

Universidad de las ciencias informáticas.

Facultad 1.



Título: Implementación de un Servicio Web para la
Administración de los departamentos docentes.

Proyecto Intranet Facultad 1.

Trabajo de diploma para optar por el título de ingeniero en ciencias informáticas.

Autores: Héctor Martín Hernández Fariñas.

Marilenys Izquierdo Castillo.

Tutor: Ing. Carlos Tonet Groero.

Cuidad de la habana, Cuba.

Junio, 2008.

DECLARACIÓN DE AUTORÍA

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

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

Héctor M Hernández Fariñas

Marilenys Izquierdo Castillo

Carlos Tonet Groero

Firma del autor

Firma del autor

Firma del tutor

DATOS DE CONTACTO

Tutor: Ing. Carlos T. Groero Carmona (cgroero@uci.cu <mailto:cgroero@uci.cu>).

Graduado de Ingeniero en Ciencias Informáticas en la Universidad de Ciencias Informáticas (UCI), carretera de San Antonio, Bolleros, C. Habana. Desde su graduación se ha desempeñado como profesor de Base de Datos y Ética Informática, a su vez es líder del proyecto Fuerza de Trabajo Calificada, resultado de un contrato de trabajo entre el Ministerio de Economía y Planificación y la UCI.

AGRADECIMIENTOS

Marilenys:

A mi mamá y mi papá porque gracias a ellos se está haciendo realidad este sueño que construimos juntos. Mil Gracias por guiarme siempre por el mejor camino y por ser los mejores padres del mundo, sin ustedes no hubiese logrado esto.

A mi novio Yadier por soportarme en este tiempo de tantas tensiones brindándome su apoyo y cariño.

A toda mi familia, abuelos, hermanos, primos, tíos, parientes cercanos por apoyarme siempre.

A las amistades que he cosechado durante estos 5 años.

A mi compañero de tesis por su dedicación y entrega total a la misma.

A las Martínez de Consolación por la energía positiva que me trasmitían.

Héctor Martín Hernández Fariñas:

A mis padres, por ser los principales responsables de que este gran sueño se haya hecho realidad, sin su apoyo incondicional, amor y entrega seguramente no hubiera podido llegar a donde estoy hoy, gracias por ser los mejores padres del mundo. Todas las palabras no serían suficientes para expresarles cuánto los amo y los admiro. Gracias por estar siempre donde y cuando los necesité.

A mi hermano, que a pesar de las discusiones lo quiero con la vida.

A mi novia Ariagna Rodríguez Donatién, por su comprensión y apoyo durante gran parte de mi carrera, por esta siempre a mi lado, en los momentos buenos y malos, por saber entenderme mejor que nadie.

A mis tías que siempre dicen que yo tengo tres madres, por quererme tanto y estar siempre atentas de mis resultados.

A mis abuelos, que aunque unos están lejos y los otros ya no están, me han ayudado mucho a lo largo de mi carrera y de mi formación como persona.

A toda mi familia en general, tías, primos, parientes cercanos, en fin, a todos muchas gracias por preocuparse siempre por mí, por darme su apoyo incondicional.

A todas mis amistades, pero muy especialmente a Ana Marys y Mirka, dos personas muy especiales para mí que desde hace mucho tiempo me han brindado su apoyo y amistad.

Al resto de mis amigos de manera general. Todos los que he conocido durante la carrera que de una forma u otra me han ayudado a lograr esta meta. Entramos en esta universidad hace cinco años y desde entonces soñábamos con este momento. Muchos han quedado a lo largo del camino y es ahora que más se siente su ausencia. Donde quiera que estén les deseo suerte y éxitos en la vida.

A todos los profesores que a lo largo de mi vida estudiantil han puesto su grano de arena en mi formación como un futuro profesional de esta revolución. Muchas gracias a todos.

A esta universidad, que desde que llegué me acogió y me dio la posibilidad de adquirir los conocimientos que me han hecho la persona que soy hoy.

En fin, a todas aquellas personas que de alguna manera han tenido que ver conmigo a lo largo de mi vida como estudiante, muchas gracias a todos de corazón.

DEDICATORIA

**“La revolución no se lleva en los labios para vivir de ella, se lleva
En el corazón para morir por ella”**

Ernesto Che Guevara.

**“A mis padres, por confiar siempre en mí. Muy especialmente a la Revolución
Cubana y al Comandante en Jefe por haberme dado la posibilidad de formarme
como un profesional de la revolución”.**

Héctor M

**“A veces sentimos que lo que hacemos es tan solo una gota en el mar,
Pero el mar sería menos si le faltara esa gota.”**

Madre Teresa de Calcuta.

A mis padres por su confianza y apoyo incondicional.

A mis abuelos y hermanos por estar siempre pendientes de mí.

A Dadier por estar siempre a mi lado.

Marilenys

RESUMEN:

En los departamentos docentes de la facultad 1 de la Universidad de las Ciencias Informáticas (UCI), la mayoría de los procesos actualmente se ejecutan de manera manual, lo que trae como consecuencias posibles inconsistencias en la información, demoras, ineficiencias, gasto innecesario de recursos humanos y de materiales de oficina.

Como parte del proceso de informatización que se lleva a cabo en esta facultad mediante la implementación de una intranet, se propuso la creación de un módulo para la administración de los departamentos. En este trabajo se presenta la solución de un servicio web que brinda funcionalidades para la informatización de la mayoría de los procesos que tiene lugar en los departamentos docentes de manera eficiente.

En el documento se reflejan los resultados de la investigación realizada. Se explican un conjunto de conceptos y se muestran los artefactos generados en cada etapa de construcción del software. Se exponen también un conjunto de recomendaciones para mejoras futuras del mismo.

INDICE

INTRODUCCIÓN.....	1
1. FUNDAMENTACIÓN TEÓRICA.....	6
INTRODUCCIÓN.....	6
1.1- CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA	6
1.2- DEFINICIONES DE GESTIÓN	8
1.3.- TENDENCIAS ACTUALES DE LAS TECNOLOGÍAS	10
1.3.1- <i>¿Porque un servicio web?</i>	10
1.4- ANÁLISIS DE METODOLOGÍAS Y SOFTWARE	11
1.4.1- SOAP.....	11
1.4.2- NuSOAP: E.....	11
1.4.3- XML.....	11
1.4.4- HTML.....	12
1.4.5- WSDL:.....	12
1.4.6- <i>Editor web orientado a la programación en PHP. Zend Studio</i>	12
1.4.7- <i>Servidor de aplicaciones WEB. Apache</i>	13
1.4.8- PHPMYADMIN.....	13
1.4.9- <i>Lenguaje de programación. PHP</i>	14
1.4.10- <i>Sistema gestor de Base de Datos. MySQL</i>	15
1.4.11- <i>Metodología de desarrollo.RUP</i>	15
1.4.12- UML.....	16
1.4.13- VISUAL PARADIGM.....	17
1.5- ESTADO DEL ARTE.....	18
1.5.1- <i>En el mundo</i>	18
1.5.2- <i>A nivel nacional</i>	20
1.5.3- <i>En la UCI</i>	20
CONCLUSIONES.....	21
2. CARACTERÍSTICAS DEL SISTEMA.....	22
INTRODUCCIÓN.....	22
2.1- ANÁLISIS DE LA SITUACIÓN PROBLÉMICA Y EL PROBLEMA.....	22
2.2- FLUJO ACTUAL Y ANÁLISIS DE LOS PROCESOS DEL NEGOCIO INVOLUCRADOS EN EL CAMPO DE ACCIÓN.....	23
2.3- OBJETO DE INFORMATIZACIÓN.....	24
2.3.1- <i>Descripción de los procesos a informatizar</i>	24
2.3.2- <i>Información que se maneja</i>	25
2.4- PROPUESTA DE SISTEMA.....	25
2.5- REGLAS DEL NEGOCIO.....	26
2.6- MODELO DEL NEGOCIO.....	27
2.7- ACTORES DEL NEGOCIO.....	27
2.8- TRABAJADORES DEL NEGOCIO.....	27
2.9- DIAGRAMA DE CU DEL NEGOCIO.....	28
2.10- DESCRIPCIÓN DE CASOS DE USO DEL NEGOCIO	29
2.11- DIAGRAMA DE ACTIVIDADES	29
2.12- DIAGRAMA DE CLASES DEL MODELO DE OBJETOS.....	36
2.13- ESPECIFICACIÓN DE LOS REQUISITOS	36
2.13.1- <i>Requisitos funcionales</i>	36
2.13.2- <i>Requisitos no funcionales</i>	41
2.14- DEFINICIÓN DE LOS CU DEL SISTEMA	43

2.14.1- Casos de uso del sistema	43
2.14.2- Casos de uso expandidos	49
2.15- ESTUDIO DE LA FACTIBILIDAD	49
CONCLUSIONES.....	55
3. DISEÑO DEL SISTEMA PROPUESTO	56
INTRODUCCIÓN.....	56
3.1- MODELO DE DISEÑO.....	56
3.2- DIAGRAMAS DE INTERACCIÓN.....	56
3.3- DIAGRAMA DE CLASES DEL DISEÑO.....	56
3.4- DESCRIPCIÓN DE CADA UNO DE LOS PAQUETES	57
3.5- DESCRIPCIÓN DE LAS CLASES DEL DISEÑO	58
3.6- DISEÑO DE LA BASE DE DATOS	84
3.7- DESCRIPCIÓN DE LAS TABLAS DE LA BASE DE DATOS	84
CONCLUSIONES.....	90
4. IMPLEMENTACIÓN DEL SISTEMA PROPUESTO.....	91
INTRODUCCIÓN.....	91
4.1- MODELO DE DESPLIEGUE	91
4.2- MODELO IMPLEMENTACIÓN.....	91
4.3- DIAGRAMA DE COMPONENTES EN PAQUETES	92
4.3.1- Diagramas de Componentes separados por paquetes.....	92
4.3.2- Descripción de cada uno de los Paquetes.....	93
CONCLUSIONES.....	93
CONCLUSIONES:.....	94
RECOMENDACIONES.....	95
CITAS BIBLIOGRÁFICAS.....	96
BIBLIOGRAFÍA.....	97

Índice de Tablas.

TABLA2. 1 ACTORES DEL NEGOCIO.....	27
TABLA2. 2 TRABAJADORES DEL NEGOCIO.....	28
TABLA2. 3 ACTORES DEL SISTEMA:	43
TABLA2. 4 DESCRIPCIÓN DEL CU DEL SISTEMA GESTIONAR PROFESOR INTERNO	43
TABLA2. 5 DESCRIPCIÓN DEL CU DEL SISTEMA GESTIONAR PROFESOR EXTERNO.....	43
TABLA2. 6 DESCRIPCIÓN DEL CU DEL SISTEMA GESTIONAR PROFESOR COLABORADOR.....	44
TABLA2. 7 DESCRIPCIÓN DEL CU DEL SISTEMA GESTIONAR ALUMNO AYUDANTE	44
TABLA2. 8 DESCRIPCIÓN DEL CU DEL SISTEMA BUSCAR PERSONA.....	45
TABLA2. 9 DESCRIPCIÓN DEL CU DEL SISTEMA GESTIONAR PLANES DE TRABAJO DM	45
TABLA2. 10 DESCRIPCIÓN DEL CU SISTEMA GESTIONAR ACTIVIDADES	45
TABLA2. 11 DESCRIPCIÓN DEL CU SISTEMA GESTIONAR DEPARTAMENTOS.....	46
TABLA2. 12 DESCRIPCIÓN DEL CU SISTEMA GESTIONAR DISCIPLINAS	46
TABLA2. 13 DESCRIPCIÓN DEL CU SISTEMA GESTIONAR ASIGNATURAS	46
TABLA2. 14 DESCRIPCIÓN DEL CU SISTEMA GESTIONAR GRUPOS.....	47
TABLA2. 15 DESCRIPCIÓN DEL CU SISTEMA GESTIONAR ASIGNACIÓN DE GRUPOS	47
TABLA2. 16 DESCRIPCIÓN DEL CU SISTEMA GESTIONAR POSTGRADOS	48
TABLA2. 17 DESCRIPCIÓN DEL CU SISTEMA GENERAR REPORTES	48

