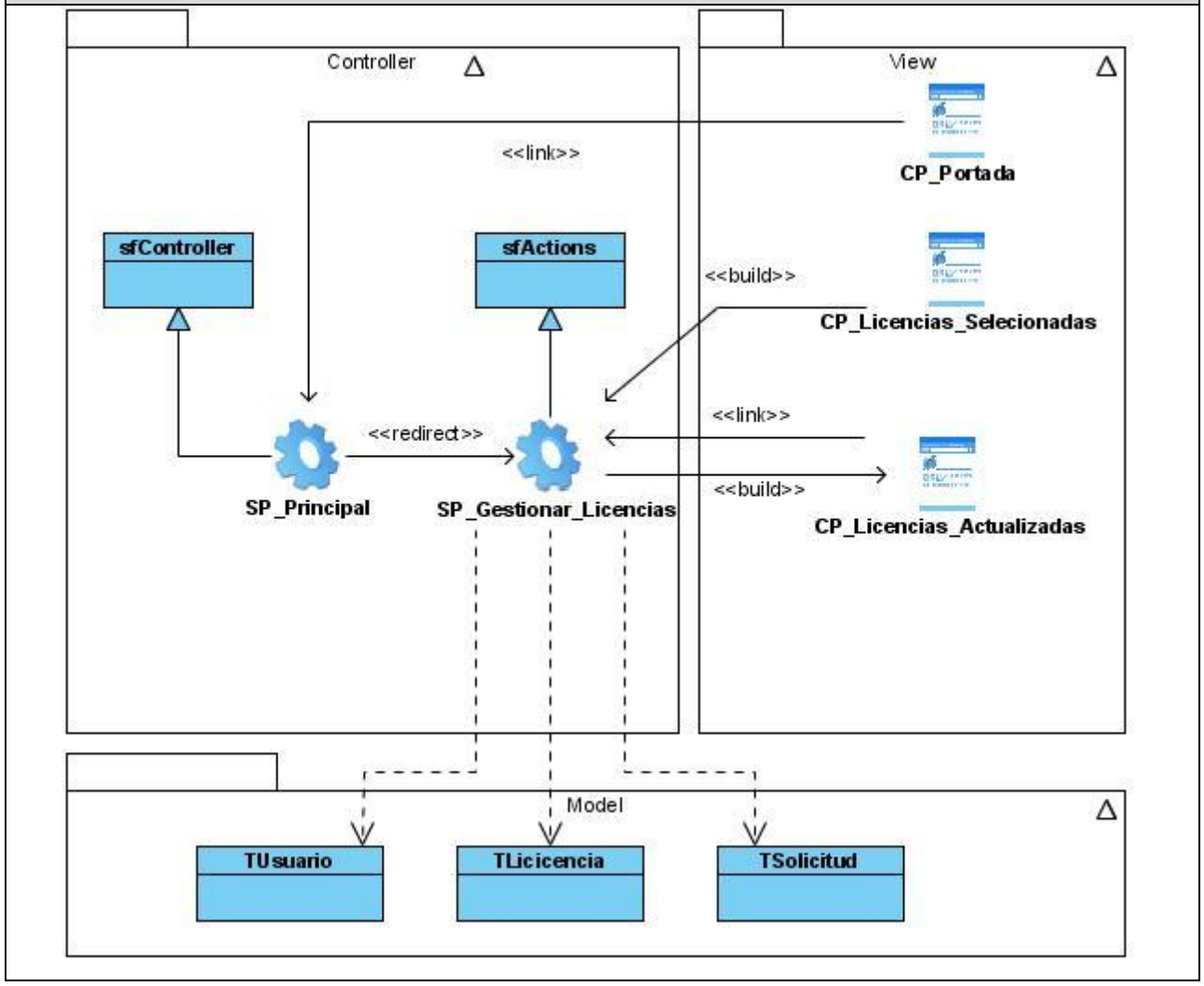


Figura 3.21 CU Gestionar Licencia: Sección Actualizar licencia.

Breve descripción del caso de uso:

Esta sección del caso de uso Gestionar licencia permite al administrador actualizar los datos una licencia a través de un formulario y queda actualizada la tabla TLicencia y muestra los datos del la licencia, ya modificados. Para comprender mejor la manera en que ocurren los procesos remítase al ANEXO X.

Diagrama de Clases del Diseño Web: Eliminar licencia

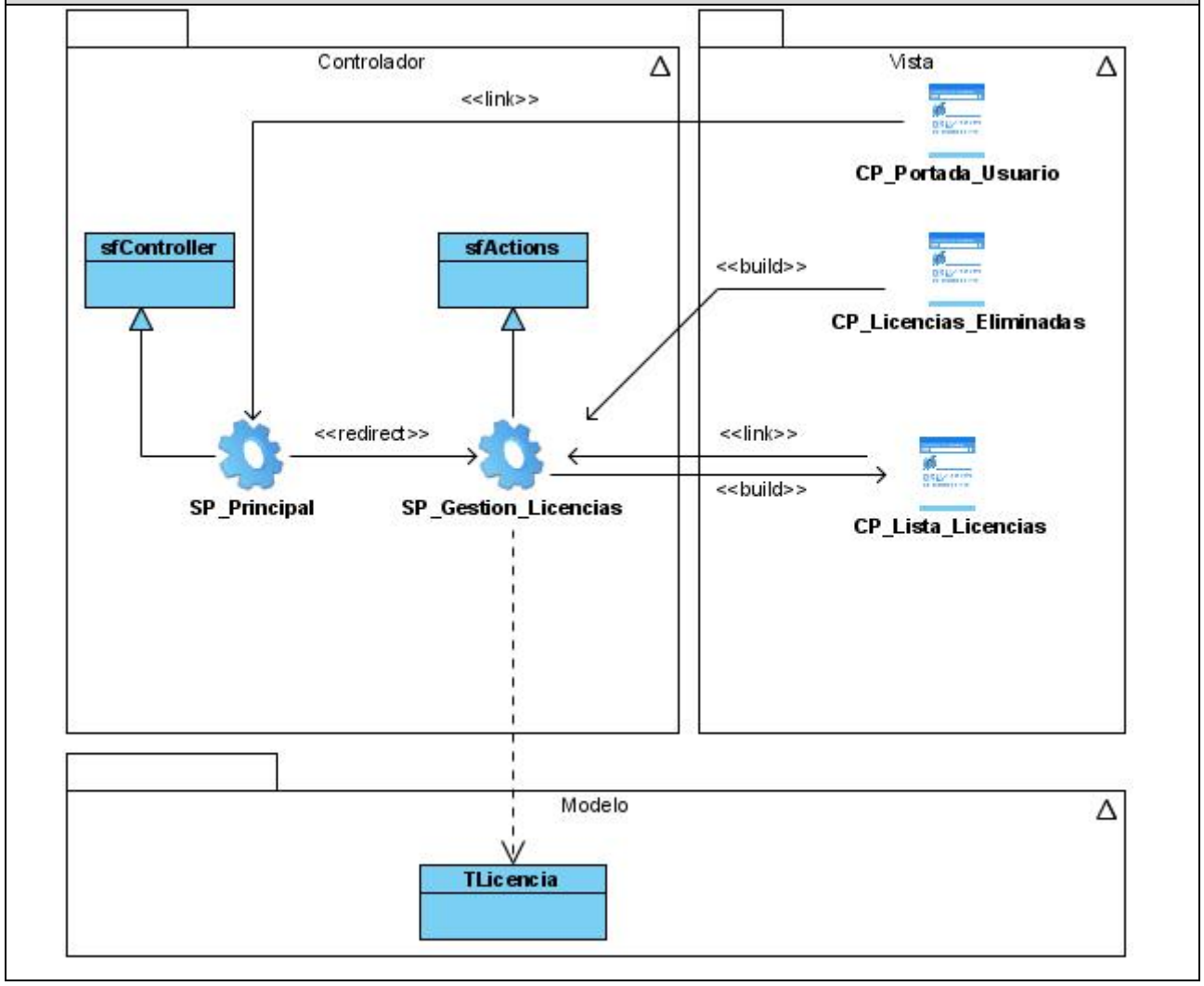


Figura 3.22 CU Gestionar Licencia: Sección Eliminar licencia.

Breve descripción del caso de uso:

Esta sección del caso de uso Gestionar licencia permite al administrador eliminar una licencia del catalogo escogiéndola de forma sencilla y queda actualizada la tabla TLicencia, además muestra una lista de todas las licencias eliminadas en ese momento. Para comprender mejor la manera en que ocurren los procesos remítase al ANEXO XI.

Diagrama de Clases del Diseño Web: Búsqueda general

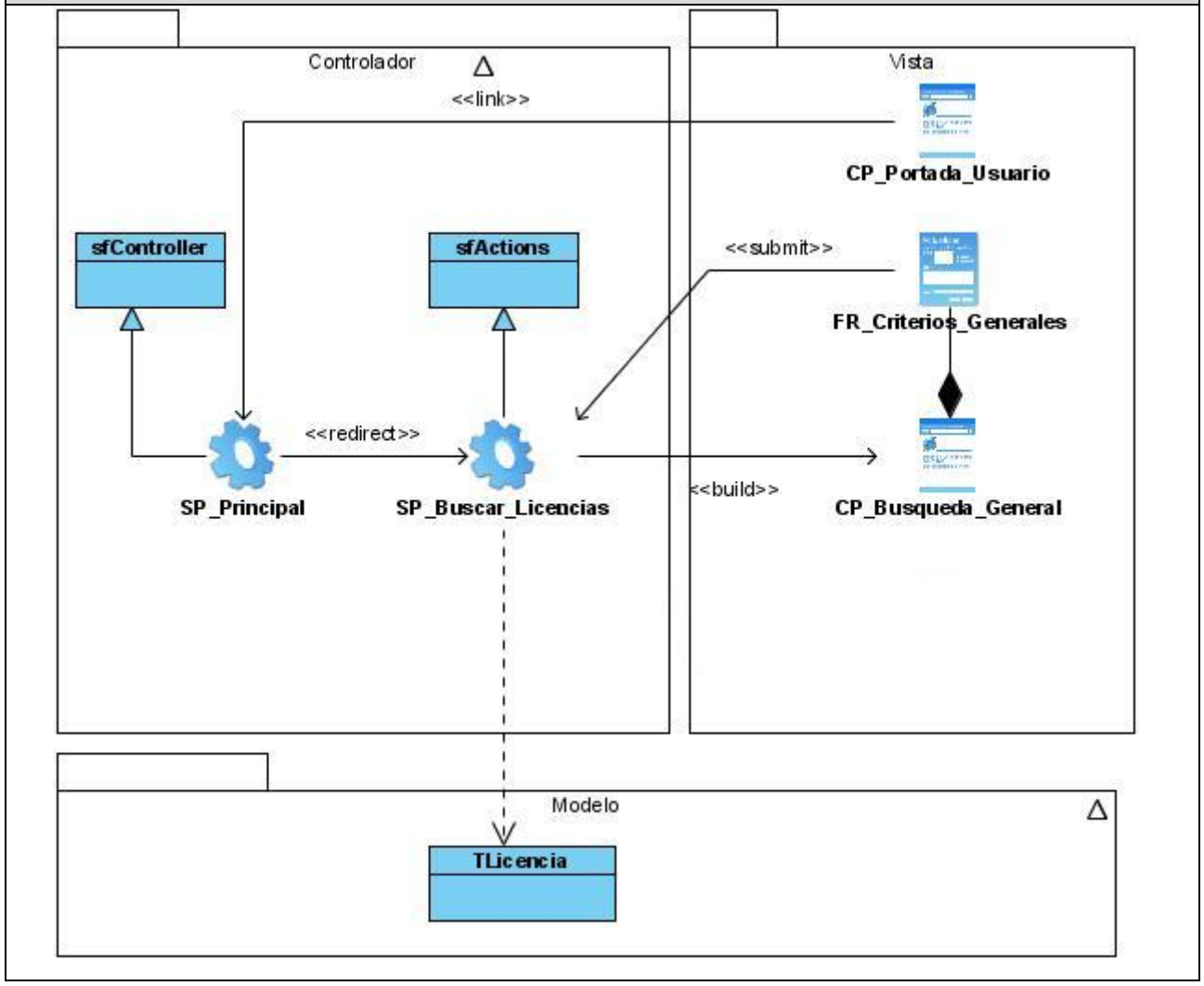


Figura 3.23 CU Buscar Licencia: Sección Búsqueda general.

Breve descripción del caso de uso:

Esta sección del caso de uso Buscar licencia permite a los usuarios hacer una búsqueda de licencias sin especificar parámetros muy particulares de las mismas, y muestra los resultados alcanzados en el proceso. Para comprender mejor la manera en que ocurren los procesos remítase al ANEXO XII.

