

# Universidad de las Ciencias Informáticas

## Facultad 1



**Título:** Análisis y diseño del Módulo Administración del Becado

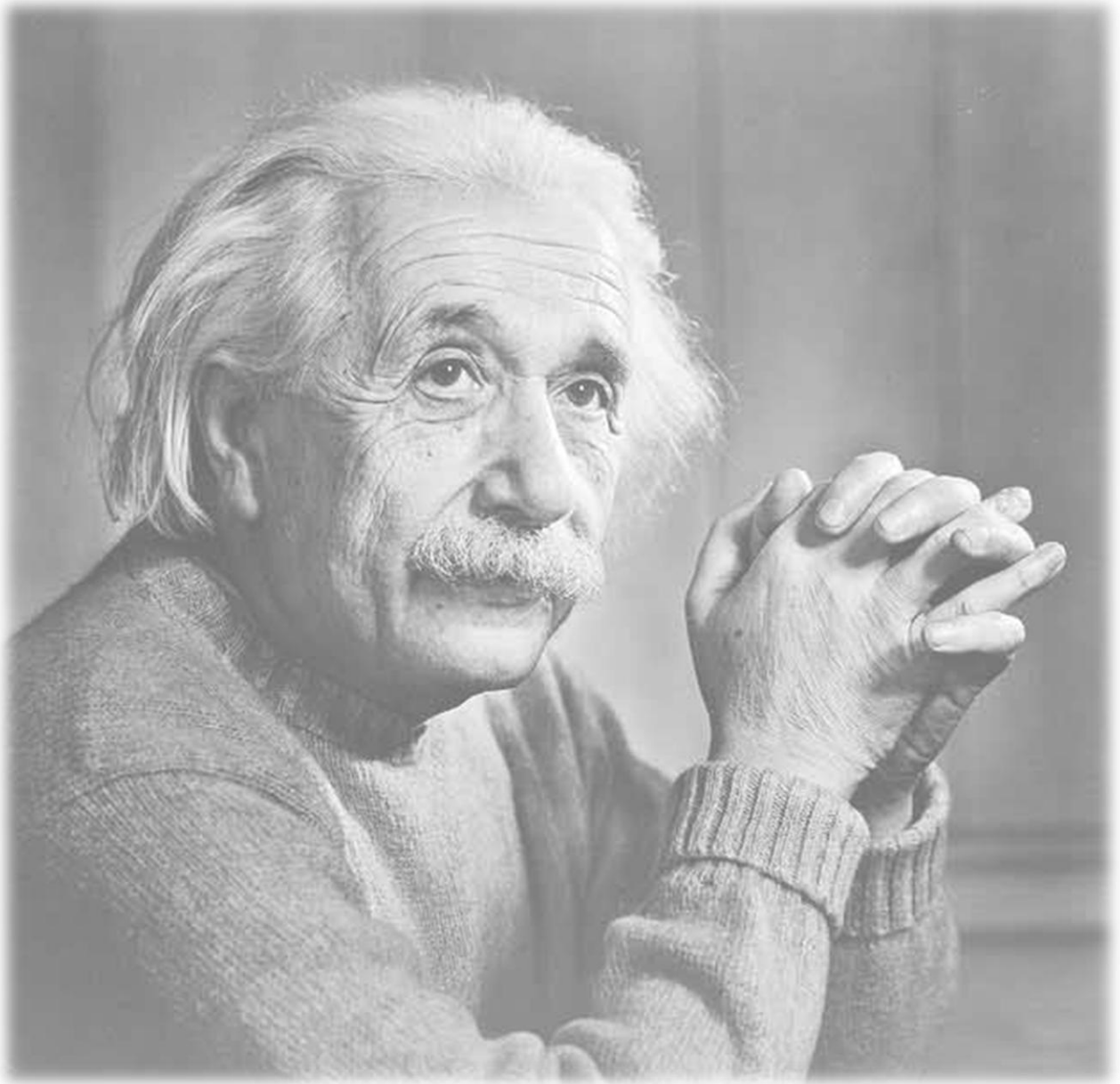
**Trabajo de Diploma para optar por el título de  
Ingeniero en Ciencias Informáticas**

**Autor(es): Wendy Rafael Bokly Plana**

**Tutor(es): Ing. Yuniel Cedeño Mendoza**

**Ciudad de la Habana**

**Junio 2008**



*"Nunca consideres el estudio como una obligación sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber."*

*Albert Einstein*

## DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al Facultad # 1 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Wendy Rafael Bokly Plana

Ing. Yuniel Cedeño Mendoza

---

---

Firma del autor

Firma del tutor

## DATOS DE CONTACTO

Ing. Yuniel Cedeño Mendoza Graduado de Ingeniería Informática en el 2007, profesor de Máquinas Computadoras en la facultad # 1. Esta es su primera experiencia como tutor.

# Agradecimientos

---

## AGRADECIMIENTOS

*A mi tío Bimbe y mi tía Mita por darme todo su amor, por ser los principales guías y educadores de mi vida, Agradecerles siempre su apoyo incondicional, por acogerme y tenerme siempre como su hijo, por eso todo mi amor y respeto hacia ustedes.*

*A mi mamá por darme la vida, educarme, siempre estar pendiente de mí, apoyarme en todo momento y brindarme todo su amor, por eso todo mi amor y respeto hacia ti mami.*

*A mi “niña”, por apoyarme en las buenas y las malas, por ser el completamiento de mi vida, por brindarme todo su amor y estar pendiente de mis cosas, por ser la mujer, compañera, amante, y la amiga ideal, agradecerle por haberme hecho un hombre mejor, por todo esto: “Ingrid Te Amo Mucho!!!”*

*A mi suegra, por tenerme como un hijo, por ser la mejor suegra del mundo, por ser parte de mi apoyo en estos momentos que no tengo a mi familia cerca, por todo esto Tere, mis afectos y respetos hacia usted*

*A toda mi familia que de una manera u otra han colaborado para que este sueño se haga realidad, principalmente a: (mi herma, tia Rosa, a mi abuela Natalia, a mi padrastro Titi, a mi primo Lisán, a mi Javá por existir). Agradecele a la familia de Ingrid que ahora es mi familia también: (Marta, Marta Rosa, Orlando, Julia, Ediana, Mirian e Indira), por sentirme como en mi casa.*

*A mis amigos, amigas y compañeros que a pesar de nuestras diferencias nos convertimos en una pequeña familia donde compartimos nuestras alegrías y tristezas.*

*A mi tutor por hacer su trabajo todo lo mejor que pudo, por estar siempre presente en todos los momentos.*

*A los profesores que sin ellos esto no sería posible.*

*A la Revolución y a Fidel por darme esta oportunidad.*

*Y a la vida que sin ella no hubiese podido escribir estos agradecimientos. Gracias a la Vida que me ha dado tanto!!!*

## Dedicatoria

---

*A mi mamá por todo su apoyo*

*A mi tía Mita y mi tío Bimbe por ser mis guías*

*A mi novia linda por todo su amor, consagración*

*hacia mí*

*A mi herma Darlis para que siga estudiando y*

*superándose*

*A Teresa por todo el apoyo que me ha brindado*

*todos estos años.*

*A mi abuela Luisa que seguro me hubiese querido*

*ver en este momento*

*Y a todas las personas que de alguna forma son*

*parte de mi vida*

## *RESUMEN*

En la residencia estudiantil se controlan un conjunto de actividades donde el estudiante debe participar, además se controlan los datos personales de cada becado. El control de estas actividades y de los datos personales de los becados. Actualmente este control se hace de forma manual, lo que trae como consecuencia desorganización y demora en la actualización de toda la información.

Como parte de las informatizaciones que está realizando la facultad en todas sus áreas, se crea el Módulo de Administración del Becado, con el objetivo de automatizar el control de las actividades y administrar los datos del becado en la residencia.

Este trabajo realiza el análisis y diseño de un servicio web que brinda un conjunto de funcionalidades que permitirá llevar a cabo el control de las actividades y administración de los datos del becado. Los servicios web son aplicaciones, están basados en una arquitectura orientada a servicios, que brindan servicios a aplicaciones multiplataformas mediante la localización y coordinación de los servicios disponibles en la red.

## ÍNDICE

<i>AGRADECIMIENTOS</i> .....	IV
<i>RESUMEN</i> .....	VI
INTRODUCCIÓN .....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA .....	6
INTRODUCCIÓN .....	6
1.1 CONTROL DEL BECADO .....	6
1.2 SISTEMAS AUTOMATIZADOS EXISTENTES .....	7
1.2.1 Sistemas existentes en el Extranjero .....	7
1.2.2 Sistemas existentes en Cuba.....	9
1.2.3 Sistemas existentes en la UCI .....	10
1.3 TENDENCIAS Y DESARROLLO DEL SOFTWARE LIBRE .....	12
1.3.1 Comportamiento en Cuba y en la Universidad de las Ciencias Informáticas (UCI) .....	12
1.4 TECNOLOGÍAS ACTUALES .....	14
1.4.1 ¿Qué es un servicio web?.....	14
1.4.2 PROTOCOLO SOAP.....	15
1.4.3 ¿QUÉ ES NUSOAP? .....	16
1.4.4 ¿QUÉ ES WSDL? .....	16
1.4.5 UDDI.....	16
1.4.6 ARQUITECTURA ORIENTADA A SERVICIO.....	17
1.4.7 ¿POR QUÉ UTILIZAR SERVICIOS WEB?.....	18
1.4.8 LENGUAJES DE PROGRAMACIÓN PARA WEB .....	19
1.4.9 LANGUAGES DE PROGRAMACIÓN A UTILIZAR.....	19
1.4.10 ZEND STUDIO COMO IDE A UTILIZAR .....	22
1.4.11 SISTEMA GESTORES DE BASE DATOS .....	23
1.4.12 MYSQL COMO GESTOR DE BASE DATOS .....	23
1.4.14 METODOLOGÍAS PARA EL DESARROLLO DE SOFTWARE. RUP .....	24
1.4.15 UML COMO LENGUAJE DE MODELADO .....	26
CONCLUSIÓN .....	27
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA .....	29
INTRODUCCIÓN .....	29
2.1 ESTADO ACTUAL DEL NEGOCIO.....	29
2.2 PROPUESTA DE AUTOMATIZACIÓN .....	30
2.3 REGLAS DEL NEGOCIO.....	30
2.4 MODELO DE NEGOCIO.....	31



2.5 ACTOR DEL NEGOCIO.....	31
2.6 TRABAJADORES DEL NEGOCIO .....	32
2.7 MODELO DE CASOS DE USOS DEL NEGOCIO .....	32
2.8 DESCRIPCIÓN TEXTUAL DE LOS CASOS DE USO .....	33
2.9 DIAGRAMA DE CLASES DEL MODELO DE OBJETO .....	45
2.10 DIAGRAMA DE ACTIVIDADES.....	46
2.11 REQUERIMIENTOS FUNCIONALES.....	47
2.12 REQUERIMIENTOS NO FUNCIONALES .....	50
2.13 DEFINICIÓN DE LOS CASOS DE USOS DEL SISTEMA.....	52
2.14 CASOS DE USOS DEL SISTEMA .....	52
2.15 DESCRIPCIONES DE LOS CASOS DE USO DEL SISTEMA .....	53
2.16 CASOS DE USOS DEL SISTEMA EXPANDIDOS .....	58
CONCLUSIONES.....	58
<b>CAPÍTULO 3: ANÁLISIS Y DISEÑO .....</b>	<b>59</b>
INTRODUCCIÓN .....	59
3.1 MODELO DE ANÁLISIS .....	59
3.2 DIAGRAMA DE CLASES DEL ANÁLISIS.....	60
3.2 MODELO DE DISEÑO.....	63
3.3 DIAGRAMAS DE INTERACCIÓN.....	64
3.3 DIAGRAMAS DE CLASES DEL DISEÑO.....	64
3.4 DESCRIPCIÓN DE LAS CLASES .....	66
3.5 DISEÑO DE LA BASE DATOS.....	81
3.6 DESCRIPCIÓN DE LAS TABLAS DE LA BASE DATOS .....	81
3.7 ESTUDIO DE LA FACTIBILIDAD.....	86
CONCLUSIÓN .....	93
<b>CONCLUSIONES GENERALES.....</b>	<b>94</b>
<b>RECOMENDACIONES .....</b>	<b>95</b>
<b>BIBLIOGRAFÍA REFERENCIADA.....</b>	<b>96</b>
<b>BIBLIOGRAFÍA .....</b>	<b>97</b>
<b>ANEXO 1 .....</b>	<b>99</b>
<b>ANEXO 2 .....</b>	<b>105</b>
<b>ANEXO 3 .....</b>	<b>113</b>
<b>GLOSARIO DE TÉRMINOS .....</b>	<b>120</b>

## ÍNDICE DE TABLA

TABLA 1 CRONOGRAMA DE INVESTIGACIÓN .....	4
TABLA 2 ACTORES DEL NEGOCIO.....	31
TABLA 3 TRABAJADORES DEL NEGOCIO.....	32
TABLA 4 DIAGRAMA DE CASO DE USO. “CONTROLAR CUARTELERÍA” .....	34
TABLA 5 CASO DE USO “CONTROLAR GUARDIA ESTUDIANTIL” .....	36
TABLA 6 CASO DE USO “CONTROLAR TSU“ .....	38
TABLA 7 CASO DE USO “REGISTRAR LA EVALUACIÓN DEL MES” .....	39
TABLA 8 CASO DE USO “INSPECCIONAR LA LIMPIEZA DEL APTO” .....	42
TABLA 9 CASO DE USO “REINSPECCIONAR LA LIMPIEZA” .....	43
TABLA 10 CUS GESTIONAR DATOS DEL ESTUDIANTE.....	53
TABLA 11 CUS CONTROLAR CUARTELERÍA .....	54
TABLA 12 CUS CONTROLAR GUARDIA .....	54
TABLA 13 CUS CONTROLAR TSU .....	55
TABLA 14 CUS CONTROLAR LIMPIEZA DEL APTO.....	56
TABLA 15 CUS GENERAR REPORTE DE ESTUDIANTE .....	56
TABLA 16 CUS REGISTRAR EVALUACIÓN MENSUAL .....	57
TABLA 17 CE_ESTUDIANTE .....	66
TABLA 18 CE_INCIDENCIA .....	67
TABLA 19 CE_EVALUACIÓN_MENSUAL.....	68
TABLA 20 CE_EVALUACIÓN_CUARTELERÍA .....	68
TABLA 21 CE_TSU .....	68
TABLA 22 CE_GUARDIA.....	69
TABLA 23 CE_ASISTENCIA_GUARDIA .....	69
TABLA 24 CE_ASISTENCIA_TSU .....	70
TABLA 25 CC_ESTUDIANTE .....	70
TABLA 26 CC_CONTROLAR_CUARTELERÍA .....	71
TABLA 27 CC_CONTROLAR_TSU .....	72
TABLA 28 CC_CONTROLAR_LIMPIEZA.....	73
TABLA 29 CC_CONTROLAR_GUARDIA .....	74
TABLA 30 ADO_ESTUDIANTE.....	75
TABLA 31 ADO_TSU.....	76
TABLA 32 ADO_CONTROLAR_GUARDIA.....	77
TABLA 33 ADO_CONTROLAR_LIMPIEZA.....	77
TABLA 34 ADO_CONTROLAR_CUARTELERÍA .....	78
TABLA 35 CI_ADMINISTRACION_DEL_BECADO.....	79
TABLA 36 Tb_ESTUDIANTE .....	81
TABLA 37 Tb_INCIDENCIA .....	82
TABLA 38 Tb_EVALUACION_MENSUAL.....	82
TABLA 39 Tb_TSU .....	83

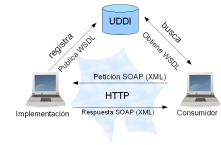
## Índice de Tabla

---

TABLA 40 Tb_GUARDIA .....	83
TABLA 41 Tb_EVALUACIÓN_CUARTELERO.....	83
TABLA 42 Tb_ASISTENCIA_TSU .....	84
TABLA 43 Tb_TIPO_MILITANCIA .....	84
TABLA 44 Tb_TIPO_DIRIGENTE .....	84
TABLA 45 Tb_TIPO_EVALUACIÓN.....	85
TABLA 46 Tb_ASISTENCIA_GUARDIA .....	85
TABLA 47 Tb_ESTUDIANTE_Tb_GUARDIA.....	85
TABLA 48 Tb_ESTUDIANTE_Tb_TSU.....	85
TABLA 49 FACTOR DE PESO DE LOS ACTORES SIN AJUSTAR.....	86
TABLA 50 FACTOR DE PESO DE LOS CASOS DE USO SIN AJUSTAR. ....	87
TABLA 51 FACTOR DE COMPLEJIDAD TÉCNICA .....	88
TABLA 52 FACTOR DE AMBIENTE.....	89
TABLA 53 ESFUERZO DEL PROYECTO. ....	91

## ÍNDICE DE FIGURA

FIGURA 1 FUNCIONAMIENTO DE LOS SERVICIOS WEB.....	15
FIGURA 2 ARQUITECTURA TRADICIONAL DE SOA .....	18
FIGURA 3 DIAGRAMA DE CASOS DE USO DEL NEGOCIO.....	33
FIGURA 4 MODELO DE OBJETO .....	45
FIGURA 5 MODELO DE OBJETO [CONTINUACIÓN].....	46
FIGURA 6 DIAGRAMA DE CASOS DE USO DEL SISTEMA.....	52
FIGURA 7 DCA CONTROLAR CUARTELERÍA .....	60
FIGURA 8 DCA CONTROLAR GUARDIA.....	61
FIGURA 9 DCA CONTROL TSU.....	62
FIGURA 10 DCA GESTIONAR DATOS .....	62
FIGURA 11 DCA GENERAR REPORTE DE ESTUDIANTE.....	63
FIGURA 12 DCA REGISTRAR EVALUACIÓN MENSUAL.....	63
FIGURA 13 DIAGRAMA DE CLASES EN PAQUETES .....	65



# Introducción

---

## INTRODUCCIÓN

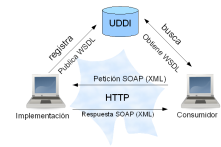
El desarrollo de la informática es actualmente una necesidad para el desarrollo de la humanidad; esta contribuye a hacer la vida más simple y más amena, aumenta la fiabilidad y efectividad de los bienes y servicios que se brindan en la actualidad. Por la vigencia e importancia, cada día hay más empresas involucradas en el desarrollo de software, para facilitar el intercambio de información y contribuir al desarrollo de la tecnología, ampliando el conocimiento humano.

Nuestro país está construyendo cada día una sociedad más eficiente y eficaz, introduciendo en todas las esferas y procesos la informática. Cuba ha puesto en mano de toda sociedad las Tecnologías de la Información y las Comunicaciones (TIC), y así lograr una cultura general en este ámbito, lo que facilitará que la sociedad se acerque cada vez más hacia el objetivo de un desarrollo sostenible.

La Industria Cubana del Software está llamada a convertirse en una fuente importante de ingresos de la nación, como resultado del aprovechamiento de las ventajas del alto capital humano existente y disponible en la nación. La Universidad de la Ciencias Informática (UCI) está vinculada a los procesos de docencia-investigación-producción en esta esfera, y jugará un papel fundamental en la informatización del país.

La UCI ha empezado a automatizar todos sus procesos en general, para esto se han creado diferentes grupos de desarrollo para los proyectos existentes. Esta estrategia ha logrado que el personal este más capacitado.

La Residencia Estudiantil de la (UCI) constituye un medio educativo muy rico, por lo tanto, en ella se realiza un conjunto de actividades con los estudiantes, que requieren del control de estas, así como la administración de los datos personales. Este control se hace forma manual, lo que lleva tiempo en organizar y mantener actualizada dicha información. Estas actividades son esenciales para determinar la evaluación final del estudiante.



## Introducción

---

Se hace necesario mejorar la eficiencia y la rapidez para obtener las informaciones de los estudiantes, que son necesarias para determinar sus evaluaciones en la residencia. Para esto se crea el Módulo de Administración del Becado de la Intranet de la facultad 1, su objetivo es controlar las actividades y administrar los datos del estudiante en la residencia estudiantil.

Con el análisis de la situación actual se busca realizar un trabajo que cumpla con los requerimientos que los trabajadores necesitan para mejorar su calidad en el trabajo.

La situación problemática, es que actualmente el control de las actividades y los datos de los estudiantes se hace de forma manual, lo que lleva tiempo en organizar y mantener actualizado dicha información. Además no se tiene un estricto control de las actividades que se realizan en la residencia, y no se asegura que todos los datos de los estudiantes se puedan conservar mientras los estudiantes sean alumnos de la universidad.

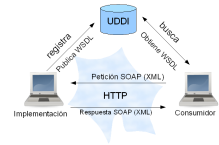
Teniendo esta situación existente se plantea el siguiente **problema**:

¿Cómo desarrollar el análisis y diseño del Módulo de Administración del Becado, para llevar el control de las actividades y la administración de los datos de los becados en la residencia estudiantil, de manera rápida y eficiente?

El **objeto de estudio** de esta investigación es realizar un estudio profundo y consistente de los procesos de control de las actividades y administración de los datos de los estudiantes en la residencia estudiantil. Derivándose que el **campo de acción** son los procesos de control de las actividades y administración de los datos de los estudiantes en la residencia estudiantil de la

# Introducción

---



Universidad de las Ciencias Informáticas. La **idea a defender** es que; Si se desarrolla un correcto modelado del negocio y una buena especificación de los requisitos, que permita el análisis y diseño de los servicios, es posible garantizar la implementación del Módulo de Administración del Becado

El **objetivo general** de esta investigación es, desarrollar el análisis y diseño de una propuesta de un servicio web para su futura implementación que permita, de una forma más cómoda gestionar y controlar las informaciones de los becados en la residencia.

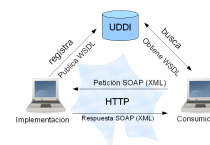
A partir del análisis del objetivo general se derivaron los siguientes **objetivos específicos**:

- Realizar el proceso de análisis y diseño de los servicios
- Diseñar servicios web accesibles a otras aplicaciones que necesiten procesar los procesos de control de las actividades y administración de los datos del becado en la residencia.

Para satisfacer estos objetivos y resolver el problema planteado anteriormente, se proponen las siguientes tareas:

1. Realizar un estudio sobre el control y gestión de las actividades y los datos en la residencia
2. Analizar otros sistemas con características similares al sistema propuesto.
3. Realizar búsquedas bibliográficas sobre la metodología, herramienta y lenguaje del modelado que se usarán para el desarrollo de la modelación y especificación del servicio web
4. Realizar análisis y diseño de un servicio web para el control actividades y administración de los datos del estudiante.

Como **métodos científicos** de investigación, se utilizaron los métodos teóricos y los métodos empíricos.



## Introducción

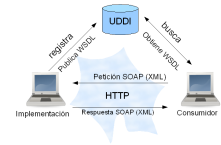
Como **métodos teóricos** se utilizaron los siguientes métodos: Inductivo-Deductivo, este permitió a través de un razonamiento, llegar a un grupo de conocimientos particulares y generales de la investigación. El método Histórico-Lógico, permitió iniciar esta investigación mediante la observación y análisis de los antecedentes del objeto de estudio. El Método de la Modelación, permitió la creación de modelos, es decir representar lo que se quiere estudiar de forma más simple, explicando lo que pasas de una manera lógica.

Como **método empírico**, se utilizó la Entrevista, este método permitió realizar el levantamiento de requisitos, necesario para realizar esta investigación, así como el funcionamiento del negocio y las herramientas a utilizar.

**Tabla 1** Cronograma de Investigación

No.	Acciones a realizar	responsable	participantes	fecha	
				inicio	Term.
1.	Realizar un estudio sobre el control y gestión de las actividades y los datos	Wendy Bokly		enero/2007	febrero/2008
2.	Analizar otros sistemas con características similares al sistema propuesto	Wendy Bokly		febrero/2008	marzo/2008
3.	Realizar búsquedas bibliográficas sobre la metodología, herramienta y lenguaje del modelado que se usaran para el desarrollo de la	Wendy Bokly		marzo/2008	abril/2008





## Introducción

---

	modelación y especificación del sistema.				
4.	Realizar análisis y diseño de la aplicación web para el control y administración de las actividades y datos del estudiante	Wendy Bokly		abril/2008	mayo/2008



## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

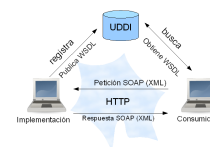
### Introducción

En este capítulo se brinda una panorámica de los semblantes relacionados con la gestión del becado, el control de los datos y las principales actividades, y de la forma que se realiza en la residencia de la UCI, es decir, sostener aspectos importantes de la teoría en que se basa la solución del problema, facilitando de esta forma la solución del mismo. Se analizan también sistemas informáticos que se han desarrollado con ese mismo propósito a nivel internacional, nacional y de universidad, así como las tendencias y tecnologías actuales que ayuden al desarrollo del módulo propuesto.

### 1.1 Control del Becado

Nuestra universidad cuenta con la residencia estudiantil más grande del país, ya que en ella se encuentran estudiantes de toda Cuba, la gestión de los datos del becado adquiere una importancia vital para el control y organización de dicha información, así como su constante actualización. Esto les permite a los trabajadores de la residencia agilizar este proceso para la toma de decisiones de los directivos, y constituye una vía fácil y rápida de llevar a cabo este proceso.

El control de los datos del becado consiste en la entrada, eliminación y actualización de los datos personales del becado. La universidad siempre ha optado por un estricto control en la residencia, ya que por su característica, propicia que personas ajenas a la universidad puedan entrar en ella.



## Capítulo 1: Fundamentación Teórica

---

En la residencia se realizan un conjunto de actividades en las cuales el becado debe participar. El control de las principales actividades es una excelente forma que controla la participación del becado en actividades, como: cuartelaría, guardia, limpieza del apartamento, TSU etc. Las actividades son fundamentales para obtención de la evaluación de la residencia, esa evaluación puede decidir la permanencia del estudiante en la universidad

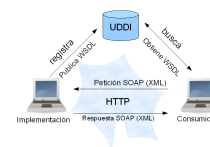
### 1.2 Sistemas Automatizados Existentes

Cada vez es más frecuente que los países estén interesados en tener sistemas automatizados para el control y gestión escolar, principalmente en el control de los estudiantes, manteniendo una constante actualización de la información y facilitar el trabajo de las personas que están al frente de la situación, mejorando la eficiencia y calidad de dicho trabajo.

#### 1.2.1 Sistemas existentes en el Extranjero

**DocCF:** Es un sistema que permite la gestión escolar, es construido por la empresa colombiana **Grupo CF Developer** dedicada a desarrollar productos y servicios informáticos al sector educativo. La gestión del sistema es por módulo permitiendo gestionar sin limitaciones información de alumnos, docentes, salones, grupos, horarios, matrículas, biblioteca, calificaciones, boletines escolares, cobros, pagos, inventario escolar, gestión de docencia, ausentismo y envío de email.

Los módulos que utiliza el sistema **DocCF** son los siguientes:



## Capítulo 1: Fundamentación Teórica

---

- **El Módulo de Gestión de Alumnos y Matriculas:** Permite el registro de información general del alumno, información académica, control de la disciplina, control del pago, informes de matriculas y control de ausencias.
- **El Módulo de Gestión Docente:** Permite una consulta de información del docente, Historia Académica por Escalafones, Perfil Profesional, Instituciones donde ha enseñado y evaluaciones docentes.

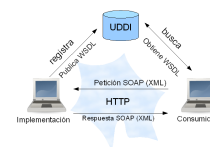
Este sistema es muy flexible. Se adapta a cualquier tipo de institución y permite intercambio y recepción a través de la Web. Puede centralizar información varios sistemas en uno solo y generar e imprimir informes.

**CALMECAC:** es un software diseñado para la administración de información básica escolar, aplicable a escuelas primarias, secundarias y bachillerato (posiblemente hasta nivel profesional), tanto públicas como privadas. Creado por Krunnware que es una consultora de TI (Tecnologías de la Información).

El sistema tiene las siguientes funcionalidades:

- Llevar el control de asistencias y evaluaciones de los alumnos.
- Tener un registro sobre los datos personales del alumno y el personal administrativo.
- Llevar un registro sobre los eventos peculiares (incidencias) del alumno y personal administrativo.
- Crear respaldos de su información.
- Emisión por periodo de las evaluaciones.
- Estadísticas por sexo y edad.
- Estadísticas de rendimiento por alumno y materia.

**CALMECAC** emite los siguientes reportes y estadísticas:



## Capítulo 1: Fundamentación Teórica

---

- Emisión por periodo de las evaluaciones.
- Credenciales de identificación para los estudiantes.
- Horarios por: profesor, grupo, asignaturas y, día.
- Catálogos: empleados, materias, grupos, alumnos, e incidencias.
- Cuadros de concentración.
- Estadísticas por sexo y edad.
- Estadísticas de rendimiento por alumno y materia.
- Evaluaciones de Conocimientos Básicos (escuelas primarias públicas).

**CALMECAC** cuenta con seguros internos que evitan que el usuario capture información incorrecta y los cálculos se realizan de manera automática. Además es gratuito no hay costos ocultos ni limitaciones en el uso de **CALMECAC**.

### 1.2.2 Sistemas existentes en Cuba

**SAIDO**: (Sistema Automatizado de Información Docente) es un sistema creado por **CECAM** (Centro de Cibernética Aplicada a la Medicina). Este sistema gestiona toda la información docente en las facultades de medicina del país. De este sistema se derivan diferentes paquetes que realizan funcionalidades específicas.

Algunos de los paquetes que utiliza **SAIDO** son los siguientes:

- **SAIDO-F** (Sistema automatizado de información docente, facultades): Es un sistema que incluye módulos de matrícula, actualización y notas de los alumnos, permitiendo obtener por selección listados sobre los datos de los mismos.



## Capítulo 1: Fundamentación Teórica

---

- **SICRE\_F** (Sistema de control de residentes, facultad): Mantiene datos sobre residentes activos, bajas y graduados, sus datos generales, evaluaciones, cursos, estancias y otras actividades.

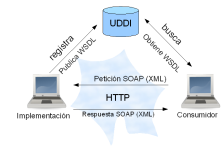
**Sistema de Control de Actividades Protocolares “Ceremonial”:** Fue creado por trabajadores del MIC (Ministerio de la Informática y las comunicaciones). El sistema se creó con el fin de mantener una supervisión estricta de las actividades planificadas para el cuerpo diplomático acreditado en nuestro país, ya sean ofrendas florales, recibimientos diplomáticos, entrevistas con funcionarios cubanos, reuniones, etc. Además, este sistema permite la gestión de solicitudes de reuniones en los salones del ministerio, y la planificación de estos últimos por esquemas diarios de trabajo y personalidades implicadas.

