



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 2

*Sistema Automatizado para la Gestión de las
Solicitudes de Servicios Telemáticos*

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

AUTORES

Iris Alemán Betancourt
Juan Carlos Rodríguez Gutiérrez

TUTOR

Ing. Eduard Palomo Gené

Ciudad de la Habana
Junio del 2007

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 _____.

Iris Alemán Betancourt

Juan Carlos Rodríguez Gutiérrez

Ing. Eduard Palomo Gené

Firma del Autor

Firma del Autor

Firma del Tutor

"Si el presente es de lucha, el futuro es nuestro."

Ernesto Che Guevara.

Agradecimientos

Agradecemos a todas aquellas personas que nos han acompañado a lo largo de nuestras vidas y que de una forma u otra han contribuido a nuestra formación profesional y a la realización de este trabajo. En especial a Eduardo Estévez y Yalian Rodríguez, por sus imprescindibles aportes, que han hecho posible la consumación de este sueño.

A todos, gracias, ustedes forman parte de este logro.

Dedicatoria

A mis padres, ejemplos de dedicación y abnegación, guías ineludibles y motivos de orgullo, admiración e inspiración siempre.

A mi hermana, personita que quiero mucho y siempre llevo en mi corazón a pesar de estar la mayor parte del tiempo alejadas.

A Yalian, compañero de la vida y amigo incondicional.

A toda mi familia, fuente de apoyo y amor inagotable ante las disímiles situaciones de la vida.

Iris.

A mis padres, quienes confiaron ciegamente en mí y supieron darme todo el amor y educación del mundo.

A mi hermana y mi cuñado, que me han apoyado incondicionalmente en todo momento.

A Giselle, cuyo amor y comprensión enaltecen mi existencia a diario.

A toda mi familia.

Juan Carlos.

Resumen

La vertiginosa evolución de las tecnologías de la información ha traído consigo el consecuente desarrollo de las redes telemáticas existentes, así como en el crecimiento acelerado de los avances y aplicaciones de los servicios telemáticos.

De igual modo la automatización de la gestión de los servicios telemáticos ha pasado a ser una necesidad para las instituciones u organizaciones que se dedican a la prestación de los mismos, debido a las nuevas necesidades ciudadanas y la expansión global de los servicios.

Este trabajo ha sido realizado en la Universidad de las Ciencias Informáticas, donde se ofrecen diferentes servicios telemáticos a la comunidad universitaria y se carece de un sistema para gestionar las solicitudes de los mismos.

Brindar una solución automatizada, flexible y única a los diferentes problemas presentados en el proceso de gestión de las solicitudes de dichos servicios, específicamente en lo que a solicitud de cuenta de correo electrónico e Internet, acceso remoto, publicación de sitios Web y apertura de puertos se refiere, es el objetivo de este trabajo.

Se propone como solución una aplicación Web, basada en tecnología PHP y con gestor de base de datos MySQL, la cual contribuye al eficiente manejo de las solicitudes, ganando en comodidad, rapidez y confiabilidad a la hora del llenado, búsqueda y verificación de las mismas, en un mayor control por parte de los directivos, así como una mayor agilidad en el cálculo estadístico.

Índice

Introducción	1
Capítulo 1: Fundamentación del Tema	5
1.1 Introducción.....	5
1.2 Servicios Telemáticos.....	5
1.3 Proceso de solicitudes de servicios telemáticos.....	6
1.4 Firma digital.....	7
1.5 Tendencias y tecnologías actuales utilizadas.....	8
1.5.1 Algoritmos de firma digital.....	8
1.5.2 Aplicaciones Web.....	11
1.5.3 Modelo Cliente/Servidor.....	13
1.5.4 Lenguajes de Programación para la Web.....	13
1.5.4.1 PHP (PHP: Hypertext Preprocessor).....	14
1.5.5 Servicios Web (Web Services).....	15
1.5.6 Servidor Web Apache.....	16
1.5.7 Sistema de Gestión de Base de Datos (SGBD).....	18
1.5.7.1 MySQL.....	18
1.5.8 Patrones de arquitectura.....	19
1.5.8.1 Modelo Vista Controlador (MVC).....	20
1.5.9 Proceso de desarrollo de Software.....	22
1.5.9.1 Metodología RUP.....	23
1.5.10 Herramientas utilizadas.....	25
1.5.10.1 Macromedia Dreamweaver MX.....	25
1.5.10.2 Zend Studio.....	25
1.5.10.3 Rational Rose.....	26
1.6 Conclusiones.....	27
Capítulo 2: Características del Sistema.....	28
2.1 Introducción.....	28

2.2 Objeto de Estudio	28
2.2.1 Descripción del proceso actual.	28
2.2.2 Análisis crítico del proceso actual.	30
2.3 Procesos objeto de automatización.	31
2.3.1 Información que se maneja.	32
2.4 Modelo del negocio.....	32
2.4.1 Actores del negocio.....	32
2.4.2 Trabajadores del negocio.....	33
2.4.3 Diagrama de casos de uso del negocio.	33
2.4.4 Descripción de los casos de uso del negocio.	34
2.4.4.1 Especificación textual del Caso de Uso “Solicitar Servicio Telemático”.....	34
2.4.4.2 Diagrama de Actividades del Caso de Uso “Solicitar Servicio Telemático”.	36
2.4.4.3 Diagrama de Clases del Modelo de Objetos del Caso de Uso “Solicitar Servicio”.	37
2.5 Especificación de los requisitos de software.....	38
2.5.1 Requerimientos Funcionales.....	38
2.5.2 Requerimientos no Funcionales.....	40
2.6 Descripción del Sistema propuesto.....	41
2.7 Modelo de casos de usos del sistema.	43
2.7.1 Definición de los actores.	43
2.7.2 Listado de casos de uso.....	44
2.7.3 Diagrama de casos de uso.....	45
2.7.4 Casos de uso expandidos.....	45
2.8 Conclusiones.....	52
Capítulo 3: Análisis y Diseño del Sistema.....	53
3.1 Introducción.	53
3.2 Modelo de análisis.	53
3.2.1 Diagrama de clases del análisis.....	53
3.2.1.1 Diagrama de clases del caso de uso “Solicitar Servicio Telemático”.....	54
3.2.1.2 Diagrama de clases del caso de uso “Gestionar Solicitud”.....	55

3.2.1.3 Diagrama de clases del caso de uso “Generar Reporte”	56
3.2.1.4 Diagrama de clases del caso de uso “Gestionar Servicio”	57
3.2.1.5 Diagrama de clases del caso de uso “Gestionar Usuario”	58
3.2.1.6 Diagrama de clases del caso de uso “Autenticar Usuario”	58
3.3 Modelo de diseño.....	59
3.3.1 Clases Base.....	59
3.3.1.1 Diagrama de clases base.....	60
3.3.1.2 Diagrama de secuencia.....	60
3.3.2 Diagrama de clases del Diseño Web.....	61
3.3.2.1 Caso de Uso”Solicitar Servicio Telemático”	62
3.3.2.2 Caso de Uso”Gestionar Solicitud”	63
3.3.2.3 Caso de Uso”Generar Reporte”	64
3.3.2.4 Caso de Uso”Gestionar Servicio”	65
3.3.2.5 Caso de Uso”Gestionar Usuario”	66
3.3.2.6 Caso de Uso”Autenticar Usuario”	67
3.4 Diseño de la Base de Datos.....	67
3.4.1 Diagrama de Clases Persistentes.....	68
3.4.2 Modelo de Datos.....	69
3.5 Principios de Diseño.....	69
3.5.1 Estándares en la interfaz de la aplicación.....	70
3.5.2 Formatos de reportes.....	71
3.5.3 Tratamiento de errores.....	71
3.6 Conclusiones.....	72
Capítulo 4: Implementación del Sistema	73
4.1 Introducción.....	73
4.2 Diagrama de Despliegue.....	73
4.3 Diagrama de Componentes.....	74
4.3.1 Diagrama de componentes de Base de Datos.....	74
4.3.1 Diagrama de componentes de Código Fuente.....	75
4.3.2 Diagrama de componentes de Código Ejecutable.....	75

4.5 Conclusiones.....	76
Capítulo 5: Estudio de Factibilidad.....	77
5.1 Introducción.....	77
5.2 Estimación basada en Puntos de Casos de Uso.....	77
5.3 Beneficios tangibles e intangibles.....	82
5.4 Análisis de costos y beneficios.....	82
5.5 Conclusiones.....	83
Conclusiones	84
Recomendaciones	85
Referencias Bibliográficas	86
Bibliografía	88
Glosario de Términos	89
Anexos	92

Introducción

La telemática, además de la diversidad de servicios que ofrece, presenta numerosas ventajas: aporta rapidez y flexibilidad en la transmisión de informaciones escritas, y permite la interacción con redes telemáticas de fácil manejo, como Internet. Además, dota a todos sus servicios de programas informáticos avanzados que ofrecen lenguajes de acceso simplificados, accesibles para todos.

La rápida evolución tecnológica, las nuevas necesidades ciudadanas y la expansión global de los servicios han conllevado a que los servicios telemáticos y la industria de contenidos relacionada con los mismos conformen una de las principales áreas de desarrollo estratégico y motor económico de la sociedad actual en el ámbito mundial.

Aunque desde hace décadas se habla de la telemática y de los impactos de las tecnologías de la información, la realidad es que el mercado de los servicios telemáticos se encuentra todavía en sus primeras fases de desarrollo, y como se mencionaba anteriormente, constituyen un proceso cuya fase más rápida se está produciendo en la actualidad.

El desarrollo de los servicios telemáticos en nuestro país, como en el resto del mundo, viene condicionado por el marco que rige las actividades de investigación y desarrollo tecnológico en el contexto nacional, se le presta un marcado interés al asunto, por considerar que la investigación y el desarrollo en este sector contribuirán a la expansión del mercado interior y, simultáneamente, mejorará el rendimiento de los grandes servicios públicos. Existen algunas empresas como CITMATEL dedicadas a impulsar el desarrollo de aplicaciones y servicios telemáticos avanzados con vistas a contribuir a la informatización de nuestro país.

La Universidad de las Ciencias Informáticas (UCI) tiene como misión preparar capital humano con una formación avanzada y de excelente nivel en las distintas áreas involucradas, en aras de lograr los objetivos de informatizar la sociedad cubana y posibilitar la inserción del país en el difícil mercado del software.

La UCI cuenta con tecnologías de punta, una amplia gama de información y servicios telemáticos, puestos a disposición de toda la comunidad universitaria, con la finalidad de fortalecer el proceso docente educativo. El uso del correo electrónico, de Internet, de la mensajería instantánea, la publicación de sitios Web, la apertura de puertos, el acceso remoto, la búsqueda de software son algunos de los servicios que disfrutaban los profesores, trabajadores y estudiantes de la Universidad que los soliciten, en dependencia de las labores que desempeñen y de la necesidad que tengan de utilizar los mismos en su actividad cotidiana. Es el Dpto. de Servicios Telemáticos el responsable de establecer y coordinar todos estos servicios de la red universitaria.

El proceso de solicitud de los servicios telemáticos en la actualidad no resulta suficientemente efectivo. Uno de los mayores problemas existentes es el hecho de que, tanto el llenado de solicitudes por los interesados, como el registro y revisión de todo el volumen de información acumulada al respecto, se efectúa manualmente, lo que provoca que el proceso sea lento y tedioso, y muy complicado a la hora de realizar búsquedas ante alguna reclamación por parte de los usuarios. La no-automatización del proceso de solicitud de los servicios telemáticos dificulta, además, el control estadístico de las solicitudes realizadas, la correspondiente ubicación espacio-temporal según las necesidades y la comprobación de firmas, así como la integridad del documento en cuestión.

La necesidad de dar solución a las situaciones antes expuestas, ha servido de motivación para la realización de este trabajo. El problema queda planteado del siguiente modo: ¿Cómo erradicar los problemas en el proceso de gestión de solicitudes de servicios telemáticos en la UCI?

La automatización de las solicitudes de los mencionados servicios beneficiaría a la comunidad universitaria, integrando la tecnología al servicio de los usuarios y restándole dificultad al proceso, por lo que se propone construir una herramienta capaz de gestionar las solicitudes de servicios telemáticos en la UCI de manera eficiente.

Se identifica como objeto de estudio de este trabajo el proceso de gestión de las solicitudes de servicios telemáticos en la UCI, mientras el campo de acción que barca el mismo es la automatización del proceso de gestión de las solicitudes de cuenta de electrónico e Internet, acceso remoto, publicación de sitios Web y apertura de puertos en dicha institución.

El objetivo general del trabajo consiste en desarrollar un sistema que permita automatizar la gestión de las solicitudes de servicios telemáticos en la UCI.

Los objetivos específicos que sirvieron de guía en todo el proceso investigativo fueron:

1. Definir y fundamentar de los principales elementos teóricos - metodológicos para el análisis del problema a resolver.
2. Desarrollar el análisis y diseño de una aplicación Web para la gestión de las solicitudes de los servicios telemáticos en la UCI.
3. Implementar el sistema con las características definidas en los procesos de análisis y diseño.

Para cumplimentar los objetivos planteados y darle solución a la situación problemática fue necesario ejecutar todo un conjunto de tareas científicas, entre las que se destacan las siguientes:

1. Análisis de las tecnologías que se utilizan para llevar a cabo sistemas como el que se pretende desarrollar.
2. Selección de la metodología de desarrollo de sistemas informáticos, que facilite la creación y garantice la calidad del sistema.
3. Selección y fundamentación de las herramientas que se utilizan para llevar a cabo el proyecto y la elección de la plataforma en la que se desarrolla la aplicación.
4. Estudio y concepción de flujos óptimos de las distintas solicitudes.
5. Diseño de una base de datos que soporte la mayoría de las funcionalidades del sistema.
6. Implementación de una aplicación Web que gestione las solicitudes de los servicios telemáticos en la UCI.

Los métodos científicos que han sido utilizados en el transcurso de la investigación permitieron apreciar el problema con objetividad, integralmente, en todas sus dimensiones. El método Analítico – Sintético, fue importante por la gran cantidad de documentos, artículos y libros consultados con el objetivo de fundamentar el tema correspondiente y hacer una selección correcta de las tecnologías, metodologías de desarrollo y herramientas a utilizar. La Entrevista al cliente es esencial a la hora de definir qué es lo que el sistema debe hacer, identificando las funcionalidades requeridas por el mismo y las restricciones que se imponen. El método de la Modelación resultó imprescindible por el gran peso que tienen los modelos en el trabajo, donde la realidad es representada a través de un modelo que guía todo el proceso hasta la

obtención del producto esperado, el cual será capaz de resolver los problemas planteados en la gestión de las solicitudes de los servicios telemáticos mencionados.

Como resultado de este trabajo, se obtiene, por primera vez en la Universidad de las Ciencias Informáticas, una aplicación Web para gestionar las solicitudes de los servicios telemáticos. Esta herramienta informática proporciona comodidad, rapidez y confiabilidad a la hora del llenado, búsqueda y verificación de las solicitudes, un mayor control por parte de los directivos, así como un análisis certero de los comportamientos de dichas solicitudes.

El contenido de este trabajo se encuentra estructurado en cuatro capítulos de la siguiente manera:

En el capítulo 1, “Fundamentación del tema”, se hace una descripción detallada del objeto de estudio, se valoran sistemas existentes vinculados al campo de acción, se enuncian las tendencias y tecnologías actuales seleccionadas a emplear en el desarrollo de la propuesta y se fundamenta el por qué de su utilización.

El capítulo 2, “Características del sistema”, describe los procesos mediante un modelo del negocio, identificando actores, trabajadores y casos de uso del negocio. Se definen además las funcionalidades del sistema, a través de los requerimientos funcionales y no funcionales que se describen detalladamente.

El capítulo 3, “Análisis y diseño del sistema”, trata sobre la elaboración de la propuesta de solución. Se modelan diagramas de clases, de datos, y se plantean los principios de diseño que se seguirán en la implementación del sistema.

En el capítulo 4, “Implementación del sistema” se procede con la construcción de la propuesta de solución. Se modela el diagrama de despliegue y el diagrama de componentes para algunas partes del sistema.

En el capítulo 5, “Estudio de factibilidad” se determina si es factible o no el desarrollo del software propuesto según los resultados que arrojan el cálculo del esfuerzo, tiempo de desarrollo y costo del proyecto.

Capítulo 1

Fundamentación del Tema

1.1 Introducción.

En el presente capítulo se brinda una visión general de los aspectos relacionados con los servicios telemáticos, los conceptos necesarios para la comprensión del problema planteado y la solución propuesta. Se aborda también sobre los sistemas existentes vinculados al campo de acción, y se describen además las tecnologías, metodologías y herramientas actuales adecuadas para llevar a cabo el sistema que se pretende desarrollar.

1.2 Servicios Telemáticos.

El transcurrir de los años fue testigo de los innumerables avances que el ser humano obtuvo con el propósito de recopilar y hacer llegar la información a mayor cantidad de personas en el menor tiempo posible. Desde el telégrafo hasta las transmisiones de radio y televisión, desde el papel y las cintas magnéticas hasta las computadoras, ha sido extraordinario el desarrollo alcanzado en los medios de transmisión, almacenamiento y procesamiento de la información; dividido en dos grandes ramas: las telecomunicaciones y la informática.

En un principio las telecomunicaciones y la informática tuvieron una evolución y avance paralelos, marcados por el aislamiento entre ellas, donde cada una utilizaba los adelantos tecnológicos y científico-técnicos en su propio desarrollo como un ente aislado. El surgimiento, a finales de la década del 60 del siglo pasado, de las redes de computadoras dio al traste con este divorcio; dando origen a la telemática, definida como un conjunto de servicios y técnicas que asocian las telecomunicaciones y la informática, y que se centra en el estudio, diseño y gestión de las redes y servicios de comunicación de datos, transportando texto, audio, video, o combinaciones de los mismos.

A partir de ese momento aparece una inmensa gama de servicios telemáticos, cuyo objetivo primordial es el intercambio de información hacia cualquier lugar del planeta por medio de una computadora. Entre ellos están: la conexión a un ordenador desde otro lugar (telnet); la transferencia de ficheros entre una

computadora local y una computadora remota (protocolo de transferencia de ficheros, o FTP); la lectura e interpretación de ficheros de una máquina remota: no sólo texto sino imágenes, sonidos o secuencias de vídeo (protocolo de transferencia de hipertexto, o HTTP); el intercambio de mensajes de correo electrónico (e-mail); el acceso a grupos de noticias y foros de debates (news) y conversaciones en tiempo real (Chat, IRC), entre otros.

La vertiginosa expansión de Internet brinda la posibilidad de que aumente cada día el número de personas que tenga acceso a los servicios telemáticos, encontrando en ellos una vía rápida de comunicación, una fuente inagotable de información y conocimientos, así como una forma mejor y más eficiente de desarrollo.

1.3 Proceso de solicitudes de servicios telemáticos.

El vertiginoso desarrollo logrado por la telemática en nuestros días, ha tributado en favor de un mayor intercambio de información en el ámbito mundial, que avanza cada día más y pretende alcanzar límites insospechados. Tal situación presupone el aumento y desarrollo de políticas de control del acceso a este valioso recurso, dado el nivel de confidencialidad e integridad que debe tener la información generada, que se encuentra disponible y accesible por medio de las redes de computadoras actuales.

Es por ello que la persona interesada en obtener un determinado servicio telemático, generalmente, debe llenar una planilla, así se dispone en el caso que nos ocupa. Los trámites a que es sometida esa solicitud hasta ser aprobado o denegado finalmente el servicio al cliente, es a lo que nos referimos con proceso de solicitud.

El proceso de solicitud para el acceso a los servicios telemáticos por parte de los usuarios es algo que no sigue un esquema general que haya sido planteado. Cada empresa, organismo o institución dedicada a la prestación de estos servicios, se encarga de establecer las políticas a seguir hasta el establecimiento del acceso de sus usuarios a los servicios que se brindan. Es por ello que la labor de gestionar las solicitudes de los servicios telemáticos de forma automatizada constituye una tarea puntual para cada centro que desee implantar un sistema de este tipo.

En la Universidad de las Ciencias Informáticas no ha existido ningún sistema encargado de gestionar las solicitudes realizadas por los usuarios que desean beneficiarse con un determinado servicio telemático. Es por esto que se realiza un estudio sin ningún antecedente en nuestro centro.

1.4 Firma digital.

La firma digital es una herramienta tecnológica que permite garantizar la autoría e integridad de los documentos digitales, pero que no implica asegurar la confidencialidad del mensaje; un documento firmado digitalmente puede ser visualizado por otras personas, al igual que cuando se firma holográficamente.

Una firma digital es un conjunto de datos asociados a un mensaje digital que permite garantizar la identidad del firmante y la integridad del mensaje, y engloba dos procesos fundamentales:

Firma de los datos. Se lleva a cabo por medio de la clave privada, añadiendo la firma al mensaje enviado.

Verificación de la firma. Se utiliza la clave pública. En este proceso se comprueba que el contenido no ha sido modificado.

¿Cómo funciona? [1]

La firma digital funciona utilizando complejos procedimientos matemáticos que relacionan el documento firmado con información propia del firmante, y permiten que terceras partes puedan reconocer la identidad del firmante y asegurarse de que los contenidos no han sido modificados.

El firmante genera, mediante una función matemática, una huella digital del mensaje, la cual se cifra con la clave privada del firmante. El resultado es lo que se denomina firma digital, que se enviará adjunta al mensaje original. De esta manera el firmante adjuntará al documento una marca que es única para dicho documento y que sólo él es capaz de producir.

Para realizar la verificación del mensaje, en primer término el receptor generará la huella digital del mensaje recibido, luego descifrará la firma digital del mensaje utilizando la clave pública del firmante y obtendrá de esa forma la huella digital del mensaje original; si ambas huellas digitales coinciden, significa que no hubo alteración y que el firmante es quien dice serlo.

¿Claves privadas y claves públicas? [1]

En la elaboración de una firma digital y en su correspondiente verificación se utilizan complejos procedimientos matemáticos basados en criptografía asimétrica, donde cada usuario cada usuario posee un par de claves propio (clave privada y clave pública).

El sistema opera de tal modo que la información cifrada con una de las claves sólo puede ser descifrada con la otra. De este modo si un usuario cifra determinada información con su clave privada, cualquier persona que conozca su clave pública podrá descifrar la misma. En consecuencia, si es posible descifrar un mensaje utilizando la clave pública de una persona, entonces puede afirmarse que el mensaje lo generó esa persona utilizando su clave privada (probando su autoría).

1.5 Tendencias y tecnologías actuales utilizadas.

Teniendo en cuenta las necesidades vistas y las características del entorno donde se aplicará la solución propuesta, se realizó un estudio de las tendencias y tecnologías actuales posibles a emplear, las cuales se describen a continuación.

1.5.1 Algoritmos de firma digital.

Como bien se mencionaba anteriormente, la firma digital es, en la transmisión de mensajes telemáticos, un método criptográfico que asegura su integridad así como la identidad del remitente.

Para firmar los mensajes se emplean algoritmos de clave pública y funciones de dispersión. [2]

- ◆ Algoritmos de clave pública o asimétricos.

Encriptan un mensaje generando un texto cifrado del mismo tamaño que el original. Usan una clave para cifrar el mensaje (clave privada) y otra para descifrar (clave pública). La idea fundamental es que las claves están ligadas matemáticamente pero es computacionalmente imposible obtener una a partir de la otra. Tienen un coste computacional alto y se suelen emplear para distribuir las claves de los algoritmos simétricos.

- ◆ Algoritmos de resumen de mensajes o función de dispersión criptográfica.

Transforman mensajes de tamaño variable a textos cifrados de tamaño fijo sin emplear claves. Se emplean para convertir mensajes grandes en representaciones más manejables.

Estos algoritmos deben tener tres propiedades para ser criptográficamente seguros:

1. No debe ser posible averiguar el mensaje de entrada basándose sólo en su resumen, es decir, el algoritmo es una función irreversible de una sola dirección.
2. Dado un resumen debe ser imposible encontrar un mensaje que lo genere.
3. Debe ser computacionalmente imposible encontrar dos mensajes que generen el mismo resumen.

Dentro de los algoritmos asimétricos se destacan:

◆ **RSA**

El algoritmo RSA es un algoritmo de dominio público, tras haber expirado la patente RSA, luego de 17 años, el 20 de Septiembre del 2000. Es el más conocido y usado de los sistemas de clave pública, y también el más rápido de ellos.

El sistema RSA se basa en el hecho matemático de la dificultad de factorizar números primos muy grandes y funciona de la siguiente manera: [3]

1. Inicialmente es necesario generar aleatoriamente dos números primos grandes, a los que llamaremos p y q .
2. A continuación calcularemos n como producto de p y q : $n = p * q$ (n debe ser computacionalmente imposible de factorizar).
3. Se calcula ϕ : $\phi(n)=(p-1)(q-1)$
4. Se calcula un número natural e de manera que $\text{MCD}(e, \phi(n))=1$, es decir e debe ser primo relativo de $\phi(n)$.
5. Es lo mismo que buscar un número impar por el que dividir $\phi(n)$ que de cero como resto.
6. Mediante el algoritmo extendido de Euclides se calcula d : $e \cdot d \bmod \phi(n)=1$ Puede calcularse $d = ((Y \cdot \phi(n)) + 1) / e$ para $Y=1, 2, 3, \dots$ hasta encontrar un d entero.
7. El par de números (e, n) son la clave pública.
8. El par de números (d, n) son la clave privada.
9. Cifrado: La función de cifrado es $C = M^e \bmod n$
10. Descifrado: La función de descifrado es $M = C^d \bmod n$

La seguridad de este algoritmo radica en ser una función computacionalmente segura, ya que si bien realizar la exponenciación modular es fácil, su operación inversa, la extracción de raíces de módulo \emptyset no es factible a menos que se conozca la factorización de e , clave privada del sistema.

En cuanto a las longitudes de claves, el sistema RSA permite longitudes variables, siendo aconsejable actualmente el uso de claves de no menos de 1024 bits (se han roto claves de hasta 512 bits, aunque se necesitaron más de 5 meses y casi 300 ordenadores trabajando juntos para hacerlo). Por lo tanto, si n es suficientemente grande el algoritmo RSA es seguro.

◆ **DSA** (Digital Signature Algorithm, en español Algoritmo de Firma digital)

Fue un algoritmo propuesto por el Instituto Nacional de Normas y Tecnología de los Estados Unidos para su uso en su Estándar de Firma Digital (DSS) y se hizo público el 30 de agosto de 1991. Este algoritmo como su nombre lo indica, sirve para firmar y no para cifrar información. [4]

DSA usa una clave privada para que forme la firma digital con su correspondiente clave pública para así poder verificar la firma. A diferencia del cifrado, la operación de firma no es reversible. El DSA no hace encriptación de clave pública, y el standard DSS no provee capacidades para distribución ni intercambio de claves. Una desventaja de este algoritmo es que requiere mucho más tiempo de cómputo que RSA. [5]

◆ **EIGamal**

El algoritmo ElGamal es de libre uso. Fue descrito por Taher Elgamal en 1984 y se basa en la dificultad de calcular algoritmos discretos en un campo finito. En un principio fue ideado para producir firmas digitales, aunque después se extendió su uso para utilizarlo en el cifrado de mensajes. Generalmente no se utiliza de forma directa, ya que la velocidad de cifrado y autenticación es inferior a la obtenida con RSA, y además las firmas producidas son más largas.

Algunos algoritmos de resumen de mensajes son:

◆ **MD5** (Message Digest 5)

El MD5 es un algoritmo de reducción criptográfico ampliamente usado. Fue desarrollado como reemplazo del algoritmo MD4 después de que se descubriese su debilidad.