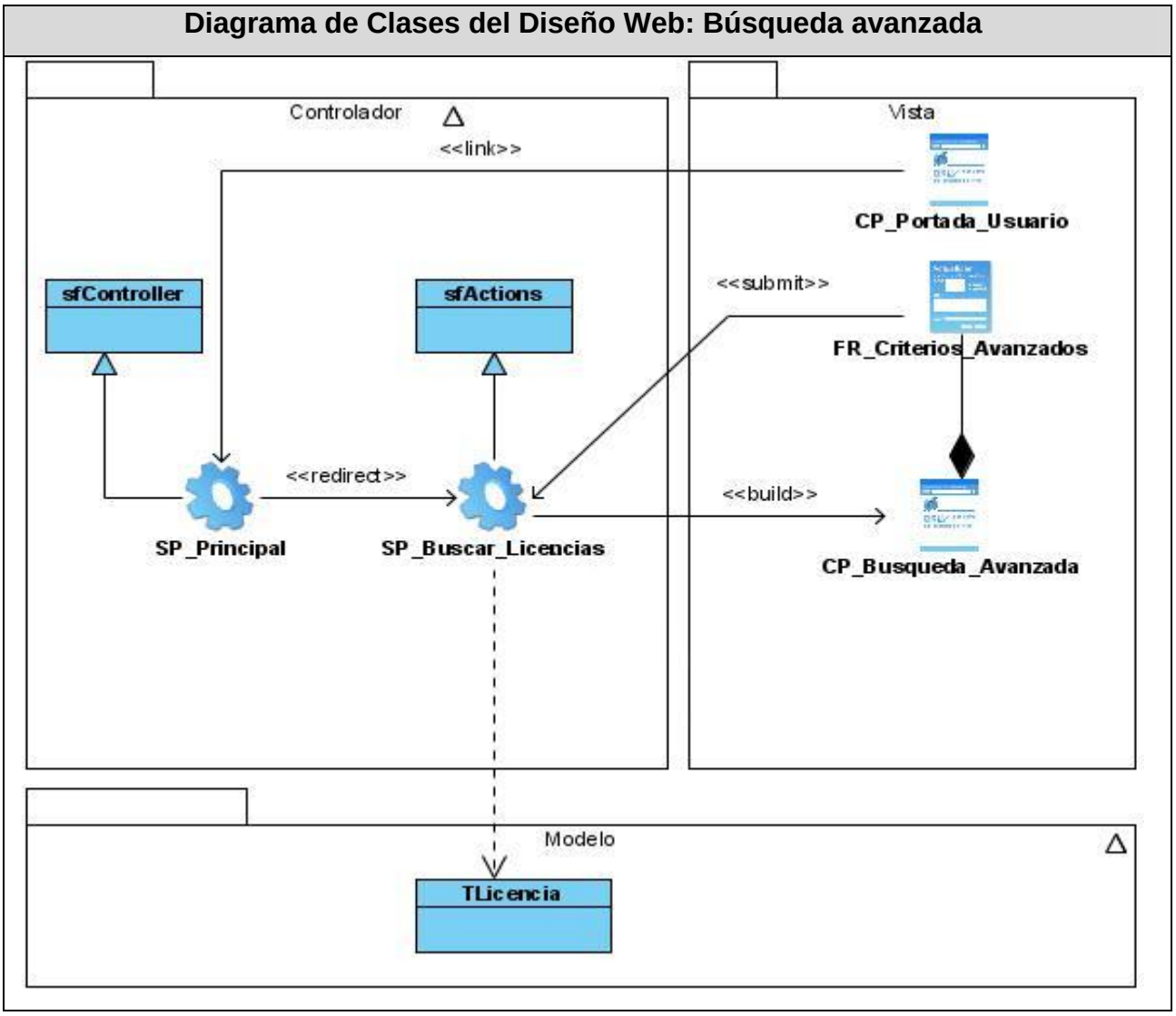


Figura 3.24 CU Buscar Licencia: Sección Búsqueda avanzada.

Breve descripción del caso de uso:

Esta sección del caso de uso Buscar licencia permite a los usuarios hacer una búsqueda de licencias de manera especifica, fijando parámetros de búsquedas mas restringidos. Para comprender mejor la manera en que ocurren los procesos remítase al ANEXO XIII

Diagrama de Clases del Diseño Web: Confeccionar reporte

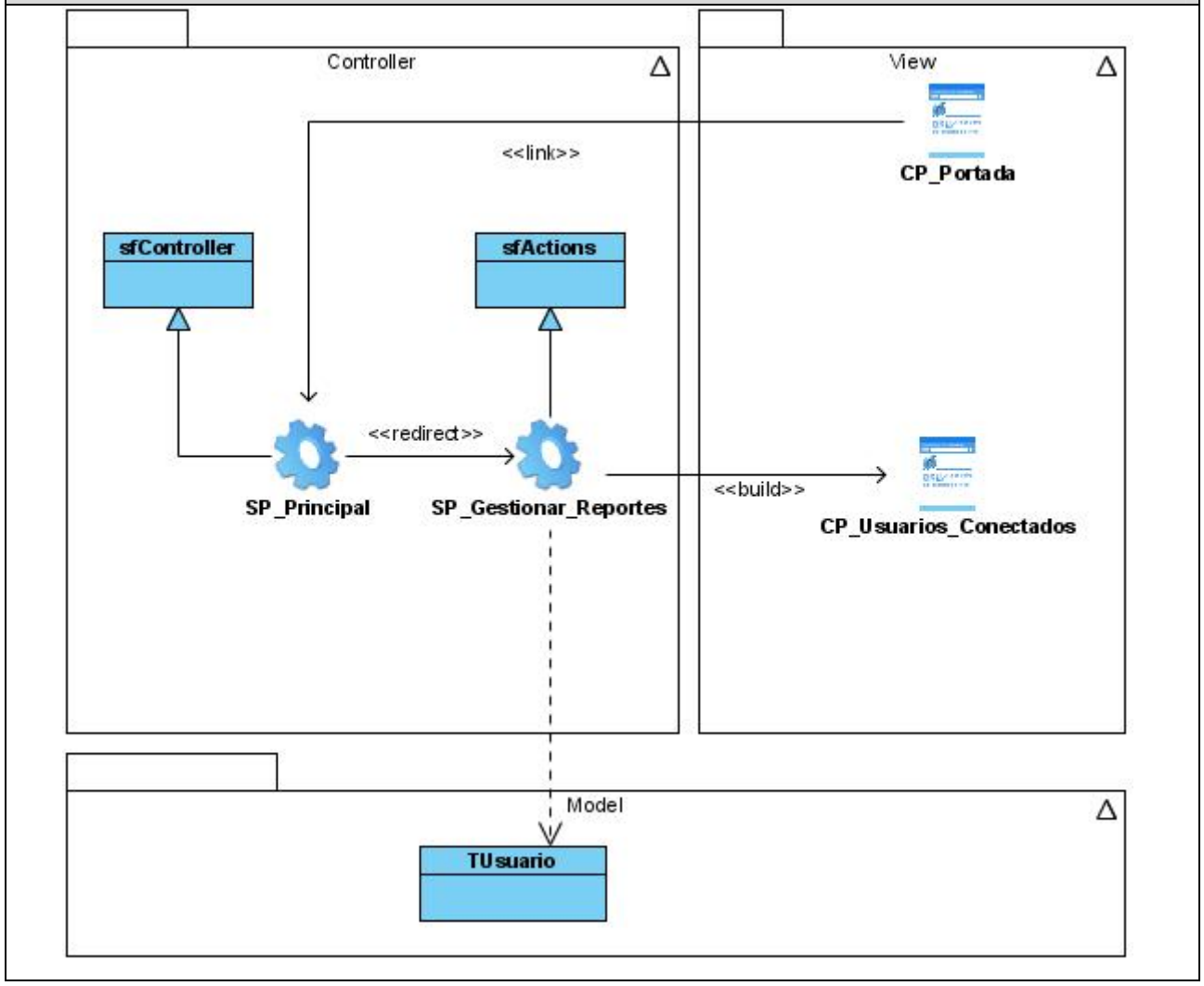


Figura 3.25 CU Confeccionar reporte: Sección Mostrar usuarios conectados.

Breve descripción del caso de uso:

Esta sección del caso de uso Confeccionar reporte permite al administrador consultar los usuarios autenticados en el sistema en un momento determinado. Para comprender mejor la manera en que ocurren los procesos remítase al ANEXO XIV.

Diagrama de Clases del Diseño Web: Confeccionar reporte

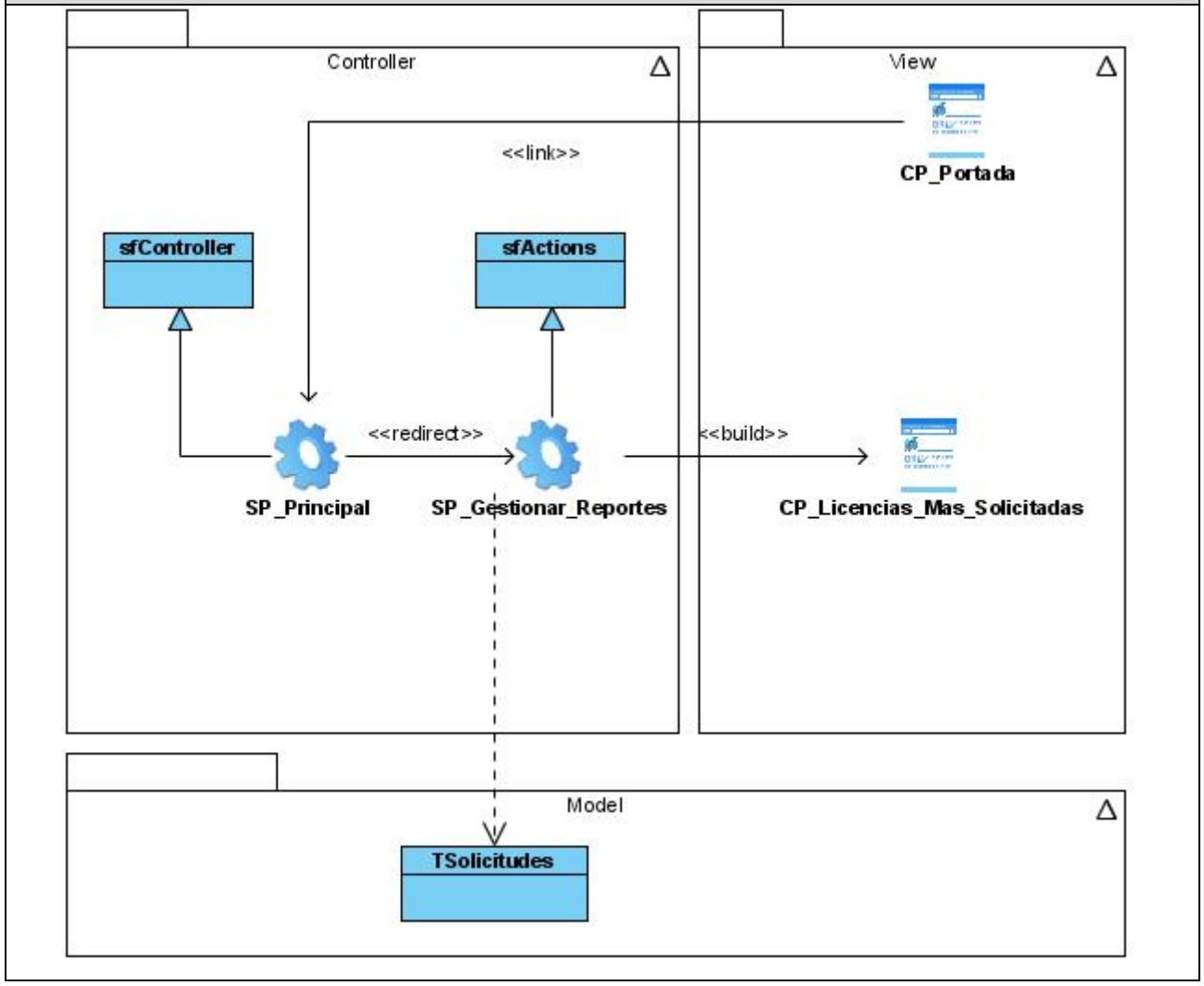


Figura 3.26 CU Confeccionar reporte: Sección Mostrar ranking de licencias mas descargadas.

Breve descripción del caso de uso:

Esta sección del caso de uso Confeccionar reporte permite al administrador hacer una lista de las licencias mas solicitadas en por los solicitantes, para ello consulta la tabla TSolicitudes. Para comprender mejor la manera en que ocurren los procesos remítase al ANEXO XV.

3.2.1 Descripción de las clases del diseño

Nombre: SP_Gestión_Usuarios	
Tipo de clase: Controladora	
Por cada responsabilidad:	
Nombre:	Adicionar_usuario (id_usuario, contraseña, privilegios, nombre, apellido, área, correo).
Descripción:	Permite adicionar un nuevo usuario al sistema.
Nombre:	Eliminar_usuario (id_usuario)
Descripción:	Permite eliminar usuario y actualiza la base de datos.
Nombre:	Modificar_usuario (contraseña, privilegios, nombre, apellido, área, correo)
Descripción:	Permite modificar los datos del usuario.

Tabla 3.1 Descripción de las clases del diseño: SP_Gestión_Usuarios.

Nombre: SP_Principal	
Tipo de clase: Controladora	
Por cada responsabilidad:	
Nombre:	Autenticar_usuario (id_usuario, contraseña)
Descripción:	Permite verificar que el usuario exista y le asigna los permisos pertinentes.

Tabla 3.2 Descripción de las clases del diseño: SP_Principal.