TABLA3. 1 DESCRIPCIÓN DE LA CLASE INTERFAZ: CI_SADE.....	58
TABLA 3. 2 DESCRIPCIÓN DE LA CLASE ENTIDAD: CE_ACTIVIDADES.....	62
TABLA 3. 3 DESCRIPCIÓN DE LA CLASE ENTIDAD: CE_ALUMNO_AYUDANTE.....	63
TABLA 3. 4 DESCRIPCIÓN DE LA CLASE ENTIDAD: CE_ASIGNATURA.....	63
TABLA 3. 5 DESCRIPCIÓN DE LA CLASE ENTIDAD: CE_ASISTENCIA_ACTIVIDADES.....	63
TABLA 3. 6 DESCRIPCIÓN DE LA CLASE ENTIDAD: CE_DEPARTAMENTO.....	64
TABLA 3. 7 DESCRIPCIÓN DE LA CLASE ENTIDAD: CE_DIRECCION.....	64
TABLA3. 8 DESCRIPCIÓN DE LA CLASE ENTIDAD: CE_DISCIPLINA.....	65
TABLA3. 9 DESCRIPCIÓN DE LA CLASE ENTIDAD: CE_PERSONA.....	65
TABLA3. 10 DESCRIPCIÓN DE LA CLASE ENTIDAD: CE_GRUPO_PROFESOR.....	66
TABLA3. 11 DESCRIPCIÓN DE LA CLASE ENTIDAD: CE_GRUPOS.....	66
TABLA3. 12 DESCRIPCIÓN DE LA CLASE ENTIDAD: CE_MESES.....	67
TABLA3. 13 CLASE ENTIDAD: CE_PLAN_ACTIVIDADES.....	67
TABLA3. 14 DESCRIPCIÓN DE LA CLASE ENTIDAD: CE_PLANDM.....	67
TABLA3. 15 DESCRIPCIÓN DE LA CLASE ENTIDAD: CE_POSTGRADO.....	68
TABLA3. 16 DESCRIPCIÓN DE LA CLASE ENTIDAD: CE_PROFESOR.....	69
TABLA3. 17 DESCRIPCIÓN DE LA CLASE ENTIDAD: CE_PROFESORCOLABORADOR.....	69
TABLA3. 18 DESCRIPCIÓN DE LA CLASE ENTIDAD: CE_PROFESOR_EXTERNO.....	69
TABLA3. 19 DESCRIPCIÓN DE LA CLASE ENTIDAD: CE_PROFESORINTERNO.....	70
TABLA3. 20 DESCRIPCIÓN DE LA CLASE ENTIDAD: CE_PROFESORPOSTGRADO.....	70
TABLA3. 21 DESCRIPCIÓN DE LA CLASE CONTROLADORA: DAO_ASIGNATURA.....	71
TABLA3. 22 DESCRIPCIÓN DE LA CLASE CONTROLADORA: DAO_DEPARTAMENTOS.....	71
TABLA3. 23 DESCRIPCIÓN DE LA CLASE CONTROLADORA: DAO_DIRECCION.....	71
TABLA3. 24 DESCRIPCIÓN DE LA CLASE CONTROLADORA: DAO_DISCIPLINA.....	72
TABLA3. 25 DESCRIPCIÓN DE LA CLASE CONTROLADORA: DAO_GRUPO_PROFESOR.....	72
TABLA3. 26 DESCRIPCIÓN DE LA CLASE CONTROLADORA: DAO_GRUPOS.....	72
TABLA3. 27 DESCRIPCIÓN DE LA CLASE CONTROLADORA: DAO_PERSONA.....	73
TABLA3. 28 DESCRIPCIÓN DE LA CLASE CONTROLADORA: DAO_PLANTRABAJO.....	75
TABLA3. 29 DESCRIPCIÓN DE LA CLASE CONTROLADORA: DAO_POSTGRADOS.....	76
TABLA3. 30 DESCRIPCIÓN DE LA CLASE CONTROLADORA: DAO_SELECT.....	76
TABLA3. 31 DESCRIPCIÓN DE LA CLASE CONTROLADORA: CC_ALUMNO_AYUDANTE.....	77
TABLA3. 32 DESCRIPCIÓN DE LA CLASE CONTROLADORA: CC_ASIGNATURA.....	77
TABLA3. 33 DESCRIPCIÓN DE LA CLASE CONTROLADORA: CC_DEPARTAMENTO.....	78
TABLA3. 34 DESCRIPCIÓN DE LA CLASE CONTROLADORA: CC_DIRECCION.....	78
TABLA3. 35 DESCRIPCIÓN DE LA CLASE CONTROLADORA: CC_DISCIPLINA.....	78
TABLA3. 36 DESCRIPCIÓN DE LA CLASE CONTROLADORA: CC_GRUPOS.....	79
TABLA3. 37 DESCRIPCIÓN DE LA CLASE CONTROLADORA: CC_PERSONA.....	79
TABLA3. 38 DESCRIPCIÓN DE LA CLASE CONTROLADORA: CC_PROFESOR.....	80
TABLA3. 39 DESCRIPCIÓN DE LA CLASE CONTROLADORA: CC_PROFESOR_COLABORADOR.....	80
TABLA3. 40 DESCRIPCIÓN DE LA CLASE CONTROLADORA: CC_PROFESOR_EXTERNO.....	81
TABLA3. 41 DESCRIPCIÓN DE LA CLASE CONTROLADORA: CC_PROFESORINTERNO.....	81
TABLA3. 42 DESCRIPCIÓN DE LA CLASE CONTROLADORA: CC_PLANESTRABAJO.....	81
TABLA3. 43 DESCRIPCIÓN DE LA CLASE CONTROLADORA: CC_POSTGRADOS.....	83
TABLA3. 44 DESCRIPCIÓN DE LA CLASE CONTROLADORA: CC_GRUPOPROFESOR.....	83
TABLA 3. 45 DESCRIPCIÓN DE LA TABLA DDISCIPLINA:.....	84
TABLA 3. 46 DESCRIPCIÓN DE LA TABLA DASIGNATURA.....	84

TABLA3. 47 DESCRIPCIÓN DE LA TABLA DDEPARTAMENTO.....	84
TABLA3. 48 DESCRIPCIÓN DE LA TABLA DMESES.....	85
TABLA3. 49 DESCRIPCIÓN DE LA TABLA PLANTRABAJODM.....	85
TABLA3. 50 DESCRIPCIÓN DE LA TABLA DASISTENCIA.....	85
TABLA3. 51 DESCRIPCIÓN DE LA TABLA PLANACTIVIDADES.....	86
TABLA3. 52 DESCRIPCIÓN DE LA TABLA DDIRECCION.....	86
TABLA3. 53 DESCRIPCIÓN DE LA TABLA DPERSONA.....	87
TABLA3. 54 DESCRIPCIÓN DE LA TABLA DGRUPOPROFESOR.....	87
TABLA3. 55 DESCRIPCIÓN DE LA TABLA DGRUPOS.....	87
TABLA3. 56 DESCRIPCIÓN DE LA TABLA DPROFESORPOSTGRADO.....	88
TABLA3. 57 DESCRIPCIÓN DE LA TABLA DPROFESOR.....	88
TABLA3. 58 DESCRIPCIÓN DE LA TABLA DPOSTGRADOS.....	88
TABLA3. 59 DESCRIPCIÓN DE LA TABLA DPROFESOR_EXTERNO.....	89
TABLA3. 60 DESCRIPCIÓN DE LA TABLA DPROFESORCOLABORADOR.....	89
TABLA3. 61 DESCRIPCIÓN DE LA TABLA ALUMNOAYUDANTE.....	89
TABLA3. 62 DESCRIPCIÓN DE LA TABLA DPROFESOR_INTERNO.....	89

Índice de Figuras.

FIGURA 1 DIAGRAMA DE CASO DE USO DEL NEGOCIO.....	29
FIGURA 2 DIAGRAMA DE ACTIVIDADES. GESTIONAR ACTIVIDADES.....	30
FIGURA 3 DIAGRAMA DE ACTIVIDADES. GESTIONAR MIEMBROS.....	31
FIGURA 4 DIAGRAMA DE ACTIVIDADES CURSAR POSTGRADO.....	32
FIGURA 5 DIAGRAMA DE ACTIVIDADES IMPARTIR POSTGRADO.....	33
FIGURA 6 DIAGRAMA DE ACTIVIDADES ELABORAR PLAN DE TRABAJO DM.....	34
FIGURA 7 DIAGRAMA DE ACTIVIDADES GESTIONAR GRUPOS.....	35
FIGURA 8 DIAGRAMA DE CLASES DEL MODELO DE OBJETOS.....	36
FIGURA 9 DIAGRAMA DE CASOS DE USO DEL SISTEMA.....	49
FIGURA 10 DIAGRAMA DE CLASES EN PAQUETES.....	57
FIGURA 11 DIAGRAMA DE DESPLIEGUE.....	91
FIGURA 12 DIAGRAMA DE COMPONENTES EN PAQUETES.....	92

Introducción.

Las Tecnologías de la Información y las Comunicaciones (TIC) ocupan un lugar central en la sociedad y la economía. La necesidad del uso de estas crece cada día más, ya que posibilitan la implementación de sistemas informatizados para facilitar y organizar el trabajo de una determinada entidad. Un ejemplo del uso de las TIC para la organización del trabajo de una institución son las Intranets: una red interna de comunicación e información que emplea la misma tecnología que se utiliza en Internet, siendo accesible únicamente para los usuarios miembros de la organización. El uso de las intranets se está extendiendo hoy en el mundo entre empresas de diversos campos (software, editoriales, finanzas, seguros, salud, etc.) pues estas posibilitan: reducir costes, ahorrar tiempo, centralizar la información, y organizarla mediante páginas web.

La Facultad 1 de la Universidad de las Ciencias Informáticas se ha propuesto la implementación y puesta en práctica de una intranet, que tiene como objetivos la informatización de varios de los procesos que tienen lugar en la misma y la publicación de informaciones importantes para sus estudiantes y profesores, entre otros. Para ello se ha dado la tarea de implementar varios módulos. Uno de ellos está destinado precisamente a la administración de los departamentos docentes, cuyo objetivo es informatizar el proceso de gestión y control de la información dentro de los mismos.

Situación Problemática:

Con la actual división de los departamentos docentes por facultades y siguiendo la estrategia de que los profesores pertenecen a estos y no a los departamentos centrales de las asignaturas, en los departamentos de las facultades se genera un conjunto de actividades con los profesores y alumnos ayudantes (AA) que requieren de un control. También se hace necesario la administración de sus datos personales e informaciones. Este control hasta ahora se lleva de forma manual, generando un gran volumen de información, que requiere de mucho tiempo para organizarla y mantenerla actualizada. Esto también implica un gasto significativo en recursos de oficina.

Tomando en cuenta la situación existente se plantea el siguiente **problema**:

¿Cómo lograr agilizar y organizar la administración de los procesos que ocurren en los departamentos docentes de la facultad 1?

Con el planteamiento del problema anterior se va a definir como **objeto de estudio**: Proceso de administración de los departamentos.

El **campo de acción** es el Proceso de administración y control de los departamentos de la facultad 1.

Con la terminación de este trabajo se espera obtener un producto que sea capaz de responder a las necesidades administrativas de los departamentos de la facultad.

Por lo que podemos definir como **objetivo general**: Implementar un servicio web relacionado con la gestión y el control de los diferentes procesos que tienen lugar en los departamentos docentes de la facultad 1.

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

1. Revisar la bibliografía sobre el tema y elaboración del marco teórico referencial.
2. Realizar un estudio acerca del desarrollo de servicios web usando PHP como lenguaje de programación y MySQL como sistema gestor de Base de Datos.
3. Desarrollar el servicio web en cuestión.
4. Proponer el servicio web para ser implementado de manera experimental.

Para guiar esta investigación se plantea la **hipótesis**: La implementación de un servicio web, permitirá agilizar y organizar el proceso de administración y control de los departamentos docentes de la facultad 1.

Partiendo de la hipótesis se definieron las siguientes **variables**:

Independiente

- Software basado en tecnologías WEB.

Dependientes

- Proceso de administración y control de los departamentos.

Operacionalización de las variables:

Variable Conceptual	Dimensión	Indicadores	UM
Software basado en tecnología WEB.	Factibilidad	Tiempo de desarrollo	Largo Moderado Breve
		Costo	Costoso Moderado Barato
		Esfuerzo	Alto Moderado Despreciable
Proceso de administración y control de los departamentos.	Mejor rendimiento	Complejidad	Alta Media Baja
		Control	Bueno Malo
		Organización del trabajo	Bueno Malo
		Tiempo de Ejecución	Rápido Medio Lento

Para dar cumplimiento a estos objetivos se llevarán a cabo las siguientes **tareas**:

1. Estudio de los antecedentes y el estado del arte de la investigación.
2. Selección de las herramientas necesarias para dar cumplimiento a los requisitos planteados.
3. Diseño del módulo cumpliendo con la arquitectura y los patrones predefinidos en el proyecto.
4. Realización de la implementación del módulo para la administración y control de los departamentos de la facultad 1.
5. Entrega del sistema en el tiempo solicitado.

Para realizar la investigación se tomó como **población**: Los departamentos de la Universidad de las Ciencias Informáticas (UCI).

Se tomó como **muestra** los departamentos de la Facultad 1.

Como **unidad de estudio** el proceso de administración y control de los departamentos.

Como **métodos científicos** utilizamos los métodos teóricos y los métodos empíricos.

Como **métodos teóricos** se emplearon el analítico – sintético, logrando con este una mejor comprensión del trabajo, dividiéndolo en partes para su solución e integrándolas para lograr el resultado final. También se empleó el método hipotético – deductivo ya que se guiará la investigación a partir de la hipótesis planteada. Otro método seleccionado fue el histórico – lógico ya que partiendo de la observación y análisis de los antecedentes del objeto de estudio se dio inicio a esta investigación.

Se emplearon además los **métodos empíricos**. Uno de ellos fue la observación para conocer el avance y desarrollo del sistema.

Como **método particular** se usó la entrevista para conocer exactamente cuáles eran los requisitos necesarios para desarrollar la investigación y para comprender mejor el funcionamiento del negocio.

