

Instituto Superior Politécnico “José Antonio Echeverría”

Facultad de Ingeniería Industrial  
Centro de Estudios de Ingeniería y Sistemas



**Universidad de las Ciencias Informáticas**



## **SISTEMA DE GESTIÓN DE SERVICIOS COMUNITARIOS**

**Trabajo de diploma para optar por el título de Ingeniería en Informática**

**Autores:** Yobannys Cabrera González  
Vladir Antonio Parrado Cruz

**Tutor:** Ing. José Fidalgo Hidalgo  
Dirección de Informatización  
Universidad de Ciencias Informáticas  
[josfh@uci.cu](mailto:josfh@uci.cu)

Ciudad de La Habana, Cuba  
Junio, 2006

*"En la vida hay algo peor que el fracaso: el no haber intentado nada."*

*Franklin D. Roosevelt.*

# *Agradecimientos*

*Agradezco:*

*A Maida, mi madre, por todo el amor y ternura que siempre me ha dado, por su apoyo incondicional aún en los momentos mas difíciles y por ser mi inspiración.*

*A Kenia, mi hermana, por toda su ayuda y por ser la niña de mis ojos.*

*A Yunior, mi hermano, por haber nacido y hacerme sentir orgulloso.*

*A Raúl, mi casi otro hermano, por estar presente cada vez que lo necesité.*

*A Vicente, mi padre, por ser más que padre, amigo.*

*A Litsandra, mi esposa, por todo su amor y...*

*A mis abuelos y tíos, por su cariño.*

*A Marcel por su sabiduría, a Pedro por sus berrinches, a ambos por su fútbol.*

*A mi nueva familia, Eva y Evelio, por aceptarme con mis defectos (cientos).*

*A Arturo, un gran amigo.*

*A José, mi tutor, por toda su ayuda y apoyo.*

*A todos los profesores de la Universidad de Camagüey que de una forma u otra ayudaron a mi formación como profesional, en especial a Geysel.*

*A Jeiser, Biasmey, Mandý, Vladir, Yunier, Dieter, Lester y demás compañeros de guerrilla.*

*A todos los que confiaron en este momento y a los que no, gracias.*

*Yobannys Cabrera González.*

*Agradezco:*

*A María del Carmen, mi mamá, por habérmelo dado todo en esta vida y principalmente en mis estudios, por todos sus consejos, por todo el amor que he recibido de ella y porque es lo mas grande que tengo.*

*A Yadir mi hermano, porque de una forma u otra me ha ayudado, y es mi hermano preferido.*

*A Humberto mi papá, por haberme dado su ayuda y porque se que me quiere mucho y es alguien especial para mi.*

*A Mariangela mi hermanita del alma por llenar de alegría mi tiempo.*

*A Miguel, mi tío por haberme ayudado tanto en mi carrera con sus consejos y además porque es al tío que más quiero, como se que me quiere a mi también.*

*A Alcides uno de mis hermanos de sentimiento que le debo mi decisión de estudiar esta carrera y que me ha ayudado a ser mejor y a comprender muchas cosas y por ser mi ídolo.*

*A Eberth otro de mis hermanos que quiero mucho, y que me ha enseñado a ser mucho de lo que soy actualmente, y me ha ayudado en todo momento que he necesitado su ayuda.*

*A Mailín una gran amiga que me ha ayudado mucho y a la que quiero con la vida.*

*A José, mi tutor, por toda su ayuda y apoyo.*

*A todos los profesores de la Universidad de Camagüey que de una forma u otra ayudaron a mi formación como profesional.*

*A Josué y su familia, Jeiser, Yunier, Yobannys y demás compañeros que compartieron conmigo todo este tiempo.*

*A los que no he mencionado no se sientan mal, ustedes también están aquí.*

*Vladir Antonio Parrado Cruz.*

## *Dedicatoria*

*A Kenia, que esté donde esté, siempre estará a mi lado.*

*Yobannys.*

*A María del Carmen como un granito de arena entre tantas cosas que se merece.*

*Vladir.*

## Resumen

El presente trabajo está encaminado a desarrollar e implantar un sistema para automatizar la prestación de servicios comunitarios en la UCI, así como sentar las bases para la migración del mismo a software libre.

La prestación de servicios en la UCI se resume como una secuencia de actividades que van desde la contratación de los servicios, la recepción de la solicitud del servicio, la planificación de las necesidades, la elaboración de los reportes y órdenes de trabajo de servicios, el aseguramiento de la calidad incluyendo los plazos de garantía así como el pago del servicio brindado.

Para lograr los objetivos se hace un estudio del estado del arte de sistemas similares existentes en el mundo y en Cuba específicamente, de las tecnologías y herramientas, así como del entorno del negocio. También un estudio preliminar para determinar las características del sistema, describir las etapas de análisis y diseño del ciclo de desarrollo del Sistema de Gestión de Servicios Comunitarios.

De esta forma se automatiza un proceso que en estos momentos tiene grandes problemas a la hora de llevarse a cabo, entre las principales razones encontramos que los clientes no saben precisamente a qué lugar dirigirse ni cómo hacer una solicitud de servicio, incluso ni acerca de la existencia de los servicios que se prestan en la UCI, la información estadística acerca de los problemas que han existido y los historiales de los reportes de trabajo se llevan a cabo manualmente, lo que equivale a una pérdida enorme de tiempo por parte del personal que labora en estos temas.

Como se aprecia existen deficiencias que son producidas tanto por el actuar del ser humano como por las condiciones reales de trabajo y por tanto ameritan la consideración para la realización de este trabajo.

# Índice

<b>INTRODUCCIÓN.....</b>	<b>14</b>
<b>1 FUNDAMENTACIÓN.....</b>	<b>19</b>
1.1 INTRODUCCIÓN.....	19
1.2 OBJETO DE ESTUDIO .....	19
1.2.1 <i>Objetivos estratégicos de la organización.</i> .....	19
1.2.2 <i>Flujo actual de los procesos.</i> .....	20
1.2.3 <i>Análisis crítico de la ejecución de los procesos.</i> .....	20
1.3 PROCESOS OBJETO DE AUTOMATIZACIÓN. ....	21
1.4 SISTEMAS AUTOMATIZADOS EXISTENTES VINCULADOS AL CAMPO DE ACCIÓN.....	22
1.4.1 <i>Software para la Gestión de Servicios Técnicos: STECNICO.</i> .....	22
1.4.2 <i>Solución para técnicos de soporte en servicios externos: EASYWORK.</i> .....	22
1.4.3 <i>Análisis de los sistemas relacionados a nuestro campo de acción.</i> .....	23
1.5 FUNDAMENTACIÓN DE LOS OBJETIVOS. ....	24
1.6 TENDENCIAS Y TECNOLOGÍAS ACTUALES .....	25
1.6.1 <i>.NET Framework</i> .....	25
1.6.2 <i>C-Sharp (C#)</i> .....	27
1.6.3 <i>ASP.NET.</i> .....	28
1.6.4 <i>Web Services.</i> .....	30
1.6.5 <i>NHibernate.</i> .....	31
1.6.6 <i>SQL</i> .....	32
1.6.7 <i>Metodología RUP (Rational Unified Process).</i> .....	32
1.6.8 <i>UML (Unified Modeling Language)</i> .....	33
1.6.9 <i>Fundamentación de la tecnología a utilizar.</i> .....	34
1.7 CONCLUSIONES .....	35
<b>2 MODELO DEL NEGOCIO.....</b>	<b>36</b>
2.1 INTRODUCCIÓN.....	36
2.2 MODELO DEL NEGOCIO ACTUAL .....	36
2.3 REGLAS DEL NEGOCIO A CONSIDERAR .....	36
2.3.1 <i>Conceptos asociados al modelo de negocios.</i> .....	37
2.4 ACTORES DEL NEGOCIO.....	37
2.5 DIAGRAMA DE CASOS DE USO DEL NEGOCIO.....	38
2.6 TRABAJADORES DEL NEGOCIO.....	38
2.7 CASOS DE USO DEL NEGOCIO .....	38

	<i>Tabla 3. Caso de uso: Realizar Solicitud</i> .....	38
	<i>Tabla 4. Caso de uso: Solicitar Estado</i> .....	40
2.8	MODELO DE OBJETOS .....	43
2.9	CONCLUSIONES .....	43
<b>3</b>	<b>REQUISITOS .....</b>	<b>44</b>
3.1	INTRODUCCIÓN.....	44
3.2	DEFINICIÓN DE LOS REQUISITOS FUNCIONALES. ....	44
3.3	DEFINICIÓN DE LOS REQUISITOS NO FUNCIONALES .....	47
3.4	ACTORES DEL SISTEMA A AUTOMATIZAR .....	48
3.5	DIAGRAMA DE CASOS DE USO DEL SISTEMA A AUTOMATIZAR .....	49
3.6	DESCRIPCIÓN DE LOS CASOS DE USO.....	50
3.7	CONCLUSIONES .....	62
<b>4</b>	<b>DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA .....</b>	<b>63</b>
4.1	INTRODUCCIÓN.....	63
4.2	DIAGRAMA DE CLASES DEL DISEÑO.....	63
	4.2.1 <i>Paquete Gestión de Servicios Comunitarios</i> .....	64
	4.2.2 <i>Paquete Acceso a Datos</i> .....	65
	4.2.3 <i>Paquete Entidades</i> . ....	69
	4.2.4 <i>Paquete Procesamiento Lógico</i> . ....	72
	4.2.5 <i>Paquete presentación</i> .....	75
4.3	PRINCIPIOS DE DISEÑO.....	88
	4.3.1 <i>Interfaz de usuario</i> .....	88
	4.3.2 <i>Formato de salida de los reportes</i> .....	90
	4.3.3 <i>Ayuda</i> .....	91
4.4	TRATAMIENTO DE ERRORES .....	91
4.5	DISEÑO DE LA BASE DE DATOS .....	92
	4.5.1 <i>Modelo lógico de datos</i> .....	92
	4.5.2 <i>Modelo físico de datos</i> .....	94
4.6	DIAGRAMA DE DESPLIEGUE.....	95
4.7	CONCLUSIONES .....	96
<b>5</b>	<b>ESTUDIO DE FACTIBILIDAD .....</b>	<b>97</b>
5.1	INTRODUCCIÓN.....	97
5.2	PLANIFICACIÓN BASADA EN CASOS DE USO .....	97
	5.2.1 <i>Cálculo de los Puntos de casos de uso desajustados</i> .....	97
	5.2.2 <i>Estimación de esfuerzo a través de los puntos de casos de uso</i> .....	100



5.3	BENEFICIOS TANGIBLES E INTANGIBLES .....	102
5.4	ANÁLISIS DE COSTOS Y BENEFICIOS.....	102
5.5	CONCLUSIONES .....	103
<b>CONCLUSIONES .....</b>		<b>104</b>
<b>RECOMENDACIONES .....</b>		<b>105</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>		<b>106</b>
<b>BIBLIOGRAFÍA .....</b>		<b>108</b>
<b>GLOSARIO DE TÉRMINOS .....</b>		<b>109</b>
<b>ANEXO I.....</b>		<b>I</b>
<b>ANEXO II .....</b>		<b>II</b>
<b>ANEXO III.....</b>		<b>III</b>
<b>ANEXO IV .....</b>		<b>IV</b>

## Índice de tablas

Tabla 1. Descripción de los actores del negocio.....	37
Tabla 2. Descripción de los trabajadores del negocio .....	38
Tabla 3. Caso de uso: Realizar Solicitud .....	38
Tabla 4. Caso de uso: Solicitar Estado .....	40
Tabla 5. Definición de actores del sistema a automatizar.....	48
Tabla 6. Descripción del caso de uso: Administrar áreas de servicio.....	50
Tabla 7. Descripción del caso de uso: Administrar los jefes del área de servicio.....	51
Tabla 8. Descripción del caso de uso: Administrar estados .....	51
Tabla 9. Descripción del caso de uso: Administrar técnicos del área de servicios.....	52
Tabla 10. Descripción del caso de uso: Administrar supervisores del área de servicios. .....	52
Tabla 11. Descripción del caso de uso: Administrar servicios del área de servicios. ....	53
Tabla 12. Descripción del caso de uso: Administrar medios del problema.....	53
Tabla 13. Descripción del caso de uso: Administrar problemas. ....	54
Tabla 14. Descripción del caso de uso: Ver solicitudes personales. ....	54
Tabla 15. Descripción del caso de uso: Realizar reclamación.....	55
Tabla 16. Descripción del caso de uso: Ver solicitudes del área de servicio.....	55
Tabla 17. Descripción del caso de uso: Generar orden de trabajo.....	56
Tabla 18. Descripción del caso de uso: Ver reportes .....	56
Tabla 19. Descripción del caso de uso: Autenticar usuario. ....	57
Tabla 20. Descripción del caso de uso: Generar certificación de trabajo. ....	57
Tabla 21. Descripción del caso de uso: Administrar materiales .....	58
Tabla 22. Descripción del caso de uso: Administrar Unidades de Medida .....	58
Tabla 23. Descripción del caso de uso: Administrar locales.....	59
Tabla 24. Descripción del caso de uso: Administrar usuarios. ....	59
Tabla 25. Descripción del caso de uso: Administrar grupos.....	60
Tabla 26. Descripción del caso de uso: Administrar usuarios de un grupo. ....	60
Tabla 27. Descripción del caso de uso: Administrar grupos de un grupo.....	61
Tabla 28. Descripción del caso de uso: Realizar solicitud.....	61
Tabla 29. Factor de peso de los actores sin ajustar. ....	97
Tabla 30. Factor de peso de los casos de uso sin ajustar .....	98
Tabla 31. Factor de complejidad técnica. ....	99

Tabla 32. Factor de ambiente. ....	100
Tabla 33. Esfuerzo del proyecto. ....	101

## Índice de figuras

Figura 1. Diagrama de casos de uso del negocio.....	38
Figura 2. Diagrama de actividad del caso de uso: Realizar Solicitud .....	40
Figura 3. Diagrama de actividad del caso de uso: Solicitar Estado .....	42
Figura 4. Modelo de objeto .....	43
Figura 5. Diagrama de los casos de uso de servicio. ....	49
Figura 6. Diagrama de los casos de uso de administración .....	50
Figura 7. Diagrama de clases del paquete GSC.....	64
Figura 8. Diagrama de clases del paquete Acceso a Datos. ....	65
Figura 9. Diagrama de clases del paquete Acceso WS.....	66
Figura 10. Diagrama de clases del paquete Proxys. ....	67
Figura 11. Diagrama de clases del paquete Referencias Web.....	68
Figura 12. Diagrama de clases del paquete Entidades .....	69
Figura 13. Diagrama de clases del paquete Entidades Seguridad .....	70
Figura 14. Diagrama de clases del paquete Entidades Servicios.....	71
Figura 15. Diagrama de clases del paquete Procesamiento Lógico.....	72
Figura 16. Diagrama de clases del paquete Procesamiento Lógico Servicios .....	73
Figura 17. Diagrama de clases del paquete Procesamiento Lógico Seguridad .....	74
Figura 18. Diagrama de clases del paquete presentación, módulo general. ....	75
Figura 19. Diagrama de clases del paquete presentación, módulo general, cambiar de usuario. ....	76
Figura 20. Diagrama de clases del paquete presentación, módulo navegación.....	77
Figura 21. Diagrama de clases del paquete presentación, módulo Administración General. ....	78
Figura 22. Diagrama de clases del paquete presentación, módulo Sistema, Administrar Áreas de Servicio.....	79
Figura 23. Diagrama de clases del paquete presentación, módulo Sistema, Administrar Locales.....	80
Figura 24. Diagrama de clases del paquete presentación, módulo Sistema, Administrar Estados, materiales y Unidades de medida .....	81
Figura 25. Diagrama de clases del paquete presentación, módulo Área de Servicios, Administrar Servicios que brinda el área. ....	82

Figura 26. Diagrama de clases del paquete presentación, módulo Área de Servicios, Administrar Supervisores y Técnicos.....	83
Figura 27. Diagrama de clases del paquete presentación, módulo Área de Servicios, Solicitudes hechas al área. ....	84
Figura 28. Diagrama de clases del paquete presentación, módulo Área de Servicios, Órdenes de Trabajo del Área.....	85
Figura 29. Diagrama de clases del paquete presentación, módulo Área de Servicios, Certificaciones de Trabajo del Área.....	86
Figura 30. Diagrama de clases del paquete presentación, módulo de Servicios, Nueva solicitud, Historial de solicitudes y reclamaciones. ....	87
Figura 31. Diagrama de clases del paquete presentación, módulo de Reportes. ....	88
Figura 32. Uso del color rojo para mensajes de error, operaciones incorrectas.....	89
Figura 33. Prototipo de los botones, botones para editar, adicionar y eliminar. ....	90
Figura 34. Ejemplo de un Reporte. ....	90
Figura 35. Acceso directo a la ayuda de la página actual.....	91
Figura 36. Utilización de mensajes de confirmación.....	91
Figura 37. Diagrama de despliegue .....	96

## Introducción

En la Universidad de las Ciencias Informáticas (UCI) existe hoy en día un gran volumen de personal. A su vez, existe también una serie de Servicios Comunitarios encaminados a resolver los problemas que allí se presentan, y a mejorar el sistema de vida de las personas que allí conviven.

En la actualidad, estos servicios son brindados por las áreas correspondientes de forma independiente. Hay áreas en las que existe un software para resolver en cierta medida el problema, mientras que otras no cuentan ni siquiera con eso.

Hoy en día, gestionar los servicios comunitarios en la UCI es una tarea difícil de desarrollar. Por otra parte en el centro se realiza todo el proceso de forma manual, por el **problema** de que no existe actualmente un software capaz de llevar el control de forma eficiente de las solicitudes que se hacen actualmente, y que a su vez, le brinde al usuario de forma fácil y sencilla la posibilidad de realizar una solicitud y seguir el estado de la misma de una forma eficiente. Como consecuencia existe la siguiente **situación problemática**: El proceso de solicitud de cualquier servicio se realiza vía telefónica o directamente en el departamento correspondiente, esto provoca que haya ineficiencia y demora en la atención a los servicios solicitados; tampoco existe un sistema de información para mantener a los usuarios al tanto de los posibles problemas presentados en las áreas de servicios, además la inexistencia de algún mecanismo que permita a los directivos conocer la situación de los servicios en cierto momento, o un historial de los problemas que se han ido presentando. También se torna difícil el proceso de control de los gastos materiales por cada área de servicio al tener que depender del área de contabilidad y hacer este manualmente. Los cambios que puedan sufrir determinados servicios brindados son desconocidos por la población de la UCI, al no contar con un medio que informe el estado de los mismos de forma dinámica. En otras palabras no existe un mecanismo que le brinde al usuario la posibilidad de solicitar cualquier servicio con una alta eficiencia, obteniendo una respuesta del estado de su solicitud, así como; que sea posible que el cliente pueda evaluar el trabajo finalizado ya, para que de esta forma las entidades puedan conocer el nivel de satisfacción del personal; además no existe un mecanismo que lleve el control de los gastos materiales de forma automática para la mayoría de las áreas que prestan servicios en la UCI.

Por todo esto, se hace necesario desarrollar un sistema que permita darle solución a los problemas aquí mencionados, y de esta forma eliminar el engorroso trabajo que hasta ahora se realiza.

Para ello se creará el **Sistema de Gestión de Servicios Comunitarios** el cual será capaz de satisfacer todas las necesidades anteriormente planteadas.

Para la implementación del mismo se ha decidido desarrollar una aplicación Web, debida a su fácil acceso, y además para que el usuario pueda acceder a la versión más reciente en todo momento, de forma tal que la aplicación siempre estará actualizada y a disposición de todos los que la necesiten.

Este trabajo tiene como antecedentes el trabajo de tesis de diploma del ingeniero José Fidalgo Hidalgo y Yudiel Alfredo Tamayo Agramonte, "Sistema de Gestión de Servicios Comunitarios" y para su realización fueron consultados los libros: UML y Patrones, Introducción al análisis y diseño orientado a objetos de Craig Larman y El Proceso Unificado de Desarrollo de Software, de los autores Ivar Jacobson, Grady Booch y James Rumbaugh, además de gran parte de la documentación sobre Ingeniería de Software que se encuentra en la red de la CUJAE, la UCI e Internet.

Con este sistema se espera automatizar todos los procesos que hasta ahora se hacían de forma manual, aumentando la confiabilidad y seguridad de los datos, proporcionando además facilidades a los usuarios del sistema, a través de una interfase amigable y sencilla la cual a su vez reduce los costos económicos que hasta ahora se hacían en materiales de oficina .

Las personas que realizan las solicitudes de un servicio comunitario en la UCI, son todas aquellas que trabajan en el centro (estudiantes, profesores, dirigentes, trabajadores), ya sean internos ó externos, mientras que las mismas son atendidas por el personal del área a que corresponda dicha solicitud (estos son designados por el jefe del área).

El Sistema de Gestión de Servicios Comunitarios le brindará al usuario la posibilidad de que este especifique los datos del problema que presente, estos pueden ser:

- Medios (medios asociado a un problema).
- Problema
- Local (Ubicación donde se encuentra el medio con problemas).

Además ofrecerá la posibilidad de seguir el estado de la solicitud hecha, así como un historial de todas las solicitudes hechas por el usuario en cualquier momento, permitiéndoles también, hacer reclamaciones en caso de no estar conforme con la solución a cierta solicitud.

El sistema ofrecerá reportes acerca del estado de la situación con los servicios comunitarios en todo momento, y de esta forma todas las estadísticas se realizan de forma automatizada, eliminando gran parte del trabajo que se realiza actualmente de forma manual.

El Sistema de Gestión de Servicios Comunitarios formará parte de la infraestructura interna de la UCI, ayudando en la toma de decisiones importantes dentro del centro.

El **objeto de estudio** de este trabajo esta enmarcado en lo referente a la gestión de servicios comunitarios en la UCI, de esta forma el **campo de acción** de este trabajo son todas las áreas que prestan servicios en esta institución.

Tenemos como **hipótesis** que si desarrollamos una aplicación con interfaz Web basada en un gestor de bases de datos potente con seguridad para el almacenamiento de datos como lo es SQL Server, y en una plataforma poderosa como lo es .NET; es posible lograr la gestión de servicios comunitarios en la UCI de una manera rápida, eficiente y segura, eliminando muchos problemas de seguridad y control presentes en la actualidad.

El **objetivo general** de este trabajo es realizar un software con interfaz Web, para la gestión de los servicios comunitarios en la UCI. Para ello nos hemos trazado los siguientes **objetivos específicos**: estudio detallado del funcionamiento de los servicios comunitarios, estudio sobre la tecnología necesaria y factible para la construcción del sistema que se propone, y realizar todo el proceso de desarrollo del sistema hasta la puesta en marcha.

Para cumplir con los objetivos, se trazaron las siguientes **tareas**:

- Estudio de las tendencias actuales que brindan solución a problemas similares.
- Fundamentación teórica del sistema.
- Reconocimiento de requisitos funcionales y confección de los casos de uso.
- Diseño del sistema



- Implementación del sistema
- Prueba e implantación del sistema.

Se pretende finalmente obtener un producto de software a la altura de las exigencias actuales de producción de software en nuestra sociedad, acorde con los estándares internacionales de catalogación y los estándares de diseño y presentación de aplicaciones Web [1].

El presente trabajo ha sido organizado de la siguiente forma:

**Capítulo 1:** Fundamentación, recoge los conceptos que se necesitan dominar para explicar en detalles cómo surge y de que se encarga el Sistema de Gestión de Servicios Comunitarios, así como las entidades que la forman y con las que se relaciona. También incluye como aspectos de actualidad una descripción de los lenguajes a utilizar para la implementación del sistema. Trata la situación de las tecnologías a utilizar en el desarrollo de la aplicación, se comparan y seleccionan las mejores propuestas para el trabajo, y se explican los conceptos principales que se van a tratar.

**Capítulo 2:** Modelo del negocio, describe el negocio a través de un Modelo de Negocio, se hace el análisis del sistema a desarrollar. Se definen las funcionalidades del sistema y se describen detalladamente, utilizando herramientas de modelación, los principales procesos del mismo.

**Capítulo 3:** Requisitos, muestra todo lo referente a los requisitos del sistema propuesto: los actores, los casos de uso, su estructuración por paquetes y las relaciones entre ellos.

**Capítulo 4:** Descripción de la solución propuesta, trata sobre la elaboración de la solución mediante diagramas de clases, se plantean los principios para el diseño y la implementación. Aquí se desglosan y explican las funcionalidades que se definieron en el capítulo anterior.

**Capítulo 5:** Estudio de factibilidad, es un estudio de la construcción del sistema, observando los beneficios tangibles e intangibles y analizando los costos del desarrollo de la propuesta.

# 1 Fundamentación

## 1.1 *Introducción*

El contenido de este capítulo constituye la base teórica del presente trabajo. En él se describen los principales conceptos y lineamientos, como resultado de la investigación realizada para la concepción del Sistema de Gestión de Servicios Comunitarios.

Se brinda una vista global de los temas relacionados con los Servicios Comunitarios, así como los principales conceptos asociados al dominio del problema, que son necesarios para entender el modelo de negocio y la propuesta de la solución. Además se realiza una comparación de las herramientas existentes y se determina cuáles van a ser las utilizadas en el sistema.

## 1.2 *Objeto de estudio*

El Objeto de estudio de este trabajo esta enmarcado en lo referente a la gestión de servicios comunitarios.

### 1.2.1 **Objetivos estratégicos de la organización.**

La UCI es un centro de estudios de nuevo tipo. Su principal reto es la formación de miles de jóvenes de todo el país como futuros profesionales en la rama de la informática.

Esta es actualmente una de las escuelas mas grandes de Cuba, en ella conviven más de 8000 personas y se tiene previsto que siga aumentando considerablemente la matrícula. Para atender todo tipo de problemas que puedan existir en las residencias, docentes, y demás locales, es decir en cualquiera de los lugares que existen actualmente en el centro se han creado una serie de áreas de servicios para darle respuesta y solución a dichos problemas, entre las que se encuentran mantenimiento constructivo, soporte de software, entre otras. Cada área en particular tiene la misión de dar respuesta de forma eficiente a las solicitudes que a ella llegan, garantizando la calidad del trabajo y la satisfacción de los clientes de acuerdo a sus necesidades.

### **1.2.2 Flujo actual de los procesos.**

El proceso de prestación de servicios se lleva de manera independiente en cada una de las áreas existentes, siendo similar en todas. Actualmente los procesos fluyen de la siguiente manera:

En la UCI existen una serie de áreas de servicios destinadas a la recepción y solución de las solicitudes de los clientes, estas solicitudes se pueden realizar o por vía telefónica o presentándose personalmente al área. El cliente puede realizar solicitudes por problemas en las cuales se deben especificar una serie de datos necesarios para la solución de su situación dentro de los cuales tenemos: el local donde se ha generado dicho problema, dígame local a la dirección física dentro de la universidad en que se encuentra dicho problema y el medio que tiene problemas, cuyos datos son anotados por el recepcionista que esté trabajando en ese momento, una vez realizado este proceso el usuario obtiene el número del reporte de su solicitud el cual debe conservar para en caso de que quiera averiguar por el estado en que se encuentra su solicitud, con este número puede obtener la información. Por otra parte el área se encarga de generar una certificación de trabajo y/o una orden de trabajo, en dependencia de lo que haga falta, la orden de trabajo se puede omitir, pues los datos que esta tiene son los concernientes a la dirección del local donde se encuentra el problema y la descripción del mismo, pero los técnicos conocen bien la localización de los locales de la UCI y la descripción la pueden obtener de la certificación de trabajo que sí debe generarse siempre debido a que esta lleva la constancia de los materiales que se emplearán en el trabajo y que justifica el uso de los mismos, por lo que no es totalmente imprescindible la generación de la orden de trabajo, pero en ocasiones es necesaria.

### **1.2.3 Análisis crítico de la ejecución de los procesos**

Los procesos de gestión de servicios que son llevados a cabo actualmente en la universidad, poseen una estructura de realización y seguimiento que no resuelven todos los problemas que se originan en el centro, ni lo hacen con la rapidez y la eficiencia que los usuarios exigen y necesitan.