MD5 produce un número de 128 bits a partir de un texto de cualquier longitud. Comienza rellenando el mensaje a una longitud congruente con $448 \bmod 512$, es decir la longitud del mensaje es 64 bits menos que un entero múltiplo de 512. El relleno consiste en un bit en 1 seguido por cuantos bits en 0 sean necesarios. La longitud original del mensaje es almacenada en los últimos 64 bits del relleno. [6]

Adicionalmente se inicializa, con un valor fijo, un buffer de 128 bits. Este buffer puede verse como 4 registros de 32 bits (A, B, C, D) y son inicializados con los siguientes valores hexadecimales: A=67452301; B=EFCDAB89; C=98BADCFE; D=10325476

Durante varias rondas de procesamiento el algoritmo toma bloques de 512 bits de la entrada y los mezcla con los 128 bits del buffer. Este proceso es repetido hasta que todos los bloques de entrada han sido consumidos. El valor resultante en el buffer es el hash del mensaje. [6]

◆ **SHA** (Secure Hash Algorithm, en español Algoritmo de Hash Seguro)

SHA es un algoritmo no libre desarrollado como parte integrante del Secure Hash Standar (SHS) y el Digital Signature Standar (DSS) por la Agencia de Seguridad Nacional Norteamericana (NSA).

A diferencia de MD5, SHA genera números hash de 160 bits y se basa en principios similares a los usados en el diseño de los algoritmos de resumen del mensaje MD4 y MD5. La versión, SHA-1, actual se considera segura (por lo menos hasta que se demuestre lo contrario).

En el sistema que se propone se ha implementado la firma digital combinando el algoritmo de hashing MD5 con RSA.

1.5.2 Aplicaciones Web.

Las aplicaciones Web se desarrollan como una extensión de los Sistemas Web para agregar funcionalidad de negocio al proceso. En términos más simples, las Aplicaciones Web utilizan las tecnologías existentes para generar contenidos dinámicos y permitir a los usuarios del sistema modificar la lógica del negocio en el servidor. La arquitectura de una aplicación Web incluye la aplicación en el Servidor, que es la que permite al sistema manejar lógica de negocio y tener un estado.

Por tanto, las aplicaciones Web no son más que una especialización y concreción de las aplicaciones cliente-servidor, o sea, su arquitectura general es la de un sistema cliente/servidor, donde tanto el cliente

(el navegador) como el servidor (el servidor Web), y el protocolo mediante el que se comunican (el HTTP: HyperText Transfer Protocol) son estándar, y no han de ser creados por el desarrollador. [7]

Por consiguiente, generan dinámicamente una serie de páginas en un formato estándar, soportado por navegadores Web comunes como HTML o XHTML. Se utilizan lenguajes interpretados del lado del cliente, tales como JavaScript, para añadir más funcionalidad a la interfaz de usuario, especialmente para crear una experiencia interactiva que no requiera recargar la página cada vez.

Una ventaja significativa en la construcción de aplicaciones Web que soporten las características de los browsers estándar es que deben funcionar igual independientemente de la versión del sistema operativo instalado en el cliente. En vez de crear clientes para Windows, MacOS X, GNU/Linux, y otros sistemas operativos, la aplicación es escrita una vez y es mostrada casi en todos lados. Sin embargo, aplicaciones inconsistentes de HTML, CSS, DOM y otras especificaciones de browsers pueden causar problemas en el desarrollo y soporte de aplicaciones Web. [8]

Se mencionarán otras ventajas que han contribuido a fomentar la elección. [9]

- Multiplataforma: Con un solo programa, un único ejecutable, nuestras aplicaciones pueden ser utilizadas a través de múltiples plataformas, tanto de hardware como de software.
- Actualización instantánea: Debido que todos los usuarios de la aplicación hacen uso de un sólo programa que radica en el servidor, los usuarios siempre utilizarán la versión más actualizada del sistema.
- Suave curva de aprendizaje: Los usuarios, como utilizan la aplicación a través de un navegador, hacen uso del sistema tal como si estuvieran navegando por Internet, por lo cual su acceso es más intuitivo.
- Fácil de integrar con otros sistemas: Debido a que se basa en protocolos estándares, la información manejada por el sistema puede ser accedida con mayor facilidad por otros sistemas.
- Acceso móvil: El usuario puede acceder a la aplicación con la única restricción de que cuente con un acceso a la red privada de la organización o a Internet, dependiendo de las políticas de dicha organización; puede hacerlo desde una computadora de escritorio, una laptop o desde una agenda electrónica; desde su oficina, hogar u otra parte del mundo.

1.5.3 Modelo Cliente/Servidor.

La arquitectura Cliente/Servidor es la integración distribuida de un sistema en red, con los recursos, medios y aplicaciones que, definidos modularmente en los servidores, administran, ejecutan y atienden las solicitudes de los clientes; todos interrelacionados física y lógicamente, compartiendo datos, procesos e información. Se establece así un enlace de comunicación transparente entre los elementos que conforman la estructura. [10]

Entre las principales características de la arquitectura cliente/servidor se pueden destacar: [11]

- ◆ El servidor presenta a todos sus clientes una interfaz única y bien definida.
- ◆ El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- ◆ El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- ◆ Los cambios en el servidor implican pocos o ningún cambio en el cliente.

Ventajas de la arquitectura cliente-servidor: [12]

- ◆ Centralización del control: los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema.
- ◆ Escalabilidad: se puede aumentar la capacidad de clientes y servidores por separado.
- ◆ Se reduce el tráfico de red considerablemente. Idealmente, el cliente se comunica con el servidor utilizando un protocolo de alto nivel de abstracción como por ejemplo SQL

La tecnología cliente/servidor es ampliamente difundida y muy utilizada por las aplicaciones Web.

1.5.4 Lenguajes de Programación para la Web.

Hoy en día existen diferentes lenguajes de programación para Web. Dichos lenguajes se clasifican en dos partes fundamentales que reconocen la propia arquitectura Cliente/Servidor de esta plataforma de desarrollo: los lenguajes del lado del Servidor y los lenguajes del lado del Cliente.

Entre los lenguajes del lado del servidor podemos encontrar entre los más sobresalientes por el auge que estos han tenido, algunos como PERL, ASP, PHP, Java, JSP, los módulos CGIs e ISAPIs, etc. Estos se caracterizan por desarrollar la lógica de negocio dentro del Servidor, además de ser los encargados del

acceso a Bases de Datos, tratamiento de la Información etc. Del lado del cliente se encuentran principalmente el JavaScript y el Visual Basic Script, que son los encargados de aportar dinamismo a la aplicación en los navegadores. [13]

Luego de haber analizado las características fundamentales de los lenguajes de programación candidatos para la implementación de la propuesta de este trabajo se considera que PHP resulta el más idóneo. A continuación se exponen las razones que respaldan la decisión tomada.

1.5.4.1 PHP (PHP: Hypertext Preprocessor).

PHP (Personal Home Page) es el acrónimo de Hypertext Preprocessor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. Es también un lenguaje interpretado y embebido en el HTML. [13]

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como UNIX (y de ese tipo, como Linux), Windows y Mac OS X, y puede interactuar con los servidores de Web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI. [14]

Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite; lo cual permite la creación de Aplicaciones Web muy robustas. [14]

Sin dejar de mencionar además que: [13]

- ◆ Es software libre, lo que implica menos costes y servidores más baratos que otras alternativas.
- ◆ Es muy rápido. Su integración con la base de datos MySQL y el servidor Apache, le permite constituirse como una de las alternativas más atractivas del mercado.
- ◆ Su sintaxis está inspirada en C, ligeramente modificada para adaptarlo al entorno en el que trabaja, de modo que si se está familiarizado con esta sintaxis, le resultara muy fácil aprender PHP.
- ◆ Su librería estándar es realmente amplia, lo que permite reducir los llamados "costes ocultos", uno de los principales defectos de ASP.
- ◆ PHP tiene una de las comunidades más grandes en Internet, con lo que no es complicado encontrar ayuda, documentación, artículos, noticias, y más recursos.

PHP es la gran tendencia en el mundo de Internet. Últimamente se puede observar un ascenso imparable, ya que cada día son muchísimas más las aplicaciones Web que lo utilizan para su funcionamiento.

1.5.5 Servicios Web (Web Services).

El objetivo de los servicios Web es lograr la interoperatividad, o sea la capacidad de que sistemas dispares puedan comunicarse y compartir información.

Un servicio Web es una entidad programable que proporciona alguna funcionalidad determinada, y es accesible a cualquier número de sistemas que usen las normas de Internet XML y HTTP. Estos servicios permiten la ejecución de sus funcionalidades sin importar la plataforma, sistema operativo, o lenguaje en el cual estén implementados. [13] La interoperabilidad se consigue mediante la adopción de estándares abiertos.

Estándares empleados: [15]

- ◆ Web Services Protocol Stack: Así se denomina al conjunto de servicios y protocolos de los servicios Web.
- ◆ XML: Es el formato estándar para los datos que se vayan a intercambiar.
- ◆ SOAP o XML-RPC: Protocolos sobre los que se establece el intercambio.

Los datos en XML también pueden enviarse de una aplicación a otra mediante protocolos normales como HTTP, FTP, o SMTP.

- ◆ WSDL: Es el lenguaje de la interfaz pública para los servicios Web. Es una descripción basada en XML de los requisitos funcionales necesarios para establecer una comunicación con los servicios Web.
- ◆ UDDI: Protocolo para publicar la información de los servicios Web. Permite a las aplicaciones comprobar qué servicios web están disponibles.
- ◆ WS-Security: Protocolo de seguridad aceptado como estándar por OASIS. Garantiza la autenticación de los actores y la confidencialidad de los mensajes enviados.

Gracias a los servicios Web se logra que sistemas heterogéneos trabajen conjuntamente como una sola aplicación computacional; en el caso de este trabajo, conociendo que existe una aplicación de búsqueda soportada por un catálogo que contiene toda la información referente a los trabajadores y usuarios de la

Universidad, se hace uso de esta tecnología para que este sistema contribuya al objetivo que se quiere lograr con la propuesta que se presenta.

1.5.6 Servidor Web Apache.

Un servidor Web es un programa que implementa el *protocolo HTTP (hypertext transfer protocol)* y se encarga de atender las *peticiones HTTP* llevada a cabo por un *cliente*, y responder con el contenido que el cliente solicita. Este protocolo HTTP está diseñado para transferir lo que llamamos hipertextos, páginas Web o páginas HTML.

Algunos servidores Web importantes son el Apache, Internet Information Server (IIS), y el Cherokee. Para desarrollar nuestra propuesta, se ha seleccionado el servidor Web Apache por ser el más factible de acuerdo a sus características.

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation y hoy en día es el servidor Web más utilizado del mundo ofreciendo una perfecta combinación entre estabilidad y sencillez, encontrándose muy por encima de sus competidores, tanto gratuitos como comerciales. Es un software de código abierto que funciona sobre cualquier plataforma. Por supuesto, se distribuye prácticamente con todas las implementaciones de Linux.

Las principales características de Apache son: [\[16\]](#)

- ◆ Funcionalidad en múltiples plataformas.
- ◆ Elaborado índice de directorios.
- ◆ Soporte del último protocolo http 1.1.
- ◆ Sencilla administración basada en la configuración de un único archivo.
- ◆ Soporte para CGI (Common Gateway Interface) y FastCGI
- ◆ Mensajes de error altamente configurables.
- ◆ Bases de datos de autenticación y negociado de contenido.

Apache está diseñado para ser un servidor Web potente y flexible que pueda funcionar en la más amplia variedad de plataformas y entornos, a través de su diseño modular. Es decir, está dividido en muchas porciones de código que hacen referencia a diferentes aspectos o funcionalidades del servidor Web,

permitiendo que una misma característica o funcionalidad sea implementada de diferente manera para obtener una mayor eficiencia.

Los módulos del Apache se pueden clasificar en tres categorías: [10]

- ◆ Módulos Base: Módulo con las funciones básicas del Apache.
- ◆ Módulos Multiproceso: Son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender a las peticiones.
- ◆ Módulos Adicionales: Cualquier otro módulo que le añada una funcionalidad al servidor.

Las funcionalidades más elementales se encuentran en el módulo base, siendo necesario un módulo multiproceso para manejar las peticiones. Se han diseñado varios módulos multiprocesos para cada uno de los sistemas operativos sobre los que se ejecuta el Apache, optimizando el rendimiento y rapidez del código. [10]

El resto de funcionalidades del servidor se consigue por medio de módulos adicionales que se pueden cargar. Para añadir un conjunto de utilidades al servidor, simplemente hay que añadirle un módulo, de forma que no es necesario volver a instalar el software.

Este diseño permite a los administradores, elegir que características van a ser incluidas en el servidor seleccionando que módulos se van a cargar, ya sea al compilar o al ejecutar el servidor. [17]

Beneficios: [17]

- ◆ Apache puede soportar de una forma más fácil y eficiente una amplia variedad de sistemas operativos. En concreto, la versión de Windows de Apache es mucho más eficiente, porque el módulo `mpm_winnt` puede usar funcionalidades nativas de red en lugar de usar la capa POSIX como hace Apache 1.3. Este beneficio se extiende también a otros sistemas operativos que implementan sus respectivos MPMs.
- ◆ El servidor puede personalizarse mejor para las necesidades de cada sitio Web. Por ejemplo, los sitios Web que necesitan más que nada escalabilidad pueden usar un MPM hebrado como `worker`, mientras que los sitios Web que requieran por encima de otras cosas estabilidad o compatibilidad con software antiguo pueden usar `prefork`. Además, se pueden configurar funcionalidades especiales como servir diferentes hosts con diferentes identificadores de usuario (`perchild`).

1.5.7 Sistema de Gestión de Base de Datos (SGBD).

Un Sistema de Gestión de Bases de Datos (SGBD) puede definirse como un paquete generalizado de software, que se ejecuta en un sistema computacional anfitrión, centralizando los accesos a los datos y actuando de interfaz entre los datos físicos y el usuario. [10] Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta y tiene como propósito general manejar de manera clara, sencilla y ordenada un conjunto de información.

Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad. [10]

Ventajas que ofrecen los SGBD: [10]

- ◆ Facilidad de manejo de grandes volúmenes de información.
- ◆ Gran velocidad en muy poco tiempo.
- ◆ Independencia del tratamiento de información.
- ◆ Seguridad de la información (acceso a usuarios autorizados), protección de información, de modificaciones, inclusiones, consulta.
- ◆ No hay duplicidad de información, comprobación de información en el momento de introducir la misma.
- ◆ Integridad referencial al terminar los registros.

Oracle, MySQL, Microsoft SQL Server, PostgreSQL, Internase, son algunos de los SGBD comúnmente utilizados en el mundo. Todos estos presentan un enfoque relacional con un buen basamento matemático centrado en el Álgebra Relacional.

1.5.7.1 MySQL

Como SGBD se seleccionó MySQL ya que es un sistema de gestión de bases de datos libre, multihilo y multiusuario con más de seis millones de instalaciones.

MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. 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. [18]

MySQL funciona sobre múltiples plataformas y existen varias APIs que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, incluyendo C, C++, C#, Pascal, Delphi (vía dbExpress), Eiffel, Smalltalk, Java (con una implementación nativa del driver de Java), Lisp, Perl, PHP, Python, Ruby, REALbasic (Mac), FreeBASIC, y Tcl; cada uno de estos utiliza una API específica. También existe un interfaz ODBC, llamado MyODBC que permite a cualquier lenguaje de programación que soporte ODBC comunicarse con las bases de datos MySQL. [18]

Son muchas las características de MySQL que se tuvieron en cuenta a la hora de la elección, por ejemplo ofrece un sistema de contraseñas y privilegios seguro mediante verificación basada en el host y el tráfico de contraseñas está encriptado al conectarse a un servidor, soporta gran cantidad de datos, contiene su propio paquete de pruebas de rendimiento proporcionado con el código fuente de la distribución de MySQL, pero son las características distintivas las que hacen la diferencia, características que son implementadas únicamente por MySQL:

- ◆ 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.

1.5.8 Patrones de arquitectura.

La utilización de los patrones ha tomado un gran auge a partir del desarrollo del modelo orientado a objetos. Un patrón describe un problema que ocurre una y otra vez en nuestro entorno y describe también el núcleo de la solución al problema, de forma que puede reutilizarse continuamente. Los patrones capturan la experiencia existente y probada para promover buenas prácticas.

La arquitectura es el esqueleto o base de una aplicación, en esta se analiza la aplicación desde varios puntos de vista.

Por ende, se puede decir que los patrones de arquitectura expresan el esquema fundamental de organización para sistemas de software. Proveen un conjunto de subsistemas predefinidos; especifican sus responsabilidades e incluyen reglas y guías para organizar las relaciones entre ellos. Ayudan a especificar la estructura fundamental de una aplicación. Cada actividad de desarrollo es gobernada por esta estructura; por ejemplo, el diseño detallado de los subsistemas, la comunicación y colaboración entre diferentes partes del sistema, etc. Cada patrón de arquitectura ayuda a conseguir una propiedad específica en el sistema global. [19]

Constituyen ejemplos de patrones de arquitectura el modelo de capas y el modelo vista controlador, resultando este último el más idóneo para guiar este trabajo.

1.5.8.1 Modelo Vista Controlador (MVC).

El **Model-View-Controller** (Modelo-Vista-Controlador, en adelante MVC) fue introducido inicialmente en la comunidad de desarrolladores de Smalltalk-80. MVC divide una aplicación interactiva en 3 áreas: procesamiento, salida y entrada. Para esto, utiliza las siguientes abstracciones: [19]

- ◆ **Modelo (Model):** Encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento de entrada.
- ◆ **Vista (View):** Muestra la información al usuario. Obtiene los datos del modelo. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador.
- ◆ **Controlador (Controller):** Reciben las entradas, usualmente como eventos que codifican los movimientos o pulsación de botones del ratón, pulsaciones de teclas, etc. Los eventos son traducidos a solicitudes de servicio (“**service requests**” en el texto original) para el modelo o la vista. El usuario interactúa con el sistema a través de los controladores.

Las Vistas y los Controladores conforman la interfaz de usuario. Un mecanismo de propagación de cambios asegura la consistencia entre la interfaz y el modelo. La separación del modelo de los componentes vista y del controlador permite tener múltiples vistas del mismo modelo. Si el usuario cambia el modelo a través del controlador de una vista, todas las otras vistas dependientes deben reflejar los cambios. Por lo tanto, el modelo notifica a todas las vistas siempre que sus datos cambien. Las vistas, en cambio, recuperan los nuevos datos del modelo y actualizan la información que muestran al usuario. [19]

Para una mejor comprensión del funcionamiento del patrón, ver anexo 1.

Este patrón es muy popular y ampliamente utilizado en múltiples plataformas y lenguajes. Algunos de sus principales beneficios son: [19]

- Menor acoplamiento
 - Desacopla las vistas de los modelos
 - Desacopla los modelos de la forma en que se muestran e ingresan los datos
- Mayor cohesión
 - Cada elemento del patrón está altamente especializado en su tarea (la vista en mostrar datos al usuario, el controlador en las entradas y el modelo en su objetivo de negocio)
- Las vistas proveen mayor flexibilidad y agilidad
 - Se puede crear múltiples vistas de un modelo
 - Se puede crear, añadir, modificar y eliminar nuevas vistas dinámicamente
 - Las vistas pueden anidarse
 - Se puede cambiar el modo en que una vista responde al usuario sin cambiar su representación visual
 - Se puede sincronizar las vistas
 - Las vistas pueden concentrarse en diferentes aspectos del modelo
- Mayor facilidad para el desarrollo de clientes ricos en múltiples dispositivos y canales
 - Una vista para cada dispositivo que puede variar según sus capacidades
 - Una vista para la Web y otra para aplicaciones de escritorio
- Más claridad de diseño
- Facilita el mantenimiento
- Mayor escalabilidad

Patrones de diseño en el MVC

Un patrón de arquitectura puede contener varios patrones de diseño. Estos patrones resuelven problemas de diseño específicos y hacen el diseño flexible y reutilizable.

El patrón de arquitectura Model-View-Controller contiene (o puede contener) los siguientes patrones de diseño: [19]

- ◆ **Observer:** Para el mecanismo de publicación y suscripción que permite la notificación de los cambios en el modelo a las vistas.

- ◆ **Composite:** para la creación de vistas compuestas. Utilizando este patrón podemos crear una jerarquía de vistas y tratar a cada vista compuesta igual que una a una vista normal.
- ◆ **Strategy:** En la relación entre las vistas y los controladores. Utilizando este patrón podemos cambiar dinámicamente o en tiempo de compilación los algoritmos del controlador mediante los cuales responde a su entorno.
- ◆ **Factory Method:** Para especificar la clase controlador predeterminada de una vista.
- ◆ **Decorator:** Para añadir capacidades adicionales a una vista (por ejemplo, scroll).
- ◆ **Proxy:** Para distribuir la arquitectura (Modelo y Vista-Controlador) en diferentes emplazamientos.

Los patrones de diseño han contribuido a dar flexibilidad y extensibilidad a nuestros diseños. Pero en adición, han demostrado ser una forma muy útil (exitosa) de reutilizar diseño, ya que ellos no sólo nombran, abstraen e identifican aspectos claves de estructuras comunes de diseño, sino que generalmente son descritos en una forma específica documental, haciendo su comprensión y aplicación fácil para el conjunto de desarrolladores. [20]

1.5.9 Proceso de desarrollo de Software.

La tendencia actual al crecimiento del volumen y complejidad de los productos en la Industria del Software exigen mayor calidad y productividad de los mismos. Para enfrentar esta situación las empresas requieren desarrollar o adquirir una disciplina en el desarrollo del software y controlar que los ingenieros usen de forma consistente los nuevos métodos, esto solo es posible si se sigue una metodología específica, que se encargará de establecer los principios necesarios para la obtención de un software económico, fiable y que funcione eficientemente.

No existe una metodología universal para el desarrollo del software, puesto que las características de cada proyecto exigen que el proceso sea configurable.

En la actualidad existen varias metodologías orientadas a objetos (OO) basadas en UML: Rational Unified Process (RUP), OPEN, MÉTRICA 3.

Por las facilidades que aporta a todo el proceso, se decidió que el Proceso Unificado de Desarrollo de Software (RUP) será la metodología de Ingeniería de Software que guiará nuestra propuesta.

1.5.9.1 Metodología RUP.

El Proceso Unificado de Rational, (Rational Unified Process, de ahí las siglas RUP), fue publicado en 1998 como resultado de varios años de experiencia y unifica los mejores elementos de metodologías anteriores.

RUP es una propuesta de proceso para el desarrollo de software orientado a objeto que utiliza *Unified Model Language (UML)* como lenguaje de representación visual y está basado en componentes, lo que quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas. Además cubre el ciclo de vida de desarrollo de un proyecto y toma en cuenta las mejores prácticas a utilizar en el modelo de desarrollo de software.

En RUP se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo. El proceso de desarrollo esta dividido en ciclos, teniendo un producto al final de cada ciclo. A su vez cada ciclo se divide en cuatro Fases: Inicio, Elaboración, Construcción y Transición, cada una de las cuales concluye con un hito bien definido donde deben tomarse ciertas decisiones. En la Figura 1: RUP en dos dimensiones, se representa el proceso en el que se grafican los flujos de trabajo y las fases y muestra la dinámica expresada en iteraciones y puntos de control.

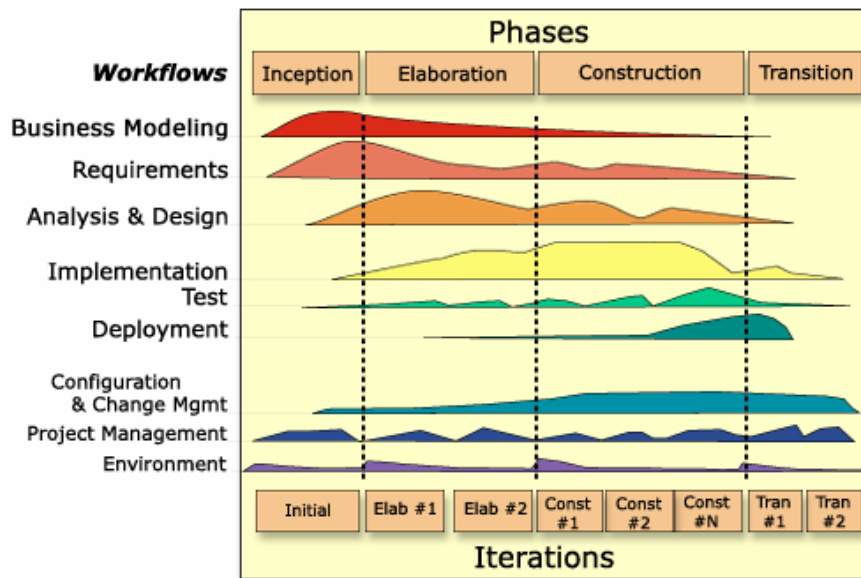


Figura 1.1: RUP en dos dimensiones.

El ciclo de vida de RUP se caracteriza por estar: [21]

- ◆ Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo).
- ◆ Centrado en la arquitectura: La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura.
- ◆ Iterativo e Incremental: Aunque la figura 1.1 puede sugerir que los flujos de trabajo se desarrollan en cascada, la “lectura” de este gráfico tiene que ser vertical y horizontal. 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. Por ejemplo, una iteración de elaboración centra su atención en el análisis y diseño, aunque refina los requerimientos y obtiene un producto con un determinado nivel, pero que irá creciendo incrementalmente en cada iteración.

Estas características constituyen los verdaderos aspectos definitorios del Proceso Unificado de Rational.

Lenguaje Unificado de Modelado (UML).

UML (Unified Modeling Language) es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. [22]

Se ha convertido en el estándar de la industria, debido a que ha sido impulsado por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh, quienes fueron contratados por la empresa Rational Software Co. para crear una notación unificada en la cual basar la construcción de sus herramientas CASE. Esta notación incorpora las principales ventajas de cada uno de los métodos particulares en los que se basa (principalmente Booch, OMT y OOSE). Con UML se fusiona la notación de estas técnicas, para formar una herramienta compartida entre todos los ingenieros software que trabajan en el desarrollo orientado a objetos. [23]

El UML esta compuesto por diversos elementos gráficos que se combinan para conformar diagramas, cuya finalidad es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo. Es importante destacar que un modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema. [21]

1.5.10 Herramientas utilizadas.

Para confeccionar la propuesta de este trabajo se hace indispensable tener en cuenta la utilización de algunas herramientas necesarias en el diseño de interfaz, el trabajo con las imágenes y el modelado del proyecto. Seguidamente se hace referencia a las mismas.

1.5.10.1 Macromedia Dreamweaver MX

Macromedia Dreamweaver MX es uno de los editores de desarrollo Web más utilizado a nivel profesional para la creación de sitios Web. Su amplio abanico de herramientas permite crear desde la más simple página Web personal hasta el sitio Web más completo y complejo para una gran empresa, así como realizar aplicaciones que se ejecuten en servidor y vinculaciones dinámicas de datos, con la posibilidad de inclusión de otras tecnologías como son, ASP.NET, PHP y JSP. [10]

Cuenta con un soporte para aplicaciones PHP y utilización de bases MySQL, lo que lo hace muy competente para nuestro caso en particular, y también con un amplio soporte para la creación y utilización de CSS, para lograr un diseño fácil y óptimo.

Este editor de HTML profesional para el diseño, codificación y desarrollo de páginas, sitios y aplicaciones Web; permite la edición visual, o sea, crear páginas rápidamente sin escribir una línea de código, así como también la codificación manual. [10] Además posibilita la creación de una interfaz de usuario integrada, que facilita la interacción dada la potencialidad de sus componentes de trabajo.