La culminación e implantación de este software traería consigo los siguientes **aportes prácticos**:

- Interacción fácil y rápida con los datos de los profesores y alumnos ayudantes de cada departamento.
- Informatización de varios de los procesos que ocurren dentro de los departamentos docentes de la facultad.
- Obtención de diferentes reportes de manera eficiente y precisa.

Cronograma de la investigación:

No.	Acciones a realizar	responsable	fecha	
			inicio	Terminación.
1.	Estudiar los antecedentes	Marilenys Izquierdo Héctor Hernández	Noviembre/2007	Diciembre/2007
2.	Realizar el diseño Teórico	Marilenys Izquierdo Héctor Hernández	Diciembre/2007	Enero/2008

3.	Levantamiento de Requisitos	Marilenys Izquierdo	Enero/2008	Febrero/2008
4.	Estudio de las herramientas necesarias para dar cumplimiento a los requisitos planteados.	Marilenys Izquierdo Héctor Hernández	Febrero/2008	Marzo/2008
5.	Llevar a cabo el diseño del módulo.	Marilenys Izquierdo Héctor Hernández	Marzo/2008	Abril/2008
6.	Realizar la implementación del módulo	Marilenys Izquierdo Héctor Hernández	Abril /2008	Junio/2008

El documento estará estructurado en cuatro capítulos y las correspondientes conclusiones, recomendaciones, referencia bibliográfica, bibliografía, anexos y glosario de términos.

Capítulo 1. Fundamentación Teórica: Se describen las tendencias actuales de las tecnologías y se hace un análisis de las herramientas seleccionadas para la implementación del sistema propuesto.

Capítulo 2. Características del sistema: Se hace una descripción de cómo se realiza el proceso objeto de estudio de nuestra investigación. Se exponen los artefactos que permiten comprender la estructura y la dinámica del proceso abordado según la metodología utilizada. Se presentan las funcionalidades que tendrá el sistema y se definen los casos de uso.

Capítulo 3. Análisis y diseño del sistema: Se exponen los diagramas de Clases del diseño, los diagramas de secuencias, el diagrama Entidad Relación de la BD, la descripción de las tablas y se explican de manera general las clases utilizadas.

Capítulo 4. Implementación del sistema: se describe el sistema que se propone a través de una óptica de programación, basada en diagramas de componentes y diagrama de despliegue.

CAPÍTULO

I

1. Fundamentación teórica.

Introducción.

En el presente capítulo haremos un análisis del fundamento teórico de la investigación. Entiéndase análisis de las herramientas a usar para el desarrollo de la solución, ventajas de cada una, porque fueron esas las seleccionadas, y cuales otras podrían haberse utilizado. Veremos de igual manera otros software con características iguales o similares desarrollados y usados dentro o fuera del país, siempre desde un punto de vista crítico. Especificaremos también una breve descripción de los conceptos más importantes asociados al dominio del problema.

1.1- Conceptos asociados al dominio del problema.

Profesor: Persona que ejerce o enseña una ciencia o arte.

Profesor adjunto: En los institutos de bachillerato y en las universidades profesor normalmente adscrito a una determinada cátedra o departamento.

Alumno ayudante:

Alumno: Discípulo respecto de su maestro, de la materia que está aprendiendo o de la escuela, colegio o universidad donde estudia.

Ayudante: Persona que auxilia a un especialista en determinada rama en algunas labores.

Alumno ayudante: Estudiante con conocimientos avanzados en determinadas materias que auxilia al profesor en algunas actividades docentes. En la UCI se encarga de impartir clases prácticas y consultas, realizar evaluaciones, entre otras tareas.

Tecnología WEB:

Tecnología:

1a. Definición de Tecnología: Conjunto de los conocimientos propios de una técnica.

2a. Definición de Tecnología: Conjunto de instrumentos, procedimientos o recursos técnicos empleados en un determinado sector o producto.

Web: World Wide Web (o la "Web") es un sistema de documentos de hipertextos y/o hipermedios enlazados y accesibles a través de internet. Con un navegador web, un usuario visualiza páginas web que pueden contener textos imágenes videos u otros contenidos multimedia, y navega a través de ellas usando hiperenlaces. [1]

Tecnología Web: Conjunto de conocimientos, instrumentos, procedimientos y recursos usados para crear un sitio WEB.

Software libre: Es la denominación del software que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. Según la Free Software Foundation, el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso se refiere a cuatro libertades de los usuarios del software:

- La libertad de usar el programa, con cualquier propósito.
- La libertad de estudiar el funcionamiento del programa, y adaptarlo a sus necesidades.
- La libertad de distribuir copias, con lo que puede ayudar a otros.
- La libertad de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie.[2]

Intranet: Una **Intranet** es una red de ordenadores privada basada en los estándares de Internet.

Las Intranets utilizan tecnologías de Internet para enlazar los recursos informativos de una organización, desde documentos de texto a documentos multimedia, desde bases de datos legales a sistemas de gestión de documentos. Pueden incluir sistemas de seguridad para la red, tableros de anuncios y motores de búsqueda.

Una Intranet puede extenderse a través de Internet. Esto se hace generalmente usando una red privada virtual (VPN).

Gestión.

Desde hace mucho tiempo el ser humano se planteó la necesidad de controlar y administrar todo lo que poseía, en un principio de manera natural se designaba a un cacique que se encargaba de organizar la distribución de los alimentos, pero con el paso del tiempo y el surgimiento de estructuras sociales y productivas más complejas no fue suficiente con una persona. En la actualidad no son solo las grandes empresas y organizaciones las necesitadas de gestionar varias cosas (Recursos, personal, etc.). Un ejemplo de ello es el presente trabajo de diploma en cual se propone una solución para informatizar los procesos relacionados con la gestión de los departamentos docentes de nuestra facultad.

1.2- Definiciones de gestión.

1ª definición de gestión: Es el proceso mediante el cual se obtiene, despliega o utiliza una variedad de recursos básicos para apoyar los objetivos de la organización. [3]

2ª definición de gestión: Proceso emprendido por una o más personas para coordinar las actividades laborales de otros individuos. [4]

3ª definición de gestión: Gestionar es coordinar todos los recursos disponibles para conseguir determinados objetivos, implica amplias y fuertes interacciones fundamentalmente entre el entorno, las estructuras, el proceso y los productos que se desean obtener. [5]

Objetivos del proceso de gestión.

Planificar: Es el proceso de establecer objetivos con el fin de alcanzar determinados resultados, así como, identificar las acciones necesarias para alcanzarlos. Dentro de este concepto se contemplan un conjunto de decisiones o una selección de alternativas para el logro de tales resultados. Esta función se registra en diversos tipos de documentos: planes, programas, pronósticos y políticas.

Organizar: Es el proceso de dividir el trabajo a realizar y de coordinar el logro de resultados que tienen un propósito común. Organizar es la química de la organización donde se mezclan todos los elementos que interactúan entre sí a fin de obtener los

resultados esperados. Es el acto de combinar habilidades, posibilidades técnicas, experiencias, recursos y todos los elementos que podrían convertirse en resultados.

Dirigir: Es el proceso de conducir y coordinar los esfuerzos laborales de las personas que integran una organización, ayudándolos a desarrollar tareas relevantes dentro de ella. La dirección es la función mediante la cual se ponen en marcha las actividades programadas. Comprende el compromiso de alcanzar un objetivo mediante el liderazgo de un grupo. La dirección ejerce una influencia notable en las personas para que trabajen voluntaria y entusiastamente para el logro de las metas colectivas de equipos y de la organización en su conjunto.

Controlar: Es el proceso de supervisar las actividades y resultados, comparándolos con los objetivos y tomando las acciones correctivas, si son necesarias. Para ello se compara el desempeño con metas y planes, se muestran las desviaciones y al emprender medidas para corregir las desviaciones, se ayuda a asegurar el logro de los planes. Esta función comprende el establecimiento de normas de desempeño como base para la medida de los resultados, investigación, análisis, diseño, implantación y operación de los sistemas de información, registros contables y estadísticos, auditorías, inspecciones, controles y otros métodos de verificación directa.

Necesidad de los Sistemas de gestión de Información.

Con los adelantos tecnológicos actuales en el mundo de la informática y las comunicaciones, es prácticamente imposible que una organización no haga uso de los mismos para el desarrollo de sus actividades cotidianas, pues constituyen un elemento fundamental para alcanzar un alto nivel de competitividad y posibilidades de desarrollo.

Cuando las organizaciones son grandes y complejas, poseen grandes cantidades de usuarios y niveles, la gestión de la información requiere el tratamiento, almacenamiento y difusión de grandes volúmenes de datos. En estos casos los métodos manuales para gestionar adecuadamente la información y responder con rapidez y agilidad a las diversas solicitudes, así como optimizar la comunicación con los demás sectores de la institución, no son viables.

En estas situaciones la generación de información confiable, precisa y oportuna, con el objetivo de facilitar el análisis y proporcionar elementos necesarios para una buena planificación requiere de sistemas de gestión información, eficaces y eficientes.

1.3.- Tendencias actuales de las tecnologías.

En el mundo actual existen publicados en la red una gran cantidad servicios instalados en un servidor y que pueden ser accedidos desde cualquier lugar del mundo. En los últimos tiempos se ve como una tendencia en el mundo el uso de web services o servicios web, pues estos de una manera muy sencilla posibilitan la administración de muchos procesos dentro de una determinada empresa o institución.

Un servicio web es un software diseñado para soportar la interoperabilidad máquina - máquina a través de una red. Posee una interfaz descrita en un formato que puede ser procesado por una computadora (específicamente WSDL). Otros sistemas pueden interactuar con el servicio web utilizando mensajes SOAP previamente establecidos.

Hay que tener en cuenta que un servicio Web no es una página con la que el usuario interactúa, un servicio web da servicios a aplicaciones y no a las personas ya que lo que hace es recibir solicitudes (mensajes XML codificados) desde una aplicación, ejecuta una tarea y devuelve como respuesta también un mensaje formateado en XML. Un mismo servicio web puede ser consumido tanto por aplicaciones que se ejecutan en un servidor y entregan código HTML al cliente, como por aplicaciones que corren directamente en la máquina del cliente.

1.3.1- ¿Porque un servicio web?

Se decidió la implementación de un servicio web teniendo en cuenta que:

- Permiten compartir datos y pueden ser consumidos desde diferentes plataformas y sistemas operativos, sin tener en cuenta el lenguaje de programación, brindando la posibilidad de gestionar cualquier tipo de información.
- Proporcionan una solución viable para habilitar la interoperabilidad de datos y sistemas. Además ofrecen funciones muy útiles a usuarios del medio web, ya que emplean un protocolo estándar (SOAP).
- Permiten describir sus interfaces detalladamente para que el usuario diseñe una aplicación cliente que capaz comunicarse con ellos y permiten ser

registrados para poder ser localizados fácilmente. Dicho registro es a través de UDDI (descripción, descubrimiento e integración universales).

- Utilizan mensajería basada en XML como medio fundamental de comunicación de datos para contribuir a reducir las diferencias existentes entre diferentes entornos de trabajo. Un servicio web puede ser implementado empleando cualquier lenguaje de programación. Entre los múltiples lenguajes se encuentran: ASP.Net, JSP (*Java Server Pages*), PHP, Python, Perl, entre otros.

1.4- Análisis de metodologías y software.

1.4.1- SOAP (Siglas de Simple Object Access Protocol): Es un protocolo estándar creado por IBM y Microsoft. Está actualmente bajo el auspicio de la W3C. Define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. SOAP es uno de los protocolos utilizados en los servicios Web. [7]

1.4.2- NuSOAP: Es un kit de herramientas (ToolKit) para desarrollar Web Services usando PHP. Está compuesto por una serie de clases que hacen 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[7]

1.4.3- XML: Sigla en inglés de **Extensible Markup Language** («lenguaje de marcas extensible»). Es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos. Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Algunos de los lenguajes que usan XML para su definición son XHTML, SVG, MathML.

XML no ha nacido solo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

Es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

1.4.4- HTML: Sigla de HyperText Markup Lenguaje (Lenguaje de Etiquetas de Hipertexto): Es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de "etiquetas", rodeadas por corchetes angulares (<,>). También puede describir, hasta un cierto punto, la apariencia de un documento y puede incluir un script (por ejemplo una función Java script), que puede afectar el comportamiento de los navegadores web y otros procesadores de HTML.

1.4.5- WSDL: Son las siglas de Web Services Description Language. Es un formato XML que se utiliza para describir servicios web (se lee wisdel).

WSDL describe la interfaz pública de 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 a menudo 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. Los tipos de datos especiales se incluyen en el archivo WSDL en forma de XML Schema. El cliente puede usar SOAP para hacer la llamada a una de las funciones listadas en el WSDL.

1.4.6- Editor web orientado a la programación en PHP. Zend Studio.

Zend Studio: Decidimos usar Zend Studio ya que se trata de un IDE de desarrollo orientado a PHP con una gran cantidad de opciones y posibilidades. Es en la actualidad uno de los más usados a nivel mundial.

Zend estudio Consta de dos partes que se dividen en: Funcionalidades de parte del cliente y funcionalidades de parte del servidor. Las dos partes se instalan por separado.

Contiene una ayuda contextual con todas las librerías de funciones del lenguaje que asiste en todo momento ofreciendo los nombres de las funciones y los parámetros que deben recibir. Permite editar varios archivos, y moverse fácilmente entre ellos, marcar a qué elementos corresponden los inicios y cierres de las etiquetas, paréntesis o llaves, moverse al principio o al final de una función, indentación automática del código, etc.