Primeramente comencemos por las vías por las que se canalizan estos procesos, se realizan o por vía telefónica o por vía presencial, por lo que haciendo un análisis de la cantidad de personal existente en la UCI y la cantidad de medios, locales y trabajadores disponibles para dar solución a los servicios brindados, notamos que existe un gran desequilibrio, pues teniendo en cuenta que una solicitud como mínimo tiene un

ciclo de vida desde que se inicia el proceso hasta que es resuelto el problema de 2 a 3 días en cada una de las áreas , y que cada área como promedio recibe de 30 a 33 solicitudes por día, tenemos una razón de culminación diaria de 11 a 15 solicitudes, por lo que solo se resuelven un poco menos del 50% de los problemas, y que para conocer el estado en el que se encuentra la solicitud solo puede saberse llamando por teléfono o de forma presencial presentando el número de reporte; al conjugarse el personal que realiza nuevas solicitudes con el que reclama sus servicios la situación se torna un caos, es imposible atenderlos a todos al mismo tiempo, pues no tenemos la suficiente cantidad de líneas telefónicas disponibles para ello(casi siempre están ocupados los teléfonos) ni la cantidad de personal, además de que estas solicitudes tienen que ser manuscritas y el usuario tiene que conservar de alguna manera el número del reporte de su solicitud, es algo engorroso la forma en que se hacen estos procesos en la actualidad; por otra parte contamos con una red de gran velocidad, además disponemos de una gran capacidad de almacenamiento, suficiente como para a través de una aplicación como la que proponemos se satisfaga cualquier tipo de requerimiento en cuanto a prestación de servicios se trata y de la forma más eficiente posible. Es necesario señalar que cada área de servicios necesita saber los gastos materiales en determinados momentos, estos se llevan de forma manual basándose en las certificaciones de trabajo, lo cual es verdaderamente trabajoso para el personal de dichas áreas, con la realización de nuestra propuesta este problema quedaría resuelto completamente al contar con reportes que automáticamente realizarán todos los cálculos concernientes al área que corresponda.

### ***1.3 Procesos objeto de automatización.***

Se desean automatizar los procesos, que son los que ayudarán a resolver las disyuntivas que se plantean en este trabajo. En primera instancia la realización de solicitudes por el usuario, la solicitud se podrá ver en cualquier momento no solo el estado de su solicitud, sino también el listado de todas las solicitudes realizadas por él y los posibles estados que han tenido, así como realizar reclamaciones a solicitudes y dar su opinión de la calidad del trabajo realizado. Por parte de las áreas de servicio el sistema debe permitir emitir órdenes de trabajo y generar certificaciones de trabajo, además de permitir la obtención de los gastos de materiales que se llevan hasta un momento determinado y cuando lo deseen.

Como primer paso se pretende dar culminación a las actividades que hacen funcional y extensible al sistema, de manera que sea flexible a cambios estructurales en el centro y a múltiples interfaces, haciendo posible el trabajo descentralizado y concurrente y garantizando la seguridad e integridad de la información. El objetivo final es obtener un sistema que englobe todas las actividades de los procesos, que brinde además una interfaz de configuración con facilidades para la utilización de los usuarios y por parte de las áreas de servicio, para así lograr automatizar de la forma más idónea la UCI.

## ***1.4 Sistemas automatizados existentes vinculados al campo de acción***

### **1.4.1 Software para la Gestión de Servicios Técnicos: STECNICO.**

Este sistema se utiliza para SAT (Servicios de Asistencia Técnica) de todo tipo de equipos electrodomésticos. STECNICO gestiona todo lo referente a talleres de servicios técnicos SAT, gestiona la reparación de electrodomésticos; genera los números de reparación desde el 10.000 hasta 9.999.999[1]; imprime resguardo para el cliente y hoja para el taller al entrar una reparación; busca las reparaciones por ocho campos distintos; permite corregir cualquier dato en cualquier momento; lista las reparaciones pendientes de presupuestos; reporta los totales facturados a un mismo cliente; controla las reparaciones para ser entregadas; lista las facturas emitidas entre dos fechas; lista las reparaciones en espera de presupuestos visibles en unos segundos; devuelve un listado de reparaciones por operario; realiza confecciones de presupuestos y facturas; contiene una base de datos para componentes y/o repuestos, este se añade al presupuesto con una pulsación; contiene una base de datos para clientes; brinda la opción de elegir el N° de código de cliente, manual o automático; el programa genera un número de factura automáticamente; realiza la impresión en papel de presupuestos y facturas con total claridad. [19]

### **1.4.2 Solución para técnicos de soporte en servicios externos: EASYWORK.**

Permite agilizar la transferencia de información de soporte de campo, y organizar los reportes de las actividades ejecutadas.

Este sistema se aplica para los diferentes tipos de agentes de soporte y asistencia técnica. Los tipos de usuarios que se benefician del sistema EASYWORK son:

Técnico de soporte de manutención: atiende llamados, es decir, se le llaman por móvil, vía voz, datos o por SMS.

Técnico en prevención de manutención: ejecuta labores programadas, ya sea para la sustitución de piezas o para la limpieza, calibración o ajuste de los equipos.

Agente de campo: provee materiales que son vendidos, como máquinas de venta, freezers, post-mix y otros.

EASYWORK está compuesto por dos softwares que controlan las actividades de campo:

EASYWORK-Mobile, para técnicos de campo. Los usuarios de este módulo son los técnicos de campo que dan soporte correctivo y manutención preventiva de los equipos, y otras actividades relacionadas al control de equipos y materiales de repuesto.

EASYWORK-Server, sirve de interfaz entre el módulo EASYWORK-Mobile y los sistemas existentes, en el cliente, tal como el de Control de Equipos y el de Control de las Actividades de Asistencia Técnica. En caso que el cliente no posea estos sistemas, se podrá ejecutar los controles por el EASYWORK-Server.

El sistema EASYWORK tiene como funciones las comentadas a continuación:

Proceso de preparación de datos: Planifica las actividades de soporte y otros servicios externos. En caso de soporte correctivo, abre un llamado para que se envíe al técnico.

Transmisión de datos: La transmisión de llamados de soporte correctivo, se puede hacer por teléfono convencional o por SMS. Los demás datos, se pueden transferir por el mismo medio o por medio de sincronización de datos.

Proceso de selección de tarea ejecutables y registro de las actividades: El sistema registra y controla tanto el tiempo de desplazamiento y ejecución, como los kilómetros recorridos en cada trayecto.

Permite también, registrar datos cualitativos, obtenidos de las mediciones de parámetros y también de ajustes y arreglos, hechos en equipos especializados.

### **1.4.3 Análisis de los sistemas relacionados a nuestro campo de acción.**

Después de haber analizado los softwares que tienen similitud con nuestros requerimientos y necesidades, podemos percatarnos de que tienen varias

características que de cierta manera se asemejan a nuestra situación pero que no satisfacen en su totalidad nuestro proceso. STECNICO gestiona todo lo referente a talleres de servicios técnicos SAT, pero solamente gestiona la reparación de electrodomésticos; es una aplicación de escritorio, por lo que imposibilita la realización de solicitudes por parte del cliente y que de forma automática pueda ser atendida, que es la principal funcionalidad del sistema que se requiere. EASYWORK por su parte permite la transmisión de datos por teléfono convencional o por SMS, pero solo se limita a ellas, imposibilitando la realización de solicitudes vía Web. Estos sistemas no dan la flexibilidad de localizar el lugar donde se presente un problema, además de que son más específicos para los servicios técnicos, nuestros requerimientos son mucho más amplios. Por lo que concluimos en que es necesaria e imprescindible la realización de un nuevo sistema que cumpla con las características requeridas para satisfacer lo concerniente a la gestión de servicios comunitarios en la universidad.

### ***1.5 Fundamentación de los objetivos.***

Partiendo de que el proceso de solicitud de cualquier servicio se realiza vía telefónica o directamente en el departamento correspondiente, provocando que haya ineficiencia y demora en la atención a los servicios solicitados; y que tampoco existe un sistema de información para mantener a los usuarios al tanto de los posibles problemas presentados en las áreas de servicios, además la inexistencia de algún mecanismo que permita a los directivos conocer la situación de los servicios en cierto momento, o un historial de los problemas que se han ido presentando y que se torna difícil el proceso de control de los gastos materiales por cada área de servicio al tener que depender del área de contabilidad y hacer este manualmente, los cambios que puedan sufrir determinados servicios brindados son desconocidos por la población de la UCI, al no contar con un medio que informe el estado de los mismos de forma dinámica, en otras palabras no existe un mecanismo que le brinde al usuario la posibilidad de solicitar cualquier servicio con una alta eficiencia, obteniendo una respuesta del estado de su solicitud, así como; que sea posible que el cliente pueda evaluar el trabajo finalizado ya, para que de esta forma las entidades puedan conocer el nivel de satisfacción del personal; además de que no existe un mecanismo que lleve el control de los gastos materiales de forma automática para la mayoría de las áreas que prestan servicios en la UCI; si desarrollamos una aplicación con interfaz Web basada en un gestor de



bases de datos potente con seguridad para el almacenamiento de datos como lo es SQL Server, y en una plataforma poderosa como lo es .NET; es posible lograr la gestión de servicios comunitarios en la UCI de una manera rápida, eficiente y segura, eliminando muchos problemas de seguridad y control presentes en la actualidad para lo que se nos hace imprescindible y necesario trazarnos los siguientes objetivos: realizar un estudio detallado del funcionamiento de los servicios comunitarios, la tecnología necesaria y factible para la construcción del sistema que se propone, y realizar todo el proceso de desarrollo del sistema hasta la puesta en marcha.

## **1.6 Tendencias y tecnologías actuales**

Una variada gama de tecnologías posibilita brindar una solución efectiva en las aplicaciones que requieren máxima seguridad, robustez y flexibilidad de programación. Para el desarrollo de esta aplicación utilizaremos las siguientes.

### **1.6.1 .NET Framework**

**Microsoft.NET** es el conjunto de nuevas tecnologías en las que Microsoft ha estado trabajando durante los últimos años, con el objetivo de obtener una plataforma sencilla y potente para distribuir el software en forma de servicios que puedan ser suministrados remotamente y que puedan comunicarse y combinarse unos con otros de manera totalmente independiente de la plataforma, lenguaje de programación y modelo de componentes con los que hayan sido desarrollados. Ésta es la llamada *Plataforma.NET (.NET Framework)*, y a los servicios antes comentados se les denomina *servicios Web (Web Services)*. [10]

El **.NET Framework** es un entorno de desarrollo y ejecución, que permite a diferentes lenguajes trabajar en conjunto para crear aplicaciones fáciles de construir, administrar, desplegar e integrar con otros sistemas.

El **.NET Framework** consiste en:

- Common Language Runtime (CLR).
- Framework Class Library (FCL).

### 1.6.1.1 Common Language Runtime (CLR).

Es el núcleo de la plataforma .NET. Es el motor encargado de gestionar la ejecución de las aplicaciones para ella desarrolladas y a las que ofrece numerosos servicios que simplifican su desarrollo y favorecen su fiabilidad y seguridad. Las principales características y servicios que ofrece el CLR son:

- Modelo de programación consistente.
- Modelo de programación sencillo.
- Eliminación del “infierno de las DLLs.
- Ejecución multiplataforma.
- Integración de lenguajes.
- Gestión de memoria.
- Seguridad de tipos.
- Aislamiento de procesos.
- Tratamiento de excepciones.
- Seguridad avanzada.
- Interoperabilidad con código antiguo.

### 1.6.1.2 Framework Class Library (FCL).

El FCL es una librería incluida en el *.NET Framework*, formada por cientos de tipos de datos que permiten acceder a los servicios ofrecidos por el CLR y a las funcionalidades más frecuentemente usadas a la hora de escribir programas. Además, a partir de estas clases prefabricadas el programador puede crear nuevas clases que mediante herencia extiendan su funcionalidad y se integren a la perfección con el resto de clases de la FCL. Por ejemplo, implementando ciertos interfaces podemos crear nuevos tipos de colecciones que serán tratadas exactamente igual que cualquiera de las colecciones incluidas en la FCL.

Esta librería está escrita en MSIL, por lo que puede usarse desde cualquier lenguaje cuyo compilador genere MSIL. A través de las clases suministradas en ella es posible

desarrollar cualquier tipo de aplicación, desde las tradicionales aplicaciones de ventanas, consola o servicio de Windows NT hasta los novedosos servicios Web y páginas ASP.NET. Es tal la riqueza de servicios que ofrece que puede crearse lenguajes que carezcan de librería de clases propia y sólo usen la FCL - como C# [11].

### 1.6.2 C-Sharp (C#).

C# es un lenguaje de programación que toma las mejores características de lenguajes existentes como *Visual Basic*, *Java* y *C++*, combinándolas en uno solo. El hecho de ser relativamente reciente no implica que sea inmaduro, pues *Microsoft* ha escrito la mayor parte de la FCL usándolo, por lo que su compilador es el más depurado y optimizado de los incluidos en el *.NET Framework SDK*. [4]

C# (leído en inglés “C Sharp” y en español “C Almohadilla”) es el nuevo lenguaje de propósito general diseñado por *Microsoft* para su plataforma *.NET*. Sus principales creadores son **Scott Wiltamuth** y **Anders Hejlsberg**, éste último también conocido por haber sido el diseñador del lenguaje Turbo Pascal y la herramienta RAD Delphi. [20]

Aunque es posible escribir código para la plataforma *.NET* en muchos otros lenguajes, C# es el único que ha sido diseñado específicamente para ser utilizado en ella, por lo que programarla usando este es mucho más sencillo e intuitivo que hacerlo con cualquiera de los otros lenguajes.

La sintaxis y estructuración de C# es muy similar a la C++, ya que la intención de *Microsoft* es facilitar la migración de códigos escritos en estos lenguajes a C# y facilita su aprendizaje a los desarrolladores habituados a ellos. Sin embargo, su sencillez y el alto nivel de productividad son equiparables a los de *Visual Basic*.

#### Características principales del C#

- Sencillez.
- Modernidad.
- Orientación a componentes.
- Gestión automática de memoria.

- Seguridad de tipos.
- Instrucciones seguras.
- Sistema de tipos unificado.
- Extensibilidad de tipos básicos.
- Extensibilidad de operadores.
- Versionable.
- Eficiente.
- Compatible.

### 1.6.3 ASP.NET.

ASP.NET es un marco de trabajo de programación generado en CLR que puede utilizarse en un servidor para generar eficaces aplicaciones Web. ASP.NET ofrece varias ventajas importantes acerca de los modelos de programación Web anteriores [7]:

- **Mejor rendimiento.** *ASP.NET* es un código de CLR, compilado que se ejecuta en el servidor. A diferencia de sus predecesores, *ASP.NET* puede aprovechar las ventajas del enlace anticipado, la compilación *just-in-time*, la optimización nativa y los servicios de caché desde el primer momento. Esto supone un incremento espectacular del rendimiento antes de siquiera escribir una línea de código.
- **Compatibilidad con herramientas de primer nivel.** El marco de trabajo de ASP.NET se complementa con un diseñador y una caja de herramientas muy completos en el entorno integrado de desarrollo (*Integrated Development Environment*, IDE) de Visual Studio. La edición WYSIWYG, los controles de servidor de arrastrar y colocar y la implementación automática son sólo algunas de las características que proporciona esta eficaz herramienta.
- **Eficacia y flexibilidad.** Debido a que ASP.NET se basa en CLR, la eficacia y la flexibilidad de toda esa plataforma se encuentra disponible para los programadores de aplicaciones Web. La biblioteca de clases de *.NET Framework*, la Mensajería y las soluciones de Acceso a datos, se encuentran

accesibles desde el Web de manera uniforme. ASP.NET es también independiente del lenguaje, por lo que puede elegir el lenguaje que mejor se adapte a la aplicación o dividir la aplicación en varios lenguajes. Además, la interoperabilidad de CLR garantiza que la inversión existente en programación basada en COM se conserva al migrar a ASP.NET.

- **Simplicidad.** ASP.NET facilita la realización de tareas comunes, desde el sencillo envío de formularios y la autenticación del cliente hasta la implementación y la configuración de sitios. Por ejemplo, el marco de trabajo de página de ASP.NET permite generar interfaces de usuario, que separan claramente la lógica de aplicación del código de presentación, y controlar eventos en un sencillo modelo de procesamiento de formularios de tipo Visual Basic. Además, CLR simplifica la programación, con servicios de código administrado como el recuento de referencia automático y el recolector de elementos no utilizados.
- **Facilidad de uso.** ASP.NET emplea un sistema de configuración jerárquico, basado en texto, que simplifica la aplicación de la configuración al entorno de servidor y las aplicaciones Web. Debido a que la información de configuración se almacena como texto sin formato, se puede aplicar la nueva configuración sin la ayuda de herramientas de administración local. Esta filosofía de "administración local cero" se extiende asimismo a la implementación de las aplicaciones ASP.NET Framework. Una aplicación ASP.NET Framework se implementa en un servidor sencillamente mediante la copia de los archivos necesarios al servidor. No se requiere el reinicio del servidor, ni siquiera para implementar o reemplazar el código compilado en ejecución.
- **Escalabilidad y disponibilidad.** ASP.NET se ha diseñado teniendo en cuenta la escalabilidad, con características diseñadas específicamente a medida, con el fin de mejorar el rendimiento en entornos agrupados y de múltiples procesadores. Además, el motor de tiempo de ejecución de ASP.NET controla y administra los procesos de cerca, por lo que si uno no se comporta adecuadamente (filtraciones, bloqueos), se puede crear un proceso nuevo en su lugar, lo que ayuda a mantener la aplicación disponible constantemente para controlar solicitudes.

- **Posibilidad de personalización y extensibilidad.** ASP.NET presenta una arquitectura bien diseñada que permite a los programadores insertar su código en el nivel adecuado. De hecho, es posible extender o reemplazar cualquier sub-componente del motor de tiempo de ejecución de ASP.NET con su propio componente escrito personalizado. La implementación de la autenticación personalizada o de los servicios de estado nunca ha sido más fácil.
- **Seguridad.** Con la autenticación de Windows integrada y la configuración por aplicación, se puede tener la completa seguridad de que las aplicaciones están a salvo.

#### 1.6.4 Web Services.

Los *XML Web Services* permiten que las aplicaciones compartan información y que además invoquen funciones de otras aplicaciones independientemente de cómo se hayan creado las aplicaciones, cuál sea el sistema operativo o la plataforma en que se ejecutan y cuales son los dispositivos utilizados para obtener acceso a ellas. Aunque los *XML Web Services* son independientes entre sí, pueden vincularse y formar grupos de colaboración para realizar tareas determinadas [3].

Para integración, los “*Servicios Web XML*” sobre *.NET* hacen posible que diferentes piezas de software trabajen en conjunto para:

- Unir aplicaciones. Convertir aplicaciones independientes a constelaciones de aplicaciones para usar datos reales.
- Intercambiar datos. Los datos de clientes residen en aplicaciones aisladas, impidiendo ofrecer nuevos servicios que exploten esos datos.

¿Para qué sirve un *Web Service*? La respuesta puede ser otra pregunta: ¿Para qué sirve en programación una rutina? Todos sabemos que una rutina es como una caja negra, que encierra cierto proceso o algoritmo, y que cumple una función clara. Muchas rutinas y un guión central componen un programa en lo que se llama "programación estructurada". Un *Web Service* viene a ser una rutina en Internet.

¿Por qué se llama "*Web Service*" y no "*Rutina en Internet*"? Los protocolos que soportan los *Servicio Web* se comunican normalmente por el puerto 80, y basándose en *HTTP*, métodos *GET* y *POST*. Esto hace que podamos acceder a ellos al igual que lo

hacemos en una página Web. La diferencia entre una página Web y un Web Service, es que la página la visita cualquier individuo interesado, mientras que el servicio sólo lo visitan programas que lo requieren.

De modo, que el conjunto de *Web Services* en Internet es una *World Wide Web* paralela, de carácter no humano, sino cibernético. Veamos, que los ordenadores ya hablan solos a través de Internet.

Los *Web Services* se actualizan de forma transparente para el programador y para el encargado de mantenimiento de la aplicación. Además, mediante un *Web Services* puedes implementar a tu programa funciones imposibles de contemplar bajo el uso de rutinas de librerías, como por ejemplo, incorporar un buscador de páginas Web. Por otro lado, la carga de CPU que supone la ejecución de una rutina, desaparece al usar *Web Services*. La carga se reparte por Internet, sobre el servidor del *Web Services*. Esto es un comienzo de "Computación Distribuida".

Los *Web Services* permiten a los usuarios usar aplicaciones que comparten datos con otros programas modulares. Son aplicaciones independientes de la plataforma que pueden ser fácilmente publicadas, localizadas e invocadas mediante protocolos Web estándar, como XML. El objetivo final es la creación de un directorio *on-line* de *Web Services*, que pueda ser localizado de un modo sencillo y que tenga una alta fiabilidad. La integración de aplicaciones hará posible obtener información demandada en tiempo real, acelerando el proceso de toma de decisiones [13].

### 1.6.5 NHibernate.

NHibernate es un framework de Object-Relational-Mapping open-source que resuelve en forma automática la persistencia de mis objetos de dominio .NET. NHibernate esta basado en el popular framework open-source Hibernate surgido en la comunidad Java en el año 2002. Este framework realiza el mapeo entre el mundo orientado a objetos de las aplicaciones y el mundo entidad-relación de las bases de datos en entornos .Net. El término utilizado es **ORM** (object/relational mapping) y consiste en la técnica de realizar la transición de una representación de los datos de un modelo relacional a un modelo orientado a objetos y viceversa.

NHibernate no solo realiza esta transformación sino que nos proporciona capacidades para la obtención y almacenamiento de datos de la base de datos que nos reducen el

tiempo de desarrollo. NHibernate nos proporciona además un lenguaje con el que realizar consultas a la base de datos. Este lenguaje es similar a SQL y es utilizado para obtener objetos de la base de datos según las condiciones especificadas en el **HQL**.

El uso de **HQL** nos permite usar un lenguaje intermedio que según la base de datos que usemos y el dialecto que especifiquemos será traducido al SQL dependiente de cada base de datos de forma automática y transparente [21].

### 1.6.6 SQL.

*SQL Server* es el eje principal de la administración y análisis de datos de la siguiente generación de productos y servicios de *Microsoft .NET*.

Es una solución integral de base de datos y análisis. *SQL Server* ofrece el rendimiento, escalabilidad y confiabilidad que requiere la Web y los entornos empresariales de línea de negocios. El nuevo soporte de *XML* y *HTTP* simplifica el acceso a datos y el intercambio, mientras que las poderosas capacidades de análisis mejoran el valor de los datos. Las características de disponibilidad mejoradas maximizan el tiempo de actividad, las funciones de la administración avanzadas automatizan las tareas rutinarias y las herramientas mejoradas de programación y los servicios aceleran el desarrollo.

### 1.6.7 Metodología RUP (Rational Unified Process)

Para desarrollar un software se necesita una forma coordinada de trabajo. Un proceso que integre las múltiples facetas del desarrollo. Se necesita un método común, un proceso que:

- Proporcione una guía para ordenar las actividades de un equipo.
- Dirija las tareas de cada desarrollador por separado y del equipo como un todo.
- Especifique los artefactos que deben desarrollarse.
- Ofrezca criterios para el control y la medición de los productos y actividades de proyectos.
- El Proceso Unificado de Desarrollo es una solución al problema del software.



“El Proceso Unificado es un proceso de desarrollo de Software. O sea, es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Sin embargo, el Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organización, diferentes niveles de aptitud y diferentes tamaños de proyecto. El Proceso Unificado está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas.”

“El Proceso Unificado utiliza el Lenguaje Unificado de Modelado (Unified Modeling Language, **UML**) para preparar todos esquemas de un sistema software , De hecho, UML, es una parte esencial del Proceso Unificado – sus desarrollos fueron paralelos.” **[2]**.

No obstante, los verdaderos aspectos definitorios del Proceso Unificado se resumen en tres fases claves – dirigido por caso de uso, centrado en la arquitectura e iterativo e incremental. Esto es lo que hace único al Proceso Unificado.

### **1.6.8 UML (Unified Modeling Language)**

“UML son las siglas de Unified Modeling Language (Lenguaje Unificado de Modelado), notación (esquemática en su mayor parte) con que se construyen sistemas por medio de conceptos orientados a objetos”. **[2]**

“El UML (Lenguaje Unificado para la Construcción de Modelos) se define como un lenguaje que permite especificar, visualizar y construir los artefactos de los sistemas de software...” **[2]**. “Es un sistema notacional (que, entre otras cosas, incluye el significado de sus notaciones) destinado a los sistemas de modelado que utilizan conceptos orientados a objetos”.

EL UML (Lenguaje Unificado de Modelado) es una de las herramientas más atractivas y utilizadas en el mundo del desarrollo de software, esto se debe a que permite a los desarrolladores de sistemas generar diseños que capturen sus ideas en una forma convencional y comunicarlas a terceras personas. Básicamente el UML es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende

el desarrollo de software. UML entrega una forma de modelar cosas conceptuales como lo son procesos de negocio y funciones de sistema, además de cosas concretas como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables.

El UML prescribe un conjunto de notaciones y diagramas estándares para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación.

“El UML es un lenguaje para construir modelos; no guía al desarrollador en la forma de realizar el análisis y diseño orientados a objetos ni le indica cual proceso de desarrollo a adoptar.” [2]

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. Diagramas de Casos de Uso para modelar los procesos del negocio.

Además UML prescribe una notación estándar y semántica esencial para el modelado de un sistema orientado a objetos. Previamente, un diseño orientado a objetos podría haber sido modelado con cualquiera de la docena de metodologías populares, causando a los revisores tener que aprender las semánticas y notaciones de la metodología empleada antes que intentar entender el diseño en sí. Ahora con UML, diseñadores diferentes modelando sistemas diferentes pueden sobradamente entender cada uno los diseños de los otros [17].

En resumen UML es el resultado de la unión de tres metodologías, *Booch*, *OMT*, y *OOSE*. Estas han tenido una aplicación extensa en el campo del la POO, tienen su historia, y han sido aplicadas en una gran variedad de industrias y problemas, por lo que pueden ser clasificadas como muy maduras.

### **1.6.9 Fundamentación de la tecnología a utilizar.**

Teniendo en cuenta las características de la tecnología anteriormente explicada podemos resumir en que usaremos la plataforma .NET como entorno de desarrollo primeramente porque la misma permite un rápido desarrollo de aplicaciones y el

servicio que se pretende automatizar es de máxima prioridad en el centro, además esta plataforma nos brinda un lenguaje de programación (C#) sencillo y con alto nivel de productividad, un marco de trabajo para aplicaciones Web (ASP.Net) que genera eficaces aplicaciones y por último, esta plataforma permite suministrar servicios remotamente, los que a su vez pueden comunicarse y combinarse unos con otros de manera totalmente independiente de la plataforma, lenguaje de programación y modelo de componentes con los que hayan sido desarrollados, este aspecto es de gran importancia pues evitaremos repetir datos que se encuentran en otras bases de datos del centro y que ocupan gran espacio, como es el caso de la información referente al personal, a los medios y a los locales. Debido a la naturaleza de los datos que se almacenarán en el sistema propuesto, se ha decidido utilizar Microsoft SQL Server 2000 como gestor de base de datos, por ser una aplicación poderosa, robusta, que permite gran seguridad de los datos, ostenta marcas de referencia en cuanto a escalabilidad y confiabilidad, que son críticas para el éxito de bases de datos de gran tamaño; se ha decidido el uso del NHibernate framework pues este nos permitirá cambiar de gestor con gran facilidad al no tener que utilizar lenguajes específicos de consultas en los gestores, lo cual nos será de gran importancia a la hora de migrar a software libre. La metodología RUP ha sido la elegida para el desarrollo del software pues además de ser puntera a nivel mundial permite una mejor interacción con el usuario en todo el proceso de desarrollo software, lo cual nos puede ayudar a maximizar en tiempo y calidad el desarrollo del sistema propuesto [9].

## **1.7 Conclusiones**

En el presente capítulo se han analizado las tecnologías actuales y se profundizó en algunos conceptos necesarios para la comprensión de la solución de este trabajo. Además se ha fundamentado la elección de cuales herramientas se utilizaran para el desarrollo de la aplicación. Finalmente se ha llegado a la conclusión de que el sistema se desarrollará sobre la plataforma **.NET**, usando como lenguaje de programación **C#**, y como gestor de bases de datos, se usará el **SQL Server 2000**.