Nombre: SP_Editar_perfil	
Tipo de clase: Controladora	
Por cada responsabilidad:	
Nombre:	Editar_perfil (contraseña, nombre, apellido, correo).
Descripción:	Permite modificar el perfil del usuario.

Tabla 3.3 Descripción de las clases del diseño: SP_Editar_perfil.

Nombre: SP_Solicitar_licencia	
Tipo de clase: Controladora	
Por cada responsabilidad:	
Nombre:	Solicitar_instalación (id_licencia).
Descripción:	Permite solicitar la instalación de una licencia existente en el catalogo.
Nombre:	Solicitar_compra (nom_licencia, autor, categoría, sitio, versión).
Descripción:	Permite solicitar la compra de una licencia no existente en el catalogo.
Nombre:	Solicitar_actualización (id_licencia)
Descripcion:	Permite solicitar la actualización de una licencia ya adquirida con anterioriad.

Tabla 3.3 Descripción de las clases del diseño: SP_Solicitar_licencia.

Nombre: SP_Gestionar_licencia	
Tipo de clase: Controladora	
Por cada responsabilidad:	
Nombre:	Adicionar_licencia (id_licencia, categoría, sitio, nombre, autor, versión, fecha, descripción).
Descripción:	Permite adicionar una licencia al catálogo.
Nombre:	Actualizar_licencia (nombre, autor, categoría, sitio, versión, descripción, fecha).
Descripción:	Permite actualizar los datos de una licencia.
Nombre:	Eliminar_licencia (id_licencia)
Descripcion:	Permite eliminar una licencia existente en el catalogo.

Tabla 3.4 Descripción de las clases del diseño: SP_Gestionar_licencia.

Nombre: SP_Buscar_licencia	
Tipo de clase: Controladora	
Por cada responsabilidad:	
Nombre:	Búsqueda_general (palabra_clave).
Descripción:	Permite hacer una búsqueda general de licencias.
Nombre:	Búsqueda_avanzada (id_licencia, categoría, autor, nombre).
Descripción:	Permite buscar indicando parámetros específicos.

Tabla 3.5 Descripción de las clases del diseño: SP_Buscar_licencia.

Nombre: SP_Gestionar_reportes	
Tipo de clase: Controladora	
Por cada responsabilidad:	
Nombre:	Mostrar_usuarios()
Descripción:	Permite mostrar la lista de usuarios online del sistema.
Nombre:	Ranking()
Descripción:	Permite saber las licencias mas solicitadas.

Tabla 3.6 Descripción de las clases del diseño: SP_Gestionar_reportes.

3.3 Diseño de la Base de Datos.

En este epígrafe se representa el diseño de la base de datos a través del diagrama de clases persistentes y el diagrama entidad relación.

3.3.1 Diagrama de clases persistentes de la base de datos.

Diagrama de clases persistentes de la base de datos.

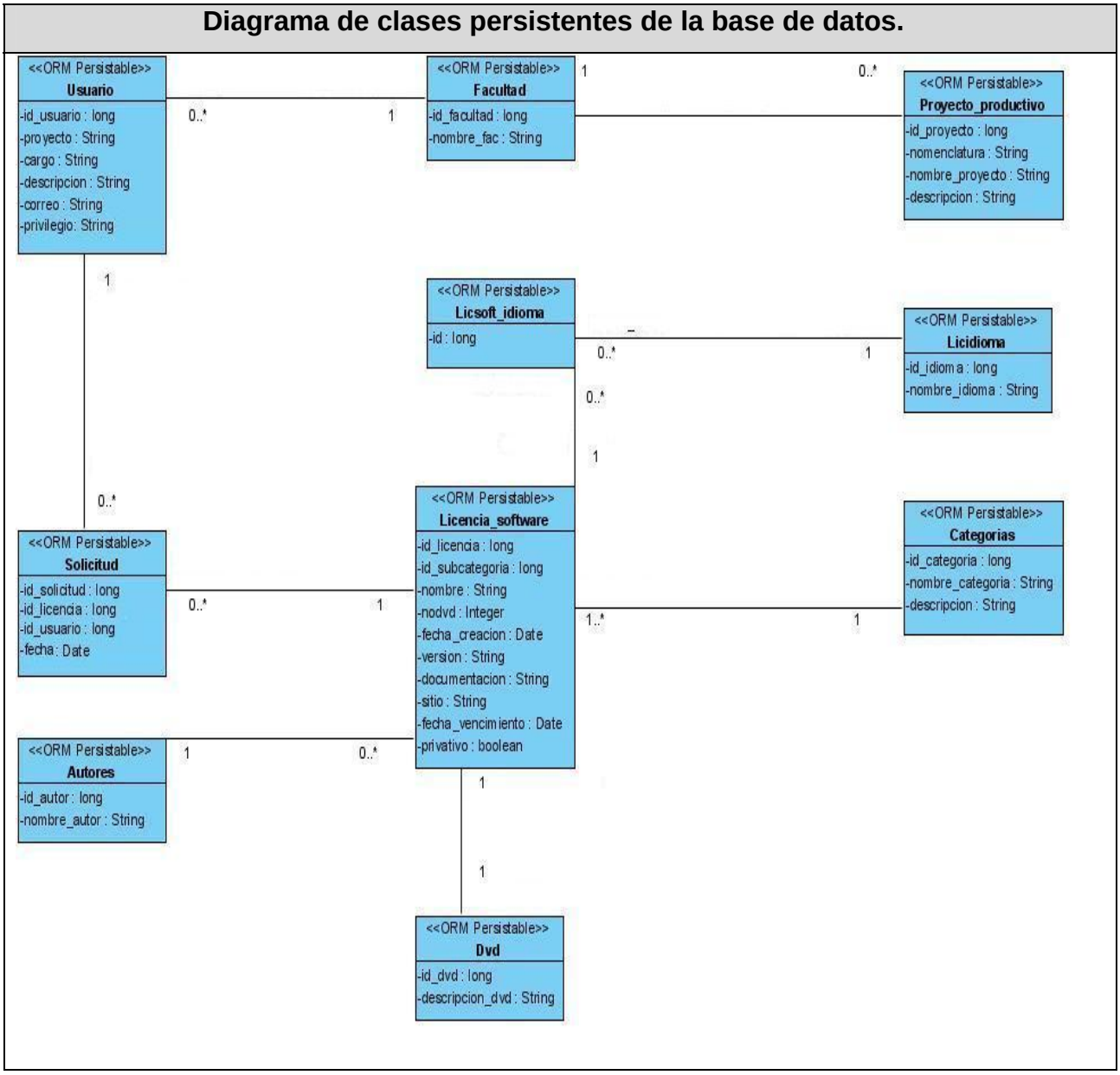


Figura 3.27 Diagrama de clases persistentes de la base de datos.

3.3.2 Diagrama entidad de relación.

Diagrama entidad de relación.

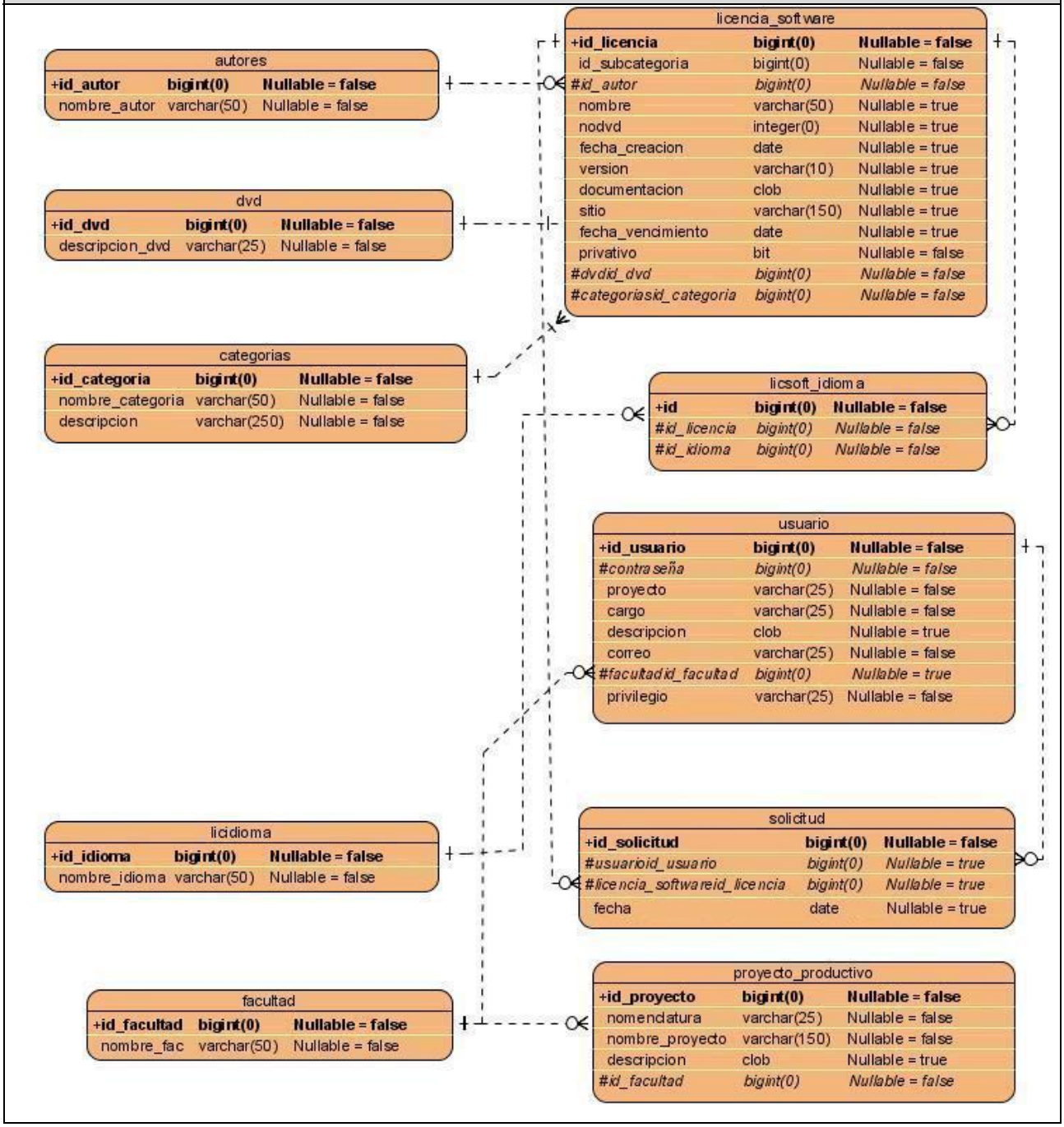


Figura 3.28 Diagrama entidad de relación.

3.4 Descripción de las tablas de la Base de Datos.

Nombre: usuario		
Descripción: Esta tabla contiene los datos de los usuarios		
Atributo	**Tipo**	**Descripción**
id_usuario	int	Identificador de usuario.
contraseña	varchar	Clave para acceder al sistema.
proyecto	varchar	Nombre del proyecto al que pertenece.
cargo	varchar	Rol que desempeña en su proyecto.
descripcion	clob	Breve descripción referente al usuario.
correo	varchar	Correo correspondiente al usuario en el centro.
id_facultad	varchar	Identificador de la facultad a que le corresponde.
pribilegio	varchar	Permisos que le corresponde dentro de la aplicación.

Tabla 3.7 Descripción de la tabla de la base de datos: Usuario

Nombre: licencia_software		
Descripción: Esta tabla contiene los datos de las licencias		
Atributo	**Tipo**	**Descripción**
id_licencia	int	Identificador licencia.
id_categoria	int	Identificador de la categoría a la cual pertenece.
id_autor	Int	Identificador del autor de la licencia.
nombre	varchar	Nombre de la licencia.
nodvd	int	Numero del DVD en el cual se encuentra la licencia.
id_dvd	int	Identificador del DVD que contiene la licencia.

fecha_creacion	date	Fecha en la cual fue creada la licencia.
version	varchar	Es la versión en la cual fue adquirida la licencia.
documentacion	clob	Es la documentación asociada a la licencia.
sitio	varchar	Es la URL de la cual fue tomada la licencia.
fecha_vencimiento	date	Fecha en la cual la licencia deja de ser valida.
privativo	bit	Es para saber bajo que concepto de privacidad esta la licencia.

Tabla 3.8 Descripción de la tabla de la base de datos: licencia_software

Nombre: solicitud		
Descripción: Esta tabla contiene las solicitudes hechas por los solicitantes.		
Atributo	**Tipo**	**Descripción**
id_solicitud	int	Identificador de la solicitud.
id_usuario	Int	Identificador del solicitante.
id_licencia	Int	Identificador de la licencia solicitada.
fecha	date	Fecha en la que fue realizada la solicitud.

Tabla 3.9 Descripción de la tabla de la base de datos: solicitud

Nombre: autores		
Descripción: Esta tabla contiene los datos de los autores de licencias.		
Atributo	**Tipo**	**Descripción**
id_autor	int	Identificador del autor.
nombre	varchar	Nombre del autor.

Tabla 3.10 Descripción de la tabla de la base de datos: autores

Nombre: dvd		
Descripción: Esta tabla contiene los datos de los DVD en los que se almacenan las licencias.		
Atributo	**Tipo**	**Descripción**
id_dvd	int	Identificador del DVD.
descripcion_dvd	varchar	Breve descripción del contenido del DVD.