1.5.10.2 Zend Studio.

Zend Studio es uno de los ambientes de desarrollo integrado o Integrated Development Environment (IDE) disponible para desarrolladores profesionales que agrupa todos los componentes necesarios para un ciclo de desarrollo de aplicaciones PHP. [10]

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, lo que acelera los ciclos de desarrollo y simplifica los proyectos complejos.

Zend Studio consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor. Las dos partes se instalan por separado, la del cliente contiene el interfaz de edición y la ayuda. Permite además hacer depuraciones simples de scripts, aunque para disfrutar de toda la potencia de la herramienta de depuración habrá que disponer de la parte del servidor, que instala Apache y el módulo PHP o, en caso de que estén instalados, los configura para trabajar juntos en depuración. Existen versiones del producto para Windows, Linux y MacOS. [24]

Los expertos en PHP consideran a Zend Studio como el entorno IDE más maduro y con más características útiles. Si deseamos aumentar la productividad en nuestros desarrollos PHP no cabe duda que este programa puede ayudarnos seriamente. Todas las opciones que dispone están pensadas con acierto por personas que conocen como nadie la tecnología. [24]

1.5.10.3 Rational Rose.

Rational Rose es una herramienta CASE para el desarrollo de aplicaciones informáticas y tiene como base UML.

Actualmente es la herramienta líder en el mundo para el modelado de sistemas complejos y de tiempo real. Integra todos los elementos que propone la metodología RUP para cubrir el ciclo de vida de un proyecto y proporciona mecanismos para realizar Ingeniería Inversa. Además utiliza un proceso de desarrollo iterativo controlado que permite que haya varias personas trabajando a la vez, posibilitando que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo y tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo.

Se decidió que se utilizaría el Rational Rose Enterprise Edition 2003, para sustentar la documentación, como modelador visual de la notación UML (Unified Modeling Language) para la confección de los diagramas que se ilustran en este documento. Es una potente herramienta para el desarrollo de Software.

1.6 Conclusiones.

En este capítulo se ha profundizado en el conocimiento de algunos conceptos necesarios para la comprensión de este trabajo. Además se realizó un análisis completo de las tecnologías actuales adecuadas para llevar a cabo el sistema que se pretende desarrollar, seleccionando y fundamentando dichas elecciones.

Basándonos en los argumentos antes expuestos y dado que a que se hará uso de tecnologías cliente/servidor sobre plataforma Web para implementar este trabajo, se ha escogido al PHP como lenguaje de programación producto de su portabilidad y eficiencia, y como SGBD a MySQL. En esta elección ha tenido un fuerte peso, el hecho de que ambos, tanto PHP como MySQL, son software libre, lo que confirma las ventajas en comparación con otros sistemas, debido a que esta práctica es gratuita y los requerimientos de hardware son relativamente bajos, además la implantación y adquisición en nuestro país del software libre es una de nuestras principales metas. RUP será la metodología que guiará el proceso de desarrollo.

Una vez conocidas las herramientas óptimas, y los conceptos a utilizar se puede empezar a desarrollar la propuesta de sistema.

Capítulo 2

Características del Sistema

2.1 Introducción.

En el presente capítulo se identifican los principales problemas que fundamentan la propuesta de solución, abordando aspectos como el objeto de estudio y el flujo actual del proceso. Se realiza un Modelo del Negocio, para capturar correctamente los requisitos y poder construir un sistema correcto. Además se enumeran los requisitos funcionales y no funcionales que debe tener el sistema propuesto, lo que permite tener una concepción general del sistema, e identificar mediante un Diagrama de Caso de Uso, las relaciones de los actores que interactúan con el sistema, y las secuencias de acciones con las que interactúan.

2.2 Objeto de Estudio.

La Universidad de las Ciencias Informáticas (UCI), basada en su estrategia de desarrollo, está enfrascada en el proyecto UCI-Ciudad Digital cuyo objetivo es crear una micro-sociedad completamente informatizada, que pudiera constituir el modelo de sociedad cubana del futuro. Como parte de esta misión el área de Gestión de servicios realiza diferentes proyectos para la automatización de los servicios que se brindan en el centro.

Justamente, el objeto de estudio de este trabajo lo constituye el proceso de gestión de solicitudes de los servicios telemáticos que son brindados en dicha universidad, proceso que actualmente se realiza de forma manual utilizando para ello una planilla de solicitud, lo que conlleva a una pérdida sustancial de tiempo desde la solicitud hasta la autorización y correspondiente ejecución de los servicios.

2.2.1 Descripción del proceso actual.

Actualmente el proceso de gestión de solicitudes de los servicios telemáticos en la UCI se tramita por medio de una planilla de solicitud que transita por una serie de niveles de autorización, hasta que resulta finalmente aprobada y se le habilita el correspondiente servicio al solicitante.

De manera general, existen 3 niveles de autorización y un nivel para el habilitador del servicio, aunque existen algunos tipos de solicitudes que no requieren pasar por todos los niveles y basta con ser aprobadas en dos de ellos para que la persona facultada para habilitar el servicio proceda. Cada nivel está representado por un directivo responsabilizado con la modificación y aprobación o no de la solicitud, directivo que va a variar según el tipo de solicitud de que se trate.

La siguiente figura pretende esclarecer la relación existente entre el tipo de solicitud, niveles de autorización y directivos por cada nivel.

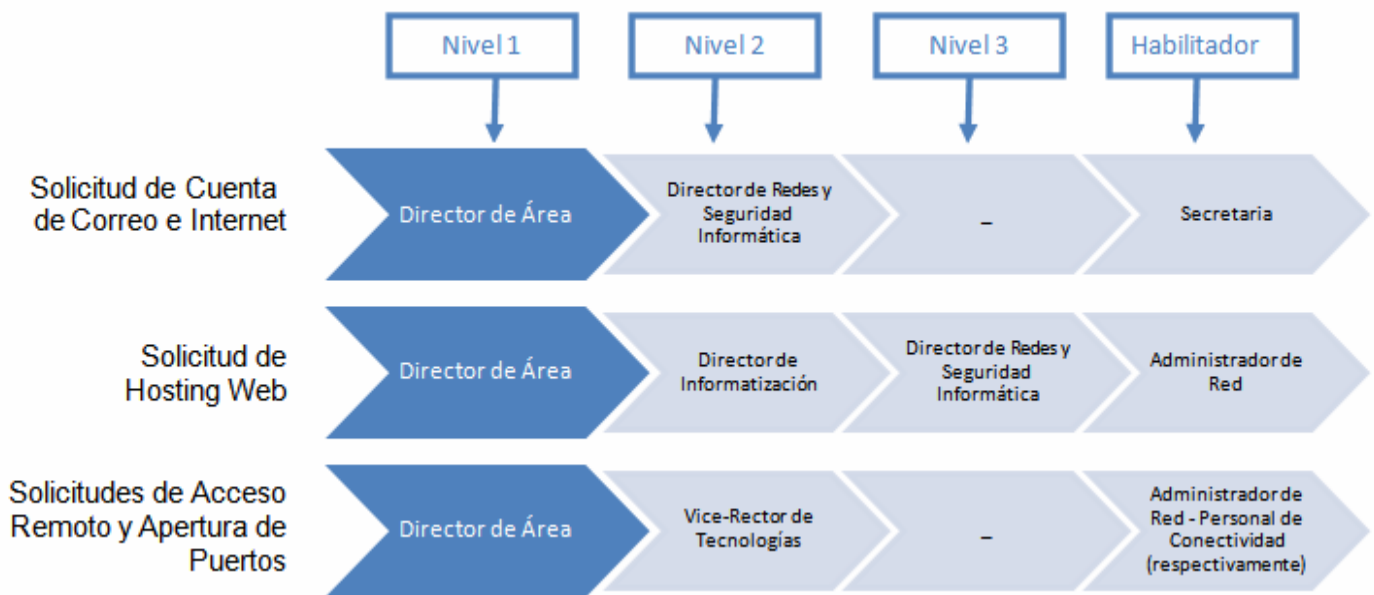


Figura 2.1: Relación tipo solicitud - nivel de autorización y directivos por niveles.

O sea, el trabajador interesado (solo trabajadores, para los estudiantes se ejecuta esta tarea de forma masiva de otro modo) debe llenar, y firmar como constancia de su comprometimiento a cumplir con el código de ética establecido por el centro para la utilización de estos servicios, la planilla que le hará llegar a su Director de Área, que sería el directivo en el primer nivel de autorización en cualquier variante.

Una vez aprobada por el director de Área, en dependencia del tipo de solicitud que sea, la jerarquía de aprobación a la que es sometida varía y la misma circula por los pasos que se describen a continuación: En caso de ser una solicitud de Cuenta de Correo Electrónico e Internet, le corresponde al Director de

Redes y Seguridad Informática realizar la aprobación final o no de la misma, pasando entonces a las manos de la secretaria que habilita el servicio correspondiente y archiva la solicitud.

Si se trata de una solicitud de Apertura de Puertos o de Acceso Remoto, entonces el Vice-Rector de Tecnologías es el responsable de realizar la aprobación o no de la misma antes de llegar a las manos de los administradores de redes o el personal de conectividad, según sea de un tipo o del otro mencionado respectivamente, que son los encargados de habilitar el servicio.

Si por el contrario, es una solicitud de Hosting Web primero es atendida por el Director de Informatización y en caso de que este la apruebe, pasa manos del Director de Redes y Seguridad Informática y luego a los administradores de redes para que procedan a la habilitación del servicio.

Aclarar que cuando la solicitud es firmada por algún director es porque ha sido aprobada por el mismo, en caso de ser anulada se archiva hasta que alguien efectuó alguna reclamación.

2.2.2 Análisis crítico del proceso actual.

El mayor problema en el proceso de gestión de las solicitudes de los servicios telemáticos en la universidad radica en el hecho de que es una labor que se realiza completamente manual, provocando que las actividades de llenado, revisión, búsqueda e incluso el establecimiento del servicio sean sumamente ineficientes, lentas y tediosas. Por ejemplo, el trámite a que es sometida la solicitud hasta llegar a su destino final puede ser demorado por la falta de datos en las mismas o por problemas de legibilidad; ante la reclamación de algún interesado, resulta incómodo realizar la búsqueda de la misma por el gran cúmulo de papel archivado.

Otro problema, es el hecho de que el interesado no es retroalimentado con las gestiones realizadas sobre la solicitud; o sea, durante el trámite de la misma, no es informado en ningún momento acerca de si su solicitud ha sido rechazada en alguno de los niveles de aprobación o en cual de dichos niveles se encuentra, por lo que ante cualquier inconformidad, la persona se tiene que dirigir hasta el lugar donde se archivan todas la solicitudes y realizar su reclamación, desencadenando entonces el tedioso proceso de búsqueda que se mencionaba anteriormente.

Por otro lado, a los directores les resulta imposible verificar que la solicitud que les llega a sus manos haya sido firmada con anterioridad por quien realmente debe efectuar esta acción, como tampoco pueden identificar si el documento ha sido transformado, incluso creado por personal no autorizado quizás suplantando la identidad de otra persona.

Finalmente se puede señalar la imposibilidad que tienen los directivos de controlar estadísticamente el comportamiento que tienen las solicitudes (cantidad de solicitudes realizadas por área, en un determinado período de tiempo, etc.) por lo engorroso que puede resultar este proceso debido al volumen de información acumulada en formato duro.

2.3 Procesos objeto de automatización.

Con el sistema se pretende automatizar los siguientes procesos:

- ◆ Llenado y trámite de las solicitudes.

Los interesados en obtener algún servicio telemático llenan una planilla de solicitud que transita por una jerarquía de aprobación hasta que el servicio solicitado pueda ser habilitado. El sistema que se pretende implementar tratará de agilizar y amenizar este proceso.

- ◆ Firmado de las solicitudes.

Durante todo el proceso que sigue la solicitud las personas autorizadas deben firmar la misma como constancia de que ha sido aprobada. Con la propuesta de este trabajo, se trataría de realizar esta acción por medio de la tecnología de firma digital, garantizando de esta forma la autenticidad de la persona que ha firmado y la integridad del documento.

- ◆ Retroalimentar al interesado con las gestiones realizadas.

La persona que llena una solicitud no es notificada en ningún momento del estado en que se encuentra su solicitud. A través del software que se pretende realizar, se trataría de que los solicitantes puedan acceder por medio de una búsqueda, siguiendo determinado criterio, a su solicitud y verificar el estado en el que se encuentra la misma.

- ◆ Control y estadísticas de las solicitudes.

No es posible llevar un control estadístico certero de las solicitudes por la gran cantidad de información acumulada en papel, lo que propicia la generación de errores y demora en la realización de la tarea. El

sistema propuesto trataría de resolver estos problemas permitiendo mostrar estadísticas de las solicitudes por tipo, por área, por fecha y por estado fundamentalmente.

2.3.1 Información que se maneja.

La solicitud constituye el documento en el que se basan todos los procesos descritos anteriormente. Por contener información sensible y fácilmente alterable como los datos personales del solicitante y datos del servicio solicitado, se debe lograr, y es uno de los objetivos que persigue nuestra propuesta, que la manipulación de la misma sea de una manera segura y confiable, que no atente contra la autenticidad e integridad de la misma.

2.4 Modelo del negocio.

A fin de lograr una mejor comprensión del problema a resolver y el común entendimiento entre clientes y desarrolladores se realiza la modelación del negocio, lo que posibilita obtener una visión más clara del proceso que nos ocupa en cuestión.

El modelo del negocio describe los procesos de negocio, en términos de casos de usos del negocio, identificando quiénes participan y las actividades que requieren automatización.

2.4.1 Actores del negocio.

Un actor del negocio es cualquier individuo, grupo, organización, máquina o sistema de información externo que interactúa con el negocio. El término *actor* significa el rol que algo o alguien juega cuando interactúa con el negocio para beneficiarse de sus resultados. [25]

A continuación se muestra los actores correspondientes a nuestro negocio con su respectiva justificación:

Actores del negocio	Justificación
Solicitante	Es un trabajador de la UCI interesado en obtener algún servicio telemático, por eso llena la planilla de solicitud correspondiente. Es el que inicia todas las acciones que dan comienzo a los procesos de negocio analizados, y al mismo tiempo es el principal beneficiado con el resultado de dichos procesos de negocio.

Tabla 2.1: Justificación de los actores del negocio.

2.4.2 Trabajadores del negocio.

Un trabajador del negocio es una abstracción de una persona (o grupo de personas), una máquina o un sistema automatizado; que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores del negocio y manipulando entidades del negocio. Representa un rol.

A continuación se muestran los trabajadores del negocio y su correspondiente justificación:

Trabajadores del negocio	Justificación
Directivo _ Nivel 1_Autorización	Persona encargada de revisar y aprobar(o rechazar) cualquier tipo de solicitud realizada por los trabajadores de su área. (Jefe de Área)
Directivo _ Nivel 2_Autorización	Persona que recibe todas las solicitudes que han sido aprobadas por Directivo _ Nivel 1_Autorización (los diferentes Directores de Áreas). Revisa, modifica, aprueba o no dichas solicitudes.
Directivo _ Nivel 3_Autorización	Persona que recibe todas las solicitudes que han sido aprobadas por Directivo _ Nivel 2_Autorización. Revisa, modifica, aprueba o no dichas solicitudes.
Habilitador _ Servicio	Persona facultada para habilitar el servicio solicitado luego de haber sido aprobada la solicitud por todos los directivos responsabilizados de esta actividad en cada una de los niveles.

Tabla 2.2: Justificación de los trabajadores del negocio.

2.4.3 Diagrama de casos de uso del negocio.

El diagrama de casos de uso del negocio representa gráficamente los procesos del negocio y su interacción con los actores del negocio. A continuación se muestra la figura 2.1 correspondiente al diagrama de casos de uso del negocio.

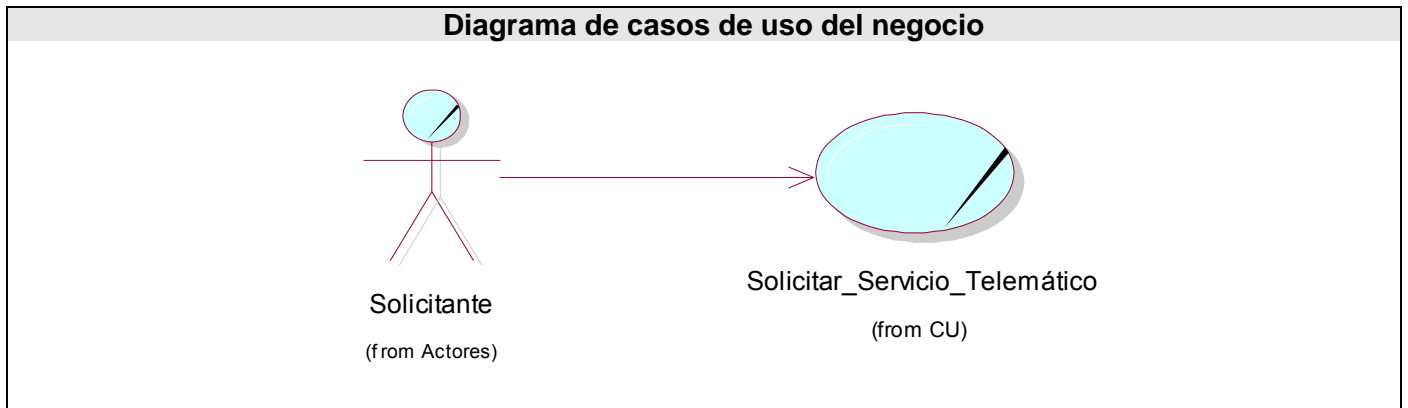


Figura 2.2: Diagrama de CUN

2.4.4 Descripción de los casos de uso del negocio.

Un caso de uso del negocio representa a un proceso de negocio, por lo que se corresponde con una secuencia de acciones que producen un resultado observable para ciertos actores del negocio. Desde la perspectiva de un actor individual, define un flujo de trabajo completo que produce resultados deseables. Para el caso de uso del negocio se tendrá una especificación textual con su correspondiente diagrama de actividades y el diagrama de clases del Modelo de Objeto.

2.4.4.1 Especificación textual del Caso de Uso “Solicitar Servicio Telemático”.

Nombre del caso de uso del negocio:	Solicitar servicio telemático.
Actores del negocio:	Solicitante.
Trabajadores del negocio:	Directivo_Nivel1_Autorización, Directivo_Nivel2_Autorización, Directivo_Nivel3_Autorización, Habilitador _ Servicio.
Resumen: El caso de uso se inicia cuando un solicitante llena una planilla para solicitar algún servicio. La solicitud va pasando por los directivos en los niveles de autorización requeridos para ser aprobada o no, hasta llegar a manos del habilitador del servicio quien procede a implementar el mismo, dándole culminación al caso de uso.	
Curso normal de los eventos	
Acción del actor	Respuesta del negocio
1. El solicitante llena la planilla de solicitud.	

2. Entrega dicha solicitud firmada a su Director de Área (Directivo_Nivel1_Autorización)	3. Directivo_Nivel1_Autorización (Director de Área) revisa contenido de la solicitud.
	4. Directivo_Nivel1_Autorización (Director de Área) aprueba la solicitud.
	5. Directivo_Nivel1_Autorización (Director de Área) firma la solicitud.
	6. Directivo_Nivel2_Autorización recibe la solicitud firmada por el Directivo_Nivel1_Autorización (Director de Área) y la revisa.
	7. Directivo_Nivel2_Autorización aprueba la solicitud.
	8. Directivo_Nivel2_Autorización firma la solicitud.
	9. La solicitud no requiere de otro nivel de autorización y pasa a manos del habilitador del servicio.
	10. Habilitador _ Servicio habilita el servicio correspondiente.
	11. Habilitador _ Servicio archiva la solicitud final.
Curso alternativo de los eventos	
Acción del actor	Respuesta del negocio
Línea 4.	4. Directivo_Nivel1_Autorización (Director de Área) rechaza la solicitud. 4.1 Archiva solicitud rechazada.
Línea 7.	7. Directivo_Nivel2_Autorización rechaza la solicitud. 7.1 Archiva solicitud rechazada.
Línea 9.	9. La solicitud requiere de otro nivel de autorización. Ver sección "3r. Nivel".
Curso alternativo de los eventos	
Acción del actor	Respuesta del negocio
Línea 4.	4. Directivo_Nivel1_Autorización (Director de Área) modifica contenido de la solicitud. Retornar a paso 5.
Línea 7.	7. Directivo_Nivel2_Autorización modifica contenido de la

	solicitud. Retornar a paso 8.
Sección “3r. Nivel”	
Curso normal de los eventos	
Acción del actor	Respuesta del negocio
	1. Directivo_Nivel3_Autorización recibe la solicitud firmada por el Directivo_Nivel2_Autorización y la revisa.
	2. Directivo_Nivel3_Autorización aprueba la solicitud.
	3. Directivo_Nivel3_Autorización firma la solicitud.
	4. Habilitador _ Servicio habilita el servicio correspondiente.
	5. Habilitador _ Servicio archiva la solicitud final.
Curso alternativo de los eventos	
Acción del actor	Respuesta del negocio
Línea 2.	2. Directivo_Nivel3_Autorización (Director de Área) rechaza la solicitud. 2.1 Archiva solicitud rechazada.
Curso alternativo de los eventos	
Acción del actor	Respuesta del negocio
Línea 2.	2. Directivo_Nivel3_Autorización modifica contenido de la solicitud. Retornar a paso 3.
Prioridad:	Crítico

Tabla 2.3: Descripción textual del CU Solicitar Servicio Telemático.

2.4.4.2 Diagrama de Actividades del Caso de Uso “Solicitar Servicio Telemático”.

Un diagrama de actividad describe el detalle de qué es lo que pasa dentro del negocio, se examinan los roles específicos que juegan los trabajadores del negocio y las actividades que realizan. Además ayuda a identificar qué funciones deberá asumir el producto de software, y quiénes serán los actores del futuro sistema.

A continuación se muestra el Diagrama de actividades del caso de uso del negocio “Solicitar Servicio Telemático”.

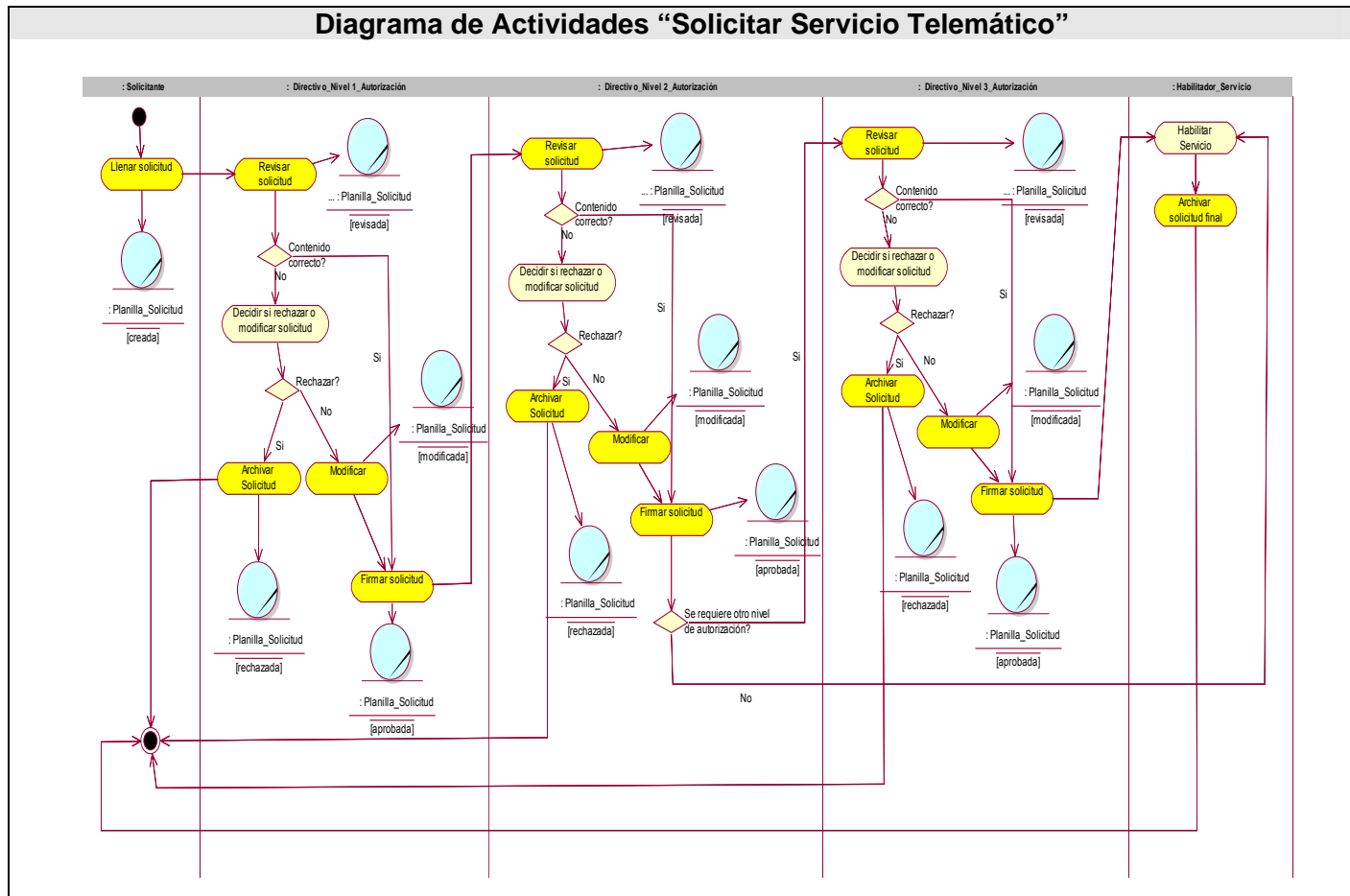


Figura 2.3: Diagrama de Actividades CU Solicitar Servicio.

2.4.4.3 Diagrama de Clases del Modelo de Objetos del Caso de Uso "Solicitar Servicio Telemático".

Un modelo de objetos del negocio es un modelo interno a un negocio. Describe cómo colaboran los trabajadores y entidades del negocio dentro del flujo de trabajo del proceso del negocio.

A continuación se muestra el Diagrama de Clases del Modelo de Objetos correspondiente a nuestro negocio.