### 1.2.3 Sistemas existentes en la UCI

**Akademios:** El software, construido por un equipo de trabajo de la Dirección de Informatización de la UCI, se denomina Sistema Automatizado para la Gestión Académica. Este software permite la gestión automatizada de los elementos que intervienen en la labor académica. *Akademios* está distribuido en diferentes módulos que realizan actividades específicas.

Los principales módulos que utiliza **Akademios** son los siguientes:

- **Módulo de matrícula:** Se encarga del control de los datos de los estudiantes y la gestión de los movimientos a que son sometidos.
- **Módulo de registro:** Permite el control del desarrollo de un período académico con el registro de las evaluaciones y la asistencia.
- **Módulo reportes:** Obtiene reportes ordenados de la información almacenada en el sistema.



## Capítulo 1: Fundamentación Teórica

---

Es un sistema Web distribuido, desarrollado en la plataforma .NET, que utiliza SQL Server 2000 para el almacenamiento de los datos, IIS (Internet Information Services) como servidor Web y Servicios WEB XML para el intercambio con otras aplicaciones.

**Sistema de Gestión de Residentes:** Este sistema es creado por el departamento de informatización de la UCI. Este sistema gestiona la ubicación de los estudiantes en la residencia estudiantil, genera reportes de estudiantes por apartamentos, edificios, cuartos, lo que permite saber exactamente la ubicación de los estudiantes y saber el estado de la capacidad de cada apartamento.

El sistema es una aplicación Web desarrollada con PHP, utiliza PostgreSQL como gestor de base de datos para el almacenamiento de los datos, utiliza el Apache como servidor web y servicio web para intercambiar información con otras aplicaciones.

Al analizar estos sistemas se tomaron algunas características positivas como son:

La administración y control de los datos generales del estudiante, ya que contiene las principales características de estos, que son la entrada, eliminación y actualización de los datos.

Los reportes permiten obtener la información de una forma rápida y oportuna que facilita el intercambio de información así como la toma de decisiones.

A pesar que estos sistemas tienen algunos aspectos positivos, no reúnen todas las condiciones y características necesarias para satisfacer el trabajo de los trabajadores de la residencia de la universidad. Estos sistemas están enmarcados a las particularidades de sus organizaciones o no cumplen con todos los requisitos que debe tener este proyecto, que son bastante particulares.



## Capítulo 1: Fundamentación Teórica

---

### 1.3 Tendencias y desarrollo del software libre

Actualmente existen numerosas empresas y compañías informáticas en todo el mundo que han apostado por el uso del software libre. Este es el caso de IBM, Hewlett Packard, Silicon Graphics, la Administración del Espacio Aeronáutico Nacional (NASA) en los sistemas de misión crítica, el motor de búsqueda Google, el portal Yahoo, el Departamento de Energía de Estados Unidos, la multinacional British Petroleum y muchas otras. Estas confirmaron la calidad de los programas que son desarrollados sobre software libre, su gran estabilidad, seguridad y confiabilidad.

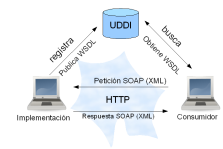
Las compañías y las empresas no son las únicas en reconocer las ventajas del software libre, sino que existen un gran número de países que están emigrando al software libre con el objetivo de eliminar la dependencia tecnológica, en este caso tenemos la región de Extremadura en España, donde se ha adoptado de manera oficial en los colegios y oficinas, el Gobierno Malayo ha decidido utilizar software libre desde noviembre del 2001, China ha contratado el desarrollo del software libre, Alemania es el país con mayor uso del software libre, en Venezuela el Gobierno Bolivariano ha establecido el uso obligatorio del software libre en empresas estatales, la ONU le recomienda a todos sus países miembro el uso del software libre.

Cada vez se observa un gran auge en el uso del software libre a nivel mundial por las grandes ventajas que el software libre propone, y por ser confiable y seguro.

#### 1.3.1 Comportamiento en Cuba y en la Universidad de las Ciencias Informáticas (UCI)

Nuestro país hace varios años que viene buscando la independencia tecnológica, introduciendo cada vez más en las instituciones el software libre, principalmente en las instituciones gubernamentales para garantizar la seguridad y la confiabilidad de la información. El Bloqueo Norteamericano es uno de los factores que han obligado a Cuba a migrar al software libre, Microsoft como toda empresa radicada en los Estados Unidos de América, está sujeta a las leyes de ese país, incluyendo las anti-cubanas que intentan





## Capítulo 1: Fundamentación Teórica

---

entre otras cosas, aislarnos del mundo tanto política como tecnológicamente. Microsoft es un monopolio imperial y como tal actúa, por lo que nadie está seguro en sus manos, y menos nosotros.

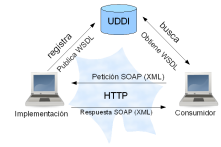
En Cuba las empresas que trabajan con conectividad tienen Linux o FreeBSD en sus servidores como: Infomed, la red telemática de salud, el Ministerio de Educación Superior, DESOFT, ETECSA, etc.

Además Cuba se encuentra enfrascada a realizar un crecimiento en la industria del software para fortalecer su economía, cuya única alternativa sostenible para que se logre es la migración hacia el software libre en todas las entidades estatales del país.

En los últimos años la UCI se ha convertido en la principal empresa de software de Cuba, buscando cada día ampliar más su mercado internacional para sus productos informáticos, y ha comenzado a migrar paulatinamente hacia las plataformas del software, actualmente existe la facultad 10, que es la encargada del estudio y desarrollo y promoción del software libre en la universidad.

En los laboratorios docentes se están instalando Windows y Linux a la vez para que los estudiantes se vayan relacionando con los sistemas operativos libres, también los principales servidores de la universidad están utilizando Linux.

En la UCI se realizan eventos y se han creado comunidades donde se debaten e intercambian conocimientos sobre software libre.



## 1.4 Tecnologías actuales

### 14.1 ¿Qué es un servicio web?

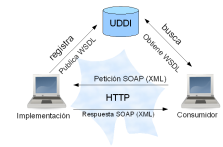
Los servicios Web son la revolución informática de la nueva generación de aplicaciones que trabajan en colaboración, las cuales el software está distribuido en diferentes servidores. Los servicios Web se comunican de forma estándar a través de los protocolos (http) y de manera independiente del lenguaje de programación, plataforma de implantación, formato de presentación o sistemas operativos. Los servicios web tienen una interfaz (WSDL) descrita en XML y es capaz de ser interpretado por cualquier máquina. Un servicio web no es una aplicación con interfaz que interactúa directamente con el usuario, sino que brinda servicios a aplicaciones clientes a través de mensajes XML codificados.

Los servicios web realizan funciones, las cuales van desde simples peticiones hasta procesos de negocios complicados. Cuando los servicios web son publicados, otros servicios pueden invocarlos usando sus servicios. Los servicios web tienen un aspecto importante a destacar, que es la capacidad de integrarse con otros servicios web para proveer un conjunto de mayores características, funcionalidades más complejas que a la vez se hacen más sencillas, gracias a la capacidad de estos de poder integrarse.

Un servicio Web es un contenedor que encapsula funciones específicas y hace que estas funciones puedan ser utilizadas por otros servidores.

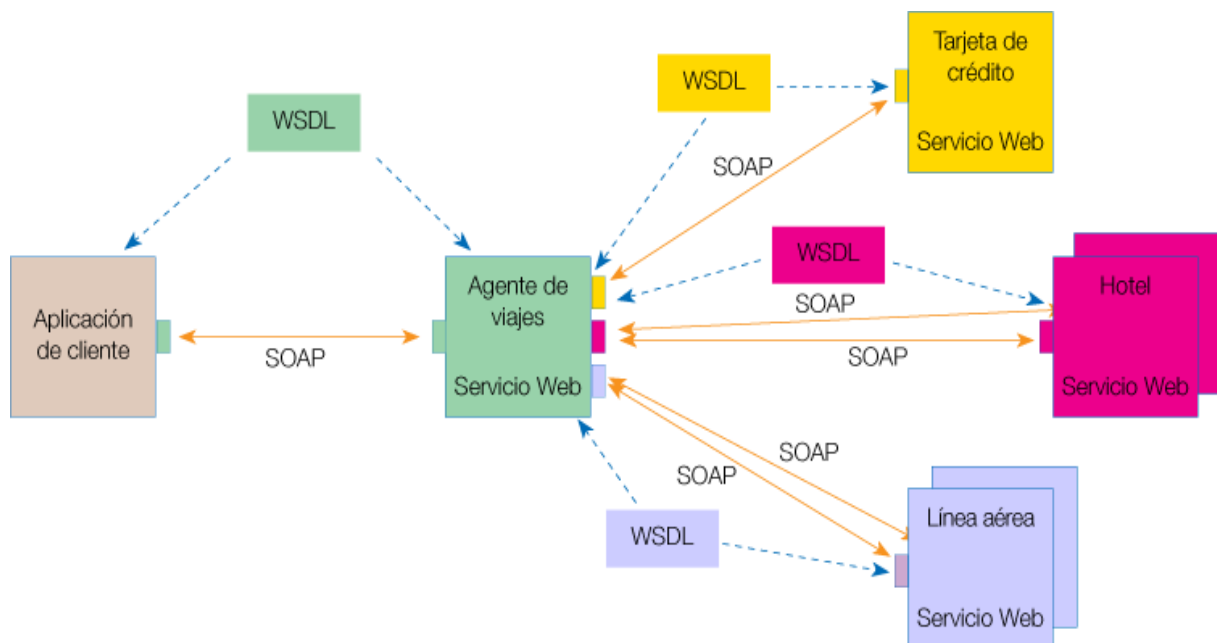
Algunas ventajas de los servicios Web son:

- Son programable
- Están basados en XML, que es un lenguaje abierto
- Pueden buscar registros de otros servicios web
- Son auto descriptivos
- El cliente no necesita conocer nada acerca de la implementación del servicio web
- El servidor y el cliente no necesitan estar escrito en el mismo lenguaje de programación.



## Capítulo 1: Fundamentación Teórica

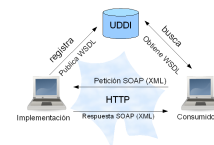
- Soporte para múltiples Lenguajes: Usted puede escribir SW en cualquier lenguaje
- Independencia en el modo de transporte. SOAP (Protocolo de Acceso Simple a Objetos) puede funcionar sobre múltiples protocolos de transportes, como por ejemplo: HTTP, HTTPS, HTTP-R, BEEP, JABBER, FTP, etc.



**Figura 1** Funcionamiento de los Servicios Web

### 1.4.2 Protocolo SOAP

Es un protocolo que permite la comunicación entre aplicaciones a través de mensajes por medio de Internet. Existen varios modelos de mensajes SOAP, pero el más común es el RPC, en donde un nodo de



## Capítulo 1: Fundamentación Teórica

---

le red (el cliente) envía un mensaje de solicitud a otro nodo (el servidor) y este último responde al mensaje del cliente. Estos mensajes son independientes de la (Brea, 2008) plataforma, y del lenguaje. SOAP está basado en XML y es la base principal de los servicios web. (1)

### 1.4.3 ¿Qué es NuSOAP?

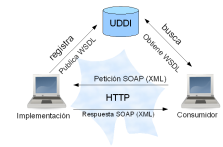
NuSOAP es un kit de herramientas (Toolkit) para desarrollar servicio web bajo el lenguaje PHP, desarrollada por la empresa NuSphere, licenciada bajo LGPL. Está compuesto por una serie de clases que nos harán mucho más fácil el desarrollo de Web Services. Provee soporte para el desarrollo de clientes (aquellos que consumen los Web Services) y de servidores (aquellos que los proveen). NuSOAP está basado en SOAP 1.1, WSDL 1.1 y HTTP 1.0/1.1. (1)

### 1.4.4 ¿Qué es WSDL?

WSDL (*Web Services Description Language*) es un formato XML que se utiliza para describir servicios web. WSDL describe la interfaz pública a los servicios Web. Está basado en XML y describe la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. Las operaciones y mensajes que soporta se describen en abstracto y se ligan después al protocolo concreto de red y al formato del mensaje. WSDL se usa en combinación con SOAP y XML Schema. Un programa cliente que se conecta a un servicio web puede leer el WSDL para determinar que funciones están disponibles en el servidor

### 1.4.5 UDDI

(Descripción Universal, Descubrimiento e Integración) es un registro público diseñado para almacenar de forma estructurada información sobre empresas y los servicios que estas ofrecen. A través de UDDI, se puede publicar y descubrir información de una empresa y de sus servicios. Se pueden utilizar sistemas taxonómicos estándar para clasificar estos datos y poder encontrarlos posteriormente en función de la



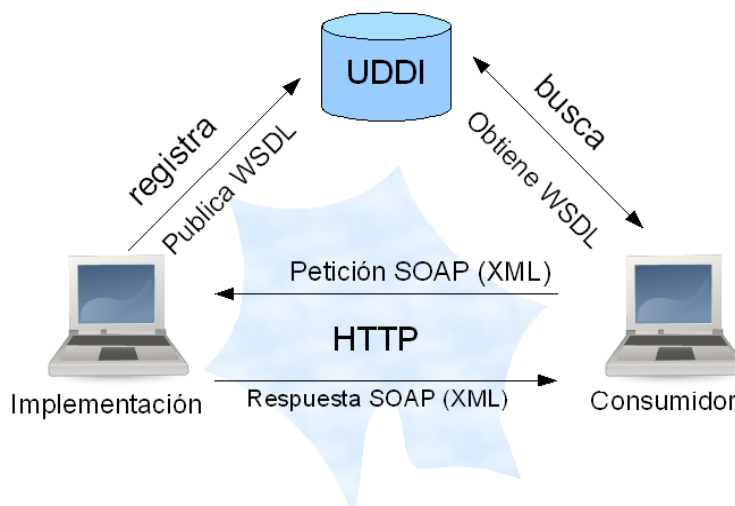
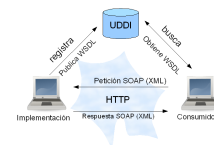
## Capítulo 1: Fundamentación Teórica

---

categorización. Lo más importante es que UDDI contiene información sobre las interfaces técnicas de los servicios de una empresa. A través de un conjunto de llamadas a basadas en SOAP, se puede interactuar con UDDI tanto en tiempo de diseño como de ejecución para descubrir datos técnicos de los servicios que permitan invocarlos y utilizarlos.

### 1.4.6 Arquitectura orientada a servicio

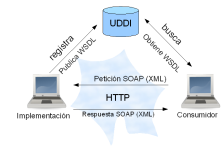
La Arquitectura Orientada a Servicios (SOA), es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requerimientos de software del usuario. Al contrario de las arquitecturas orientadas a objetos, los SOAs están formados por servicios de aplicación débilmente acoplados y altamente interoperables. Los Servicios Web se han convertido en el estandarte de SOA, ya que esta tecnología posee un conjunto de características que permiten cubrir todos los principios de la orientación a servicios. La definición de la interfaz encapsula las particularidades de una implementación, lo que la hace independiente del fabricante, del lenguaje de programación o de la tecnología de desarrollo.



**Figura 2** Arquitectura Tradicional de SOA

### 1.4.7 ¿Por qué utilizar Servicios Web?

Se utiliza el servicios web teniendo en cuenta que existen servicios de aplicaciones existentes que permiten compartir datos y pueden ser solicitados a través de diferentes plataformas y sistemas operativos, sin tener en cuenta el lenguaje de programación, lo que posibilita que se pueda gestionar cualquier tipo de información. Los servicios web proporcionan conectividad con cualquier software de un modo transparente por el paso de mensaje a través del protocolo SOAP, pueden implementar todas sus funcionalidades y mantenerse seguro, ya que por cada mensaje simple que se intercambia realiza múltiples saltos y es ruteado a través de numerosos puntos antes de que alcance su destino final. Los servicios web se registran para que sean localizados fácilmente y dicho registro es a través de UDDI (descripción, descubrimiento e integración universales).



## Capítulo 1: Fundamentación Teórica

---

Están basados en XML, que ha sido universalmente aceptado. Un servicio web puede ser implementado con el empleo de cualquier lenguaje de programación, teniendo en cuenta las características y facilidades del utilizado.

Entre los múltiples lenguajes se encuentran: **ASP.Net**, **JSP** (*Java Server Pages*), **Perl**, **PHP**, **Python**, entre otros.

### 1.4.8 Lenguajes de Programación para Web

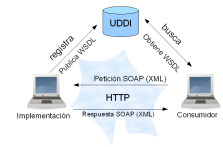
Hoy en día el Internet se ha convertido en el canal de comunicación más usado del mundo, por todas las ventajas y potencialidades que brindan todos los sistemas que soporta, permitiendo la interacción con los usuarios y la personalización.

Existen varios lenguajes de programación que le permiten la gran interactividad a las aplicaciones Web, tanto en el lado del servidor como del cliente. Los lenguajes de programación más usados en software libre, que corren en el servidor y procesan toda la lógica del negocio son PHP, JAVA y Perl. Los lenguajes de programación Web del lado del cliente, más utilizados, encargados de visualizar la información en el navegador y la validación de la información en los formularios, son HTML, JavaScript y Ajax.

A continuación se describen los lenguajes, herramientas y metodología seleccionada por el proyecto:

### 1.4.9 Languages de programación a utilizar

**PHP:** El lenguaje de programación PHP denominado procesador de hipertexto es de lenguaje de código abierto y multiplataforma, es uno de los más utilizado para la creación de páginas Web. Este lenguaje se encuentra por versión 5.0 y en Internet se encuentra una amplia documentación que posibilita un mejor entendimiento del lenguaje. PHP es un lenguaje orientado a objeto y permite la herencia, incluye también la programación estructurada a servicios Web.



## Capítulo 1: Fundamentación Teórica

---

PHP tiene la capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.

Cuenta con una biblioteca nativa de funciones sumamente amplia e incluida (acceso a base datos, encriptación, envío de correo, XML, creación de PDF, etc.).

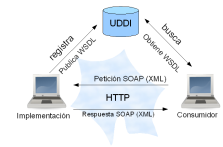
Debido a su amplia distribución PHP está perfectamente soportado por una gran comunidad de desarrolladores. Como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparan rápidamente. El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.

**XML:** En sus siglas en inglés por Extensible Markup Language (lenguaje de marcas extensible), no es realmente un lenguaje en particular, sino un protocolo de comunicación entre aplicaciones Web. Existen algunos lenguajes de programación que utilizan XML, entre ellos se encuentran PHP y JAVA.

XML es una tecnología en realidad muy sencilla, y permite una manera distinta de hacer las cosas, más avanzada, cuya principal novedad consiste en permitir compartir los datos con los que se trabaja a todos los niveles, por todas las aplicaciones y soportes.

*XML no es un lenguaje, sino varios lenguajes, no es una sintaxis, sino varias y no es una manera totalmente nueva de trabajar, sino una manera más refinada que permitirá que todas las anteriores se puedan comunicar entre si sin problemas, ya que los datos cobran sentido. (DesarrolloWeb.com 2008).*





## Capítulo 1: Fundamentación Teórica

---

**HTML:** Hipertexto Parking Language. [Lenguaje de marcado de hipertextos]. Lenguaje de programación que permite la inclusión de textos, imágenes fijas y móviles, vídeo, archivos, etc., y su enlace mediante hipertexto por el usuario de la computadora, independientemente de la estructura de la máquina o del sistema. Base de la estructura de aplicación WWW.

Algunas ventajas que presenta, son la facilidad con que se pueden actualizar los contenidos y que permite utilizar estilos en formato CSS (hojas de estilos en cascada) en las páginas para una mayor facilidad en su modificación.

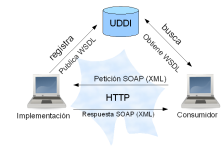
HTML no es un Lenguaje de Programación, aunque si permite incluirle código en Lenguaje de Programación, bajo ciertos criterios, extendiendo su capacidad y funcionalidad, aunque eso se logre excediendo los alcances del HTML en si.

El lenguaje HTML puede ser creado y editado con cualquier editor de textos básico, como puede ser el Bloc de Nota de Windows, Gedit, o cualquier otro editor que admita texto sin formato como GNU Emacs, Microsoft Word, TextPad, etc.

**JavaScript:** Es un lenguaje de programación que permite a los desarrolladores crear acciones en sus páginas Web y en programas más grandes, orientados a objetos mucho más complejos. Con Javascript podemos crear diferentes efectos e interactuar con nuestros usuarios. (3)

JavaScript es un lenguaje basado en acciones que posee menos restricciones. Además gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas entre otros.

Su sintaxis es similar a la usada en Java y C, al ser un lenguaje del lado del cliente este es interpretado por el navegador, no se necesita tener instalado ningún Framework. El código script tiene capacidades limitadas, por razones de seguridad, por lo cual no es posible hacer todo con Javascript, sino que es necesario usarlo conjuntamente con HTML



## Capítulo 1: Fundamentación Teórica

---

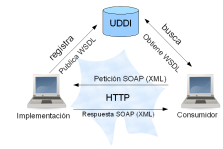
**Ajax:** (Asynchronous JavaScript And XML) corresponde a dos lenguajes, Java y XML que interactúan asincrónicamente. Ajax es una técnica de desarrollo Web para crear aplicaciones interactivas. Éstas se ejecutan en el cliente, es decir en el navegador de los usuarios y mantiene una comunicación asincrónica con el servidor en un segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla.

Ajax es una combinación de cuatro tecnologías ya existentes; HTML y hojas en cascadas CSS para el diseño que acompaña a la información, DOM (Document Object Model) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada, el objeto XMLHttpRequest para intercambiar datos asincrónicamente con el servidor web, y XML es formato usado comúnmente para la transferencia de vuelta al servidor, aunque cualquier formato puede funcionar, incluyendo HTML preformateado, texto plano, JSON, y hasta EBML.

### 1.4.10 Zend Studio Como IDE a utilizar

Es un programa construido por zend, que han sido los impulsores la tecnología de servidor PHP, orientada a desarrollar aplicaciones Web. El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código.

El programa entero está escrito en Java, y zend ha lanzado varias versiones del producto para Windows, Linux, y MacOs. Zend Studio consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor. La parte del cliente contiene el interfaz de edición y la ayuda. Permite además hacer depuraciones simples de scripts, pero para aprovechar toda la potencia de la herramienta de depuración, habrá que disponer de la parte del servidor que instala Apache y el módulo PHP.



## Capítulo 1: Fundamentación Teórica

---

Zen Studio posee el Zend Framework, que se trata de un framework para desarrollo de aplicaciones Web y servicios Web con PHP, te brinda soluciones para construir sitios Web modernos, robustos y seguros. Además es Open Source y trabaja con PHP 5.

### 1.4.11 Sistema Gestores de Base Datos

Un Sistema Gestor de base de datos (SGBD) es un conjunto de programas que permiten crear y mantener una Base Datos asegurando su integridad, confidencialidad y seguridad. Por tanto debe permitir:

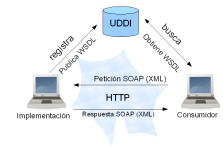
- Definir una base de datos: especificar tipos, estructuras y restricciones de datos...
- Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD
- Manipular la base de datos: realizar consultas, actualizarla, generar informes.

Algunas de las características deseables en un Sistema Gestor de base de datos SGBD son:

- Control de la redundancia: La redundancia de datos tiene varios efectos negativos (duplicar el trabajo al actualizar, desperdicia espacio en disco, puede provocar inconsistencia de datos) aunque a veces es deseable por cuestiones de rendimiento.
- Restricción de los accesos no autorizados: cada usuario ha de tener unos permisos de acceso y autorización.
- Cumplimiento de las restricciones de integridad: el SGBD ha de ofrecer recursos para definir y garantizar el cumplimiento de las restricciones de integridad. (4)

### 1.4.12 MySQL Como Gestor de Base Datos

Es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo, debido a su gran rapidez y facilidad de uso. Esta gran



## Capítulo 1: Fundamentación Teórica

---

aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración. Este gestor está hecho para aplicaciones de Internet pequeñas, que no realicen muchas transacciones al mismo tiempo, presenta una estabilidad bastante alta y una seguridad en los datos. MySQL es multiplataforma, y con PHP hace la combinación perfecta.

### 1.4.14 Metodologías para el desarrollo de software. RUP

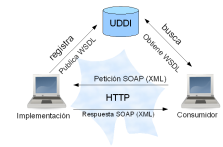
*Un proceso define quién está haciendo qué, cuándo y cómo alcanzar un determinado objetivo. En la Ingeniería del software el objetivo es construir un producto software o mejorar uno existente (IVAR JACOBSON).*

Un proceso efectivo proporciona normas para el desarrollo eficiente de software de calidad y presenta las mejores prácticas que el estado actual que la tecnología permite; en consecuencia, reduce el riesgo y hace el proyecto más predecible.

Las metodologías fueron diseñadas bajo un conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para lograr la calidad, que es el principal objetivo estratégico en las organizaciones. Su objetivo es elevar la calidad del software (en todas las fases por las que pasa) a través de una mayor transparencia y control sobre el proceso.

Todo desarrollo de software es riesgoso y difícil de controlar, pero utilizando un proceso o una metodología, entonces podremos tener a los clientes y desarrolladores más satisfechos con el resultado del trabajo. Para el desarrollo del módulo propuesto se analizarán las dos corrientes que más se utilizan en el mundo, conocidas como: metodologías pesadas y metodologías ligeras.

0 La metodología RUP (Rational Unified Process) es una metodología pesada, de IBM Rational para el desarrollo y construcción de software basado íntegramente en UML, el cual permite especificar, construir,



## Capítulo 1: Fundamentación Teórica

---

visualizar y documentar los artefactos de un sistema de software orientado a objetos. Sus principales características son:

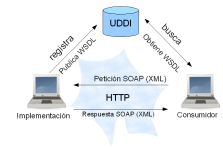
- **Centrados en los modelos:** Los diagramas son un vehículo de comunicación más preciso que las descripciones en lenguaje natural. Se trata de minimizar el uso de descripciones y especificaciones textuales del sistema.
- **Guiados por los casos de uso:** Los casos de uso son el instrumento para describir el comportamiento del software y extraer los casos de prueba con los que se valida el sistema.
- **Centrados en la arquitectura:** Los modelos son proyecciones del análisis y el diseño describe la arquitectura del producto a desarrollar.
- **Iterativo e incremental:** Durante todo el proceso de desarrollo se producen versiones superiores.

*El Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto (IVAR JACOBSON).*

Se divide en ciclos de trabajo, teniendo un producto superior como resultado de cada ciclo. Estos se componen en su interior por 4 fases, en las cuales se llevan a cabo un conjunto de flujos para el desarrollo de todo el proyecto.

Las 4 fases de RUP para el desarrollo del software son las siguientes:

- Fase Inicio, tiene como objetivo determinar la visión del proyecto.
- Fase Elaboración, el objetivo es determinar la arquitectura óptima.
- Fase Construcción, el objetivo es llevar a obtener la capacidad operacional inicial.
- Fase Transmisión, el objetivo es llegar a obtener el release del proyecto.



## Capítulo 1: Fundamentación Teórica

---

RUP es la metodología de desarrollo de software más usada a nivel mundial, por las grandes potencialidades que presenta para el análisis, implementación y documentación de sistemas orientados a objetos, es el Proceso Unificado de Desarrollo o RUP (Rational Unified Process) con el Lenguaje Unificado de Modelado (UML).

Con RUP se presentan al cliente los artefactos del final de una fase y se valorarán las precondiciones para la siguiente (definición de riesgos, aceptación del plan de iteración, prototipos, etc.) y solo después de que el cliente acepte los artefactos generados se pasará a la siguiente fase. Además RUP brinda una guía para encontrar, organizar, documentar, y seguir los cambios de los requisitos funcionales y las restricciones. Utiliza una notación de Caso de Uso y escenarios para representar los requisitos.

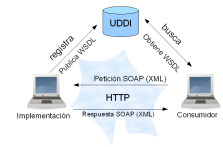
RUP permite seleccionar fácilmente el conjunto de componentes de proceso que se ajustan a las necesidades específicas del proyecto y se podrán alcanzar resultados predecibles unificando el equipo con procesos comunes que optimicen la comunicación y creen un entendimiento común para todas las tareas, responsabilidades y artefactos.

La herramienta CASE (ingeniería de software asistida por el ordenador) a utilizar para el modelado de la propuesta del módulo, es el Visual Paradigm for UML 6.0 Enterprise Edition, ya que Visual Paradigm para UML es una herramienta UML profesional, soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue, es una herramienta libre, permite establecer una trazabilidad real entre el modelo (análisis y diseño) y el código ejecutable. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor costo.

### 1.4.15 UML Como lenguaje de modelado

Es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Se usa para entender, diseñar, configurar, mantener y controlar la





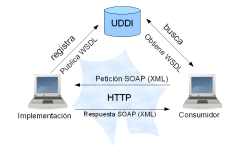
## Capítulo 1: Fundamentación Teórica

---

Este servicio Web utilizará el lenguaje PHP en el servidor y Javascript y AJAX en el cliente, para que el servidor no esté muy sobrecargado y las mayores funcionalidades la realice el navegador. MySQL como gestor de base de datos.

La metodología de desarrollo del sistema será RUP, estándar más utilizado para el análisis, implementación y documentación de sistemas orientados a objetos a nivel internacional.





## Capítulo 2: Características del Sistema

---



## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

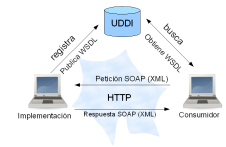
### Introducción

En este capítulo se realizará una breve descripción del estado actual del negocio, se describirán las reglas del negocio a considerar para la automatización del módulo. Se presenta el modelo del negocio, se define el diagrama de caso de uso del negocio, la descripción de cada proceso y los diagramas de actividades de cada uno de estos, que describen la secuencia de actividades que tienen lugar. Además se identifican los requerimientos funcionales y no funcionales, y a partir de éstos se definen los casos de uso del sistema y sus descripciones.

### 2.1 Estado actual del negocio

Actualmente en la residencia estudiantil el control de las actividades en la que participan los estudiantes se hacen de forma manual generando así un gran volumen de información, en la cual se invierte tiempo en organizarla y mantenerla actualizada y también recursos de oficina. Esto posibilita la pérdida de información y además de que la evaluación otorgada al estudiante no sea la correcta.