Tabla 3.11 Descripción de la tabla de la base de datos: dvd

Nombre: categorias		
Descripción: Esta tabla contiene los datos relacionados con las categorías a las que puede pertenecer una licencia.		
Atributo	**Tipo**	**Descripción**
id_categoria	int	Identificador de la categoría.
nombre_categoria	varchar	Nombre completo de la categoría.
descripcion	varchar	Breve descripción de la categoría.

Tabla 3.12 Descripción de la tabla de la base de datos: categoria

Nombre: licidioma		
Descripción: Esta tabla contiene los datos del idioma en que puede estar una licencia.		
Atributo	**Tipo**	**Descripción**
id_idioma	int	Identificador de un idioma en específico.
nombre_idioma	varchar	Nombre completo del idioma.

Tabla 3.13 Descripción de la tabla de la base de datos: licidioma

Nombre: licsoft_idioma		
Descripción: Esta tabla contiene los datos del proyecto a cual pertenece el solicitante.		
Atributo	**Tipo**	**Descripción**
id	int	Identificador del idioma de la licencia de

		software en particular.
id_licencia	int	Identificador de la licencia que esta en ese idioma.
id_idioma	int	Identificador del idioma.

Tabla 3.14 Descripción de la tabla de la base de datos: licsoft_idioma

Nombre: proyecto_productivo		
Descripción: Esta tabla contiene los datos del proyecto a cual pertenece el solicitante.		
Atributo	**Tipo**	**Descripción**
id_proyecto	int	Identificador del proyecto.
nomenclatura	varchar	Código que recibe cada proyecto en particular.
nombre_proyecto	varchar	Nombre del proyecto.
descripcion	clob	Breve descripción del proyecto.
id_facultad	int	Identificador de la facultad a la que pertenece el proyecto.

Tabla 3.15 Descripción de la tabla de la base de datos: proyecto_productivo

Nombre: facultad		
Descripción: Esta tabla contiene los datos de las facultades.		
Atributo	**Tipo**	**Descripción**
id_facultad	int	Identificador de la facultad.
nombre_fac	varchar	Nombre de la facultad.

Tabla 3.16 Descripción de la tabla de la base de datos: facultad

3.5 Principios del diseño.

3.5.1 Interfaz de usuario.

En el sistema propuesto todo el diseño está determinado fundamentalmente por el principio de la usabilidad, teniendo en cuenta que no se trata de un sitio Web, sino de una aplicación de trabajo donde el diseño tiene como principal propósito facilitar su uso, comprensión y navegación, por encima de ornamentos inútiles, aunque manteniendo pautas estéticas, orgánicas y agradables.

El interés general es mantener el diseño y la estructura del sitio lo más simple posible, la simplicidad es entendimiento del contenido, es facilidad para encontrar lo que se busca, es también velocidad de descarga, por eso se recomienda que las página no estén muy cargadas de imágenes.

3.5.2 Tratamiento de errores.

Una excepción es lanzada durante la ejecución del programa interrumpiendo el flujo normal de las sentencias. Permite de forma clara controlar los errores que ocurren en el momento de ejecución.

Cada formulario se encarga de la validación de sus datos para evitar errores de concepto. Y se utilizan mensajes de confirmación, para acciones que son irreversibles como es el caso de las eliminaciones.

3.6 Conclusiones

En este capítulo se ha mostrado los diagramas de clases tanto del análisis como del diseño, se hiso la descripción de las clases y demás elementos necesarios para la implementación. Se definieron, cuáles son las clases que serán persistentes, luego, a partir de esto, se construyó el modelo de datos. Se expusieron las pautas ha seguir para el diseño de la interfaz, y se explicó cómo ha de llevarse a cabo el tratamiento de errores.

Conclusiones

Una vez concluido satisfactoriamente este trabajo y después de haber cumplido con los objetivos propuestos, se puede percibir los siguientes resultados:

- Se describieron de forma clara y concisa los procesos que serán implementados.
- Se hizo una modelación conceptual de las clases implicadas en la aplicación.
- Se desarrollaron los diagramas que describen el diseño web propuesto para la aplicación y la descripción de las clases del diseño.
- Se propuso un sistema que resolverá los problemas relacionados con la solicitud de licencias existentes hasta el momento en la Universidad de las Ciencias Informáticas.
- Con la solución que se propone se tendrá un mayor control sobre las licencias que sean solicitadas por los proyectos productivos de la universidad.

.

Recomendaciones

Con vista al desarrollo futuro de este proyecto se listan a continuación una serie de recomendaciones:

- Llevar a cabo la implementación de la propuesta hecha en este trabajo.
- Seguir investigando con el objetivo de profundizar aún más en el tema y lograr la automatización de los procesos que secundan los descritos en esta investigación.
- Proponer nuevos procedimientos que agilicen los procesos de solicitud de licencias en la Universidad de las Ciencias Informáticas.

Bibliografía Consultada

Anónimo. ¿Qué es un lenguaje de programación? [En línea] 2006. http://www.alegsaonline.com/art/11.php.

Anónimo. ASP Facil. [En línea] 2004. http://www.aspfacil.com/articulos/278001.asp.

Anónimo. Comparación entre SGBD. [En línea] 2005. http://www.ilustrados.com/documentos/sgbd.pdf.

Anónimo. desarrolloweb.com. [En línea] 2001. http://www.desarrolloweb.com/articulos/541.php.

Anónimo. Eclipse. [En línea] 2008. http://www.eclipse.org/mylyn/

Anónimo. Eclipse. [En línea] 2008. http://www.eclipse.org/pdt/.

Anónimo. El blog de INEW. [En línea] 2006. http://elblogdeinwe.com/blogviejo/raiz/index.php?op=ver_noticia&noticia=56.

Anónimo. Gamarod. [En línea] 2004. http://www.gamarod.com.ar/articulos/introduccion_a_aspnet.asp.

Anónimo. Milenium. [En línea] 2008. http://www.informaticamilenium.com.mx/Paginas/espanol/sitioweb.htm.

Anónimo. Softonic. [En línea] 2008. http://quanta-plus.softonic.com/linux.

Anónimo. Wikipedia. [En línea] 2008. http://es.wikipedia.org/wiki/Eclipse_(software).

Anónimo. Wikipedia. [En línea] 2008. http://es.wikipedia.org/wiki/Framework.

Anónimo. Zend Studio. [En línea] 2003. http://www.desarrolloweb.com/articulos/1178.php.

Arambillete, Wilman. Introducción a las Java Server Pages. [En línea] 2002. http://www.webexperto.com/articulos/art/57/introduccion-a-las-java-server-pages-/.

Daniel, Pecos. PostGreSQL vs. MySQL. [En línea] http://www.netpecos.org/docs/mysql_postgres/index.html.

Hinostroza, Raul Rodas. LinuxCentro.Net. [En línea] 2007. http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP.

Ivar Jacobson, Grady Booch y James Rumbaugh. Biblioteca.uci.cu. [En línea] 1999. http://bibliodoc.uci.cu/pdf/reg03050.pdf.

Labrador, Ramón M. Gómez. Tipos de Licencias de Software. [En línea] 2005. http://www.informatica.us.es/~ramon/articulos/LicenciasSoftware.pdf.

López, Alejandro Rivera. Universidad de las Americas. [En línea] 2008. http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/rivera_l_a/capitulo2.pdf.

Murray, Pablo. Gestión-Información-Conocimiento. [En línea] 2003. http://eprints.rclis.org/archive/00002316/01/B14-01.pdf.

Patricio Letelier, Mª Carmen Penadés. Métodologías ágiles para el desarrollo de software:eXtreme Programming (XP). [En línea] http://www.willydev.net/descargas/masyxp.pdf.

Peralta, Manuel. Sistema de Información. [En línea] 1997. http://www.monografias.com/trabajos7/sisinf/sisinf.shtml#esi.

Rosas, Juan Eladio Sánchez. Tuxpuc. [En línea] 2007. http://tuxpuc.pucp.edu.pe/content/view/744/1/.

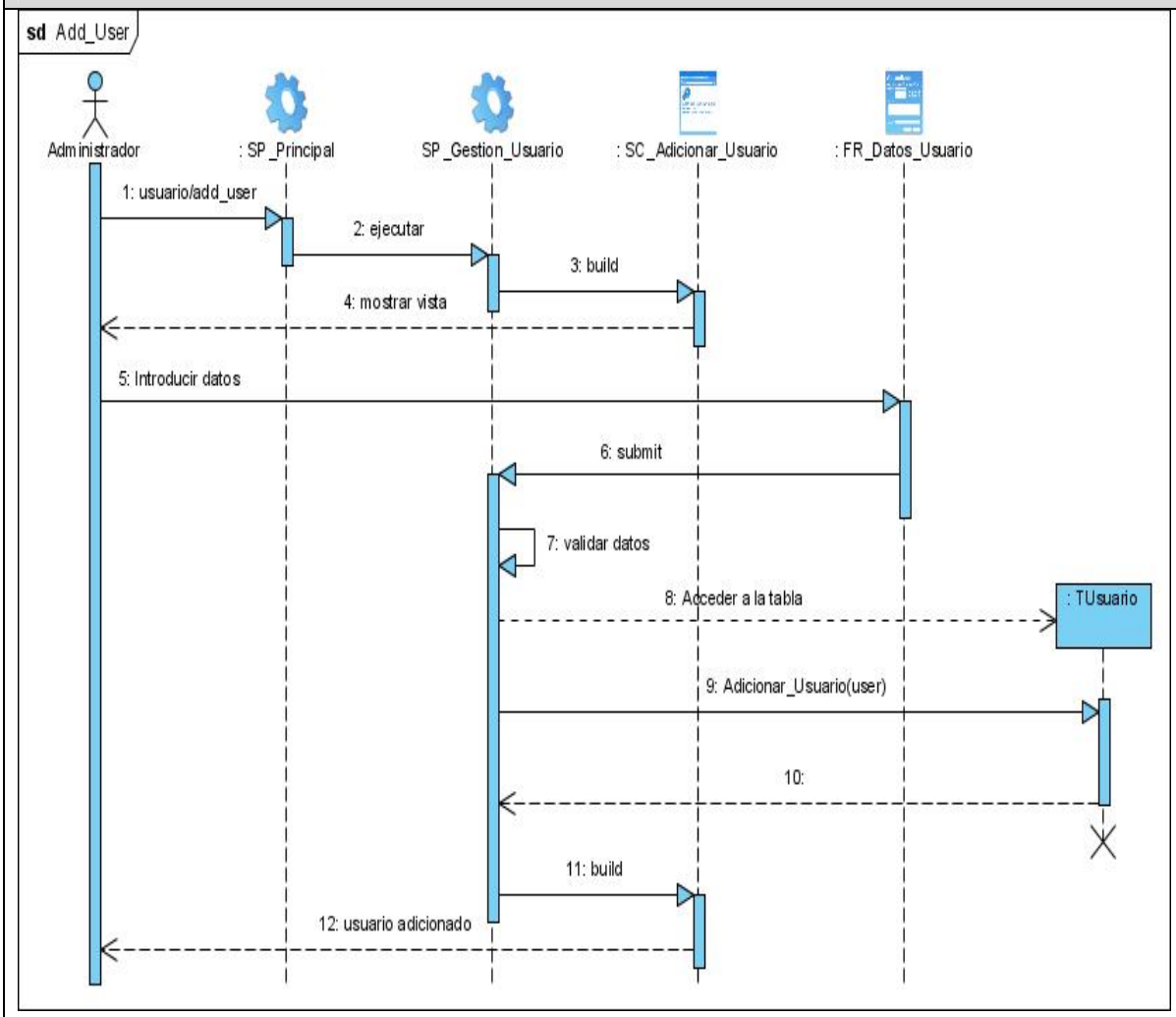
Valdés, Damián Pérez. ¿Qué es Javascript? [En línea] 2007. http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/.

Zaninotto, François. Symfony, la guía definitiva. [En línea] http://www.librosweb.es/symfony/index.html.