## **2 Modelo del negocio**

### **2.1 Introducción**

En el presente capítulo se aborda lo referente al flujo de trabajo Modelación del Negocio, según la metodología RUP; desarrollando el Modelo de Casos de Uso de Negocio y del Modelo de Objetos del Negocio el diagrama de objetos del negocio. Se describen detalladamente los procesos y las reglas del negocio; se utiliza la notación UML. Las distintas vistas están compuestas por el diagrama de casos de uso del negocio y los diagramas de actividad correspondientes a cada caso de uso. También se describen los casos de uso del negocio en formato expandido para una mayor comprensión de los procesos del negocio.

### **2.2 Modelo del negocio actual**

Actualmente en la UCI existen una serie de áreas de servicios destinadas a la recepción y solución de las solicitudes de los clientes, estas solicitudes se pueden realizar o por vía telefónica o presentándose personalmente al área. Una vez realizada la solicitud el usuario espera a que le den solución a su problema, mientras que el área se encarga de generar una orden de trabajo y una certificación de trabajo para dar solución a la solicitud del cliente y mantener el control del gasto de materiales. Además existen los reportes acerca del estado de los servicios comunitarios, los cuales se realizan de forma manual.

### **2.3 Reglas del negocio a considerar**

- Para realizar una solicitud debe registrarse la fecha en que se realiza, los datos del solicitante, se debe especificar la descripción del problema, el local donde radica y los posibles medios asociados.
- Para dar solución a una solicitud se genera al menos una orden de trabajo la cual puede tener ninguna, una o varias certificaciones de trabajo en dependencia del estado en que se encuentre dicha solicitud.

### 2.3.1 Conceptos asociados al modelo de negocios.

**Área de servicios:** En la UCI para la prestación de los servicios comunitarios existen áreas. Cada área brinda una serie de servicios, además tiene uno o varios jefes del área, supervisores y técnicos.

**Medio:** Cualquier objeto que exista en la UCI y que pueda presentar problemas.

**Problema:** Posible situación que presente uno o varios medios.

**Local:** Cualquier lugar dentro de la estructura de la UCI.

**Supervisor:** Persona que da respuesta a las solicitudes hechas por los usuarios, y además genera órdenes de trabajos.

**Técnico:** Persona que da solución a una orden de trabajo.

**Jefe de área:** Persona que dirige un área de servicios.

**Solicitud:** Petición por parte de un usuario de un servicio a un área de servicios por presentar problema.

**Estado de solicitud:** Estado que indica en un momento dado la situación en que se encuentra una solicitud.

**Orden de trabajo:** Orden definida para dar solución a un problema específico dentro de un rango de fecha.

**Certificación de trabajo:** Acta de constancia de los materiales utilizados para dar solución a una solicitud.

**Reclamación:** Queja por problemas en la solución de una solicitud.

**Usuario:** Toda persona que trabaje o estudie en la UCI.

## 2.4 Actores del negocio

Tabla 1. Descripción de los actores del negocio

Nombre del actor	Descripción
Solicitante	Persona que realiza la solicitud, interesada además en conocer el estado en que se encuentra la misma.

## 2.5 Diagrama de casos de uso del negocio

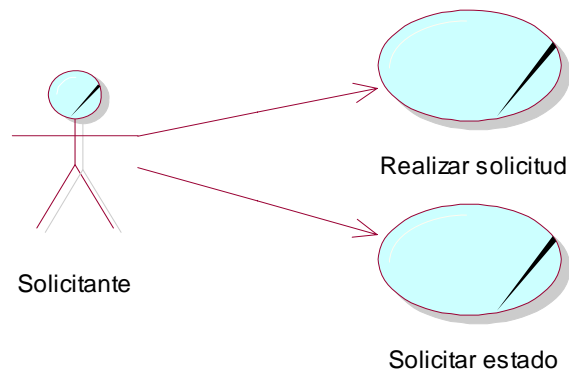


Figura 1. Diagrama de casos de uso del negocio

## 2.6 Trabajadores del negocio

Tabla 2. Descripción de los trabajadores del negocio

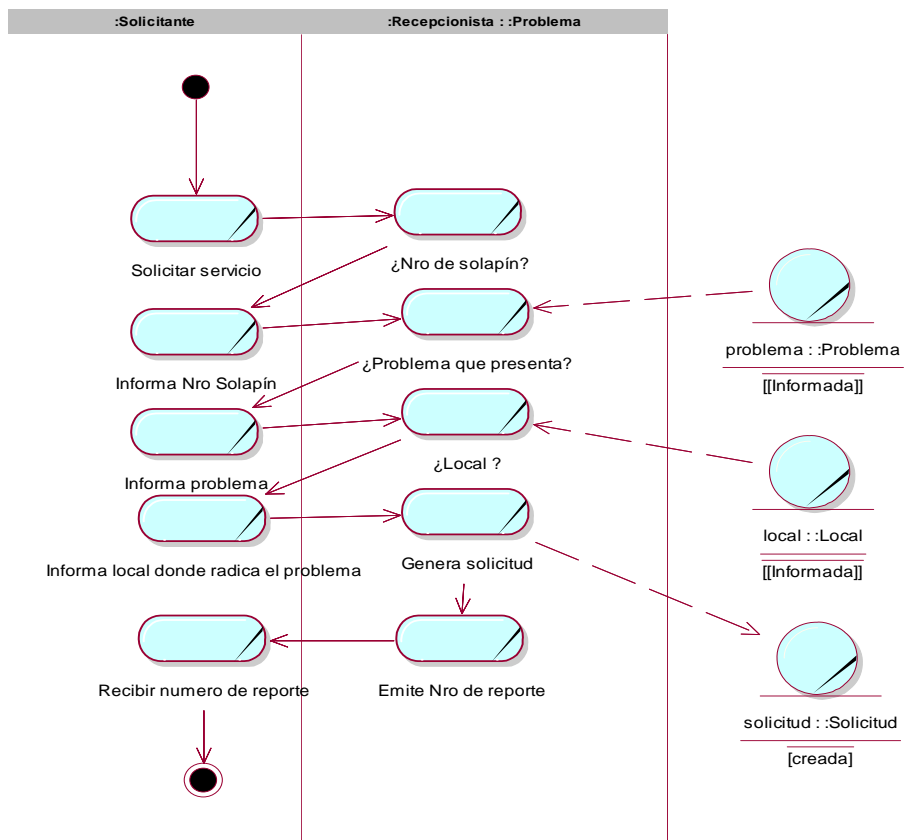
Nombre del trabajador	Descripción
Recepcionista	Personal de las áreas de servicios encargados de registrar las solicitudes de los usuarios.
Jefe de brigada	Personal encargado de obtener las solicitudes de su área y de asignar técnicos que den solución a las mismas.
Técnico	Personal encargado de dar solución a la solicitud del usuario.

## 2.7 Casos de uso del negocio

Tabla 3. Caso de uso: Realizar Solicitud

Nombre del caso de uso del negocio:	CU1 – Realizar solicitud.
-------------------------------------	---------------------------

<b>Actores del negocio:</b>	Usuario.
<b>Propósito:</b>	Solicitar un servicio comunitario.
<b>Resumen:</b>	
El caso de uso inicia cuando un Usuario se pone en contacto con el Recepcionista de un área de servicios y este registra su solicitud devolviendo un identificador de su solicitud (número de reporte).	
<b>Casos de uso asociados:</b>	
<b>Flujo de trabajo</b>	
<b>Acción del actor</b>	<b>Respuesta del negocio</b>
1 El Usuario le informa al Recepcionista los datos del problema que presenta.  3 El Usuario recibe identificador de la solicitud.	2 El Recepcionista registra los datos de la solicitud.
<b>Prioridad:</b>	Alta(De que se realice una solicitud, depende todo lo demás en el sistema)
<b>Mejoras:</b>	La automatización de este proceso de evaluación reducirá el tiempo de respuesta y permitirá a los Usuarios mejorar su gestión. El Usuario no tendrá que interactuar con el Recepcionista, pues este último desaparecerá al automatizarse este proceso.
<b>Cursos alternos:</b>	



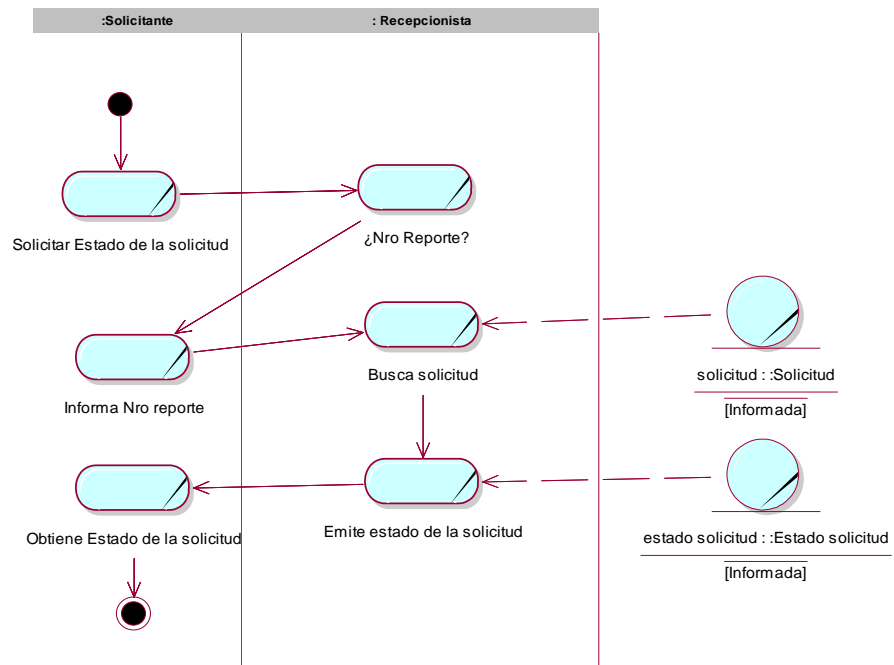
**Figura 2. Diagrama de actividad del caso de uso: Realizar Solicitud**

**Tabla 4. Caso de uso: Solicitar Estado**

<b>Nombre del caso de uso del negocio:</b>	CU2 – Solicitar estado.
<b>Actores del negocio:</b>	Usuario.
<b>Propósito:</b>	Solicitar el estado en que se encuentra una solicitud hecha anteriormente.
<b>Resumen:</b>	



<p>El caso de uso inicia cuando un Usuario se pone en contacto con el Recepcionista de un área de servicios para obtener el estado de una solicitud hecha previamente, el Usuario le proporciona el identificador de su solicitud y el Recepcionista le devuelve el estado en que se encuentra su solicitud.</p>	
<b>Casos de uso asociados:</b>	
<b>Flujo de trabajo</b>	
<b>Acción del actor</b>	<b>Respuesta del negocio</b>
<p>1 El Usuario le informa al Recepcionista el identificador de la solicitud de la cual desea conocer el estado en que se encuentra.</p> <p>3 El Usuario recibe el estado en que se encuentra la solicitud.</p>	<p>2 El Recepcionista con el identificador de solicitud suministrado por el Usuario, busca su estado y se lo devuelve.</p>
<b>Prioridad:</b>	Alta (El conocimiento por parte del Usuario del estado de su solicitud es fundamental)
<b>Mejoras:</b>	La automatización de este proceso de evaluación reducirá el tiempo de respuesta y permitirá a los Usuarios obtener el estado de su solicitud en todo momento.
<b>Cursos alternos:</b>	



**Figura 3. Diagrama de actividad del caso de uso: Solicitar Estado**

## 2.8 Modelo de objetos

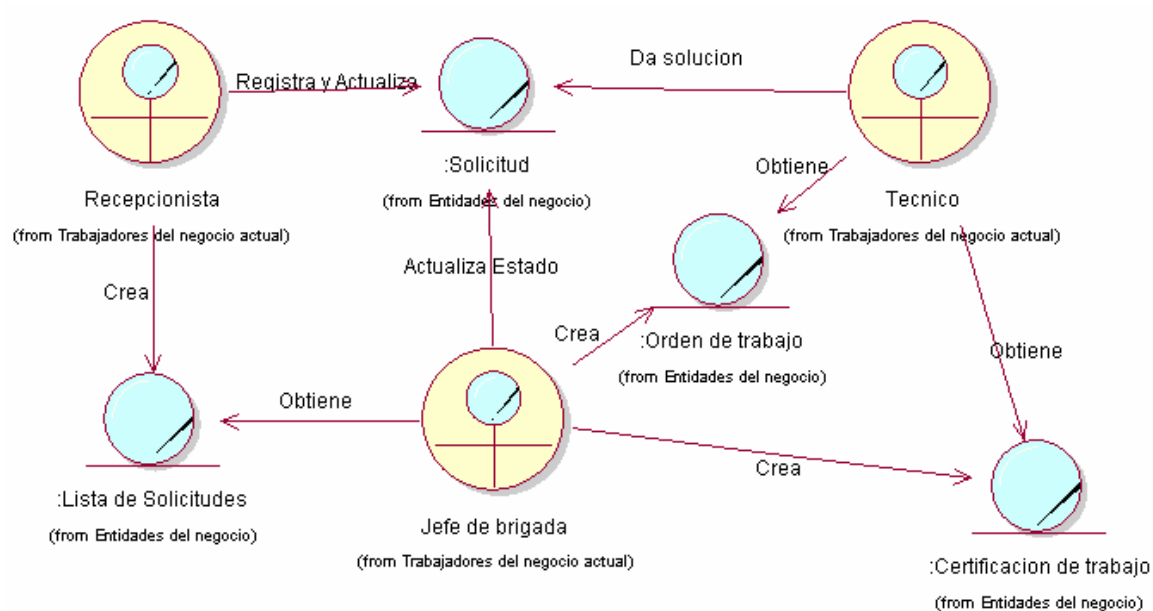


Figura 4. Modelo de objeto

## 2.9 Conclusiones

En el presente capítulo analizamos el modelo de negocio a partir de un estudio detallado del funcionamiento de los servicios comunitarios en las distintas áreas de la UCI, se detalló el proceso de negocio actual haciendo un análisis crítico del flujo actual de los procesos, de esta forma pudimos detectar las ineficiencias que se llevan a cabo en las distintas áreas y las posibles vías de solución, sentando las bases para la realización de la automatización de dichos procesos, que comenzaremos a detallar en el próximo capítulo.

## **3 Requisitos**

### **3.1 Introducción**

En este capítulo se aborda sobre los requisitos funcionales y no funcionales del sistema que se propone, lo que permite tener una concepción general del sistema y representar a través del diagrama de caso de uso, las relaciones de los actores del sistema, y las secuencias de acciones con las que interactúan.

### **3.2 Definición de los requisitos funcionales.**

1. Registrar nueva solicitud de servicio.
  - 1.1. Entrar datos de la solicitud.
    - 1.1.1. Obtener datos generales de la persona que solicita.
    - 1.1.2. Seleccionar local donde se encuentra la situación a reportar.
    - 1.1.3. Seleccionar problema
      - 1.1.3.1. Seleccionar medios asociados al problema
    - 1.1.4. Describir solicitud.
  - 1.2. Almacenar detalles de la solicitud: ip de la máquina de donde se realizó la solicitud, fecha y hora en que se realizó la solicitud.
2. Realizar seguimiento de una solicitud.
  - 2.1. Mostrar el estado en que se encuentra la solicitud.
  - 2.2. Realizar reclamación: descripción.
  - 2.3. Mostrar reclamaciones hechas a una solicitud.
  - 2.4. Mostrar el historial de las solicitudes realizadas por un usuario, filtradas por fecha y estados.
  - 2.5. Mostrar vista previa de la solicitud.
3. Atender solicitudes realizadas a un área.
  - 3.1. Listar solicitudes hechas a un área, filtradas por fecha y estado.

- 3.2. Buscar solicitud dado número de reporte.
- 3.3. Emitir orden de trabajo.
  - 3.3.1. Almacenar solicitud que la genera, fecha de asignación, fecha de inicio y fecha de fin.
  - 3.3.2. Asignar grupo de técnicos.
- 3.4. Actualizar estado de la solicitud.
4. Generar certificación de trabajo
  - 4.1. Almacenar orden de trabajo que la genera, fecha y descripción.
  - 4.2. Almacenar consumo de materiales.
5. Administrar material.
  - 5.1. Adicionar material: nombre y descripción.
  - 5.2. Modificar datos del material.
  - 5.3. Eliminar material.
6. Administrar unidad medida
  - 6.1. Adicionar unidad medida: nombre.
  - 6.2. Modificar datos de la unidad medida.
  - 6.3. Eliminar unidad medida.
7. Administrar área de servicios.
  - 7.1. Añadir nueva área de servicios: nombre del área y descripción.
  - 7.2. Modificar datos del área de servicios.
  - 7.3. Eliminar área de servicios.
8. Administrar jefe de área de servicios.
  - 8.1. Adicionar nuevo jefe de área.
  - 8.2. Eliminar jefe de área.
9. Administrar supervisor de un área de servicios.
  - 9.1. Adicionar nuevo supervisor.
  - 9.2. Eliminar supervisor.

10. Administrar técnico de un área de servicios.

10.1. Adicionar nuevo técnico.

10.2. Eliminar técnico.

11. Administrar estado de la solicitud.

11.1. Añadir nuevo estado de solicitud: nombre y descripción.

11.2. Modificar los datos del estado de solicitud.

11.3. Eliminar estado de solicitud.

12. Administrar servicio de un área de servicios.

12.1. Adicionar nuevo servicio: nombre, identificador del área de servicios y descripción.

12.2. Modificar servicio.

12.3. Eliminar servicio.

13. Administrar medios de un problema.

13.1. Adicionar nuevo medio, identificador (los demás datos se obtienen a través de un Web Services).

13.2. Eliminar medio.

14. Administrar local.

14.1. Adicionar local, nombre, identificador del local padre.

14.2. Modificar local.

14.3. Eliminar local.

15. Administrar problema.

15.1. Adicionar problema, nombre, identificador del servicio y descripción.

15.2. Modificar problema.

15.3. Eliminar problema.

16. Listar reportes destinados a directivos.

16.1. Mostrar solicitudes atendidas por área de servicios, filtradas según fecha y estados.

16.2. Listar consumo de materiales por área, filtrados por fecha.

16.2. Obtener promedio de reclamaciones por áreas filtradas por fecha.

### **3.3 Definición de los requisitos no funcionales**

#### **Interfaz de usuario:**

Se aplicará el estándar del proyecto UCI Ciudad Digital para el diseño de interfaz de usuario. Esta será legible, simple de usar, e interactiva.

#### **Usabilidad:**

Facilidad de uso para todo tipo de clientes, incluyendo personas con pocos conocimientos en el uso de las PCs.

Manual de usuario. Material de entrenamiento. Ayuda en Línea, soportada por páginas Web, que estará disponible al usuario en todo momento.

#### **Soporte:**

El sistema debe ser de fácil instalación, adaptable a numerosas plataformas y de fácil mantenimiento.

#### **Portabilidad:**

Facilidad para adaptarlos a diferentes ambientes sin necesidad de usar otros medios que los previstos. Se requiere de Windows como plataforma.

#### **Seguridad:**

Se deben implementar varios niveles de usuarios con permisos que correspondan con el rol que desempeñan en la aplicación.

Evaluar mecanismos de tolerancias a faltas. Predicción de fallos.

Protección contra los fallos. El sistema debe ser capaz de en pocos segundos recuperarse de un fallo de una operación.

El sistema permite que los usuarios pocos familiarizados con el sistema perciban sin problemas las salidas del mismo. Las salidas del sistema tienen que tener un 100 % de veracidad y precisión.

Podrá ser usado las 24 horas del día.

**Legales:**

Debe cumplir con lo establecido en las leyes del sistema de seguridad y protección de nuestro país. El sistema de soporte comunitario y la documentación del mismo pertenecen al proyecto UCI-Ciudad Digital.

**Software:**

Sistemas Operativos de la familia Windows (Windows NT 4.0, Windows 9.x, Windows 2000). La aplicación se realizará en un ambiente Web, la Base de Datos es independiente de la aplicación.

**Hardware:**

Compatible con procesadores x486 o superior, con 64 MB de memoria RAM. Un mínimo de 200 MB de espacio disponible en disco.

Soportado por una red hasta 100 Mbps de velocidad.

**Restricciones:**

Para la documentación del sistema se utilizó para realizar el análisis y el diseño del sistema la metodología RUP, y como herramienta de modelación UML (Unified Modeling Language), como herramienta de apoyo a este se utilizó el Rational Rose. Su desarrollo se llevará a cabo con .Net, y como gestor de base de datos se utilizará el SQL Server.

**3.4 Actores del sistema a automatizar****Tabla 5. Definición de actores del sistema a automatizar**

Nombre del actor	Descripción
Usuario	Cualquier persona, estudiante o trabajador de la UCI que solicite un servicio.
Supervisor	Persona que pertenece a un área de servicios, la cual se encarga de atender las solicitudes que lleguen a su área.
Jefe de área	Persona al mando de un área de servicios de la UCI.



Directivo	Persona con cargo directivo dentro de la UCI, que necesita reportes acerca de los servicios.
Administrador	Persona designada para darle mantenimiento al sistema.

### 3.5 Diagrama de casos de uso del sistema a automatizar

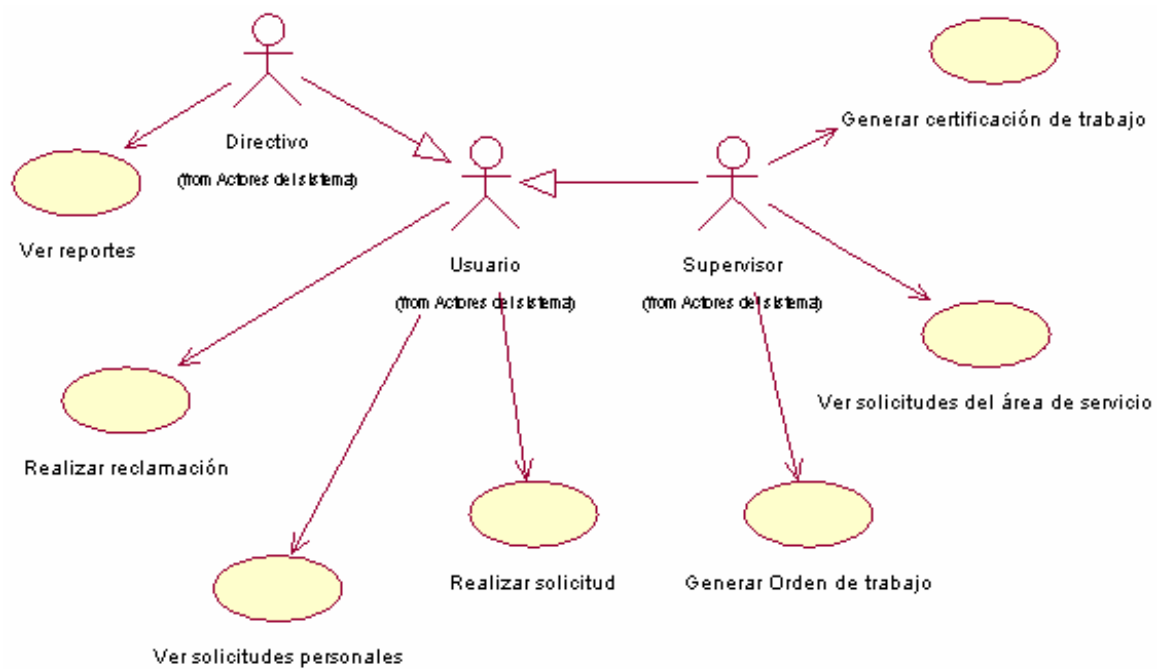


Figura 5. Diagrama de los casos de uso de servicio.

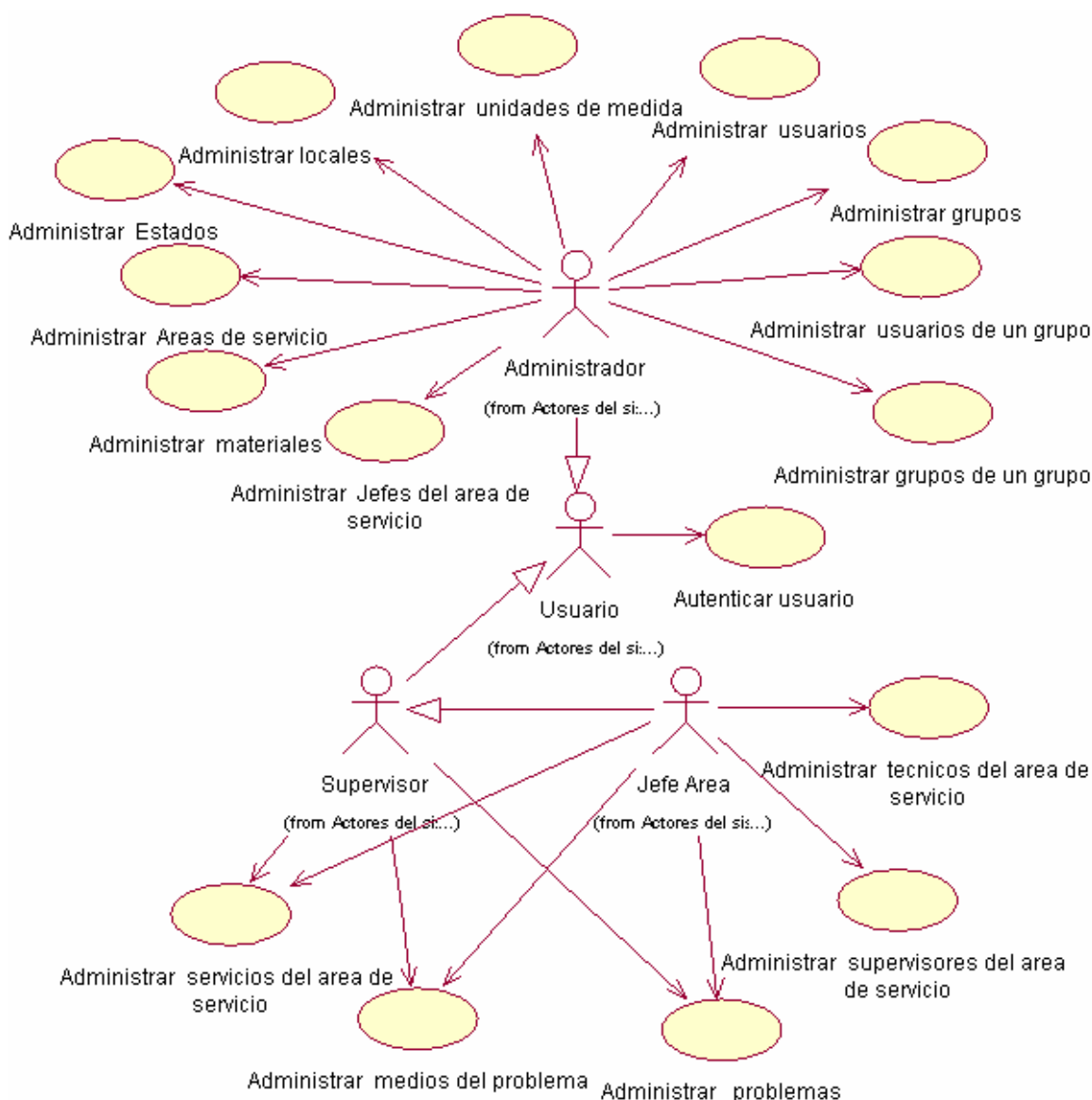


Figura 6. Diagrama de los casos de uso de administración.