Existen también algunos datos como: nombre y apellidos, carne de identidad, número de solapín, dirección particular y teléfono, grupo, año que cursa, si son militantes (UJC y/o PCC), si son dirigentes de la FEU o la UJC, edificio, apto y teléfono, proyecto en los que participan, laboratorio en que trabaja y teléfono, si son alumnos ayudantes, asignatura que imparten, evaluación del becado, que no están controlados, o están controlados de forma manual por las instructoras y directivos de la residencia y que se hace muy trabajoso buscar estas informaciones que son vitales para cualquier circunstancia.



## Capítulo 2: Características del Sistema

---

### 2.2 Propuesta de Automatización

La propuesta de automatización de este trabajo es el análisis y diseño de un servicio web para una futura implementación, que permita el control de las actividades y administración de los datos personales del estudiante en la residencia de la facultad. Estas funcionalidades pueden ser usadas por cualquier sistema informático, independientemente de la plataforma en que hayan sido desarrollados.

Las principales funcionalidades a automatizar son:

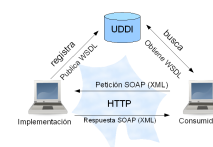
- Gestionar los datos del estudiante. Los datos son nombre y apellidos, carnet de identidad, número de solapín, dirección particular y teléfono, correo, año que cursa, si son militantes (UJC y/o PCC), si son dirigentes de la FEU o la UJC, edificio, apto y teléfono, proyecto en los que participan, laboratorio en que trabaja y teléfono, si son alumnos ayudantes, asignatura que imparten, evaluación del becado.
- Controlar las principales actividades en la que participa el estudiante: cuarterería, limpieza del apto, guardia estudiantil y TSU
- Generar reporte de estudiante: reporte por categoría:(militantes, dirigente, evaluación mensual y Alumno ayudante), por año, por edificio y por apto

### 2.3 Reglas del Negocio

Los procesos que se llevan a cabo en cualquier centro cumplen determinadas reglas que deben seguirse en el análisis y diseño de la aplicación, a fin de respetar y garantizar las restricciones que existen en el negocio.

Las siguientes reglas son:

- La instructora evalúa solamente a los estudiantes de su edificio
- La instructora controla solamente las actividades de su edificio
- El jefe de la guardia es el único que realiza el informe de asistencia de la guardia.



## Capítulo 2: Características del Sistema

- El jefe del TSU es el único que realiza el informe de asistencia del TSU.
- El estudiante debe tener una evaluación de la residencia todos los meses (evaluación mensual)
- El estudiante debe firmar la guardia estudiantil al comenzar y terminar la guardia

### 2.4 Modelo de negocio

El modelo del negocio se realiza con el propósito de comprender las características y actividades que se llevan a cabo en el contexto a automatizar. En él, se modelan los clientes y trabajadores del negocio y su interacción con los procesos y objetos que lo componen. Está formado por el modelo de casos de uso del negocio y el modelo de objetos del negocio.

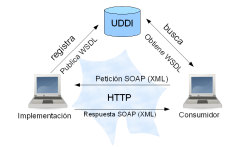
### 2.5 Actor del negocio

Los actores del negocio son aquellas personas o sistemas que obtienen un resultado de valor de uno o varios procesos del negocio.

El actor del negocio estudiado se define en la siguiente tabla:

**Tabla 2** Actores del Negocio

Actor del Negocio	
Actor del negocio	Justificación
Estudiante	El control de las actividades y de los datos es del es del estudiante.



## Capítulo 2: Características del Sistema

### 2.6 Trabajadores del negocio

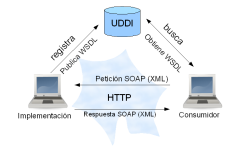
Los trabajadores del negocio son aquellas personas o sistemas que están involucrados en uno o más procesos del negocio, que participan en ellos pero no obtienen ningún beneficio, es decir no obtienen ningún resultado de valor. Se encuentran relacionados con las entidades del negocio.

**Tabla 3** Trabajadores del Negocio

Trabajadores del Negocio	
Trabajadores del negocio	Justificación
Instructora	Es la encargada de controlar los datos y actividades de los estudiantes de su edificio.
Jefe de la guardia	Es el encargado de controlar la participación de la guardia estudiantil
Jefe del TSU	Se encarga de controlar la participación de los estudiantes al TSU.
Psicopedagoga	Realiza los informes de incidencia, limpieza, cuartería y de la evaluación mensual de todos los estudiantes de la facultad

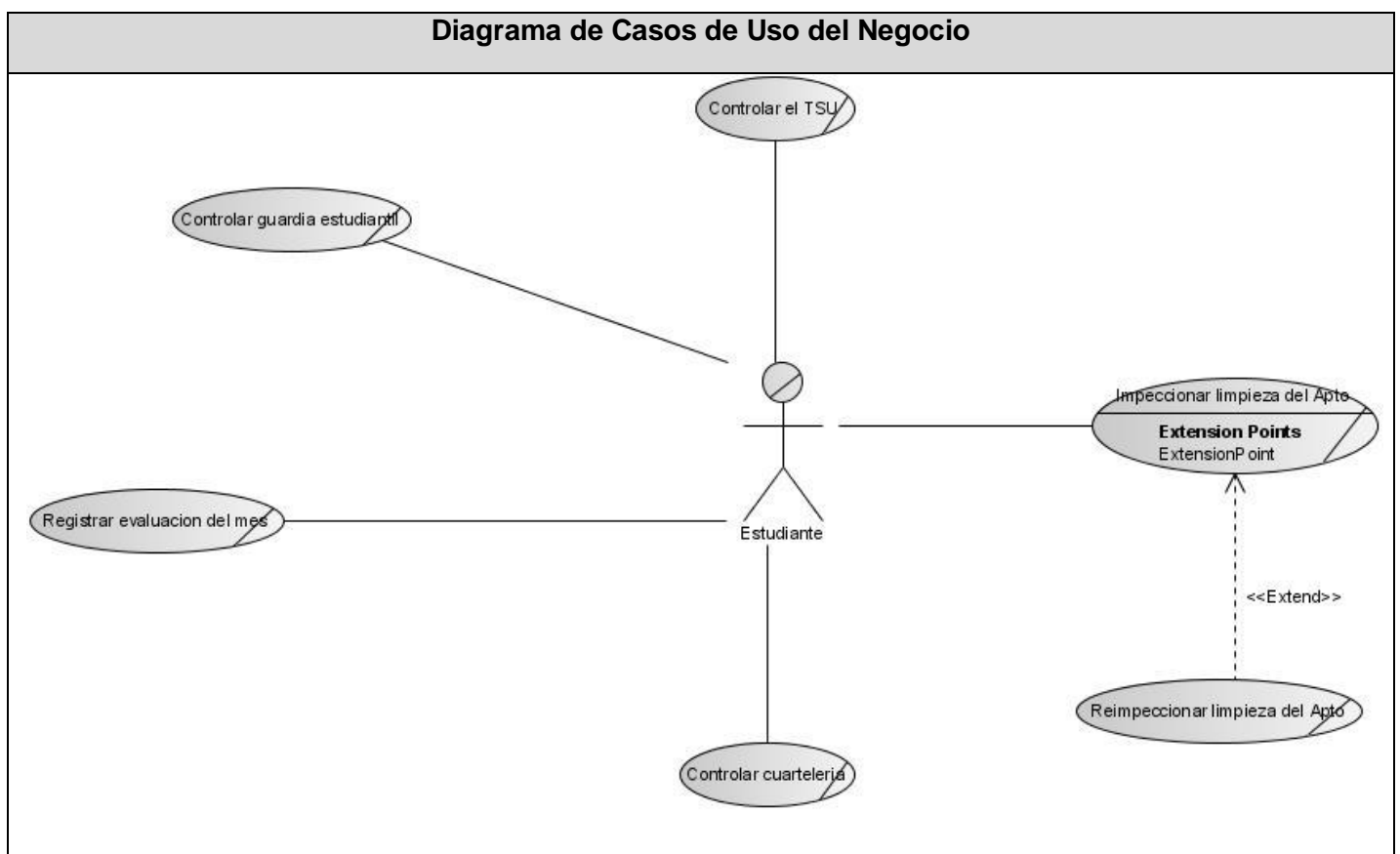
### 2.7 Modelo de casos de usos del negocio

El modelo de casos de uso del negocio describe los procesos del negocio en términos de casos de uso del negocio y actores del negocio, que se corresponden con los procesos del negocio y los clientes de éste respectivamente. Este modelo se describe mediante diagramas de casos de uso que muestran la relación entre los casos de uso y los actores del negocio.



## Capítulo 2: Características del Sistema

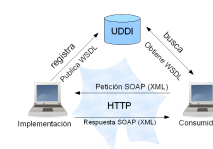
A continuación (figura 3) se muestra los distintos casos de uso que fueron identificados en el negocio, así como sus descripciones textuales de cada uno de manera detallada.



**Figura 3** Diagrama de Casos de Uso del Negocio

### 2.8 Descripción Textual de los casos de uso

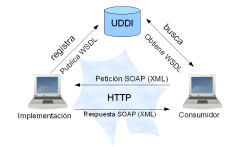
A continuación se muestra la descripción textual de los casos de uso del negocio, donde se expresan las diferentes relaciones entre actores y trabajadores del negocio, como resultado de las actividades que se realizan en el negocio



## Capítulo 2: Características del Sistema

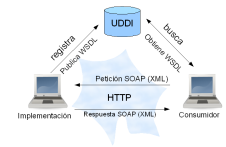
**Tabla 4** Diagrama de Caso de Uso. “Controlar Cuartelería”

<b>Caso de Uso</b>	Controlar Cuartelería	
<b>Actores del negocio:</b>	Estudiante	
<b>Propósito:</b>	Controlar el cumplimiento de la cuartelería, y registrar las incidencias que puedan surgir	
<b>Trabajadores</b>	Instructora de Residencia (Inicia) Psicopedagoga Vicedecano de Residencia	
<b>Resumen</b>	<p>El caso de uso se inicia cuando la instructora revisa a que estudiante le toca la cuartelería. La instructora revisa la limpieza de la escalera y las áreas del paso de escalera, la instructora realiza un informe y se lo envía a la psicopedagoga. La psicopedagoga añade esta información en el control de cuartelería, crea el informe semanal de la cuartelería y se lo envía al vicedecano de residencia, este revisa el informe y toma alguna decisión. Termina el caso de uso</p>	
<b>Casos de uso asociados:</b>	Ninguno	
<b>Flujo de trabajo</b>		
<b>Acción del actor</b>	<b>Respuesta del negocio</b>	
	1. La instructora revisa a que estudiante le toca la cuartelería	
	2. La instructora revisa las escaleras y las áreas	



## Capítulo 2: Características del Sistema

	3. La instructora realiza un informe.
	4. La instructora le envía el informe a la psicopedagoga
	5. La psicopedagoga recibe el informe
	6. La psicopedagoga añade la información del informe en el Control de la Cuartelaría
	7. La psicopedagoga realiza el informe semanal de la cuartelaría.
	8. La psicopedagoga le envía el informe al vicedecano de residencia
	9. El vicedecano recibe dicho informe
	10. El vicedecano lee el informe
	11. El vicedecano guarda el informe
	12. Terminó el caso de uso
<b>Prioridad:</b>	Alta
<b>Mejoras:</b>	
<b>Cursos alternos:</b>	2. En caso de que existan incidencias, la instructora registra las incidencias.

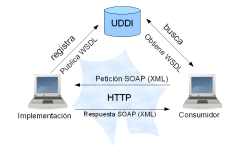


## Capítulo 2: Características del Sistema

**Tabla 5** Caso de Uso “Controlar Guardia Estudiantil”

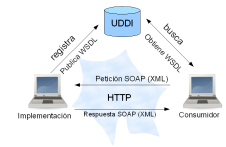
<b>Caso de Uso</b>	Controlar Guardia Estudiantil	
<b>Actores del negocio:</b>	Estudiante	
<b>Propósito:</b>	Controlar el cumplimiento de la guardia estudiantil	
<b>Trabajadores</b>	<p>Jefe de la guardia (inicia)</p> <p>Personal de guardia del puesto mando UCI</p> <p>Vicedecano de Residencia</p>	
<b>Resumen</b>	<p>El caso de uso se inicia cuando el Jefe de la guardia distribuye la guardia. El estudiante debe firmar el registro de guardia cuando comience y finalice su turno de guardia. El Jefe de la guardia debe hacer un informe de la guardia y enviárselo a vicedecano de residencia, este revisa el informe y lo archiva.</p> <p>Se terminó el caso de uso</p>	
<b>Casos de uso asociados:</b>	Ninguno	
<b>Flujo de trabajo</b>		
<b>Acción del actor</b>	<b>Respuesta del negocio</b>	
	1. El Jefe de la guardia distribuye la guardia	
	2. Entrega la distribución en el puesto de mando	
	3. El personal de guardia del puesto de mando UCI recoge la relación de la guardia	
	4. El jefe de la guardia firma la guardia	





## Capítulo 2: Características del Sistema

5. El estudiante se dirige al puesto de mando	
6. Firma la entrada de la guardia	
7. Realiza su turno de guardia	
8. Termina el turno de guardia	
9. Vuelve al puesto de mando	
10. Firma la salida de la guardia	
	11. El jefe de la guardia realiza un informe.
	12. El jefe de la guardia envía el informe al vicedecano de residencia.
	13. El vicedecano de residencia recibe el informe.
	14. El vicedecano revisa el informe
	15. Guarda el informe
<b>Prioridad:</b>	Media
<b>Mejoras:</b>	
<b>Cursos alternos:</b>	

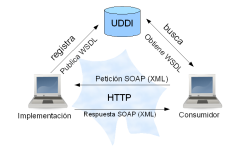


## Capítulo 2: Características del Sistema

--	--

**Tabla 6** Caso de Uso “Controlar TSU”

<b>Caso de Uso</b>	Controlar TSU	
<b>Actores del negocio:</b>	Estudiante	
<b>Propósito:</b>	Controlar la asistencia al TSU	
<b>Trabajadores</b>	Jefe TSU (inicia) Vicedecano de Residencia	
<b>Resumen</b>	El caso de uso se inicia cuando el jefe del TSU distribuye el trabajo, el estudiante realiza el trabajo, el jefe del TSU toma la asistencia, realiza un informe y se lo envía al vicedecano de residencia. El vicedecano lee y archiva el informe. Termina el caso de uso	
<b>Casos de uso asociados:</b>	Ninguno	
<b>Flujo de trabajo</b>		
<b>Acción del actor</b>	<b>Respuesta del negocio</b>	
	1. El jefe del TSU distribuye el trabajo	
2. El estudiante comienza el trabajo		
3. Termina el trabajo		
	4. El jefe del TSU toma la asistencia	

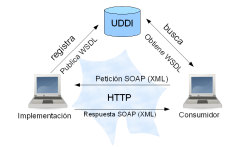


## Capítulo 2: Características del Sistema

	5. El jefe del TSU realiza un informe
	6. El jefe del TSU le envía el informe al vicedecano de residencia
	7. El vicedecano de residencia recibe el informe
	8. El vicedecano de residencia revisa el informe
	9. El vicedecano de residencia guarda el informe.
	10. Termina el caso de uso
<b>Prioridad:</b>	Media
<b>Mejoras:</b>	
<b>Cursos alternos:</b>	

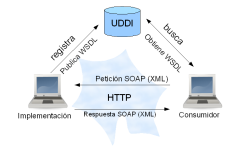
**Tabla 7** Caso de Uso “Registrar la evaluación del mes”

<b>Caso de Uso</b>	Registrar la evaluación del mes
<b>Actores del negocio:</b>	Estudiante
<b>Propósito:</b>	Evaluar el comportamiento del estudiante durante el mes en la residencia.



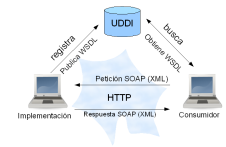
## Capítulo 2: Características del Sistema

<b>Trabajadores</b>	Instructora (inicia) Psicopedagoga Vicedecano de Residencia	
<b>Resumen</b>	<p>La instructora busca el nombre del estudiante, revisa las evaluaciones diarias que tuvo durante el mes, en caso de que el estudiante haya realizado la cuartelaría la instructora revisa la evaluación que obtuvo el estudiante. En caso de que el estudiante no haya tenido cuartelaría ese mes, la evaluación final del mes depende nada más de las evaluaciones diarias obtenidas. El estudiante firma la evaluación. La instructora realiza el informe mensual de las evaluaciones y se lo envía a la psicopedagoga. La psicopedagoga revisa el informe y añade la información al Registro General de evaluaciones mensuales del estudiante y le envía una copia de este documento al vicedecano de residencia. El vicedecano revisa el registro y lo archiva. Terminó el caso de uso</p>	
<b>Casos de uso asociados:</b>	Ninguno	
<b>Flujo de trabajo</b>		
<b>Acción del actor</b>	<b>Respuesta del negocio</b>	
	1. La instructora busca el nombre del estudiante.	
	2. La instructora revisa las evaluaciones diarias que obtuvo el estudiante durante el mes	



## Capítulo 2: Características del Sistema

	3. La instructora revisa la distribución de la cuartelaría del mes
	4. La instructora escribe la evaluación final del mes
	5. La instructora le lleva el registro al estudiante
6. El estudiante firma la evaluación	
	7. La instructora realiza el informe mensual de las evaluaciones
	8. La instructora le envía el informe a la psicopedagoga.
	9. La psicopedagoga recibe el informe.
	10. La psicopedagoga revisa el informe
	11. La psicopedagoga añade la información al registro general de las evaluaciones mensuales del estudiante
	12. La psicopedagoga le envía una copia del registro al vicedecano de residencia.
	13. El vicedecano de residencia recibe la copia del registro
	14. El vicedecano de residencia revisa el registro
	15. El vicedecano de residencia guarda el registro.
	16. Termina el caso de uso

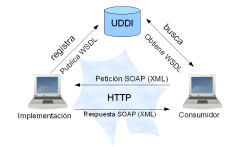


## Capítulo 2: Características del Sistema

<b>Prioridad:</b>	Media
<b>Mejoras:</b>	
<b>Cursos alternos:</b>	3. En caso de que el estudiante tuvo cuartelaría, la instructora revisa la evaluación que el estudiante obtuvo ese mes.

**Tabla 8** Caso de Uso “Inspeccionar la limpieza del Apto”

<b>Caso de Uso</b>	Inspeccionar la limpieza del Apto	
<b>Actores del negocio:</b>	Estudiante	
<b>Propósito:</b>	Controlar la limpieza y organización de los apartamentos	
<b>Trabajadores</b>	Instructora (inicia)	
<b>Resumen</b>	El caso de uso se inicia cuando la instructora llega al apartamento a realizar la primera inspección. La instructora revisa a que estudiante le toca la limpieza, revisa la limpieza, en caso de que no existan incidencias, la instructora evalúa de B al apartamento. Termina el caso de uso	
<b>Casos de uso asociados:</b>	Reinspeccionar limpieza del Apto (Extend)	
<b>Flujo de trabajo</b>		
<b>Acción del actor</b>	<b>Respuesta del negocio</b>	
	1. La instructora llega al apartamento a realizar impacción.	

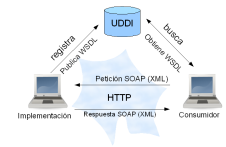


## Capítulo 2: Características del Sistema

	2. Revisa a que estudiante le toca la limpieza
	3. Revisa la limpieza
	4. Evalúa de B al apartamento
	5. Termina el caso de uso
<b>Prioridad:</b>	Alta
<b>Mejoras:</b>	
<b>Cursos alternos:</b>	3. En caso de que existan incidencias, la instructora registra las incidencias, y se retira del apartamento.

**Tabla 9** Caso de Uso “Reinspeccionar la limpieza”

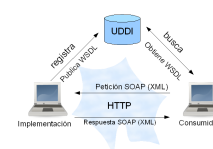
<b>Caso de Uso</b>	Reinspeccionar la limpieza
<b>Actores del negocio:</b>	Estudiante
<b>Propósito:</b>	Controlar la limpieza del apartamento nuevamente
<b>Trabajadores</b>	Instructora (inicia) Psicopedagoga Vicedecano de Residencia



## Capítulo 2: Características del Sistema

<b>Resumen</b>	El caso de uso se inicia en caso de que la instructora haya encontrado incidencias. La instructora realiza una reinspección al apartamento, revisa la limpieza, en caso de que las incidencias hayan sido erradicadas, la instructora evalúa de B al apartamento y se retira. Termina el caso de uso.	
<b>Casos de uso asociados:</b>	Ninguno	
<b>Flujo de trabajo</b>		
<b>Acción del actor</b>	<b>Respuesta del negocio</b>	
	1. La instructora llega al apartamento	
	2. La instructora revisa la limpieza	
	3. La instructora evalúa de B al apartamento	
	4. Termina el caso de uso	
<b>Prioridad:</b>	Alta	
<b>Mejoras:</b>		
<b>Cursos alternos:</b>	2. En caso de que las incidencias persistan, le envía las incidencias a la psicopedagoga. La psicopedagoga añade las incidencias en el control de incidencias, realiza el informe semanal de incidencias y se lo envía al vicedecano. El vicedecano revisa el informe, guarda el informe y toma decisiones.	

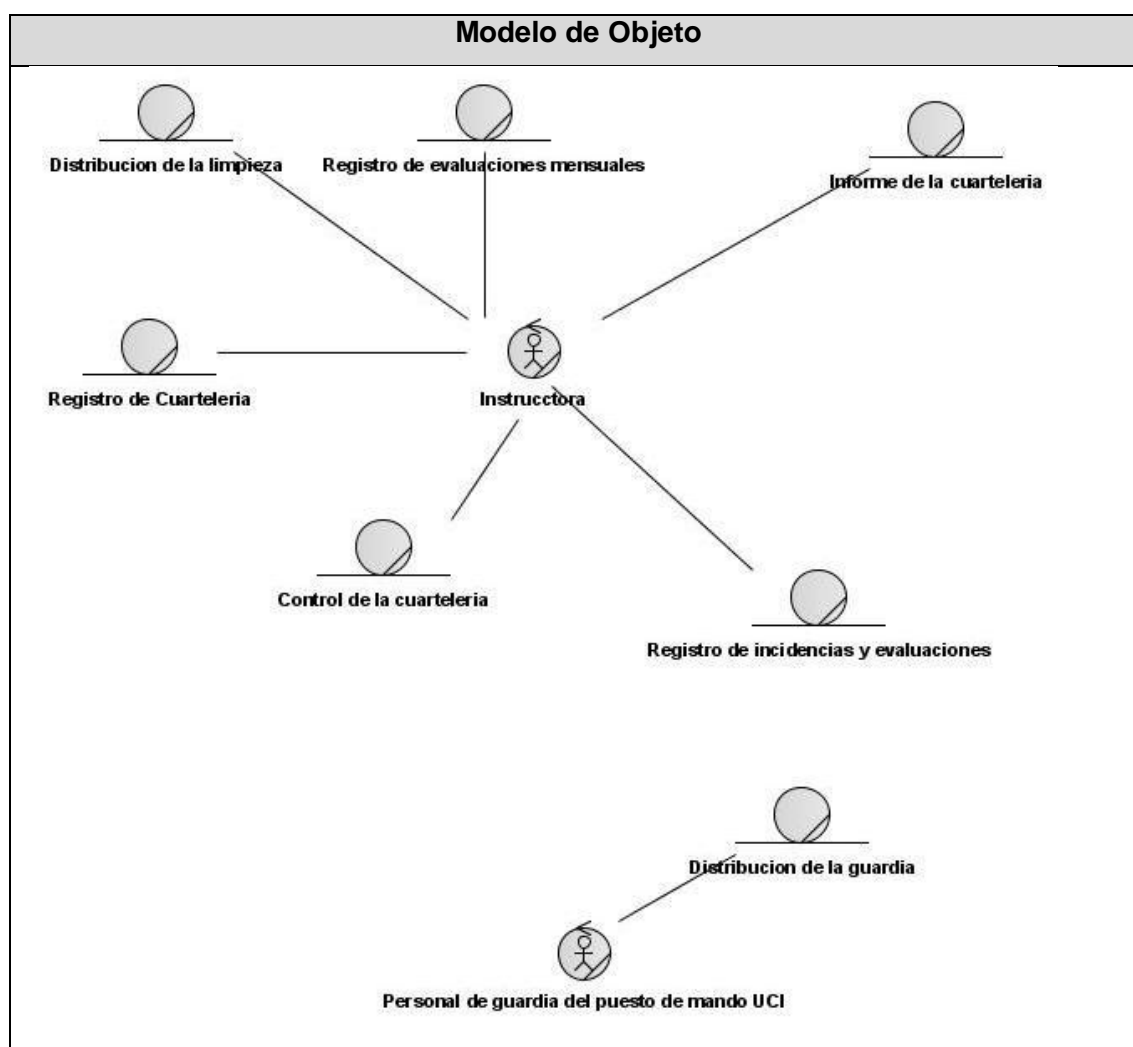




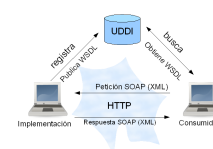
## Capítulo 2: Características del Sistema

### 2.9 Diagrama de Clases del Modelo de Objeto

El diagrama de clases del modelo de objetos es el correspondiente al negocio que se describe, en él se muestra la colaboración entre los trabajadores y entidades involucradas.



**Figura 4** Modelo de Objeto



## Capítulo 2: Características del Sistema

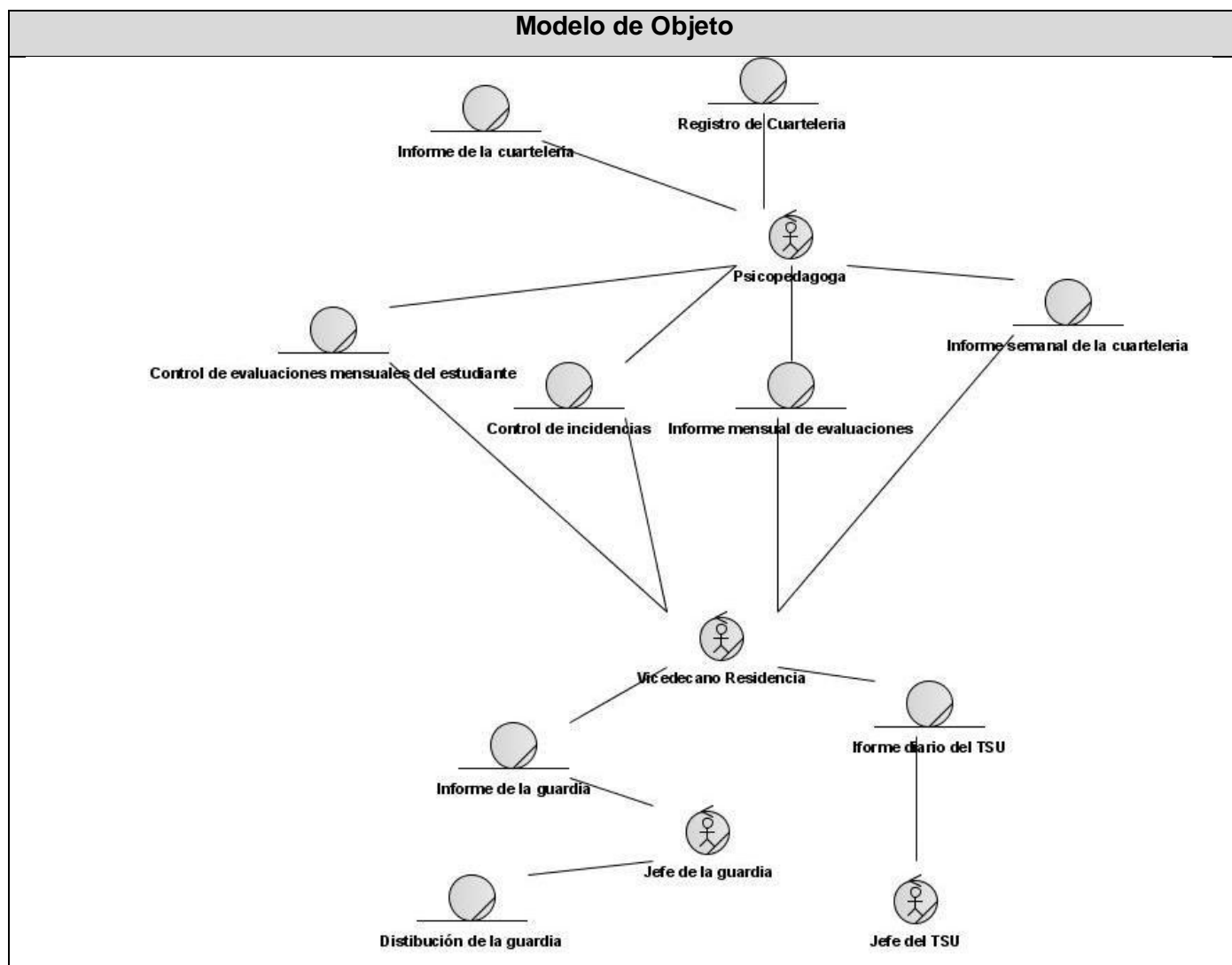
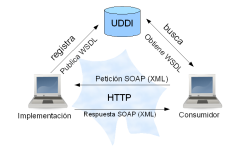


Figura 5 Modelo de Objeto [Continuación]

### 2.10 Diagrama de Actividades

Un diagrama de actividad describe un proceso que explora el orden de las tareas o actividades que logran los objetivos del negocio. **ANEXO 1**



## Capítulo 2: Características del Sistema

---

### 2.11 Requerimientos Funcionales

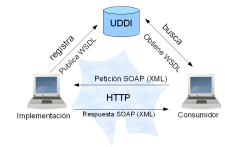
Los requerimientos funcionales son capacidades o condiciones que un sistema determinado debe cumplir. El sistema debe brindar las siguientes funcionalidades:

#### R1. Gestionar datos del estudiante

- 1.1 Permitir insertar número de solapín o nombre y apellidos del estudiante
- 1.2 Mostrar los siguientes datos del estudiante: (nombre y apellidos, carne de identidad, grupo, año, foto, número de solapín, dirección particular y teléfono, edificio, apto, Militante (UJC y/o PCC), Dirigente (FEU o UJC), teléfono de apto, proyecto en el que participa, laboratorio en el que trabaja, teléfono del laboratorio, si es alumno ayudante y asignatura que imparte en caso de ser alumno ayudante).
- 1.3 Permitir insertar los datos
- 1.4 Permitir actualizar los datos
- 1.5 Permitir eliminar los datos
- 1.6 Almacenar los datos

#### R2. Controlar cuartelería

- 2.1 Permitir insertar cuarteros los cuarteros del mes
- 2.2 Mostrar listado mensual de la cuartelería
  - 2.2.1 La lista debe estar ordenada ascendentemente por los días
  - 2.2.2 Mostrar el día y mes que le toca la cuartelería a los estudiantes.
  - 2.2.3 Señalar al cuartero que está ese día
- 2.3 Permite cambiar al estudiante que esté de cuartero ese día
  - 2.3.1 Permitir hacer cambio entre dos estudiante



## Capítulo 2: Características del Sistema

---

### 2.3.2 Permitir correr la cuartelería

2.3.2.1 El sistema debe adelantar un día a los demás estudiantes cada vez que se corra la cuartelería

2.4 Permitir insertar la evaluación del cuartelero en turno y las incidencias en caso de que existan

2.5 El sistema debe registrar la fecha de cada cuartelería

2.6 Almacenar los datos

### R3. Controlar guardia

3.1 Permitir insertar grupo a la guardia

3.2 Permitir seleccionar el grupo que está de guardia ese día

3.3 Mostrar los estudiantes que están de guardia

3.4 Permitir registrar la asistencia a la guardia.

3.5 Mostrar la asistencia a la guardia

3.5.1 El sistema de permitir configurar la fecha: (mes y año)

3.6 El sistema debe registrar la fecha de cada guardia estudiantil

3.7 Almacenar los datos

### R4. Controlar TSU

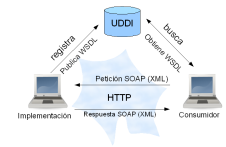
4.1 Permitir insertar el grupo al TSU

4.2 Permitir insertar la asistencia.

4.3 Mostrar asistencia al TSU

4.3.1 El sistema de permitir configurar la fecha: (mes y año)





## Capítulo 2: Características del Sistema

---

### 2.12 Requerimientos No Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

A continuación se muestran los requerimientos no funcionales:

#### Usabilidad

El uso de servicios web permite que la aplicación pueda consumir estos aunque esté escrita en una plataforma de desarrollo diferente a ellos, pues su protocolo de comunicación está basado en XML y es compatible con todas las plataformas.