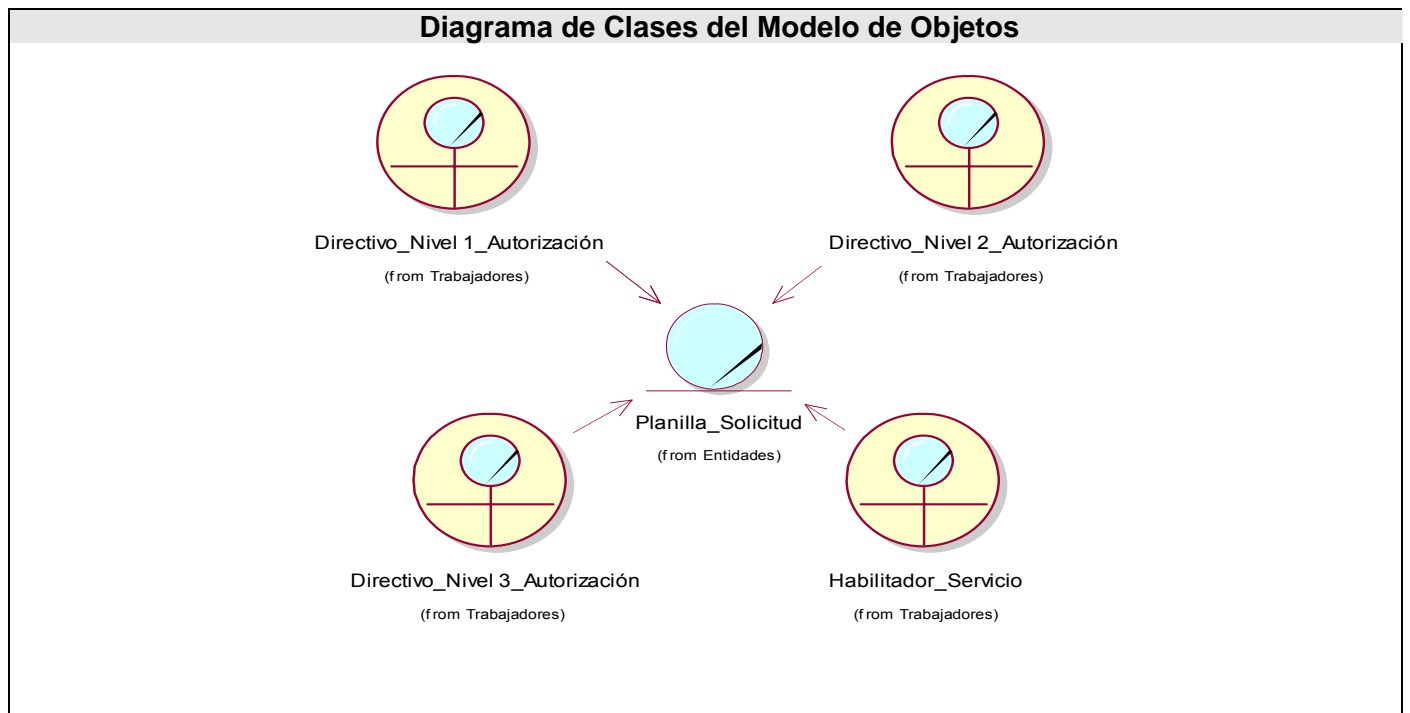


Figura 2.4: Modelo de Objetos.

2.5 Especificación de los requisitos de software.

Los requisitos son las condiciones o capacidades que tienen que ser alcanzadas o poseídas por el sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente. O sea, los requisitos son los que definen qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.

2.5.1 Requerimientos Funcionales.

Los requerimientos funcionales son aquellos requisitos que, desde el punto de vista de las necesidades del usuario, debe cumplir el sistema y que están fuertemente ligados a las opciones del programa. Para cumplir con los objetivos propuestos se prevé que el sistema tenga las siguientes funcionalidades:

R1. Solicitar servicio telemático.

- 1.1. Validar los datos del solicitante.
- 1.2. Llenar planilla de solicitud.

R2. Gestionar solicitud.

- 2.1. Modificar solicitud.
- 2.2. Firmar solicitud.
- 2.3. Aprobar o rechazar solicitud.
- 2.4. Imprimir solicitud.

R3. Generar reportes.

- 3.1. Realizar búsqueda para solicitante.
- 3.2. Mostrar estadísticas.
 - 3.2.1. Solicitudes por tipos.
 - 3.2.2. Solicitudes por área.
 - 3.2.3. Solicitudes por fecha.
 - 3.2.5. Solicitudes por directivo.
- 3.3. Realizar búsqueda avanzada para directivos.

R4. Gestionar Servicio.

- 4.1. Adicionar servicio.
 - 4.1.1. Crear datos del servicio (de la nueva planilla).
 - 4.1.2. Crear línea de aprobación de la solicitud.
- 4.2. Modificar servicio.
 - 4.2.1. Modificar datos del servicio (de la nueva planilla).
 - 4.2.2. Modificar línea de aprobación de la solicitud.

R5. Gestionar Usuario.

- 5.1. Adicionar usuario.
- 5.2. Modificar usuario.
- 5.3. Eliminar o deshabilitar usuario según privilegio.

R6. Autenticar usuario.

- 6.1. Pedir nombre de usuario y contraseña para las opciones del sistema que lo requieran.
- 6.2. Mandar directamente al usuario registrado a las opciones principales con que cuenta su nivel registrado.
- 6.3. Permitir al usuario registrado cambiar su passphrase.
- 6.4. Permitir que el usuario registrado cierre su sesión de trabajo desde cualquier lugar del sistema.

2.5.2 Requerimientos no Funcionales.

Los requerimientos no funcionales son características que describen alguna forma o restricción para la realización de algún requerimiento (funcionalidad) o conjunto de ellas e inclusive todos los requerimientos. Se consideran los atributos del sistema, propiedades que debe tener el producto. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. A continuación se muestran los requerimientos no funcionales:

Apariencia o interfaz externa

- ◆ Interfaz con pocas imágenes para no demorar las respuestas al usuario.
- ◆ Diseño sencillo y claro, con reconocimiento visual a través de elementos visibles que identifiquen cada una de sus funcionalidades.
- ◆ Formal, serio y con una navegación sugerente, teniendo en cuenta el fin con el que se desarrolla la aplicación.

Usabilidad

- ◆ El sistema puede ser usado por cualquier persona, que posea conocimientos básicos en el manejo de la computadora y de un ambiente Web en sentido general.
- ◆ Instalar el sistema trae consigo una mayor rapidez de trabajo y por consiguiente un ahorro de materiales y personal.

Rendimiento

- ◆ Disponibilidad constante de trabajo en red contra el servidor.
- ◆ Rápido acceso de búsqueda a la información en tiempos relativamente cortos.
- ◆ El sistema deberá de ser lo más estable y confiable posible.

Soporte

- ◆ Se requiere que el producto reciba mantenimiento ante cualquier fallo que ocurra.

Restricciones en el diseño y la implementación

- ◆ La aplicación se implementará utilizando el lenguaje de programación PHP, Servidor Web Apache y como Gestor de Bases de Datos MySQL Server.

Software

- ◆ Para el funcionamiento del sistema en el servidor es necesario el S.O. Windows 98 o superior, Linux o Unix, en sus versiones de S.O. servidores.
- ◆ Para el funcionamiento del sistema en las terminales cliente es necesario el S.O. Windows 95 o superior, Linux o Unix.

Hardware

- ◆ Se necesitan como requerimientos mínimos una PC con procesador Pentium II o superior.

Portabilidad

- ◆ El software podrá ser usado bajo los sistemas operativos Windows y Linux.

Seguridad

- ◆ El sistema se encarga de controlar los diferentes niveles de acceso y roles de usuarios, garantiza que la información sea vista únicamente por quien tiene derecho a verla.
- ◆ Se hacen validaciones de la información tanto en el cliente como en el servidor, no obstante los usuarios acceden de manera rápida y operativa al sistema sin que los requerimientos de seguridad se conviertan en un retardo para ellos.

Confidencialidad

- ◆ Toda la información está protegida del acceso no autorizado.

Disponibilidad

- ◆ Se garantiza a los usuarios del sistema el acceso a la información solicitada en todo momento (si tiene permiso para ello).

Confiabilidad

- ◆ La herramienta de implementación a utilizar tiene soporte para recuperación ante fallos y errores.

Legales

- ◆ La plataforma escogida para el desarrollo de la aplicación, está basada en la licencia GNU/GPL.

2.6 Descripción del Sistema propuesto.

Para cumplimentar los objetivos propuestos al inicio de este trabajo, y teniendo en cuenta todos los requerimientos planteados, el sistema que se propone debe tener cinco módulos principales: login, solicitudes, gestión de solicitudes, reportes y administración del sistema.

Para utilizar todas las funcionalidades del sistema, se considera la existencia de 3 roles; Solicitante, Directivo y Administrador; o sea, un usuario se puede comportar como uno de estos roles. En el SGBD que se utilizará se controlarán los datos y el rol de cada usuario que exista.

El módulo de login será usado obligatoriamente por los directivos y los administradores para acceder a sus secciones de trabajo. De igual modo, las personas que juegan estos roles serán los autorizados a acceder al módulo de reportes donde podrán consultar un grupo de estadísticas relacionadas con el comportamiento de las solicitudes y efectuar también una búsqueda avanzada en caso de querer comprobar algún procedimiento específico sobre las mismas. Se especializarán las búsquedas siguiendo varios criterios, como lo hacen algunos buscadores, y se podrán filtrar los resultados para obtener respuestas más específicas.

El módulo de las solicitudes se va a encontrar visible en todo momento, puede ser utilizado por el rol Solicitante para llenar alguna planilla de solicitud y para acceder a un reporte personalizado donde se listan las solicitudes efectuadas por él, posibilitándole mantenerse informado en todo momento del estado en el que se encuentra su solicitud (si se encuentra pendiente por aprobación o ha sido rechazada por un determinado directivo).

El módulo de gestión de solicitudes solo puede ser usado por los Directivos y comprende un proceso completo de los trámites a los que es sometida la solicitud. O sea, cada directivo tendrá acceso a las planillas en el debido momento para revisarlas, realizarle modificaciones si así lo estima necesario, aprobarlas o no, y firmarla con la tecnología de firma digital como constancia de las gestiones efectuadas por él. El SGBD contendrá la estructura apropiada para cumplir con la línea de aprobación que deben seguir las solicitudes.

Finalmente el módulo de administración al que solo podrá acceder el Administrador, controlará opciones de configuración; por ejemplo la inserción de un nuevo servicio (una nueva planilla de solicitud para un nuevo servicio) con su correspondiente línea de aprobación, o la modificación de uno ya existente. También tendrá el control de todos los usuarios, por lo que puede crear cualquier rol nuevo en el sistema o modificar los permisos de uno existente.

En resumen, con este sistema se podrá gestionar las solicitudes de servicios telemáticos de manera ágil y eficiente, garantizando que la persona interesada se pueda informar en todo momento del estado en el que se encuentra su solicitud y la seguridad del proceso.

2.7 Modelo de casos de usos del sistema.

Utilizando las facilidades que brinda el UML, se capturan los requisitos funcionales del sistema y se representan mediante un diagrama de casos de uso. Para ello se define de acuerdo a lo planteado en los epígrafes anteriores, cuales serían los actores que van a interactuar con el sistema, y los casos de uso que van a representar las funcionalidades.

2.7.1 Definición de los actores.

Un actor no es parte del sistema, es un rol de un usuario, que puede intercambiar información o puede ser un recipiente pasivo de información y representa a un ser humano, a un software o a una máquina que interactúa con el sistema. Con el sistema interactúan seis actores que se definen a continuación:

Actores	Justificación
Usuario	Representa un actor generalizado que puede comportarse como Solicitante, Directivo o Administrador.
Solicitante	Representa a una persona que va a utilizar el sistema para llenar una solicitud de un determinado servicio telemático. Cuando lo estime, puede realizar una búsqueda de su solicitud para ver el estado en el que se encuentra la misma.
Directivo	Representa a una persona autorizada a realizar todas las acciones sobre las solicitudes (modificarlas, aprobarlas o rechazarlas, firmarlas). Puede realizar búsquedas y consultar todo tipo de estadísticas referente a las solicitudes.
Administrador	Representa a una persona que configura y controla el comportamiento del sistema. Encargado de adicionar, modificar, eliminar servicios y usuarios en el sistema. También puede consultar los reportes de las solicitudes.

Tabla 2.4: Justificación de los actores del sistema.

2.7.2 Listado de casos de uso.

A continuación se presentan los casos de uso determinados para satisfacer los requerimientos funcionales de sistema:

CU-1	Solicitar Servicio Telemático.
Actor	Solicitante.
Descripción	El solicitante llena un formulario correspondiente a una determinada solicitud y la envía.
Referencia	R1.

CU-2	Gestionar Solicitud.
Actor	Directivo.
Descripción	Los directivos en los diferentes niveles de aprobación modifican, aprueban y firman la solicitud.
Referencia	R2.

CU-3	Generar Reporte.
Actor	Usuario.
Descripción	Un Usuario (específicamente un solicitante) puede realizar una búsqueda para conocer el estado en el que se encuentra su solicitud. Un Usuario (directivo o administrador) puede consultar las estadísticas del comportamiento de las solicitudes y realizar una búsqueda avanzada para verificar el procedimiento específico seguido con las solicitudes.
Referencia	R3.

CU-4	Gestionar Servicios.
Actor	Administrador.
Descripción	El Administrador adiciona un nuevo servicio, o sea una nueva planilla de solicitud para un nuevo servicio con su línea de aprobación, el tiempo de validez de la misma, etc. Además puede modificar un servicio (planilla de solicitud) existente.
Referencia	R4.

CU-5	Gestionar Usuarios.
Actor	Administrador.
Descripción	El Administrador puede adicionar un nuevo usuario con sus respectivos privilegios, modificar o eliminar un usuario.
Referencia	R5.

CU-6	Autenticar Usuario.
Actor	Usuario.
Descripción	El usuario proporciona el login y password definidos para entrar al sistema. El sistema verifica y si es válido accede directamente al módulo permisible.
Referencia	R6.

2.7.3 Diagrama de casos de uso.

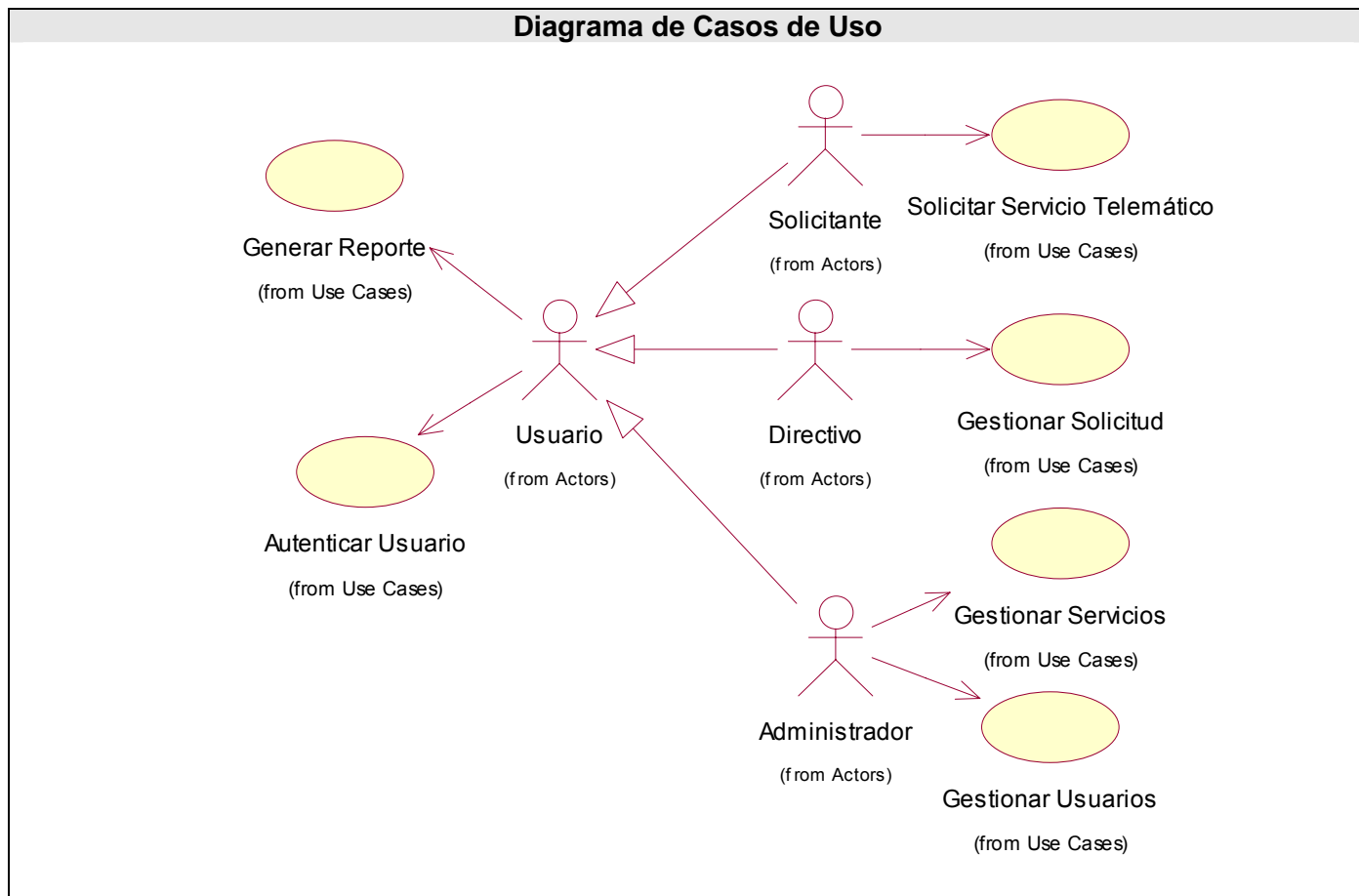


Figura 2.5: Diagrama de Casos de Uso del Sistema.

2.7.4 Casos de uso expandidos.

Mediante los casos de uso expandidos se describe paso a paso la secuencia de eventos que los actores utilizan para completar un proceso a través del sistema. Este sería el último paso en el análisis de los procesos, para pasar a la construcción de la solución propuesta.

Caso de uso	
CU-1	Solicitar Servicio Telemático.
Propósito	Permitir al Solicitante el llenado y envío de una planilla de solicitud de un determinado servicio telemático.
Actores	Solicitante (inicia).
Precondiciones	El tipo de solicitud que se desea debe estar activa.

Poscondiciones	Se crea una nueva instancia de la clase solicitud.	
Resumen: El CU se inicia cuando un Solicitante accede al sistema y llena y envía una determinada solicitud, concluyendo el caso de uso.		
Referencias	R1.	
Curso normal de los eventos.		
Acción del actor	Respuesta del sistema	
1. El solicitante selecciona la solicitud a llenar e introduce su No. Cl.	2. El sistema valida los datos entrados por el solicitante.	
	3. El sistema archiva la solicitud para pendiente aprobación.	
Curso Alterno de los eventos.		
Línea 2. Datos incompletos o incorrectos.	2. El sistema emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos. Retorna al paso 3.	
Prioridad	Crítico.	
Prototipo	Ver Anexo 2.	

Caso de uso		
CU-2	Gestionar solicitud.	
Propósito	Permitir a los directivos realizar todo el trámite de una solicitud.	
Actores	Directivo (inicia).	
Precondiciones	Para gestionar una solicitud debe existir la instancia de la clase solicitud que va a ser tramitada.	
Poscondiciones	Se actualiza el estado de la solicitud.	
Resumen: El CU se inicia cuando un directivo accede al sistema para aprobar o rechazar una nueva solicitud, quien a su vez puede realizarle modificaciones y debe firmarla. Cada vez que la solicitud pasa por manos de un directivo es archivada en el sistema con el estado correspondiente.		
Referencias	R2.	
Curso normal de los eventos.		
Acción del actor	Respuesta del sistema	
1. El directivo accede a su sesión de trabajo.	2. El sistema muestra un listado con las solicitudes pendientes por aprobación.	
3. El directivo selecciona una solicitud.	4. El sistema muestra los datos de la solicitud con los campos que puedan ser cambiados como regiones editables.	
5. El directivo revisa la solicitud y la modifica si lo desea.		
6. El directivo aprueba la solicitud.		
7. El directivo introduce su passphrase.	8. El sistema valida dato de entrada.	
9. El directivo presiona el botón firmar.	10. El sistema realiza un resumen de la solicitud y crea la firma.	
	11. El sistema archiva la solicitud con el estado de	

	aprobada, el resumen de la misma, la firma y la fecha de aprobación.
	12. El sistema envía correo de notificación a próximo directivo en línea de aprobación.
Curso Alterno de los Eventos.	
Línea 6. Solicitud rechazada. 6.1 El directivo rechaza la solicitud. Retornar al paso 7. Línea 11.	11.1. El sistema archiva la solicitud con el estado de rechazada y la firma.
Línea 8. Dato incorrecto.	8.1. El sistema le muestra al Directivo mensaje "passphrase erróneo".
	8.2. El sistema cancela todas las acciones hechas hasta el momento sobre la solicitud. Retornar al paso 6.
Prioridad	Crítico.
Prototipo	Ver Anexo 3.

Caso de uso	
CU-3	Generar Reporte.
Propósito	Permitir al Usuario (solicitante) verificar el estado en el que se encuentra su solicitud, permitir también (a los directivos y administradores) ver el comportamiento estadístico de las solicitudes y realizar una búsqueda avanzada para comprobar el procedimiento específico seguido por las mismas.
Actores	Usuario (inicia).
Precondiciones	-
Poscondiciones	Se muestra el reporte solicitado.
Resumen: El CU se inicia cuando un Usuario (Solicitante) accede al sistema para realizar la búsqueda de una solicitud siguiendo determinados criterios, mostrándose la misma con el estado en el que se encuentra en ese momento; o cuando un usuario (Directivo) decide consultar los comportamientos estadísticos que han tenido las solicitudes o realizar una búsqueda avanzada para verificar el procedimiento seguido con las solicitudes. Finaliza el CU cuando el Usuario ha realizado alguna de estas acciones.	
Referencias	R3.
Sección "Realizar Búsqueda para Solicitante"	
Curso normal de los eventos.	
Acción del actor	Respuesta del sistema
1. El Usuario selecciona la opción de ver sus solicitudes e introduce su No. CI.	2. El sistema muestra un listado con las solicitudes realizadas por el mismo.
3. El Usuario selecciona una solicitud.	4. El sistema muestra el estado de la solicitud por cada uno de los directivos que ha transitado la misma.
Curso Alterno de los eventos.	

Línea 2. Solicitud inexistente.	2. El sistema muestra un mensaje de error especificando que la solicitud no existe.
Sección “Consultar Estadísticas”	
Curso normal de los eventos.	
Acción del actor	Respuesta del sistema
1. El Usuario selecciona un criterio según las estadísticas que desea consultar. <ul style="list-style-type: none"> ◆ Comportamiento por Área. ◆ Comportamiento por Tipo de Solicitud. ◆ Comportamiento por Fechas. ◆ Comportamiento por Estado. ◆ Comportamiento por Directivo. 	2. El sistema muestra el reporte correspondiente.
Sección “Realizar Búsqueda para Directivo”	
Curso normal de los eventos.	
Acción del actor	Acción del actor
1. El Usuario selecciona los criterios para la búsqueda.	2. El sistema muestra el resultado de la búsqueda en un listado.
3. El usuario selecciona una solicitud.	4. El sistema muestra datos de la solicitud.
Prioridad	Crítico.
Prototipo	Ver Anexo 4.

Caso de uso	
CU-4	Gestionar Servicio.
Propósito	Permitirle al administrador insertar una nueva solicitud para un nuevo servicio con su correspondiente línea de aprobación y el tiempo de validez de la misma. También que pueda modificar las plantillas existentes.
Actores	Administrador (inicia).
Precondiciones	Para modificar un servicio debe existir la instancia de la clase tipo _solicitud.
Poscondiciones	Para los siguientes procesos: Insertar: se crea una instancia de la clase tipo _solicitud. Modificar: se modifican los atributos de la instancia seleccionada de la clase tipo _solicitud.
Resumen: El CU se inicia cuando el Administrador del sistema accede al mismo para realizar alguna gestión sobre los servicios, dígame crear un nuevo servicio (una nueva planilla de solicitud para un nuevo servicio) con sus datos y su correspondiente línea de aprobación, modificar los datos o la línea de aprobación de un servicio existente, finalizando el caso de uso con la realización de alguna de estas acciones.	
Referencias	R4.
Curso normal de los eventos.	
Acción del actor	Respuesta del sistema
1. El Administrador selecciona la acción a realizar.	2. a) Si elige opción adicionar servicio ver sección “Adicionar Servicio”.

	b) Si elige la opción de listar los servicios existentes ver sección "Listar Servicios".
Sección "Adicionar Servicio"	
2a 1. El administrador crea un nuevo tipo de solicitud introduciendo los datos del nuevo servicio y creando la línea de aprobación. Selecciona la opción Aceptar.	2a 2. El sistema muestra el listado de los servicios actualizado. Retorna al paso 2.
Curso alterno de los eventos	
Línea 2a 1. Operación Cancelada.	
2a 1.1. El Administrador introduce los datos del nuevo servicio, crea línea de aprobación y selecciona la opción Cancelar.	2a 1.2. El sistema cancela la operación y retorna al paso 2.
Sección "Listar Servicios"	
	2b 1. El sistema muestra el listado de los servicios con dos posibles acciones a realizar sobre cada uno de ellos: modificar datos o modificar línea de aprobación.
2b 2. El Directivo selecciona acción a realizar.	2b 3. 3.1) Si elige opción modificar datos ver sección "Modificar Datos". 3.2) Si elige opción modificar línea de aprobación ver sección "Modificar línea de aprobación".
Sección "Modificar Datos"	
	2b 3.1. El sistema muestra los atributos correspondientes a la plantilla de solicitud seleccionada por el Administrador.
2b 3.1.1. El Administrador realiza las modificaciones pertinentes y selecciona la opción Aceptar.	2b 3.1.2. El sistema actualiza los cambios. Retoma al paso 2b 1
Curso alterno de los eventos	
Línea 2b 3.1.1 Operación Cancelada.	
2b 3.1.1.1 El Administrador realiza las modificaciones pertinentes y selecciona la opción Cancelar.	2b 3.1.1.2 El sistema cancela la operación de modificar.
Sección "Modificar línea de aprobación"	
	2b 3.2. El sistema muestra los directivos.
2b 3.2.1 El Administrador selecciona los directivos para cada nivel y acepta.	2b 3.2.2 El sistema actualiza los cambios en la línea de aprobación. Retorna al paso 2b 1.
Curso alterno de los eventos	
Línea 2b 3.2.1. Operación Cancelada.	
2b 3.2.1.1 El Administrador selecciona la opción Cancelar.	2b 3.2.1.2 El sistema cancela la operación de eliminar. Retorna al paso 2b 1.
Prioridad	Crítico.
Prototipo	Ver Anexo 5.

Caso de uso	
CU-5	Gestionar Usuario.
Propósito	Permitir al Administrador mantener el control sobre los usuarios del sistema con sus respectivos permisos.
Actores	Administrador (inicia).
Precondiciones	Para modificar o eliminar un servicio debe existir la instancia de la clase tipo _ solicitud.
Poscondiciones	Para los siguientes procesos: Insertar: se crea una instancia de la clase usuario. Modificar: se modifican los atributos de la instancia seleccionada de la clase usuario.
Resumen: El CU se inicia cuando el Administrador del sistema accede al mismo para realizar alguna gestión sobre los usuarios, dígase crear un nuevo usuario, eliminar o deshabilitar un usuario existente según sus privilegios o modificarle sus permisos, finalizando el caso de uso con la realización de alguna de estas acciones.	
Referencias	R5.
Curso normal de los eventos.	
Acción del actor	Respuesta del sistema
1. El Administrador selecciona la acción a realizar.	2. a) Si elige opción adicionar usuario ver sección "Adicionar Usuario". b) Si elige la opción de listar los usuarios existentes ver sección "Listar Usuarios".
Sección "Adicionar Usuario"	
2a 1. El administrador introduce los datos del nuevo usuario y selecciona la opción Aceptar.	2a 2. El sistema valida los datos de entrada.
	2a 3. El sistema muestra el listado de usuarios actualizado. Retorna al paso 2.
Curso alterno de los eventos	
Línea 2a 2. Operación Cancelada.	
2a 1.1. El Administrador introduce los datos del nuevo usuario y selecciona la opción Cancelar.	2a 1.2. El sistema cancela la operación y retorna al paso 2.
Línea 2a 2. Datos incorrectos o incompletos.	2a 2.1. El sistema emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos. Retorna al paso 2a 1.
Sección "Listar Usuarios"	
	2b 1. El sistema muestra el listado de los usuarios con dos posibles acciones a realizar sobre cada uno de ellos: modificar o, eliminar o deshabilitar según los privilegios del usuario.
2b 2. El Directivo selecciona acción a realizar.	2b 3. 3.1) Si elige opción modificar ver sección "Modificar".