### 3.6 Descripción de los casos de uso

Tabla 6. Descripción del caso de uso: Administrar áreas de servicio

<b>Nombre del caso de uso</b>	Administrar áreas de servicio
<b>Actores</b>	Administrador
<b>Resumen</b>	El caso de uso se inicia cuando el administrador muestra la lista de las áreas de servicios definidas en el sistema, puede

	agregar, eliminar y modificar las áreas. Permite a los administrativos actualizar las áreas de servicios.
<b>Precondiciones</b>	El usuario que este autenticado en el sistema debe estar dentro del grupo de administración.
<b>Poscondiciones</b>	
<b>Requisitos especiales</b>	

**Tabla 7. Descripción del caso de uso: Administrar los jefes del área de servicio**

<b>Nombre del caso de uso</b>	Administrar jefes del área de servicio
<b>Actores</b>	Administrador
<b>Resumen</b>	El caso de uso se inicia cuando el administrador escoge el área de servicios para administrar sus jefes. El administrador puede agregar o eliminar jefes. Permite que los administradores puedan actualizar los jefes de las áreas de servicios
<b>Precondiciones</b>	El usuario que este autenticado en el sistema debe estar dentro del grupo de administración. Debe existir al menos un área de servicios para adicionarle jefes (Ver caso de uso Actualizar áreas de servicios).
<b>Poscondiciones</b>	
<b>Requisitos especiales</b>	

**Tabla 8. Descripción del caso de uso: Administrar estados**

<b>Nombre del caso de uso</b>	Administrar estados
<b>Actores</b>	Administrador
<b>Resumen</b>	El caso de uso se inicia cuando el administrador revisa los estados, puede agregar, eliminar y modificar los estados. Permite que los administradores actualicen los posibles estados por los que puede pasar una solicitud.

<b>Precondiciones</b>	El usuario que este autenticado en el sistema debe estar dentro del grupo de administración.
<b>Poscondiciones</b>	
<b>Requisitos especiales</b>	

**Tabla 9. Descripción del caso de uso: Administrar técnicos del área de servicios.**

<b>Nombre del caso de uso</b>	Administrar técnicos del área de servicios
<b>Actores</b>	Jefe de área
<b>Resumen</b>	El caso de uso se inicia cuando el Jefe de área revisa los técnicos del área, donde puede agregar o eliminar técnicos. Permite que los jefes de área puedan actualizar los técnicos de su área de servicios
<b>Precondiciones</b>	El usuario que este autenticado en el sistema debe estar dentro del grupo de jefes de área. Debe existir al menos un área de servicios para adicionarle técnicos (Ver caso de uso Actualizar áreas de servicios).
<b>Poscondiciones</b>	
<b>Requisitos especiales</b>	

**Tabla 10. Descripción del caso de uso: Administrar supervisores del área de servicios.**

<b>Nombre del caso de uso</b>	Administrar supervisores del área de servicios
<b>Actores</b>	Jefe de área
<b>Resumen</b>	El caso de uso se inicia cuando el Jefe de área lista los supervisores del área de servicios. Aquí se puede agregar y eliminar supervisores. Permite a los jefes de área actualizar los supervisores del área de servicios.
<b>Precondiciones</b>	El usuario que este autenticado en el sistema debe estar dentro del grupo de jefes de área.

	Debe existir al menos un área de servicios para adicionarle supervisores (Ver caso de uso Actualizar áreas de servicios).
<b>Poscondiciones</b>	
<b>Requisitos especiales</b>	

**Tabla 11. Descripción del caso de uso: Administrar servicios del área de servicios.**

<b>Nombre del caso de uso</b>	Administrar servicios del área de servicios
<b>Actores</b>	Supervisor, Jefe de área
<b>Resumen</b>	El caso de uso se inicia cuando el supervisor o el Jefe de área revisen los servicios del área, puede agregar, eliminar y modificar los servicios. Permite a los supervisores y jefes de área de un área de servicios actualizar los servicios del área.
<b>Precondiciones</b>	El usuario que este autenticado en el sistema debe estar dentro del grupo de supervisores o jefes de área. Debe existir al menos un área de servicios para adicionarle servicios (Ver caso de uso Actualizar áreas de servicios).
<b>Poscondiciones</b>	
<b>Requisitos especiales</b>	

**Tabla 12. Descripción del caso de uso: Administrar medios del problema**

<b>Nombre del caso de uso</b>	Administrar medios del problema
<b>Actores</b>	Supervisor, Jefe de área
<b>Resumen</b>	El caso de uso se inicia cuando el supervisor o el Jefe de área escogen un problema para actualizar sus medios. Se puede agregar y eliminar medios relacionados a un problema. Permite a los supervisores o jefes de área actualizar los medios asociados a los problemas del área.
<b>Precondiciones</b>	El usuario que este autenticado en el sistema debe estar dentro del grupo de supervisores o jefes de área.

	Debe existir al menos un problema para poder adicionarle un medio (ver caso de uso Actualizar problemas del área de servicios).
<b>Poscondiciones</b>	
<b>Requisitos especiales</b>	

**Tabla 13. Descripción del caso de uso: Administrar problemas.**

<b>Nombre del caso de uso</b>	Administrar problema
<b>Actores</b>	Supervisor, Jefe de área
<b>Resumen</b>	El caso de uso se inicia cuando el supervisor o el Jefe de área escogen un servicio y ve la lista de problemas asociados al mismo. Se puede agregar, eliminar y modificar el problema. Permite a los supervisores y a los jefes de área actualizar los problemas asociados a los servicios.
<b>Precondiciones</b>	El usuario que este autenticado en el sistema debe estar dentro del grupo de supervisores o jefes de área. Debe existir al menos un servicio para asignarle sus posibles problemas (ver caso de uso Actualizar servicio).
<b>Poscondiciones</b>	
<b>Requisitos especiales</b>	

**Tabla 14. Descripción del caso de uso: Ver solicitudes personales.**

<b>Nombre del caso de uso</b>	Ver solicitudes personales
<b>Actores</b>	Usuario
<b>Resumen</b>	El caso de uso se inicia cuando el usuario selecciona la opción de historial de solicitudes, se muestra un listado de las solicitudes realizadas por el usuario y el estado en que se encuentran en un rango de fecha determinado.  En caso de que el usuario quiera ver los detalles de la solicitud, selecciona descripción y el sistema le muestra los

	<p>detalles de la solicitud realizada</p> <p>Permite que los usuarios del sistema puedan ver el estado en que se encuentran las solicitudes realizadas por ellos.</p>
<b>Precondiciones</b>	<p>El usuario que este autenticado en el sistema debe estar al menos dentro del grupo de usuarios.</p> <p>Debe existir al menos una solicitud(Ver caso de uso Realizar solicitud)</p>
<b>Poscondiciones</b>	
<b>Requisitos especiales</b>	

**Tabla 15. Descripción del caso de uso: Realizar reclamación.**

<b>Nombre del caso de uso</b>	Realizar reclamación
<b>Actores</b>	Usuario
<b>Resumen</b>	El caso de uso se inicia cuando el usuario revisa los detalles de su solicitud, le puede agregar varias reclamaciones a la solicitud. Permite al usuario realizar una reclamación en caso de que no este satisfecho.
<b>Precondiciones</b>	<p>El usuario que este autenticado en el sistema debe estar al menos dentro del grupo de usuario.</p> <p>Debe existir al menos una solicitud realizada por el usuario para poder realizar una reclamación (Ver caso de uso Realizar solicitud).</p>
<b>Poscondiciones</b>	
<b>Requisitos especiales</b>	

**Tabla 16. Descripción del caso de uso: Ver solicitudes del área de servicio.**

<b>Nombre del caso de uso</b>	Ver solicitudes del área de servicio
<b>Actores</b>	Supervisor, Jefe área
<b>Resumen</b>	El caso de uso se inicia cuando el supervisor selecciona la opción de ver las solicitudes hechas al área en un rango de

	fechas y/o según el estado de la misma. Permite a los supervisores ver el listado de las solicitudes que han sido hechas a su área de servicios.
<b>Precondiciones</b>	El usuario que este autenticado en el sistema debe estar dentro del grupo de supervisores. Debe existir al menos una solicitud hecha al área a la que pertenece el supervisor (Ver caso de uso Realizar solicitud).
<b>Poscondiciones</b>	
<b>Requisitos especiales</b>	

**Tabla 17. Descripción del caso de uso: Generar orden de trabajo.**

<b>Nombre del caso de uso</b>	Generar orden de trabajo
<b>Actores</b>	supervisor
<b>Resumen</b>	El caso de uso se inicia cuando un supervisor ve la lista de solicitudes hechas al área de servicios, aquí para responder una solicitud, debe generar una orden de trabajo). Luego el supervisor puede ver la lista de órdenes de trabajo generadas, de aquí puede imprimirlas o editarlas. Permite a los supervisores generar órdenes de trabajo y actualizarlas además.
<b>Precondiciones</b>	El usuario que este autenticado en el sistema debe estar dentro del grupo de supervisores. Debe existir al menos una solicitud hecha al área a la que pertenece el supervisor (Ver caso de uso Realizar solicitud).
<b>Poscondiciones</b>	
<b>Requisitos especiales</b>	

**Tabla 18. Descripción del caso de uso: Ver reportes**

<b>Nombre del caso de uso</b>	Ver reportes
-------------------------------	--------------



<b>Actores</b>	Directivo, Jefe de área, Supervisor
<b>Resumen</b>	El caso de uso se inicia cuando un directivo, Jefe de área o Supervisor escoge un reporte para verlo. Permite a los directivos, jefes de área y supervisores ver y recibir en todo momento reportes que informan el estado general del sistema y de los servicios comunitarios, de acuerdo a sus intereses.
<b>Precondiciones</b>	El usuario que este autenticado en el sistema debe estar dentro del grupo de directivos, jefes de área o supervisores. Para poder ver los reportes, deben existir solicitudes (Ver caso de uso Realizar solicitud).
<b>Poscondiciones</b>	
<b>Requisitos especiales</b>	

**Tabla 19. Descripción del caso de uso: Autenticar usuario.**

<b>Nombre del caso de uso</b>	Autenticar usuario
<b>Actores</b>	Usuario
<b>Resumen</b>	El Caso de Uso se inicia cuando el usuario introduce los datos que se le piden para acceder a la aplicación, estos se verifican y finaliza dándole los permisos de acuerdo a su rol y habilitándole la entrada.
<b>Precondiciones</b>	
<b>Poscondiciones</b>	Se habilitan las funcionalidades según los privilegios.
<b>Requisitos especiales</b>	

**Tabla 20. Descripción del caso de uso: Generar certificación de trabajo.**

<b>Nombre del caso de uso</b>	Generar certificación de trabajo.
<b>Actores</b>	Supervisor
<b>Resumen</b>	El caso de uso se inicia cuando un supervisor ve la lista de las órdenes de trabajo generadas por el área de servicios, y en caso de que sea necesario generar una certificación de

	trabajo que respalde y justifique el consumo de materiales para dar solución a un problema. Luego el supervisor puede ver la lista de certificaciones de trabajo generadas, de aquí puede imprimirlas o editarlas. Permite a los supervisores generar certificaciones de trabajo y actualizarlas además.
<b>Precondiciones</b>	El usuario que este autenticado en el sistema debe estar dentro del grupo de supervisores. Debe existir al menos una orden de trabajo hecha al área a la que pertenece el supervisor (Ver caso de uso Realizar solicitud).
<b>Poscondiciones</b>	
<b>Requisitos especiales</b>	

**Tabla 21. Descripción del caso de uso: Administrar materiales**

<b>Nombre del caso de uso</b>	Administrar Materiales
<b>Actores</b>	Administrador
<b>Resumen</b>	El caso de uso se inicia cuando el administrador del sistema necesite adicionar, eliminar o modificar materiales. Permite a los administradores actualizar materiales.
<b>Precondiciones</b>	El usuario que este autenticado en el sistema debe estar dentro del grupo de administradores.
<b>Poscondiciones</b>	
<b>Requisitos especiales</b>	

**Tabla 22. Descripción del caso de uso: Administrar Unidades de Medida**

<b>Nombre del caso de uso</b>	Administrar unidades de medida
<b>Actores</b>	Administrador
<b>Resumen</b>	El caso de uso se inicia cuando el administrador del sistema necesite adicionar, eliminar o modificar Unidad de Medida.

	Permite a los administradores actualizar Unida de Medida.
<b>Precondiciones</b>	El usuario que este autenticado en el sistema debe estar dentro del grupo de administradores.
<b>Poscondiciones</b>	
<b>Requisitos especiales</b>	

**Tabla 23. Descripción del caso de uso: Administrar locales.**

<b>Nombre del caso de uso</b>	Administrar locales
<b>Actores</b>	Administrador
<b>Resumen</b>	El caso de uso se inicia cuando el administrador del sistema necesite adicionar, eliminar o modificar locales. Permite a los administradores actualizar locales.
<b>Precondiciones</b>	El usuario que este autenticado en el sistema debe estar dentro del grupo de administradores.
<b>Poscondiciones</b>	
<b>Requisitos especiales</b>	

**Tabla 24. Descripción del caso de uso: Administrar usuarios.**

<b>Nombre del caso de uso</b>	Administrar usuarios
<b>Actores</b>	Administrador
<b>Resumen</b>	El caso de uso se inicia cuando el administrador del sistema necesite adicionar, eliminar o modificar usuario. Permite a los administradores actualizar usuarios.
<b>Precondiciones</b>	El usuario que este autenticado en el sistema debe estar dentro del grupo de administradores.
<b>Poscondiciones</b>	

<b>Requisitos especiales</b>	
------------------------------	--

**Tabla 25. Descripción del caso de uso: Administrar grupos.**

<b>Nombre del caso de uso</b>	Administrar Grupos
<b>Actores</b>	Administrador
<b>Resumen</b>	El caso de uso se inicia cuando el administrador del sistema necesite adicionar, eliminar o modificar grupos. Permite a los administradores actualizar grupos.
<b>Precondiciones</b>	El usuario que este autenticado en el sistema debe estar dentro del grupo de administradores.
<b>Poscondiciones</b>	
<b>Requisitos especiales</b>	

**Tabla 26. Descripción del caso de uso: Administrar usuarios de un grupo.**

<b>Nombre del caso de uso</b>	Administrar usuarios de un grupo
<b>Actores</b>	Administrador
<b>Resumen</b>	El caso de uso se inicia cuando el administrador del sistema seleccione un grupo, permitiendo adicionar o eliminar usuario. Permite a los administradores actualizar usuarios de un grupo.
<b>Precondiciones</b>	El usuario que este autenticado en el sistema debe estar dentro del grupo de administradores.
<b>Poscondiciones</b>	
<b>Requisitos especiales</b>	

**Tabla 27. Descripción del caso de uso: Administrar grupos de un grupo.**

<b>Nombre del caso de uso</b>	Administrar grupos de un grupo
<b>Actores</b>	Administrador
<b>Resumen</b>	El caso de uso se inicia cuando el administrador del sistema selecciona un grupo para adicionar o eliminar grupos que pertenezcan a él. Permite a los administradores actualizar grupos de un grupo.
<b>Precondiciones</b>	El usuario que este autenticado en el sistema debe estar dentro del grupo de administradores.
<b>Poscondiciones</b>	
<b>Requisitos especiales</b>	

**Tabla 28. Descripción del caso de uso: Realizar solicitud.**

<b>Nombre del caso de uso</b>	Realizar solicitud
<b>Actores</b>	Usuario
<b>Resumen</b>	El caso de uso se inicia cuando el usuario selecciona Nueva Solicitud e ingresa los datos necesarios para realizar dicha solicitud, para ello es necesario que escoja un problema y de este los posibles medios asociados, también debe seleccionar el local donde radica dicha situación problémica, pudiendo ver la vista previa de la misma si lo desea. Permite a los usuarios realizar solicitud.
<b>Precondiciones</b>	El usuario que este autenticado en el sistema debe estar al menos dentro del grupo de usuarios.
<b>Poscondiciones</b>	
<b>Requisitos especiales</b>	

### **3.7 Conclusiones**

En este capítulo se desarrolló 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 ahora se puede empezar a construir el sistema, tratando de que se cumplan todos los requerimientos y las funciones que han sido consideradas necesarias en este capítulo.

## **4 Descripción de la solución propuesta**

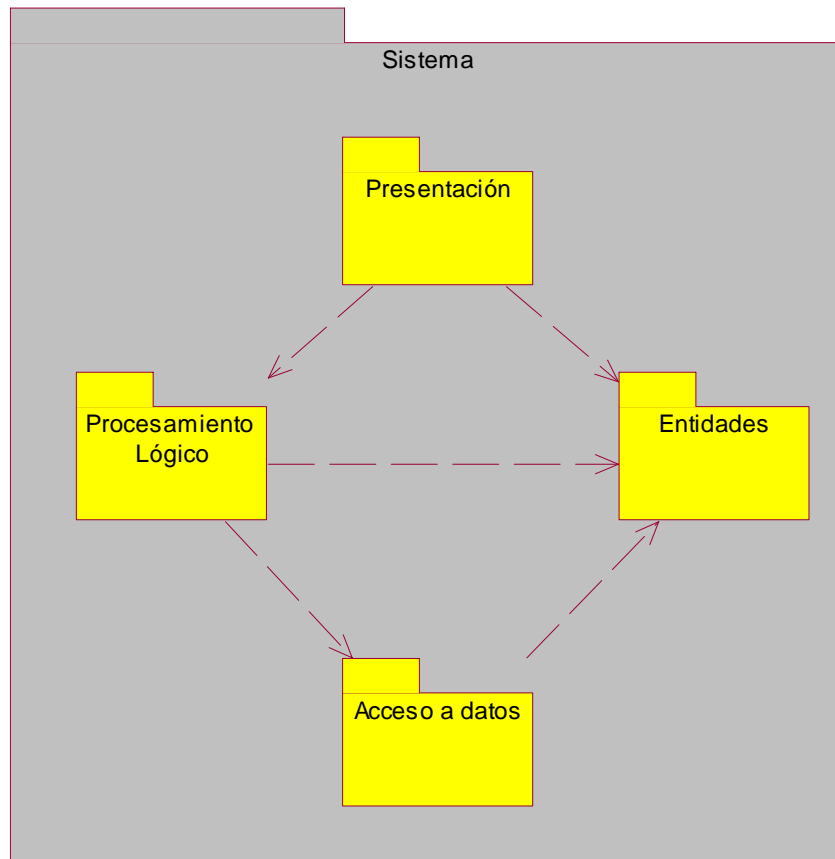
### ***4.1 Introducción***

En el presente capítulo se modelan los artefactos de diseño del sistema, utilizando para su modelado el Lenguaje Unificado de Modelado (UML). Se define el diagrama de clases, se tratan los principios del diseño del sistema, se explica el tratamiento de errores. En los últimos epígrafes, se diseña la Base de Datos y se realiza el diagrama de despliegue.

### ***4.2 Diagrama de clases del diseño***

Para un mejor entendimiento de los diagramas de clases, estos se han dividido en paquetes, los cuales también han sido divididos en sub.-paquetes.

#### 4.2.1 Paquete Gestión de Servicios Comunitarios



**Figura 7. Diagrama de clases del paquete GSC**

El paquete **Acceso a Datos** contiene las clases para hacer posible el acceso a todo tipo de datos manejados por la aplicación.

El paquete **Procesamiento Lógico** contiene las clases encargadas de manejar la lógica de negocio de la aplicación.

El paquete **Entidades** contiene las clases que representan entidades reales del dominio, no presentan comportamiento, solo propiedades, estas clases son usadas para pasar información entre las capas de la aplicación, por lo que es usada por todas.

El paquete **Presentación** contiene las clases controladoras y de presentación del sistema, en nuestro caso una interfaz Web.



## 4.2.2 Paquete Acceso a Datos.

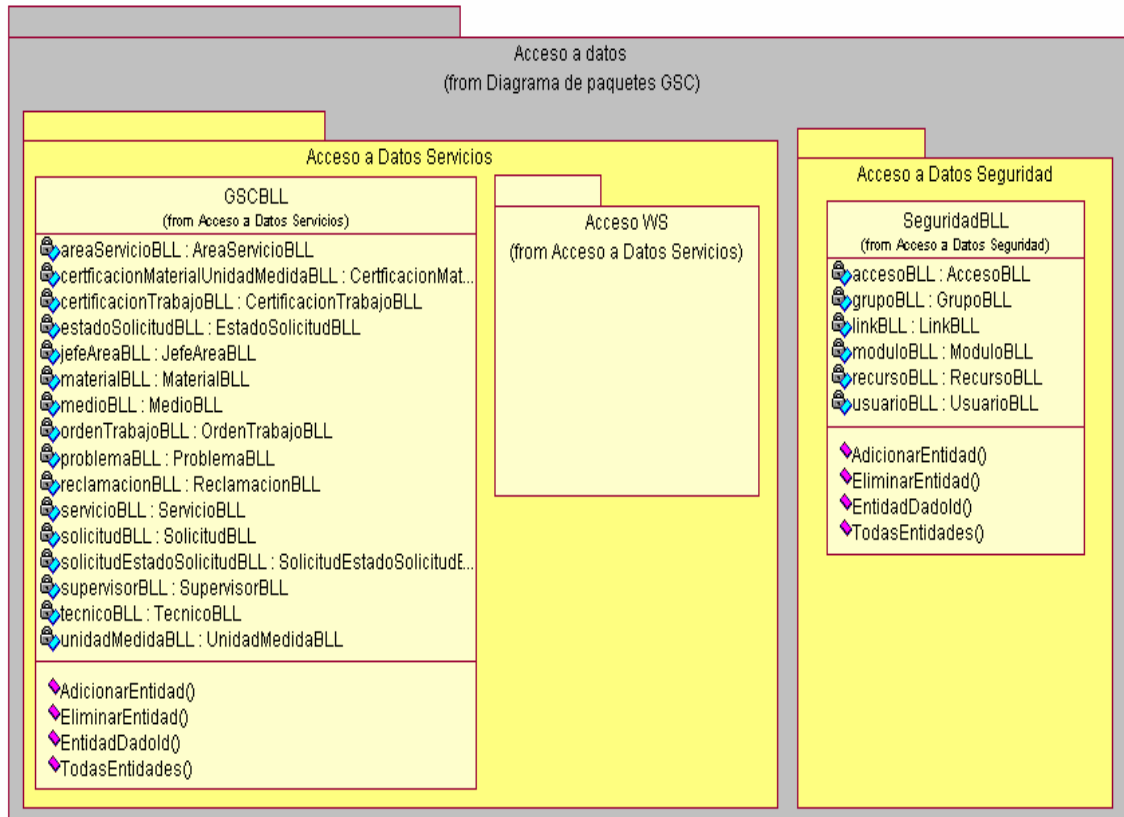
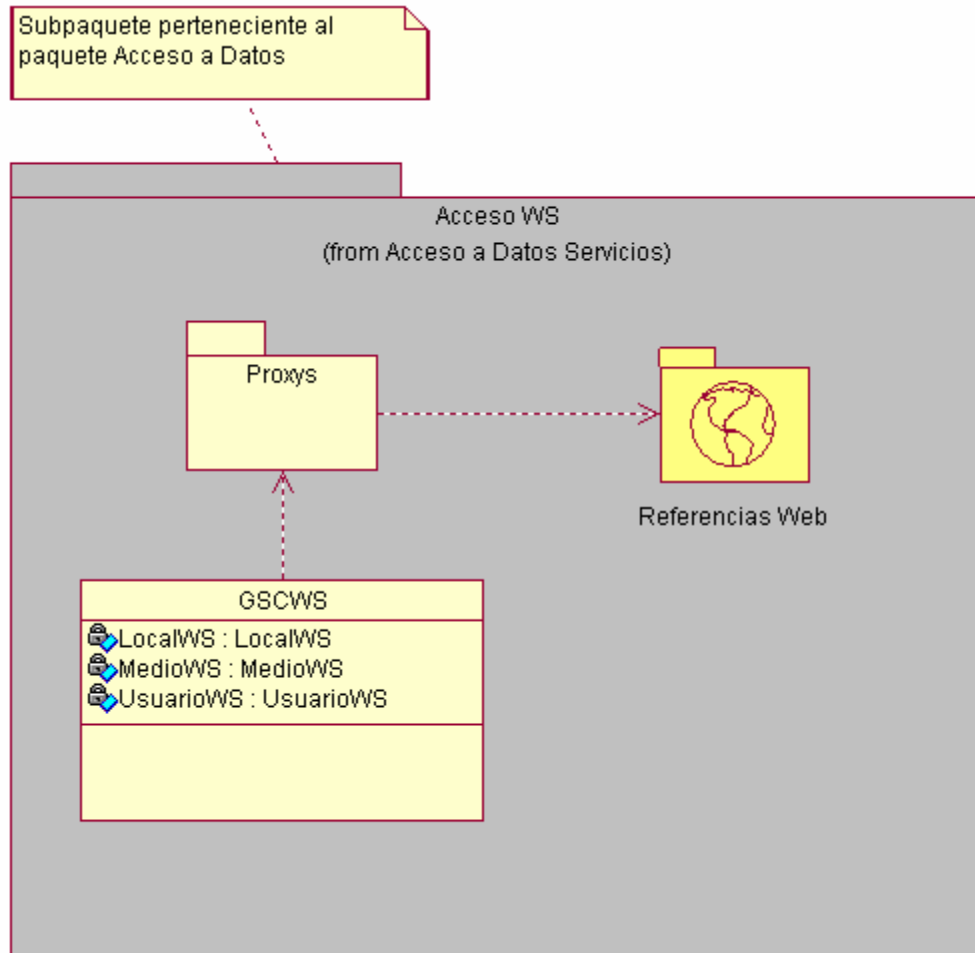


Figura 8. Diagrama de clases del paquete Acceso a Datos.

El paquete **Acceso a Datos** se dividió en dos subpaquetes: **Acceso a Datos Servicios** y **Acceso a Datos Seguridad**.

El paquete **Acceso a Datos Servicios** contiene la clase controladora **GSCBLL**, la cual posee una instancia de cada una de las clases que controlan la lógica del negocio del sistema, a través de las cuales podemos manejar los datos de todas las entidades pertenecientes a los servicios, en conjunto con el subpaquete Acceso a Web Services encargado de obtener los datos a través de los Servicios Web (Web Services) que brindan información de las Bases de Datos centrales de la Universidad. El paquete **Acceso a Datos Seguridad** contiene la clase controladora **SeguridadBLL**, la cual posee así mismo una instancia de cada una de las clases que manejan la lógica del negocio referente a la seguridad del sistema.

#### 4.2.2.1 Paquete Acceso a WS



**Figura 9. Diagrama de clases del paquete Acceso WS.**

EL paquete **Acceso WS** ha sido dividido en dos subpaquetes; el paquete **Proxys** y el paquete **Referencias Web**, además de contener la clase controladora **GSCWS**, la cual posee instancias de las clases que manejan la información de los Servicios Web.

El paquete **Proxys** contiene las clases puentes entre el sistema y las clases encargadas de acceder a los Servicios Web (Web Services).

El paquete **Referencias Web** contiene las clases encargadas de acceder a los Servicios Web (Web Services).

#### 4.2.2.1.1 Paquete Proxys (Perteneiente al paquete Acceso a WS)

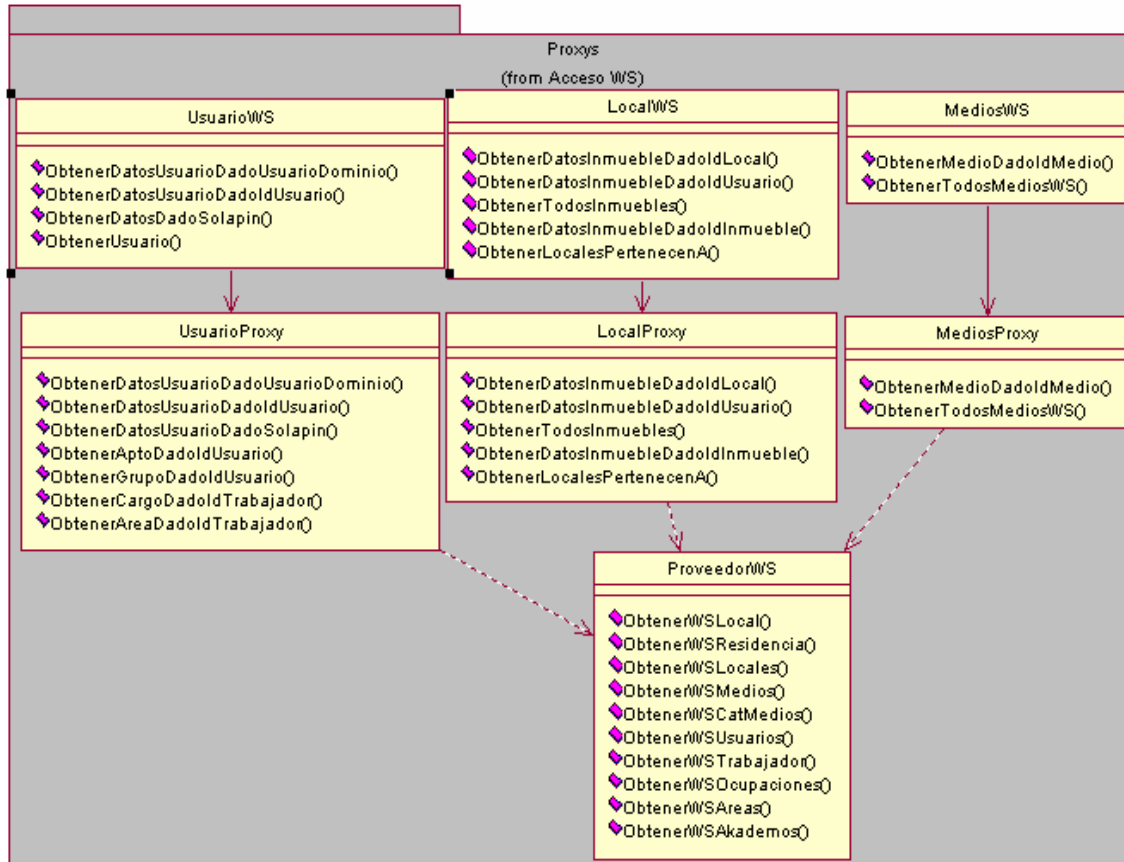


Figura 10. Diagrama de clases del paquete Proxys.