El uso de los servicios web es muy práctico ya que pueden aportar gran independencia entre la aplicación que usa el servicio web y el propio servicio. De esta forma, los cambios a lo largo del tiempo en uno no deben afectar al otro.

#### Software

Se Necesita como sistema Operativo Windows 98 o superior, Linux o Unix, en sus versiones de S.O servidores.

Los servidores están desarrollados sobre una plataforma Web con lenguaje de programación PHP 5.0.

La Base de Datos desarrollada en el Sistema Gestor de BD MySQL.

La Intranet está montada sobre CMS Drupal.

#### Hardware

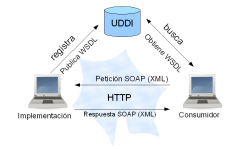
Se necesitan como requerimientos mínimos una PC con procesador Pentium IV o superior.

Requiere como mínimo RAM 256 MB.

El disco duro requiere como mínimo 5 GB para almacenar la Base de Datos.

#### Portabilidad

Los servicios web podrán ser accedidos desde cualquier protocolo de transporte que sea capaz de



## Capítulo 2: Características del Sistema

---

transportar texto, como por ejemplo (http, https, SMTP, etc.).

El diseño del servicio web permite que el mismo pueda funcionar tanto en sistemas operativos libres como propietarios.

### **Seguridad y Privacidad**

El acceso será controlado con nombres de usuario y contraseñas. Solo los usuarios con derechos de administrador podrán acceder a las funciones administrativas.

Se establecerá una llave privada entre el cliente y el proveedor del servicio web, para evitar acceso no autorizado a los servicios.

### **Confiabilidad**

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

Garantía de un tratamiento adecuado de las excepciones y validación de las entradas del usuario.

La aplicación estará disponible las 24 horas brindando así todas las funcionalidades de la misma.

### **Importación y exportación de datos**

El sistema deberá almacenar todos los datos en una base de datos MySQL Server, donde puedan ser accedidos por los servicios.

### **Restricciones del diseño y la implementación**

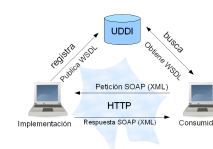
Se utiliza UML para lograr una mejor documentación del sistema y como herramienta de apoyo Visual Paradigm. Se utiliza como lenguaje de programación PHP y el gestor de base de datos MySQL.

Se usa la librería NuSoap para la creación de los servicios web.

Se usa la librería ADOdb para la abstracción a datos.

### **Legales**

El Servicio Web y toda la documentación generada pertenecen al proyecto Intranet de la Facultad 1 de la Universidad de las Ciencias Informáticas.



## Capítulo 2: Características del Sistema

### 2.13 Definición de los casos de usos del Sistema

Actores	Justificación
Intranet	La Intranet es el actor que inicia los casos de uso invocando los servicios que permiten ejecutarlos beneficiándose de ellos.

### 2.14 Casos de Usos del Sistema

Los casos de uso del sistema son las funcionalidades que el sistema brinda. En ellos se describe la secuencia determinada de eventos que realiza un actor en iteración con el sistema, en este caso es un servicio web, que se determinó de acuerdo a las funciones que pudiera brindar.

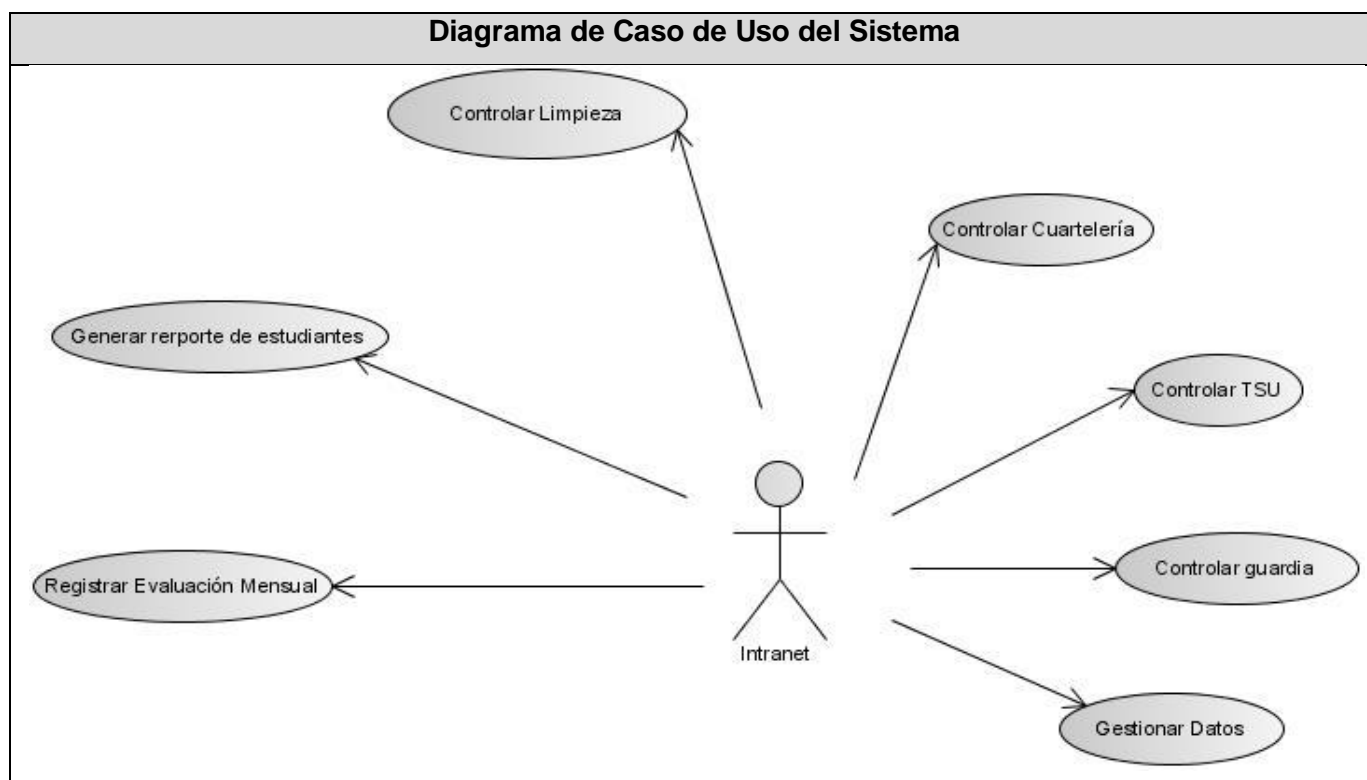
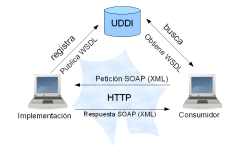


Figura 6 Diagrama de Casos de Uso del Sistema





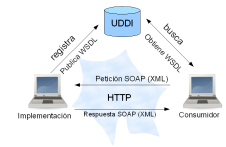
## Capítulo 2: Características del Sistema

### 2.15 Descripciones de los Casos de Uso del Sistema

A continuación se muestran las descripciones de los casos de uso del sistema donde se expresan las acciones del actor y las respuestas del sistema.

**Tabla 10** CUS Gestionar datos del estudiante

Especificación de Caso de Uso de Sistema	
<b>Identificador</b>	CUS_1
<b>Nombre</b>	Gestionar datos del estudiante
<b>Descripción</b>	El caso de uso se inicia cuando el actor invoca el servicio obtener estudiante pasándole el solapín. El sistema muestra los datos del estudiante. En caso de insertar datos, el actor invoca el servicio insertar datos del estudiante, en caso de eliminar o actualizar datos invoca el servicio eliminar estudiante y actualizar estudiante, verificando previamente que el estudiante se encuentre en el sistema, y dados estos datos actualiza los datos o elimina. Termina el caso de uso.
<b>Actor</b>	Intranet Facultad 1
<b>Prioridad de Implantación</b>	Alta
<b>Requisitos Funcionales</b>	R1.1, R1.2, R1.3, R1.4, R1.5, R1.6

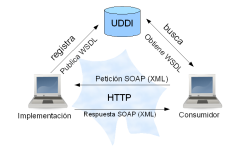


## Capítulo 2: Características del Sistema

**Tabla 11** CUS Controlar cuarterería

Especificación de Caso de Uso de Sistema	
<b>Identificador</b>	CUS_2
<b>Nombre</b>	Controlar cuarterería
<b>Descripción</b>	<p>El caso de uso se inicia cuando el actor desee insertar los cuartereros, invoca el servicio, insertar cuartereros almacenado los datos en la base de datos, en caso de que el actor desee cambiar la cuarterería, invoca el servicio cambiar cuarterería, el sistema permite hacer cambios entre dos estudiantes. En caso de que el actor desee correr la cuarterería, invoca el servicio correr cuarterería, el sistema adelanta un día la cuarterería. En caso de insertar alguna incidencia, el actor invoca el servicio insertar incidencias del cuarterero almacenando las incidencia. El actor invoca el servicio insertar evaluación para insertar la evaluación del cuarterero. El sistema almacena los datos en la base de datos En caso que el estudiante desee obtener la evaluación, invoca el servicio mostrar evaluación del cuarterero. Termina el caso de uso</p>
<b>Actor</b>	Intranet Facultad 1
<b>Prioridad de Implantación</b>	Alta
<b>Requisitos Funcionales</b>	R2.1, R2.2..., R2.3..., R2.4, R2.5, R2.6

**Tabla 12** CUS Controlar Guardia

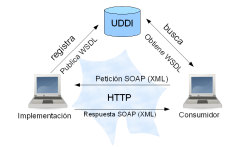


## Capítulo 2: Características del Sistema

Especificación de Caso de Uso de Sistema	
<b>Identificador</b>	CUS_3
<b>Nombre</b>	Controlar Guardia
<b>Descripción</b>	El caso de uso se inicia cuando el actor invoca el servicio insertar grupo a la guardia. En caso de que el actor desee insertar la asistencia, invoca el servicio insertar asistencia de la guardia. El sistema almacena a los estudiantes que participaron en la guardia. En caso de que el actor desee obtener la asistencia de la guardia, invoca el servicio mostrar asistencia de la guardia. El sistema muestra a los estudiantes que participaron en la guardia. Termina el caso de uso
<b>Actor</b>	Intranet Facultad 1
<b>Prioridad de Implantación</b>	Alta
<b>Requisitos Funcionales</b>	R3.1, R3.2, R3.3, R3.4, R3.5, R3.6, R3.7

Tabla 13 CUS Controlar TSU

Especificación de Caso de Uso de Sistema	
<b>Identificador</b>	CUS_4
<b>Nombre</b>	Controlar TSU
<b>Descripción</b>	El caso de uso se inicia cuando el actor invoca el servicio insertar grupo al TSU. En caso de que el actor desee insertar la asistencia, invoca el servicio insertar asistencia. El sistema almacena a los estudiantes que participaron en el TSU. En caso de que el actor



## Capítulo 2: Características del Sistema

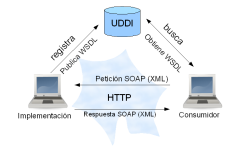
	desea obtener la asistencia del TSU, invoca el servicio mostrar asistencia. El sistema muestra a los estudiantes que participaron en el TSU. Termina el caso de uso
<b>Actor</b>	Intranet Facultad 1
<b>Prioridad de Implantación</b>	Alta
<b>Requisitos Funcionales</b>	R4.1, R4.2, R4.3, R4.4, R4.5, R4.6

**Tabla 14** CUS Controlar limpieza del Apto

Especificación de Caso de Uso de Sistema	
<b>Identificador</b>	CUS_5
<b>Nombre</b>	Controlar limpieza del Apto
<b>Descripción</b>	El caso de uso se inicia cuando el actor desea controlar la limpieza del apto. En caso de que el actor desea insertar alguna incidencia, invoca el servicio insertar incidencia. El sistema inserta las incidencias en la base de datos. En caso de que el actor desea eliminar alguna incidencia, invoca el servicio eliminar incidencias. El sistema elimina la incidencia de la base de datos. Termina el caso
<b>Actor</b>	Intranet Facultad 1
<b>Prioridad de Implantación</b>	Alta
<b>Requisitos Funcionales</b>	R5.1, R5.2, R5.3, R5.4

**Tabla 15** CUS Generar reporte de estudiante

Especificación de Caso de Uso de Sistema
--

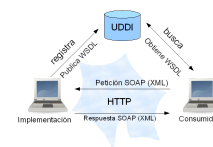


## Capítulo 2: Características del Sistema

<b>Identificador</b>	CUS_6
<b>Nombre</b>	Generar reporte de estudiante
<b>Descripción</b>	El caso de uso se inicia cuando el actor desee obtener algún reporte del estudiante. En caso de que el actor desee obtener estudiantes por categoría, invoca el servicio mostrar estudiante por categoría. El sistema muestra un listado de estudiantes según la categoría. En caso de que desee obtener los estudiantes por apartamentos o por edificio, invoca el servicio obtener estudiantes, pasándole el apartamento o el edificio. El sistema muestra un listado de estudiantes. En caso de que el actor desee obtener a los estudiantes con incidencias, invoca el servicio mostrar estudiantes incidentes. El sistema muestra a los estudiantes con incidencias. En caso de que el actor desee obtener las incidencias de un estudiante seleccionado, invoca el servicio. En caso de que el actor desee obtener los estudiantes por año, invoca el servicio mostrar estudiantes por año. El sistema muestra a los estudiantes Termina el caso de uso
<b>Actor</b>	Intranet Facultad 1
<b>Prioridad de Implantación</b>	Media
<b>Requisitos Funcionales</b>	R6.1, R6.2, R6.3, R6.4, R6.5, R6.6

**Tabla 16 CUS** Registrar evaluación mensual

Especificación de Caso de Uso de Sistema	
<b>Identificador</b>	CUS_1



## Capítulo 2: Características del Sistema

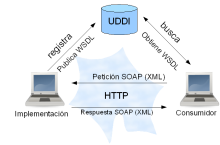
<b>Nombre</b>	Registrar evaluación mensual
<b>Descripción</b>	El caso de uso se inicia cuando el actor invoca el servicio insertar evaluación mensual, el sistema almacena la evaluación. En caso de obtener la evaluación, invoca le servicio mostrar evaluación mensual Termina el caso de uso
<b>Actor</b>	Intranet Facultad 1
<b>Prioridad de Implantación</b>	Alta
<b>Requisitos Funcionales</b>	R7.1, R7.2..., R7.3

### 2.16 Casos de Usos del Sistema Expandidos

En el **ANEXO 2** se encuentra las descripciones de los casos de uso del sistema que pertenecen al ciclo de desarrollo especificado

### Conclusiones

En este capítulo se empezó a profundizar en el desarrollo de la propuesta de solución, se describió los requisitos con los que deben cumplir los servicios. La elaboración del modelo de negocio ha permitido un mejor estudio del entorno en que se va a desarrollar el sistema. El diagrama de caso de uso del sistema, muestra de manera gráfica y detallada las funcionalidades que va a brindar el servicio web.



## CAPÍTULO 3: ANÁLISIS Y DISEÑO

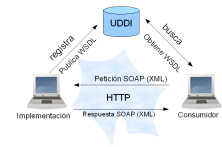
### Introducción

En el presente capítulo se pretende realizar el Modelo de Análisis y Diseño del servicio web propuesto. Este modelo establece la realización de los casos de uso en clases, pasando desde una representación en términos del análisis (diagrama de clases del análisis) hacia un modelo del diseño, en el cual se refleja una vista interna de lo que sería el sistema, representando los diagramas de secuencias, los diagramas de clases con sus descripciones, el diseño de la base de datos a través del Modelo de Datos, con las descripciones de las tablas correspondientes y se realizará la estimación de costo del proyecto para determinar la factibilidad y rentabilidad del proyecto.

### 3.1 Modelo de Análisis

En el modelo de análisis no se toma en cuenta el lenguaje de programación que se va utilizar en el desarrollo de la aplicación ni otros aspectos como la plataforma. El objetivo general del análisis es comprender los requisitos del software y no precisar cómo se realizará la implementación.

El modelo del análisis se considera como la aproximación al Modelo de Diseño y se describe en un lenguaje comprensible para los desarrolladores; se compone además por clases del análisis y sus objetos organizados en paquetes que colaboran entre sí.



## Capítulo 3: Análisis y Diseño

### 3.2 Diagrama de Clases del Análisis

El diagrama de clases del análisis es un artefacto en el que se representan los conceptos del dominio de un problema y que representa las cosas del mundo real. Está compuesto por clases del análisis y sus relaciones. Las clases del análisis están centradas en los requisitos funcionales y son evidentes en el dominio del problema porque representan conceptos y relaciones del dominio. Dichas clases tienen atributos y existen relaciones de asociación, agregación / composición, generalización / especialización y tipos asociativos entre ellas. Estas clases se clasifican en:

- **Entidad:** Modelan información que posee larga vida y que es a menudo persistente.
- **Interfaz:** Modelan la interacción entre el sistema y sus actores.
- **Control:** Coordinan la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso

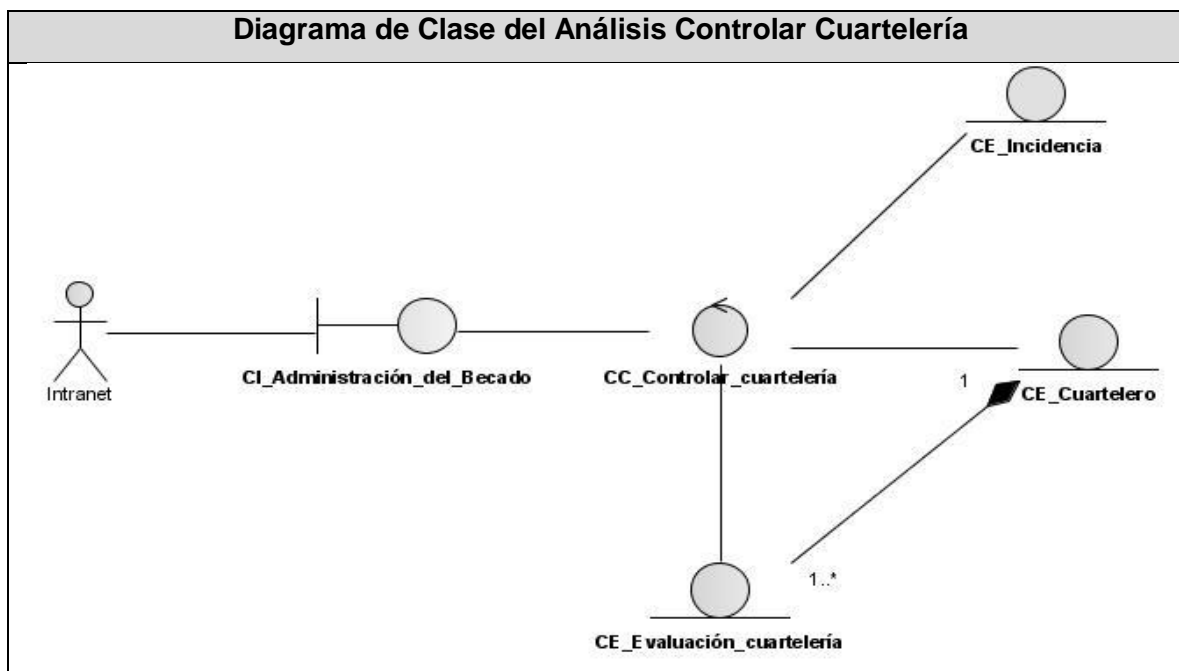
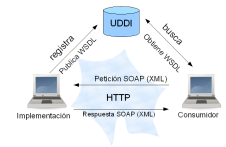
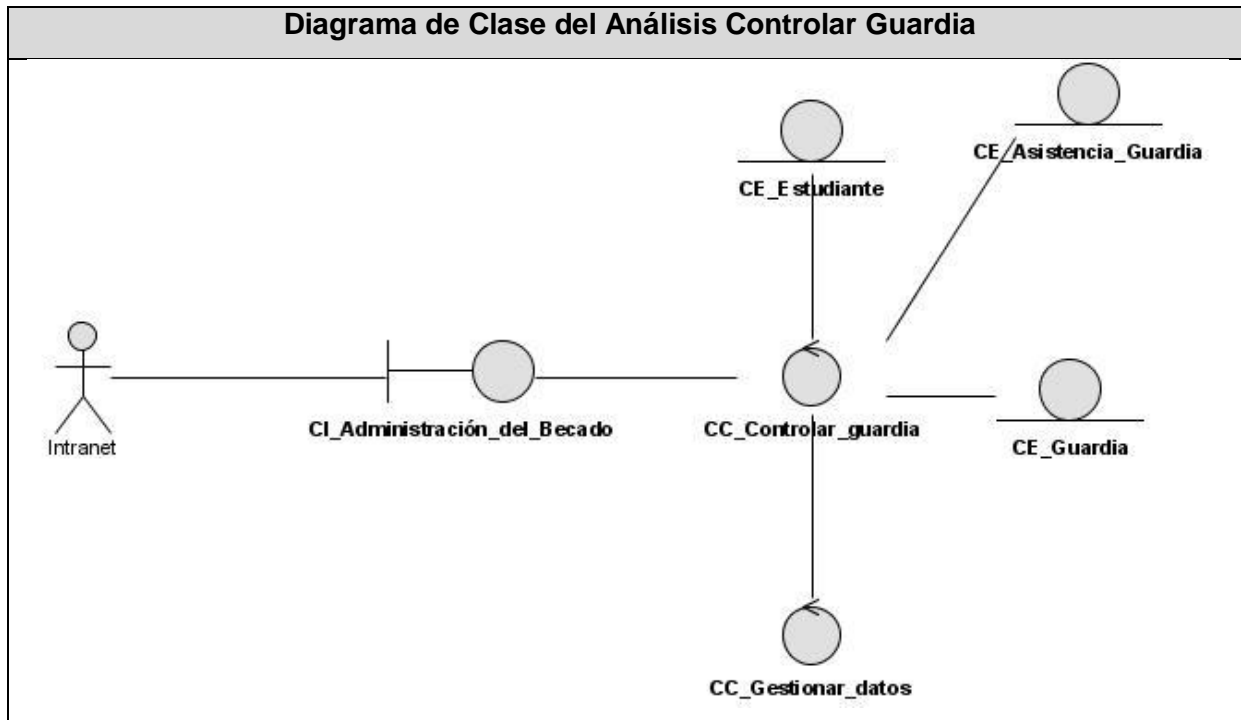


Figura 7 DCA Controlar Cuarterería



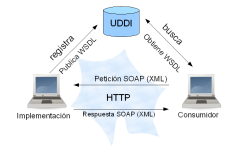


## Capítulo 3: Análisis y Diseño



**Figura 8** DCA Controlar Guardia





## Capítulo 3: Análisis y Diseño

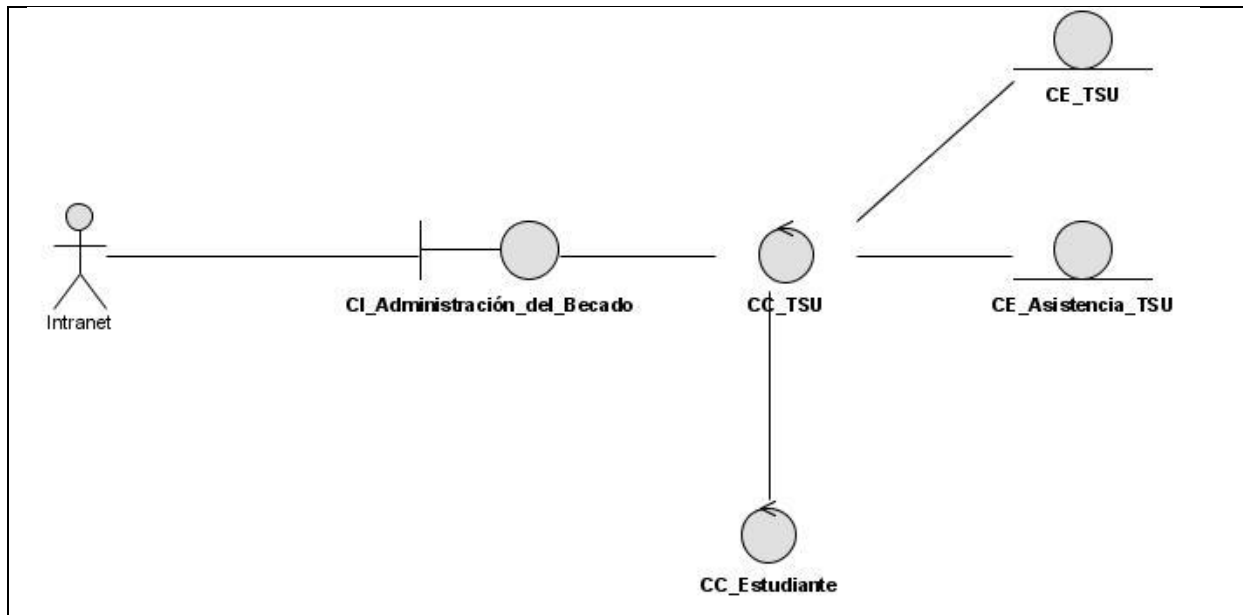


Figura 9 DCA Control TSU

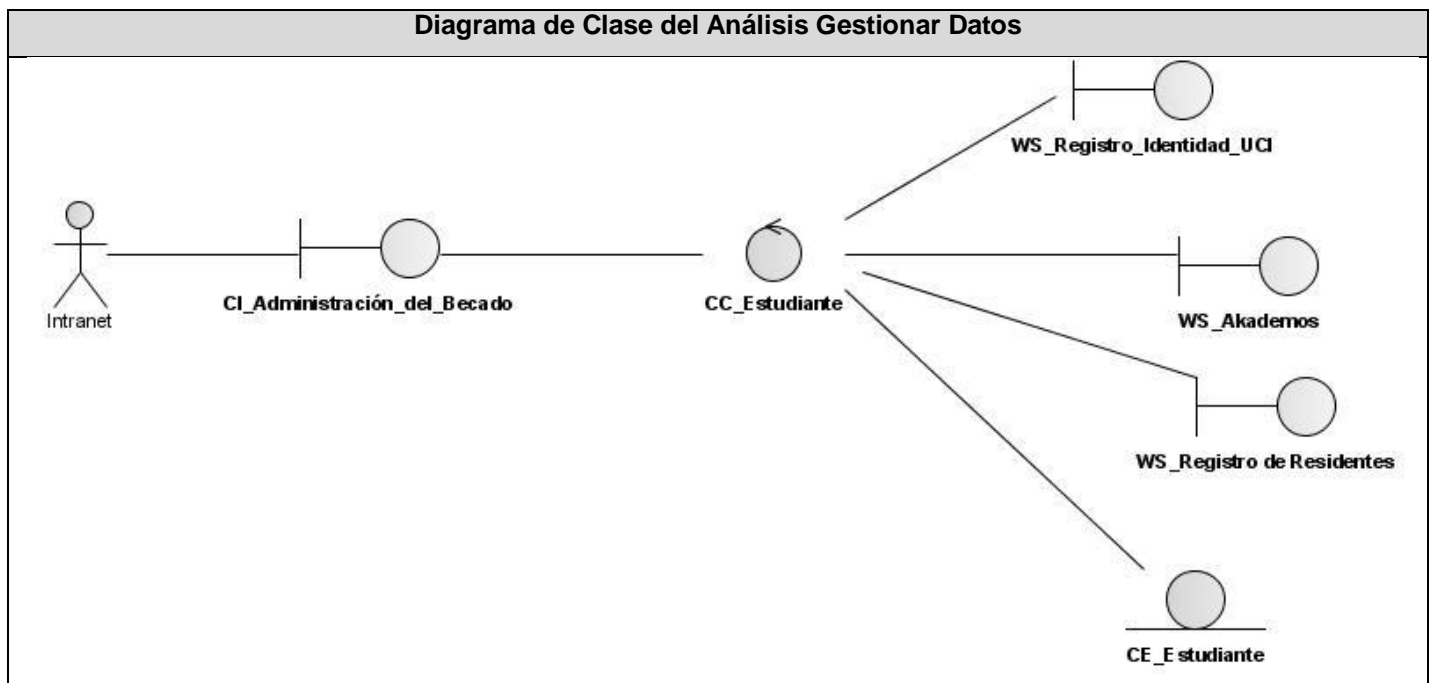
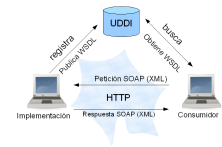
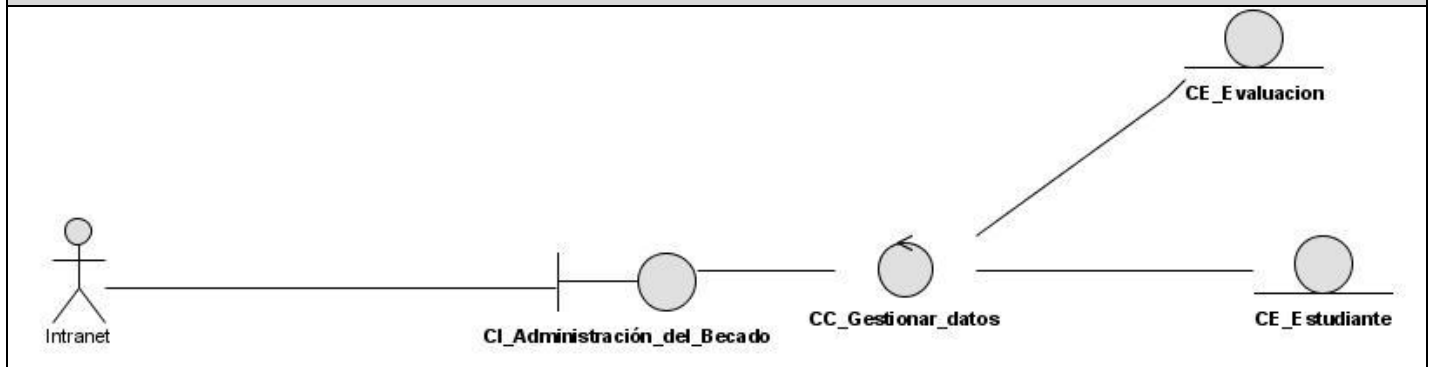


Figura 10 DCA Gestionar Datos



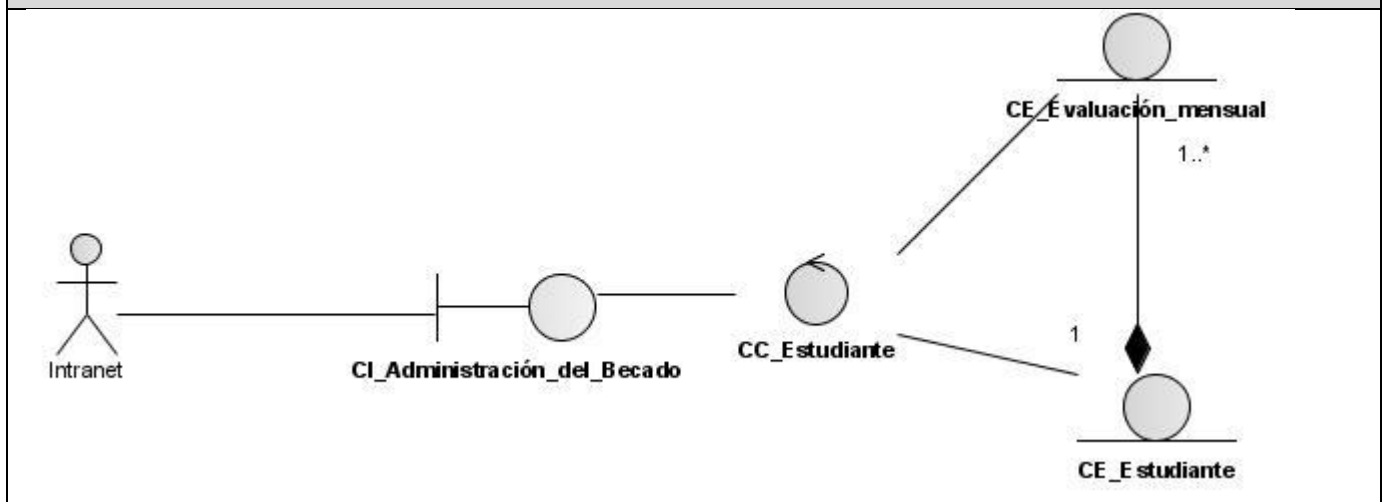
## Capítulo 3: Análisis y Diseño

**Diagrama de Clase del Análisis Generar Reporte de Estudiante**



**Figura 11 DCA Generar Reporte de Estudiante**

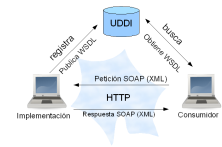
**Diagrama de Clase del Análisis Registrar Evaluación Mensual**



**Figura 12 DCA Registrar Evaluación Mensual**

### 3.2 Modelo de Diseño

El Modelo de Diseño es un modelo de objetos que describe la realización física de los casos de uso, centrándose en cómo los requisitos, junto a otras restricciones, tienen impacto en el sistema a considerar. La



## Capítulo 3: Análisis y Diseño

---

esencia del Diseño es la elaboración de los diagramas de interacción, que describen cómo los objetos se comunican entre ellos para dar cumplimiento a los requerimientos que fueron definidos y permiten la realización de los diagramas de clases del diseño, que contienen las clases que se pueden implementar en un software.