Para su selección nos basamos fundamentalmente en que el Zend Studio se distribuye bajo la licencia de Open Source.

1.4.7- Servidor de aplicaciones WEB. Apache.

Apache es un servidor web gratuito desarrollado por el apache server Project cuyo objetivo es la creación de un servidor web seguro, eficiente y fácilmente extensible con código fuente abierto. Este proyecto es manejado además por un grupo de voluntarios localizados en distintas partes del mundo que a través de internet desarrollan y mejoran el servidor y la documentación relacionada con este.

Ventajas de apache:

- 1- Su licencia: esta es de código abierto del tipo BSD que permite el uso comercial y no comercial del Apache.
- 2- Una talentosa comunidad de desarrolladores siguiendo un proceso abierto de desarrollo.
- 3- Arquitectura modular: Los usuarios de Apache pueden adicionar fácilmente funcionalidad a sus ambiente específicos.
- 4- Portabilidad: Apache trabaja sobre todas las versiones recientes de Windows, Linux y Unix.
- 5- Es robusto y seguro.

1.4.8- PHPMYADMIN.

PHPMYADMIN: Es un programa de distribución libre, creado por una comunidad sin ánimo de lucro, que sólo trabaja en el proyecto por amor al arte. Es una herramienta

muy completa que permite acceder a todas las funciones típicas de una base de datos MySQL a través de una interfaz web muy intuitiva.

Este programa presenta varias ventajas que nos motivaron a seleccionarlo para su utilización. Algunas de ellas son:

- Es totalmente gratuito (open source)
- Es muy completo: Permite crear, borrar y modificar las tablas
- Permite administrar los permisos a las personas y ejecutar comandos SQL.

1.4.9- Lenguaje de programación. PHP.

Php Es un lenguaje de programación interpretado usado normalmente para la creación de páginas web dinámicas. PHP es un acrónimo recursivo que significa "PHP Hypertext Pre-processor" (inicialmente PHP Tools o Personal Home Page Tools). Actualmente también se puede utilizar para la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+. Es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor.

Php presenta muchas ventajas que lo hacen atractivo y que nos motivaron a usarlo como lenguaje de programación para desarrollar el trabajo de curso. Entre las principales ventajas que presenta encontramos:

- Muy sencillo de aprender.
- Similar en sintaxis a C y a PERL
- Soporta la POO.
- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad (InterBase, mSQL, MySQL, Oracle, Informix, PostgreSQL, entre otros)
- Permite leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.

Otros lenguajes como Perl (Practical Extraction and Report Language), ASP (Active Server Pages) y JSP (Java Server Pages) tienen características similares al PHP aunque poseen rasgos que los marcan y distinguen. Pero analizando por ejemplo que la velocidad de ejecución PHP es mucho mayor, que tiene una mayor disponibilidad de recursos en la actualidad en Internet, que tiene un soporte sólido y real para la Programación Orientada a Objetos, mejor soporte para MySQL con extensión completamente reescrita, y mejor soporte para XML hemos seleccionado PHP como lenguaje de programación.

1.4.10- Sistema gestor de Base de Datos. MySQL.

SGBD: Un Sistema Gestor o Manejador de Bases de Datos (SGBD) es un conjunto de programas que permite a los usuarios crear y mantener una BD. Es un software de propósito general que facilita el proceso de definir, construir y manipular la BD para diversas aplicaciones. Se compone de un lenguaje de definición de datos, un lenguaje de manipulación de datos y de un lenguaje de consulta.

MySQL es la base de datos open source más popular en los últimos tiempos. Es un sistema de gestión de base de datos relacional, multihilo y multiusuario. La versión 5 de MySQL, incorpora funcionalidades muy importantes, como los procedimientos almacenados, nuevos motores, mejoras en tipos de datos (por ejemplo el VARCHAR) y optimizaciones en operaciones de búsqueda de datos.

Algunas de las principales ventajas que presenta MySQL como SGBD son por ejemplo que tiene un rendimiento alto, múltiples utilidades de administración, integración perfecta con PHP (que por demás es el lenguaje de programación a usar), mecanismos para un mejor control de acceso de usuarios, y tamaño ilimitado en los registros. Todo esto unido al hecho de que es una tecnología bajo la licencia de open source, hizo que eligiéramos MySQL como SGBD para la solución a implementar.

1.4.11- Metodología de desarrollo.RUP.

Con el desarrollo de las tecnologías y la creación cada día de proyectos de gran envergadura que necesitan de una buena planificación y gestión, han surgido en el mundo varias metodologías para guiar el proceso de desarrollo de software. La mayoría de estas metodologías están orientadas al paradigma más importante de la

programación en los últimos años: el paradigma orientado a objetos. Algunas de estas metodologías son Rational Unified Process (RUP), Extreme Programming (XP), OPEN, MÉTRICA 3, Microsoft Solution Framework (MSF) entre otras.

De todas estas metodologías la que hemos seleccionado para guiar el proceso de desarrollo del sistema es RUP. La hemos seleccionado debido a que es una metodología que:

- Proporciona una guía para el orden de todas las actividades de un equipo de desarrollo de software.
- Se encarga de dirigir las tareas de cada desarrollador por separado y del equipo como un todo.
- Especifica los artefactos que deben desarrollarse.
- Ofrece criterios para el control y la medición de los productos y las actividades del proyecto.

Entre sus principales características tenemos:

- Centrado en los Modelos: Los diagramas son un vehículo de comunicación más expresivo que las descripciones en lenguaje natural. Se trata de minimizar el uso de descripciones y especificaciones textuales del sistema.
- Guiado por los casos de uso: Los casos de uso son el instrumento para validar la arquitectura del software y extraer los casos de prueba.
- Centrado en la arquitectura: Los modelos son proyecciones del análisis y el diseño constituye la arquitectura del producto a desarrollar.
- Iterativo e incremental: Durante todo el proceso de desarrollo se producen versiones incrementales (que se acercan al producto terminado) del producto en desarrollo.

1.4.12- UML.

Como lenguaje de modelado de usará UML. Este es un lenguaje gráfico que permite visualizar, especificar y documentar cada una de las partes que comprende el desarrollo del software. Permite modelar elementos conceptuales como son los

procesos de negocio, las funciones de sistema, clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables, así como distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. Dentro de sus objetivos tiene: Brindar un material de apoyo que le permita al lector poder definir diagramas propios como también entender diagramas ya existentes. Prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan.

1.4.13- VISUAL PARADIGM.

Es una herramienta CASE que utiliza UML como lenguaje de modelado. Es básicamente una herramienta diseñada para cumplir las mismas funcionalidades que el Rational Rose, con la diferencia de que es compatible con sistemas Linux y está registrada bajo la licencia de software libre. Algunas de las principales ventajas que presenta son:

- Presenta un diseño centrado en casos de uso y enfocado al negocio.
- Usa un lenguaje estándar común a todo el equipo de desarrollo y facilita la comunicación.
- Tiene capacidades de ingeniería directa (versión profesional) e inversa.
- Tiene modelos y códigos que permanecen sincronizados en todo el ciclo de desarrollo.
- Presenta disponibilidad de múltiples versiones, para cada necesidad y múltiples plataformas.[6]

La razón fundamental por la cual lo seleccionamos para el moldeamiento del sistema además de las muchas facilidades que brinda es que se trata de una herramienta case muy poderosa cuya distribución es libre.

Está disponible en varias ediciones, cada una destinada a diferentes tipos de necesidades: Enterprise, Professional, Community, Standard, Modeler y Personal.

Nota aclaratoria: Las herramientas de desarrollo especificadas anteriormente fueron seleccionadas teniendo en cuenta todas las ventajas y características especificadas además de ser las que se utilizan en el proyecto. Su uso fue orientado por la dirección del mismo.

1.5- Estado del arte.

1.5.1- En el mundo.

SIGA (Software Integrado de Gestión Académica Web): Es un sistema modular para la administración académica y curricular, diseñado especialmente para instituciones de educación superior. Funciona completamente a través de internet e integra tanto datos como procesos en una solución completa, eliminando barreras de espacio y tiempo. SIGA cumple a cabalidad con las características necesarias que garantizan su calidad. Ellas son: funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad.

Como SIGA es un producto diseñado para interactuar a través de la Internet, permite compartir información de manera eficiente y segura entre dependencias, evitando los problemas de inconsistencia originados por la redundancia de información. Las interfaces para los usuarios son amigables y tienen uniformidad en su presentación induciéndole a navegar por el sistema, facilitando a los mismos enfrentarse a las nuevas tecnologías de la informática con ánimo y confianza.

Debemos señalar que este software no está solo enfocado a la gestión de los profesores, sino que también se centra en la informatización de los procesos académicos. Este sistema fue desarrollado por la Universidad Técnica Federico Santa María.

Profesorado V 1.0: Este software dispone de múltiples funciones para registrar los datos de los estudiantes de manera organizada. Gestiona toda la información relacionada con las materias de enseñanza impartidas, datos de los profesores, alumnos, asignaturas, cursos, notas, horarios, anotaciones personales, etc.

Permite configurar de forma totalmente personalizada los datos introducidos. Tiene funcionalidades para la modificación de tablas (colores, anchos, columnas y filas), modificación de notas y datos personales, entre otras.

Al igual que el SIGA no solo se centra en la gestión de los profesores, sino también en la gestión académica, pero aun así consideramos que constituye un buen antecedente para nuestra investigación.

Virtuali eSchool™: Es una herramienta muy útil con funcionalidades que permiten gestionar un centro de educación de manera sencilla y ágil. Entre sus funcionalidades se haya la gestión de los profesores por lo que lo consideramos un antecedente de nuestra investigación y la solución que desarrollaremos para nuestro proyecto de tesis. Sus facilidades se pueden dividir en varias categorías:

Para la Administración:

ESchool le proporciona a la administración una precisa base de datos de todos los estudiantes, los profesores y todo el material educativo que se halla disponible en la escuela. Le permite observar el desarrollo y desempeño de cada estudiante, así como mantener todos los datos filiales de los mismos. También permite comunicarse con los padres, profesores y estudiantes a través de un sistema de correo electrónico interno.

Para los Profesores o Maestros:

ESchool es una herramienta desarrollada para que los profesores hagan más productivo su tiempo y su trabajo más eficiente. Con las herramientas de evaluación que posee el profesor podrá evaluar en forma individual el rendimiento de cada estudiante, así como administrar sus exámenes, pruebas y hojas de trabajo en forma más eficiente y rápida. ESchool contiene herramientas que le ofrecen de forma automática calificaciones e información a nivel estadístico. Con estas herramientas, el profesor puede crear fácilmente el contenido educativo que desee, manteniéndose en todo momento en contacto con sus estudiantes, comunicando sus ideas y diferentes puntos de vista, todo esto a través de los sistemas internos de "chat" y correo electrónico.

Para los Estudiantes:

Todo el contenido educativo está disponible para el estudiante, ayudándole a mantenerse al día y actualizado con los diferentes cursos y materias que le son asignados. Los estudiantes podrán tener acceso a esta información suministrada por los profesores, podrán Interactuar con otros estudiantes y con sus profesores a los efectos de evacuar preguntas o dudas. ESchool permite tener acceso a ejercicios domiciliarios, hojas de trabajo, pruebas "online" que luego de ser corregidas automáticamente, le serán devueltas al estudiante para su auto evaluación.

DocCF: Software de Gestión Escolar. Tiene incluido funcionalidades para la gestión de alumnos, registro de matriculas, gestión de profesores, evaluación de docencia, asignación de horarios, administración de la biblioteca, registro de calificaciones, generación de boletines escolares, envío de emails, mensajes de texto, control de pagos y pensiones, inventario escolar, control de ausentismo, estadísticas, listados e informes, etc. Todo en una sola aplicación.

Como se ve se trata de un software de propósito general que incluye muchas funcionalidades. Lo consideramos dentro de los antecedentes de nuestra investigación debido a que incluye también un modulo dedicado a la gestión de profesores.

1.5.2- A nivel nacional.

Sistema de gestión de la información de un Departamento docente: Desarrollado en la universidad Marta Abreu de Villa Clara es un sistema para realizar la gestión de la información en un departamento docente. Surge teniendo en cuenta la necesidad del intercambio de información, rápido acceso y nivel de actualización. Se desarrolla en un ambiente Web utilizando PHP e Interbase. Está constituido por cuatro módulos: • Módulo de Capital Humano • Módulo de Formación del Profesional • Módulo de Educación de Postgrado • Módulo de Ciencia y Técnica. Estos módulos facilitan la gestión de la información del Capital Humano de un Departamento Docente. Dentro de ellos quedan registrados los datos personales de los trabajadores del Departamento, cumplimiento del Plan de Trabajo, los objetivos del profesor, su evaluación, etc. Se controlan los datos referentes a sus asesoramientos a otros profesores. También se desarrolla todo el trabajo de planificación de la carga docente de los profesores.

1.5.3- En la UCI.