#### 4.2.2.1.2 Paquete Referencias Web

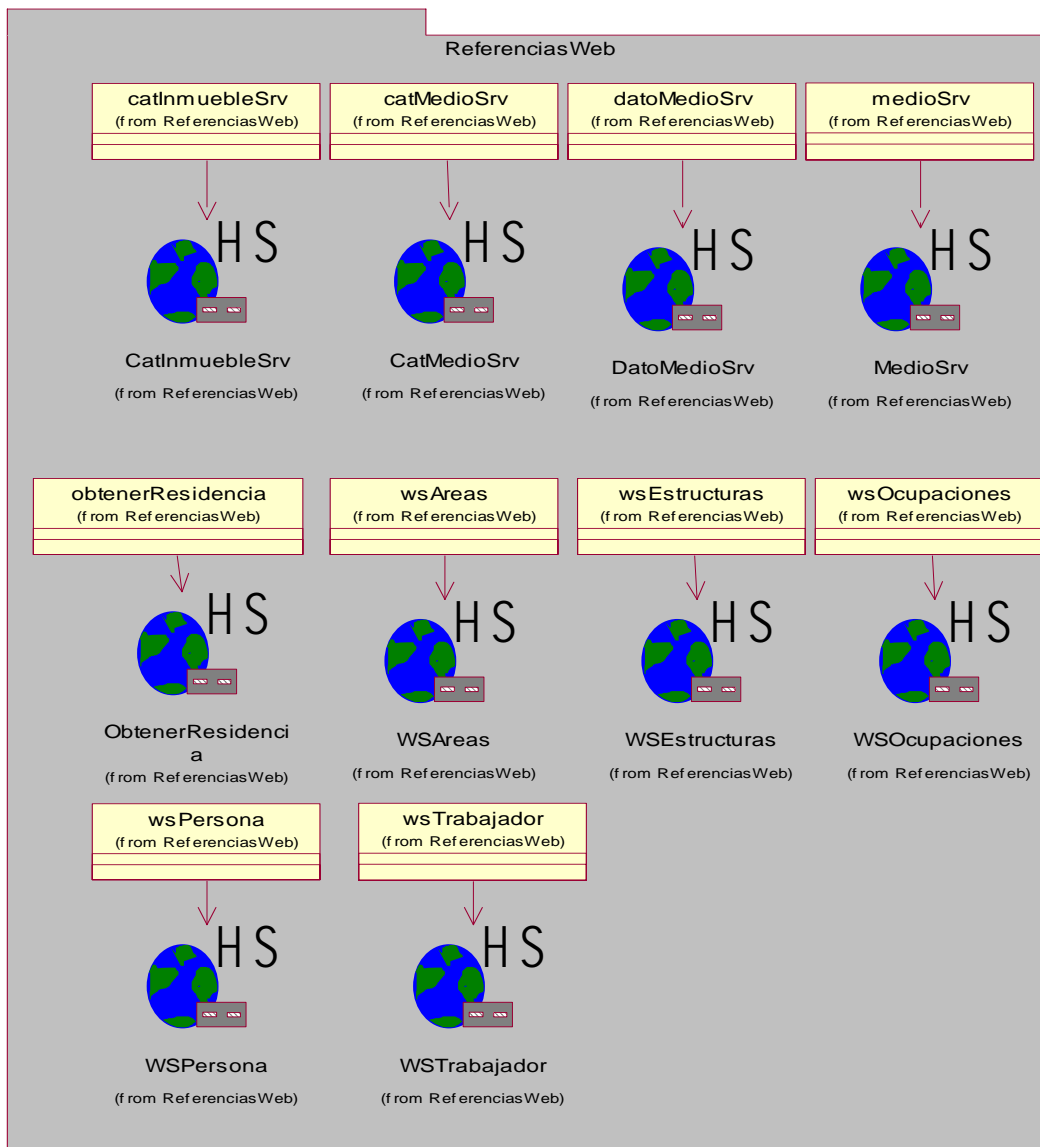


Figura 11. Diagrama de clases del paquete Referencias Web.

### 4.2.3 Paquete Entidades.

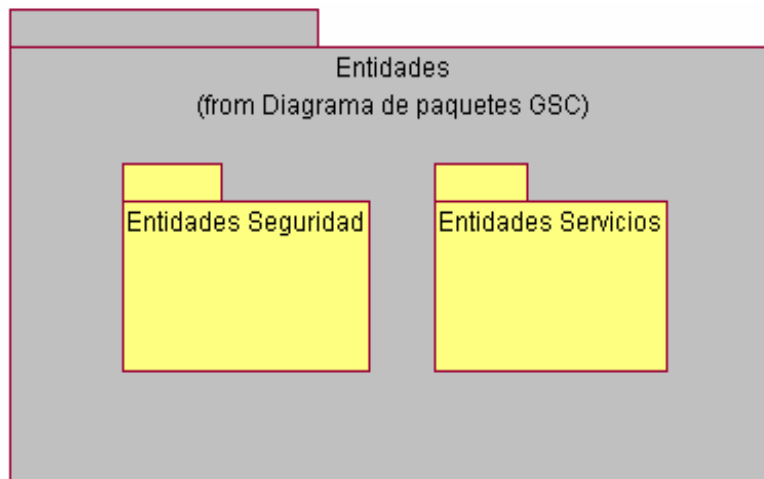


Figura 12. Diagrama de clases del paquete Entidades

#### 4.2.3.1 Paquete Entidades Seguridad.

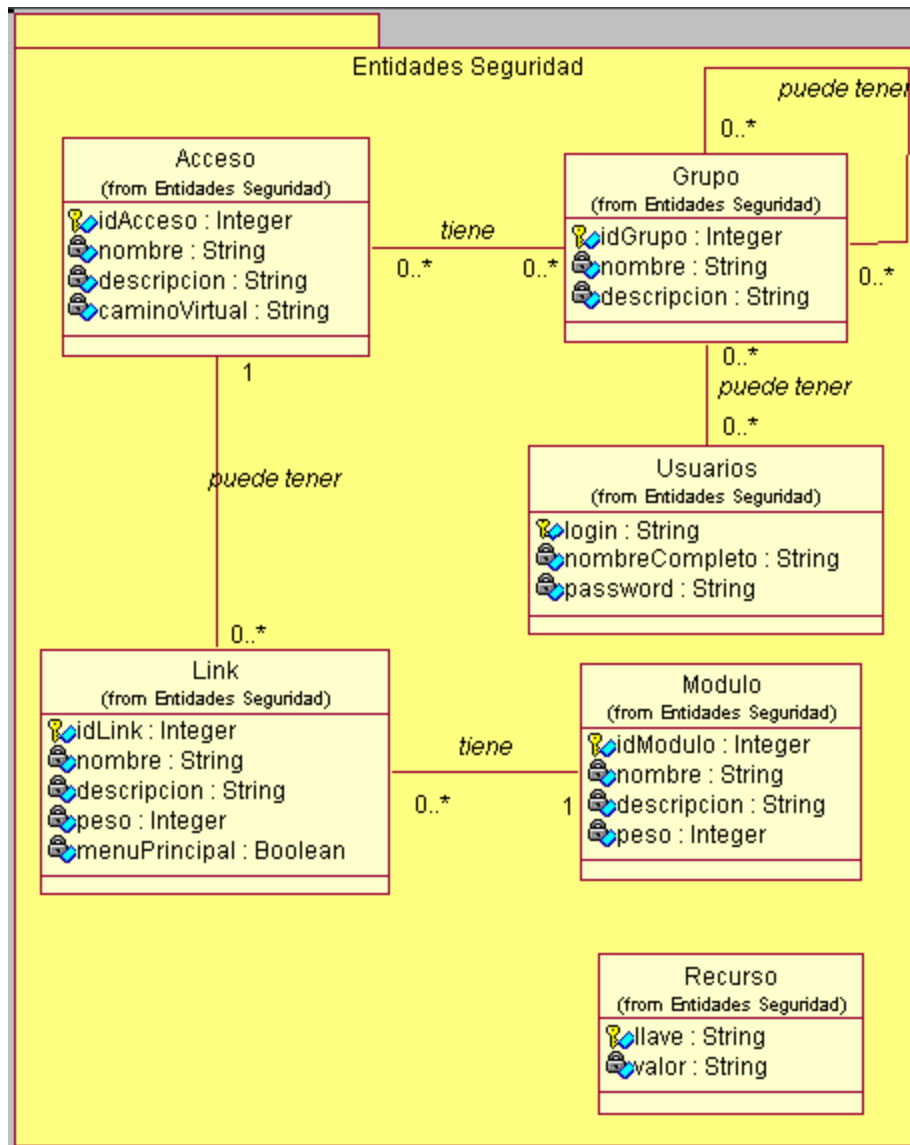


Figura 13. Diagrama de clases del paquete Entidades Seguridad

#### 4.2.3.2 Paquete Entidades Servicios.

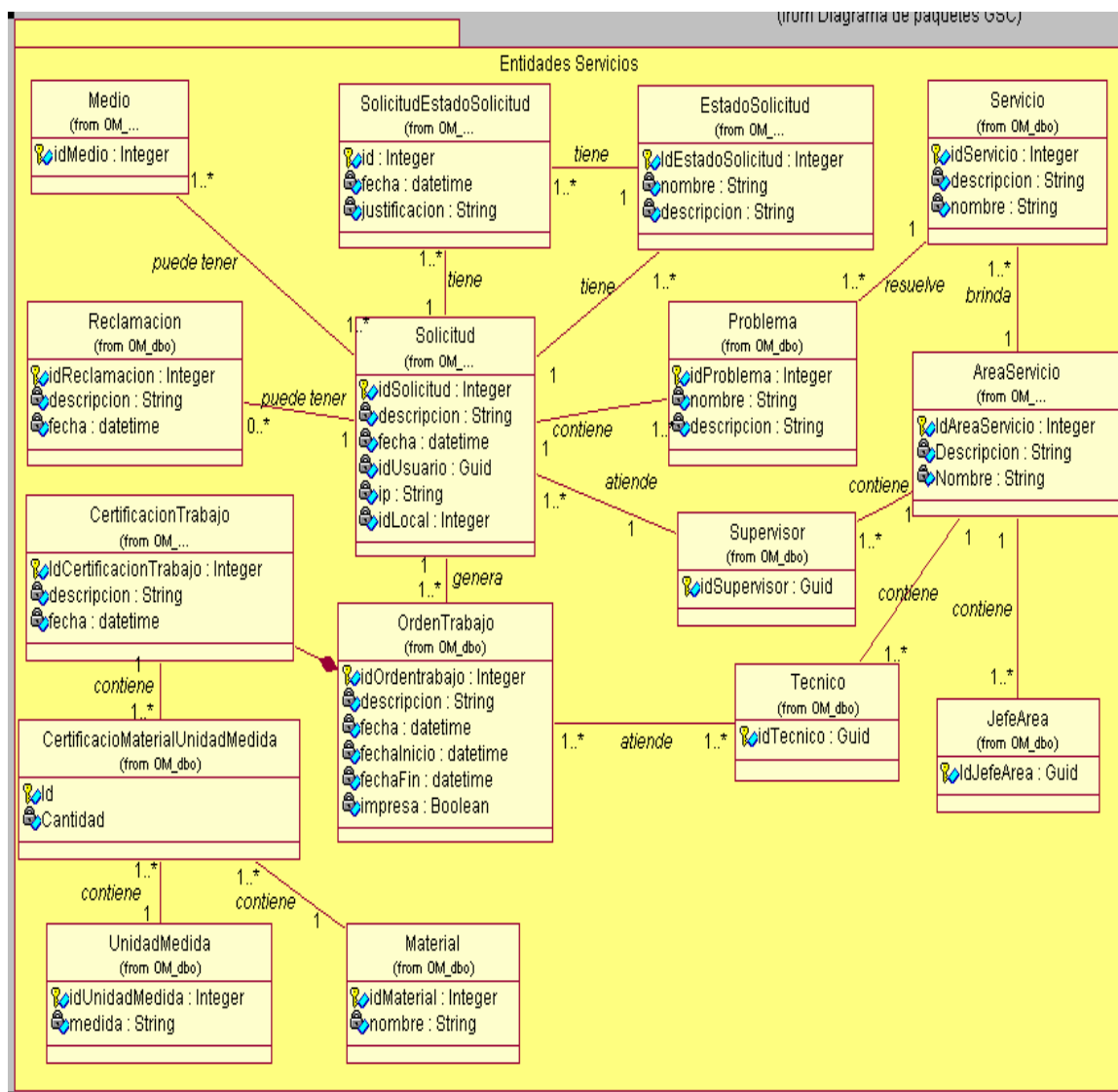


Figura 14. Diagrama de clases del paquete Entidades Servicios

#### 4.2.4 Paquete Procesamiento Lógico.

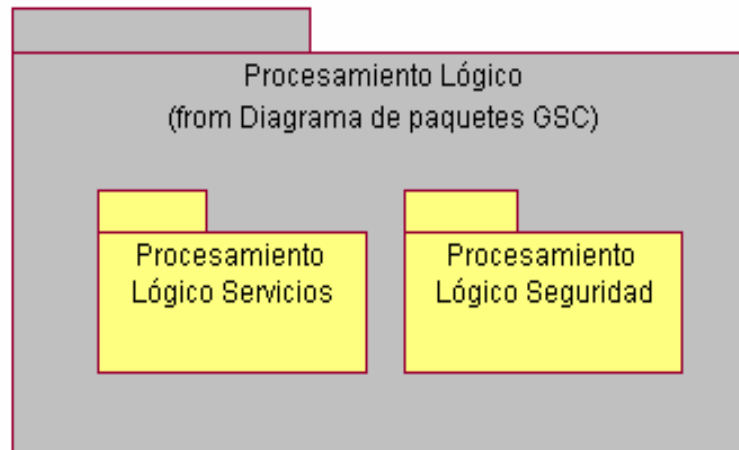


Figura 15. Diagrama de clases del paquete Procesamiento Lógico



#### 4.2.4.1 Procesamiento lógico Servicios (Pertenece al paquete procesamiento Lógico)

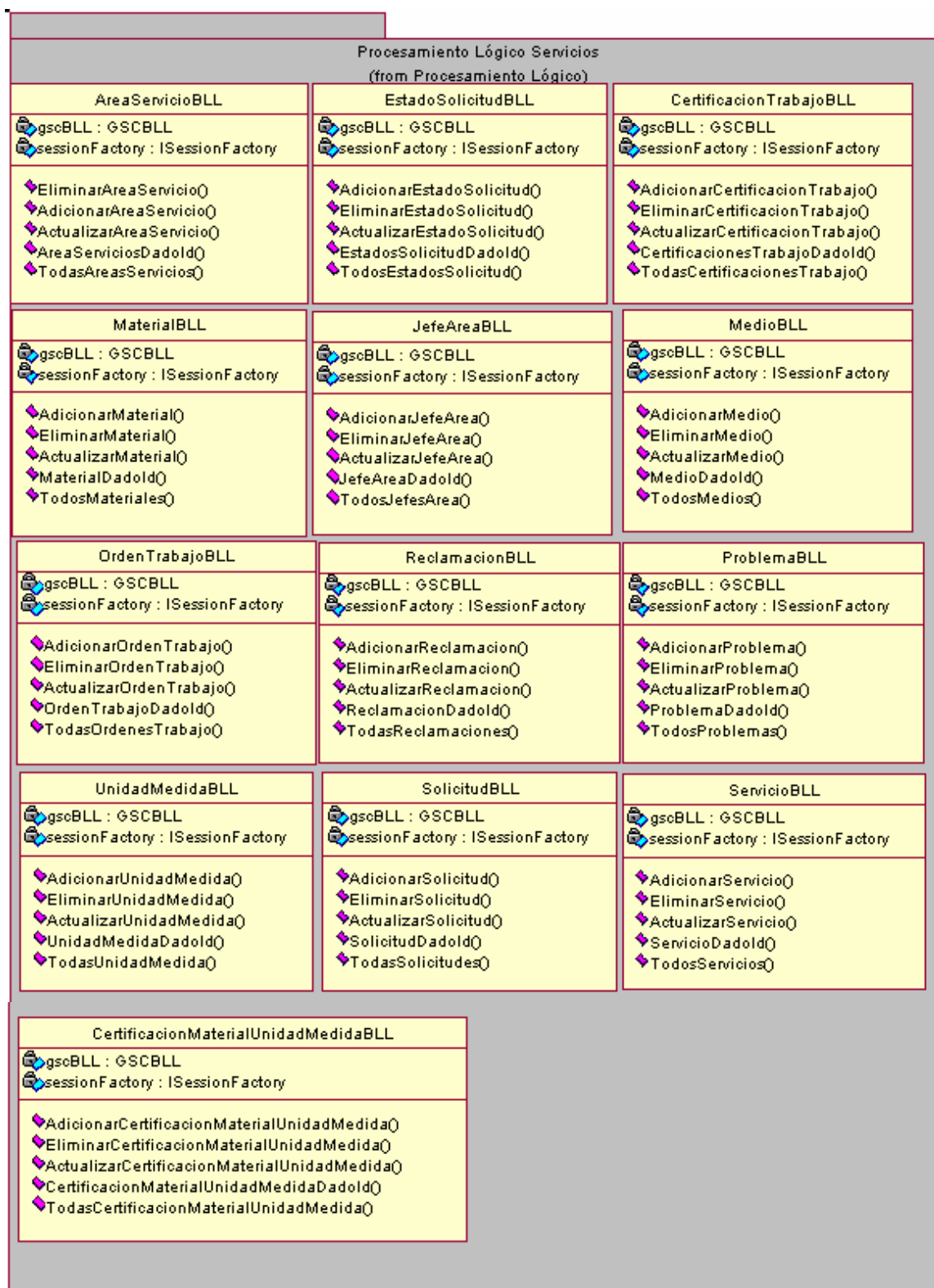


Figura 16. Diagrama de clases del paquete Procesamiento Lógico Servicios.

#### 4.2.4.2 Procesamiento lógico Seguridad (Pertenece al paquete Procesamiento Lógico)

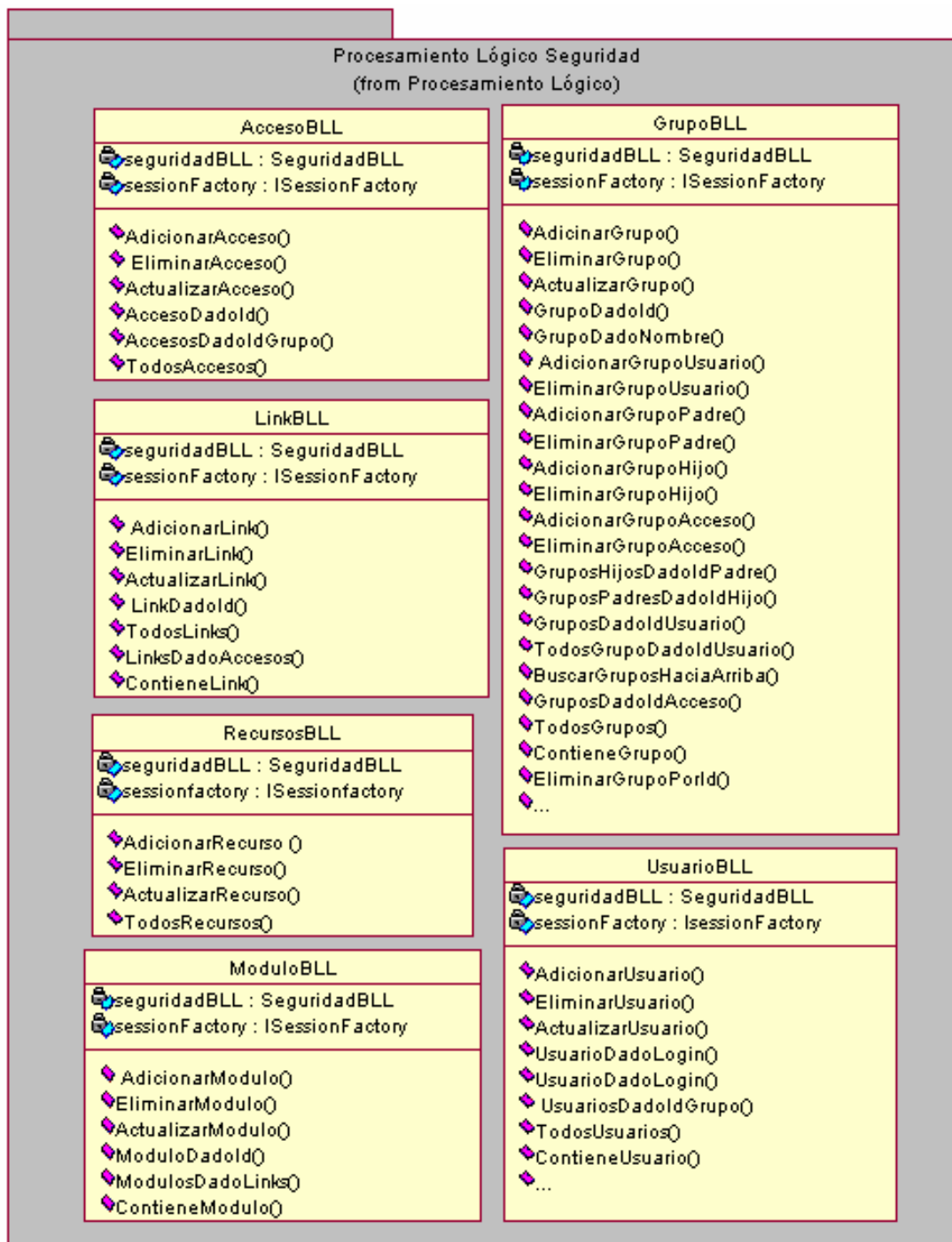


Figura 17. Diagrama de clases del paquete Procesamiento Lógico Seguridad

### 4.2.5 Paquete presentación

El paquete de presentación ha sido dividido en varios diagramas por su complejidad y tamaño. Cada diagrama representa un módulo o parte de este.

#### 4.2.5.1 Módulo General.

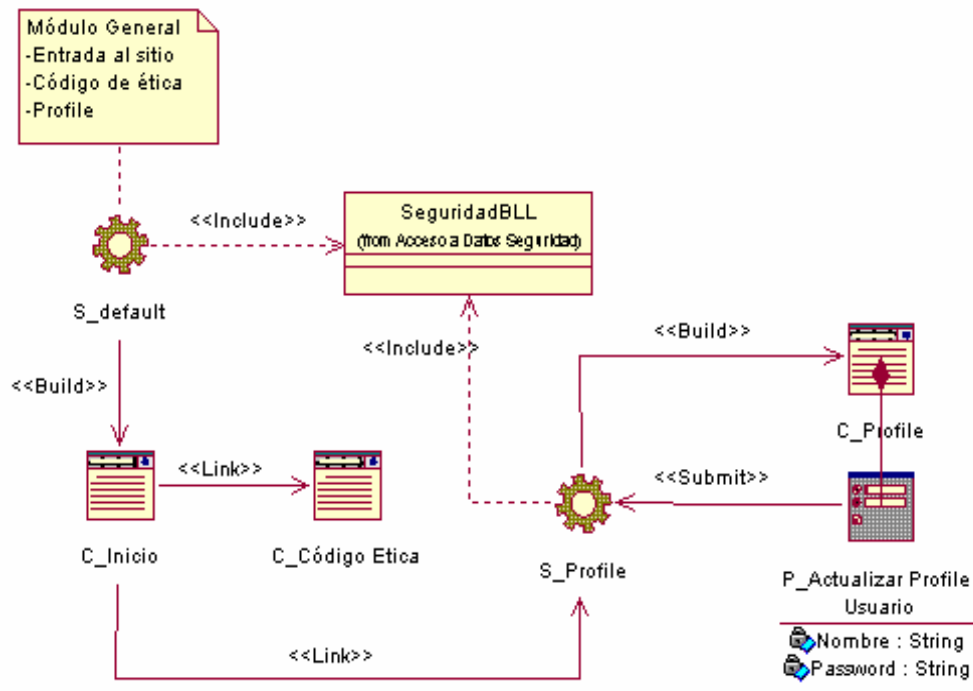


Figura 18. Diagrama de clases del paquete presentación, módulo general.

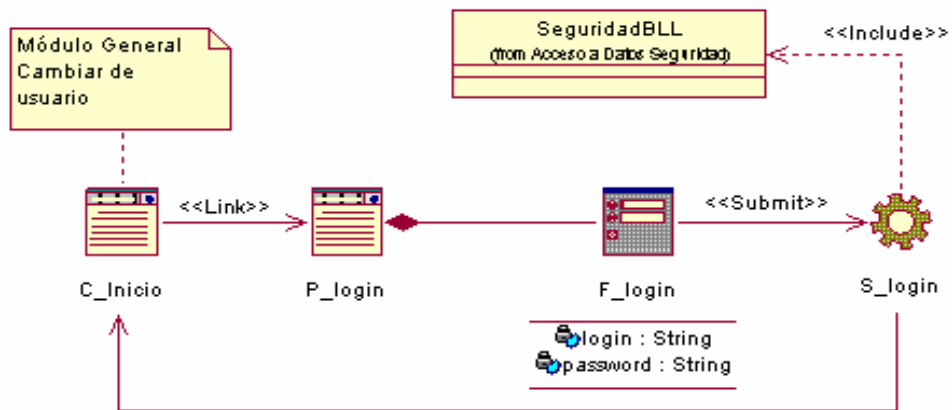


Figura 19. Diagrama de clases del paquete presentación, módulo general, cambiar de usuario.