Anexos

ANEXO I

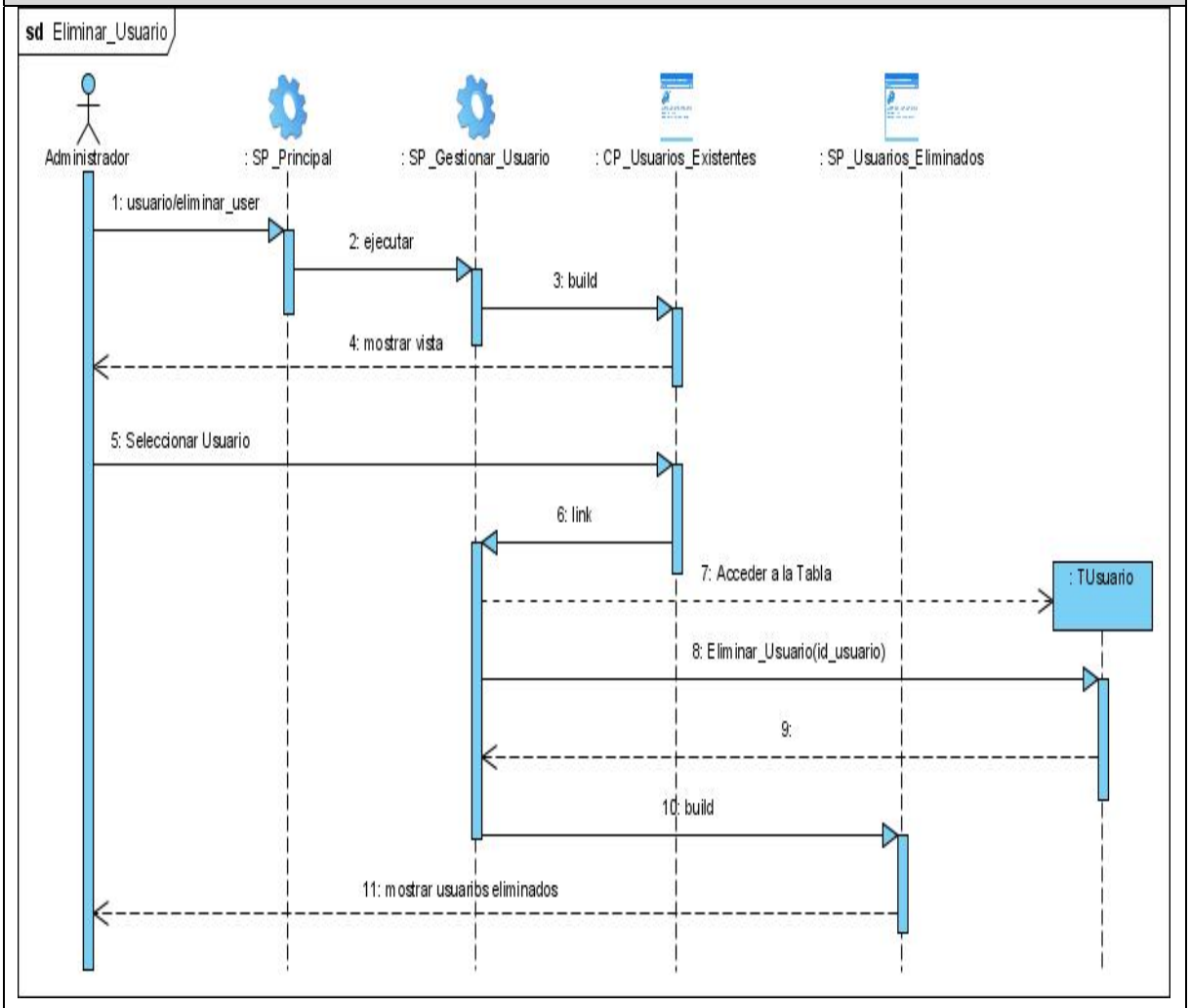
Diagrama de Secuencia: Adicionar Usuario



CU Gestionar Usuario: Sección Adicionar usuario.

ANEXO II

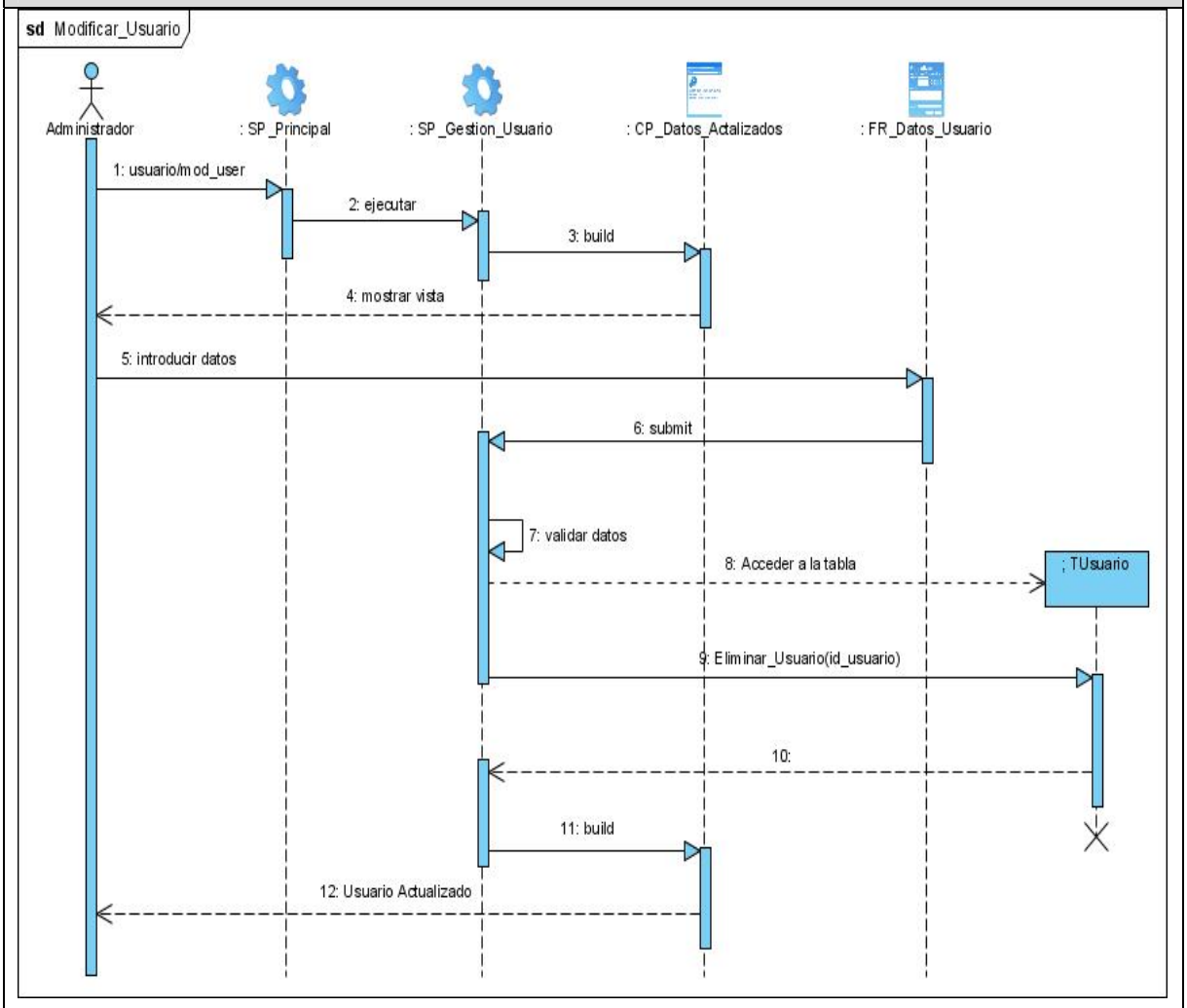
Diagrama de Secuencia: Eliminar Usuario



CU Gestionar Usuario: Sección Eliminar usuario.

ANEXO III

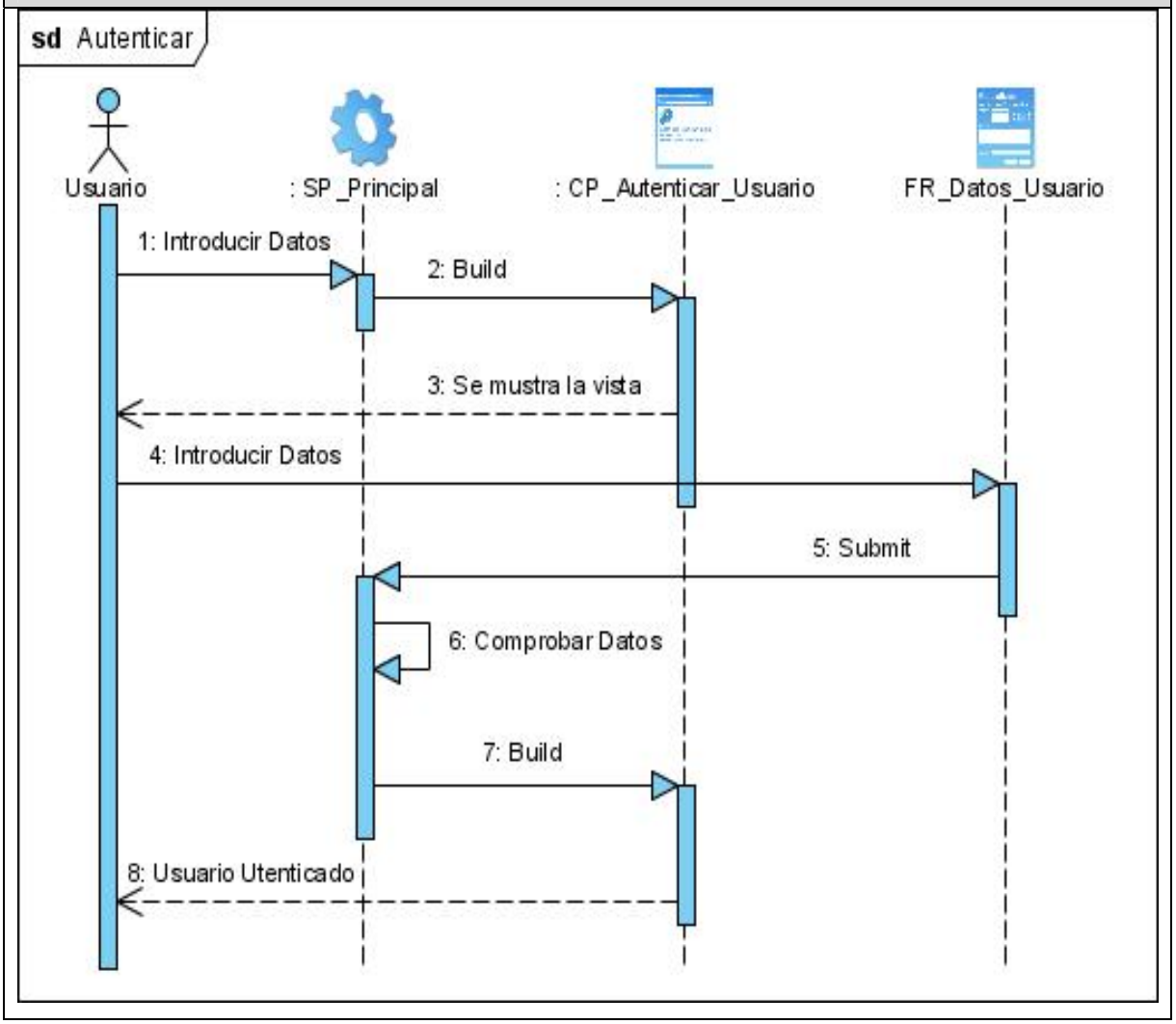
Diagrama de Secuencia: Modificar Usuario



CU Gestionar Usuario: Sección Modificar usuario.

ANEXO IV

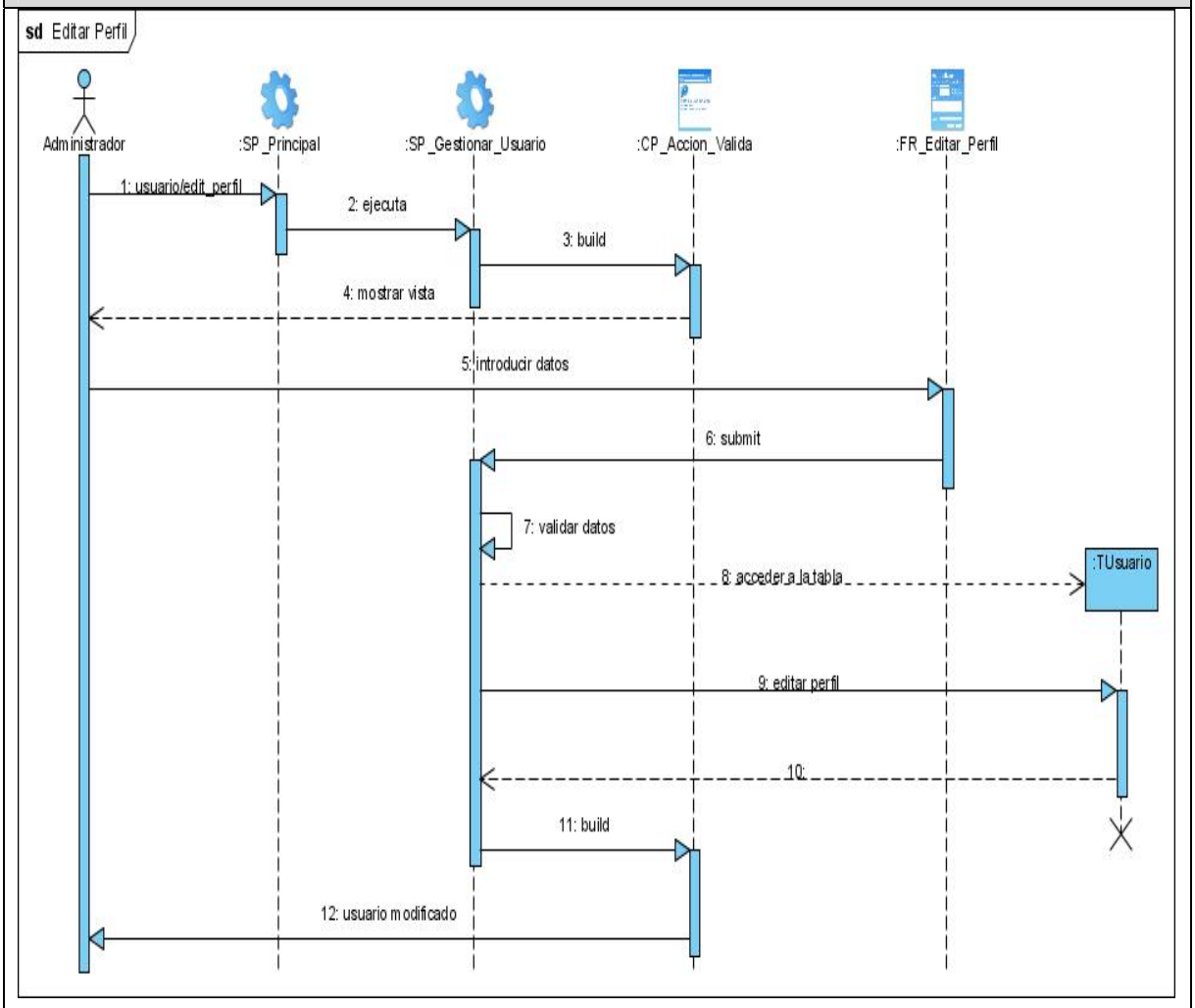
Diagrama de Secuencia: Autenticar Usuario



CU Autenticar Usuario.