Sistema Automatizado para la Gestión Académica "Akademos": Se caracteriza por su gran dinamismo, forma de adaptarse a los cambios de la nueva universidad y la incorporación de todos los implicados en la gestión académica a la misma (estudiantes, profesores, secretarias, docente, directivos y otros). Este sistema hoy en día es funcional y estable, su base de datos es grande y debe atender gran cantidad de peticiones simultáneas, pues el sistema está disponible a toda la comunidad universitaria, la cual puede acceder desde cualquier lugar del centro al mismo. La

misión principal de este sistema es la gestión y administración de todos los datos de alumnos y profesores de nuestra universidad.

Conclusiones.

En este capítulo se realizó un análisis de las tendencias actuales de las tecnologías. Se describieron las principales herramientas a utilizar para la implementación de la solución que se propone y se abordaron un conjunto de conceptos asociados al dominio del problema que pudieran ser de ayuda a los interesados en conocer más a fondo la presente investigación.

CAPÍTULO

II

2. Características del sistema.

Introducción.

En el presente capítulo se realizará un análisis del funcionamiento del negocio a informatizar como parte del primer flujo de trabajo en el desarrollo de cualquier software. Se definirán los procesos a informatizar así como los actores, trabajadores y casos de uso del negocio. También se definirán los requisitos funcionales y no funcionales de a partir de los cuales se obtendrán los casos de uso del sistema para el servicio web a implementar.

2.1- Análisis de la situación problemática y el problema.

En el mundo actual con el auge que existe en el desarrollo de las TIC muchas empresas y organizaciones han dedicado grandes esfuerzos a la informatización de sus procesos.

Nuestra universidad no se ha quedado atrás en este aspecto y muchos de los procesos que al inicio se hacían de manera manual se han informatizado posibilitando así el ahorro de gran cantidad de materiales de oficina y esfuerzo humano.

En la facultad 1 de nuestra universidad como parte de este esfuerzo por informatizar todo lo que pueda ser informatizado se ha propuesto la implementación de una intranet a través de la cual se puedan gestionar sus procesos internos.

Nuestro trabajo en específico está dirigido a la informatización de los procesos de gestión y control en los departamentos de la facultad. Actualmente estos se llevan a cabo manualmente generando así un gran volumen de información muy difícil de controlar.

Es por ello que se planteó la necesidad de implementar un sistema capaz de gestionar toda la información referente a los profesores y alumnos ayudantes vinculados a un determinado departamento y que a su vez permita la gestión y control de todas las actividades que se realizan en el mismo.

Como un objetivo primario para la implementación de este sistema se surge la necesidad de implementar un servicio a través de cual se puedan gestionar todos estos procesos de manera eficiente.

2.2- Flujo actual y análisis de los procesos del negocio involucrados en el campo de acción.

El proceso de **registro de nuevos profesores y alumnos ayudantes** se inicia al llegar un nuevo profesor o alumno ayudante al departamento. El jefe del departamento registra en un documento Excel los datos del mismo.

Esto puede traer como consecuencia que los datos se registren incorrectamente además de esfuerzo humano que implica tener constancia de todos los profesores y alumnos ayudantes del departamento de manera organizada.

El proceso de **gestión de actividades** igualmente se realiza manualmente. Para la confección del plan de trabajo del departamento se reúnen los miembros del mismo, se proponen un conjunto de actividades para el mes cada una con su responsable y fecha de ejecución. El jefe del departamento registra las actividades propuestas en el plan de trabajo del departamento. En caso de que alguna de las actividades deba ser modificada o eliminada el proceso igualmente resulta ineficiente e implica un esfuerzo humano innecesario.

Dentro del proceso de gestión de las actividades también se tiene un control de asistencia de los miembros del departamento a las mismas. Este registro lo controla en responsable de la actividad el día en que se realiza y una vez terminada se lo entrega el jefe de departamento para que este lo compute en el registro de asistencia a actividades del departamento.

Todo este proceso es muy engorroso y requiere un gran esfuerzo humano.

En el proceso de **Elaboración del plan de trabajo DM:** El mismo es realizado por el jefe del departamento Este elabora el plan de trabajo docente metodológico para el departamento y se lo envía al vicedecano de docencia de la facultad, este lo revisa y si esta correcto lo aprueba. Una vez aprobado se le envía a los miembros del departamento para que tengan constancia del mismo.

Este proceso igualmente es engorroso pues todo se hace manualmente, además en caso haber alguna modificación en el plan de trabajo el proceso se tendría que repetir de nuevo.

El proceso de **gestión de los postgrados** actualmente se realiza en los departamentos de una forma ineficiente. La selección de los profesores que cursarán los postgrados se realiza de dos maneras: por solicitud y por designación. Cuando llega un nuevo postgrado el jefe de departamento en dependencia de las características del mismo puede informar a los profesores que recibirán el mismo (por designación) o enviar la información del nuevo postgrado a los miembros del departamento para que estos hagan sus solicitudes si están interesados (por solicitud). Una vez determinado los profesores que cursarán el postgrado, se registra esta información en el registro de postgrados del departamento.

En caso de que se necesite seleccionar un profesor para que imparta el postgrado, al recibir la información del mismo el jefe de departamento determina cuales profesores pueden impartirlo y de acuerdo a la carga docente de cada uno decide cuál de ellos impartirá el postgrado. Se le informa al profesor acerca de la designación, si este está de acuerdo se registra la información en un documento destinado para ello.

2.3- Objeto de informatización.

2.3.1- Descripción de los procesos a informatizar.

Con la implementación del servicio que se propone se informatizarán varios procesos dentro de los departamentos docentes que mejorarán su funcionamiento. Algunos de estos procesos son la gestión de toda la información referente a los profesores (internos, externos o colaboradores) y alumnos ayudante vinculados a un departamento en específico así como la gestión de las diferentes actividades programadas por el departamento.

También se informatizará la gestión de los grupos de facultad, de los postgrados que imparten y recibe cada profesor y de los planes de trabajo docente metodológicos de cada departamento.

El servicio tendrá funcionalidades para generar reportes de miembros por departamento y por categoría de persona.

2.3.2- Información que se maneja.

Para la realización del servicio web propuesto se necesita controlar toda la información referente a los profesores y alumnos ayudantes de un departamento. Algunos de los datos más importantes son CI, numero de solapín, correo, asignatura, grupos a los que imparte clases entre otros.

Se necesita además un control de las actividades que se planifican y ejecutan en el departamento. Este control incluye fecha de realización, responsable de la actividad y asistencia a la misma.

Algunas de las entidades que se manejan son:

- 1- Registro de miembros del departamento.
- 2- Plan de trabajo del departamento.
- 3- Registro de asistencia a las actividades.
- 4- Registro Postgrados.

En el nuevo sistema que se propone los datos de los profesores y alumnos ayudantes son obtenidos de la Base de Datos de la UCI (los que en ella se encuentran).

2.4- Propuesta de sistema.

El propósito de la presente investigación es crear una aplicación mediante la cual se pueda dar respuesta al problema científico plantado. La aplicación en si será un web services mediante el cual se pueda informatizar todo el proceso de gestión de los profesores y alumnos ayudantes en los departamentos docente de la facultad, entre otras funcionalidades.

El servicio web estará diseñado para soportar la interoperabilidad máquina - máquina a través de la red, sin dependencia del lenguaje o del sistema operativo a usar por el cliente. O sea debe estar disponible lo mismo para Linux que para Windows.

Entres las principales funcionalidades que debe brindar el servicio web tenemos la gestión de los datos de los profesores y alumnos ayudantes. Cada vez que llegue un nuevo miembro al departamento o se deje de pertenecer al mismo esa información podrá ser actualizada rápidamente en la BD.

También deberá ser capaz de gestionar toda la información relacionada con las actividades planificadas y realizadas por cada departamento, este control incluye el choqueo de la asistencia a la misma.

Proveeremos un conjunto de funcionalidades destinadas a la gestión de los postgrados en los cuales se halle involucrado algún profesor de la facultad, la gestión de los planes docente metodológicos de cada departamento y la gestión de los grupos de la facultad.

Contará además con un sistema de reportes que permitirá al jefe del departamento o a algún directivo de la facultad obtener informaciones importantes de manera rápida y organizada.

Para llevar a cabo el desarrollo de esta propuesta, se partió del análisis realizado de los procesos que se desarrollan en la institución referentes al control de todos los procesos descritos y las necesidades que existían. Se tuvo en cuenta el estudio de algunos software y sistemas similares que existen en Cuba y el Mundo, los cuales fueron reflejados en el capítulo 1, con el objetivo de tener conocimientos de los mismos en cuanto a funcionalidades y características, destacando que dentro de las investigaciones efectuadas no se detectaron servicios web similares al que se propone implementar.

De este análisis se concluyó que los sistemas estudiados en el capítulo 1 no cumplen con los requisitos que demanda el sistema. Nuestra propuesta de utilizar un servicio web está basada fundamentalmente en la necesidad de tener este módulo en un entorno desacoplado, que permita una independencia de lenguaje y sistema operativo, de acuerdo con los requerimientos del sistema.

2.5- Reglas del negocio.

Los procesos presentes en el negocio siempre presentan determinadas reglas cuyo cumplimiento es obligatorio a la hora de llevar a cabo su informatización. En el sistema a desarrollar en la presente investigación tenemos las siguientes:

- Solo el jefe del departamento puede modificar o añadir información de los miembros de su departamento.
- Solo el jefe de departamento podrá actualizar el registro de asistencia.

- Solo se puede modificar una actividad con la autorización del jefe del departamento.
- Solo el jefe de departamento puede elaborar o modificar el plan de trabajo DM.
- Solo el jefe de departamento puede actualizar el registro de postgrados.

2.6- Modelo del negocio.

Su objetivo es describir los procesos, existentes u observados, con el propósito de comprenderlos. Aquí se especifican qué procesos del negocio soportará el sistema, se identifican los objetos del modelo de dominio y del negocio. Este modelo establece las competencias que se requieren de cada proceso: sus trabajadores, sus responsabilidades y las operaciones que llevan a cabo.

2.7- Actores del negocio.

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos con los que el negocio interactúa.

Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados.

Tabla2. 1 Actores del Negocio.

Actor	Descripción
Miembro del Departamento	Este actor incluye al Profesor y al Alumno Ayudante. Ambos pertenecen al mismo departamento e imparten alguna materia.
Vicedecano de Docencia	Persona que a nivel de facultad interactúa con el negocio.

2.8- Trabajadores del negocio.

Un Trabajador del negocio representa a personas, o sistemas (software) dentro del negocio que son las que realizan las actividades que están comprendidas dentro de un caso de uso. Estos trabajadores están dentro de la frontera del negocio, son los que en un futuro se convertirán en usuarios del sistema que se quiere construir.

Tabla2. 2 Trabajadores del negocio

Trabajador	Descripción
Jefe de Departamento	Es el encargado de gestionar la información de todos los miembros de su departamento, ya sea actualizar información, adicionar información o eliminar información.
Responsable de Actividad	Es el encargado de controlar la información referente a la actividad de la cual es responsable. Debe controlar su cumplimiento y tener control de la asistencia a la misma.
Secretaria de docencia	Es el encargado de gestionar la información referente a los grupos de la facultad, los departamentos, las disciplinas, las asignaturas y los postgrados.

2.9- Diagrama de CU del negocio.

Un Caso de Uso del Negocio representa un proceso dentro del negocio que se estudia, por lo que se corresponde con una secuencia de acciones con un orden lógico y que producen un resultado observable para ciertos actores del negocio.

El diagrama de casos de uso del negocio representa gráficamente los procesos del negocio y su interacción con los actores del negocio.