### 4.2.5.3 Módulo Administración General.

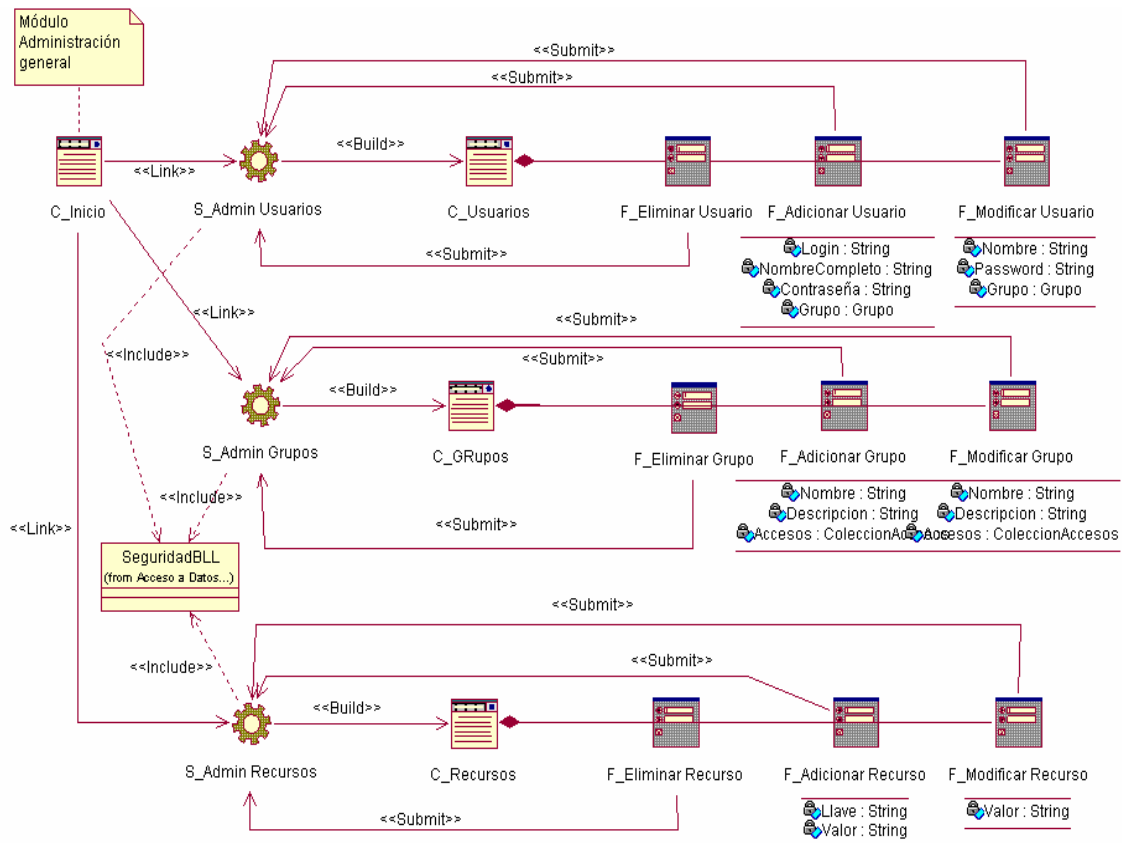


Figura 21. Diagrama de clases del paquete presentación, módulo Administración General.

#### 4.2.5.4 Módulo Sistema.

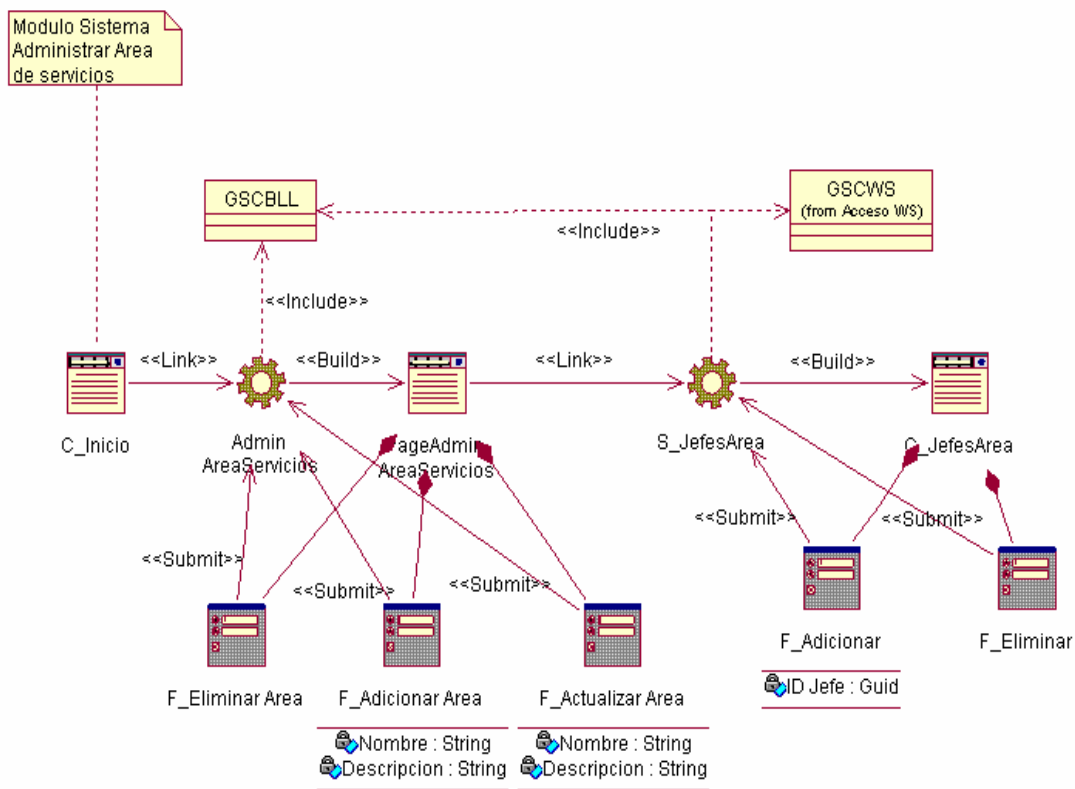
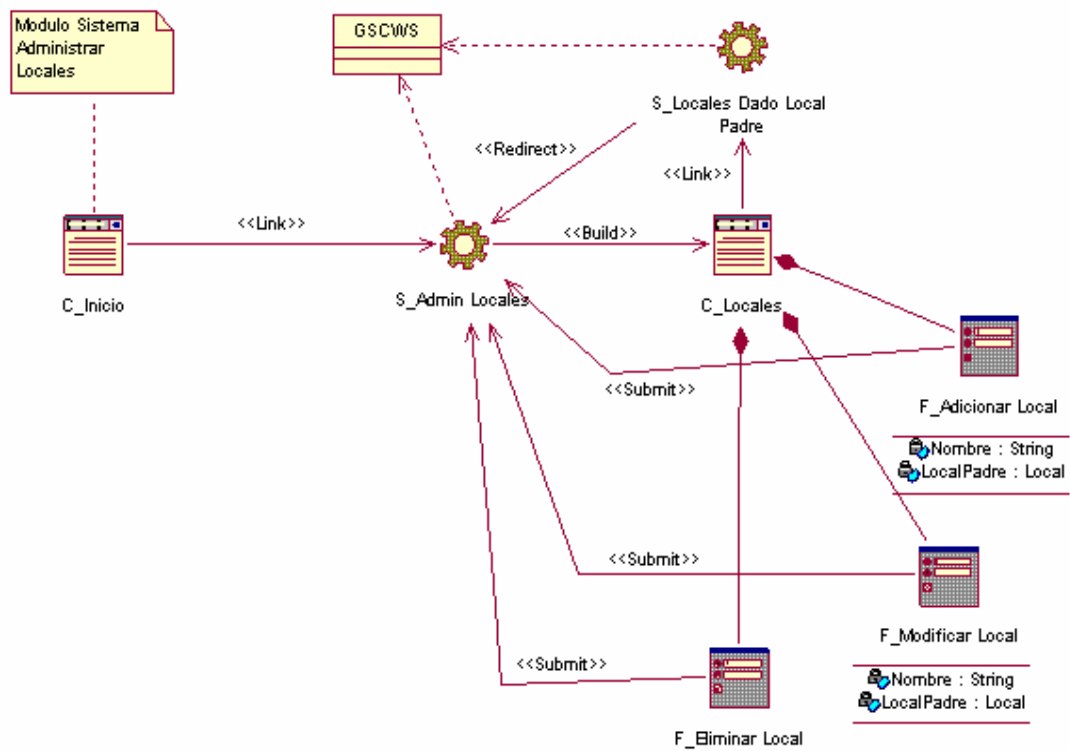
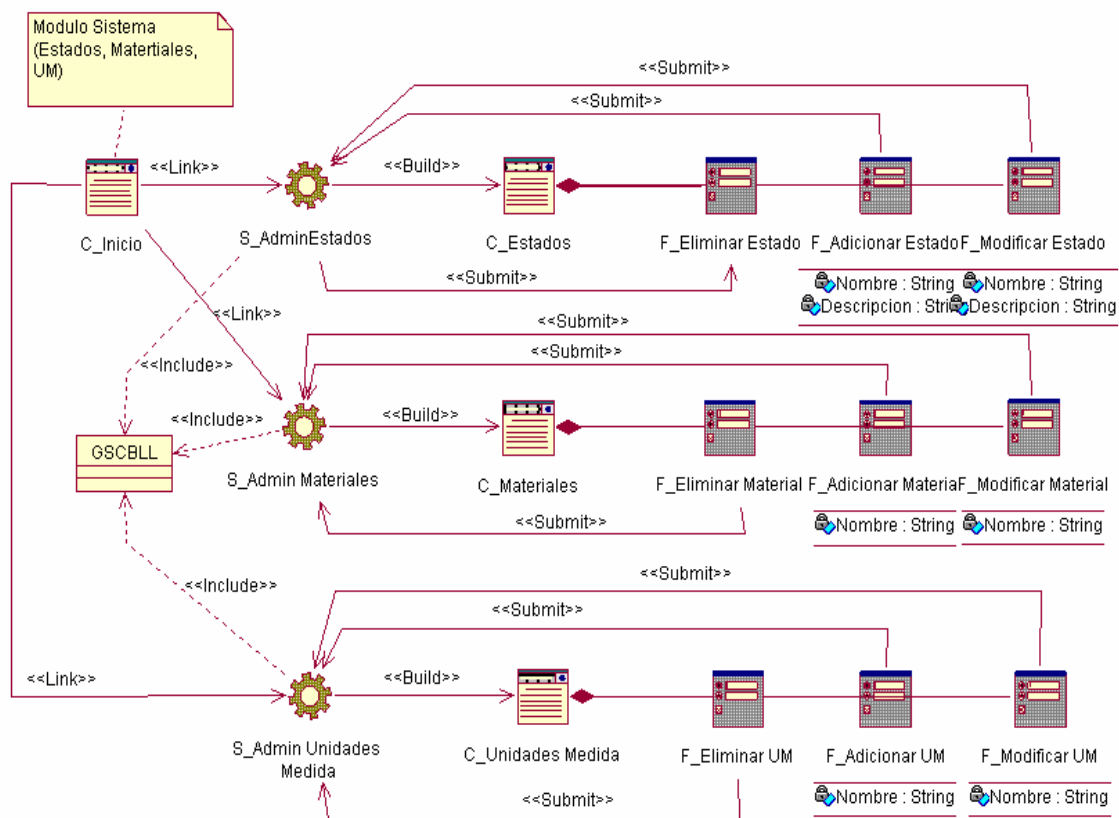


Figura 22. Diagrama de clases del paquete presentación, módulo Sistema, Administrar Áreas de Servicio.



**Figura 23. Diagrama de clases del paquete presentación, módulo Sistema, Administrar Locales.**





**Figura 24.**Diagrama de clases del paquete presentación, módulo Sistema, Administrar Estados, materiales y Unidades de medida.

#### 4.2.5.5 Módulo Área de Servicios.

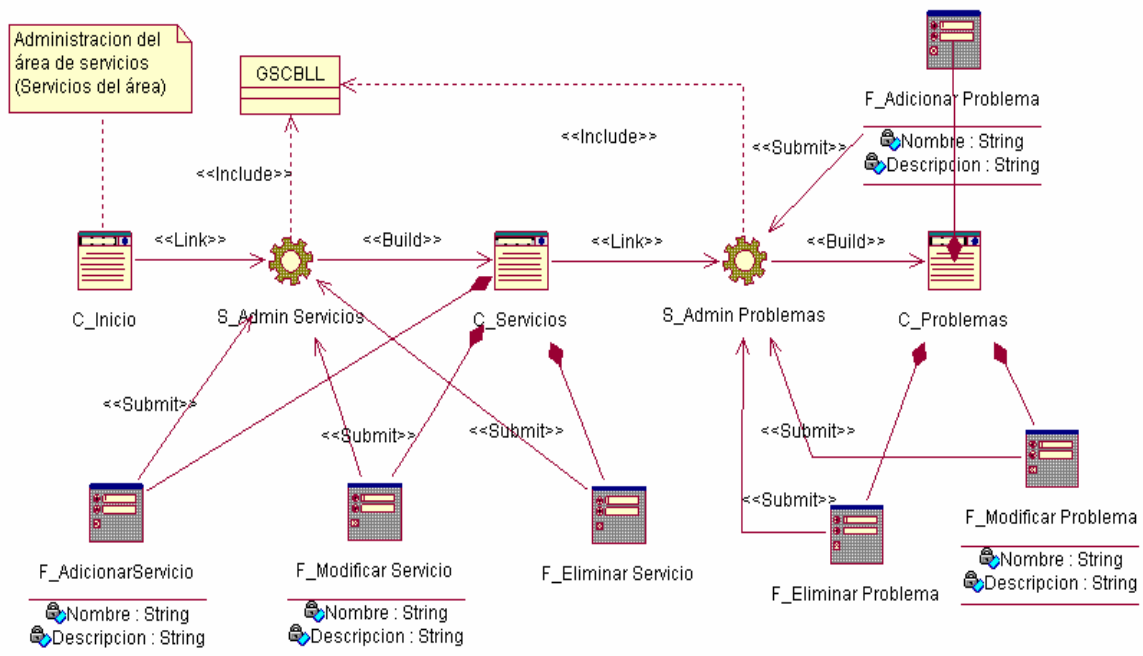


Figura 25. Diagrama de clases del paquete presentación, módulo Área de Servicios, Administrar Servicios que brinda el área.

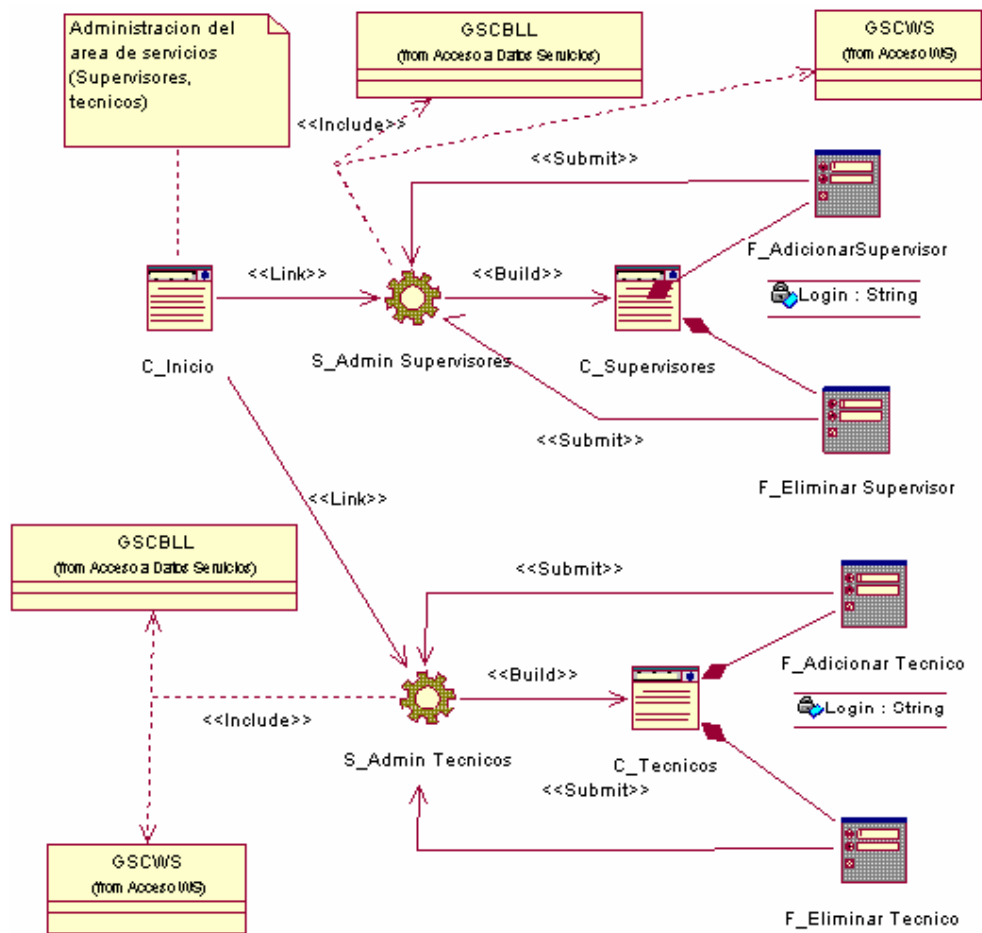


Figura 26. Diagrama de clases del paquete presentación, módulo Área de Servicios, Administrar Supervisores y Técnicos.

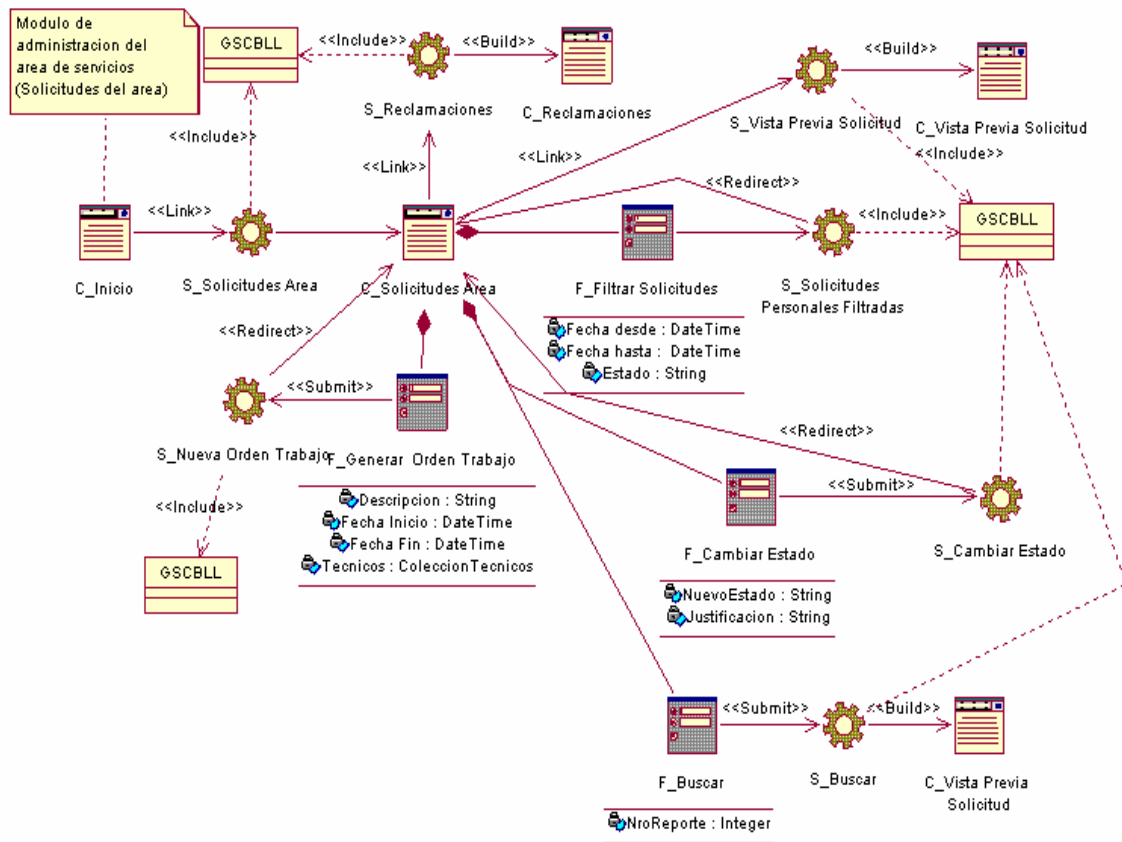


Figura 27. Diagrama de clases del paquete presentación, módulo Área de Servicios, Solicitudes hechas al área.

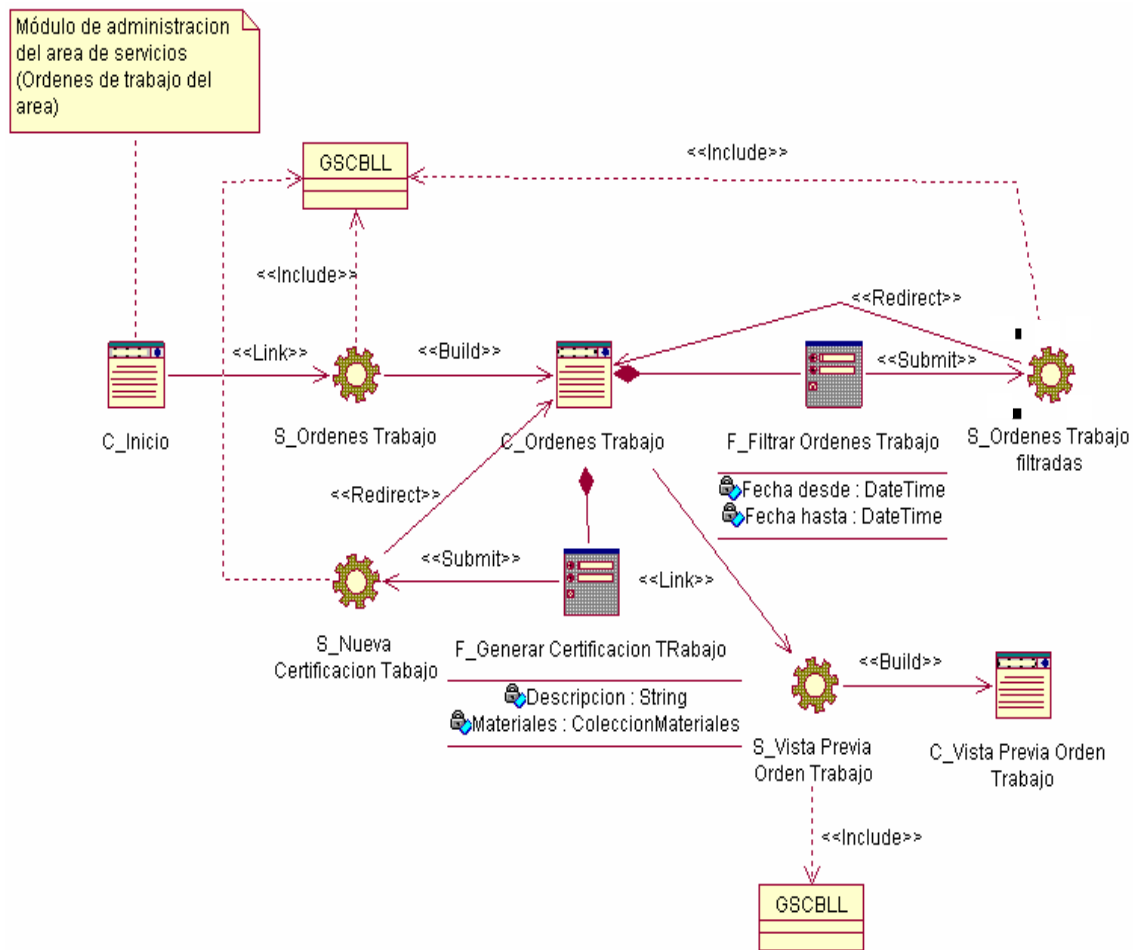
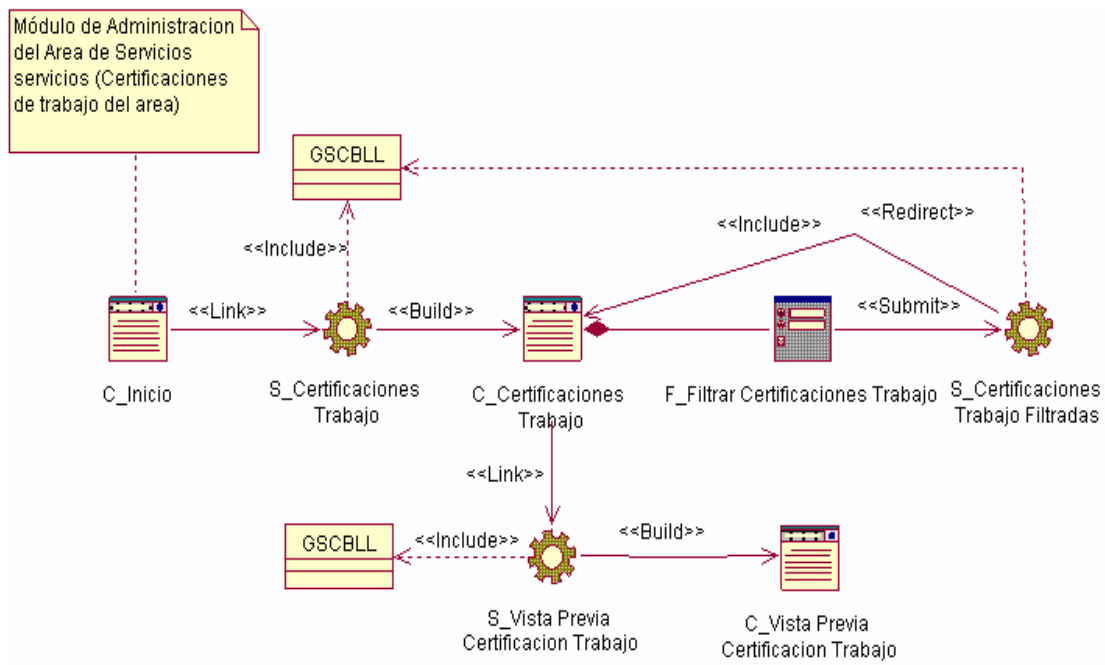


Figura 28. Diagrama de clases del paquete presentación, módulo Área de Servicios, Órdenes de Trabajo del Área.



**Figura 29. Diagrama de clases del paquete presentación, módulo Área de Servicios, Certificaciones de Trabajo del Área.**

4.2.5.6 Módulo Servicios.

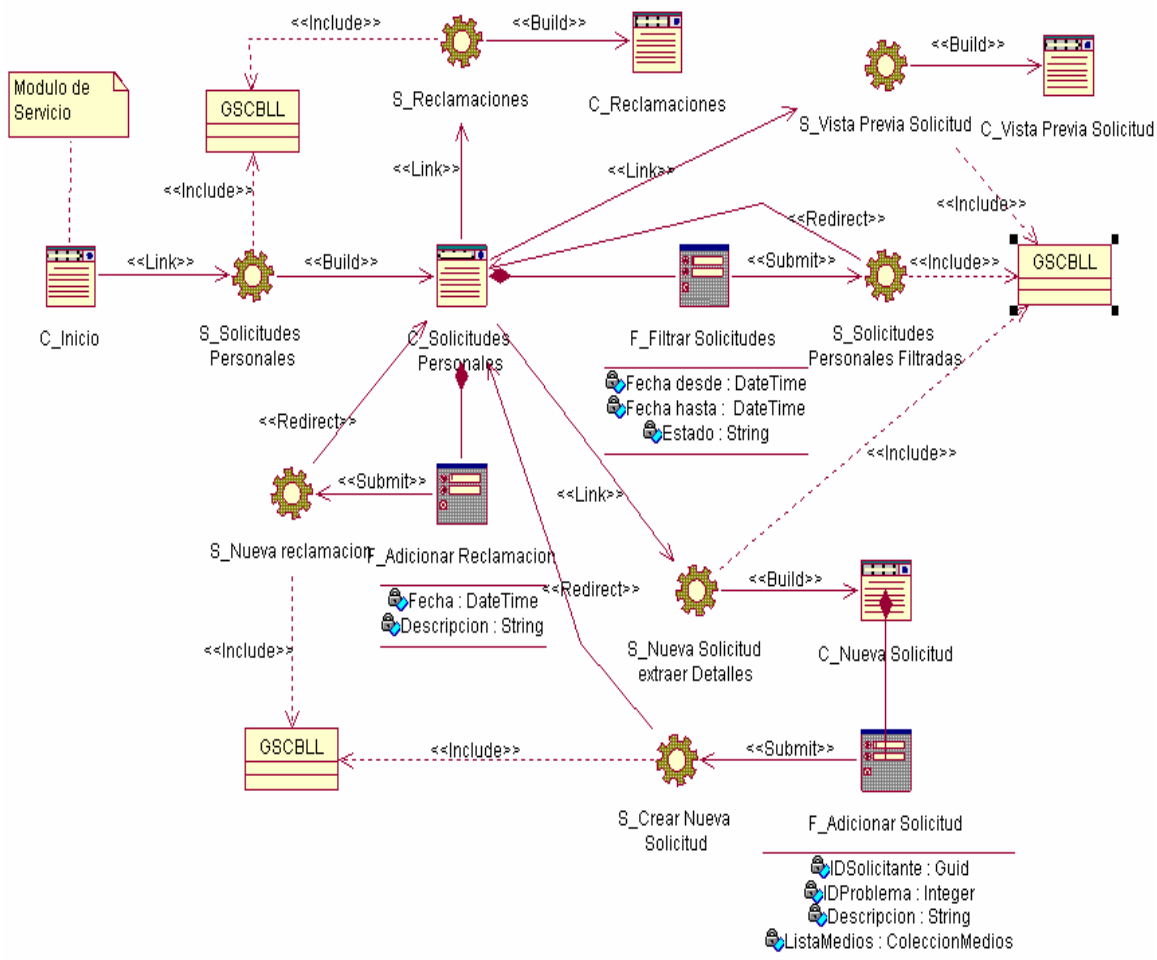


Figura 30. Diagrama de clases del paquete presentación, módulo de Servicios, Nueva solicitud, Historial de solicitudes y reclamaciones.