ANEXO V

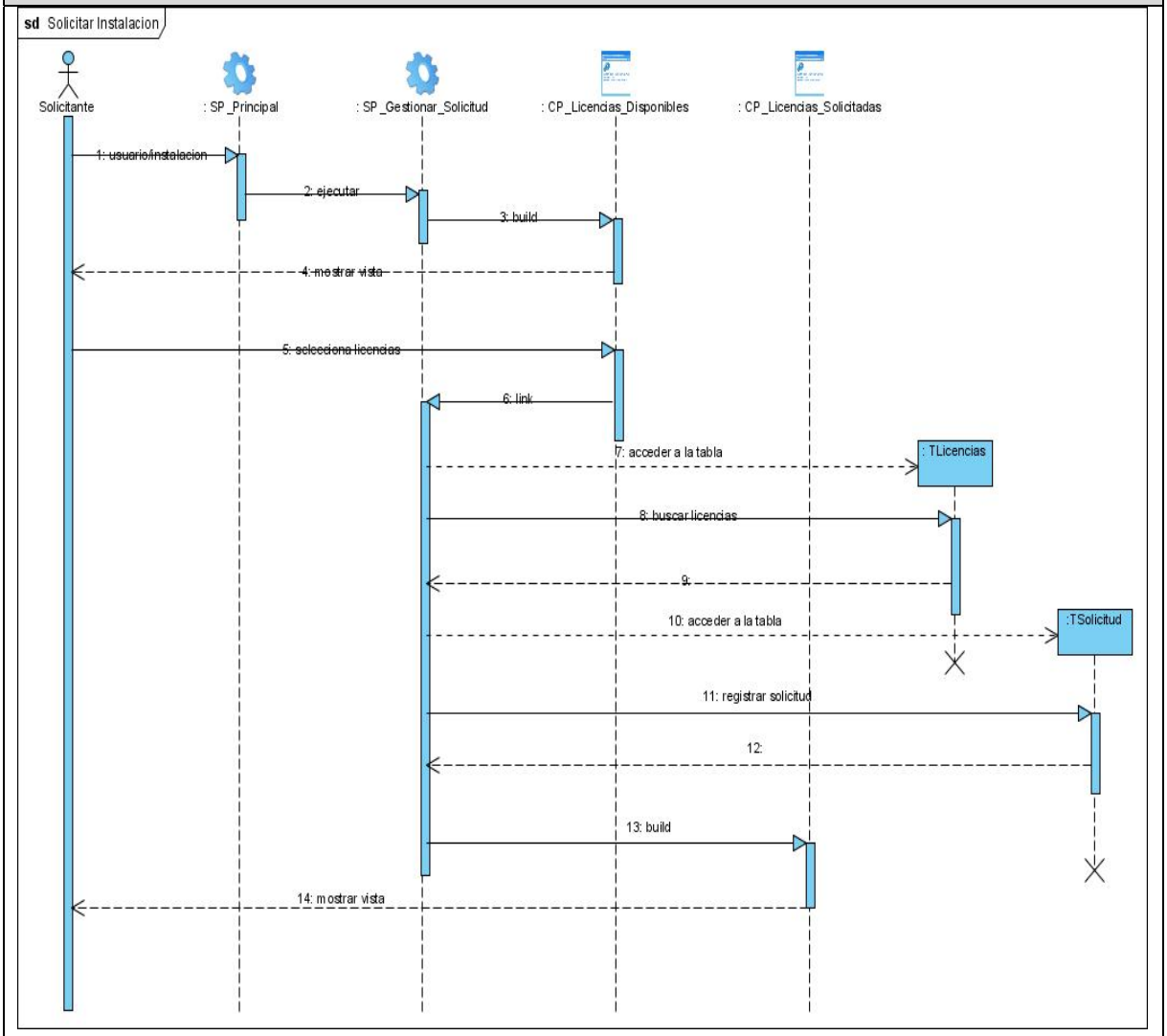
Diagrama de Secuencia: Editar Perfil



CU Editar Perfil

ANEXO VI

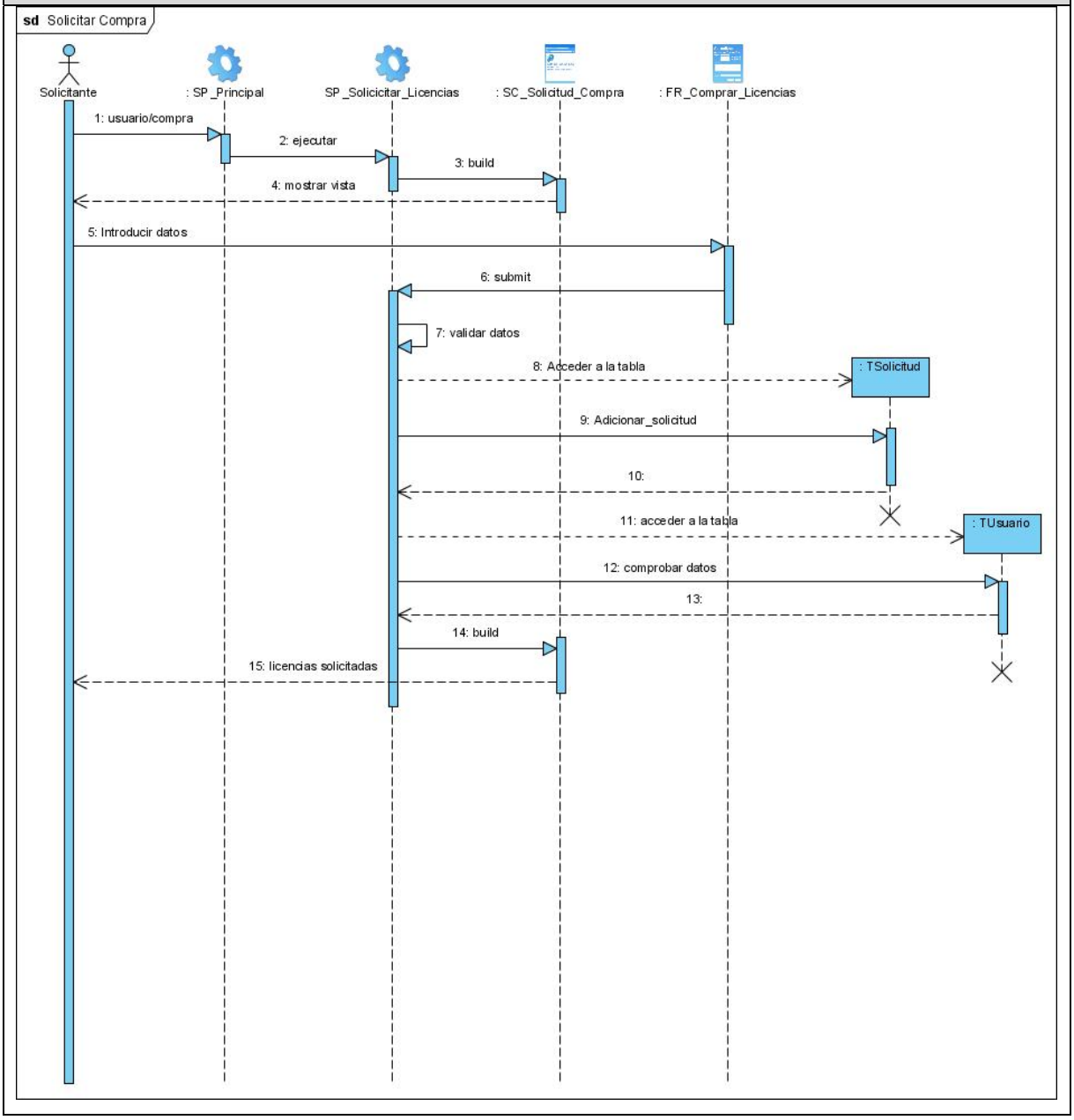
Diagrama de Secuencia: Solicitar Instalación o préstamo



CU Solicitar licencia: Sección Solicitar préstamo o instalación.

ANEXO VII

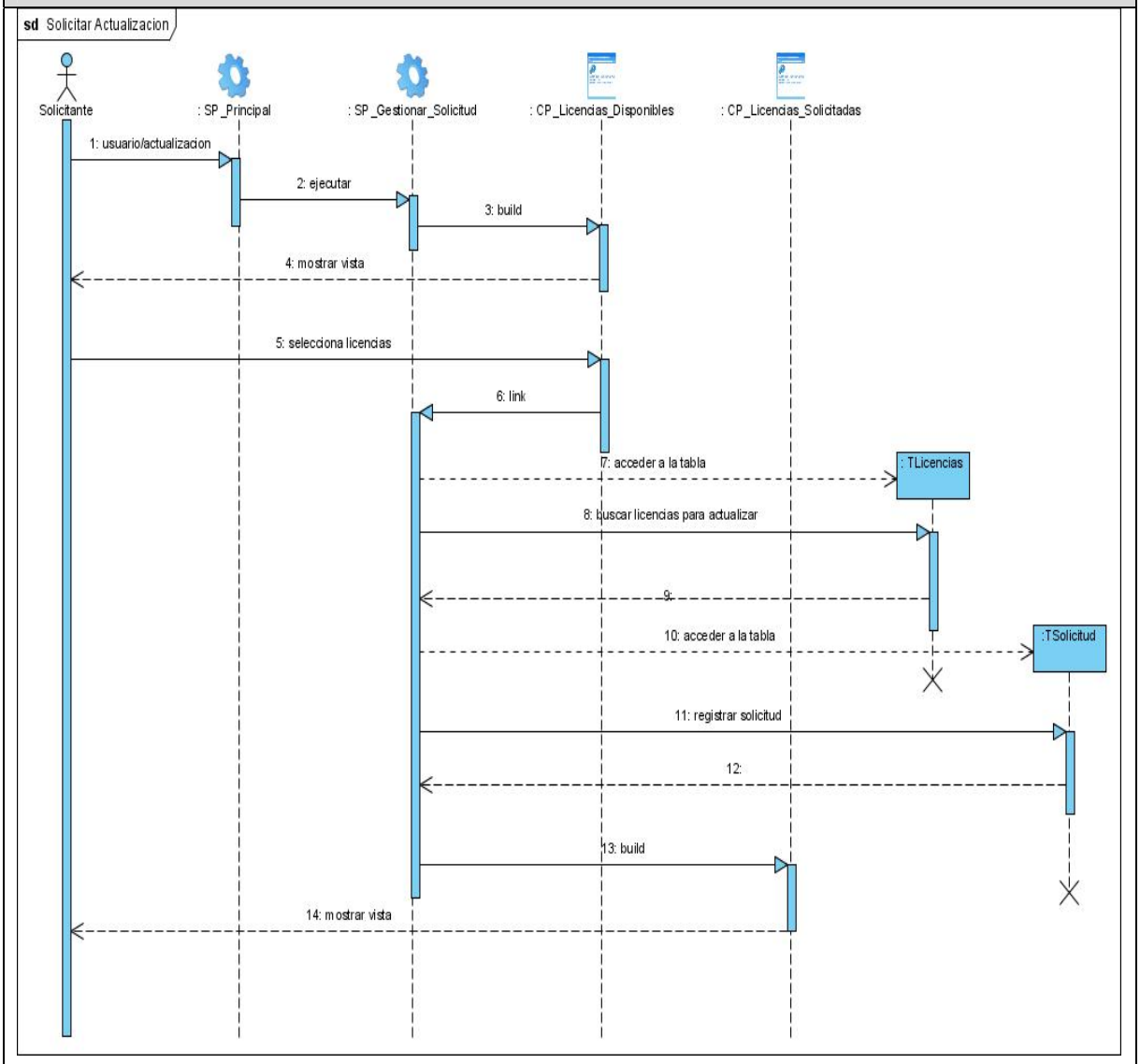
Diagrama de Secuencia: Solicitar compra



CU Solicitar licencia: Sección Solicitar compra.