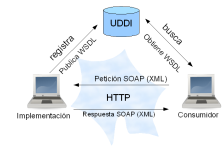
### 3.3 Diagramas de Interacción

Los diagramas de interacción consisten en un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos. Estos diagramas se dividen en dos tipos de diagramas de UML: los diagramas de secuencia y los diagramas de colaboración. Para modelar los aspectos dinámicos de los servicios se utilizaron diagramas de secuencia por cada caso de uso. Ver **ANEXO 3**

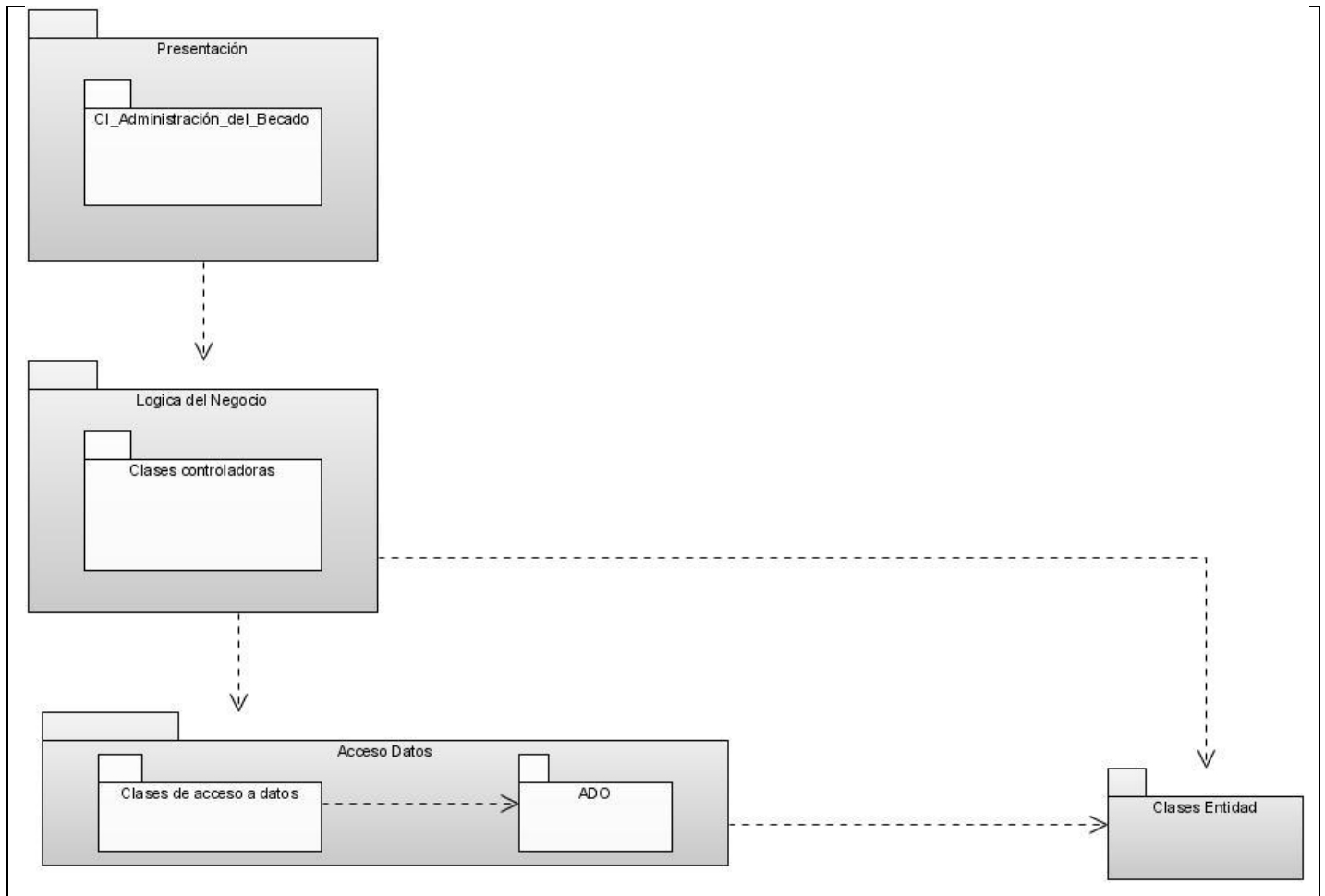
### 3.3 Diagramas de Clases del Diseño

Para entender mejor el diagrama de clase, el diagrama se separó en 4 paquetes según las funcionalidades, estos paquetes a la vez se han dividido en otros subpaquetes para una mejor organización y legibilidad.

#### Diagrama de Clases en Paquetes

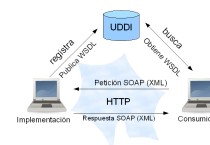


## Capítulo 3: Análisis y Diseño



**Figura 13** Diagrama de Clases en Paquetes

En el paquete **Presentación** se encuentra la clase **interfaz** del servicio web, donde están registrados todos los servicios que se expondrán en la WSDL (Lenguaje de descripción del servicio web). El paquete **Lógica del Negocio**, se encuentra la lógica del negocio del servicio web. En este paquete se encuentra el paquete de **clases Controladoras**: estas son las clases controladoras de las entidades relacionadas en el negocio, estas clases permiten diversas funciones como la validación de los datos y algunas consumen servicios, de otros servicios web externos. El paquete de **Acceso a Datos**, contiene las clases para hacer posible la persistencia y recuperación de objetos. Este paquete está dividido en dos subpaquetes: el



## Capítulo 3: Análisis y Diseño

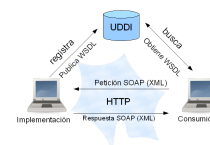
subpaquete **Clases de Acceso a Datos**, contiene las clases encargadas de acceder a la base de datos para manipular la persistencia de las entidades. El subpaquete **ADO**, utilizado por el paquete anterior, es donde se encuentran un conjunto de clases que permiten conectarse a diferentes bases de datos. El paquete de acceso a datos permite en general a la aplicación abstraerse del origen de los datos y la lógica de su persistencia, logrando un bajo acoplamiento entre sus componentes. El paquete Entidad contiene las clases que no tienen comportamiento, solo propiedades y son representaciones de entidades del dominio, son clases persistentes que son accedidas por las clases los paquetes **Lógica del Negocio** y **Acceso a Datos**.

En el **ANEXO 4** se muestra cada uno de los diagramas de clases correspondiente a cada paquete del diagrama general representado anteriormente.

### 3.4 Descripción de las clases

Tabla 17 CE\_Estudiante

Nombre: CE_Estudiante	
Tipo de clase: Entidad	
Atributo	Tipo
id_estud	var
telef_apto	var
nom_proye	var
num_lab	var
telef_lab	var
a_ayudante	var
asignatura	var
solapin	var
nom_completo	var
Obj_tipo_militante	var
Obj_tipo_dirigente	var
edificio	var
apto	var

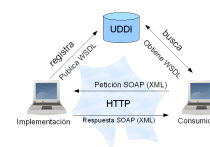


## Capítulo 3: Análisis y Diseño

<b>Responsabilidades</b>
Get_Id_estud()
<b>Descripción:</b> Responsable de devolver el id del estudiante
Get_telef_apto()
<b>Descripción:</b> Responsable de devolver el teléfono del apartamento donde vive el estudiante
Get_nom_proye()
<b>Descripción:</b> Responsable de devolver el proyecto al que pertenece el estudiante
Get_num_lab()
<b>Descripción:</b> Responsable de devolver el laboratorio donde trabaja el estudiante
Get_telef_lab()
<b>Descripción:</b> Responsable de devolver el teléfono del laboratorio donde trabaja el estudiante
Get_a_ayudante()
<b>Descripción:</b> Responsable de devolver si el estudiante es alumno ayudante
Get_asignatura()
<b>Descripción:</b> Responsable de devolver la asignatura que imparte el estudiante
Get_solapin()
<b>Descripción:</b> Responsable de devolver el solapín del estudiante
Get_nom_completo()
<b>Descripción:</b> Responsable de devolver nombre y apellido del estudiante
Get_edificio()
<b>Descripción:</b> Responsable de devolver el edificio donde vive el estudiante
Get_apto()
<b>Descripción:</b> Responsable de devolver el apartamento donde vive el estudiante

Tabla 18 CE\_Incidencia

<b>TablaNombre: CE_Incidencia</b>	
<b>Tipo de clase: Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
id_incidencia	var
descrip	var
fecha	var
Obj_estudiante	var
<b>Responsabilidades</b>	
Get_id_incidencia()	
<b>Descripción:</b> Responsable de devolver el id de la incidencia	
Get_descrip()	
<b>Descripción:</b> Responsable de devolver la descripción de la incidencia	



## Capítulo 3: Análisis y Diseño

Get_fcha()
<b>Descripción:</b> Responsable de devolver la fecha de la incidencia

Tabla 19 CE\_Evaluación\_mensual

<b>Nombre: CE_Evaluación_mensual</b>	
<b>Tipo de clase: Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
id_eval	var
fecha_eval	var
Obj_tipo_evalucion	var
Obj_estudiante	var
<b>Responsabilidades</b>	
Get_Id_eval()	
<b>Descripción:</b> Responsable de devolver el id de la evaluación	
Get_fecha_eval ()	
<b>Descripción:</b> Responsable de devolver la fecha de la evaluación	

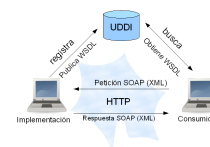
Tabla 20 CE\_Evaluación\_cuartelería

<b>Nombre: CE_Evaluación_cuartelería</b>	
<b>Tipo de clase: Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
id_eval	var
fecha_eval	var
Obj_tipo_evalucion	var
Obj_cuartelero	var
<b>Responsabilidades</b>	
Get_Id_eval()	
<b>Descripción:</b> Responsable de devolver el id de la evaluación	
Get_fecha_eval ()	
<b>Descripción:</b> Responsable de devolver la fecha de la evaluación	

Tabla 21 CE\_TSU

<b>Nombre: CE_TSU</b>	
<b>Tipo de clase: Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
id_tsu	var
fecha_tsu	var





## Capítulo 3: Análisis y Diseño

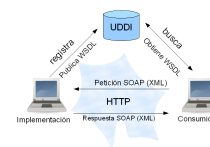
grupo	var
Obj_tipo_sesion	var
<b>Responsabilidades</b>	
Get_id_tsu ()	
<b>Descripción:</b> Responsable de devolver el id del TSU	
Get_fecha_tsu ()	
<b>Descripción:</b> Responsable de devolver la fecha del TSU	
Get_grupo ()	
<b>Descripción:</b> Responsable de devolver el grupo que participó en el tsu	

Tabla 22 CE Guardia

<b>Nombre: CE_Guardia</b>	
<b>Tipo de clase: Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
id_guardia	var
fecha_guar	var
grupo	var
<b>Responsabilidades</b>	
Get_id_guardia ()	
<b>Descripción:</b> Responsable de devolver el id de la guardia	
Get_fecha_guar ()	
<b>Descripción:</b> Responsable de devolver la fecha de la guardia	
Get_asistencia ()	
<b>Descripción:</b> Responsable de devolver la asistencia a la guardia	
Get_grupo ()	
<b>Descripción:</b> Responsable de devolver el grupo que participó en la guardia	

Tabla 23 CE\_Asiencia\_guardia

<b>Nombre: CE_Asiencia_guardia</b>	
<b>Tipo de clase: Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
solapin	var
Obj_guardia	var
<b>Responsabilidades</b>	
Get_solapin()	
<b>Descripción:</b> Responsable de devolver el solapín del estudiante que participó en la guardia	



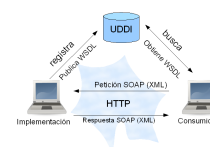
## Capítulo 3: Análisis y Diseño

Tabla 24 CE\_Asiistencia\_tsu

<b>Nombre: CE_Asiistencia_tsu</b>	
<b>Tipo de clase: Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
solapin	var
Obj_tsu	var
<b>Responsabilidades</b>	
Get_solapin()	
<b>Descripción:</b> Responsable de devolver el solapín del estudiante que participó en el tsu	

Tabla 25 CC\_Estudiante

<b>Nombre: CC_Estudiante</b>	
<b>Tipo de clase: Clase controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
-----	-----
<b>Responsabilidades</b>	
Inserta_Estudiante(datos: var)	
<b>Descripción:</b> Responsable de insertar un estudiante si el dato enviado es correcto	
Elimina_Estudiante(id_estud: var)	
<b>Descripción:</b> Responsable de eliminar un estudiante si el dato enviado es correcto	
Actualiza_Datos(datos: var)	
<b>Descripción:</b> Responsable de actualizar los datos del estudiante si los datos enviados son correctos	
Obtiene_Estudiantes(apto: var, edificio: var, nombre: var)	
<b>Descripción:</b> Responsable de buscar los estudiantes si los datos enviados son correctos	
Inserta_Evaluacion(tipo_eval: var, fecha: var)	
<b>Descripción:</b> Responsable de insertar la evaluación si los datos enviados son correctos	
Obtinene_Estudiantes_Categoria(categoría: var, localidad: var)	
<b>Descripción:</b> Responsable de buscar estudiantes si los datos enviados son correctos	
Busca_Datos_Estudiante(solapin: var)	
<b>Descripción:</b> Responsable de buscar los datos de un estudiante si los datos enviados son correctos	
Obtiene_Evaluacion(solapin: var, fecha: var)	
<b>Descripción:</b> Responsable de devolver la evaluación del estudiante si los datos enviados son correctos	

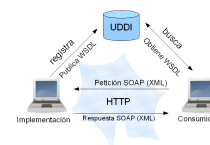


## Capítulo 3: Análisis y Diseño

Validar_Negocio_InsertaEstudiante(datos: var)
<b>Descripción:</b> Responsable de validar que los datos para insertar son correctos
Validar_Negocio_EliminaEstudiante(id_estud: var)
<b>Descripción:</b> Responsable de validar de que los datos para eliminar son correctos
Validar_Negocio_ActualizaDatos(datos: var)
<b>Descripción:</b> Responsable de validar que los datos para actualizar sean correctos
Validar_Negocio_BuscaEstudiante(solapin: var)
<b>Descripción:</b> Responsable de validar que el dato para buscar los datos del estudiante sea correcto
Validar_Negocio_ObtieneEstudiantes(apto: var, edificio: var, nombre: var)
<b>Descripción:</b> Responsable de validar que los datos para buscar los estudiantes sean correctos
Validar_Negocio_ObtieneEstudCateg(categoría: var, localidad: var)
<b>Descripción:</b> Responsable de validar que los datos para buscar estudiantes sean correctos
Validar_Negocio_InsertaEval(tipo_eval: var, fecha: var)
<b>Descripción:</b> Responsable de validar que los datos para insertar la evaluación sean correctos
Validar_Negocio_ObtieneEval(solapin: var, fecha: var)
<b>Descripción:</b> Responsable de validar que los datos para devolver la evaluación del estudiante sean correctos

Tabla 26 CC\_Controlar\_cuartelería

<b>Nombre: CC_Controlar_cuartelería</b>	
<b>Tipo de clase: Clase controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
-----	-----
<b>Responsabilidades</b>	
Inserta_Evaluacion(eval: var)	
<b>Descripción:</b> Responsable de Insertar la evaluación si el dato enviado es correcto	
Inserta_Incidencia(array_incidencia: var)	
<b>Descripción:</b> Responsable de insertar incidencia si los datos enviados son correctos	
Listar_Cuarteleros(edificio: var, paso_escalera: var)	
<b>Descripción:</b> Responsable de buscar los cuarteros si los datos enviados son correctos	
Cambia_Cuarteleria(id_cuart1, id_cuart2)	
<b>Descripción:</b> Responsable de cambiar la cuartería si los datos enviados son correctos	

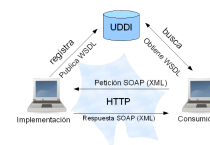


## Capítulo 3: Análisis y Diseño

Obtiene_Cuartelero(edificio: var, paso_escalera: var, fecha: var)
<b>Descripción:</b> Responsable de buscar un cuartelero si los datos enviados son correctos
Busca_Estudiantes(apto: var)
<b>Descripción:</b> Responsable de buscar estudiantes si el dato enviado es correcto
Corre_Cuarteleria(edificio: var, paso_escalera: var)
<b>Descripción:</b> Responsable de correr la cuarterería si los datos enviados son correctos
Validar_Negocio_Cambia_Cuartelería(id_cuart1: var, id_cuart2: var)
<b>Descripción:</b> Responsable de validar que los datos para cambiar la cuarterería sean correctos
Validar_Negocio_ListarCuarteleros(edificio: var, paso_escalera: var)
<b>Descripción:</b> Responsable de validar que los datos para listar cuartereros sean correctos
Valida_Negocio_BuscaEstudiante(apto: var)
<b>Descripción:</b> Responsable de validar que el dato para buscar los estudiantes sea correcto
Validar_Negocio_InsertaEvaluación(eval: var)
<b>Descripción:</b> Responsable de validar que el dato para insertar la evaluación sea correcto
Validar_Negocio_InsertarIncidencias(array_incidencias: var)
<b>Descripción:</b> Responsable de validar que los datos para insertar incidencias sean correctos
Validar_Negocio_ObtieneCuartelero(edificio: var, paso_escalera: var, fecha: var)
<b>Descripción:</b> Responsable de validar que los datos para buscar un cuartelero sean correctos
Validar_Negocio_CorreCuarteleria(edificio: var, paso_escalera: var)
<b>Descripción:</b> Responsable de validar que los datos para correr la cuarterería sean correctos

Tabla 27 CC\_Controlar\_TSU

<b>Nombre: CC_Controlar_TSU</b>	
<b>Tipo de clase: Clase controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
-----	-----
<b>Responsabilidades</b>	
Busca_Estudiantes(grupo: var)	
<b>Descripción:</b> Responsable de buscar estudiantes si el dato enviado es correcto	

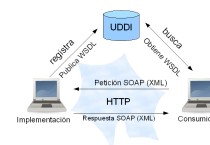


## Capítulo 3: Análisis y Diseño

Inserta_Grupo(grupo: var, sesion: var, fecha: var)
<b>Descripción:</b> Responsable de insertar un grupo si los datos enviados son correctos
Insertar_Asiistencia(id_tsu: var, array_id_estud: var)
<b>Descripción:</b> Responsable de insertar la asistencia si los datos enviados son correctos
Obtiene_TSU(fecha: var, sesión: var)
<b>Descripción:</b> Responsable de buscar el grupo que participó en el tsu si los datos enviados son correctos
Obtiene_Asiistencia(id_tsu: var)
<b>Descripción:</b> Responsable de mostrar la asistencia del tsu si el dato enviado es correcto
Validar_Negocio_InsertarAsistencia(id_tsu: var, id_estud: var)
<b>Descripción:</b> Responsable de validar que los datos para insertar la asistencia sean correctos
Validar_Negocio_BusacaEstudiante(grupo: var)
<b>Descripción:</b> Responsable de validar que el dato para buscar estudiante sea correcto
Validar_Negocio_InsertaGrupo(grupo: var, sesion: var, fecha: var)
<b>Descripción:</b> Responsable de validar que los datos para insertar el grupo sean correctos
Validar_Negocio_ObtieneTSU(fecha: var, sesión: var)
<b>Descripción:</b> Responsable de validar que los datos para obtener el grupo que participó en tsu sean correctos
Validar_Negocio_ObtieneAsistencia(id_tsu)
<b>Descripción:</b> Responsable de validar que los datos para obtener la asistencia del tsu sean correctos

**Tabla 28** CC\_Controlar\_limpieza

<b>Nombre: CC_Controlar_limpieza</b>	
<b>Tipo de clase: Clase controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
-----	-----
<b>Responsabilidades</b>	
Inserta_Incidencias(id_estud: var, inc: var)	
<b>Descripción:</b> Responsable de insertar incidencia del estudiante si los datos enviados son correctos	
Obtiene_Estudiante_Incidentes(fecha: var)	
<b>Descripción:</b> Responsable de obtener a los estudiantes con incidencia si los datos enviados son correctos	
Elimina_Incidencia(id_inc: var)	
<b>Descripción:</b> Responsable de eliminar las incidencia del estudiante si los datos	

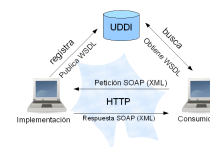


## Capítulo 3: Análisis y Diseño

enviados son correctos
<b>Descripción:</b> Responsable de obtener las incidencia del estudiante si los datos enviados son correctos
Muestra_Incidencia(id_estud: var)
<b>Descripción:</b> Responsable de mostrar las incidencia de un estudiante si el dato enviado son correctos
Validar_Negocio_InsertaIncidencias(array_inc: var)
<b>Descripción:</b> Responsable de validar que los datos para insertar incidencias sean correctos
Validar_Negocio_ObtieneElIncidentes(fecha: var)
<b>Descripción:</b> Responsable de validar que el dato para obtener los estudiantes con incidencias sea correcto
Validar_Negocio_EliminaIncidencia(id_inc: var)
<b>Descripción:</b> Responsable de validar que el dato para eliminar las incidencias sea correcto
Validar_Negocio_MuestraIncidentes(id_estud: var)
<b>Descripción:</b> Responsable de validar que el dato para mostrar la incidencia del estudiante sea correcto

**Tabla 29** CC\_Controlar\_guardia

<b>Nombre: CC_Controlar_guardia</b>	
<b>Tipo de clase: Clase controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
-----	-----
<b>Responsabilidades</b>	
Inserta_Grupo(grupo: var, fecha: var)	
<b>Descripción:</b> Responsable de insertar un grupo docente si los datos enviados son correctos	
Obtiene_Guardia(fecha: var)	
<b>Descripción:</b> Responsable de obtener el grupo docente de guardia si el dato enviado es correcto	
Obtiene_Estudiantes(grupo: var)	
<b>Descripción:</b> Responsable de obtener los estudiantes de la guardia si el dato enviados es correcto	
Inserta_Asiencia(id_guard: var, array_id_estud: var)	
<b>Descripción:</b> Responsable de insertar la asistencia de la guardia si los datos enviados son correctos	
Validar_Negocio_InsertaGrupo(grupo: var, fecha: var)	
<b>Descripción:</b> Responsable de validar que los datos para insertar el grupo sean	



## Capítulo 3: Análisis y Diseño

correctos
Validar_Negocio_InsertaAsistencia(id_guard: var, array_id_estud: var)
<b>Descripción:</b> Responsable de validar que los datos para insertar la asistencia sean correctos
Validar_Negocio_ObtenerGuardia(fecha: var)
<b>Descripción:</b> Responsable de validar que el dato para obtener el grupo docente de guardia sea correcto
Validar_Negocio_ObtieneEstud(grupo: var)
<b>Descripción:</b> Responsable de validar que el dato para obtener los estudiantes de guardia sea correcto

**Tabla 30** ado\_Estudiante

<b>Nombre: ado_Estudiante</b>	
<b>Tipo de clase: Clase controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
-----	-----
<b>Responsabilidades</b>	
dao_Gestionar_datos()	
<b>Descripción:</b> Constructor de la clase	
Insertar_Estudiante(estudiante: var)	
<b>Descripción:</b> Responsable de insertar un estudiante dado un objeto del mismo tipo pasado por parámetro	
Eliminar_Estudiante(id_estud: var)	
<b>Descripción:</b> Responsable de eliminar los datos de un estudiante pasando el id del estudiante y los datos a eliminar por parámetro	
Actualizar_Estudiante(estudiante: var)	
<b>Descripción:</b> Responsable de actualizar un estudiante dado un objeto del mismo tipo pasado por parámetro.	
Get_Datos_Estudiante(solapin: var)	
<b>Descripción:</b> Responsable de devolver los datos de un estudiante pasando el solapín por parámetro	
Get_Solapin(id_estud: var)	
<b>Descripción:</b> Responsable de devolver un solapín pasando el id del estudiante como parámetro	
Get_Estudiantes(apto: var, edificio: var, nombre: var, grupo: var)	
<b>Descripción:</b> Responsable de devolver un listado de estudiantes dado cualquiera de estos datos pasados como parámetros	
Get_Estudiante(solapin: var)	



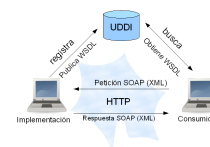
## Capítulo 3: Análisis y Diseño

<b>Descripción:</b> Responsable de devolver un estudiante pasando como parámetro el solapín
Get_Estudiante_Categoria(categoria: var, localidad: var)
<b>Descripción:</b> Responsable de devolver un listado de estudiante pasando como parámetro la categoría y la localidad (apto ó edificio)
Existe(solapin: var)
<b>Descripción:</b> Responsable de verificar que el solapín pasado por parámetro exista
Get_Evaluacion(solapin: var, fecha: var)
<b>Descripción:</b> Responsable de devolver la evaluación del estudiante pasando como parámetro el solapín y la fecha
Insertar_Evaluacion(evaluacion: var)
<b>Descripción:</b> Responsable de insertar la evaluación del estudiante pasando como parámetro la evaluación

Tabla 31 ado\_TSU

<b>Nombre: ado_TSU</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
-----	-----
<b>Responsabilidades</b>	
dao_TSU()	
<b>Descripción:</b> Constructor de la clase	
Insertar_TSU(tsu: var)	
<b>Descripción:</b> Responsable de insertar un tsu dado un objeto del mismo tipo pasado por parámetro	
Insertar_Asistencia(asistencia: var)	
<b>Descripción:</b> Responsable de insertar la asistencia al TSU pasando la asistencia del tsu como parámetro	
Get_Grupo(fecha: var, sesion: var)	
<b>Descripción:</b> Responsable de devolver el grupo que participó en el TSU pasando la fecha y la sesión como parámetro	
Get_Asistencia(id_tsu: var)	
<b>Descripción:</b> Responsable de mostrar la asistencia del TSU pasando el id del tsu como parámetro	





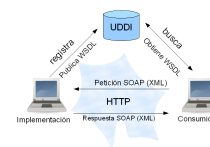
## Capítulo 3: Análisis y Diseño

**Tabla 32** ado\_Controlar\_guardia

<b>Nombre: ado_Controlar_guardia</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
-----	-----
<b>Responsabilidades</b>	
dao_Guardia()	
<b>Descripción:</b> Constructor de la clase	
Insertar_Guardia(guardia: var)	
<b>Descripción:</b> Responsable de insertar la guardia dado un objeto del mismo tipo pasado por parámetro	
Get_Asiistencia(id_guard: var)	
<b>Descripción:</b> Responsable de mostrar la asistencia de la guardia pasando el id de la guardia como parámetro	
Insertar_Asiistencia(asistencia: var)	
<b>Descripción:</b> Responsable de insertar la asistencia de la guardia pasando un objeto asistencia por parámetro	
Get_Datos_Guardia(fecha: var)	
<b>Descripción:</b> Responsable de devolver un objeto guardia, pasando la fecha como parámetro	

**Tabla 33** ado\_Controlar\_limpieza

<b>Nombre: ado_Controlar_limpieza</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
-----	-----
<b>Responsabilidades</b>	
dao_Controlar_Limpieza()	
<b>Descripción:</b> Constructor de la clase	
Insertar_Incidencias(incidencia: var)	
<b>Descripción:</b> Responsable de insertar las incidencias de la limpieza pasado un objeto incidencia por parámetro	
Eliminar_Incidencias(incidencia: var)	
<b>Descripción:</b> Responsable de eliminar las incidencias de la limpieza pasado un objeto incidencia por parámetro	
Get_Estudiantes_Incidentes(fecha: var)	
<b>Descripción:</b> Responsable de obtener los estudiantes con incidencias pasando la fecha por parámetro	

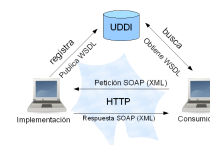


## Capítulo 3: Análisis y Diseño

Get_Descripcion(id_estud: var)
<b>Descripción:</b> Responsable de obtener las incidencia de un estudiante pasando el id por parámetro.