#### 4.2.5.7 Módulo Reportes.

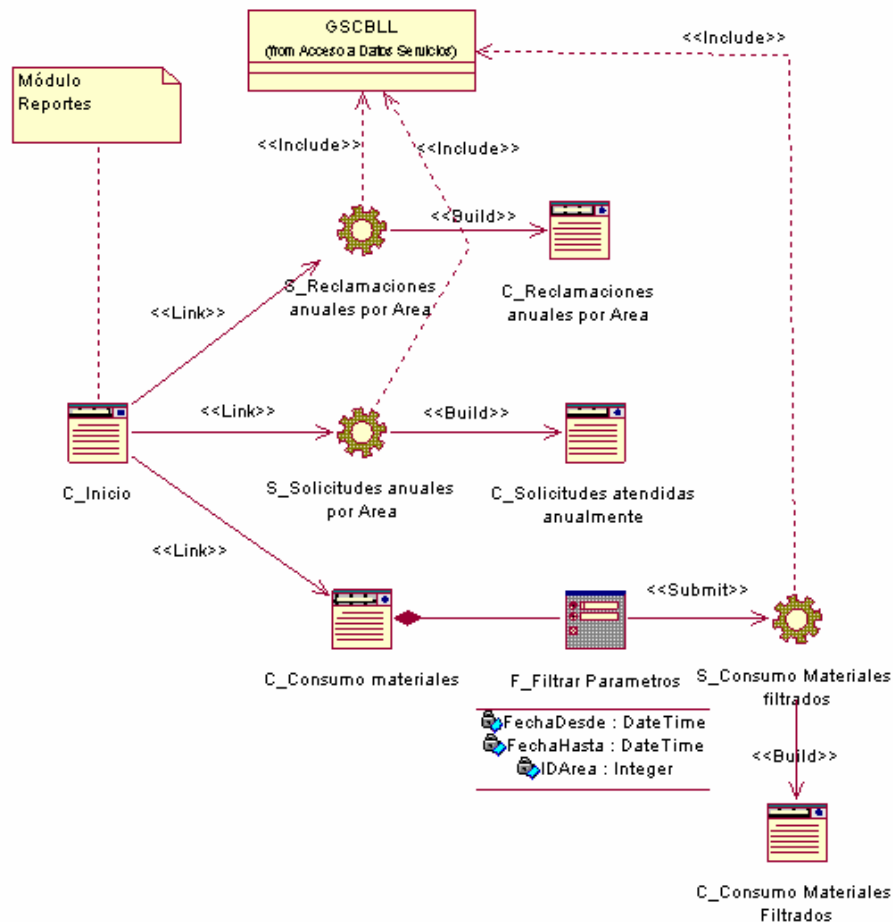


Figura 31. Diagrama de clases del paquete presentación, módulo de reportes.

### 4.3 Principios de diseño

El diseño ha sido elaborado pensando en los usuarios finales, que serán: estudiantes, profesores y trabajadores, los cuales, no en todos los casos poseen conocimientos sobre computación, por tanto, se ha elegido una interfaz amigable e intuitiva. Se ha mantenido un diseño consistente en todas las páginas, para lograr que el usuario se sienta cómodo y logre acostumbrarse rápidamente a la aplicación.

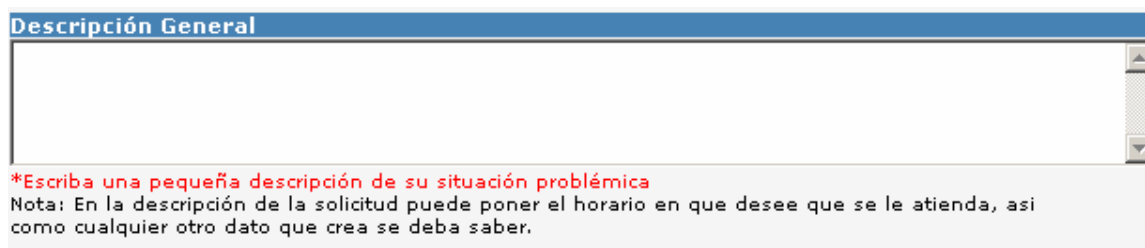
#### 4.3.1 Interfaz de usuario

Para el diseño de la interfaz de usuario fueron aplicadas las siguientes pautas:



## Páginas:

Para la aplicación se definió un estándar para todas las páginas, existe una principal, la cual tiene opciones comunes para todos los tipos de usuario (*Ver anexo I*). Además del diseño de la página principal se crearon dos nuevos diseños de páginas, una para la entrada de datos con características especiales (entiéndase por características especiales las entradas de solicitudes, órdenes de trabajo y certificaciones de trabajo, pues las mismas llevan un nivel de complejidad superior a las demás entradas de datos (*Ver anexo II*)) y otra para la salida de reportes. Se utiliza el color rojo para resaltar errores de campos requeridos (Fig. 32), con formato incorrecto o mensajes de operaciones no válidas (Fig. 33). De forma general se realizan varias operaciones por página, de forma que el usuario no tenga que moverse tanto dentro de la aplicación. Todos los prototipos de páginas se rigen por la misma hoja de estilos con el fin de ajustarse rigurosamente a los estándares planteados por la aplicación.

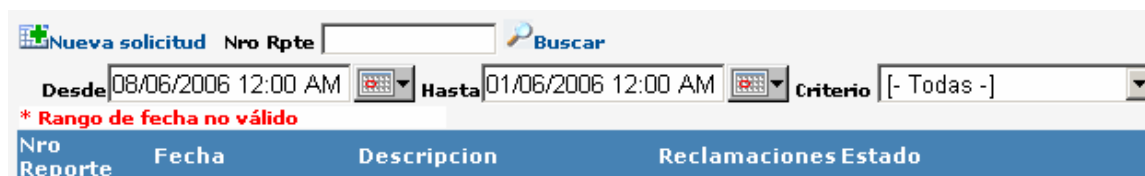


**Descripción General**

\*Escriba una pequeña descripción de su situación problemática

Nota: En la descripción de la solicitud puede poner el horario en que desee que se le atienda, así como cualquier otro dato que crea se deba saber.

**Figura 32. Uso del color rojo para mensajes de error, campos requeridos.**



Nueva solicitud Nro Rpte  Buscar

Desde 08/06/2006 12:00 AM Hasta 01/06/2006 12:00 AM Criterio [- Todas -]

\* Rango de fecha no válido

Nro Reporte	Fecha	Descripción	Reclamaciones Estado
-------------	-------	-------------	----------------------

**Figura 33. Uso del color rojo para mensajes de error, operaciones incorrectas.**

## Hoja de estilo:

La hoja de estilo utilizada se basa principalmente en el uso de colores azules, grises y blancos con el fin de dar claridad al diseño, determinado rigurosamente el estilo de tablas, encabezamientos de tablas, de páginas, etiquetas y cuadros de textos, así como los títulos de páginas y sus márgenes.

### Menú lateral:

El menú lateral contendrá el acceso a las páginas fundamentales de la aplicación, este será dinámico en dependencia rol del usuario que se encuentre en el sistema. (Ver anexo III)

### Botones:

El diseño de los botones de las páginas variará en dependencia de la operación a realizar, mostrando una pequeña imagen de 16 x 16, para una orientación visual de la operación a realizar, no obstante los mismos tendrán un texto descriptivo puesto que el sistema esta dirigido no solo a personas vinculadas al campo de la informática. (Fig. 34)



**Figura 34. Prototipo de los botones, botones para editar, adicionar y eliminar.**

### 4.3.2 Formato de salida de los reportes

El sistema brinda reportes en forma de tablas. En estos se da la posibilidad de filtrar los resultados mostrados por uno o más campos, de manera que el usuario pueda obtener de forma sencilla y sin dificultad la información que desea ver. Las filas de los reportes son de color alterno con el objetivo de facilitar la lectura del mismo. Los colores de los reportes fueron escogidos igualmente para que permitan una cómoda lectura de este. Los reportes además permiten paginado, por lo que se muestran un número limitado de registros, permitiendo moverse adelante y hacia atrás (Fig. 35).

Solicitudes hechas a Mantenimiento Constructivo				
Desde	01/06/2006 12:00 AM	Hasta	30/06/2006 12:00 AM	Criterio [- Todas -] <span>Mostrar</span>
Nro Reporte	Fecha	Descripcion	Reclamaciones	Estado
71	05/06/2006	Problemas eléctricos en el cuarto	0	Pendiente por revisión
72	05/06/2006	Se rompio el brequer del cuarto	0	Pendiente por revisión
73	05/06/2006	Problemas de pase a tierra de corriente.	0	Pendiente por revisión

**Figura 35. Ejemplo de un Reporte.**

Además el sistema de reportes se mostrará en forma de gráfico, para un mejor entendimiento de los reportes, y con el objetivo de ser usados en la toma de decisiones por los directivos de la universidad.

### 4.3.3 Ayuda

La ayuda se muestra como parte del menú principal de cada página de la aplicación (Fig. 36). Además existe un manual de usuario, disponible para descargar desde cualquier lugar de la universidad, el cual contiene la lista de preguntas frecuentes con sus respuestas.

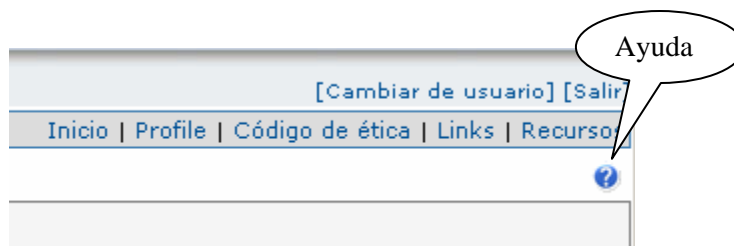


Figura 36. Acceso directo a la ayuda de la página actual.

## 4.4 Tratamiento de errores

Con el objetivo de prevenir los errores por parte del usuario, solo se le piden los datos necesarios, por ejemplo, se deshabilitan ciertos botones si el usuario no tiene que utilizarlos en ese momento, además se verifica la integridad de los mismos antes de ser almacenados para evitar la inconsistencia. Los campos obligatorios son verificados, para ello el sistema usa validaciones, tanto en la parte del cliente, como en la parte del servidor. En el caso de las excepciones en tiempo de ejecución, estas son capturadas por el sistema y almacenadas en una base de datos, para su posterior revisión por parte de los desarrolladores del sistema, con el objetivo de mejorar la aplicación. Se utilizan mensajes de confirmación, para acciones que son irreversibles como es el caso de las eliminaciones (Fig. 37).

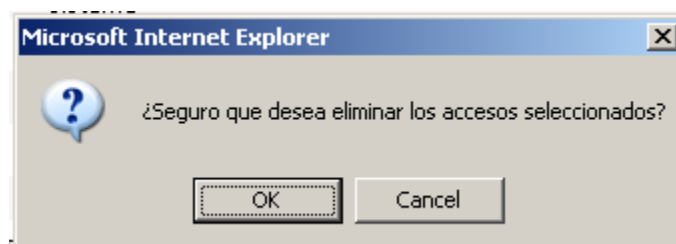
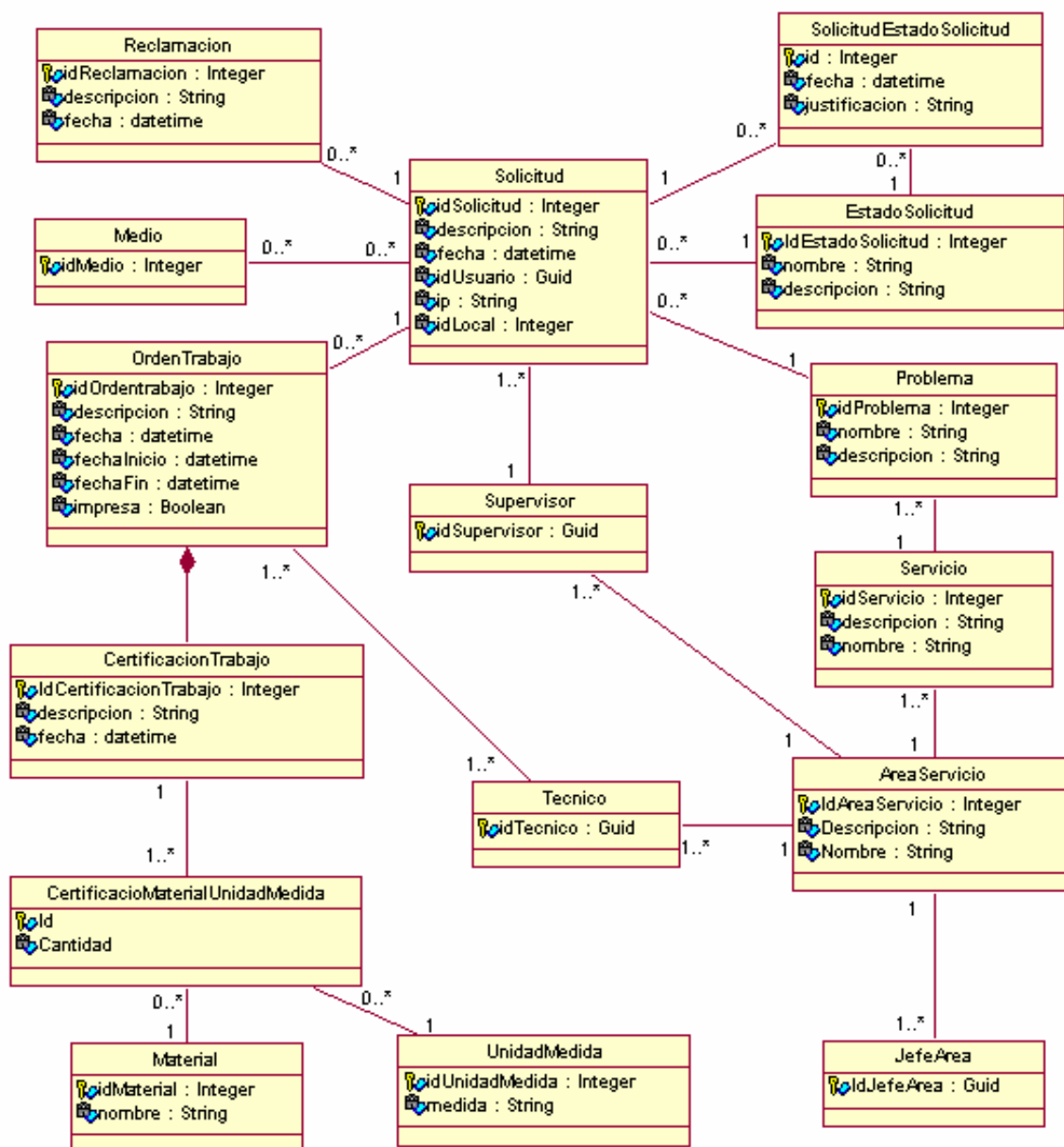
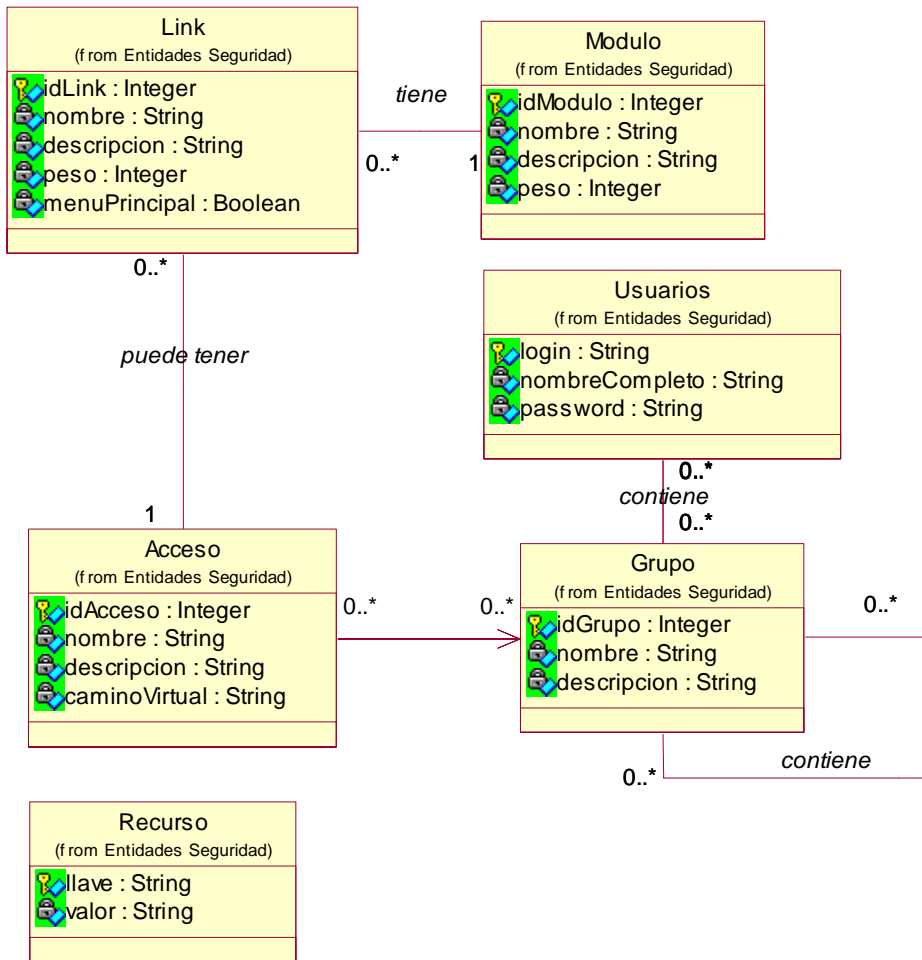


Figura 37. Utilización de mensajes de confirmación.

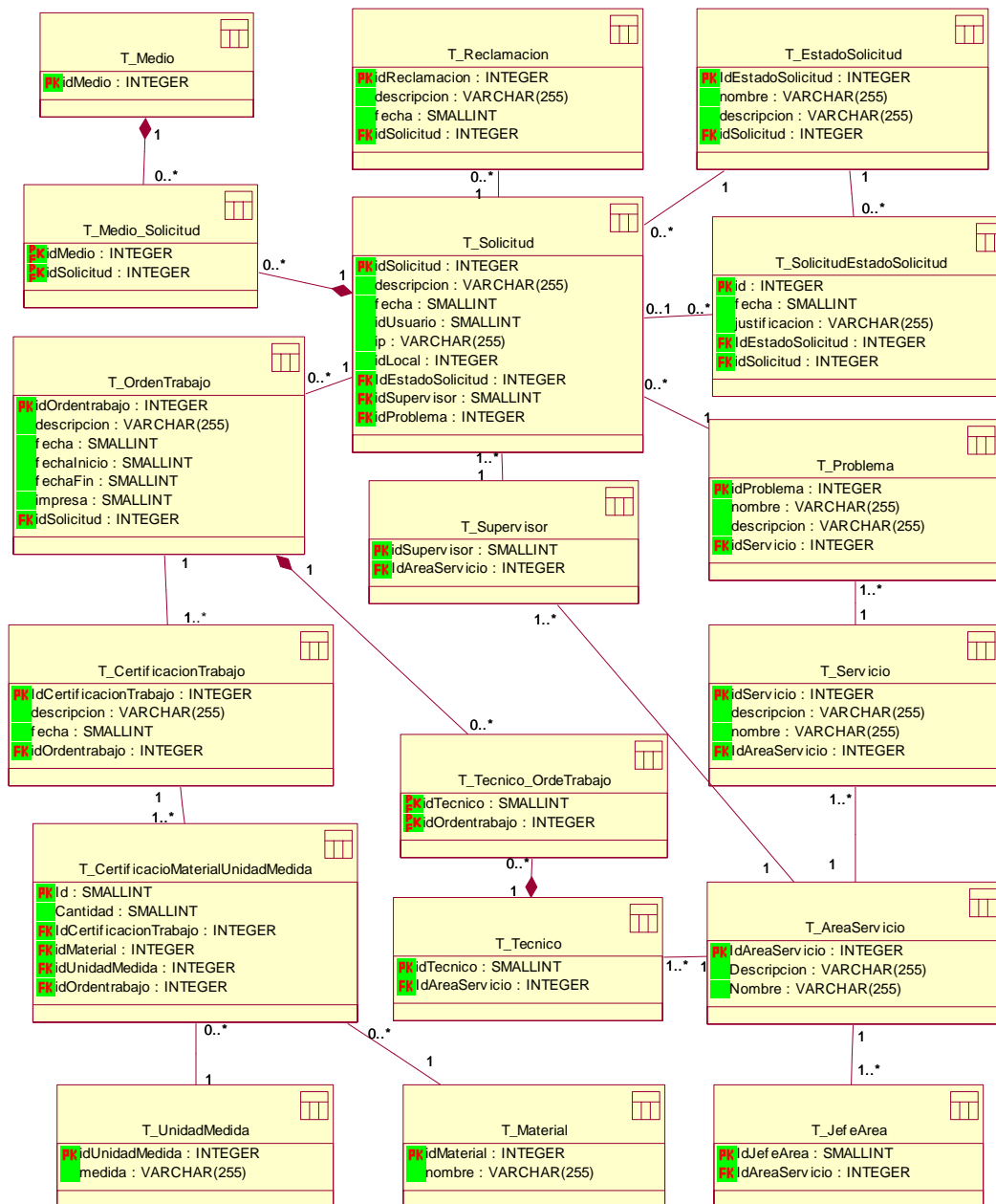
## 4.5 Diseño de la base de datos

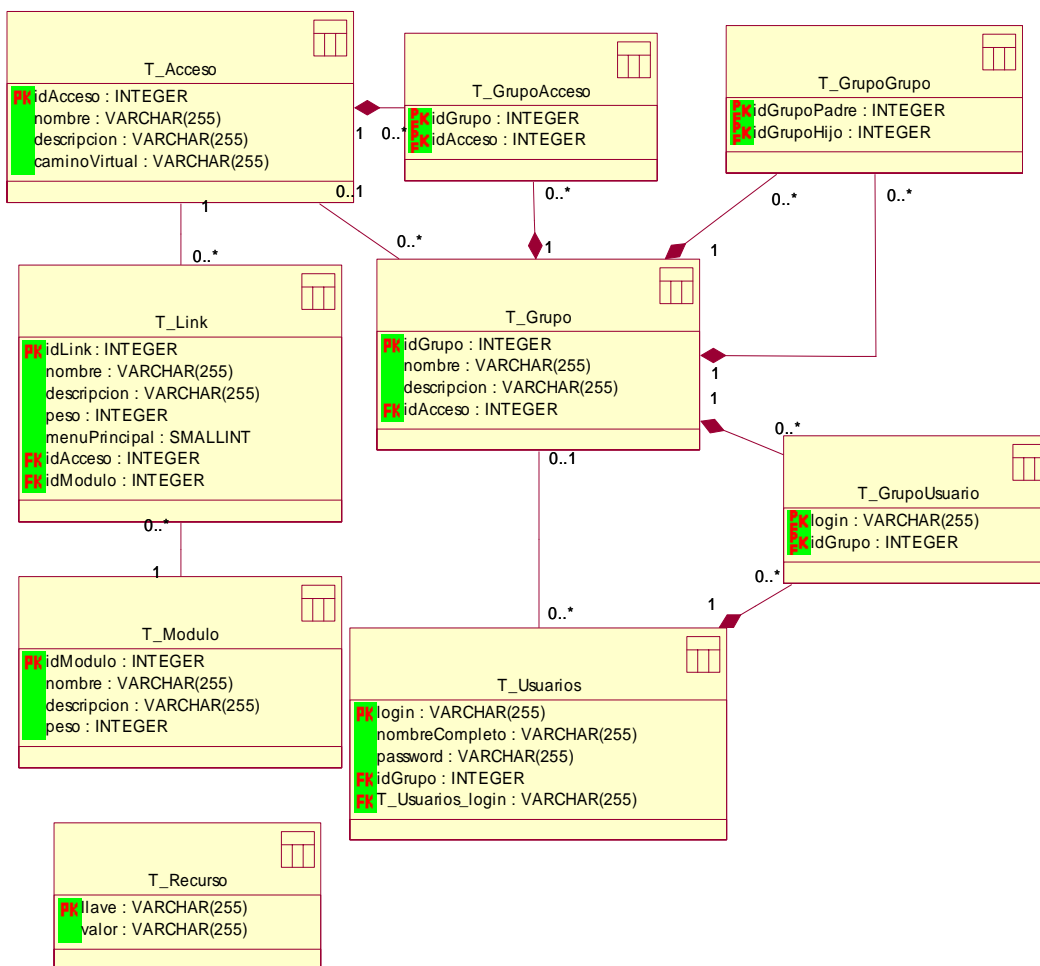
### 4.5.1 Modelo lógico de datos.





### 4.5.2 Modelo físico de datos



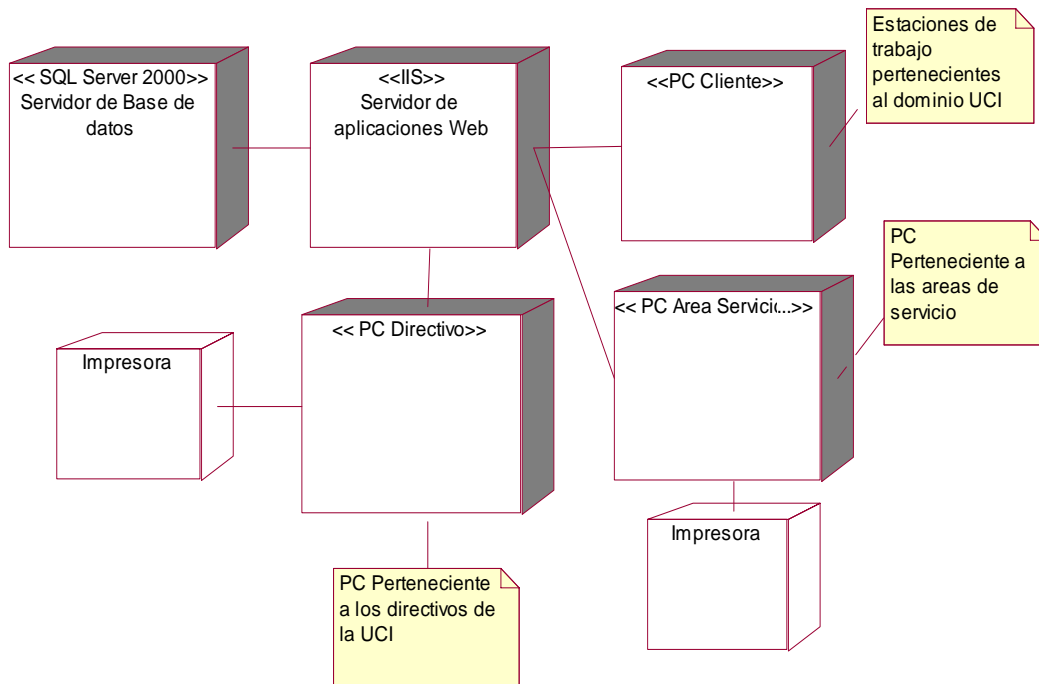


## 4.6 Diagrama de despliegue

El siguiente diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software.

Como observamos en la figura el diagrama de despliegue nos indica que para el correcto funcionamiento de la aplicación es necesario la presencia de tres nodos elementales, los cuales no son más que medios de cómputo: el servidor de bases de datos, el servidor de aplicaciones y las PC, ya sean estas últimas de usuarios, de áreas de servicio o de directivos, además se representa un dispositivo indispensable para las áreas de servicio y para los directivos, la impresora. Entre las PC y el servidor de aplicaciones Web existe una comunicación por HTTP, entre el servidor de bases de datos y el servidor de aplicaciones hay una comunicación por TCP/IP y la comunicación entre las PC y la impresora es por puerto USB o puerto paralelo. Se decidió utilizar esta

arquitectura de capas por la naturaleza de la aplicación, pues la misma se desarrolla con interfaz Web. La aplicación cuenta con tres capas, la primera es la interfaz de usuario, la que se encontrará en todas las máquinas que soliciten el servicio, la misma solicitará a la capa de negocio una petición, esta se encontrará en el nodo que representa el servidor de la aplicación y esta a su vez interactuará con la capa de datos, representada por el servidor de bases de datos. Para una mejor descripción ver anexo IV.