ANEXO VIII

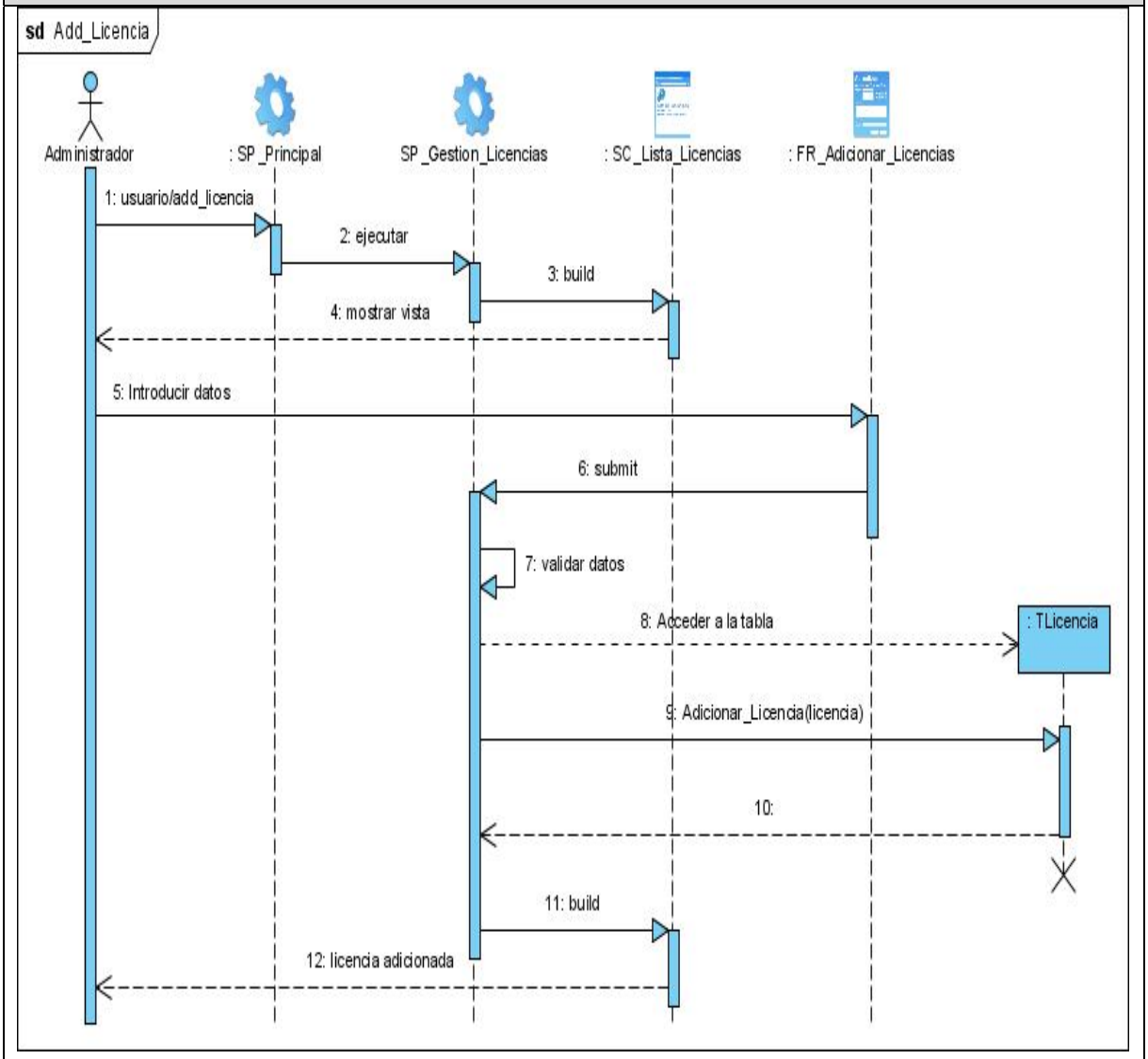
Diagrama de Secuencia: Solicitar actualización



CU Solicitar licencia: Sección Solicitar actualización.

ANEXO IX

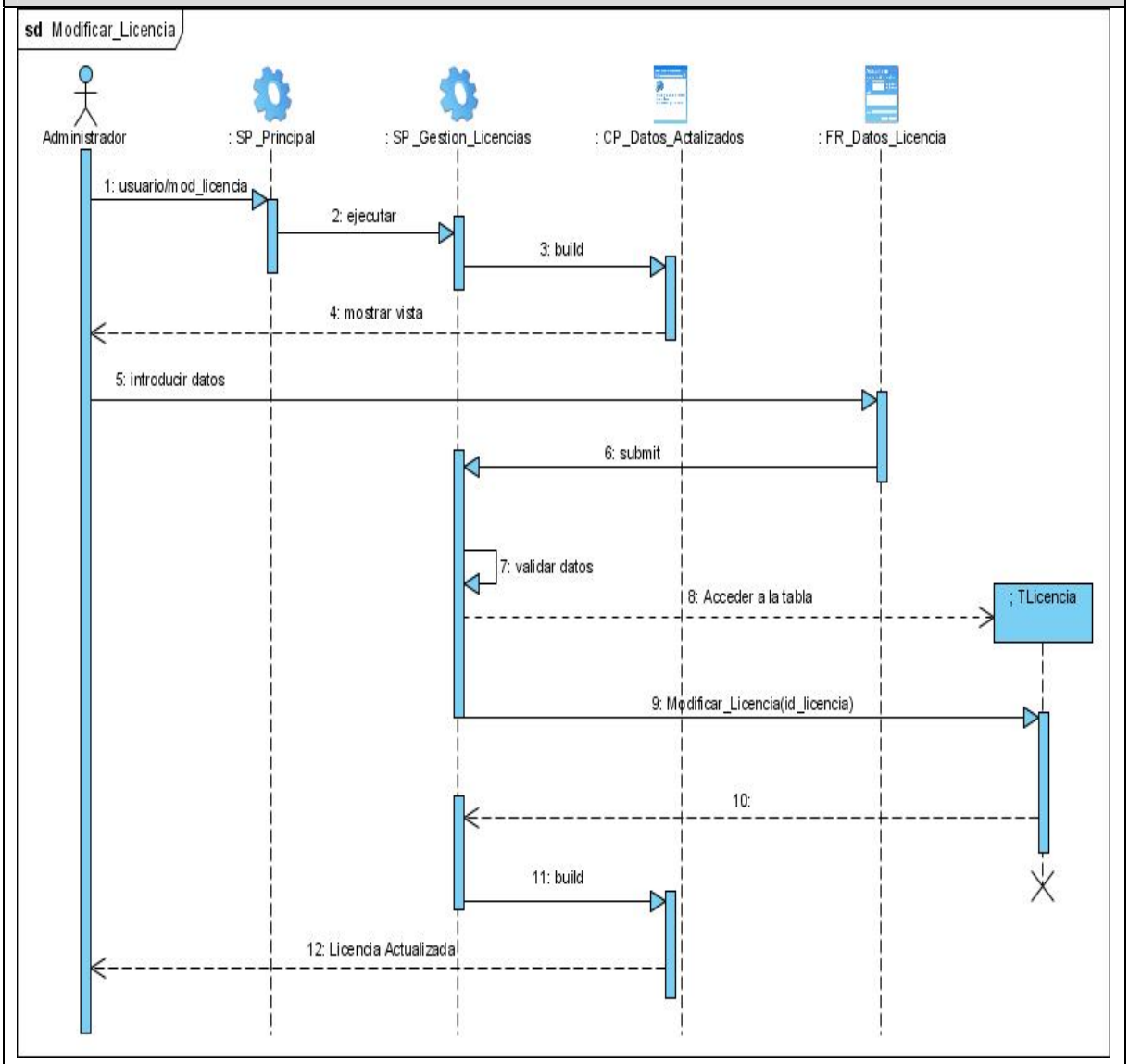
Diagrama de Secuencia: Adicionar licencia



CU Gestionar Licencia: Sección Adicionar licencia.

ANEXO X

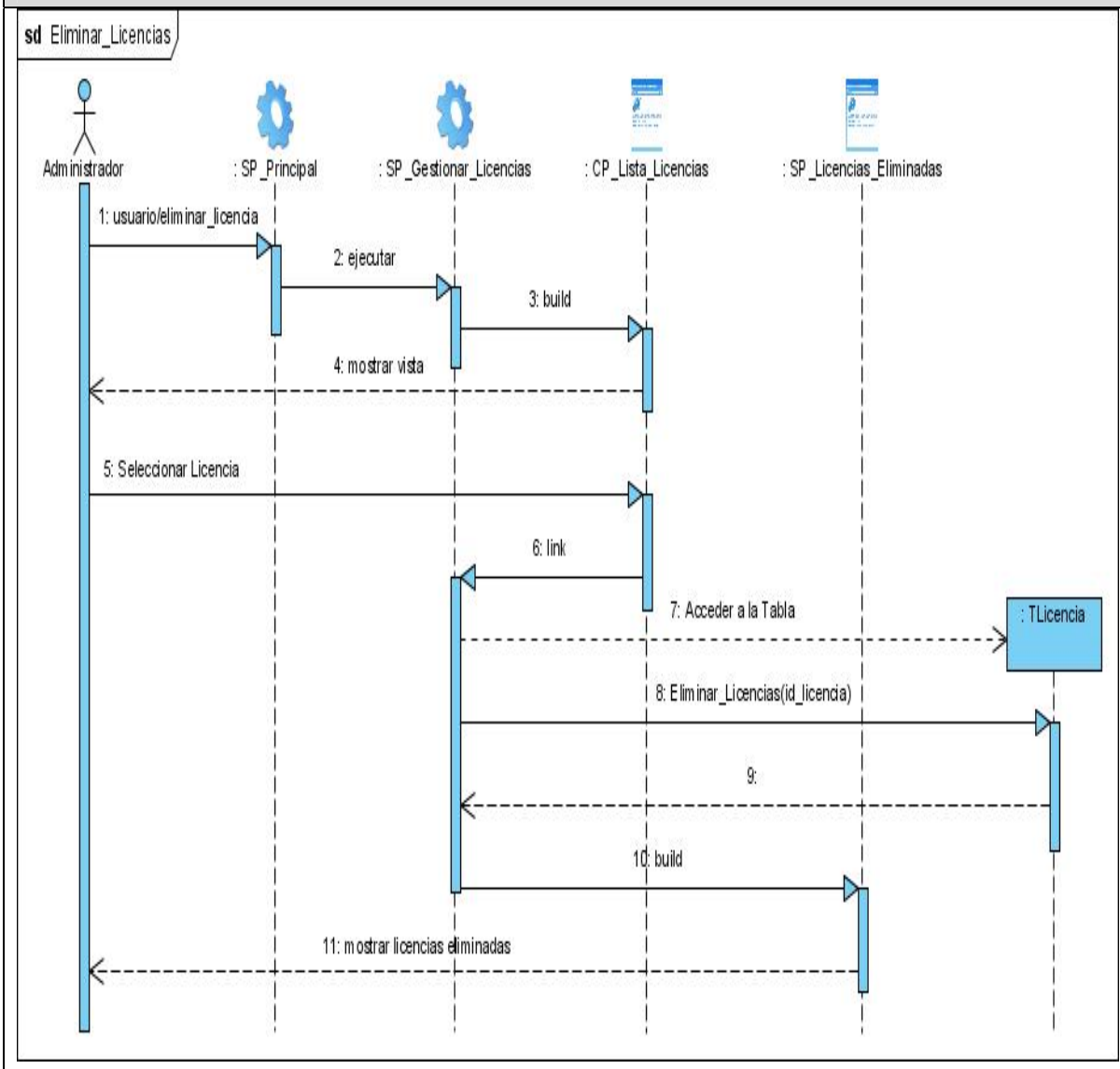
Diagrama de Secuencia: Actualizar licencia



CU Gestionar Licencia: Sección Actualizar licencia.