	3.2) Si elige opción eliminar ver sección “Eliminar Usuario”. (En caso de que sea la opción deshabilitar sencillamente se continua mostrando el este usuario en el listado pero está deshabilitado, se considera que por la sencillez de esta opción no es necesario describir una sección.
Sección “Modificar Permisos”	
	2b 3.1. El sistema muestra los permisos correspondientes al usuario seleccionado por el Administrador.
2b 3.1.1. El Administrador realiza las modificaciones pertinentes y selecciona la opción Aceptar.	2b 3.1.2. El sistema actualiza los cambios. Retoma al paso 2b 1
Curso alterno de los eventos	
Línea 2b 3.1.1 Operación Cancelada.	
2b 3.1.1.1 El Administrador realiza las modificaciones pertinentes y selecciona la opción Cancelar.	2b 3.1.1.2 El sistema cancela la operación de modificar permisos. Retorna al paso 2b 1.
Sección “Eliminar Usuario”	
	2b 3.2. El sistema muestra un cartel de confirmación para la eliminación.
2b 3.2.1 El Administrador selecciona la opción Aceptar.	2b 3.2.2 El sistema muestra el listado de usuarios actualizado. Retorna al paso 2b 1.
Curso alterno de los eventos	
Línea 2b 3.2.1. Operación Cancelada.	
2b 3.2.1.1 El Administrador selecciona la opción Cancelar.	2b 3.2.1.2 El sistema cancela la operación de eliminar. Retorna al paso 2b 1.
Prioridad	Crítico.
Prototipo	Ver Anexo 6.

Caso de uso	
CU-7	Autenticar Usuario.
Propósito	Permitir a los usuarios el acceso al sistema según los permisos establecidos para cada uno.
Actores	Usuario (inicia).
Precondiciones	-
Poscondiciones	El usuario accede a un área restringida.
Resumen: El CU se inicia cuando el Usuario introduce su usuario y contraseña para acceder a módulos restringidos del sistema. El sistema comprueba los datos de entrada y le permite el acceso al usuario según sus privilegios, finalizando el CU.	
Referencias	R7.
Curso normal de los eventos.	

Acción del actor	Respuesta del sistema
1. El Usuario introduce su usuario y contraseña.	2. El sistema valida los datos entrados por el solicitante contra Ldap.
	3. El sistema verifica la existencia de usuario en su BD.
	4. El sistema permite el acceso del Usuario Registrado según los permisos que le correspondan.
Curso Alterno de los eventos.	
Línea 2. Datos incompletos o incorrectos.	2.1. El sistema emite un mensaje de error. Retorna a paso 1.
Línea 3. Usuario inexistente en BD del sistema.	3.1. El sistema no permite el acceso del usuario.
Prioridad	Crítico.
Prototipo	Ver Anexo 7.

2.8 Conclusiones.

En este capítulo se comenzó a desarrollar la propuesta de solución, obteniéndose a partir del análisis de los procesos del negocio, un listado con las funciones que debe tener el sistema, que se representaron mediante un Diagrama de Casos de Uso, y finalmente se describieron paso a paso todas las acciones de los actores del sistema con los casos de uso con los que interactúan. Gracias a esto, se puede comenzar a construir el sistema, tratando de que se cumplan todos los requerimientos establecidos en este capítulo.

Capítulo 3

Análisis y Diseño del Sistema

3.1 Introducción.

En este capítulo se traducen los requisitos a una especificación que describe cómo implementar el sistema. Con este fin se modelan los artefactos que ayudan a manejar las complicaciones que implican la construcción de aplicaciones Web. Para ello los componentes de la aplicación se tratan como clases y se presentan a través de diagramas de clases Web, que detallan la interacción de las distintas páginas. Además se estructura la información que se desea persista a través del diseño de la base de datos. Finalmente se tratan los principios de diseño de la aplicación.

3.2 Modelo de análisis.

El modelo de análisis es un modelo conceptual que esboza cómo llevar a cabo la funcionalidad dentro del sistema, incluida la funcionalidad significativa para la arquitectura. Es descrito por el lenguaje del desarrollador y proporciona la estructura a la vista interna del sistema.

En la construcción del modelo de análisis se identifican las clases que describen la realización de los casos de uso, los atributos y las relaciones entre ellas. Con esta información se construye el Diagrama de clases del análisis.

3.2.1 Diagrama de clases del análisis.

Un Diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Representa las cosas del mundo real, no de la implementación automatizada de estas cosas.

A continuación se representan los diagramas de clases del análisis para cada uno de los casos de usos que comprende la propuesta.

3.2.1.1 Diagrama de clases del caso de uso “Solicitar Servicio Telemático”

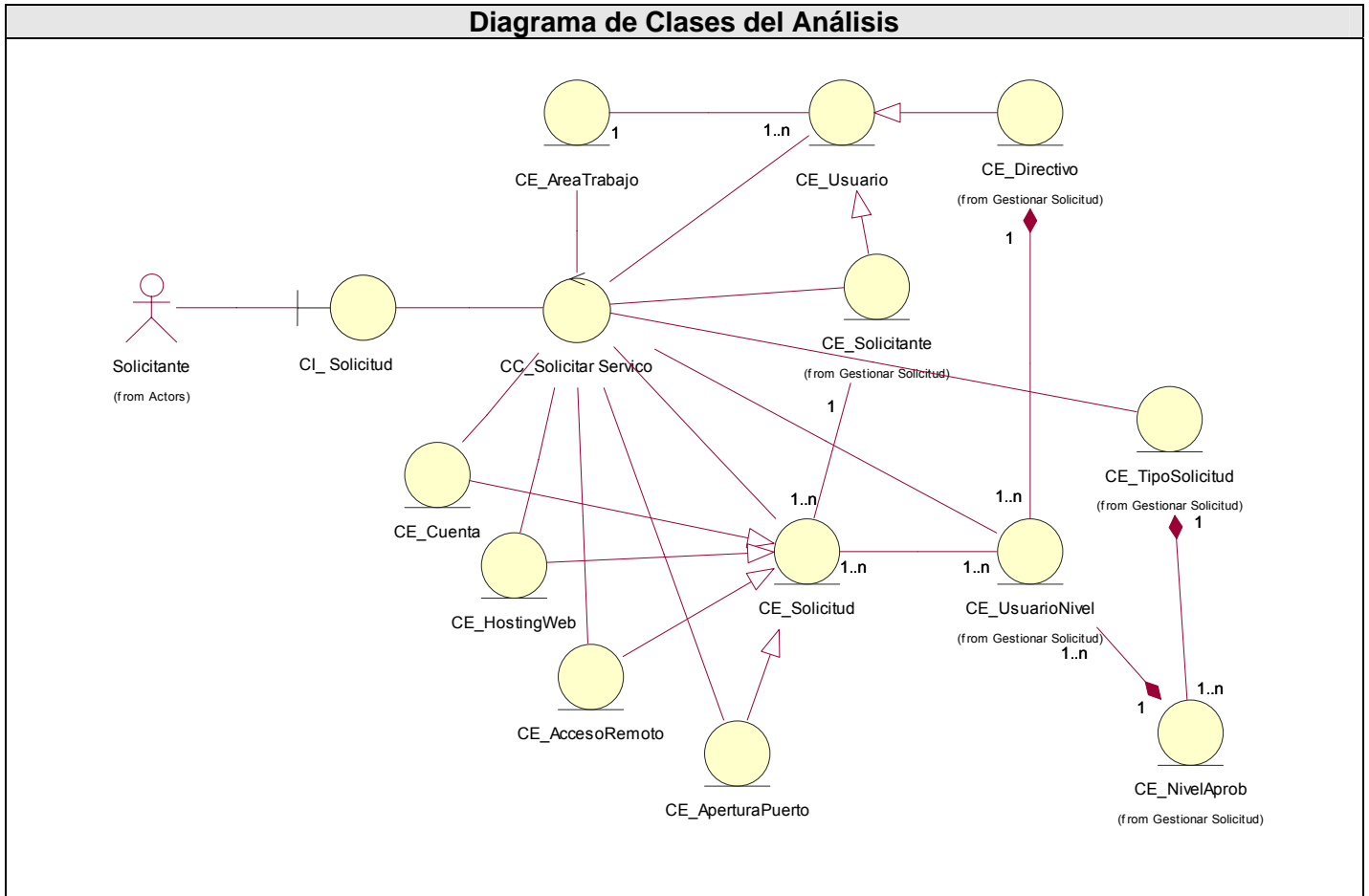


Figura 3.1: Diagrama de clases del análisis del CU Solicitar Servicio Telemático.

3.2.1.2 Diagrama de clases del caso de uso “Gestionar Solicitud”

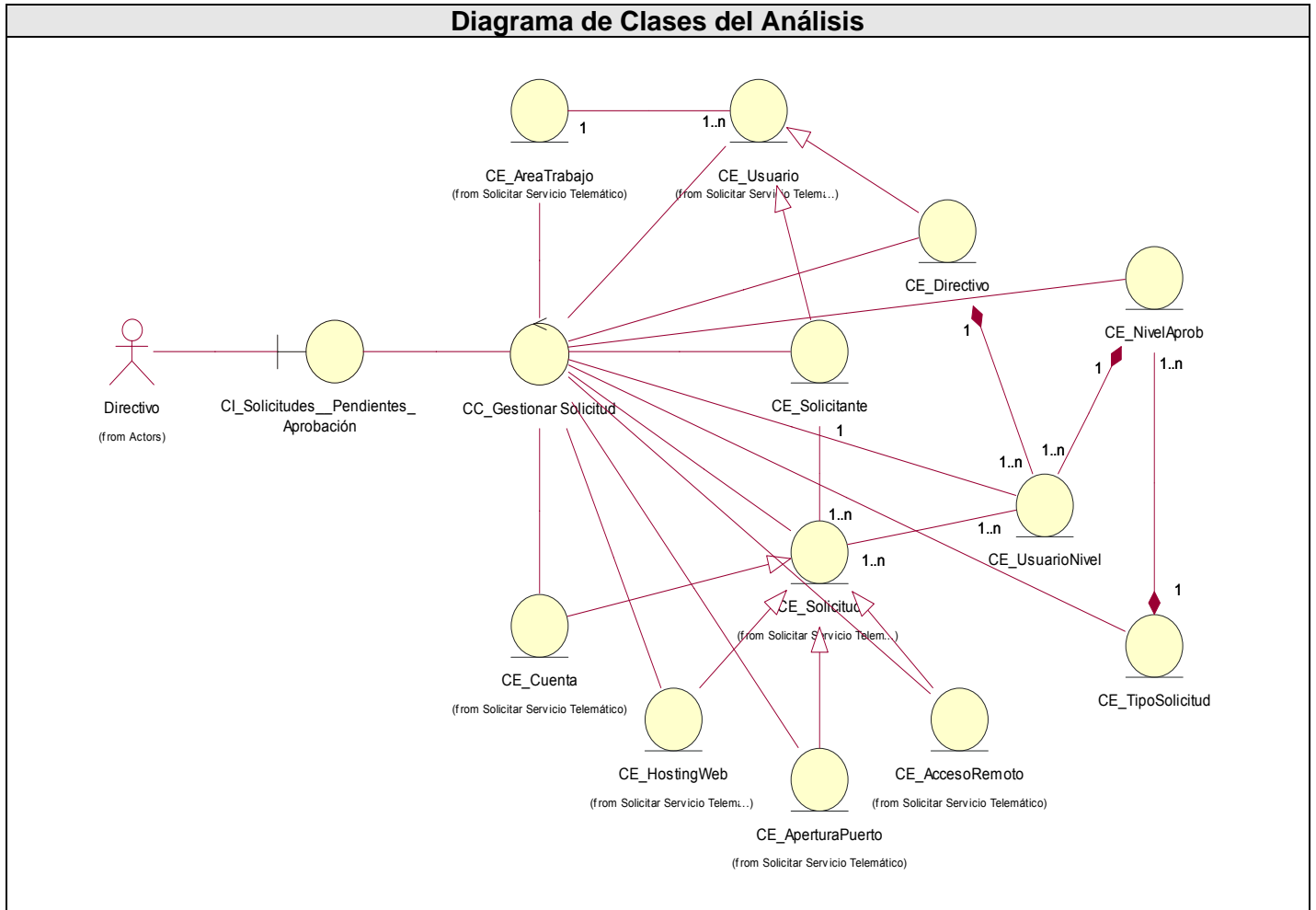


Figura 3.2: Diagrama de clases del análisis del CU Gestionar Solicitud.

3.2.1.3 Diagrama de clases del caso de uso “Generar Reporte”

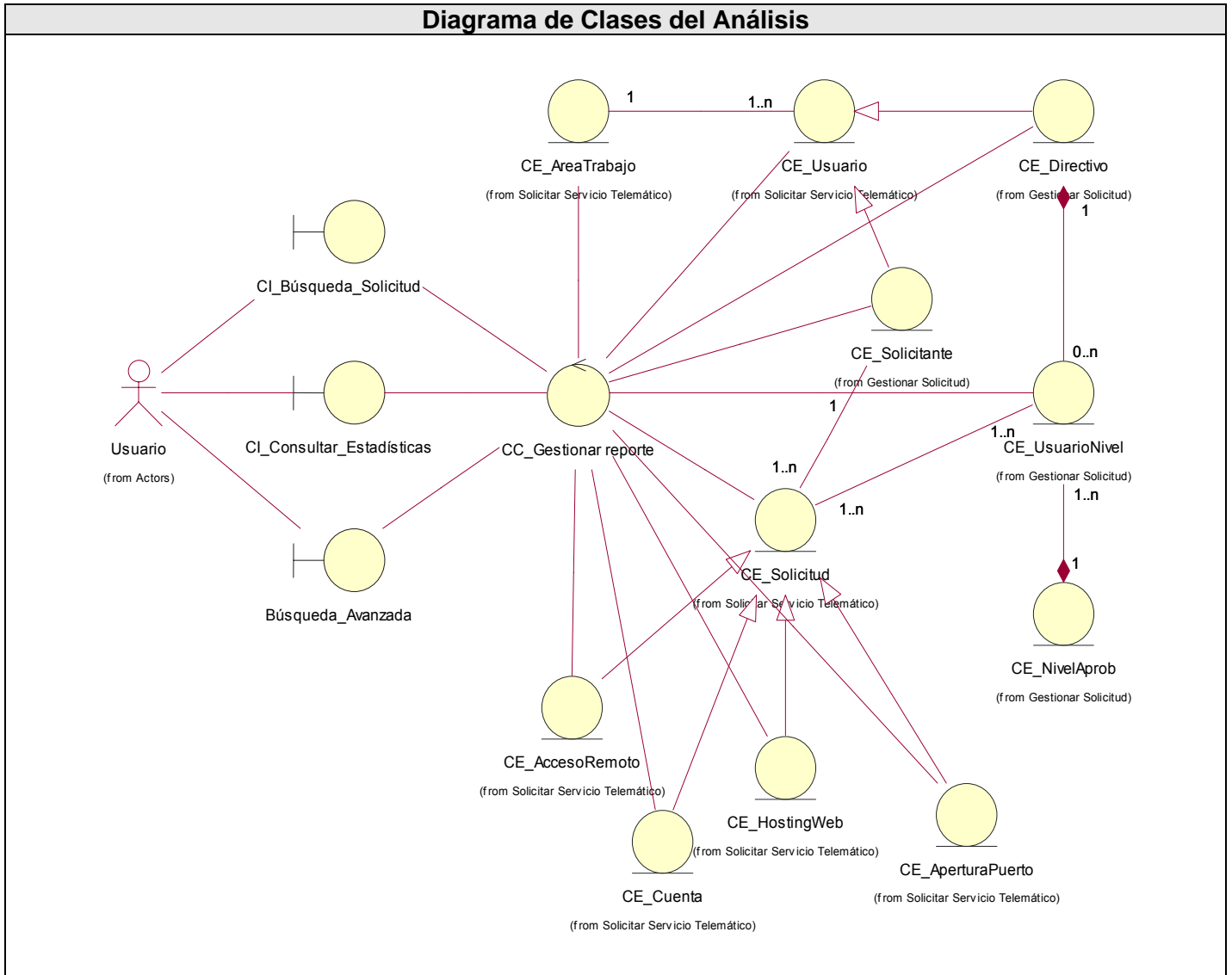


Figura 3.3: Diagrama de clases del análisis del CU Generar Reporte.

3.2.1.4 Diagrama de clases del caso de uso “Gestionar Servicio”

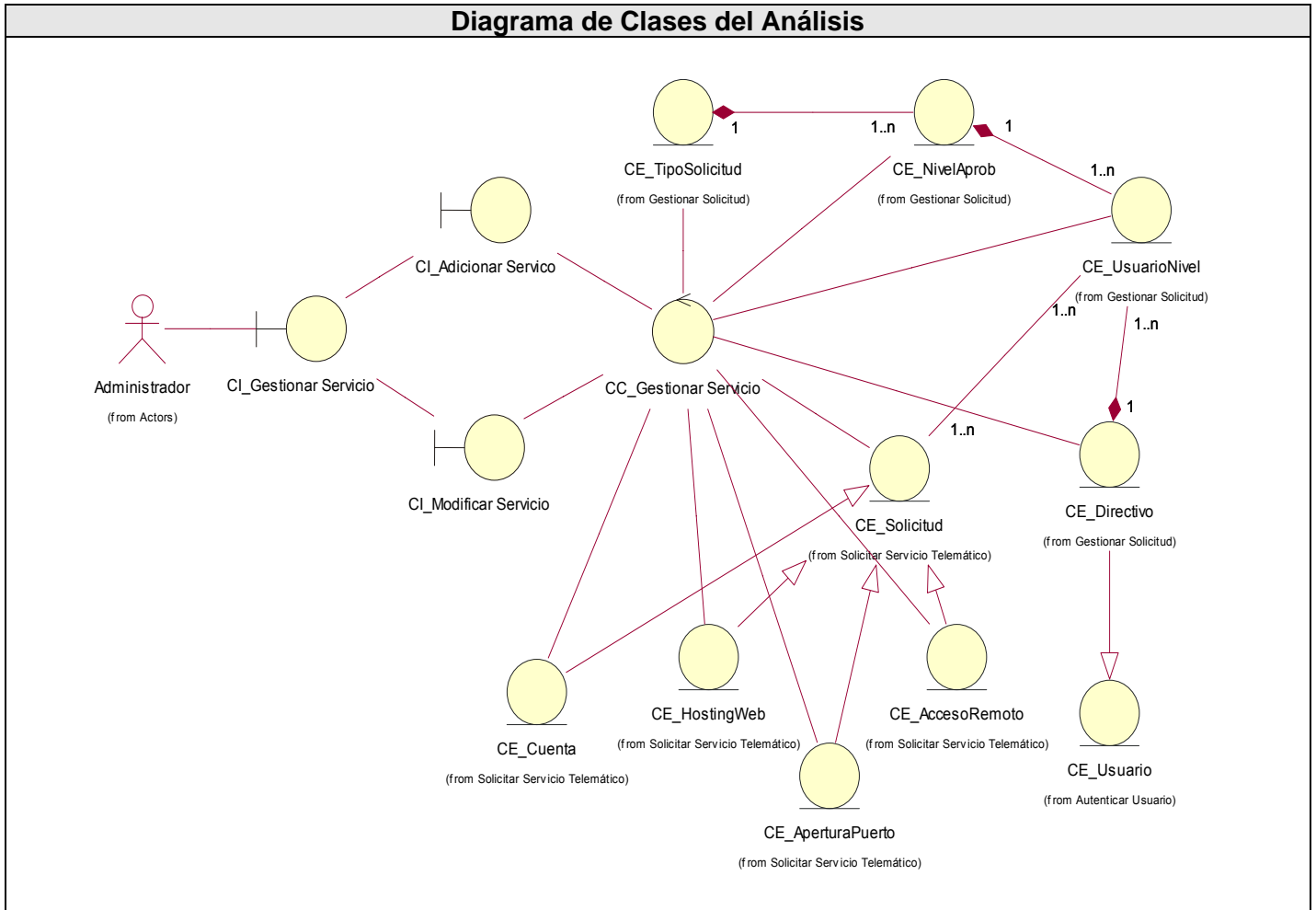


Figura 3.4: Diagrama de clases del análisis del CU Gestionar Servicio.

3.2.1.5 Diagrama de clases del caso de uso “Gestionar Usuario”

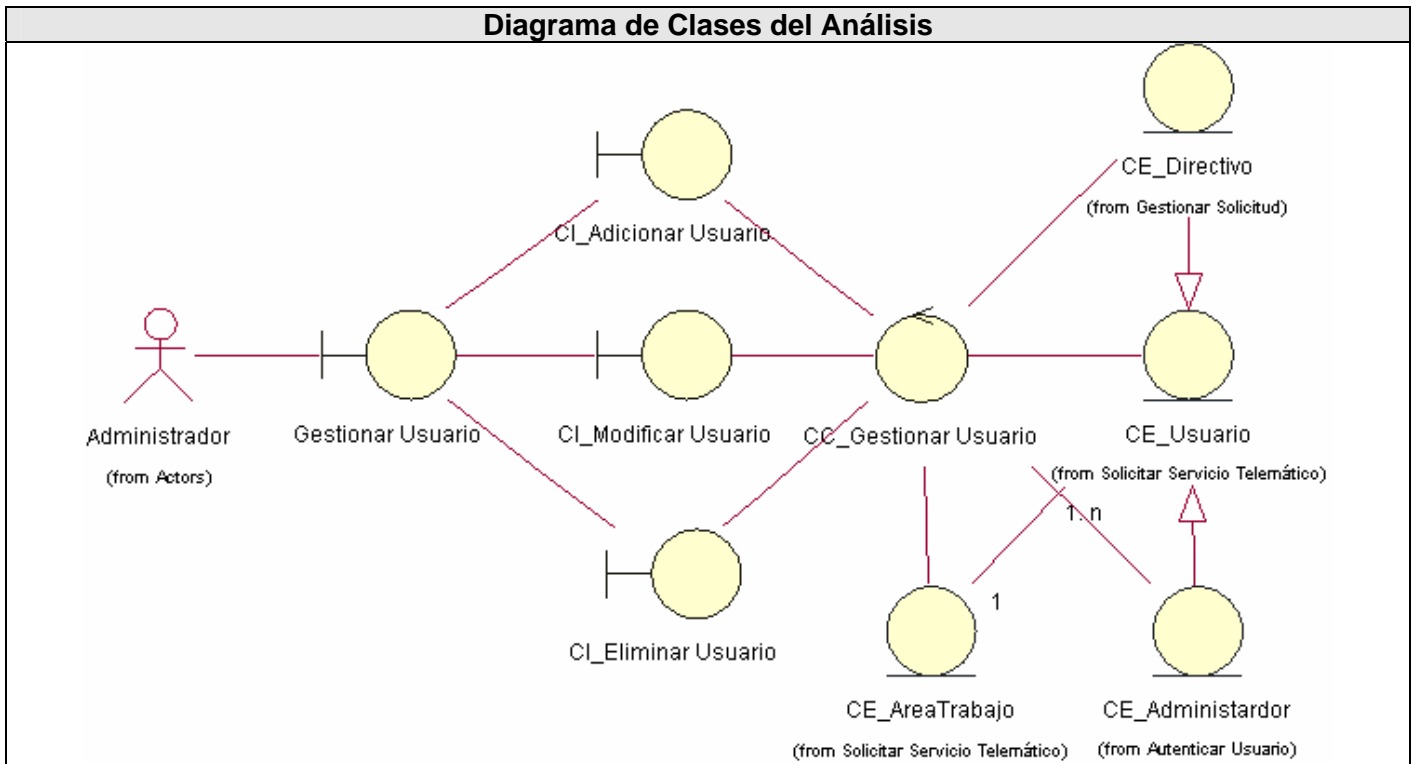


Figura 3.5: Diagrama de clases del análisis del CU Gestionar Usuario.

3.2.1.6 Diagrama de clases del caso de uso “Autenticar Usuario”

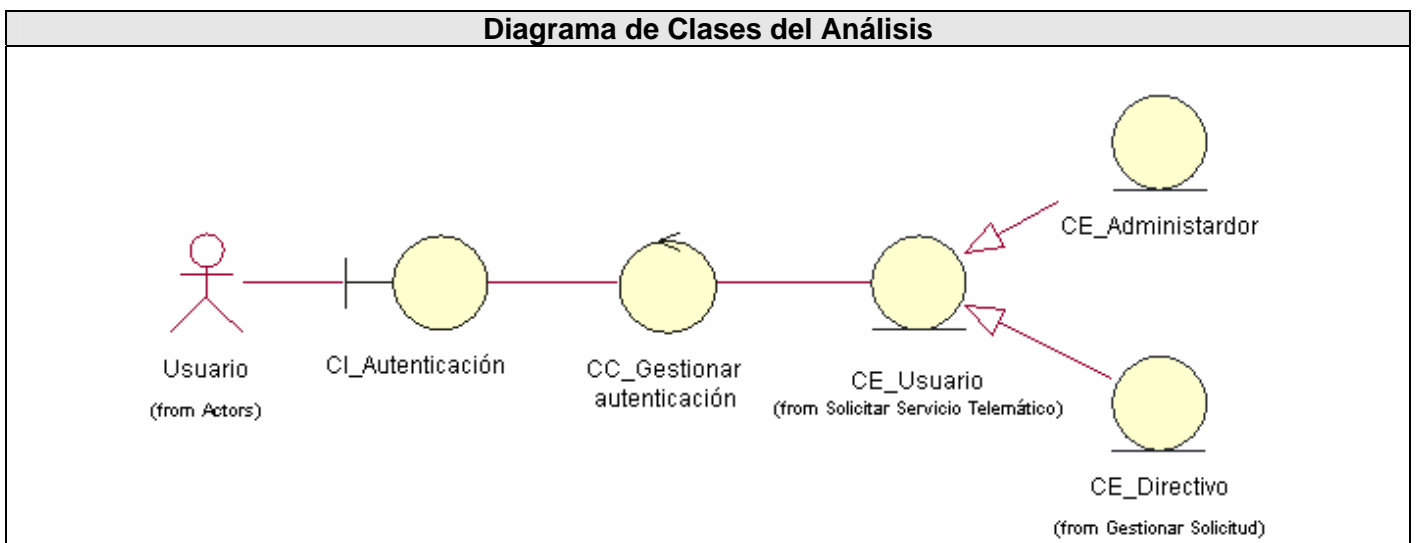


Figura 3.6: Diagrama de clases del análisis del CU Autenticar Usuario.

3.3 Modelo de diseño.

El modelo de diseño es un modelo de objetos que 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 a considerar.

Una entrada esencial en el diseño es el resultado del análisis, o sea el modelo de análisis, que proporciona una comprensión detallada de los requisitos.

En el modelo del diseño, los casos de uso son realizados por las clases del diseño y sus objetos. Esto se representa por colaboraciones en el modelo de diseño y denota la realización de casos de uso del diseño

3.3.1 Clases Base.

Los casos de uso del sistema definidos en el capítulo anterior, deben encajar en la arquitectura cuando se llevan a cabo, mientras que la arquitectura debe permitir el desarrollo de los casos de uso requeridos ahora y en el futuro. De esa manera, la arquitectura debe diseñarse para permitir que el sistema evolucione, que los desarrolladores puedan progresar hasta obtener una visión común, que se organice el desarrollo del software y que se fomente la reutilización.

Teniendo en cuenta las posibilidades que ofrece el patrón arquitectónico Modelo-Vista-Controlador, descrito en el Capítulo 1, se decide hacer uso de un framework (Code Igniter) que basa su funcionamiento en dicho patrón, para definir el sistema de clases base de la aplicación, pues se persigue facilidad de mantenimiento, escalabilidad, rapidez de desarrollo e independencia entre las distintas capas del sistema para facilitar su futura evolución.