**Figura 38. Diagrama de despliegue**

## 4.7 Conclusiones

En el presente capítulo se mostraron los resultados de la etapa de diseño del sistema se obtuvo el diagrama de clases donde se representaron las clases y sus asociaciones. Se plantean principios de diseño que ayudan a un mejor diseño del Sistema. Se presentó la concepción del tratamiento de errores, y el sistema de ayuda. Además se realizó el diseño de la base de datos y el diagrama de despliegue. Todos estos elementos obtenidos brindan una idea mucho más clara de la estructura del sistema e influyen en el logro de una mejor herramienta.



## 5 Estudio de factibilidad

### 5.1 Introducción

Es importante evaluar la factibilidad de un proyecto antes de su elaboración, para conocer si es conveniente llevarlo a cabo. La viabilidad y el análisis de riesgo están relacionados de muchas maneras. Si el riesgo del proyecto es alto, la viabilidad de producir software de calidad se reduce. En el presente capítulo se hace un estudio de factibilidad, beneficios y costo del sistema propuesto, utilizando estimación basada en puntos de casos de uso que es un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de pesos a determinados factores que lo afectan para contabilizar el tiempo total estimado para ese proyecto a partir de esos factores.

### 5.2 Planificación basada en casos de uso

#### 5.2.1 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

**Tabla 29. Factor de peso de los actores sin ajustar.**

Tipo de actor	Descripción	Factor de peso	Actores	UAW
Simple	Otro sistema que interactúa con el sistema a desarrollar, mediante una interfaz de programación.	1	0	0
Medio	Sistema con sistema mediante protocolo de interfaz basada en texto.	2	0	0

Complejo	Persona que interactúa con el sistema mediante interfaz gráfica.	3	5	15
			Total	15

$$UAW = \sum cant \text{ actores} * peso$$

$$UAW = 15$$

**Tabla 30. Factor de peso de los casos de uso sin ajustar**

Tipo de CU	Descripción	Peso	Cantidad de CU	Total
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	19	95
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	4	40
Complejo	El caso de uso tiene más de 8 transacciones.	15	-	0
			Total	135

$$UUCW = \sum cant \text{ CU} * peso$$

$$UUCW : 135$$

$$UUCP = UAW + UUCW$$

$$UUCP = 15 + 135$$

$$UUCP = 150$$

**Cálculo de los Puntos de casos de uso ajustados.**

$$UCP = UUCP * TCF * EF$$

Donde:

*UCP : Factor de Complejidad Técnica*

*UUCP : Puntos de Casos de Uso Sin Ajustar*

*TCF : Factor de Complejidad Técnica*

*EF : Factor de Ambiente*

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 desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante).

**Tabla 31. Factor de complejidad técnica.**

<b>Factor</b>	<b>Descripción</b>	<b>Peso</b>	<b>Valor asignado</b>	<b>Total</b>
T1	Sistema distribuido	2	1	2
T2	Tiempo de respuesta	1	5	5
T3	Eficiencia del usuario final	1	5	5
T4	Funcionamiento Interno complejo	1	4	4
T5	El código debe ser reutilizable	1	4	4
T6	Facilidad de instalación	0,5	3	1.5
T7	Facilidad de uso	0,5	4	2
T8	Portabilidad	2	3	6
T9	Facilidad de cambio	1	3	3
T10	Concurrencia	1	5	5
T11	Incluye objetivos especiales de seguridad	1	4	4
T12	Provee acceso directo a terceras partes	1	0	0
T13	Se requieren facilidades especiales de entrenamiento de usuarios	1	3	3

$$TCF = 0.6 + 0.01 * \sum ( peso * valor \text{ asignado } )$$

$$TCF = 0.6 + 0.01 * (2 + 5 + 5 + 4 + 4 + 1.5 + 2 + 6 + 3 + 5 + 4 + 0 + 3)$$

$$TCF = 0.6 + 0.01 * (43)$$

$$TCF = 1.03$$

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).

**Tabla 32. Factor de ambiente.**

Factor	Descripción	Peso	Valor asignado	Total
E1	Familiaridad con el modelo de proyecto utilizado	1,5	4	6
E2	Experiencia en la aplicación	0,5	4	2
E3	Experiencia en la orientación a objetivos.	1	4	4
E4	Capacidad del analista líder.	0,5	5	2.5
E5	Motivación.	1	5	5
E6	Estabilidad de requerimientos	2	5	10
E7	Personal Part-Time	-1	3	-3
E8	Dificultad del lenguaje de programación	-1	3	-3
Total				23.5

$$EF = 1.4 - 0.03 * \sum ( peso * valor asignado )$$

$$EF = 1.4 - 0.03 * (6 + 2 + 4 + 2.5 + 5 + 10 - 3 - 3)$$

$$EF = 1.4 - 0.03 * 23.5$$

$$EF = 1.4 - 0.705$$

$$EF = 0.695$$

$$UCP = UUCP * TCF * EF$$

$$UCP = 150 * 1.03 * 0.695$$

$$UCP = 107,3775$$

### 5.2.2 Estimación de esfuerzo a través de los puntos de casos de uso.

$$E = UCP * CF$$

Donde:

*E* : Esfuerzo Estimado en Horas – Hombres

*UCP* : Puntos de Casos de Uso Ajustados

*CF* : Factor de Conversión

$$E1—E6= 2 \quad E7—E8=0$$

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.

En este caso se puede decir que:

$CF = 20 \text{ Horas-Hombre} / \text{Punto de Casos de uso.}$

$$E = UCP * CF$$

$$E = 107,3775 * 20$$

$$E = 2147,55 \text{ Horas - Hombre}$$

#### **Paso 4. Calcular esfuerzo de todo el proyecto.**

**Tabla 33. Esfuerzo del proyecto.**

Actividad	Porcentaje	Horas-Hombres
Análisis	10 %	536,8875
Diseño	20 %	1073,775
Implementación	40 %	2147,55
Pruebas	15 %	805,33125
Sobrecarga (otras actividades)	15 %	805,33125
Total	100 %	5368,875

Si el esfuerzo total es de 5368,875 **horas-hombre** y por cada 192 horas yo tengo 1 mes eso daría un  $E_T = 27,9628906$  **mes-hombre**.

$$E = 5368,875 \text{ Horas - Hombres}$$

$$\text{Cantidad de Horas - Hombre por mes (CH)} = 192$$

$$E = 5368,875 / 192 \text{ Meses - Hombres}$$

$$E = 27,96 \text{ Meses - Hombres}$$

Esto quiere decir que 1 persona puede realizar el sistema analizado en aproximadamente 2 años y 4 meses.

#### **Costo del Proyecto**

Se asume como salario básico mensual (SBM) es de \$225.00

$$CHM = 2 * SBM$$

$$CHM = 450.00 \text{ pesos/mes}$$

$$\text{Costo} = CHM * E_T$$

$$\text{Costo} = 450 * 27,96$$

**Costo = \$ 6291, 65**

### **5.3 Beneficios tangibles e intangibles**

El sistema de Gestión de Servicios Comunitarios no es un producto con fines comerciales aunque puede adjuntarse por sus características a algún sistema de gestión de servicios, por lo que los beneficios que aporta serán principalmente intangibles. Tiene como principal objetivo resolver uno de los grandes problemas que tiene la universidad en estos momentos que es los servicios comunitarios de la UCI.

El principal objetivo de la aplicación es mejorar la gestión de los servicios comunitarios en la UCI.

Beneficios:

- Tener un sistema global para la realización de solicitudes desde cualquier ubicación en la universidad.
- Que los solicitantes puedan ver el estado de las solicitudes en cualquier momento.
- Que los jefes de áreas y supervisores conozcan el estado de las solicitudes realizadas en su área de servicio.
- Que los directivos puedan ver el sistema de reportes, el cual los ayudará en la toma de decisiones.

### **5.4 Análisis de costos y beneficios**

El desarrollo de este sistema no supone grandes gastos de recursos, aunque si de tiempo; la base de datos que contiene la información, puede ser alojada en los servidores existentes en la universidad, ya que los mismos tienen buenas prestaciones y acceso rápido. La tecnología utilizada para el desarrollo del sistema es .NET, que aunque no es una tecnología gratis a nivel mundial, puede ser de fácil migración al proyecto Mono y a Java, 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 que se diseñaron.

## **5.5 Conclusiones**

En este capítulo se efectuó el estudio de factibilidad correspondiente al desarrollo del proyecto. Este permitió llegar a la conclusión que resultará factible implementar la aplicación. La herramienta propuesta trae consigo una serie de beneficios sobre todo intangibles para el centro, pero no menos necesarios e importantes, porque va a contribuir a mejorar su funcionamiento, lo que indica que es factible implementar la herramienta propuesta.

## Conclusiones

Con el presente trabajo se propone una solución integral al problema de la Gestión de los Servicios Comunitarios en la Universidad de las Ciencias Informáticas. Se presenta una aplicación capaz de controlar las solicitudes del personal de la UCI, brindando además reportes sobre las mismas. El buen uso de esta aplicación, puede convertirla en una poderosa herramienta para el control de las solicitudes en la universidad, permitiendo ser usada además en la toma de decisiones.

El sistema se desarrolló siguiendo la metodología RUP, y se utilizaron representaciones para la modelación de todas las fases del proyecto.

El sistema resultante tiene un ambiente fácil de entender y usar, el cual cumple con los estándares de diseño y utiliza técnicas modernas de la programación orientada a objetos.

Por todo lo anterior expuesto se concluye que los objetivos propuestos para el presente trabajo han sido cumplidos satisfactoriamente. Se incluyen además una serie de recomendaciones que deben tenerse en cuenta para el trabajo futuro.



## Recomendaciones

A pesar de haberse cumplido los objetivos generales del trabajo, nuevas ideas han ido surgiendo en el desarrollo del mismo, permitiendo esto que en un futuro se cree un sistema más eficiente, por tanto recomendamos:

1. Continuar con el desarrollo del sistema con el objetivo de adecuarlo mejor a las demandas que presenta la universidad.
2. Emigrar el sistema a software libre, con el objetivo de seguir el movimiento que actualmente se esta llevando a cabo en la universidad. El sistema fue diseñado de manera que al emigrarlo, los cambios serán mínimos.
3. Integrar el sistema con otras aplicaciones que requieran obtener la información almacenada por el mismo.
4. Aplicar el sistema a todas las áreas de la universidad donde pueda ser usado, permitiendo un mejor uso del mismo.
5. Extenderlo de forma que pueda ser utilizado no solo en la universidad, sino también en cualquier lugar o empresa que requiera de un sistema para la Gestión de los Servicios Comunitarios.

## Referencias bibliográficas

1. Fidalgo Hidalgo, José y Tamayo Agramonte, Yudiel A. Sistema de Servicio Comunitario. Trabajo para optar por el título de Ingeniero Informático. Instituto Superior Politécnico “José Antonio Echeverría”, Universidad de La Habana, Ciudad de la Habana, Junio del 2004.
2. Booch, G., Rumbaugh, J., Jacobson, I. “El Lenguaje Unificado de Modelado”. Addison-Wesley. 1999.
3. Shohoud, Yasser. Real World XML Web Services. Addison-Wesley. 2003.
4. Ferguson, Jeff, Patterson, Brian, Beres, Jason, Boutquin, Pierre, Gupta, Meeta. La Biblia del C#. Anaya. 2003.
5. Microsoft. Introduction to C# Programming for the Microsoft .Net Platform. Microsoft. 2001.
6. Joseph, Schmuller. “Aprendiendo UML en 24 horas”, Prentice-Hall, Inc. 2001.
7. *¿Qué es ASP.NET?*  
<http://www.zonagratis.com/microsoft/asp/aspnet.htm> (10/4/2006)
8. Lenguajes de Programación: “Programación Web”. <http://lenguajes-de-programacion.com/programacion-web.shtml> (16/4/2006)
9. Compare Microsoft .NET to J2EE Technology.  
<http://www.gotdotnet.com/team/compare/> (16/4/2006)
10. Características de Visual Studio .NET. Microsoft Corp. 2005.  
<http://www.microsoft.com/latam/vstudio/producto/caracteristicas.asp> (18/5/2006)
11. .NET Framework Fundamentals.  
<http://msdn.microsoft.com/netframework/programming/fundamentals/default.aspx>  
(16/5/2006)
12. URL Rewriting in ASP.NET.  
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnasp/html/urlrewriting.asp> (20/5/2006)
13. Introduction to Web User Controls.  
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vbcon/html/vbconintroductiontocustomwebcontrols.asp> (20/5/2006)

14. Tutorial: Introduction to Web Services. <http://www.c-sharpcorner.com/Tutorials/IntroductionToWebServicesT.asp> (10/5/2006)
15. Database Programming in C# with ADO.NET. <http://www.c-sharpcorner.com/Database.asp> (18/5/2006)
16. Core J2EE Patterns.  
<http://java.sun.com/blueprints/corej2eepatterns/Patterns/index.html> (15/4/2006)
17. Análisis y Diseño Orientado a Objetos.  
<http://www.dcc.uchile.cl/~luguerre/cc40b/index.html> (5/5/2006)
18. Cursos.  
<http://www.microsoft.com/spanish/msdn/comunidad/uni.net/> (25/4/2006)
19. STECNICO versión 1.4.  
[http://personal.lobocom.es/jlpl/Stec\\_jlp.html](http://personal.lobocom.es/jlpl/Stec_jlp.html) (20/5/2006)
20. C Sharp  
[http://es.wikipedia.org/wiki/C\\_Sharp](http://es.wikipedia.org/wiki/C_Sharp) (10/4/2006)
21. NHibernate  
<http://www.theserverside.net/tt/articles/showarticle.tss?id=NHibernate> (2/5/2006)

## Bibliografía

1. Cerami, Ethan. Web Services Essentials. O'Reilly. 2002.
2. Fowler, Martin, Beck, Kent, Brant, John, Opdyke, William, Roberts, Don. Refactoring: Improving Design of Existing Code. Addison-Wesley. 2002.
3. Kerievsky, Joshua. "Refactoring to patterns". Industrial Login Inc. 2002.
4. Cockburn, Alistair. Writing Effective Use Cases. Addison-Wesley. 2000.
5. North, Simon. Teach yourself XML in 21 days. Macmillan Computer Publishing. 1999.
6. Benage, Don, Socha, Jody. .Net e-Business Architecture. Sams. 2002.
7. Mitchell, Scot, Anders, Bill, Howard, Rob, Seven Doug, Walther, Stephen, Willie, Christop, Wolthuis, Don. ASP. NET: Tips, Tutorials and Code. Sams. 2001.
8. Lazy Collections  
<http://forum.hibernate.org/viewforum.php?f=25> (20/5/2006)
9. Qué es c#  
<http://www.desarrolloweb.com/articulos/561.php> (25/4/2006)
10. Tutorial UML  
[www.dcc.uchile.cl/~psalinas/uml/introduccion.html](http://www.dcc.uchile.cl/~psalinas/uml/introduccion.html) (10/5/2006)

## Glosario de términos

**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).

**CLR:** (Common Language Runtime) Núcleo de la plataforma *.Net*, motor que gestiona la ejecución de las aplicaciones para ella desarrolladas y a las que ofrece múltiples servicios.

**CSS:** Siglas en Inglés de Cascading Style Sheets (Hojas de Estilo en Cascada).

**DLL:** (Dynamic Linking Library) Bibliotecas de enlace dinámico), término con el que se refiere a los archivos con código ejecutable que se cargan bajo demanda del programa por parte del sistema operativo. Esta denominación se refiere a los sistemas operativos Windows siendo la extensión con la que se identifican los ficheros, aunque el concepto existe en prácticamente todos los sistemas operativos modernos.

**FCL :** (Framework Class Library) Librería incluida en el *.NET Framework*, formada por cientos de tipos de datos que permiten acceder a los servicios ofrecidos por el CLR y a las funcionalidades más frecuentemente usadas a la hora de escribir programas.

**HQL:** (Hibernate Query Language) Lenguaje de consultas hibernate. Es un lenguaje de interrogación. En el mundo relacional disponemos del SQL (Structured Query Language) que nos permite obtener información haciendo preguntas basadas en las tablas y sus columnas. El equivalente en el mundo objetual es el HQL, que nos permite hacer preguntas basadas en los objetos y sus propiedades.

**IDE:** (Integrated development environment). No es más que un entorno integrado de desarrollo.

**Microsoft:** Compañía de software más grande del mundo. Fue fundada en 1975 por Paul Allen y Bill Gates. Aunque también se conoce por sus lenguajes de programación y aplicaciones para computadores personales, el éxito sobresaliente de Microsoft se debe a sus sistemas operativos DOS y Windows.

**MSIL:** (Microsoft Intermediate Language). Lenguaje intermedio de Microsoft. Es el lenguaje de programación que emiten, de forma interna, los compiladores de lenguajes *.NET*, como VB.NET y C#.

**NHibernate:** Es un mapeador objeto-relacional que proporciona un puente entre la programación orientada a objetos y los sistemas de gestión de bases de datos relacionales, basado en el famoso framework Hibernate.

**OMT:** (Object Modeling Techniques). Metodología de análisis y diseño orientado a objetos, desarrollada por Jim Rumbaugh; es una de las tres grandes metodologías que contribuyeron al desarrollo del RUP.

**OOSE:** (Oriented Software Engineering) . Metodología de análisis y diseño orientado a objetos, desarrollada por Ivar Jacobson; es una de las tres grandes metodologías que contribuyeron al desarrollo del RUP.

**RAD:** (Rapid Application Development) Desarrollo rápido de aplicaciones. Proceso de desarrollo de software desarrollado inicialmente por James Martin en 1980. El método comprende el desarrollo iterativo, la construcción de prototipos y el uso de utilidades CASE. Tradicionalmente, el desarrollo rápido de aplicaciones tiende a englobar también la usabilidad, utilidad y la rapidez de ejecución.

**SAT :** Servicios de Asistencia Técnica.

**SDK:** (Software Development Kit). Kit de desarrollo de software, un conjunto de aplicaciones para desarrollar programas en un determinado lenguaje o para un determinado entorno.

**SOAP:** Acrónimo de Simple Object Access Protocol (Protocolo de acceso de objeto simple). Es un protocolo elaborado para facilitar la llamada remota de funciones a través de Internet, permitiendo que dos programas se comuniquen de una manera muy similar técnicamente a la invocación de páginas Web.

**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.

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

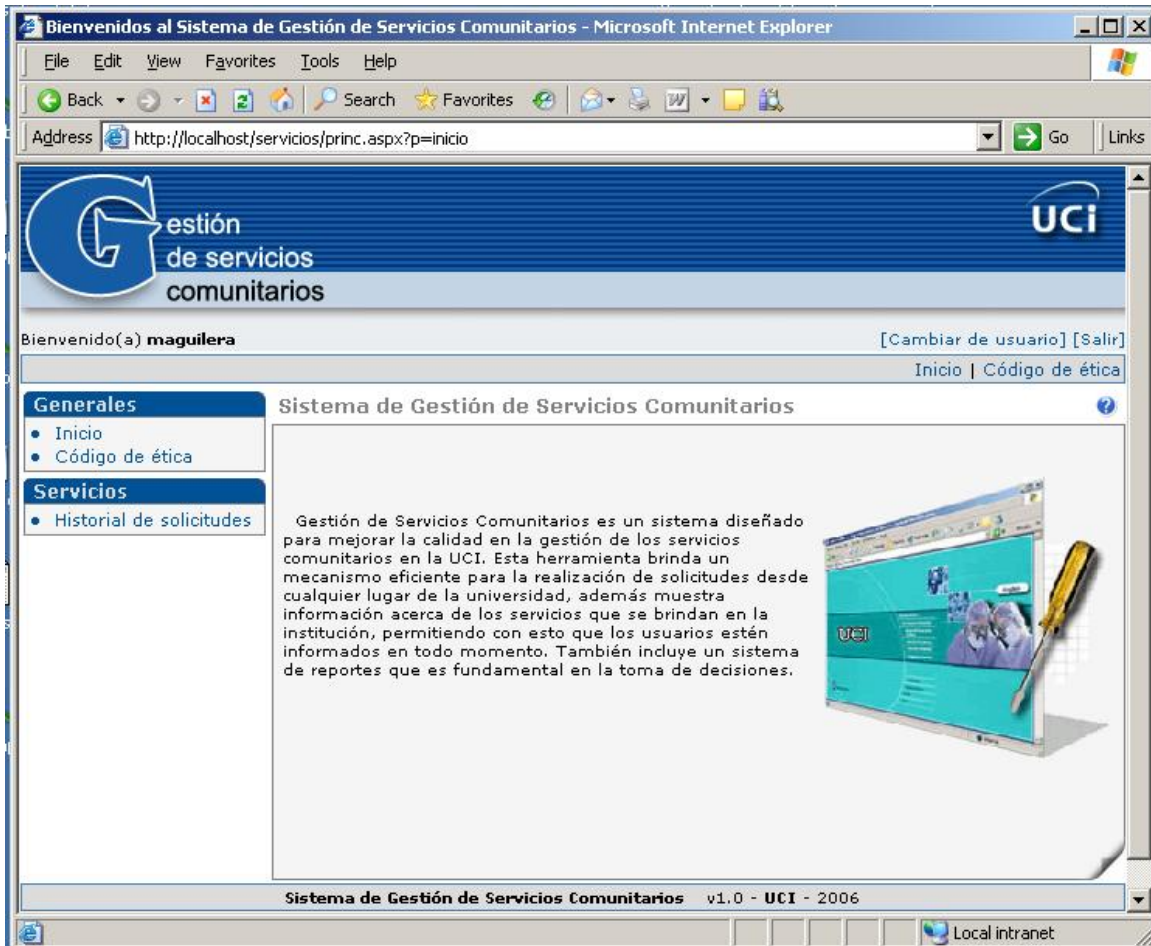
**URL:** Acrónimo de Universal Resource Locator (localizador universal de recursos), método de identificación de documentos o lugares en Internet, que se utiliza principalmente en World Wide Web (WWW). Un URL es una cadena de caracteres que

identifica el tipo de documento, la computadora, el directorio y los subdirectorios en donde se encuentra el documento y su nombre.

**Web Services:** Aplicación que realiza un cometido y que puede formar parte de otros servicios para formar un servicio más completo. La comunicación hacia y desde el WebService se realiza con XML.

**XML:** Es el acrónimo de eXtensible Markup Language (lenguaje de marcado ampliable o extensible) desarrollado por el World Wide Web Consortium (W3C).

## Anexo I



*Página de inicio.*



## Anexo II

NuevaSolicitud - Microsoft Internet Explorer

Nueva solicitud

Salvar Solicitud Cancelar

**Datos del solicitante**  
[Usuario actual][Otro usuario]

**E** ycabrerago@uci.cu  
Yobannys Cabrera Gonzalez

**Local donde se presenta el problema**  
Administración / Alimento / Almacén

**Problema presentado**  
Problemas con el Grupo electrógeno del Edificio.

**Seleccione los medios que pudieran estar relacionados**  
Grupos electrógenos y motores

**Descripción General**

Nota: En la descripción de la solicitud puede poner el horario en que desee que se le atienda, así como cualquier otro dato que crea se deba saber.

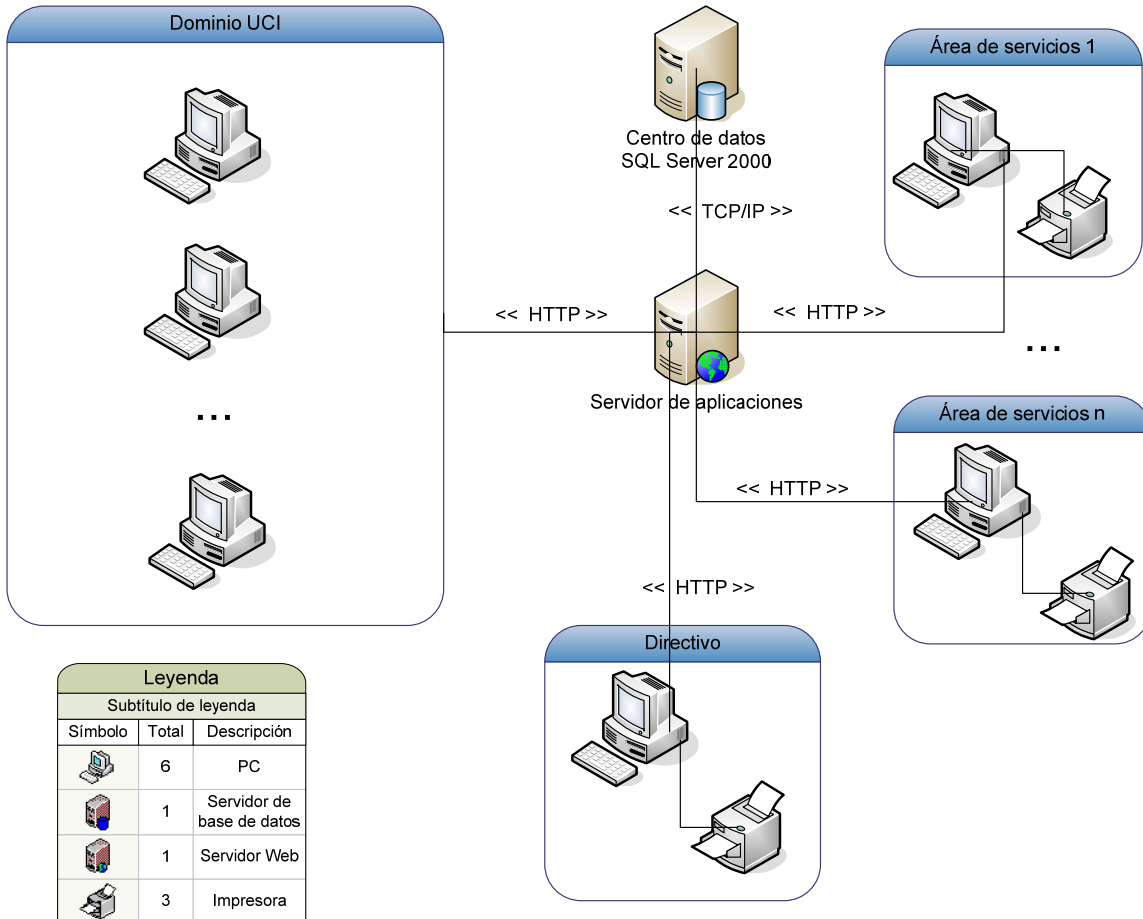
*Página nueva solicitud.*

## Anexo III

<b>Generales</b>
<ul style="list-style-type: none"><li>• Inicio</li><li>• Profile</li><li>• Código de ética</li></ul>
<b>Navegación</b>
<ul style="list-style-type: none"><li>• Accesos</li><li>• Módulos</li><li>• Links</li></ul>
<b>Administración</b>
<ul style="list-style-type: none"><li>• Usuarios</li><li>• Grupos</li><li>• Recursos</li></ul>
<b>Sistema</b>
<ul style="list-style-type: none"><li>• Locales</li><li>• Áreas de servicio</li><li>• Estados</li><li>• Materiales</li><li>• Unidades de medida</li></ul>
<b>Area de servicio</b>
<ul style="list-style-type: none"><li>• Solicitudes del área</li><li>• Ordenes de trabajo</li><li>• Certificaciones de trabajo</li><li>• Servicios</li><li>• Supervisores</li><li>• Técnicos</li></ul>
<b>Servicios</b>
<ul style="list-style-type: none"><li>• Historial de solicitudes</li></ul>
<b>Reportes</b>
<ul style="list-style-type: none"><li>• Reporte General</li></ul>

*Menú lateral con todos lo accesos de la aplicación.*

# Anexo IV



Leyenda		
Subtítulo de leyenda		
Símbolo	Total	Descripción
	6	PC
	1	Servidor de base de datos
	1	Servidor Web
	3	Impresora

*Modelo de despliegue detallado.*