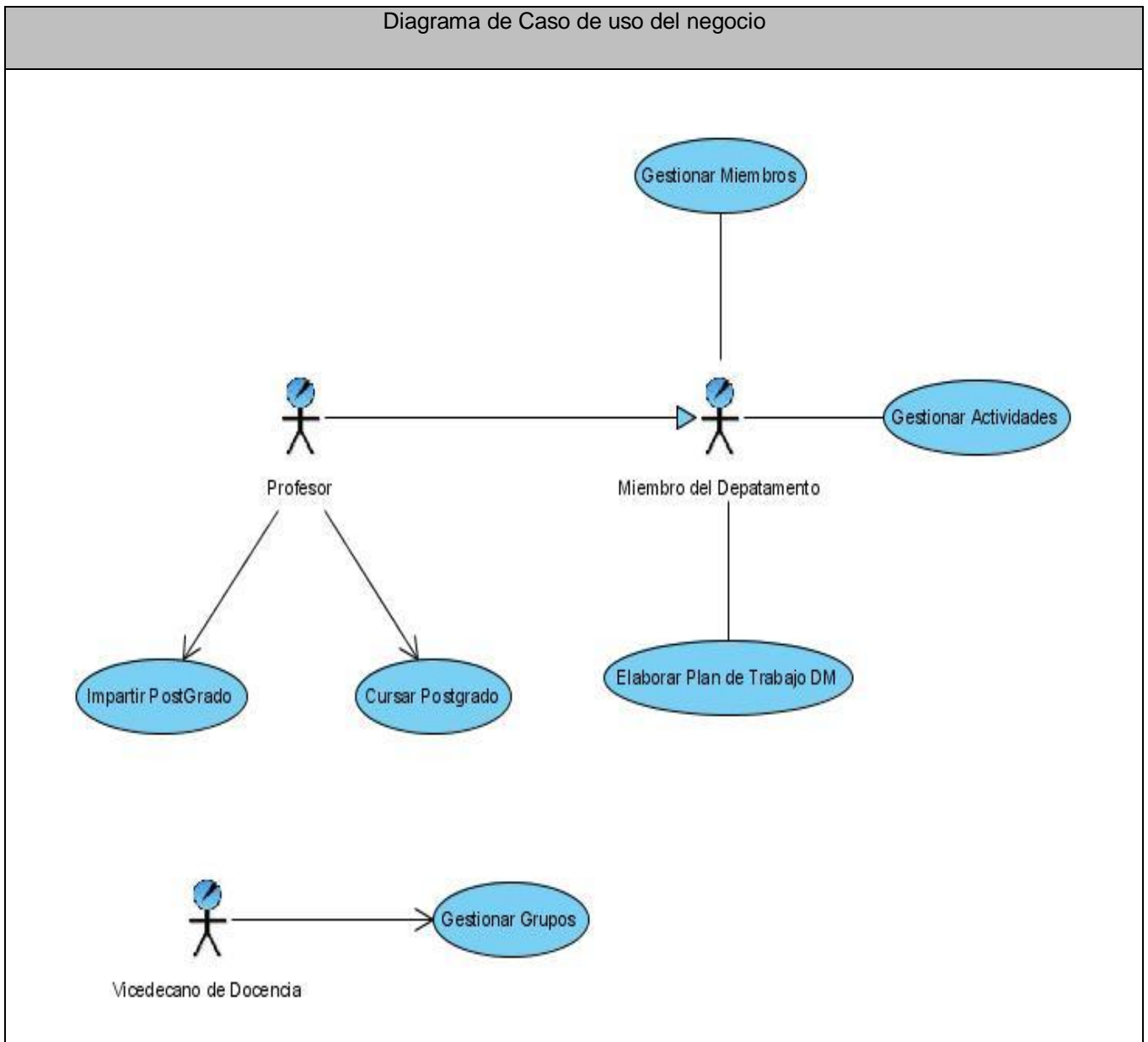


Figura 1 Diagrama de Caso de uso del negocio.

2.10- Descripción de casos de uso del negocio.

Para la comprensión de los procesos del negocio se ha realizado una descripción de los mismos teniendo en cuenta que permiten obtener una visión sobre el problema de manera general y qué es lo que realmente se necesita. Estas descripciones podrán ser encontradas en el **Anexo I**.

2.11- Diagrama de actividades.

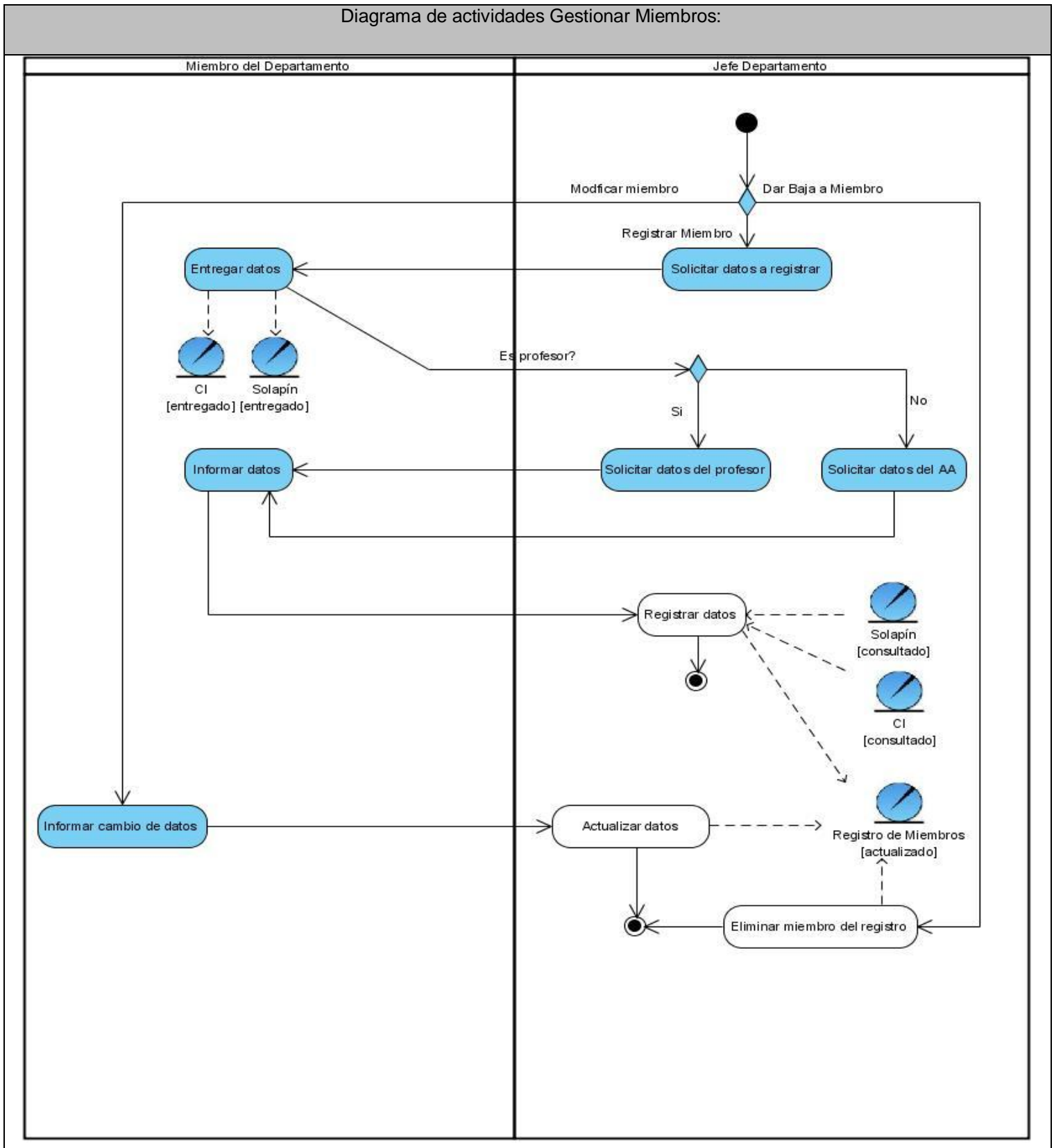


Figura 3 Diagrama de actividades. Gestionar Miembros

Diagrama de actividades Cursar Postgrado:

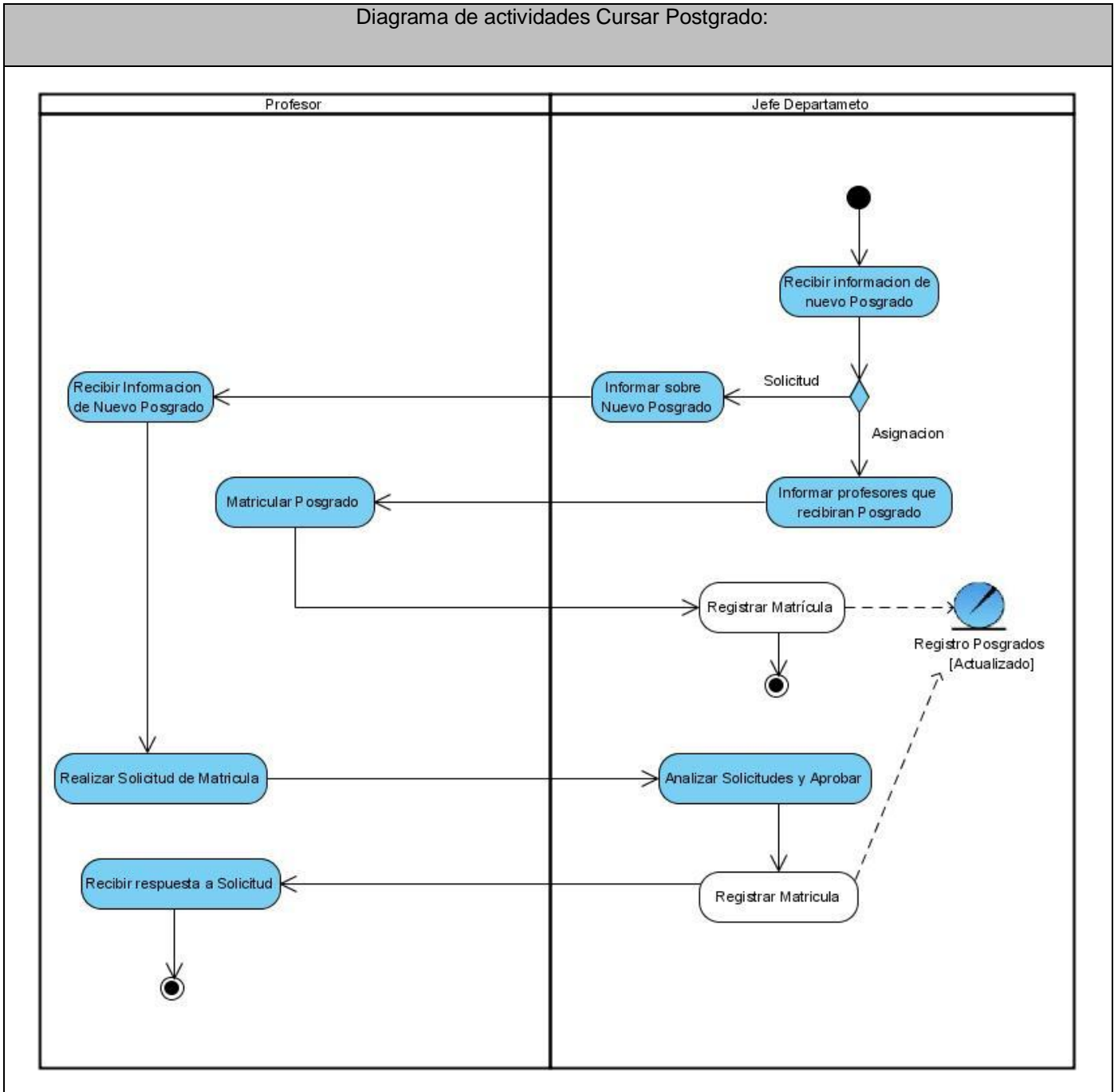


Figura 4 Diagrama de actividades Cursar Postgrado

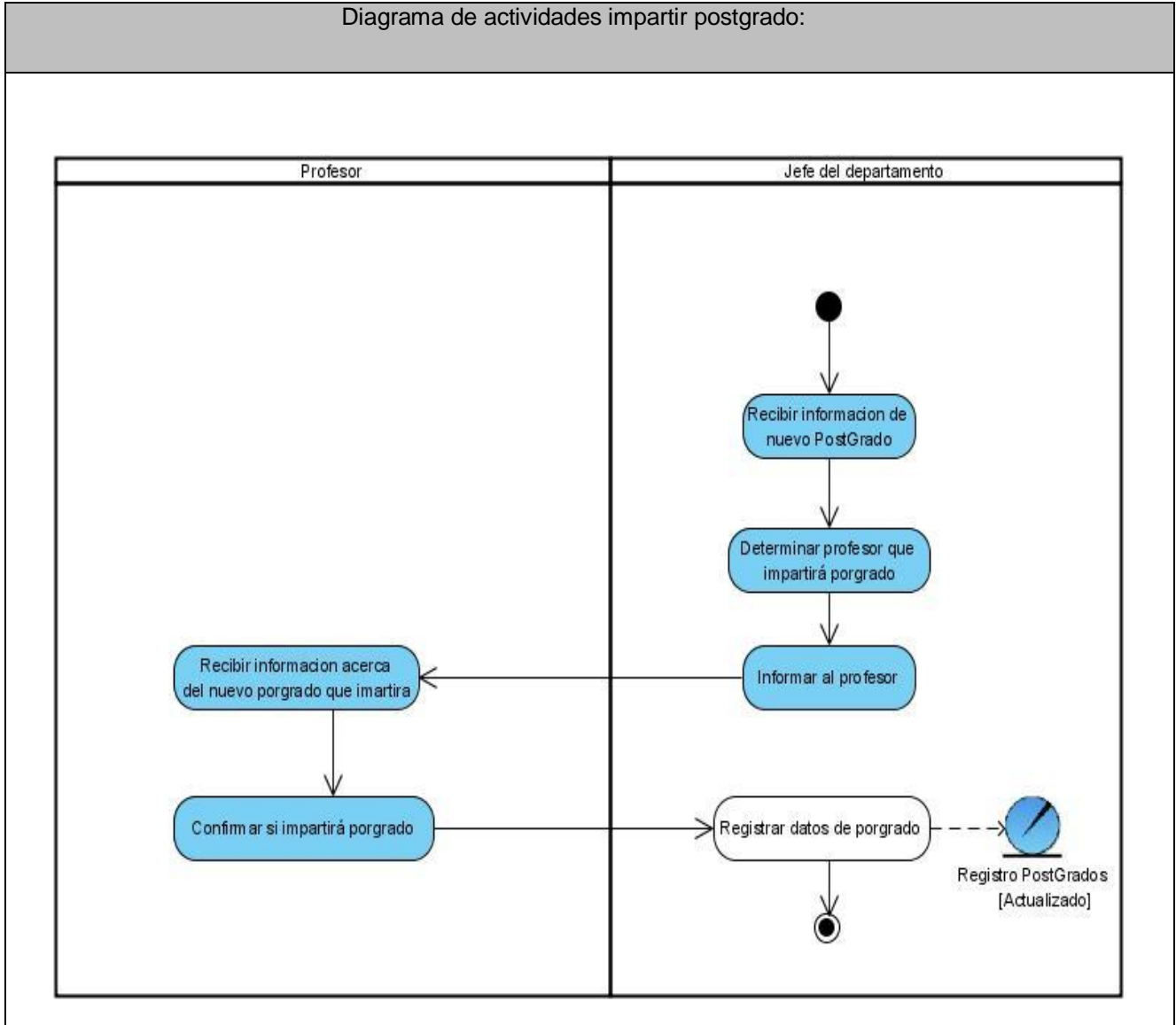


Figura 5 Diagrama de actividades impartir postgrado

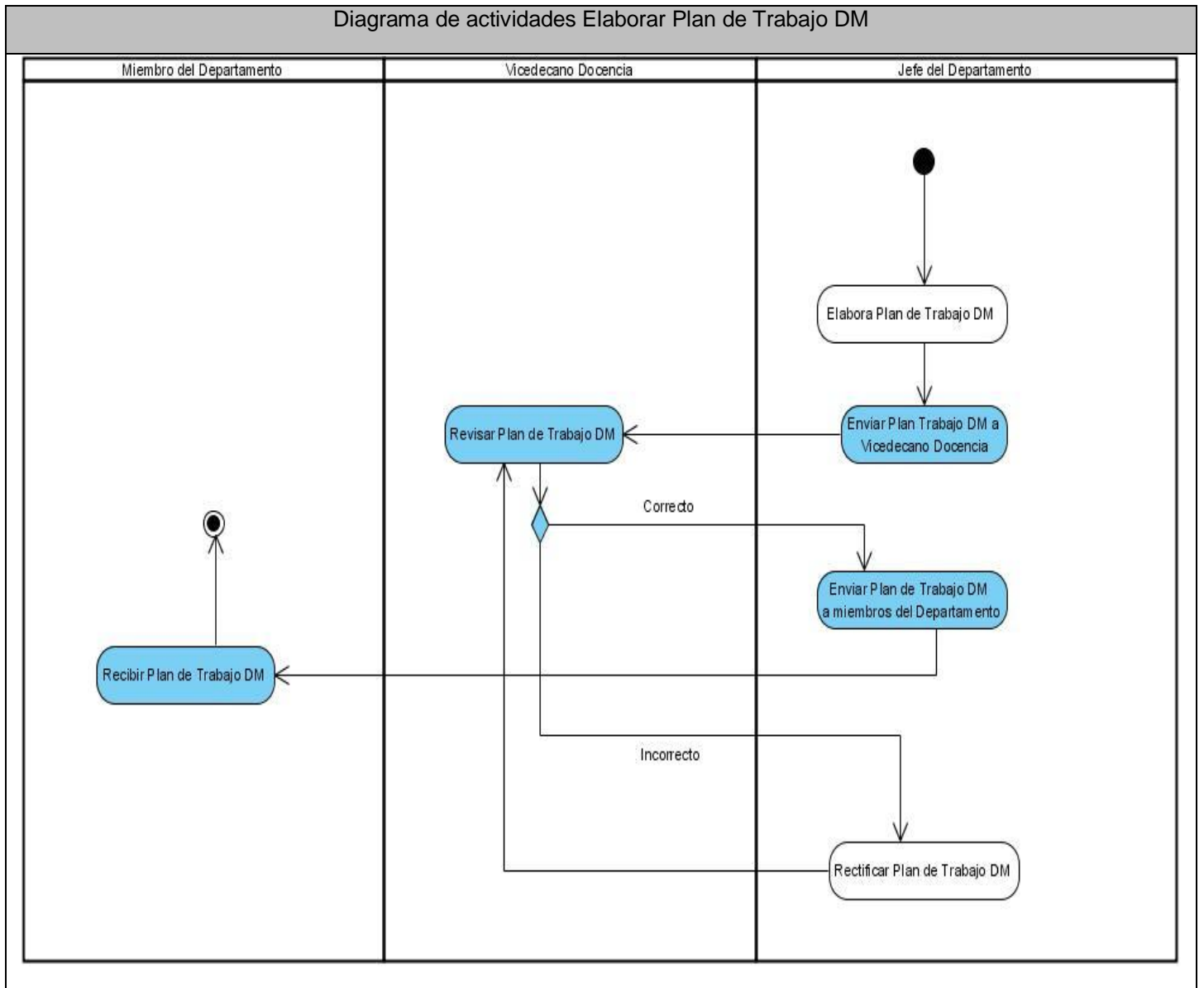


Figura 6 Diagrama de actividades Elaborar Plan de Trabajo DM

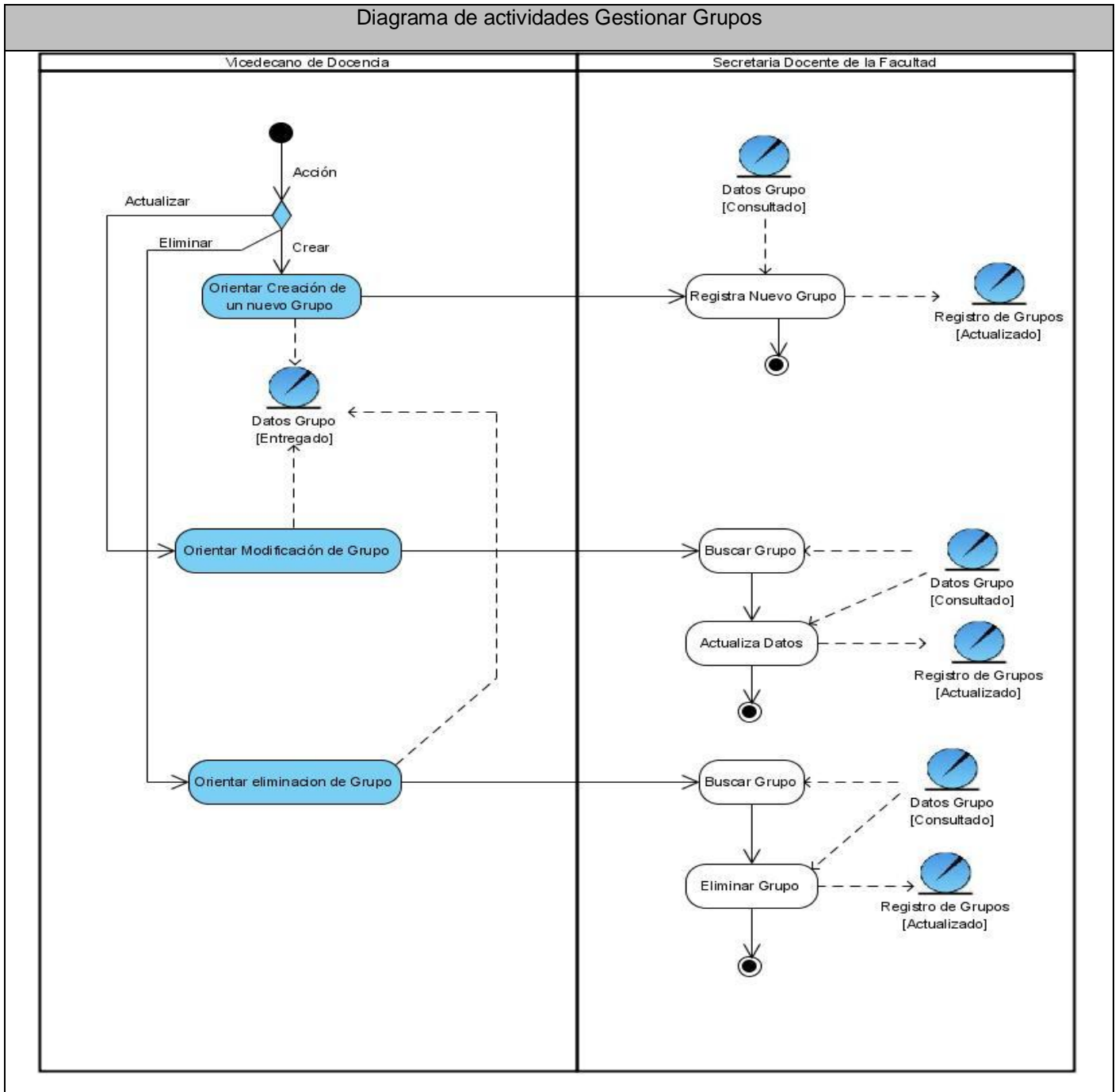


Figura 7 Diagrama de actividades Gestionar grupos

2.12- Diagrama de Clases del modelo de Objetos.

Los diagramas de clases del modelo de objetos describen el modelo de objetos del negocio, muestran la participación de los trabajadores y entidades del negocio y la relación que se establece entre ellos.

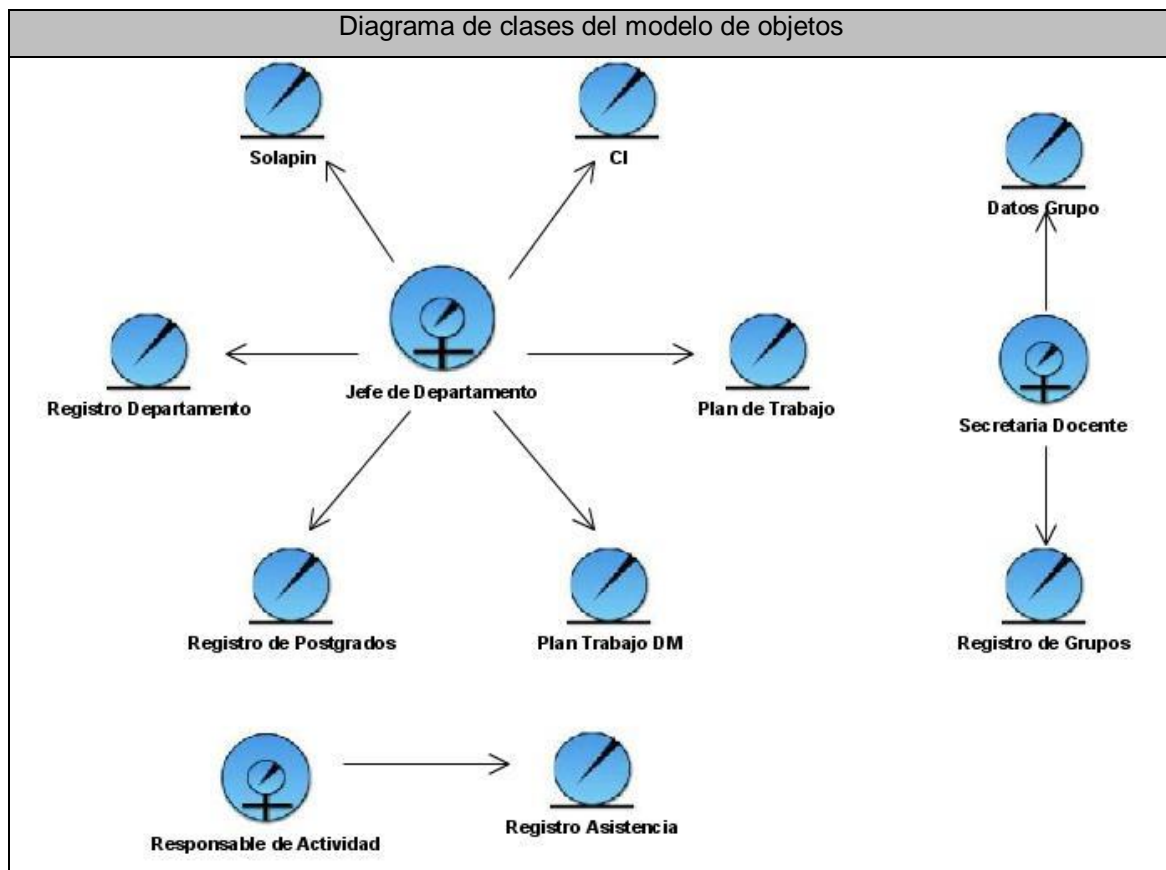


Figura 8 Diagrama de clases del modelo de objetos.

2.13- Especificación de los requisitos.

2.13.1- Requisitos funcionales.

Los requisitos funcionales especifican funciones que el sistema debe ser capaz de realizar. Nuestro sistema debe brindar las siguientes funcionalidades:

RF_1. Gestionar Profesor Interno.

RF_1.1 Insertar profesor interno dado como datos: Departamento, militancia de la UJC, nombre y apellidos, CI, solapín, correo, disciplina, asignatura, categoría docente, categoría científica, militancia del PCC, apartamento, teléfono del apartamento,

laboratorio de producción, teléfono del laboratorio, provincia, municipio, calle, entre calle, reparto, número de casa, edificio y apartamento.

RF_1.2 Actualizar datos profesor interno dado como datos: Departamento, militancia de la UJC, nombre y apellidos, CI, solapín, correo, disciplina, asignatura, categoría docente, categoría científica, militancia del PCC, apartamento, teléfono del apartamento, laboratorio de producción, teléfono del laboratorio, provincia, municipio, calle, entre calle, reparto, número de casa, edificio y apartamento.