3.3.1.1 Diagrama de clases base.

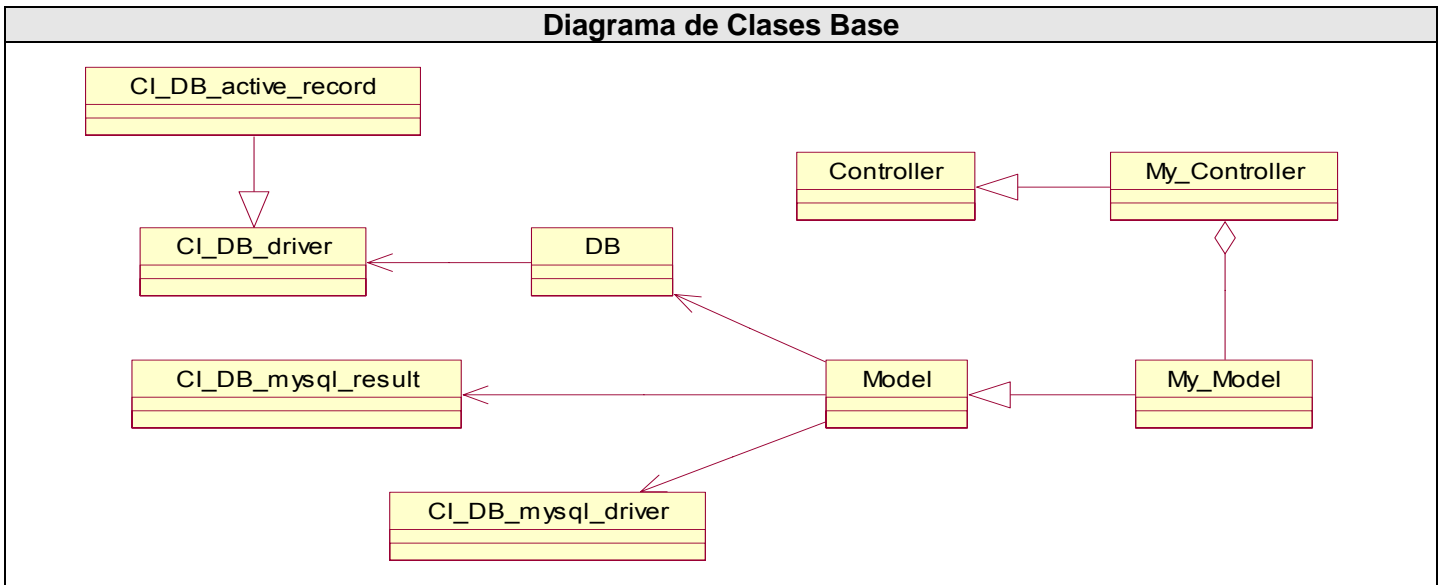


Figura 3.7: Diagrama de las clases base.

3.3.1.2 Diagrama de secuencia.

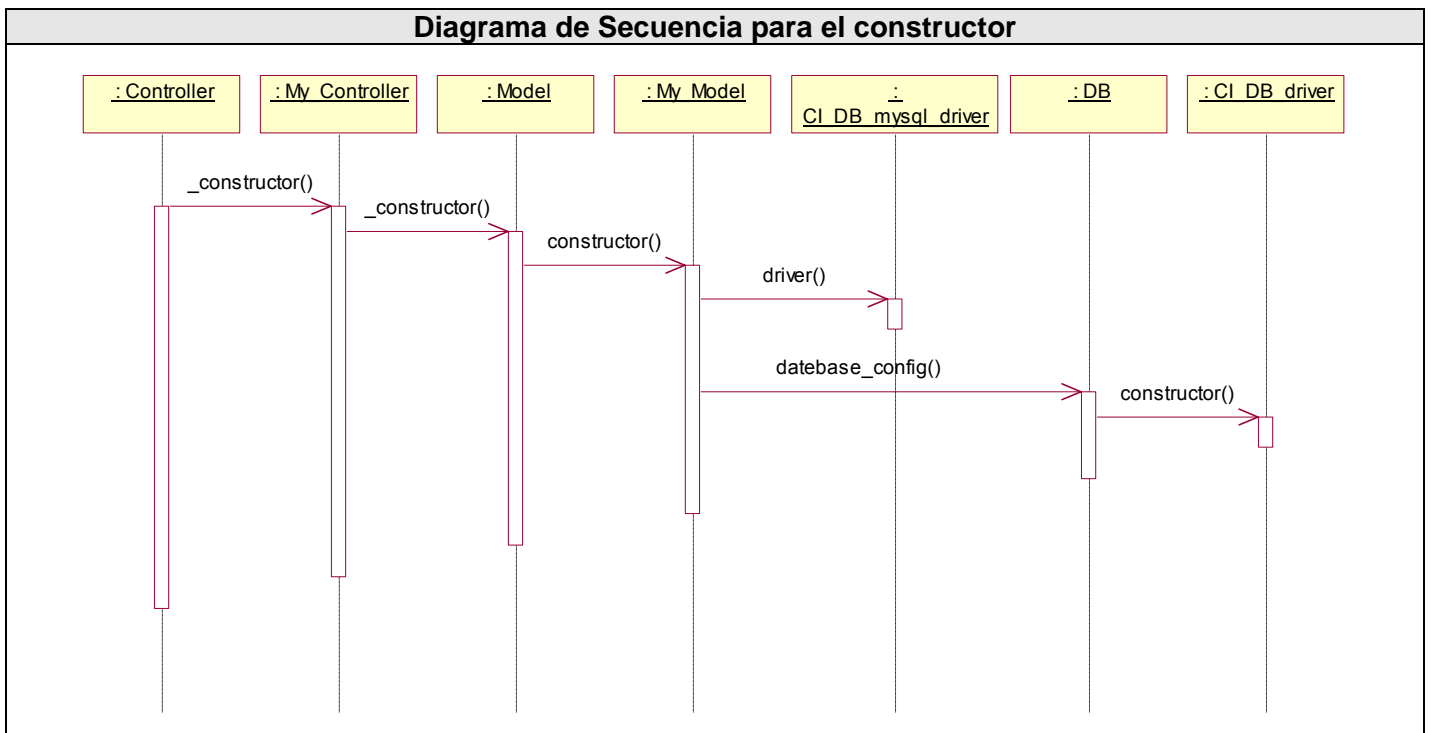


Figura 3.8: Diagrama de secuencia para el constructor.

3.3.2 Diagrama de clases del Diseño Web.

En las aplicaciones Web, donde es más importante la modelación de la lógica y el estado del negocio que los detalles de presentación, un diagrama de clases presenta las páginas, los enlaces entre estas (donde cada página lógica puede ser representada como una clase), todo el código que irán creando las páginas, así como el contenido dinámico de estas una vez que estén en el navegador del cliente.

El diagrama de clases Web, fue definido, a partir de los diferentes casos de uso del sistema y empleando las extensiones de UML para Web. A continuación se muestran los diagramas de clases para los distintos casos de uso.

3.3.2.1 Caso de Uso "Solicitar Servicio Telemático"

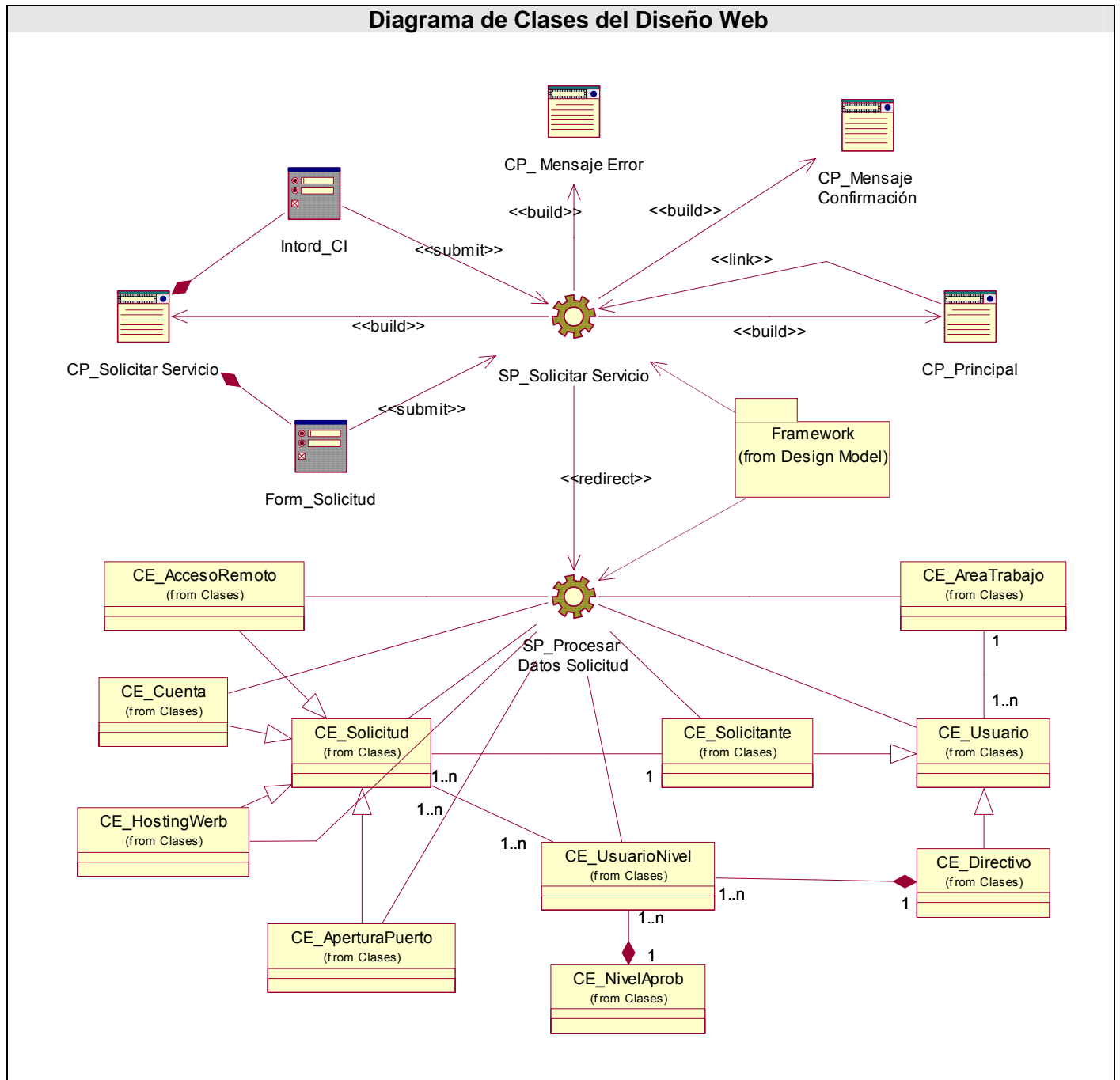


Figura 3.9: Diagrama de clases del diseño Web del CU Solicitar Servicio Telemático.

3.3.2.2 Caso de Uso "Gestionar Solicitud"

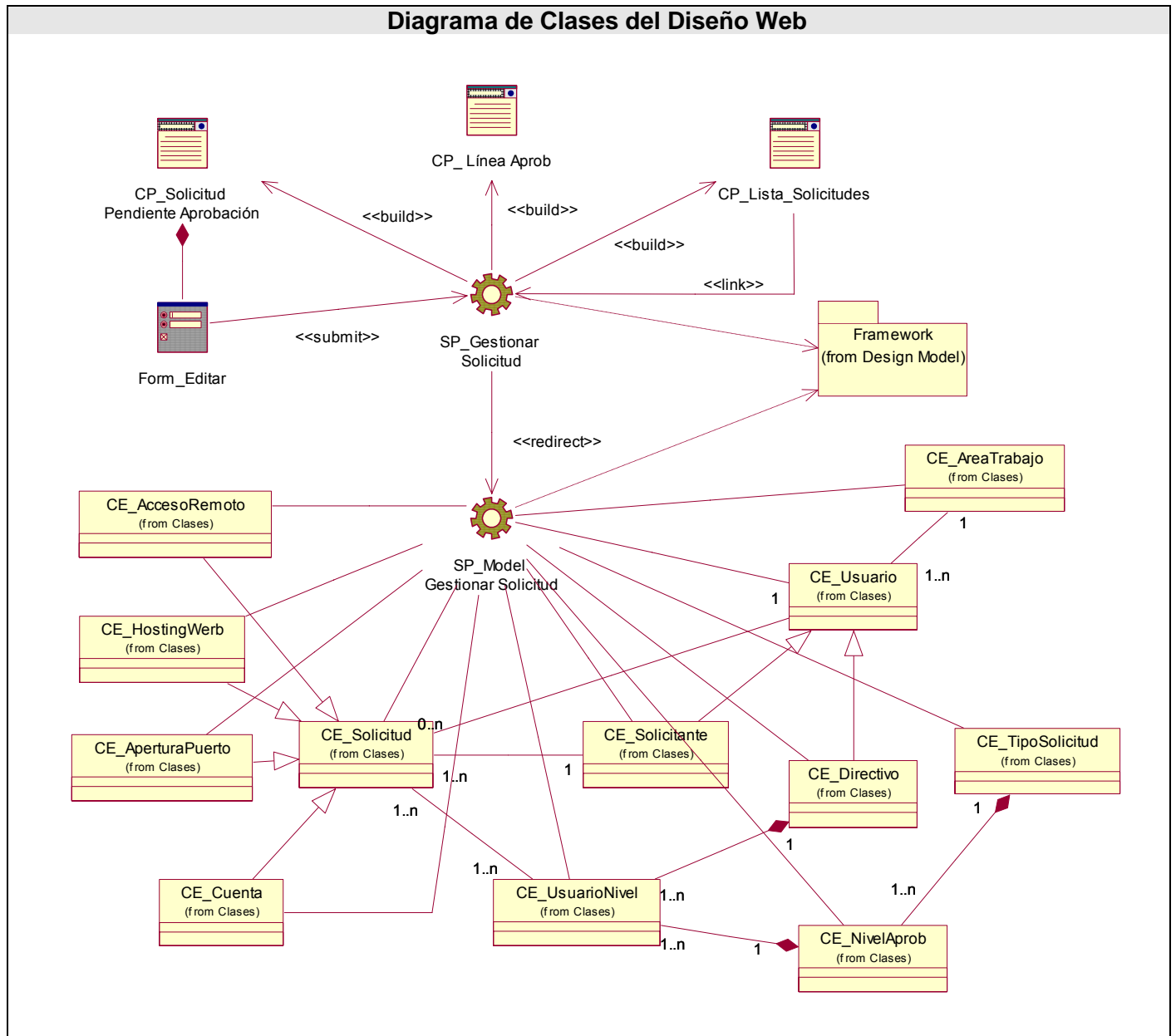


Figura 3.10: Diagrama de clases del diseño Web del CU Gestionar Solicitud.

3.3.2.4 Caso de Uso "Gestionar Servicio"

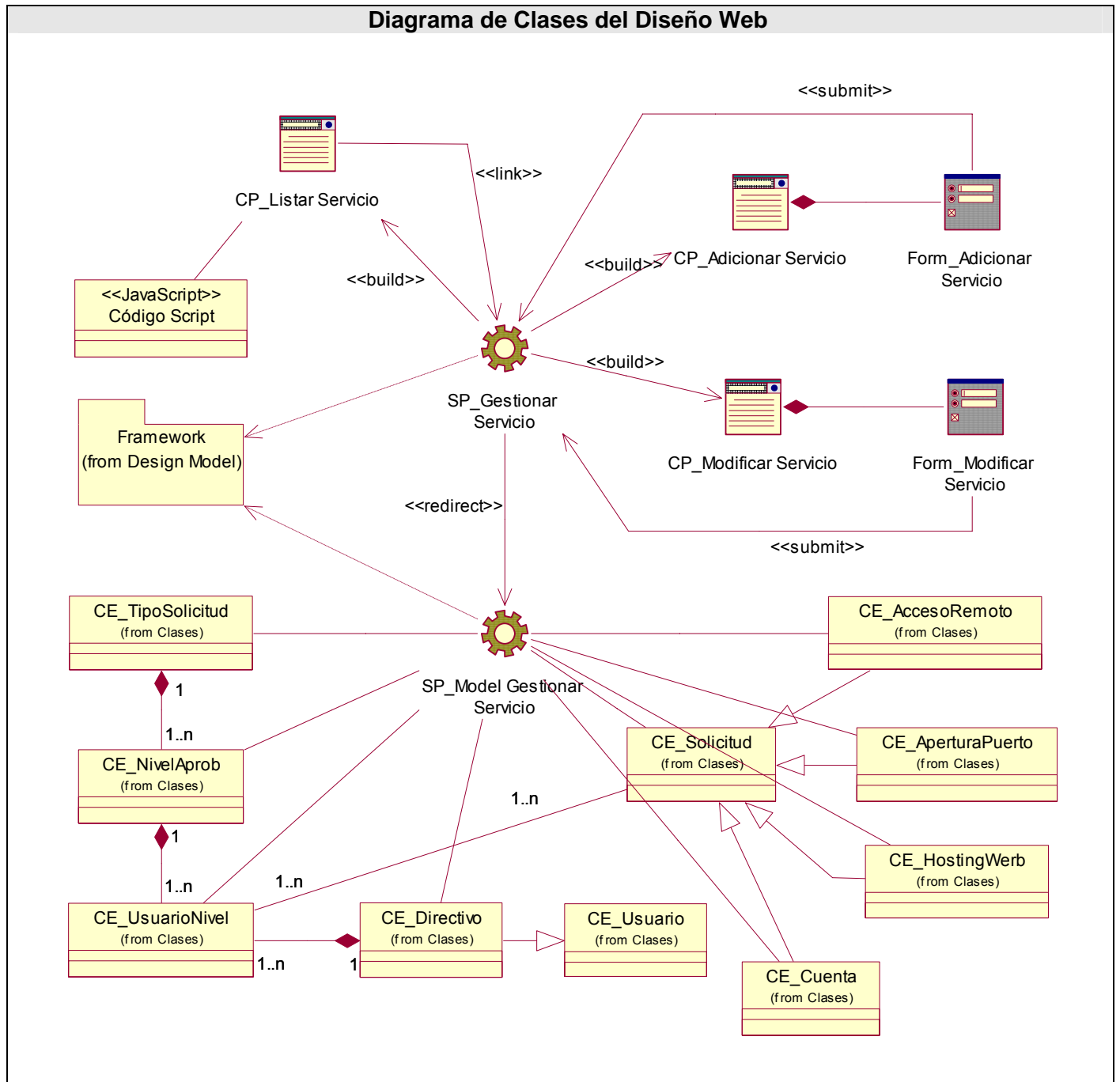


Figura 3.12: Diagrama de clases del diseño Web del CU Gestionar Servicio.

3.3.2.5 Caso de Uso "Gestionar Usuario"

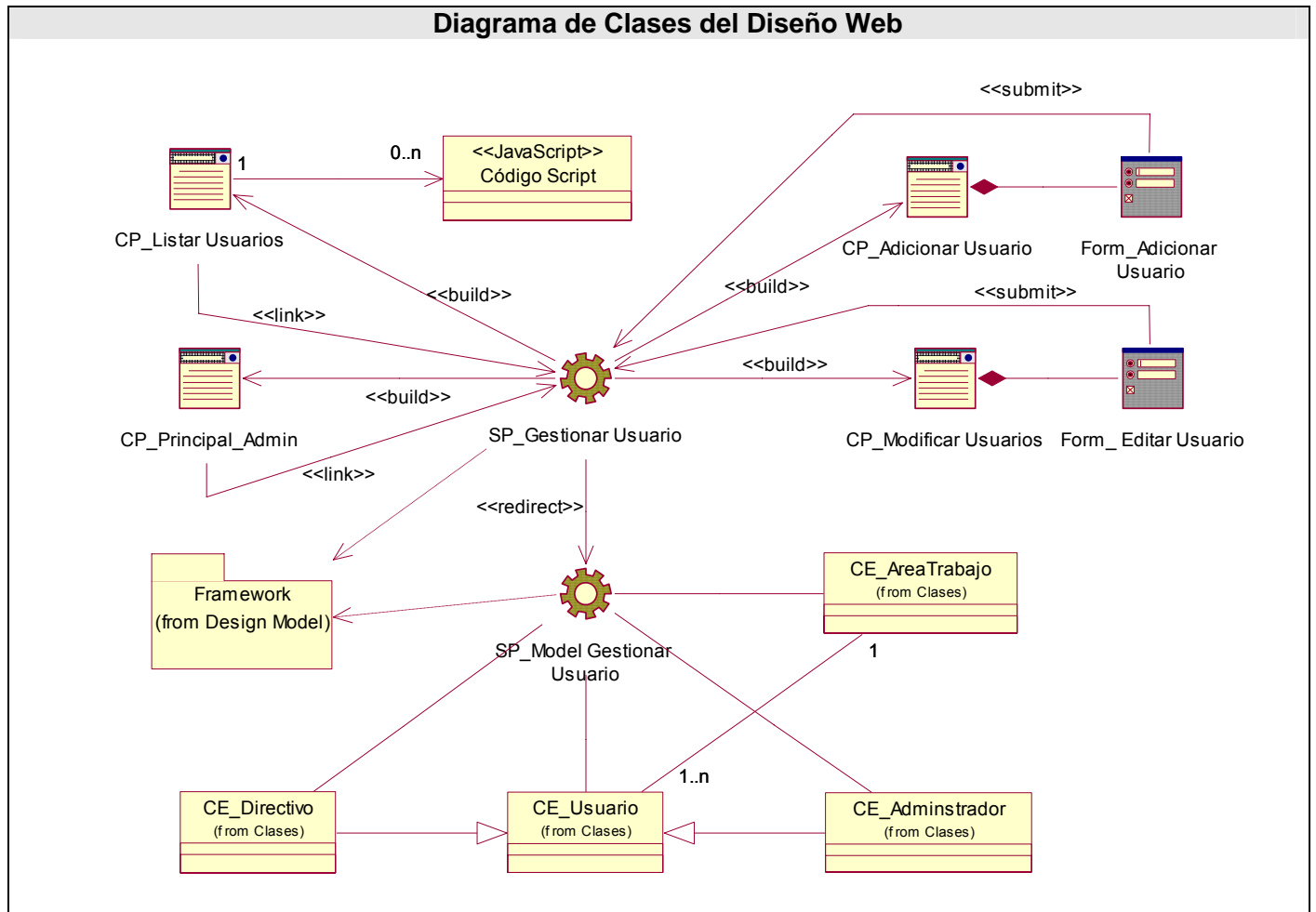


Figura 3.13: Diagrama de clases del diseño Web del CU Gestionar Usuario.

3.3.2.6 Caso de Uso "Autenticar Usuario"

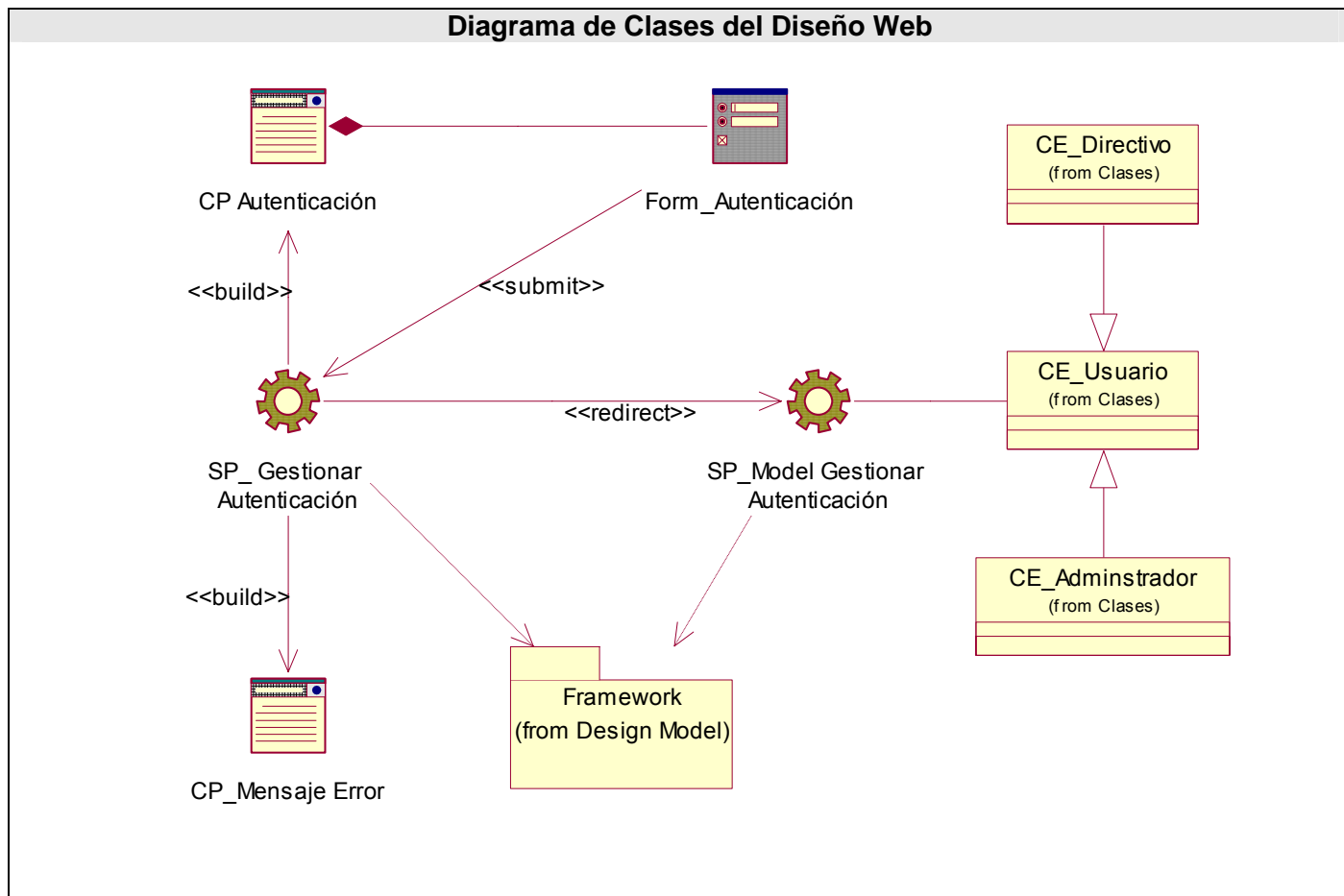


Figura 3.14: Diagrama de clases del diseño Web del CU Autenticar Usuario.

3.4 Diseño de la Base de Datos.

Para diseñar la base de datos del sistema, se ha utilizado el diagrama de clases persistentes y el modelo de datos, que están basados en el modelamiento de las clases del epígrafe anterior. Algunas de las clases representan los datos que se obtienen y almacenan durante los procesos de la aplicación, estos son las que pueden modelarse a través de un diagrama de clases persistentes, lo que permitirá ver la relación entre los datos, y completará el modelamiento de la lógica de negocio de la aplicación. Por su parte el modelo de los datos describe la representación lógica y física de dichos datos persistentes en el sistema.