Tabla 34 ado\_Controlar\_cuartelería

<b>Nombre: ado_Controlar_cuartelería</b>	
<b>Tipo de clase: Clase controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
-----	-----
<b>Responsabilidades</b>	
dao_Controlar_cuartelería()	
<b>Descripción:</b> Constructor de la clase	
Insertar_Evaluación(evaluación: var)	
<b>Descripción:</b> Responsable de insertar la evaluación de la cuarterería pasando un objeto evaluación por parámetro	
Insertar_Incidencias(incidencia: var)	
<b>Descripción:</b> Responsable de insertar las incidencias de la cuarterería pasando un objeto incidencia por parámetro	
Buscar_Cuarteleros(edificio: var , paso_escalera: var)	
<b>Descripción:</b> Responsable de obtener los cuartereros pasando el edificio y paso escalera como parámetro	
Cambiar_Cuartelería(id_cuart1: var, \$id_cuart2: var)	
<b>Descripción:</b> Responsable de cambiar la cuarterería entre dos cuartereros, pasando los cuartereros por parámetro	
Correr_Cuartelería(edificio: var, paso_escalera: var)	
<b>Descripción:</b> Responsable correr la cuarterería pasando el edificio y paso escalera por parámetro	
Existen_en_la_Lista(solapin: var)	
<b>Descripción:</b> Responsable de verificar que el estudiante se encuentre en la lista de la cuarterería pasando el solapín por parámetro	
Insertar_Cuartelos(cuartelero: var)	
<b>Descripción:</b> Responsable de insertar un cuarterero dado un objeto del mismo tipo pasado por parámetro	
Get_Cuartelero(edificio: var, paso_escalera: var, fecha: var)	
<b>Descripción:</b> Responsable de obtener un cuarterero pasando el edificio, paso de escalera y la fecha por parámetro	
Get_Evaluacion(solapin: var)	
<b>Descripción:</b> Responsable de obtener la evaluación de la cuarterería pasando el	

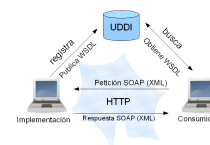


## Capítulo 3: Análisis y Diseño

solapín del estudiante por parámetro

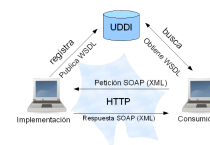
**Tabla 35** CI\_Administracion\_del\_Becado

<b>Nombre: CI_Administracion_del_Becado</b>	
<b>Tipo de clase: Interfaz</b>	
<b>Atributo</b>	<b>Tipo</b>
-----	-----
<b>Responsabilidades</b>	
obtener_estudiante(solapin: xsd: string)	
<b>Descripción:</b> Servicio encargado de devolver un estudiante dados los datos enviados	
insertar_estudiante(key: xsd: string, militante: xsd: int, dirigente: xsd: int, telef_apto: xsd: string, proye: xsd: string, solapín: xsd: string, num_lab: xsd: string, telef_lab: xsd: string, aa: xsd: bool, asignatura: xsd: string, nom_completo: xsd: string, apto: xsd: string, edificio: xsd: string)	
<b>Descripción:</b> Servicio encargado de insertar a un estudiante dados los datos enviados	
eliminar_estudiante(key: xsd: string, id_estud: xsd: int)	
<b>Descripción:</b> Servicio encargado de eliminar un estudiante dados los datos enviados	
actualizar_datos_estudiante(key: xsd: string, militante: xsd: int, dirigente: xsd: int, telef_apto: xsd: string, proye: xsd: string, solapín: xsd: string, num_lab: xsd: string, telef_lab: xsd: string, aa: xsd: bool, asignatura: xsd: string, nom_completo: xsd: string, apto: xsd: string, edificio: xsd: string)	
<b>Descripción:</b> Servicio encargado de actualizar los datos del estudiante dados los datos enviados	
insertar_cuarteleros(apto: xsd: string, paso_escalera: xsd: int, tns: array_fecha, key: xsd: string)	
<b>Descripción:</b> Servicio encargado de insertar un cuartelero dados los datos enviados	
Listar_cuarteleros(edificio: xsd: string, paso_escalera: xsd: int)	
<b>Descripción:</b> Servicio encargado de listar los cuarteleros dados los datos enviados	
cambiar_cuartelería(id_cuart1: xsd: int, id_cuart2: xsd: int)	
<b>Descripción:</b> Servicio encargado de cambiar la cuarterería entre dos cuartereros dados los datos enviados	
correr_cuartelería(edificio: xsd: string, paso_escalera: xsd: int)	
<b>Descripción:</b> Servicio encargado de correr la cuarterería dados los datos enviados	
obtener_cuartelero(edificio: xsd: string, paso_escalera: xsd: int, fecha: xsd: datetime)	
<b>Descripción:</b> Servicio encargado de devolver un cuarterero dados los datos enviados	



## Capítulo 3: Análisis y Diseño

Insertar_evalucion_cuarterero(eval: xsd: int, key: xsd: string)
<b>Descripción:</b> Servicio encargado de insertar la evaluación del cuarterero dados los datos enviados
insertar_grupo_al_tsu(grupo: xsd: string, sesión: xsd: int, fecha: xsd: datetime)
<b>Descripción:</b> Servicio encargado de insertar un grupo al tsu dados los datos enviados
mostrar_grupo_tsu(fecha: xsd: datetime, sesion: xsd: int)
<b>Descripción:</b> Servicio encargado de mostrar el grupo que participó en el tsu dados los datos enviados
obtener_estudiantes(grupo: xsd: string, apto: xsd: string, edificio: xsd: string, nombre: xsd: string)
<b>Descripción:</b> Servicio encargado de devolver un listado de estudiante dados los datos enviados
insertar_asistencia_tsu(key: xsd: string, id_tsu: xsd: int, tns: array_id_estud)
<b>Descripción:</b> Servicio encargado de insertar la asistencia del tsu dados los datos enviados
mostrar_tsu(fecha: xsd: datetime, sesion: xsd: int)
<b>Descripción:</b> Servicio encargado de devolver los datos del tsu dados los datos enviados
mostrar_asistencia(id_tsu: xsd: int)
<b>Descripción:</b> Servicio encargado de devolver la asistencia del tsu dados los datos enviados
insertar_grupo_guardia(grupo: xsd: string, fecha: xsd: datetime)
<b>Descripción:</b> Servicio encargado de insertar grupo a la guardia dados los datos enviados
mostrar_guardia(fecha: xsd: datetime)
<b>Descripción:</b> Servicio encargado de devolver los datos de la guardia dados los datos enviados
insertar_asistencia_guardia(id_guard: xsd: int, tns: array_id_estud)
<b>Descripción:</b> Servicio encargado de insertar la asistencia a la guardia dados los datos enviados
mostrar_asistencia_guardia(id_guard: xsd: int)
<b>Descripción:</b> Servicio encargado de devolver la asistencia de la guardia dados los datos enviados
insertar_incidencias(id_estud: xsd: int, inc: xsd: string, key: xsd: string)
<b>Descripción:</b> Servicio encargado de insertar la incidencia del estudiante dados los datos enviados
mostrar_estudiantes_incidentes(fecha: xsd: datetime)
<b>Descripción:</b> Servicio encargado de devolver a los estudiantes con incidencias dados los datos enviados
mostrar_incidencia(id_estud: xsd: int)
<b>Descripción:</b> Servicio encargado de devolver las incidencias de un estudiante dados



## Capítulo 3: Análisis y Diseño

los datos enviados
mostrar_estudiantes_categoria(categoria: xsd: string, localidad: xsd: string)
<b>Descripción:</b> Servicio encargado de devolver un listado de estudiantes dados los datos enviados
eliminar_incidencia(id_inc: xsd: int)
<b>Descripción:</b> Servicio encargado de eliminar incidencia dados los datos enviados
mostrar_estudiantes_anno(anno: xsd: int, localidad: xsd: string)
<b>Descripción:</b> Servicio encargado de devolver un listado de estudiante por año dados los datos enviados
insertar_eval_mensual(key: xsd: string, eval: xsd: string)
<b>Descripción:</b> Servicio encargado de insertar la evaluación mensual dados los datos enviados
mostrar_eval_cuartero(edificio: xsd: string, paso_escalera: xsd: int, fecha: xsd: datetime)
<b>Descripción:</b> Servicio encargado de mostrar la evaluación del cuartero dados los datos enviados
mostrar_eval_mensual(apto: xsd: string, edificio: xsd: string, nombre: xsd: string, fecha: xsd: datetime)
<b>Descripción:</b> Servicio encargado de mostrar la evaluación mensual dado los datos enviados

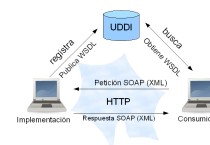
### 3.5 Diseño de la Base Datos

El modelo de base datos más usado hoy en día es el modelo relacional, con el que se puede relacionar unos datos con otros para facilitar el trabajo del programador y también el momento de realizar una búsqueda. Para construir este modelo se empleó el modelo relacional, este modelo se utilizó teniendo en cuenta que constituye la representación física de la misma. **ANEXO 4**

### 3.6 Descripción de las tablas de la Base Datos

**Tabla 36** Tb\_Estudiante

<b>Nombre:</b> Tb_Estudiante
<b>Descripción:</b> Tabla que almacena los datos de los becados



## Capítulo 3: Análisis y Diseño

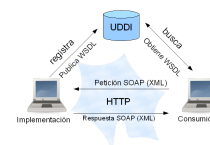
Atributo	Tipo	Descripción
Id_estud	int	Identificador de la tabla
telef_apto	varchar (20)	Número de teléfono del apto
nom_proye	varchar (50)	Nombre del proyecto que pertenece el estudiante
num_lab	varchar (11)	Número del laboratorio en que trabaja el estudiante
telef_lab	varchar (20)	Teléfono del laboratorio del estudiante
a_ayudante	varchar (2)	Si es o no alumno ayudante
asignatura	varchar (50)	Asignatura que imparte el alumno ayudante
nombre_completo_	varchar (20)	Nombres y apellidos del estudiante
edificio	varchar (11)	Edificio donde vive el estudiante
apto	varchar (11)	Apartamento donde vive el estudiante
solapin	varchar (11)	Solapín del estudiante
Id_dirigente	int	Identificador de la tabla donde se guarda el tipo de dirigente que puede pertenecer el estudiante (FEU y/o UJC)
Id_tipo_milit	int	Identificador de la tabla donde se guarda el tipo de militancia a la que pertenece el estudiante (UJC y/o PCC)

**Tabla 37** Tb\_Incidencia

Nombre: Tb_Incidencia		
<b>Descripción:</b> Tabla que almacena las incidencias que existen que puedan existir en el control de las actividades		
Atributo	Tipo	Descripción
Id_incidencia	int	Identificador de la tabla
fecha_inc	datetime	Fecha en que se registró la incidencia
descrip	varchar (50)	Descripción de la incidencia
Id_estud	int	Identificador de la tabla estudiante

**Tabla 38** Tb\_Evaluacion\_mensual

Nombre: Tb_Evaluacion_mensual		



## Capítulo 3: Análisis y Diseño

<b>Nombre: Tb_Evaluacion_mensual</b>		
<b>Descripción:</b> Tabla que almacena la evaluación mensual de los estudiantes		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_eval	int	Identificador de la tabla
fecha	datetime	Fecha de la evaluación
Id_Tipo_Eva	int	Identificador de la tabla donde se guarda el tipo de evaluación
Id_estud	int	Identificador de la tabla estudiante

**Tabla 39** Tb\_TSU

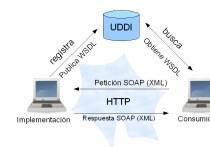
<b>Nombre: Tb_TSU</b>		
<b>Descripción:</b> Tabla que almacena los datos del TSU		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_tsu	int	Identificador de la tabla
fecha_tsu	datetime	Fecha en que se realizó el TSU
grupo	varchar (11)	Grupo que participó en el TSU
Id_sesion	int	Identificador de la tabla donde se guarda la sesión del TSU

**Tabla 40** Tb\_Guardia

<b>Nombre: Tb_Guardia</b>		
<b>Descripción:</b> Tabla que almacena los datos de la guardia		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_guardia	int	Identificador de la tabla
fecha_guar	datetime	Fecha en que se realizó la guardia
grupo	varchar (11)	Grupo que participó en la guardia

**Tabla 41** Tb\_Evaluación\_cuartelero

<b>Nombre: Tb_Evaluación_cuartelero</b>		
<b>Descripción:</b> Tabla que almacena la evaluación de los cuarteros		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_eval	int	Identificador de la tabla



## Capítulo 3: Análisis y Diseño

fecha	datetime	Fecha de la evaluación
Id_Tipo_Eva	int	Identificador de la tabla donde se guarda el tipo de evaluación
Id_cuarteria	int	Identificador de la tabla cuarterero
Id_estud	int	Identificador de la tabla estudiante

**Tabla 42** Tb\_Asistencia\_TSU

<b>Nombre: Tb_Asistencia_TSU</b>		
<b>Descripción:</b> Tabla que almacena la asistencia al TSU		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_tsu	int	Identificador de la tabla TSU
solapin	varchar (11)	Solapín del estudiante

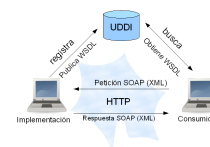
**Tabla 43** Tb\_Tipo\_militancia

<b>Nombre: Tb_Tipo_militancia</b>		
<b>Descripción:</b> Tabla nomencladora que guarda el tipo de militancia a la que puede pertenecer el estudiante		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_tipo_milit	int	Identificador de la tabla
Tipo	varchar (11)	Tipo de militancia que puede pertenecer el estudiante (UJC y/o PCC)

**Tabla 44** Tb\_Tipo\_dirigente

<b>Nombre: Tb_Tipo_dirigente</b>		
<b>Descripción:</b> Tabla nomencladora que guarda el tipo de dirigente que puede ejercer el estudiante		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_tipo_dirigente	int	Identificador de la tabla
Tipo	varchar (11)	Tipo de dirigente que puede ser el estudiante (FEU y/o UJC)





## Capítulo 3: Análisis y Diseño

**Tabla 45** Tb\_Tipo\_Evaluación

<b>Nombre: Tb_Tipo_Evaluación</b>		
<b>Descripción:</b> Tabla nomencladora que guarda las diferentes evaluaciones del estudiante		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_tipo_Eva	int	Identificador de la tabla
Tipo	char(1)	Tipo de evaluaciones que puede tener un estudiante (B, R ó M)

**Tabla 46** Tb\_Asistencia\_Guardia

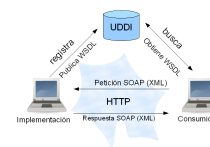
<b>Nombre: Tb_Asistencia_Guardia</b>		
<b>Descripción:</b> Tabla que almacena la asistencia a la guardia estudiantil		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_guardia	int	Identificador de la tabla guardia
solapin	varchar (11)	Solapín del estudiante

**Tabla 47** Tb\_Estudiante\_Tb\_Guardia

<b>Nombre: Tb_Estudiante_Tb_Guardia</b>		
<b>Descripción:</b> Tabla que almacena el id del estudiante y el id de la guardia		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_guardia	int	Identificador de la tabla guardia
Id_estud	int	Identificador de la tabla estudiante

**Tabla 48** Tb\_Estudiante\_Tb\_TSU

<b>Nombre: Tb_Estudiante_Tb_TSU</b>		
-------------------------------------	--	--



## Capítulo 3: Análisis y Diseño

<b>Descripción:</b> Tabla que almacena el id del estudiante y el id del TSU		
Atributo	Tipo	Descripción
Id_tsu	int	Identificador de la tabla TSU
Id_estud	int	Identificador de la tabla estudiante

### 3.7 Estudio de la Factibilidad

El estudio de la factibilidad es de suma importancia para realizar el análisis del costo y los beneficios que estos reportarán. A partir de este análisis se obtiene el tiempo de desarrollo en meses por hombre, y la estimación de costo del sistema propuesto.

#### Planificación basada en casos de uso.

Pasos para la estimación:

#### Paso 1. Calcular el Punto de Casos de Usos sin ajustar.

$$UUCP = UAW + UUCW$$

Donde:

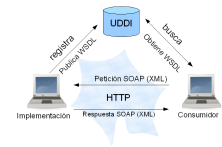
**UUCP:** Puntos de casos de uso sin ajustar.

**UAW:** Factor de peso de los actores sin ajustar.

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

**Tabla 49** Factor de peso de los actores sin ajustar.

Tipo de actor	Descripción	Factor de peso	Actores	Total
Simple	Sistema con sistema a través de interfaz	1	0	0



## Capítulo 3: Análisis y Diseño

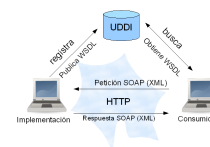
	de programación.			
Medio	Sistema con sistema mediante protocolo de interfaz basada en texto.	2	1	2
Complejo	Persona que interactúa con el sistema mediante interfaz gráfica.	3	0	0
Total			0	2

$$UAW = \sum cant \ actores * peso$$

$$UAW = 2$$

**Tabla 50** 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	0	0
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	6	60
Complejo	El caso de uso tiene más de 8 transacciones.	15	1	15
Total			7	75



## Capítulo 3: Análisis y Diseño

$$UUCW = \sum cant\ CU * P_{esc}$$

UUCW = 75

**UUCP:** Puntos de Casos de Uso sin ajustar

**UUCP = UAW + UUCW.**

UUCP = 3 + 75

UUCP = 78

**Paso 2. Calcular los Puntos de casos de uso ajustados.**

$$UCP = UUCP * TCF * EF$$

Donde:

**UCP:** Puntos de casos de uso ajustados.

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

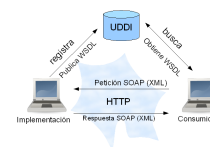
**TCF:** Factor de complejidad técnica.

**EF:** Factor de ambiente.

El factor de complejidad técnica (TCF) se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante.

**Tabla 51** Factor de complejidad técnica

Factor	Descripción	Peso	Valor asignado	Total
T1	Sistema distribuido	2	1	2
T2	Tiempo de respuesta	1	4	4
T3	Eficiencia del usuario final	1	3	3



## Capítulo 3: Análisis y Diseño

T4	Funcionamiento Interno complejo	1	3	3
T5	El código debe ser reutilizable	1	3	3
T6	Facilidad de instalación	0,5	4	2
T7	Facilidad de uso	0,5	5	2,5
T8	Portabilidad	2	5	10
T9	Facilidad de cambio	1	4	4
T10	Concurrencia	1	2	2
T11	Incluye objetivos especiales de seguridad	1	2	2
T12	Provee acceso directo a terceras partes	1	0	0
T13	Se requieren facilidades especiales de entrenamiento de usuarios	1	1	1
Total				38.5

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

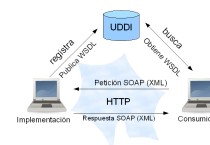
$$TCF = 0.6 + 0.01 * 38.5$$

$$TCF = 0.985$$

El factor de ambiente (EF) está relacionado con las habilidades y el entrenamiento del grupo involucrado en el desarrollo del sistema. El cálculo del mismo es similar al cálculo del Factor de complejidad técnica, se trata de un conjunto de factores que se cuantifican con valores de 0 a 5.

**Tabla 52** Factor de ambiente.

Factor	Descripción	Peso	Valor asignado	Total
--------	-------------	------	----------------	-------



## Capítulo 3: Análisis y Diseño

E1	Familiaridad con el modelo de proyecto utilizado	1,5	1	1,5
E2	Experiencia en la aplicación	0,5	1	0,5
E3	Experiencia en la orientación a objetivos.	1	2	2
E4	Capacidad del analista líder.	0,5	3	1,5
E5	Motivación.	1	5	5
E6	Estabilidad de requerimientos	2	5	10
E7	Personal Part–Time	-1	0	0
E8	Dificultad del lenguaje de programación	-1	2	-2
Total				18,5

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

$$EF = 1.4 - 0.03 * 18.5$$

$$EF = 0.845$$

$$UCP = UUCP * TCF * EF$$

$$UCP = 68 * 0.985 * 0.845$$

$$UCP = 56.5981$$

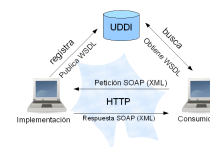
### Paso 3. Estimación de esfuerzo a través de los puntos de casos de uso

$$E = UCP * CF$$

Donde:

**E:** Esfuerzo estimado en horas hombres.

**UCP:** Punto de casos de usos ajustados.



## Capítulo 3: Análisis y Diseño

**CF:** Factor de conversión.

El factor de conversión (CF) se obtiene contando 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 utilizó el factor de conversión:

**CF = 28 Horas-Hombre / Punto de Casos de uso.**

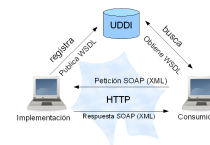
$E = 56,5981 * 28$

$E = 1584,7468$  Horas-Hombre

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

**Tabla 53** Esfuerzo del proyecto.

Actividad	Porcentaje %	Horas-Hombres
Análisis	10	396,1867
Diseño	20	792,3734
Implementación	40	1584,7468
Pruebas	15	594,28005
Sobrecarga (otras actividades)	15	594,28005
Total	100	3961,867



## Capítulo 3: Análisis y Diseño

Si  $E_T = 3961,867$  horas-hombre y cada mes tiene como promedio 264 horas laborables, eso daría un  $E_T=15,007$

Esto significa que una sola persona terminaría el proyecto en 15 meses aproximadamente.

### -Costo del Proyecto.

Se asume como salario promedio mensual \$100.00

**CH:** Cantidad de hombres.

**Tiempo:** Tiempo total del proyecto.

CH=1 hombre

**CHM = CH \* Salario Promedio**

CHM = 1 \* 100

CHM= 100.00 \$/mes

Costo = CHM \*  $E_T$  / CH

Costo = \$100.00 \* 15,007 / 1

Costo  $\approx$  \$1500,70

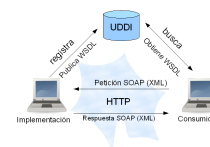
De los resultados obtenidos se interpreta que 1 hombre que trabaje en un proyecto este va a tener una duración de 15 meses y un costo de \$1500,70

### Análisis de costos y beneficios.

Desarrollar un producto informático tiene su costo y justificar dicho costo depende de los beneficios que reporten su implantación y utilización, dichos beneficios pueden ser de orden económico y social. El sistema propuesto es un servicio web para ser usado por la intranet de la facultad 1, para controlar las actividades que se realizan en el área de la residencia por lo tanto este servicio tiene un carácter social. Las herramientas y tecnologías usadas están enmarcadas al desarrollo del software libre por lo tanto no se incurre en gastos de licencias.

Todo el análisis realizado anteriormente demuestra la factibilidad del desarrollo completo de la aplicación.



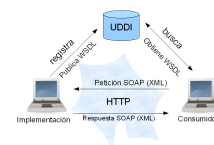


## Capítulo 3: Análisis y Diseño

---

### Conclusión

Con el desarrollo de este capítulo quedó plasmado el diseño final que tendrá el sistema propuesto, que servirá para su implementación. En este capítulo se llevó a cabo las descripciones de las clases. El diagrama de clases está diseñado por paquete, dividido en subpaquetes según sus funcionalidades. A partir de las clases entidades, se realizó el modelo de datos, basado en el modelo entidad relación. Se expusieron los servicios que va a brindar el servicio web. También se realizó el estudio de la factibilidad y el análisis del costo estimado del sistema propuesto.



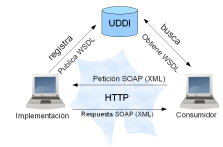
## Conclusiones Generales

---

### CONCLUSIONES GENERALES

Se hizo necesario diseñar un servicio web para su futura implementación que permita realizar con mayor comodidad, eficiencia y rapidez el control de las actividades y la administración de los datos del estudiante en la residencia estudiantil, ya que los estudios realizados de otros sistemas similares a nivel internacional, nacional y de universidad no satisfacen la necesidad del problema planteado.

El sistema se representó mediante tecnología RUP y se modelaron todos los diagramas que fueron necesarios para desarrollar el análisis y diseño del servicio web. De esta forma se cumplieron todos los objetivos planteados.



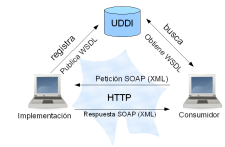
## Recomendaciones

---

### RECOMENDACIONES

A pesar de que se cumplieron todos los objetivos, surgieron ideas que se recomiendan, ya que pueden propiciar una mejora de los procesos en un futuro.

- Se recomienda que la evaluación mensual del estudiante se genere a partir de la participación del estudiante en todas las actividades que se realizan en el área de la residencia.
- Implementar las funcionalidades de los servicios diseñado para ser usado lo antes posible por el proyecto.
- Se recomienda que estas funcionalidades sean incluidas al Sistema de Gestión Residentes, para que este funcione de forma más integral y que los trabajadores de la residencia no tengan que utilizar dos sistemas diferentes con algunas funcionalidades similares.



## Bibliografía Referenciada

---

### **BIBLIOGRAFÍA REFERENCIADA**

1. ¿Qué es XML, SOAP, WSDL, UDDI? Noción Digital. [Online] 2008.
2. Alvarez, Miguel Angel. ¿Qué es XML? Desarrolloweb.com. [Online] 2007.  
<http://www.desarrolloweb.com/articulos/449.php>.
3. Pérez, Damián Valdés. ¿Que es JavaScript? Maestros del Web.com. [Online] 2008.  
<http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript>.
4. Sistema Gestor de base de datos SGBD. Garballe Collector. [Online] Noviembre 01, 2004.  
[http://www.error500.net/garbagecollector/archives/categorias/bases\\_de\\_datos/sistema\\_gestor\\_de\\_base\\_de\\_datos\\_sgbd.php](http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbd.php).
5. POLANCO, MCGILL LOZANO. ¿Que es UML? MCGILL LOZANO POLANCO. [Online] 2007.  
<http://mayi.polanco.googlepages.com/TRABAJODEINGSOFTWAREII.doc>.
6. Módulo (programación). Wikipedia. [Online] 2008.
7. ¿Qué es Intranet? Mi Respuestas.com. [Online] 2008. <http://www.misrespuestas.com/que-es-intranet.html>.
8. Interbusca. Interbusca.com. [Online] 2008. <http://antivirus.interbusca.com/glosario/HTTP.html>.

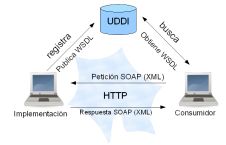


## Bibliografía

---

### BIBLIOGRAFÍA

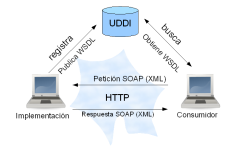
1. Jacobson, Ivar, Booch, Grady y James, Rumbaugh. El proceso unificado de software. La Habana: Felix Varela, 2004. Vol. 1.
2. Jacobson, Ivar, Booch, Grady y James, Rumbaugh. El proceso unificado de software. La Habana: Felix Varela, 2004. Vol. 2.
3. Pressman, Rogers. Ingeniería del software. La Habana: Felix Varela, 2005. Vol. 1.
4. Pressman, Rogers. Ingeniería del software. La Habana: Felix Varela, 2005. Vol. 2.
5. Madelyn Saumell Reyes, Miguel Jaeger Rodríguez Lazo. Módulo Control de Acceso del proyecto Intranet del Centro Rector de Universidad para Todos. Universidad de las Ciencias Informáticas. Ciudad de la Habana: s.n., 2007. Tesis.
6. Dinia Zayas Romero, Gerxis Sam Alcántara. Modelación de Servicios Web para el procesamiento y la representación de información geoespacial. Universidad de las Ciencias Informáticas. Ciudad de la Habana: s.n., 2007.
7. Rolando Alfredo Hernández León, Sayda Coello González. EL PARADIGMA CUANTITATIVO DE LA INVESTIGACIÓN CIENTÍFICA. Metodología de la Investigación, Universidad de las Ciencias Informáticas. Ciudad de la Habana: s.n., 2002. Tesis. 959-16-0343-6.