ANEXO XI

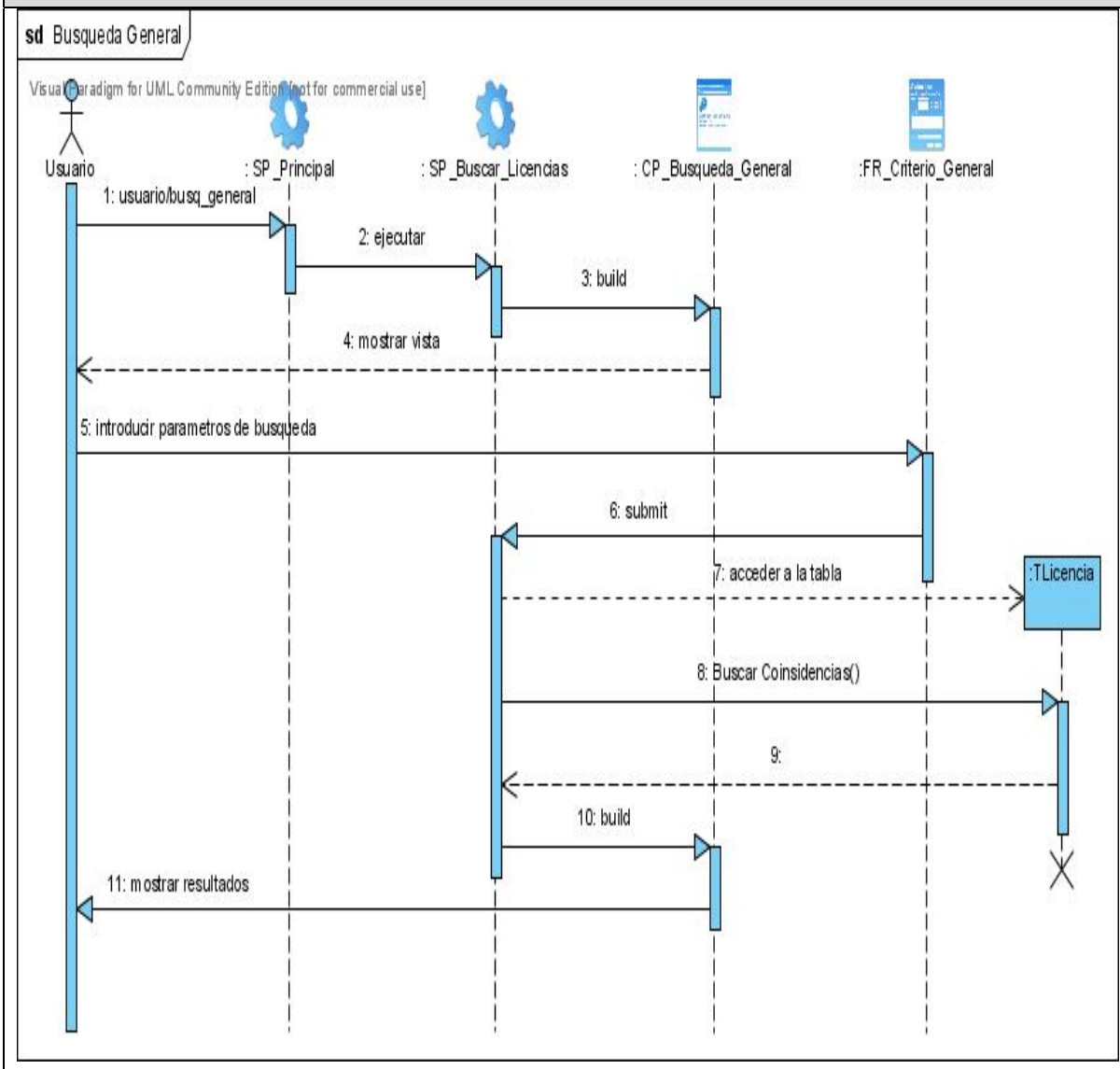
Diagrama de Secuencia: Eliminar licencia



CU Gestionar Licencia: Sección Eliminar licencia.

ANEXO XII

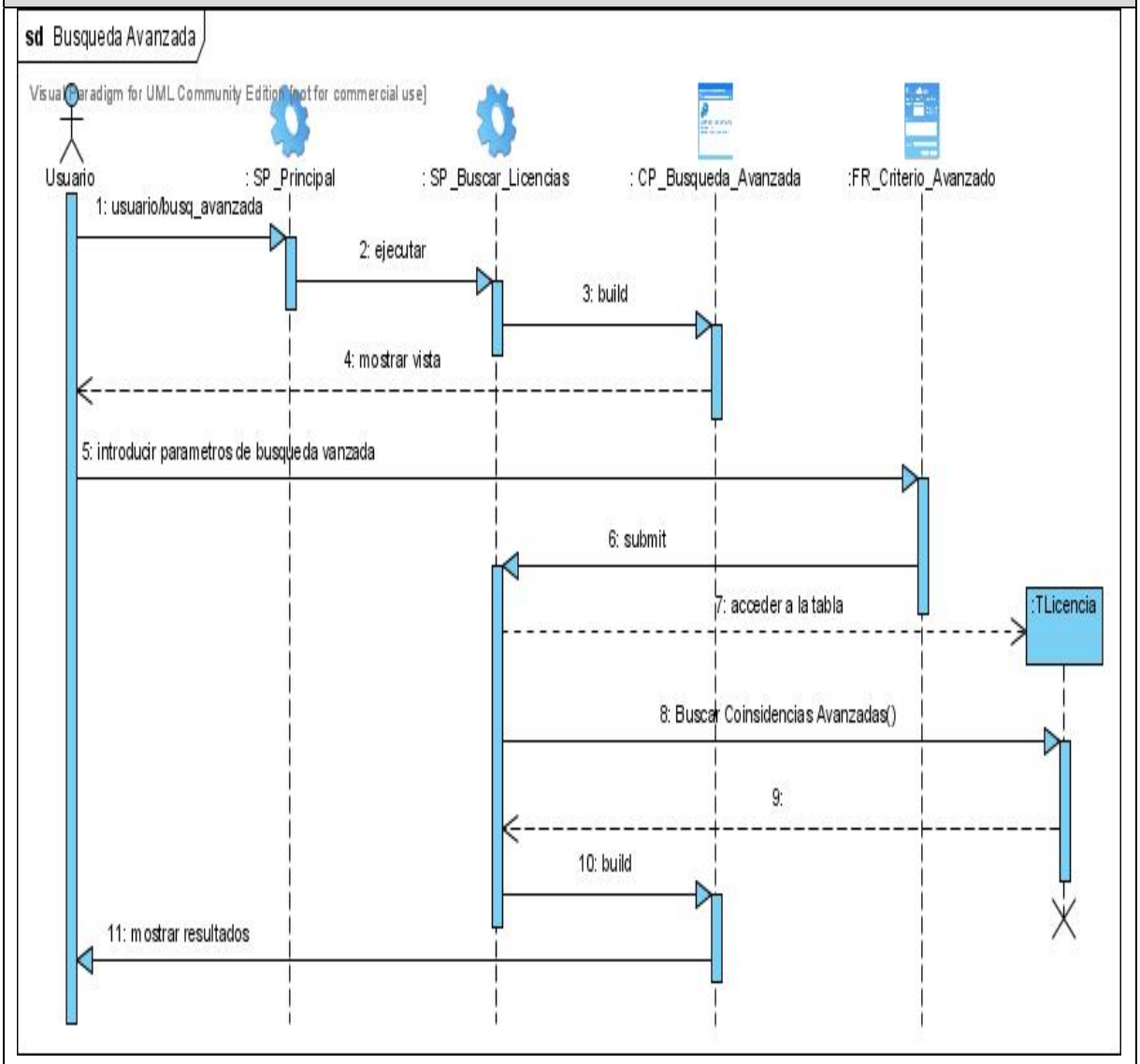
Diagrama de Secuencia: Búsqueda general



CU Buscar Licencia: Sección Búsqueda General.

ANEXO XIII

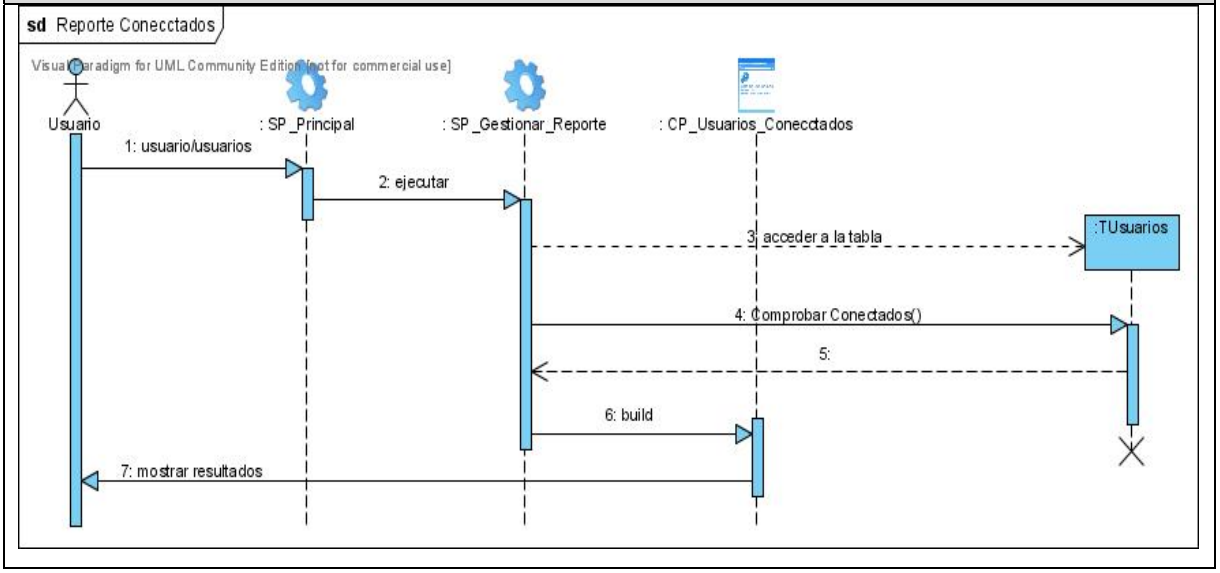
Diagrama de Secuencia: Búsqueda avanzada



CU Buscar Licencia: Sección Búsqueda Avanzada.

ANEXO XIV

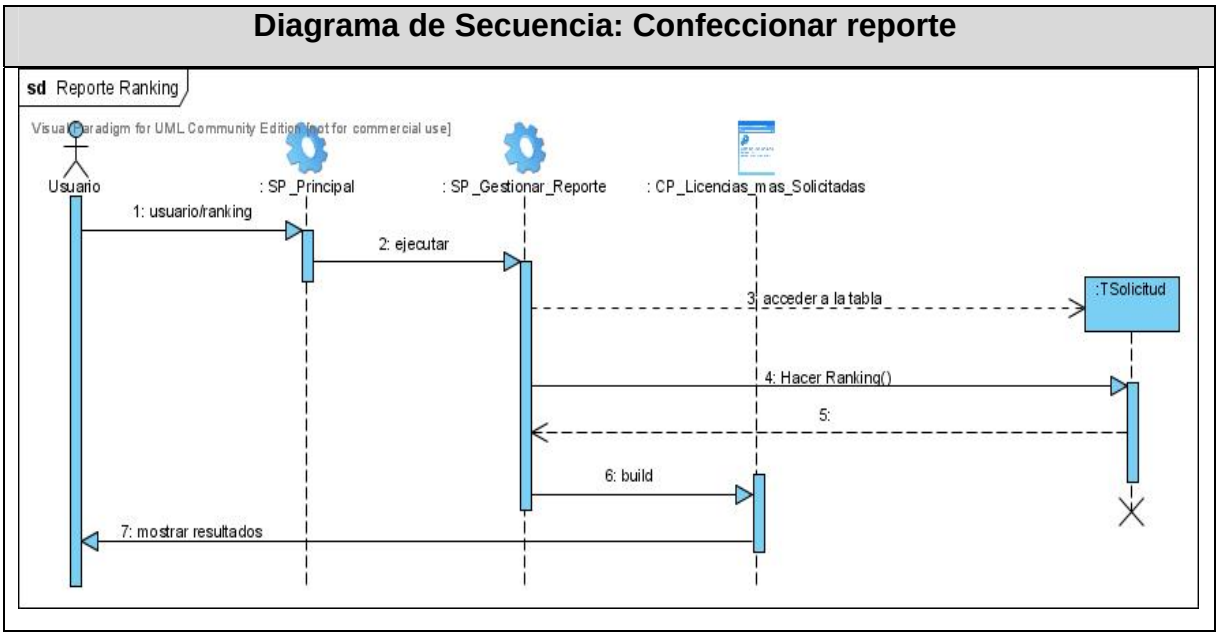
Diagrama de Secuencia: Confeccionar reporte



CU Confeccionar Reporte: Sección Mostrar Usuarios Conectados.

ANEXO XV

Diagrama de Secuencia: Confeccionar reporte



CU Confeccionar Reporte: Sección Mostrar Licencias más Descargadas.

Glosario de Términos

API- Interfaz de Programación de Aplicaciones (en inglés Application Programming Interface) es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

CGI- Interfaz de entrada común (en ingles Common Gateway Interface), es una importante tecnología que permite a un cliente (explorador web) solicitar datos de un programa ejecutado en un servidor web.

FTP- Protocolo de Transferencia de Archivos (en ingles File Transfer Protocol) es un sistema que permite enviar y recibir ficheros entre computadores a través de la red.

GUI- Interfaz gráfica de usuario (en ingles Graphical User Interface), es el artefacto tecnológico de un sistema interactivo que posibilita, a través del uso y la representación del lenguaje visual, una interacción amigable con un sistema informático.

IDE- Entorno de desarrollo integrado (en ingles Integrated Development Environment), es un programa compuesto por un conjunto de herramientas para un programador.

JUnit- Es un conjunto de librerías creadas por Erich Gamma y Kent Beck que son utilizadas en programación para hacer pruebas unitarias de aplicaciones Java.

Plugin- componente enchufable, es una aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica.

Refactorización- Es una técnica de la ingeniería de software para reestructurar un código fuente, alterando su estructura interna sin cambiar su comportamiento externo.

SQL- Lenguaje de Consulta Estructurado (en ingles Structured Query Language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas.

UML- Lenguaje Unificado de Modelado (en inglés, Unified Modelling Language) es el lenguaje de modelado de sistemas de software más conocido en la actualidad.

URL- Localizador de Recurso Uniforme (en inglés Uniform Resource Locator), la dirección global de documentos y de otros recursos en la World Wide Web.

XML- Lenguaje de marcas extensible (en ingles Extensible Markup Language) una especificación/lenguaje de programación desarrollada por la World Wide Web**.**