3.4.1 Diagrama de Clases Persistentes.

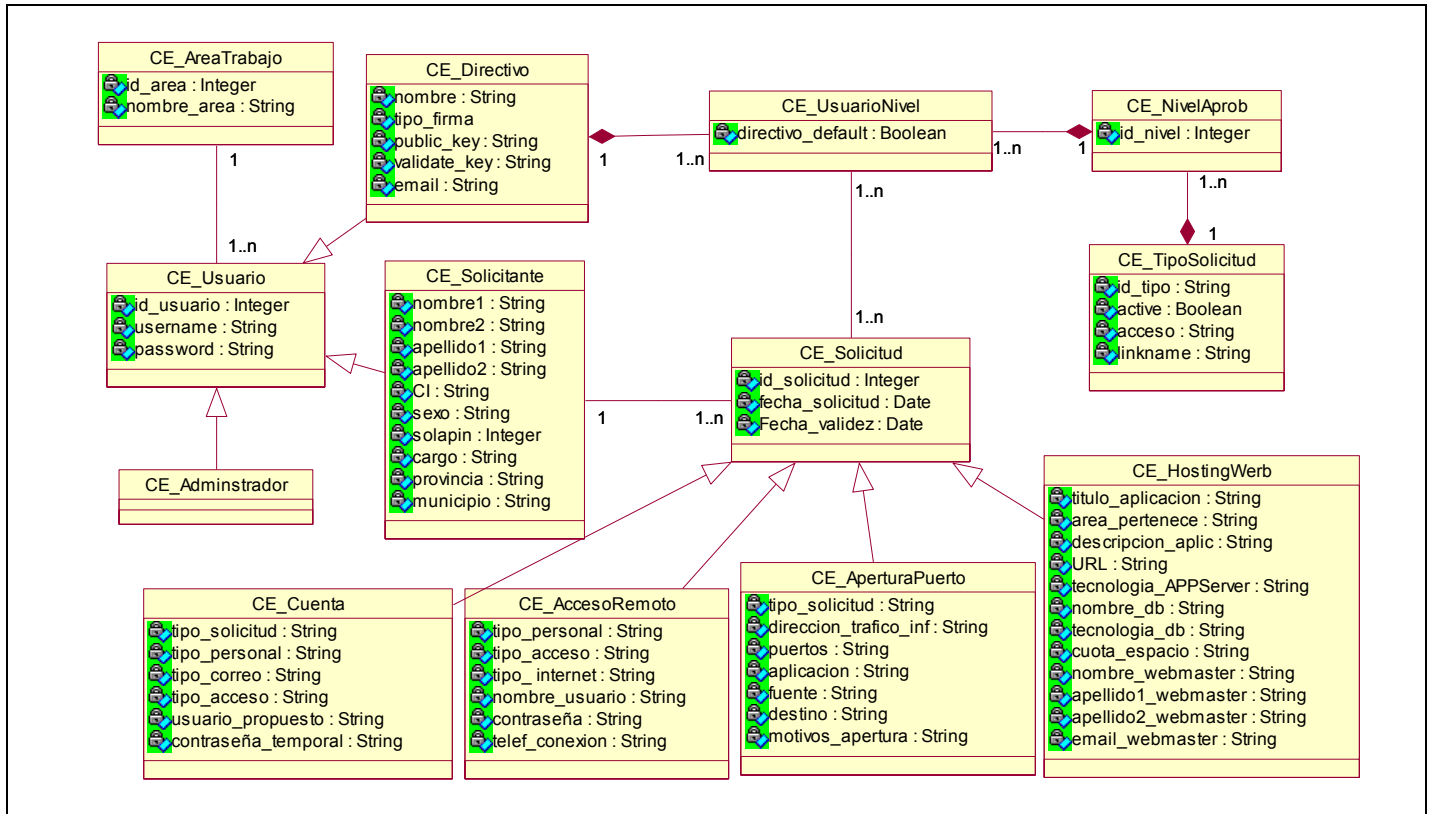


Figura 3.15: Diagrama de clases persistentes.

1. Principio de uso equiparable: donde las características de privacidad, garantía y seguridad estén igualmente disponibles para todos los usuarios, y que se proporcione las mismas maneras de uso para todos los usuarios: idénticas cuando es posible, equivalentes cuando no lo es.
2. Principio de la flexibilidad: donde se ofrezcan posibilidades de elección en los métodos de uso, que facilite al usuario la exactitud y precisión, y se adapte al paso o ritmo del usuario.
3. Principio de la simplicidad e intuición: donde se elimine la complejidad innecesaria y que sea consistente con las expectativas e intuición del usuario.
4. Principio de la Información perceptible: donde se usen diferentes modos para presentar de manera redundante la información esencial (gráfica y verbal), se proporcione contraste suficiente entre la información esencial y sus alrededores, se amplíe la legibilidad de la información esencial, y que diferencie los elementos en formas que puedan ser descritas (por ejemplo, para las funciones de catalogación).
5. Principio de tolerancia al error: donde se dispongan los elementos para minimizar los riesgos y errores, por ejemplo utilizando elementos comunes; y los elementos peligrosos eliminados, aislados o tapados, que se proporcionen advertencias sobre peligros y errores. Hay que posibilitar el descubrimiento interactivo y el aprendizaje ensayo-error, y posibilitar la reversibilidad y la recuperabilidad de las acciones.
6. Principio de esfuerzo de acceso y uso: que minimicen las acciones repetitivas, y que proporcione una línea de visión clara hacia los elementos importantes tanto para un usuario sentado como de pie.

3.5.1 Estándares en la interfaz de la aplicación.

La calidad de la interfaz de usuario puede ser uno de los motivos que conduzca a un sistema al éxito o al fracaso, es por eso que uno de los aspectos más relevantes de la usabilidad de un sistema es la consistencia de su interfaz de usuario.

Para el diseño de la interfaz de la aplicación se utilizaron colores azul claro, gris y blanco, estándares y acordes a las normas de diseño de la Universidad.

Se utilizó una hoja de estilo para guardar la configuración del diseño para todas las páginas, para los botones y las líneas se utilizaron estos estilos, eliminando así el número de imágenes que ralenticen la

presentación de la página. La letra usada fue tipo verdana de tamaño 10 y 12, porque permite una separación entre las letras óptima para presentar información.

Los formularios de entradas ocupan el centro superior, y las entradas organizadas por importancia. Se incluye alguna especificación con respecto a las entradas. Los formularios con muchas entradas se organizan y dividen de acuerdo a lo que representan.

En general se realizan múltiples operaciones en cada página, de forma que el usuario no tenga que moverse tanto dentro de la aplicación, para completar una operación. Por ejemplo se pueden hacer la modificación, eliminación en una misma página.

3.5.2 Formatos de reportes.

Generar reportes que permitan un control de las solicitudes es una de las principales funcionalidades del sistema propuesto. Estos han sido concebidos como un módulo dentro de la aplicación y se obtienen en dependencia de las necesidades del usuario.

Cada reporte tiene un encabezado que le identifica y describe brevemente, luego se muestra la información obtenida de manera legible y organizada.

3.5.3 Tratamiento de errores.

En el sistema propuesto se evitan, minimizan y tratan los posibles errores, con el fin de garantizar la integridad y confiabilidad de la información que en este se registra y muestra. Los errores se tratan en una página especial que incluye el fichero de configuración general, y está preparada para recoger el número del error y presentar la pantalla con el error que le corresponde a ese código. En algunos casos incluye la forma de solucionar el error, como es el caso de la sesión, y la autenticación de los usuarios.

Algunos errores serán generados por funciones JavaScript para evitar la ejecución de la página en vano y otros en forma de mensajes de texto en la misma página donde se ejecutó la acción, de forma que el usuario pueda corregir más fácilmente y continuar. Se utilizan mensajes de confirmación, para acciones que son irreversibles como es el caso de las eliminaciones.

Los mensajes de error que emite el sistema se muestran en un lenguaje de fácil comprensión para los usuarios. Cuando se introduce información en un formulario y faltan datos, sale un cuadro de alerta indicando el campo o dato que falta. Similar ocurre cuando se introduce información errónea en un campo numérico o e-mail.

3.6 Conclusiones.

En el este capítulo fueron tratadas las concepciones básicas para el análisis y diseño del sistema. Se desarrollaron los diagramas de clases de la aplicación y el diseño de la base de datos de la misma. Se describieron, además, los principios de diseño seguidos, específicamente, los estándares de la interfaz, los formatos para los reportes y la concepción del tratamiento de errores.

Capítulo 4

Implementación del Sistema

4.1 Introducción.

En este capítulo se describe cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue. Para ello se confeccionan los diagramas de despliegue y componentes.

4.2 Diagrama de Despliegue.

El diagrama de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Es una colección de nodos y arcos; donde cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo de hardware similar. [22]

Muestra la configuración de los componentes hardware, los procesos, los elementos de procesamiento en tiempo de ejecución y los objetos que existen en tiempo de ejecución. En este tipo de diagramas intervienen nodos, asociaciones de comunicación, componentes dentro de los nodos y objetos que se encuentran a su vez dentro de los componentes. Un nodo es un objeto físico en tiempo de ejecución, es decir una máquina que se compone habitualmente de, por lo menos, memoria y capacidad de procesamiento, a su vez puede estar formada por otros componentes. [9]

El diagrama de despliegue muestra la topología del hardware sobre el que se ejecuta el sistema. Continuación se muestra el diagrama de despliegue.

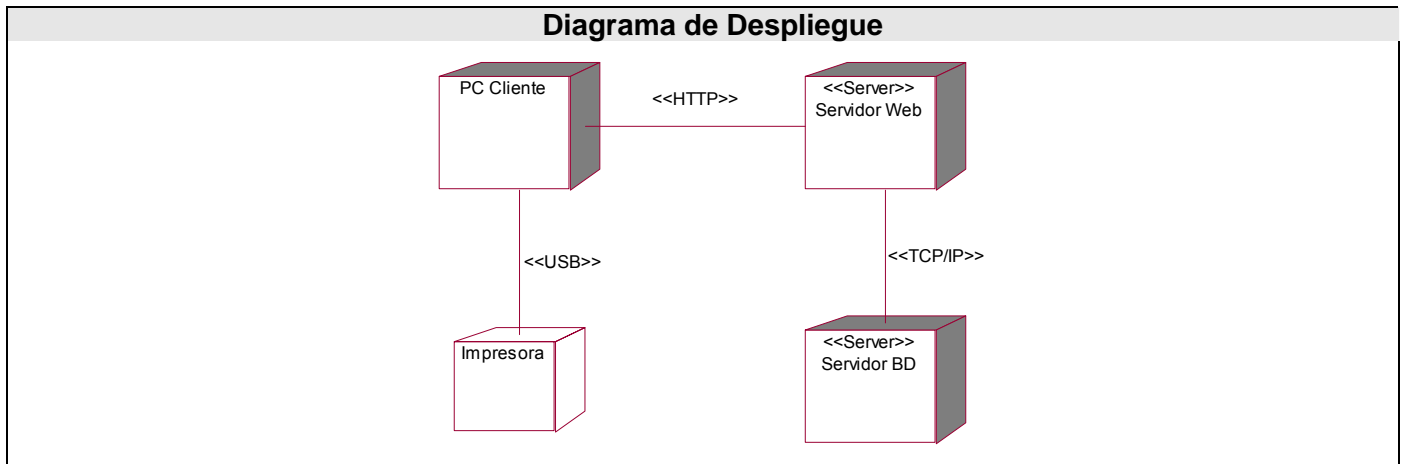


Figura 4.1: Diagrama de Despliegue.

4.3 Diagrama de Componentes.

Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación.

Un diagrama de componentes modela la vista estática de un sistema. Muestra la organización y las dependencias lógicas entre un conjunto de componentes software, sean éstos componentes de código fuente, librerías, binarios o ejecutables. No es necesario que incluya todos los componentes del sistema, normalmente se realizan por partes, cada diagrama describe un apartado del sistema (código fuente, código ejecutable, BD físicas).

4.3.1 Diagrama de componentes de Base de Datos.

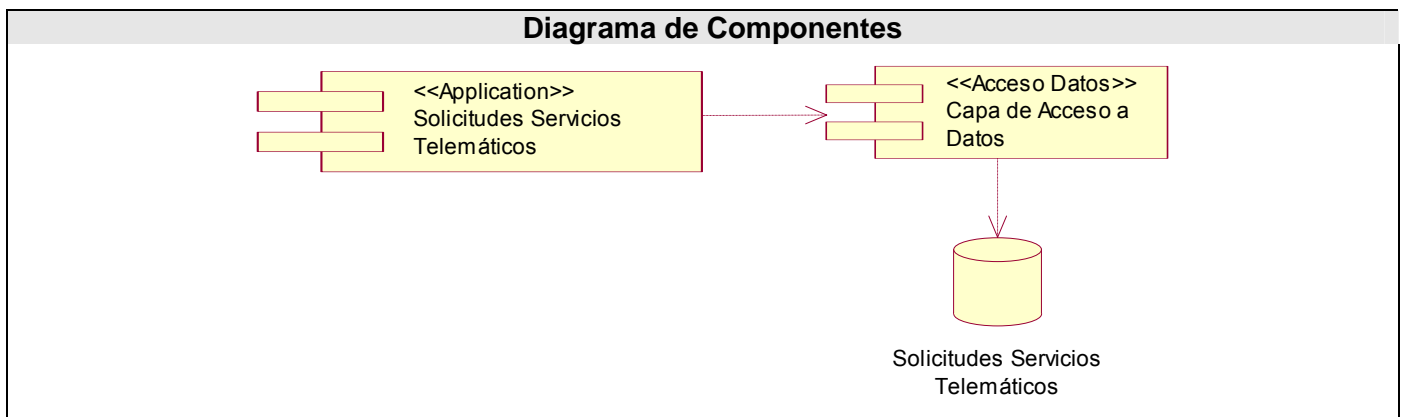


Figura 4.2: Diagrama de componentes de base de datos.

4.3.1 Diagrama de componentes de Código Fuente.

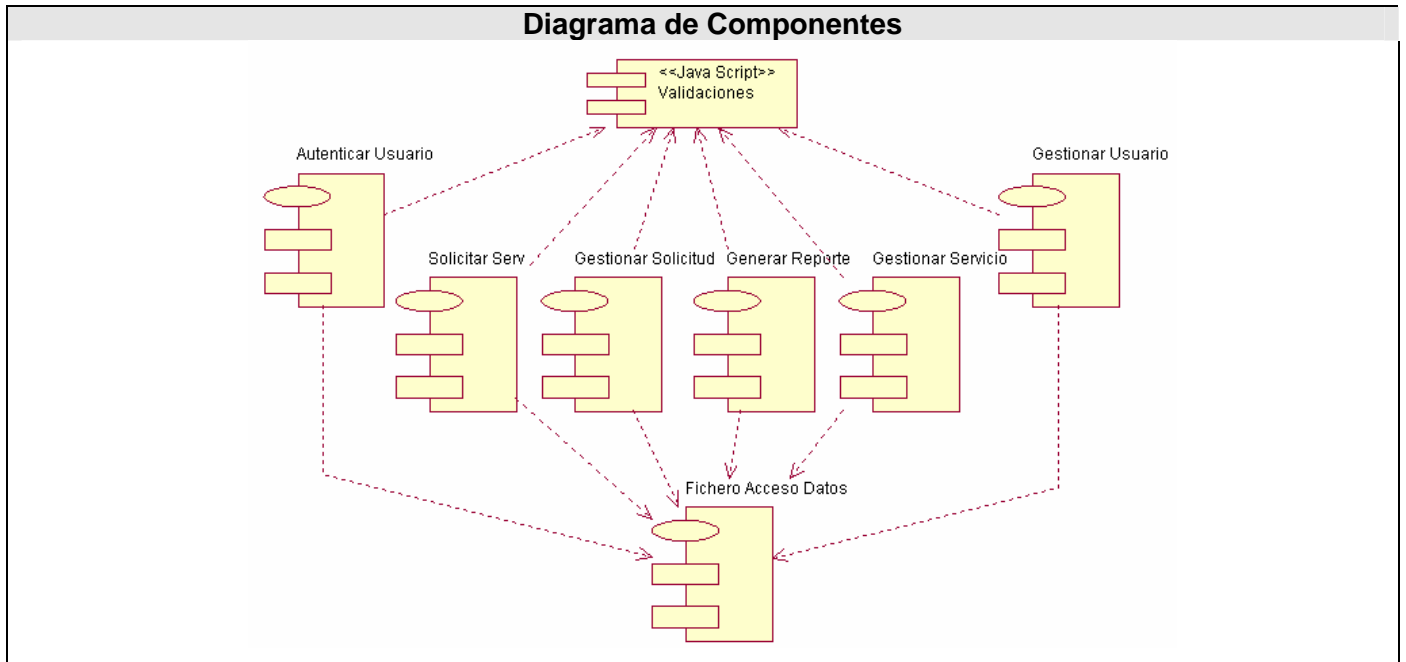


Figura 4.3: Diagrama de componentes de código fuente.

4.3.2 Diagrama de componentes de Código Ejecutable.

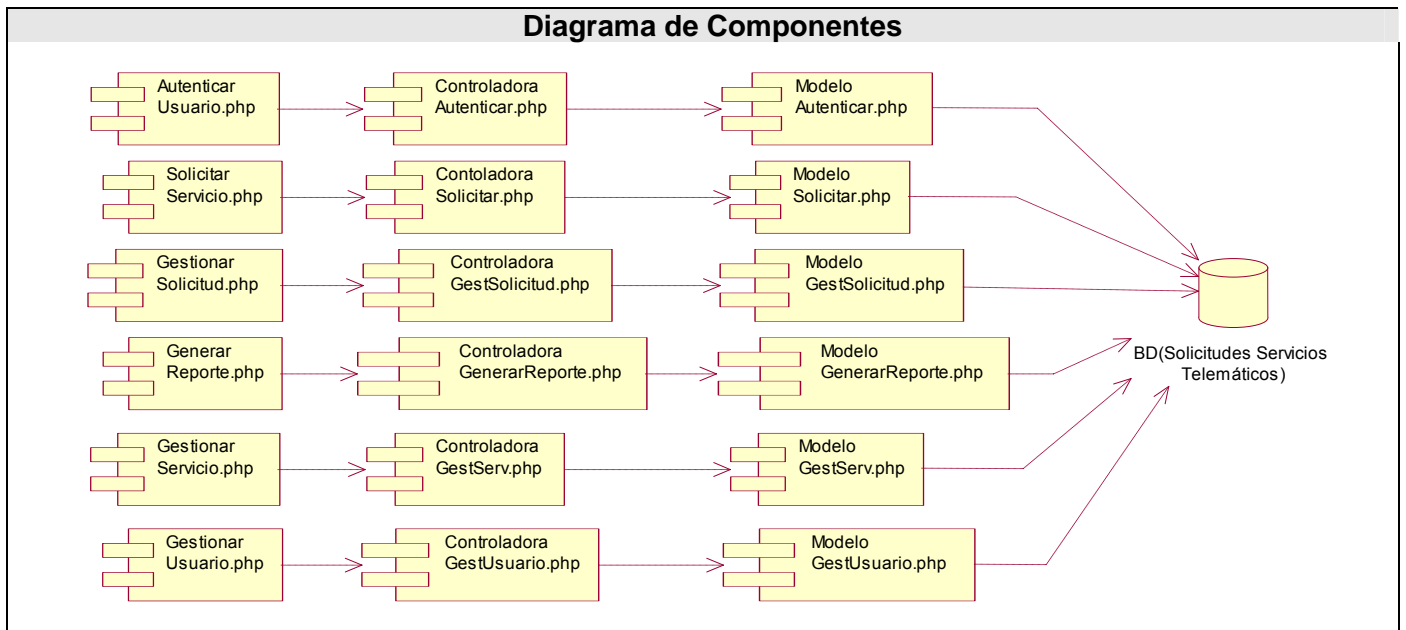


Figura 4.4: Diagrama de componentes de código ejecutable.

4.5 Conclusiones.

En este capítulo fue definida la organización de las clases y objetos en componentes, conformando los distintos diagramas de componentes para algunos apartados del sistema. También se cumplió con la descripción de la distribución física del sistema, en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo, a través del diagrama de despliegue. A partir de este momento el sistema cuenta con funcionalidad operativa.

Capítulo 5

Estudio de Factibilidad

5.1 Introducción.

Realizar un estudio de factibilidad para determinar si es factible o no implementar la propuesta de sistema resulta de vital importancia a la hora de realizar un proyecto. La estimación del costo, el esfuerzo y el tiempo necesarios para obtener el software, son factores que no se pueden obviar a la hora de emprender el trabajo. En este capítulo se llevan a cabo las actividades de estimación.

5.2 Estimación basada en Puntos de Casos de Uso.

Puntos de Casos de Uso es un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores.

A continuación, se detallan los pasos a seguir para la aplicación de éste método.

Paso 1. Cálculo de los Puntos de casos de uso Desajustados.

$$UUCP = UAW + UUCW$$

Donde: UUCP: Puntos de casos de uso sin ajustar.

UAW: Factor de peso de los actores sin ajustar.

UUCW: Factor de peso de los casos de uso sin ajustar.

- Para calcular UAW:

Tipo	Descripción	Peso	Cant * peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación.	1	0 * 1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2	0 * 2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3	3 * 3
Total			9

Tabla 5.1: Factor de peso de los actores sin ajustar.

- Para calcular UUCW:

Tipo	Descripción	Peso	Cant * peso
Simple	El Caso de Uso contiene de 1 a 3 transacciones	5	2 * 5
Medio	El Caso de Uso contiene de 4 a 7 transacciones	10	4 * 10
Complejo	El Caso de Uso contiene más de 8 transacciones	15	0 * 15
Total			50

Tabla 5.2: Factor de peso de los casos de uso sin ajustar.

Luego: UUCP = 9 + 50

$$\text{UUCP} = 59$$

Paso 2. Cálculo de los Puntos de casos de uso ajustados.

$$\text{UCP} = \text{UUCP} * \text{TCF} * \text{EF}$$

Donde: UCP: Puntos de casos de uso ajustados.

UUCP: Puntos de casos de uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

- Para Calcular TCF:

$$\text{TCF} = 0.6 + 0.01 * \sum (\text{Peso}_i * \text{Valor}_i) \text{ (Donde Valor es un número del 0 al 5)}$$

El factor de complejidad técnica (TCF) se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada factor se cuantifica en un valor.

Significado de los valores: 0: No presente o sin influencia,

1: Influencia incidental o presencia incidental

2: Influencia moderada o presencia moderada

3: Influencia media o presencia media

4: Influencia significativa o presencia significativa

5: Fuerte influencia o fuerte presencia

Factor	Descripción	Peso	Valor	$\sum (\text{Peso}_i * \text{Valor}_i)$
T1	Sistema distribuido	2	2	4
T2	Objetivos de performance o tiempo de respuesta	1	4	4
T3	Eficiencia del usuario final	1	1	1
T4	Procesamiento interno complejo	1	2	2

T5	El código debe ser reutilizable	1	3	3
T6	Facilidad de instalación	0.5	2	1
T7	Facilidad de uso	0.5	3	1.5
T8	Portabilidad	2	3	6
T9	Facilidad de cambio	1	3	3
T10	Concurrencia	1	2	2
T11	Incluye objetivos especiales de seguridad	1	4	4
T12	Provee acceso directo a terceras partes	1	3	3
T13	Se requieren facilidades especiales de entrenamiento a los usuarios	1	1	1
Total				35.5

Tabla 5.3: Factor de complejidad técnica.

Luego: $TCF = 0.6 + 0.01 * 35.5$

TCF = 0.96

- Para Calcular EF:

EF = 1.4 - 0.03 * Σ (Peso_i * Valor_i)

El factor de ambiente (EF) está relacionado con las habilidades y entrenamiento del grupo de desarrollo que realiza el sistema. Cada factor se cuantifica con un valor desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante).

Factor	Descripción	Peso	Valor	Σ (Peso _i * Valor _i)
E1	Familiaridad con el modelo de proyecto utilizado	1.5	4	6
E2	Experiencia en la aplicación	0.5	4	2
E3	Experiencia en orientación a objetos	1	4	4
E4	Capacidad del analista líder	0.5	4	2
E5	Motivación	1	5	5
E6	Estabilidad de los requerimientos	2	3	3
E7	Personal part-time	-1	1	-1
E8	Dificultad del lenguaje de programación	-1	2	-2
Total				19

Tabla 5.4: Factor de ambiente.

Luego: $EF = 1.4 - 0.03 * 19$

$$EF = 0.83$$

Finalmente $UCP = 59 * 0.96 * 0.83$

$$UCP = 47.01$$

Paso 3. Estimación de esfuerzo de FT Implementación.

$$E = UCP * CF$$

Donde: E: Esfuerzo estimado en horas hombres.

UCP: Punto de casos de usos ajustados.

CF: Factor de conversión.

- Para calcular CF:

$$\text{Total}_{EF} = \text{Cant } EF < 3 \text{ (entre E1 –E6)} + \text{Cant } EF > 3 \text{ (entre E7, E8)}$$

CF = 20 horas-hombre (si $\text{Total}_{EF} \leq 2$)

CF = 28 horas-hombre (si $\text{Total}_{EF} = 3$ ó $\text{Total}_{EF} = 4$)

CF = abandonar o cambiar proyecto (si $\text{Total}_{EF} \geq 5$)

O sea, para obtener el factor de conversión (CF) se cuentan cuantos valores de los que afectan el factor ambiente (E1...E6) están por debajo de la media (3), y los que están por arriba de la media para los restantes (E7, E8). Si el total es 2 o menos se utiliza el factor de conversión 20 Horas-Hombre / Punto de Casos de uso. Si el total es 3 o 4 se utiliza el factor de conversión 28 Horas-Hombre / Punto de Casos de uso. Si el total es mayor o igual que 5 se recomienda efectuar cambios en el proyecto ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

Como $\text{Total}_{EF} = 0 + 0$

$$\text{Total}_{EF} = 0$$

CF = 20 horas-hombre (porque $\text{Total}_{EF} \leq 2$)

Luego $E = 47.01 * 20 \text{ horas-hombre}$

$$E = 940 \text{ horas-hombre}$$

Paso 4. Calcular esfuerzo de todo el proyecto.

Actividad	Porcentaje	Horas-Hombres
Análisis	10%	235
Diseño	20%	470
Implementación	40%	940
Pruebas	15%	352.5
Sobrecarga (otras actividades)	15%	352.5
Total	100%	2350

Tabla 5.5: Esfuerzo del proyecto.

Como el valor de esfuerzo calculado representa el esfuerzo del FT implementación, por comparación salen el resto de los esfuerzo y la suma de ellos es el **esfuerzo total (E_T)**.

Suponiendo que una persona trabaje 8 horas por día, y un mes tiene como promedio 24 días hábiles; la cantidad de horas que puede trabajar una persona en 1 mes es 192 horas.

Si $E_T = 2350$ horas-hombre y por cada 192 horas yo tengo 1 mes eso daría un $E_T = 12.24$ mes-hombre. Esto quiere decir que un hombre puede realizar el sistema analizado en un período de 12.24 meses.

Tiempo de desarrollo del Proyecto.

Tiempo = E_T / CH donde: CH Cantidad de Hombres.

Tiempo = 12.24 mes-hombre / 2 hombre

Tiempo = 6.12 meses

Si en el equipo hay 2 personas y ambas realizan el mismo esfuerzo entonces el problema analizado puede terminarse en aproximadamente 183 días (6.12meses).

Costo del Proyecto.

Como la cantidad de de hombres involucrados en el desarrollo del proyecto es 2, se asume como salario promedio mensual (CHM) \$100.00.

Costo Total = CHM * (E_T / CH)

Costo Total = 100.00 * 6.12

Costo Total = \$ 612

De los resultados obtenidos se interpreta que con 2 hombres trabajando en el proyecto el mismo se desarrolla en 6.11 meses y su costo total se estima que sea \$612.

5.3 Beneficios tangibles e intangibles.

El Sistema Automatizado para la Gestión de las Solicitudes de Servicios Telemáticos no es un producto con fines comerciales, su principal objetivo es resolver los problemas existentes en el proceso de gestión de solicitudes de dichos servicios en la UCI.