## Bibliografía

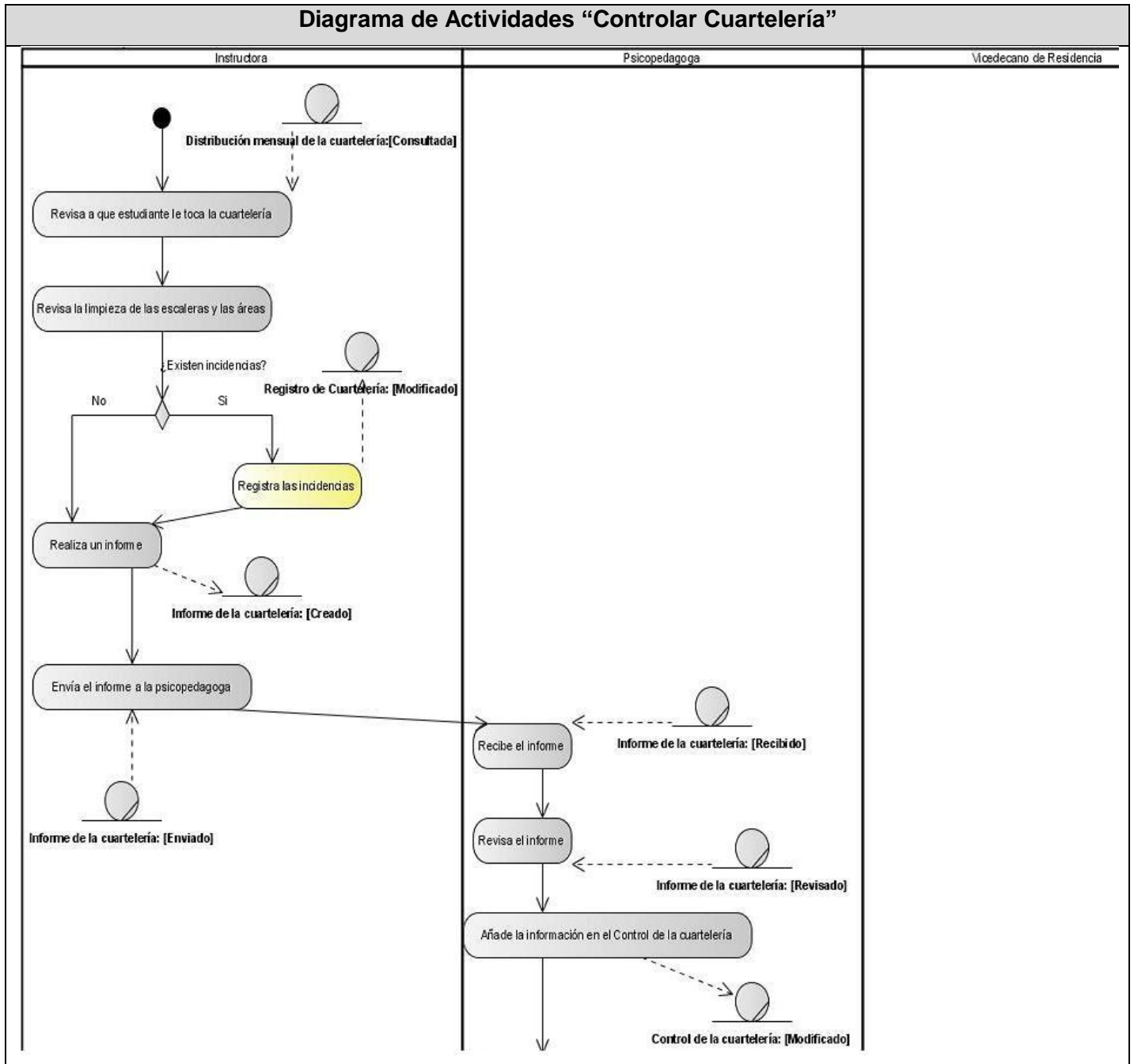
---

8. C., Benjamín González. WSDL para la documentación de Servicios Web. Desarrollo Web. [En línea] 2008. <http://www.desarrolloweb.com/articulos/1581.php>
9. DocCF. Gupo CF Developer. [En línea] 2008. [Citado el: 14 de Marzo de 2008.] <http://www.grupocfdeveloper.com>
10. Anyell Cano Ramos, Kenia Pérez Hernández, Rinaldo Moreno Valdés, Sergio Valle Hernández. Informática Evento Virtual. [En línea] 2007. [www.informaticahabana.com](http://www.informaticahabana.com)
11. Christian Van Der Henst S. Maestro del Web. [En línea] 23 de 05 de 2007. <http://www.maestrosdelweb.com/editorial/desarrollo-web/php/>
12. Lic. Armando Armijo Castro, Ladislao Armando Armijo Olavarría. CALMECAC. Krunnware. [En línea] 12 de 05 de 2008. [www.krunnware.com](http://www.krunnware.com)
13. Ajax un nuevo acercamiento a las aplicaciones web. Maestros del Web. [En línea] 2008. <http://www.maestrosdelweb.com/editorial/ajax/>
14. Metodología de Desarrollo de Software. REYNOX. [En línea] 2005. [Citado el: 15 de 03 de 2008.] [www.reynox.com](http://www.reynox.com)
15. Moral, Juan Antonio Breña. ¿Que es un Web Service? Web Taller. [En línea] 06 de 06 de 2008. [Citado el: 12 de 03 de 2008.] [http://www.webtaller.com/maletin/articulos/que\\_son\\_web\\_services.php](http://www.webtaller.com/maletin/articulos/que_son_web_services.php)
16. Váldez, Damian Péres. ¿Que es JavaScript? Maestro del Web. [En línea] 07 de 03 de 2007. <http://www.maestrosdelweb.com/editorial/%c2%bfque-es-javascript/>



# Anexo 1

## ANEXO 1



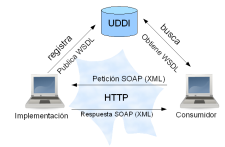
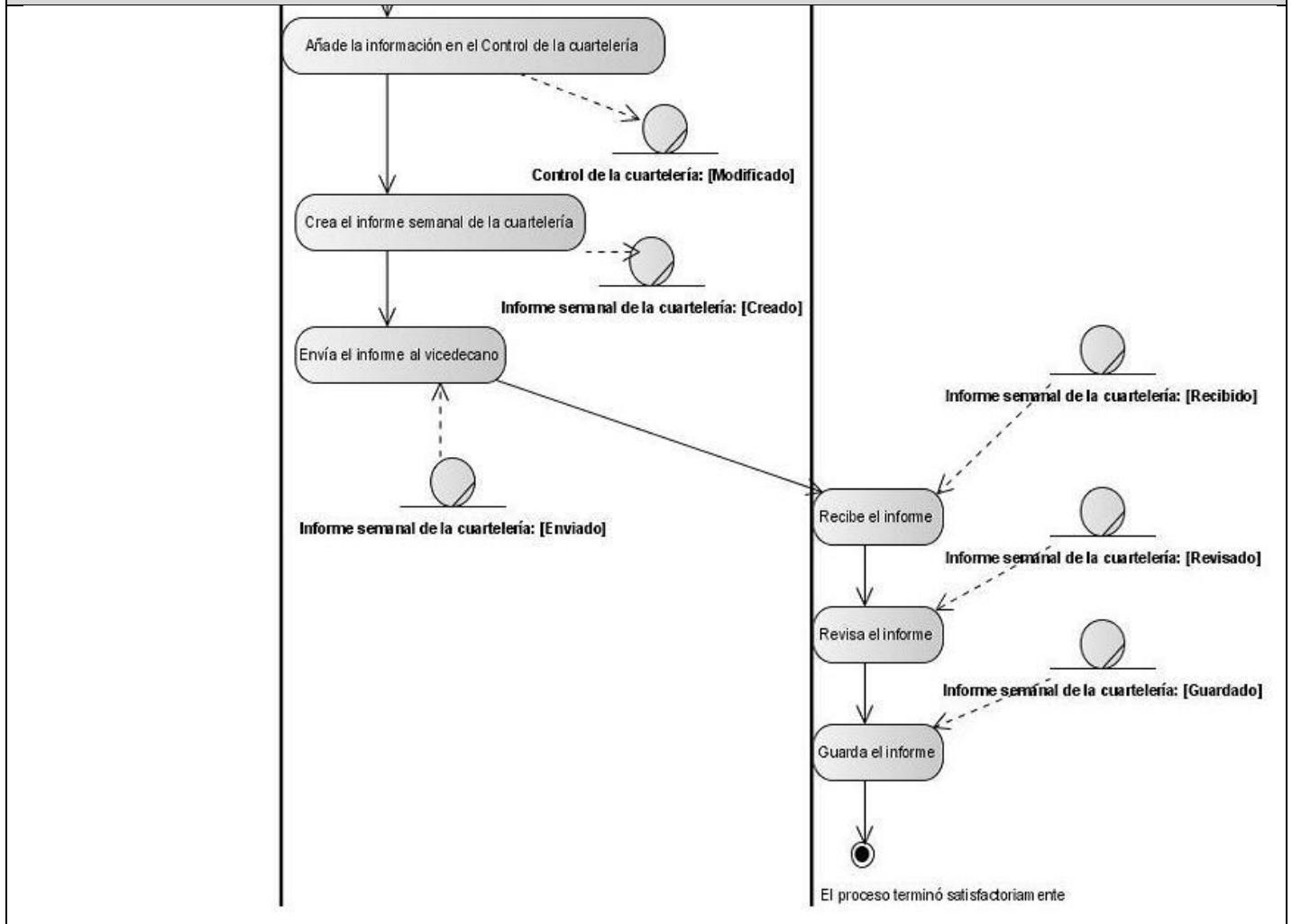


Diagrama de Actividades “Controlar Cuartelería [Continuación]”





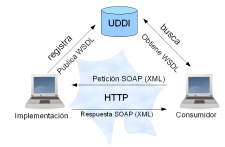
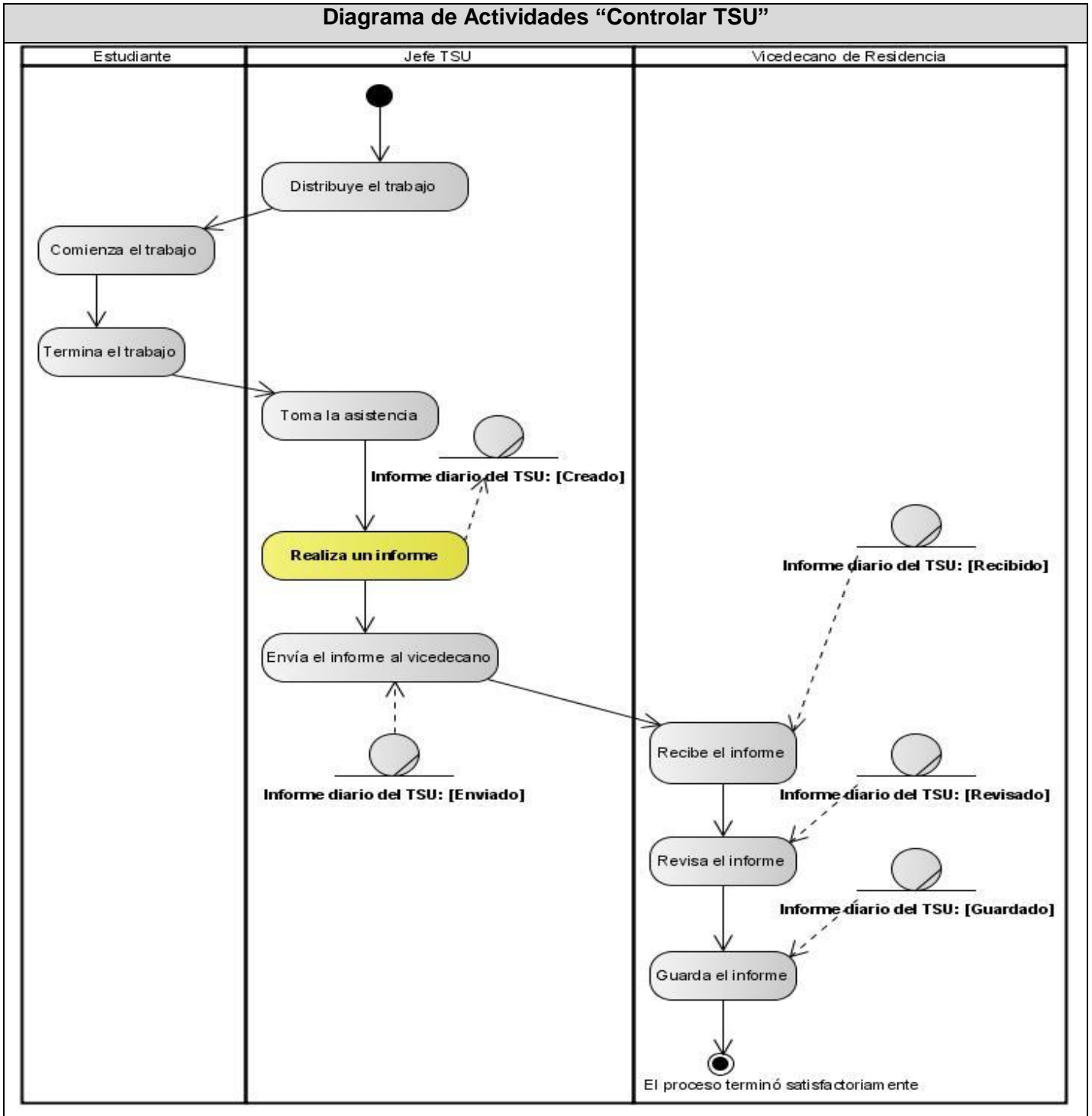
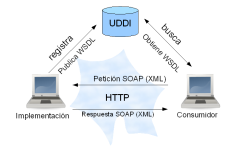


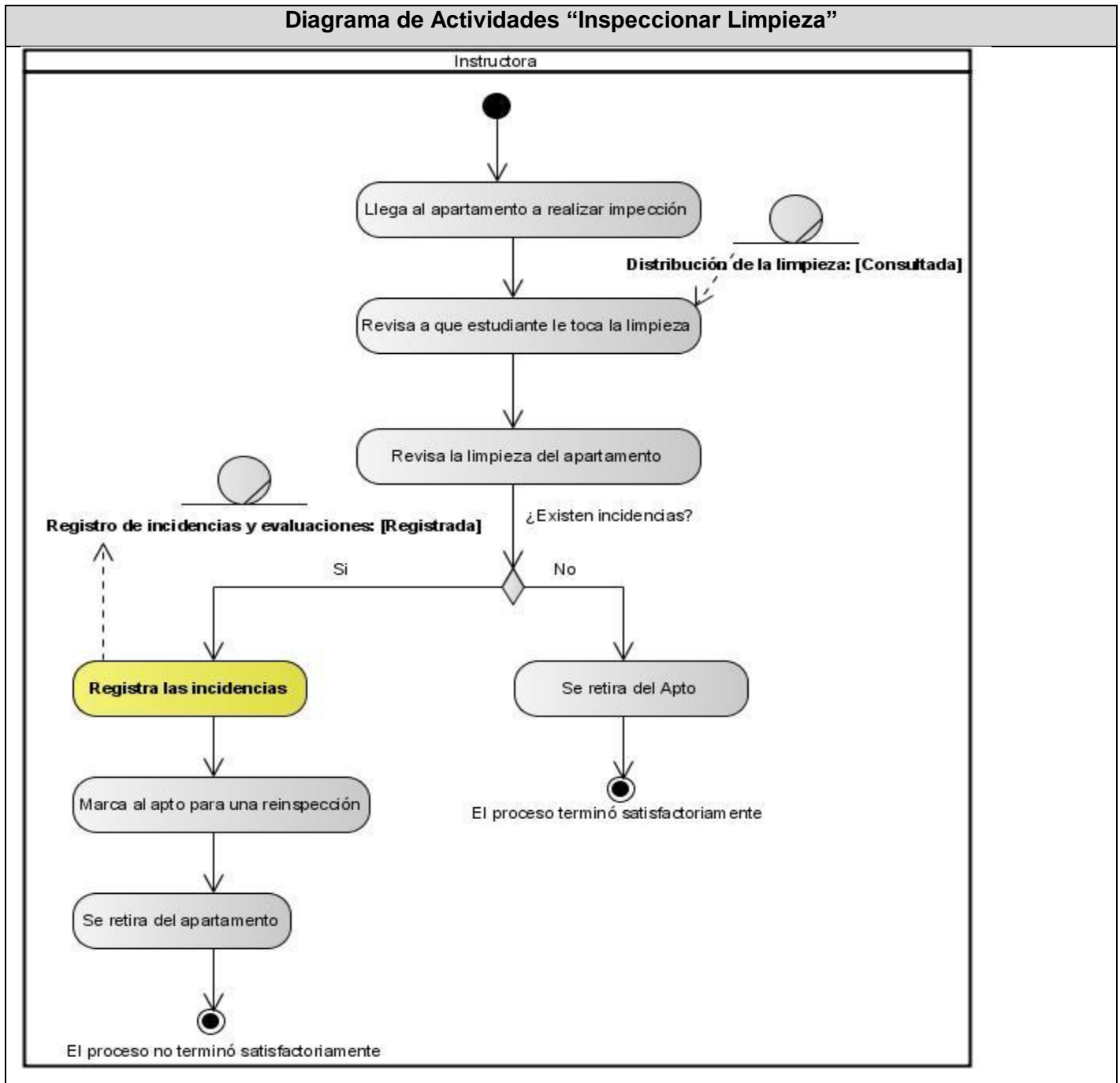
Diagrama de Actividades "Controlar TSU"



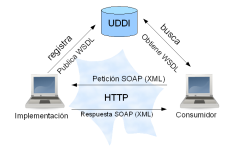


# Anexo 1

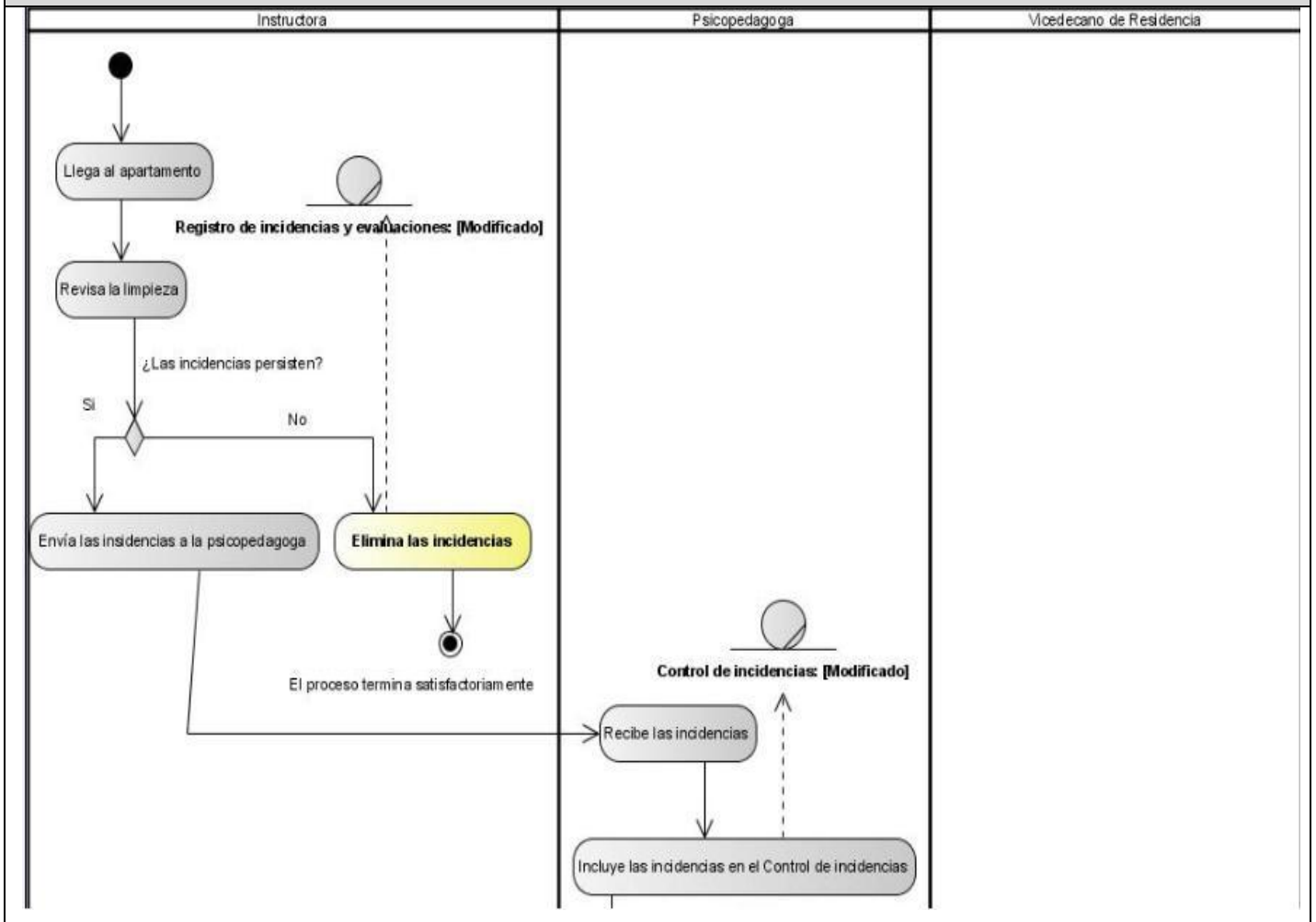
**Diagrama de Actividades “Inspeccionar Limpieza”**

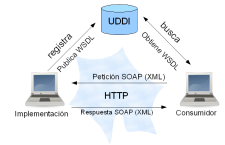


# Anexo 1

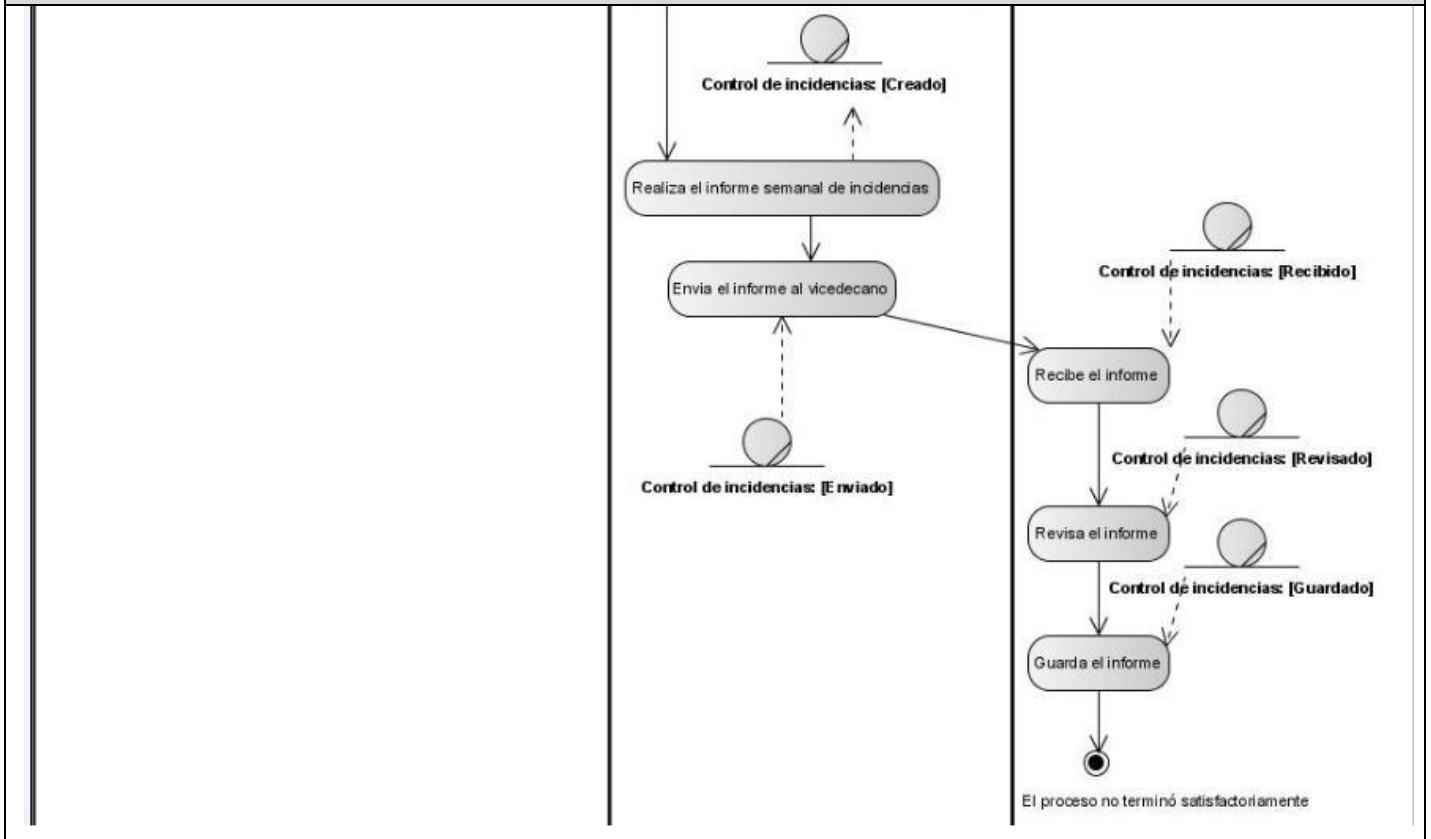


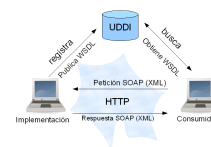
**Diagrama de Actividades “Reinspeccionar limpieza”**





**Diagrama de Actividades “Reinspeccionar limpieza [Continuación]”**



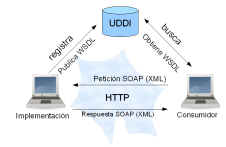


## Anexo 2

### ANEXO 2

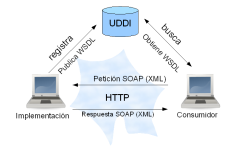
Tabla A2.1: Caso de Uso “Gestionar datos del estudiante”

Especificación de Caso de Uso de Sistema	
<b>Identificador</b>	CUS_1
<b>Nombre</b>	Gestionar datos del estudiante
<b>Descripción</b>	El caso de uso se inicia cuando el actor invoca el servicio obtener estudiante pasándole el solapín. El sistema muestra los datos del estudiante. En caso de insertar datos, el actor invoca el servicio insertar datos del estudiante, en caso de eliminar o actualizar datos invoca el servicio eliminar estudiante y actualizar estudiante, verificando previamente que el estudiante se encuentre en el sistema, y dados estos datos actualiza los datos o elimina. Termina el caso de uso.
<b>Actor primario</b>	Intranet Facultad 1
<b>Propósito</b>	Modificar, insertar, o eliminar datos del estudiante
<b>Prioridad de Implantación</b>	Alta
<b>Requisitos Funcionales</b>	R1.1, R1.2, R1.3, R1.4, R1.5, R1.6
Flujo de Eventos	
Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El actor invoca el servicio obtener estudiante, pasándole los datos pertinentes (solapín)	2. Verifica que el número de solapín es correcto. Sino ver flujo Alternativo 1 3. Envía la petición realizada por el actor 4. Muestra al estudiante



## Anexo 2

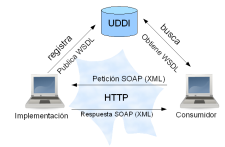
<p>7. Si el actor desea invocar el servicio Insertar datos del estudiante, ver la sección “Insertar datos”, si desea invocar el servicio actualizar datos del estudiante, ver la sección “Actualizar datos”, si desea invocar el servicio eliminar datos del estudiante, ver la sección “Eliminar datos”, si desea invocar el servicio insertar la evaluación mensual, ver la sección “Insertar Evaluación”, si desea invocar el servicio, mostrar evaluación mensual, ver la sección, “Mostrar Evaluación”</p>	
<b>Sección: Insertar datos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<p>1. Invoca el servicio insertar datos del estudiante, pasándole los datos</p>	<p>2. Verifica que los datos sean correctos. Sino ver Flujo Alternativo 2 3. Almacena los datos en la base de datos. Termina el caso de uso</p>
<b>Sección: Actualizar datos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<p>1. Invoca el servicio actualizar datos del estudiante pasándole los datos que van hacer actualizados</p>	<p>2. Verifica que los datos sean correctos. Sino ver Flujo Alternativo 2 3. Actualiza los datos en la base de datos. Termina el</p>



## Anexo 2

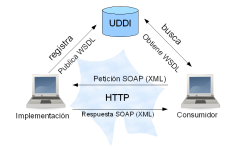
	caso de uso
<b>Sección: Eliminar datos</b>	
Acción del Actor	Respuesta del Sistema
1. Invoca el servicio eliminar datos del estudiante pasándole los datos que van hacer eliminados	2. Verifica que los datos sean correctos. Sino ver Flujo Alternativo 2 3. Solicita la confirmación para eliminar 4. Elimina los datos. Termina el caso de uso
<b>Flujo Alternativo 1</b>	
Acción del Actor	Respuesta del Sistema
	1. Muestra un mensaje diciendo que el número de solapín es incorrecto.
<b>Flujo Alternativo 2</b>	
Acción del Actor	Respuesta del Sistema
	1 Muestra un mensaje de error diciendo que existen datos incorrectos

Tabla A2.2: Caso de Uso “Controlar cuarterería”



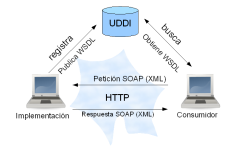
Especificación de Caso de Uso de Sistema	
<b>Identificador</b>	CUS_2
<b>Nombre</b>	Controlar cuarterería
<b>Descripción</b>	<p>El caso de uso se inicia cuando el actor desee insertar los cuartereros, invoca el servicio, insertar cuartereros almacenado los datos en la base de datos, en caso de que el actor desee cambiar la cuarterería, invoca el servicio cambiar cuarterería, el sistema permite hacer cambios entre dos estudiantes. En caso de que el actor desee correr la cuarterería, invoca el servicio correr cuarterería, el sistema adelanta un día la cuarterería. En caso de insertar alguna incidencia, el actor invoca el servicio insertar incidencias del cuarterero almacenando las incidencias. El actor invoca el servicio insertar evaluación para insertar la evaluación del cuarterero. El sistema almacena los datos en la base de datos En caso que el estudiante desee obtener la evaluación, invoca el servicio mostrar evaluación del cuarterero. Termina el caso de uso</p>
	Intranet Facultad 1
<b>Propósito</b>	Crear la cuarterería, realizar los cambios que se hacen en la misma y registra la evaluación y las incidencias en caso de que existan.
<b>Prioridad de Implantación</b>	Alta
<b>Requisitos Funcionales</b>	R2.1, R2.2..., R2.3..., R2.4, R2.5, R2.6
<b>Flujo de Eventos</b>	
<b>Flujo Básico</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>





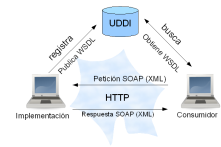
## Anexo 2

<p>1. El actor invoca el servicio insertar cuarteros, pasándole los datos pertinentes</p>	<p>2. Verifica que no existan campos en blancos. Sino ver flujo Alternativo 1</p> <p>3. El sistema almacena a los cuarteros en la base de datos</p>
<p>4. Si el actor desea invocar el servicio cambiar la cuartería, ver sección “Cambiar cuartero”, si desea invocar el servicio correr cuartería, ver sección “Correr cuartería”, si invoca el servicio insertar incidencias del cuartero, ver la sección “Insertar incidencias”, si desea invocar el servicio insertar evaluación del cuartero, ver la sección “Insertar evaluación”, si desea invocar e servicio mostrar evaluación del cuartero, ver la sección “Mostrar Evaluación”</p>	
<b>Sección: Cambiar cuartero</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<p>1. Invoca el servicio obtener cuarteros, pasándole los datos pertinentes</p>	<p>2. Valida que todos los campos hallan sido llenados correctamente. Sino ver flujo Alternativo 1</p> <p>3. Consulta a la base de datos</p> <p>4. Muestra a los cuarteros</p>
<p>5. Invoca el servicio cambiar cuartería, pasándole los dos cuarteros</p>	<p>6. Valida que los cuarteros existan en la lista de la cuartería. Sino ver flujo</p>



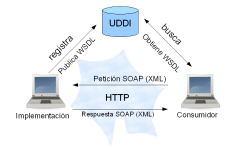
## Anexo 2

	<p>Alternativo 2</p> <p>7. El sistema intercambia las fechas a los cuartereros. Termina el caso de uso</p>
<b>Sección: Correr la cuarterería</b>	
Acción del Actor	Respuesta del Sistema
<p>1. Invoca el servicio correr cuarterería pasándole los datos pertinentes</p>	<p>2. Valida que no existan campos vacíos. Sino ver flujo Alternativo 1</p> <p>3. El sistema le adelanta en un día la fecha de la cuarterería</p> <p>4. Muestra a los cuartereros. Termina el caso de uso</p>
<b>Sección: Insertar incidencia</b>	
Acción del Actor	Respuesta del Sistema
<p>1. Invoca el servicio obtener cuarterero, pasándole los datos pertinentes</p>	<p>2. Verifica que los datos sean correctos. Sino ver flujo Alternativo 3</p> <p>3. El sistema muestra al cuarterero en turno</p>
<p>4. Invoca el servicio insertar incidencia del cuarterero.</p>	<p>5. Verifica que no existen campos vacíos. Sino ver flujo Alternativo 1</p> <p>6. Almacena los datos en la base de datos. Termina el caso de uso</p>



## Anexo 2

<b>Sección: Insertar evaluación</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Invoca el servicio obtener cuartelero, pasándoles los datos pertinentes	2. Verifica que los datos sean correctos. Sino ver flujo Alternativo 3 3. El sistema muestra al cuartelero en turno
4. Invoca el servicio insertar evaluación del cuartelero	5. Almacena la evaluación en la base de datos. Termina el caso de uso
<b>Sección: Mostrar Evaluación</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Invoca el servicio obtener cuartelero, pasándole los datos pertinentes	2. Verifica que los datos sean correctos. Sino ver flujo Alternativo 3 3. El sistema muestra al cuartelero en turno
4 Invoca el servicio mostrar evaluación mensual del cuartelero, pasándole los datos pertinentes	5 Verifica que los datos sean correctos. Sino ver flujo Alternativo 3 6 Consulta a la base datos 7 Muestra la evaluación del cuartelero. Termina el caso de uso
<b>Flujo Alternativo 1</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. Muestra un mensaje diciendo que existen campos vacíos



## Anexo 2

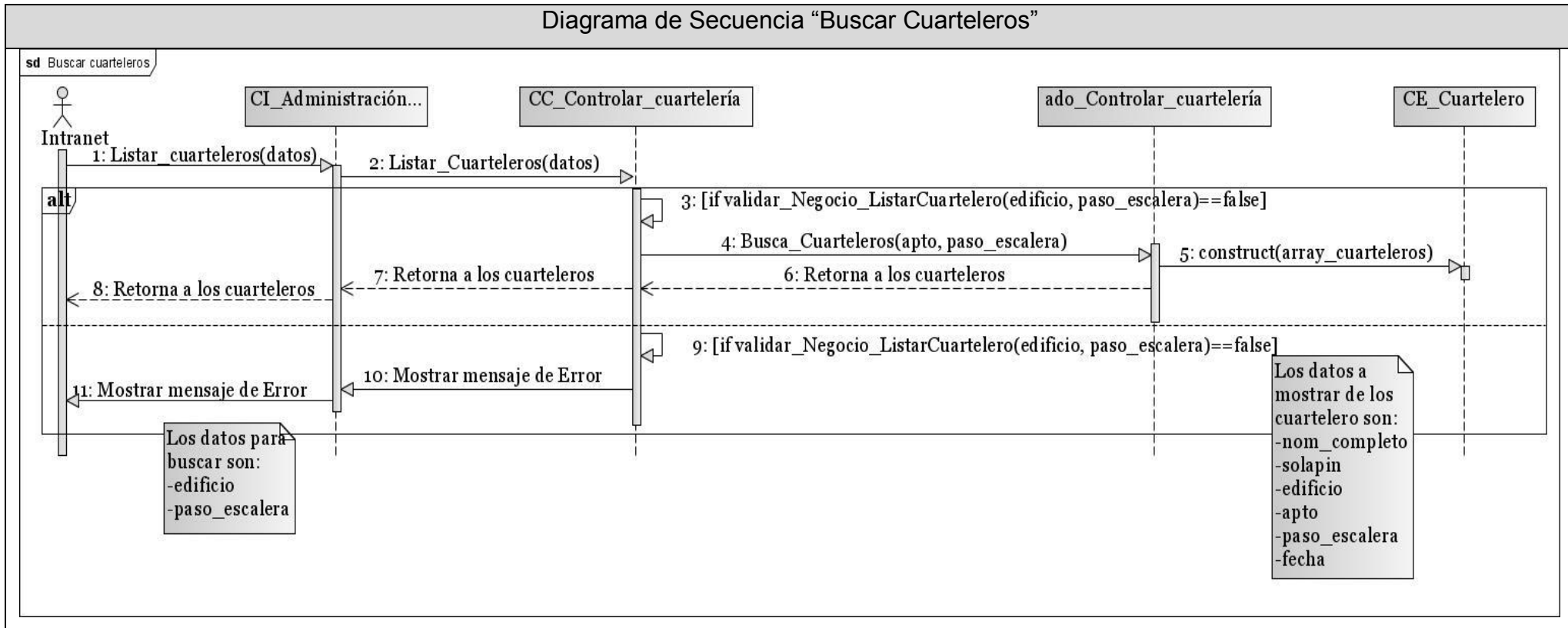
Flujo Alternativo 2	
Acción del Actor	Respuesta del Sistema
	1. Muestra un mensaje diciendo que los estudiantes no existen en la lista de la cuarterería.
Flujo Alternativo 3	
Acción del Actor	Respuesta del Sistema
	1 Muestra un mensaje de error diciendo que existen datos incorrectos.

# Anexo 3



## ANEXO 3

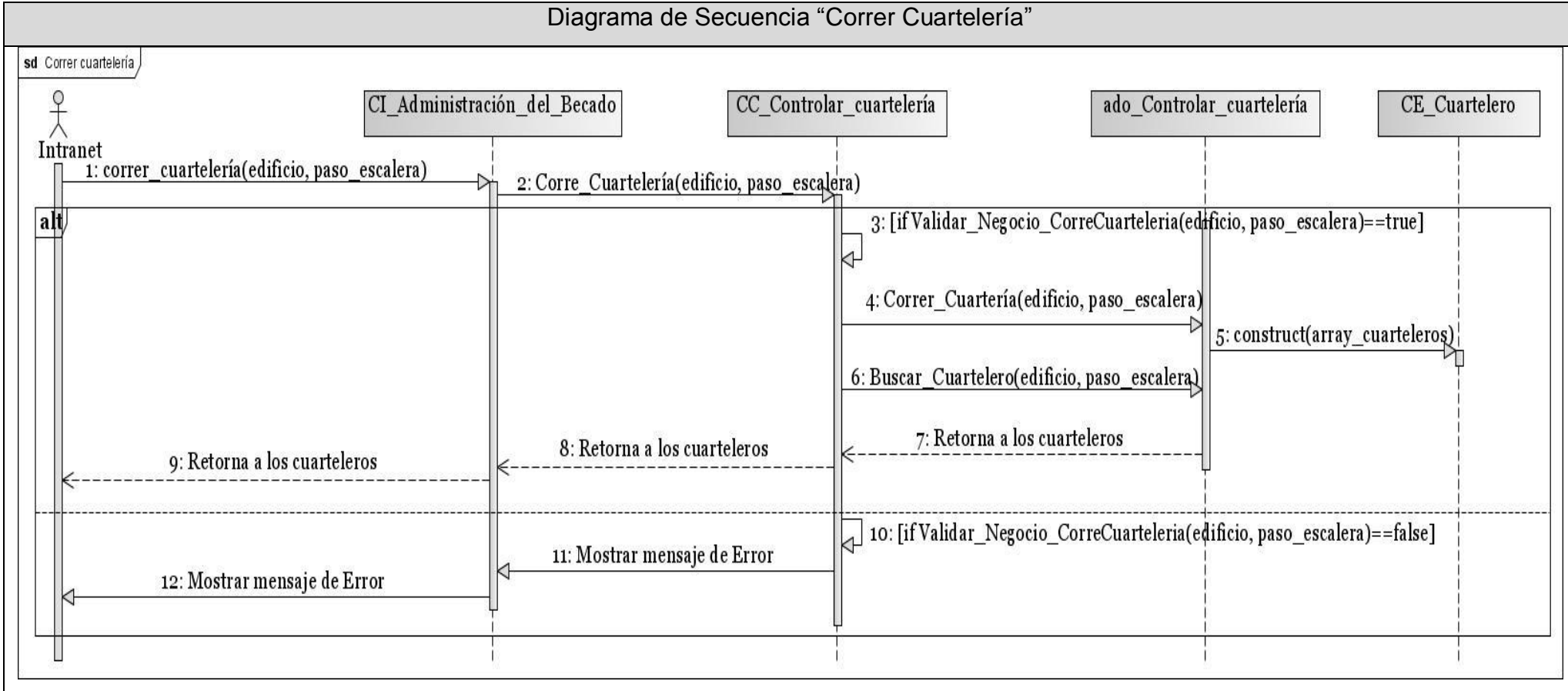
### Diagrama de Secuencia "Buscar Cuartereros"



# Anexo 3



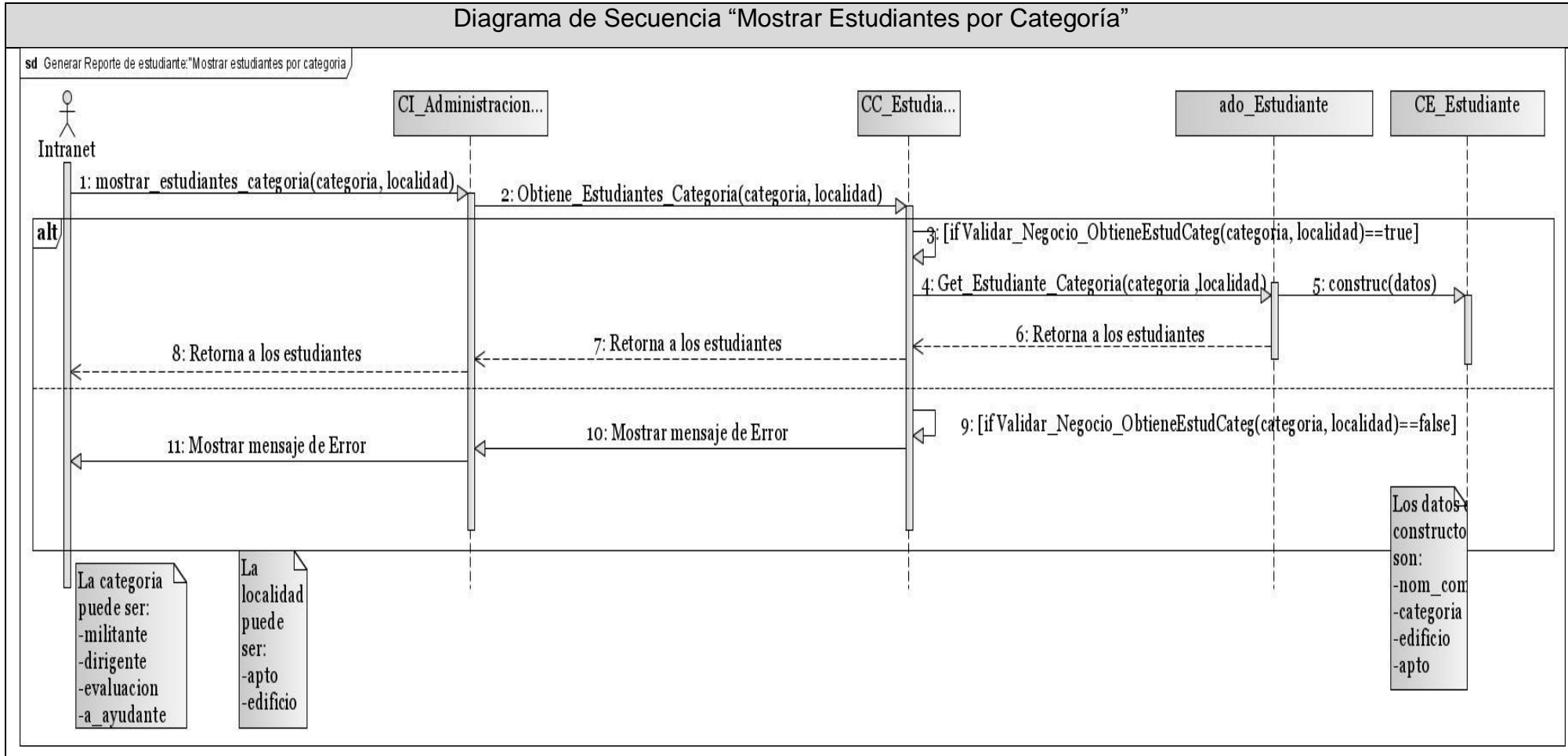
Diagrama de Secuencia "Correr Cuartelería"



# Anexo 3



Diagrama de Secuencia "Mostrar Estudiantes por Categoría"



# Anexo 3

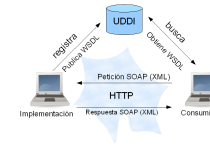
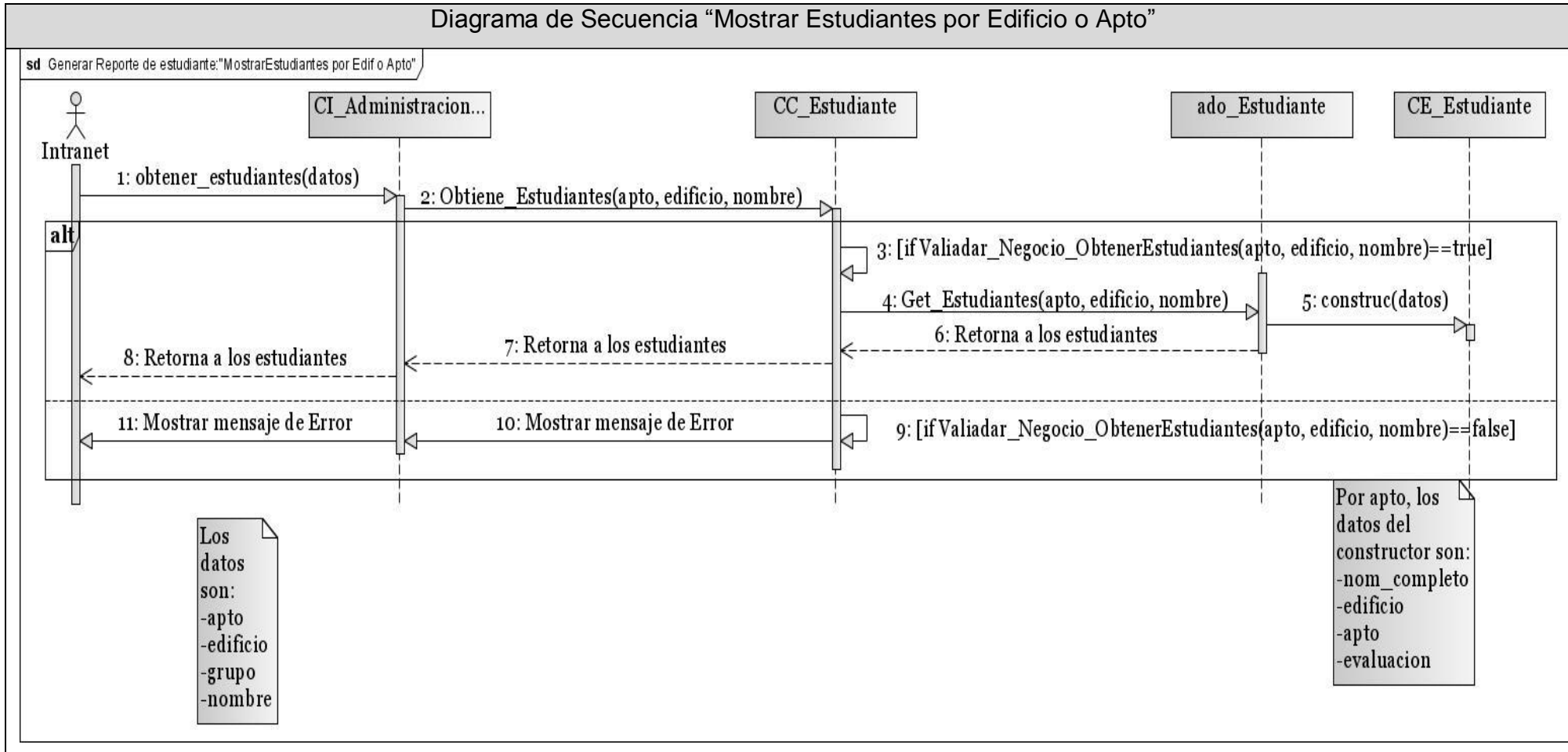


Diagrama de Secuencia "Mostrar Estudiantes por Edificio o Apto"

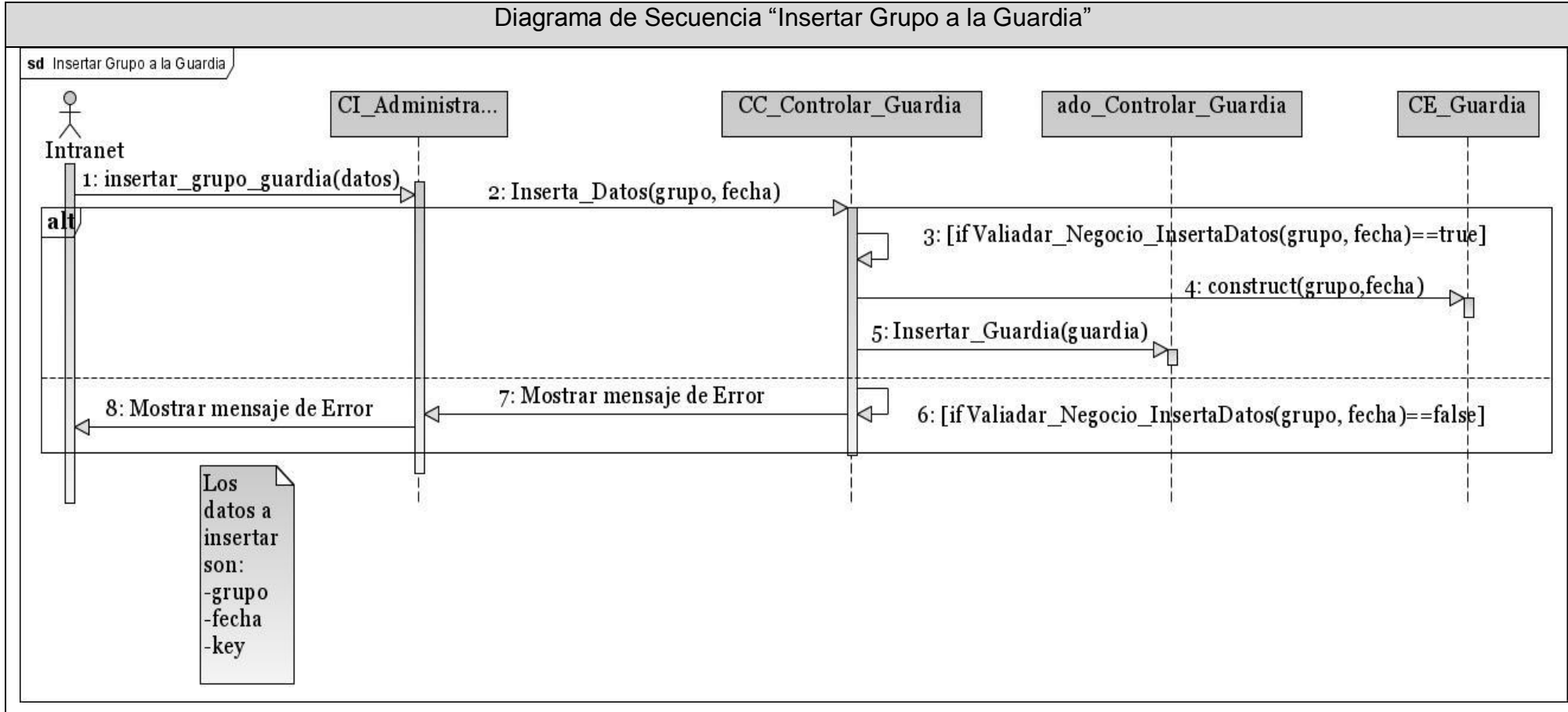


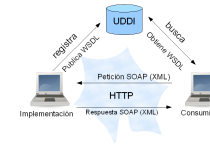


# Anexo 3



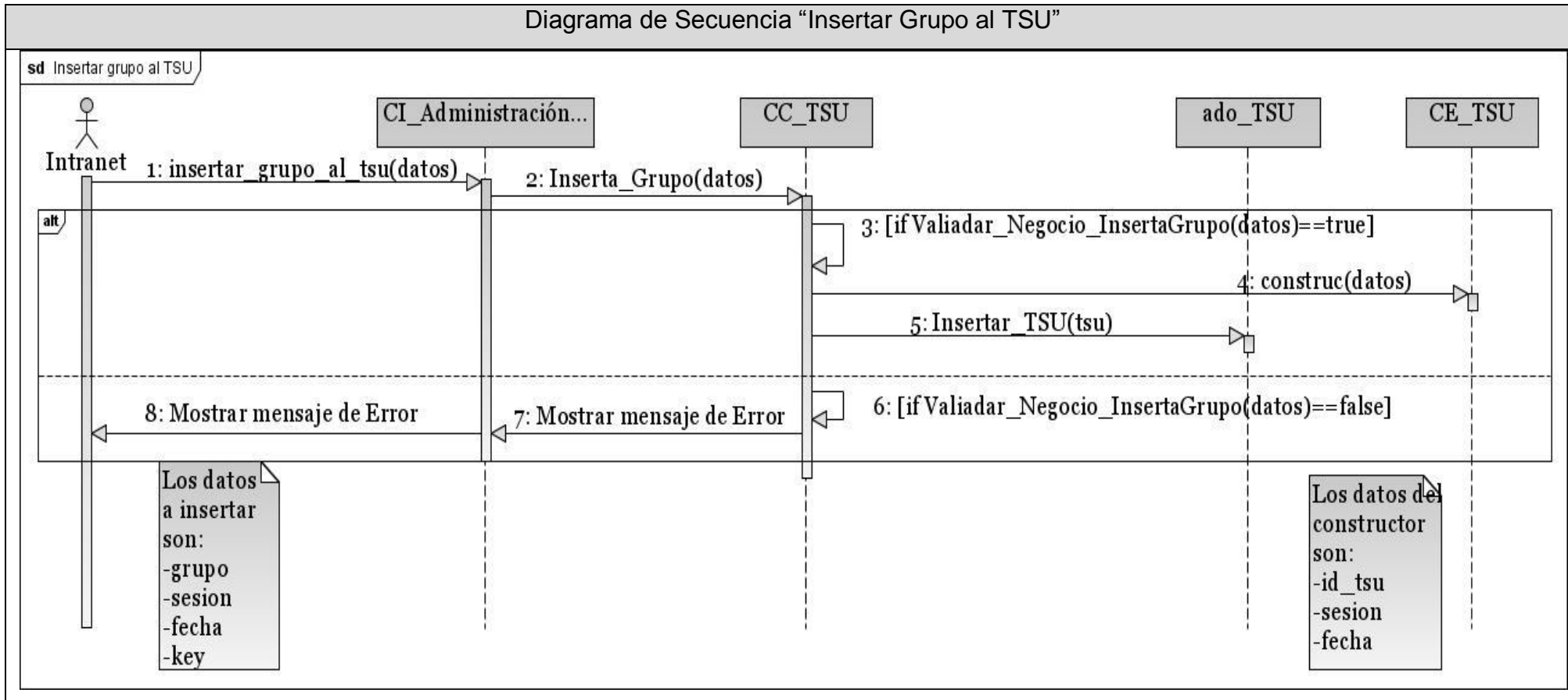
Diagrama de Secuencia "Insertar Grupo a la Guardia"



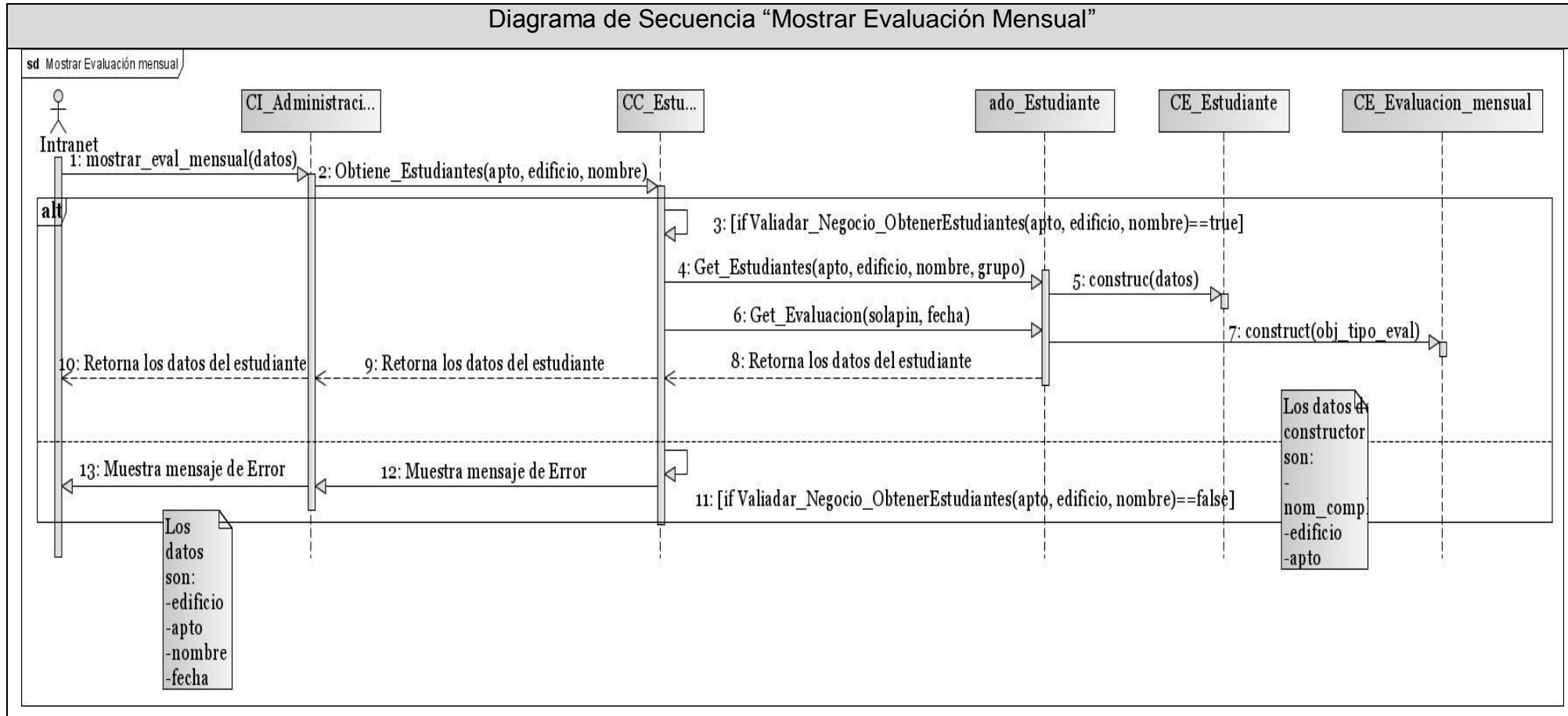


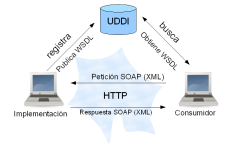
## Anexo 3

Diagrama de Secuencia "Insertar Grupo al TSU"



# Anexo 3





## Glosario de Término

---

### GLOSARIO DE TÉRMINOS

**Cuartelería:** Guardia que realiza el estudiante en el edificio donde reside.

**Cuartelero:** Estudiante que está de cuartería ese día.

**TSU:** Conjunto de actividades que realizan los estudiantes como apoyo a diferentes áreas de la facultad como: el docente, comedor, y áreas verdes de la residencia.

**Guardia Estudiantil:** Guardia nocturna que realizan los estudiantes en la residencia

**Intranet:** Una Intranet, es una red de Área Local. La cual tiene la característica, de ser de uso exclusivo, de la empresa u organización que la ha instalado. Debido a ello, es que utiliza protocolos HTML y el TCP/IP. Protocolos que permiten la interacción en línea de la Intranet, con la Internet.

**Ciente del servicio:** Es el que solicita la ejecución del servicio web, y por lo tanto el que lo consume.

**Proveedor del servicio:** Es el encargado de implementar el servicio web y ofrecerlo a los clientes.

**Registro del servicio:** Es un repositorio donde se almacenan las descripciones de los servicios, para que así los clientes puedan buscar el servicio web que mejor se adapte a sus necesidades.

**HTTP:** Es un sistema de comunicación que permite la visualización de páginas Web, desde un navegador.

**Protocolo:** Un protocolo es un método por el cual dos ordenadores acuerdan comunicarse, una especificación que describe cómo los ordenadores hablan el uno al otro en una red.



## Glosario de Término

---

**Software Libre:** Aplicaciones informáticas que pueden ser libremente copiadas, distribuidas, estudiadas y modificadas por el usuario, según los parámetros establecidos por su creador.

**Módulo:** Es una parte de un programa que realiza partes de las tareas que debe realizar el programa.

**Apto:** Apartamento donde residen los estudiantes.