El beneficio fundamental del sistema es contar con una aplicación Web flexible, dinámica y de interfaz agradable que permita registrar, actualizar y conocer de una forma más precisa, segura y en el menor tiempo posible la información referente a las solicitudes.

Por tanto, los beneficios inmediatos son generalmente intangibles:

- ◆ Disminución del tiempo y esfuerzo que se invierte en esta tarea que se realiza, hasta el momento, de forma manual.
- ◆ Fácil detección de errores.
- ◆ Fácil y rápido acceso y publicación de la información actualizada.
- ◆ Fácil procesamiento de la información y obtención, dinámica, de reportes referentes al comportamiento de las solicitudes en cualquier momento.

5.4 Análisis de costos y beneficios.

Desarrollar un producto informático resulta costoso, justificar entonces su realización depende de los beneficios que reportarían su implantación y utilización.

Como ya se ha mencionado en el epígrafe anterior, el software que se propone aportaría beneficios inmediatos intangibles de suma importancia, pues una vez implantado, contribuirá a aumentar la eficiencia en la gestión de las solicitudes de servicios telemáticos, al proporcionar comodidad, rapidez y confiabilidad

a la hora del llenado, búsqueda y verificación de las solicitudes, un mayor control por parte de los directivos, así como un análisis certero de los comportamientos de dichas solicitudes.

Por otro lado se tiene que la tecnología utilizada para el desarrollo del mismo es libre, por tanto no hay que incurrir en gastos en el pago de licencias de uso. El sistema es portable por lo que un cambio de plataforma para la implantación del mismo es viable y factible, y no hay que incurrir en muchos cambios; debido a la estructuración en capas de los procesos del negocio diseñados.

Por tanto, analizando el costo del proyecto, los numerosos beneficios que reporta, se puede concluir que su implementación es realmente factible.

5.5 Conclusiones.

En este capítulo se le ha realizado un estudio de factibilidad económica al sistema propuesto, que ha permitido arribar a la conclusión de que la herramienta reportará beneficios significativos e importantes para la gestión de las solicitudes desarrollada en la UCI, lo que indica que es factible implementar la herramienta propuesta.

Conclusiones

La culminación de una tarea, por pequeña que sea, siempre es objeto de regocijo por parte de quien la emprendió; pues en ella se ve reflejado el fruto de varias jornadas de esfuerzo y dedicación.

Como colofón de este trabajo, el cual resume un largo camino de estudios, queda lo siguiente:

- ✓ Se definió y fundamentó los principales elementos teóricos - metodológicos para el análisis del problema a resolver.
- ✓ Se desarrolló el análisis y diseño de una aplicación Web que permite la gestión de las solicitudes de los servicios telemáticos en la UCI.
- ✓ Se implementó un sistema con las características definidas en los procesos de análisis y diseño.

Por lo que se puede concluir que se alcanzó, satisfactoriamente, el objetivo propuesto: desarrollar una solución robusta, flexible y única de software que dé soporte al proceso de gestión de las solicitudes de servicios telemáticos en la UCI.

Recomendaciones

Ninguna obra humana es perfecta, como bien sentenciaba un sabio, es por ello que al culminar esta tarea lo primero que se debe tener en cuenta es realizar las recomendaciones pertinentes que permitan un posterior desarrollo y perfeccionamiento del sistema implementado. Por ello se recomienda:

- ◆ Poner a prueba el sistema durante un período de tiempo significativo, para comprobar su desempeño y que las funcionalidades del sistema se correspondan con la realidad de lo que se desea.
- ◆ Continuar el estudio con el objetivo de añadir nuevas funcionalidades.
- ◆ Desarrollar una ayuda en línea con vistas a orientar e informar al usuario.

Referencias Bibliográficas

- [1] *Firma Digital* [Disponible en: <<http://www.pki.gov.ar/index.php?option=content&task=view&id=91&Itemid=102>> [Fecha de consulta: 16 enero 2007]
- [2] Talens-Oliag, S. *Introducción a la Criptología*. [Disponible en: <<http://www.uv.es/~sto/articulos/BEI-2003-04/criptologia.html#id2442402>>] [Fecha de consulta: 16 enero 2007]
- [3] *El algoritmo RSA y la factorización de números grandes*. [Disponible en: <http://daniellerch.com/sources/doc/algoritmo_rsa.html>] [Fecha de consulta: 16 enero 2007]
- [4] *DSA* [Disponible en: <<http://es.wikipedia.org/wiki/DSA>>] [Fecha de consulta: 16 enero 2007]
- [5] Sotelo, A. *GNUpg Básico*. Última actualización, 04 de Enero, 2004 [Disponible en: <<http://www.slackware.cl/index.php?q=node/113>>] [Fecha de consulta: 16 enero 2007]
- [6] Herramientas Web para la enseñanza de protocolos de comunicación. *MD5* [Disponible en: <<http://neo.lcc.uma.es/evirtual/cdd/tutorial/presentacion/md5.html>> [Fecha de consulta: 16 enero 2007].
- [7] *Programación Web*. [Disponible en: <<http://www.arsys.es/soporte/programacion/windows.htm>> [Fecha de consulta: 20 enero 2007].
- [8] *Aplicaciones Web* [Disponible en: <http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web> [Fecha de consulta: 20 enero 2007]
- [9] Valido, Y. y Moreira, Y. *SAIMM: Sistema de Apoyo Integral a la Misión Milagro*. Trabajo de Diploma para optar por el título de Ingeniero Informático, Instituto Superior Politécnico “José Antonio Echeverría”, Ciudad de la Habana, junio 2005.
- [10] Alonso L. *Modulo Admisión del Sistema Automatizado para la Gestión de Información de la Misión Milagro*. Trabajo de Diploma para optar por el título de Ingeniero Informático, Instituto Superior Politécnico “José Antonio Echeverría”, Ciudad de la Habana, junio 2006.
- [11] *Arquitectura cliente/servidor* [Disponible en: <<http://www.csi.map.es/csi/silice/Global71.html>> [Fecha de consulta: 24 enero 2007]
- [12] *Cliente-servidor* [Disponible en: <<http://es.wikipedia.org/wiki/Cliente-servidor>> [Fecha de consulta: 24 enero 2007]

- [13] Soto, N. y Saborit, Y. *Hubble. Propuesta para un Sistema de Catalogación y Recuperación de Recursos de Información*. Trabajo de Diploma para optar por el título de Ingeniero Informático, Instituto Superior Politécnico “José Antonio Echeverría”, Ciudad de la Habana, junio 2004.
- [14] *PHP* [Disponible en: <<http://es.wikipedia.org/wiki/PHP>> [Fecha de consulta: 24 enero 2007]
- [15] *Servicio web* [Disponible en: http://wiki.prod.uci.cu/index.php/Servicio_web> [Fecha de consulta: 24 enero 2007]
- [16] Sitio Web Linalco, especialistas en Linux y software libre. *Servicios y soluciones linalco*. [Disponible en: <<http://www.linalco.com/soluciones.html>> [Fecha de consulta: 24 enero 2007]
- [17] Sitio Web *Linux para todos* [Disponible en: <<http://www.linuxparatodos.net/geeklog/staticpages/index.php?page=servidor-web>> [Fecha de consulta: 24 enero 2007]
- [18] *MySQL* [Disponible en: <<http://es.wikipedia.org/wiki/MySQL>> [Fecha de consulta: 26 enero 2007]
- [19] Welicki, L. *Patrones y Antipatrones: una Introducción - Parte II* Disponible en: <http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_3317.asp>. [Fecha de consulta: 26 enero 2007].
- [20] *Patrones de Diseño*. Conferencia 7 de Ingeniería del Software I, curso 2005-2006, UCI.
- [21] *Introducción a la Ingeniería de Software*. Conferencia 1 de Ingeniería del Software I, curso 2005-2006, UCI.
- [22] Booch, G.; Rumbaugh, J. y Jacobson, I. *El Lenguaje Unificado de Modelado*. 2000.
- [23] Sitio Web Clikear.com. Portal para desarrolladores .NET Framework. *Desarrollo Orientado a Objetos con UML*. [Disponible en: <<http://www.clikear.com/manuales/uml/introduccion.asp>> [Fecha de consulta: 29 enero 2007]
- [24] Álvarez, M. *Evaluando Zend Studio*. [Disponible en: <<http://www.maestrosdelweb.com/editorial/zendstudio/>> [Fecha de consulta: 29 enero 2007]
- [25] *Modelo del Negocio*. Conferencia 2 de Ingeniería de Software I, curso 2005-2006, UCI.

Bibliografía

- ◆ Booch, G.: Rumbaugh, J. y Jacobson, I. *El Proceso Unificado de Desarrollo de Software*. Pearson education, S. A. Madrid 2000.
- ◆ Franco, José A. *UML en acción. Modelando Aplicaciones Web*. Conferencia, CUJAE.
- ◆ Gallego Vázquez, José A. *Desarrollo Web con PHP y MySQL*. Ediciones Anaya Multimedia (Grupo Anaya) S. A. 2003. [Disponible en <http://bibliodoc.uci.cu/pdf/reg02138.pdf>]
- ◆ Larman, Craig. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. Primera Edición Prentice Hall, Hispanoamericana S.A. 1999.
- ◆ Larman, Craig. *UML y Patrones. Introducción al análisis y diseño orientado a objetos y al proceso unificado*. Segunda Edición Prentice Hall.
- ◆ Matos, Rosa María. *Introducción al trabajo con Base de Datos*. Asignatura de Sistemas de Gestión de Base de Datos.
- ◆ Peralta, Mario. *Estimación del esfuerzo basada en casos de uso*. Centro de Ingeniería de Software e Ingeniería del Conocimiento, Buenos Aires, Argentina.
- ◆ Pressman, Roger S. *Ingeniería del Software. Un enfoque práctico*. Parte I y II (Español). La Habana, Editorial Félix Varela, 2005, quinta edición, 610p.
- ◆ Quatrani, Terry. *Visual Modeling with Rational Rose 2000 and UML*, Publisher Addison Wesley, Second Edition October 19, 1999.
- ◆ Schmuller, J. *Aprendiendo UML en 24 Horas*. PEARSON EDUCACION, México, 2000.

Glosario de Términos

- ◆ APACHE: es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/1.1.
- ◆ API: *Application Programming Interface* (Interfaz de Programación de Aplicaciones) es el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta librería para ser utilizado por otro software como una capa de abstracción.
- ◆ ASP: *Active Server Pages*. Es una tecnología del lado servidor de Microsoft para páginas web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Server (IIS). Con ASP se pueden combinar páginas HTML, *scripts* y objetos COM.
- ◆ CASE: *Computer Aided Software Engineering*.
- ◆ CGI: *Common Gateway Interface*. Estándar para transferir datos entre el cliente y el programa.
- ◆ CSS: *Cascading Style Sheets*. Lenguaje de hojas de estilo en cascada utilizadas para definir la presentación de un documento HTML o XML.
- ◆ CUN: *Caso de uso del negocio*.
- ◆ CUS: *Caso de uso del sistema*.
- ◆ DOM: *Document Object Model* (Modelo de Objetos de Documento). Es una API para acceder, añadir y cambiar dinámicamente contenido estructurado en documentos con lenguajes como Javascript. Es una forma de representar los elementos de un documento estructurado (tal como una página web HTML o un documento XML).
- ◆ Framework: Estructura de soporte definida, en la cual otro proyecto de software puede ser organizado y desarrollado. Puede incluir soporte de programas, bibliotecas y un lenguaje de scripting entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.
- ◆ HTML: *HyperText Markup Language*. Lenguaje de marcas usado para escribir documentos para servidores World Wide Web. Los lenguajes de marcas no son equivalentes a los lenguajes de programación aunque se definan igualmente como "lenguajes". Son sistemas complejos de

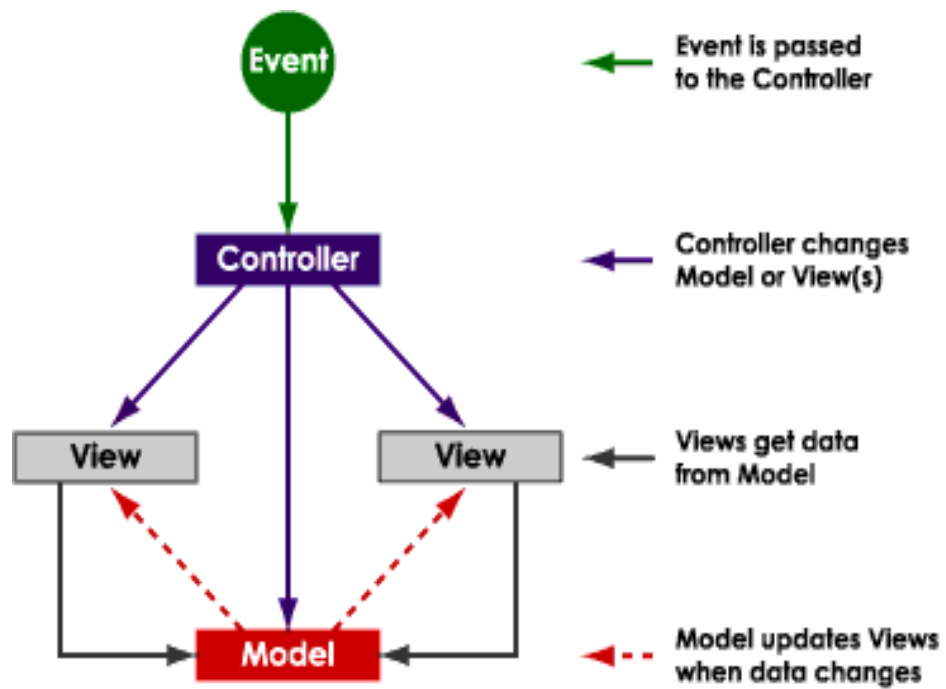
descripción de información, normalmente documentos, que se pueden controlar desde cualquier editor ASCII.

- ◆ HTTP: *HyperText Transfer Protocol*. Protocolo de Transferencia de Hipertextos. Modo de comunicación para solicitar páginas Web.
- ◆ Herramientas CASE: Herramientas utilizadas para el desarrollo de proyectos de Ingeniería de Software.
- ◆ Hardware: Componentes electrónicos, tarjetas, periféricos y equipo que conforman un sistema de computación; se distinguen de los programas (software) porque son tangibles.
- ◆ ISAPIs: *Internet Server API*. La API para servidores de Internet.
- ◆ JSP: *Java Server Pages*. 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.
- ◆ Linux: Es el nombre de un núcleo, pero se suele denominar con este nombre a un sistema operativo de libre distribución software libre, donde el código fuente está disponible públicamente y cualquier persona, con los conocimientos informáticos adecuados, puede libremente estudiarlo, usarlo, modificarlo y redistribuirlo.
- ◆ Microsoft: Compañía que manufactura los sistemas de operación DOS y Windows.
- ◆ MVC: *Modelo Vista Controlador*. Patrón de arquitectura de software.
- ◆ MPMs: Módulo de multiprocesamientos.
- ◆ MySQL: Es un sistema de gestión de bases de datos relacional que cuentan con todas las características de un motor de BD comercial: transacciones atómicas, triggers, replicación, llaves foráneas entre otras. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar.
- ◆ OASIS: Organization for the Advancement of Structured Information Standards. Consorcio internacional sin fines de lucro que orienta el desarrollo, la convergencia y la adopción de los estándares e-business.
- ◆ PHP: *Hypertext Preprocessor*. Es un ambiente script del lado del servidor que permite crear y ejecutar aplicaciones Web dinámicas e interactivas. Con PHP se pueden combinar páginas HTML y scripts.
- ◆ PostgreSQL: es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) libre.

- ◆ Perl: *Practical Extraction and Report Language*. Es un lenguaje de programación desarrollado por Larry Wall inspirado en otras herramientas de UNIX como son: sed, grep, awk, c-shell.
- ◆ RUP: *Rational Unified Process* (Proceso Unificado de desarrollo). Metodología para el desarrollo de Software.
- ◆ Web Service: (Servicio Web). Conjunto de documentos web en formatos tales como .aspx(ASP), .php(PHP), .pl(PERL), etc. accesible desde http como desde cualquier aplicación, su código es generalmente más extensivo que el de una simple web en html.
- ◆ SOAP: *Simple Object Access Protocol*. Protocolo utilizado en los servicios Web. Define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.
- ◆ Software: Programas de sistema, utilerías o aplicaciones expresados en un lenguaje de máquina.
- ◆ SQL: *Structured Query Language*. Es un lenguaje declarativo de acceso a bases de datos que permite especificar diversos tipos de operaciones sobre las mismas. Aúna características del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar información de interés de una base de datos.
- ◆ SGBD: *Sistema de Gestión de Bases de Datos*. Es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez.
- ◆ UCI: *Universidad de las Ciencias Informáticas*.
- ◆ UML: *Unified Modeling Language*. Es una notación estándar para modelar objetos del mundo real como primer paso en el desarrollo de programas orientados a objetos. Es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software.
- ◆ WEB (WWW): Red de documentos HTML intercomunicados y distribuidos entre servidores del mundo entero.
- ◆ WSDL: *Web Services Description Language*. Formato XML que se utiliza para describir servicios Web.
- ◆ XML: *Extensible Markup Language*. Es un lenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium. Orientado principalmente al almacenamiento, procesamiento y transmisión de mensajes.

Anexos

Anexo1. Funcionamiento del patrón MVC.



Anexo 2. Prototipo CU Solicitar Servicio Telemático.



Solicitudes de Servicios Telemáticos

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



[Inicio](#) | [Código de Ética](#) | [Políticas](#) | [Foro de la Comunidad](#) | [WikiProd](#) |

Solicitudes

- [Solicitud de Acceso Remoto](#)
- [Solicitud de Apertura de Puertos](#)
- [Solicitud de Cuenta](#)
- [Solicitud de Hosting Web](#)

Búsqueda de Solicitudes

Solicitud de Acceso Remoto



Nombre: EDUAR PALOMO GENÉ

Sexo: Masculino

No. Solapín: 7681

Área: FACULTAD #2

Cargo: PROFESOR

Dirección Particular

Teléfono desde el que realizará la conexión:

Tipo de Personal

- Rectoría
- Consejo Universitario
- Profesor de la UCI
- Profesor Adjunto
- Otros

Tipo de Servicios

- Acceso a la Intranet Universitaria
- Acceso al Correo Electrónico
- Sin Acceso a Internet
- Acceso a la Internet Cubana
- Acceso a la Navegación Básica
- Acceso total a Internet

(1.9220 segundos)

Anexo 3. Prototipo CU Gestionar Solicitud.



Solicitudes de Servicios Telemáticos

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



[Inicio](#) | [Código de Ética](#) | [Políticas](#) | [Foro de la Comunidad](#) | [WikiProd](#)

Directivo

[Solicitud de Acceso Remoto 1](#)

Reportes

[Por Tipo](#)

[Por Área](#)

[Búsqueda Avanzada](#)

Logout

[Salir](#)

[Cambiar passphrase](#)

Solicitud de Acceso Remoto



Nombre: YADILKA SUAREZ-INCLAN RIVER

Sexo: Femenino

No. Solapín: 7705

Área: FACULTAD #2

Cargo: VICEDECANO DE FORMACION

Dirección Particular

Teléfono desde el que realizará la conexión:

Tipo de Personal

Rectoría

Consejo Universitario

Profesor de la UCI

Profesor Adjunto

Otros

Tipo de Servicios

Acceso a la Intranet Universitaria

Acceso al Correo Electrónico

Sin Acceso a Internet

Acceso a la Internet Cubana

Acceso a la Navegación Básica

Acceso total a Internet

Aprobado

No Aprobado

Entre su Passphrase

Firmar

Cancelar

(1.9220 segundos)

Solicitudes Pendientes

Datos de las Solicitud	Línea de Aprobación
YADILKA 2007-06-26 00:33:08	 (Ver línea de Aprobación)
Aprobada por:  LESTER	Firma Válida
Datos en la BD no alterados	

Anexo 4. Prototipo CU Generar Reporte.



UCi **Solicitudes de Servicios Telemáticos**
UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Inicio | Código de Ética | Políticas | Foro de la Comunidad | WikiProd |

Directivo
Solicitud de Acceso Remoto 1

Reportes
Por Tipo
Por Área
Búsqueda Avanzada

Logout
Salir
Cambiar passphrase

Búsqueda Avanzada

Por Rango de tiempo
Desde: 01-01-2007 Hasta: 26-06-2007

Área
Escoger Area: Todo

Tipo de Servicio
Escoger tipo de servicio: Todo

Tipo de Servicio
Escoger Directivo: Todo

Solicitante
Escoger Solicitante: Todo

Buscar

(1.9220 segundos)

Reportes por Criterios

Resultado(s) de la Búsqueda

DARIAN HORACIO 2007-06-15 17:48:57
 DARIAN HORACIO 2007-06-21 15:01:30
 YADILKA 2007-06-26 00:33:08

Anexo 4. Prototipo CU Gestionar Servicio.

UCI Solicitudes de Servicios Telemáticos
UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Inicio | Código de Ética | Políticas | Foro de la Comunidad | WikiProd |

Gestionar Usuario
[Crear Usuario](#)
[Listar Usuario](#)

Gestionar Servicio
[Crear Servicio](#)
[Listar Servicios](#)

Reportes
[Por Tipo](#)
[Por Área](#)
[Búsqueda Avanzada](#)

Logout
[Salir](#)

Modificar Solicitud de Acceso Remoto

Datos del servicio

Nombre del servicio

Activo

Público

Registrado

Línea de aprobación

Modificar el nivel 1 2 3 [Agregar otro nivel](#)

(1.9220 segundos)

Nivel 1 de Solicitud de Acceso Remoto

Nombre	Usuario	Área	<input checked="" type="checkbox"/>	<input checked="" type="radio"/>
ELIURKIS	eliurkis	FACULTAD #10	<input checked="" type="checkbox"/>	<input checked="" type="radio"/>
LESTER	lesterr	FACULTAD #2	<input checked="" type="checkbox"/>	<input type="radio"/>
YADILKA	yadilka	FACULTAD #2	<input type="checkbox"/>	<input type="radio"/>

Anexo 5. Prototipo CU Gestionar Usuario.



Solicitudes de Servicios Telemáticos

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



[Inicio](#) | [Código de Ética](#) | [Políticas](#) | [Foro de la Comunidad](#) | [WikiProd](#) |

Gestionar Usuario

[Crear Usuario](#)

[Listar Usuario](#)

Crear Usuario



Nombre: DARIAN HORACIO GRASS BOADA

Sexo: Masculino

No. Solapín: 11518

Área: FACULTAD #2

Cargo: INSTRUCTOR RECIEN GRADUADO

Privilegios

Administrador

Directivo

Cargar solicitud con fichero(firma)

Passphrase

Rescriba Passphrase

Pass Admin

Administrador

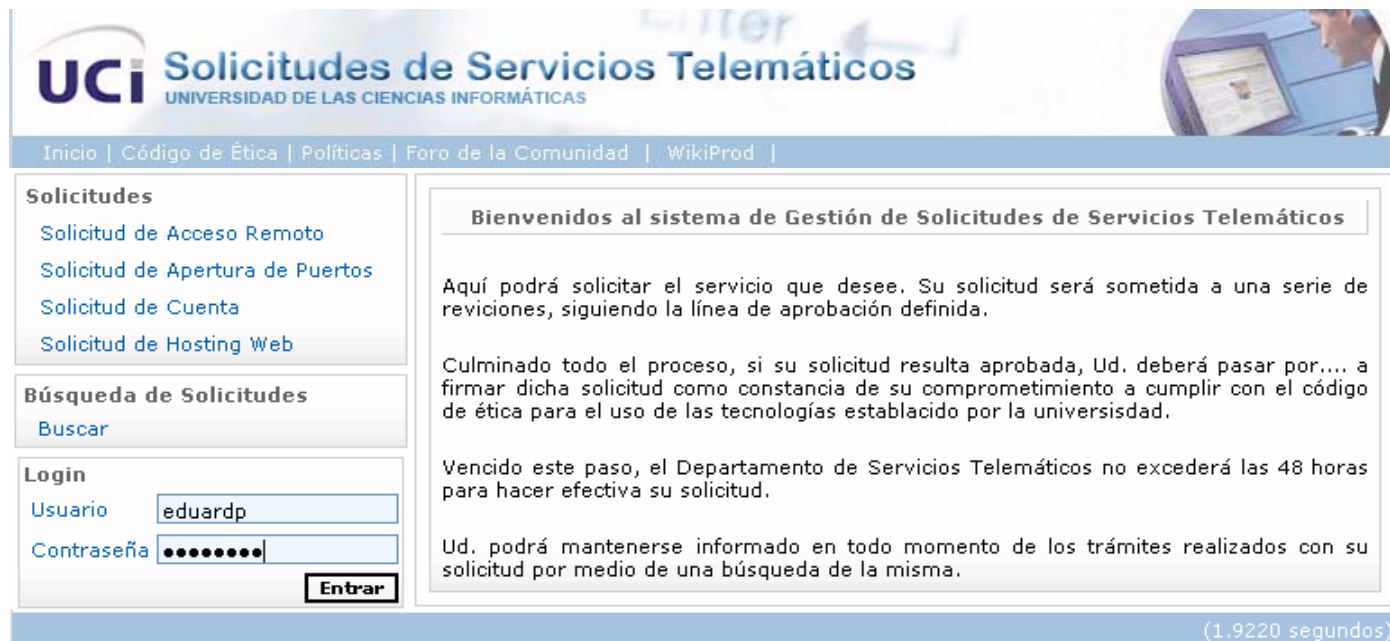
Crear

(1.9220 segundos)

Listar Usuarios

Nombre	Usuario	Rol		
Administrador	yrodriguezle	Administrador	✖	✍
EDUAR	eduardp	Directivo	🔒	✍
RAYDEL	raydelmp	Directivo/Admin	🔒	✍
ELIURKIS	eliurkis	Directivo/Admin	🔒	✍
LESTER	lesterr	Directivo	🔒	✍
YADILKA	yadilka	Directivo	🔒	✍
Administrador	jrodriguezg	Administrador	✖	✍

Anexo 6. Prototipo CU Autenticar Usuario.



The screenshot shows the login interface for the 'Solicitudes de Servicios Telemáticos' system. The header includes the UCI logo and navigation links. The main content area is divided into a left sidebar with menu items and a central login box. The login box contains fields for 'Usuario' (filled with 'eduardp') and 'Contraseña' (masked with dots), and an 'Entrar' button. A large message box on the right provides a welcome message and instructions regarding the request process and a 48-hour response time.

UCI Solicitudes de Servicios Telemáticos
UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Inicio | Código de Ética | Políticas | Foro de la Comunidad | WikiProd |

Solicitudes

- [Solicitud de Acceso Remoto](#)
- [Solicitud de Apertura de Puertos](#)
- [Solicitud de Cuenta](#)
- [Solicitud de Hosting Web](#)

Búsqueda de Solicitudes

[Buscar](#)

Login

Usuario

Contraseña

Entrar

Bienvenidos al sistema de Gestión de Solicitudes de Servicios Telemáticos

Aquí podrá solicitar el servicio que desee. Su solicitud será sometida a una serie de revisiones, siguiendo la línea de aprobación definida.

Culminado todo el proceso, si su solicitud resulta aprobada, Ud. deberá pasar por.... a firmar dicha solicitud como constancia de su comprometimiento a cumplir con el código de ética para el uso de las tecnologías establecido por la universidad.

Vencido este paso, el Departamento de Servicios Telemáticos no excederá las 48 horas para hacer efectiva su solicitud.

Ud. podrá mantenerse informado en todo momento de los trámites realizados con su solicitud por medio de una búsqueda de la misma.

(1.9220 segundos)