RF_1.3 Eliminar profesor interno dado como dato CI.

RF_2. Gestionar profesor externo.

RF_2.1 Insertar profesor externo dado como datos: Departamento, militancia de la UJC, nombre y apellidos, CI, solapín, correo, disciplina, asignatura, categoría docente, categoría científica, militancia del PCC, institución, teléfono de la institución, provincia, municipio, calle, entre calle, reparto, número de casa, edificio y apartamento.

RF_2.2 Actualizar datos de un profesor externo dado como datos: Departamento, militancia de la UJC, nombre y apellidos, CI, solapín, correo, disciplina, asignatura, categoría docente, categoría científica, militancia del PCC, institución, teléfono de la institución, provincia, municipio, calle, entre calle, reparto, número de casa, edificio y apartamento.

RF_2.3 Eliminar profesor externo dado como dato CI.

RF_3. Gestionar profesor colaborador.

RF_3.1 Insertar un profesor colaborador dado como datos: Departamento, militancia de la UJC, nombre y apellidos, CI, solapín, correo, disciplina, asignatura, categoría docente, categoría científica, militancia del PCC, área de trabajo, provincia, municipio, calle, entre calle, reparto, número de casa, edificio y apartamento.

RF_3.2 Actualizar datos de un profesor colaborador dado como datos: Departamento, militancia de la UJC, nombre y apellidos, CI, solapín, correo, disciplina, asignatura, categoría docente, categoría científica, militancia del PCC, área de trabajo, provincia, municipio, calle, entre calle, reparto, número de casa, edificio y apartamento.

RF_3.3 Eliminar profesor colaborador dado como dato CI.

RF_4. Gestionar alumnos ayudantes.

RF_4.1 Insertar un alumno ayudante dado como datos: Departamento, militancia de la UJC, nombre y apellidos, CI, solapín, correo, disciplina, asignatura, apartamento, teléfono del apartamento, profesor tutor, provincia, municipio, calle, entre calle, reparto, número de casa, edificio y apartamento.

RF_4.2 Actualizar los datos de un alumno ayudante dado como datos: Departamento, militancia de la UJC, nombre y apellidos, CI, solapín, correo, disciplina, asignatura, apartamento, teléfono del apartamento, profesor tutor, provincia, municipio, calle, entre calle, reparto, número de casa, edificio y apartamento.

RF_4.3 Eliminar alumno ayudante dado como dato: CI.

RF_5. Buscar persona dado como dato: Departamento y nombre.

RF_6. Gestionar Planes de Trabajo DM.

RF_6.1 Adicionar actividades DM al plan de trabajo DM dado como datos: id departamento, semestre, responsable, descripción de la actividad, aseguramientos, participantes, meses en que se debe desarrollar la misma.

RF_6.2 Buscar actividades DM del plan de trabajo DM dado como datos: Departamento docente y semestre.

RF_6.3 Modificar actividad DM dado como datos: id actividad, id departamento, semestre, responsable, descripción de la actividad, aseguramientos, participantes, meses en que se debe desarrollar la misma.

RF_7. Gestionar Actividades.

RF_7.1 Adicionar actividades al plan de trabajo del departamento dado como datos: id departamento, responsable, mes, fecha de cumplimiento, descripción de la actividad, lugar.

RF_7.2 Buscar actividades dado como datos: departamento y mes en que fue planificada.

RF_7.3 Modificar actividades dado como datos: id actividad, id departamento, responsable, mes, fecha de cumplimiento, descripción de la actividad, lugar.

RF_7.4 Registrar asistencia a actividades dado como datos: id actividad, id persona.

RF_7.5 Actualizar registro de asistencia a actividades dado como datos: id actividad, id persona.

RF_7.6 Obtener reporte de asistencia a actividades dado como dato: id actividad.

RF_8. Gestionar Departamentos.

RF_8.1 Insertar departamento dado como dato: nombre departamento.

RF_8.2 Buscar departamento dado como dato: nombre departamento.

RF_8.3 Actualizar departamento dado como datos: id departamento, nombre departamento.

RF_9. Gestionar Disciplinas.

RF_9.1 Insertar disciplina dado como datos: nombre departamento, nombre disciplina.

RF_9.2 Buscar disciplina dado como dato: id de departamento.

RF_9.3 Actualizar disciplina dado como datos: id departamento, id disciplina, nombre disciplina.

RF_10. Gestionar Asignaturas.

- RF_10.1 Insertar asignatura dado como datos: id de departamento, nombre disciplina, nombre asignatura, año de la asignatura.
- RF_10.2 Buscar asignatura dado como datos: id de departamento, id de disciplina.
- RF_10.3 Actualizar asignatura dado como datos: id departamento, id disciplina, id asignatura, nombre asignatura, año.
- RF_11. Gestionar Grupos.
 - RF_11.1 Insertar grupo dado como datos: Nombre grupo, año.
 - RF_11.2 Buscar grupo dado como datos: año o nombre.
 - RF_11.3 Actualizar grupo dado como datos: id grupo, nombre de grupo, año.
 - RF_11.4 Eliminar grupo dado como dato: id grupo.
- RF_12. Gestionar asignación de grupos.
 - RF_12.1 Asignar grupos a un profesor dado como datos: id grupo, id profesor.
 - RF_12.2 Eliminar asignación dado como datos: id grupo, id profesor.
- RF_13. Gestionar Postgrados.
 - RF_13.1 Insertar Postgrado dado como datos: Nombre postgrado, Horario (Descripción), curso escolar, semestre, fecha de inicio, fecha de fin.
 - RF_13.2 Buscar postgrado dado como datos: Nombre o estado.
 - RF_13.3 Actualizar postgrado dado como datos: Nombre postgrado, Horario (Descripción), curso escolar, semestre, fecha de inicio, fecha de fin, estado.
 - RF_13.4 Registrar matricula de postgrado dado como datos: id persona, id postgrado.

RF_13.5 Registrar profesor que imparte postgrado dado como datos: id persona, id postgrado.

RF_14. Generar Reportes.

RF_14.1 Obtener miembros de un departamento dado: id departamento.

RF_14.2 Obtener personas por categoría dado como datos: id departamento y categoría.

2.13.2- Requisitos 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 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.

También se puede señalar que la implementación de un servicio web posibilita una independencia de lenguaje de programación y sistema operativo para el cliente pues la comunicación con ellos se basa en mensajes XML.

• Software

Se necesita como sistema Operativo Windows 98 o superior, Linux o Unix, en sus versiones de S.O servidores para el servidor del Web Services.

Se necesita tener instalado el servidor de aplicaciones Web Apache en el servidor del web services.

Se necesita tener instalado MySQL en el servidor de Base de Datos.

• Hardware

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

Se requiere como mínimo 256 MB de memoria RAM.

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 transportar texto (http, 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.

• **Importación y exportación de datos**

El sistema deberá almacenar todos los datos en una base de datos MySQL, 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 modelado Visual Paradigm. Se usa como lenguaje de programación PHP y como sistema gestor de base de datos MySQL.

Se usa la librería NuSoap para la creación del servicio web.

Se usa la librería ADOdb para la conexión a la Base de 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.

2.14- Definición de los CU del sistema:

Tabla2. 3 Actores del sistema:

Actor	Descripción:
Intranet facultad 1	La Intranet de la facultad 1 es el actor del sistema pues es la que invoca a los servicios implementados en el web services.

2.14.1- Casos de uso del sistema:

Los casos de uso son fragmentos de funcionalidad que un sistema ofrece. En ellos se describe la secuencia determinada de eventos que realiza un actor en interacción con el sistema, en este caso el servicio web, los mismos fueron determinados de acuerdo a funciones que pudiera brindar.

Tabla2. 4 Descripción del CU del sistema Gestionar profesor interno

CU_1	Gestionar profesor interno
Actor:	Intranet facultad 1
Descripción	Se inicia cuando el actor desea realizar alguna de las siguientes operaciones: Insertar un nuevo profesor interno, actualizar los datos de un profesor interno o eliminar un profesor interno. En caso de querer insertar un nuevo profesor interno invoca al servicio insertar_profesor_interno_general, en caso de querer actualizar los datos de un profesor interno invoca al servicio actualizar_profesor_interno_general y en caso de querer eliminar un profesor interno invoca el servicio eliminar_profesor_interno_general finalizando así el CU.
Referencias	RF_1 RF_1.1 RF_1.2 RF_1.3

Tabla2. 5 Descripción del CU del sistema Gestionar Profesor externo

CU_2	Gestionar Profesor externo
Actor	Intranet facultad 1
Descripción	Se inicia cuando el actor desea realizar alguna de las siguientes operaciones: Insertar un nuevo profesor externo, actualizar los

	<p>datos de un profesor externo o eliminar un profesor externo.</p> <p>En caso de querer insertar un nuevo profesor externo invoca al servicio insertar_profesor_externo_general, en caso de querer actualizar los datos de un profesor externo invoca al servicio actualizar_profesor_externo_general y en caso de querer eliminar un profesor externo invoca el servicio eliminar_profesor_externo_general finalizando así el CU.</p>
Referencias:	RF_2 RF_2.1 RF_2.2 RF_2.3

Tabla2. 6 Descripción del CU del sistema Gestionar Profesor colaborador

CU_3	Gestionar Profesor colaborador
Actor	Intranet facultad 1
Descripción	<p>Se inicia cuando el actor desea realizar alguna de las siguientes operaciones: Insertar un nuevo profesor colaborador, actualizar los datos de un profesor colaborador o eliminar un profesor colaborador.</p> <p>En caso de querer insertar un nuevo profesor colaborador invoca al servicio insertar_profesor_colaborador_general, en caso de querer actualizar los datos de un profesor colaborador invoca al servicio actualizar_profesor_colaborador_general y en caso de querer eliminar un profesor colaborador invoca el servicio eliminar_profesor_colaborador_general finalizando así el CU.</p>
Referencias:	RF_3 RF_3.1 RF_3.2 RF_3.3

Tabla2. 7 Descripción del CU del sistema Gestionar alumno ayudante

CU_4	Gestionar alumno ayudante
Actor	Intranet facultad 1
Descripción	<p>Se inicia cuando el actor desea realizar alguna de las siguientes operaciones: Insertar un nuevo alumno ayudante, actualizar los datos de un alumno ayudante o eliminar un alumno ayudante.</p> <p>En caso de querer insertar un nuevo alumno ayudante invoca al servicio insertar_alumno_ayudante_general, en caso de querer actualizar los datos de un alumno ayudante invoca al servicio actualizar_alumno_ayudante_general y en caso de querer eliminar un alumno ayudante invoca el servicio eliminar_alumno_ayudante_general finalizando así el CU.</p>
Referencias:	RF_4 RF_4.1 RF_4.2 RF_4.3

Tabla2. 8 Descripción del CU del sistema Buscar Persona

CU_5	Buscar Persona
Actor	Intranet facultad 1
Descripción	Se inicia cuando el actor desea buscar una persona en el sistema e invoca el servicio buscar_persona finalizando así el CU.
Referencias:	RF_5

Tabla2. 9 Descripción del CU del sistema Gestionar Planes de Trabajo DM

CU_6	Gestionar Planes de Trabajo DM
Actor	Intranet facultad 1
Descripción	Se inicia cuando el actor del sistema desea realizar alguna de las siguientes acciones: Adicionar una nueva actividad al plan de trabajo docente metodológico, modificar alguna de las actividades o buscar las actividades dado departamento y semestre. En caso de querer adicionar una nueva actividad al plan de trabajo DM se invoca al servicio insertar_Actividad_docente_metodologica. En caso de querer modificar una actividad se invoca el servicio actualizar_Actividad_docente_metodologica y en caso de desear buscar las actividades DM se invoca el servicio buscar_actividadesDM que dado el id de departamento y semestre devuelve las actividades DM finalizando así el CU.
Referencias	RF_6 RF_6.1 RF_6.2 RF_6.3

Tabla2. 10 Descripción del CU Sistema Gestionar Actividades

CU_7	Gestionar Actividades
Actor	Intranet facultad 1
Descripción	Se inicia cuando el actor desea realizar una de las siguientes acciones: Adicionar una nueva actividad al plan de trabajo del departamento, actualizar una actividad, registrar la asistencia, modificar el registro de asistencia, obtener el reporte de asistencia y buscar las actividades planificadas dado id de departamento y mes. En caso de desear adicionar una nueva actividad de invoca al servicio llamado insertar_actividad, en caso de desear modificar un actividad se invoca al servicio llamado actualizar_actividad, en caso de querer registrar la asistencia se invoca el servicio llamado registrar_asistencia, en caso de querer modificar el registro de asistencia se invoca el servicio llamado

	<p>actualizar_registro_asistencia y en caso de desear obtener el reporte de asistencia a una actividad invoca el servicio registrosXidactividad.</p> <p>En caso de querer buscar las actividades planificadas se invoca el servicio buscar_actividad.</p>
Referencias	RF_7 RF_7.1 RF_7.2 RF_7.3 RF_7.4 RF_7.5 RF_7.6

Tabla2. 11 Descripción del CU sistema Gestionar Departamentos

CU_8	Gestionar Departamentos
Actor	Intranet facultad 1
Descripción	<p>Se inicia cuando el actor desea realizar una de las siguientes acciones: Crear un nuevo departamento, modificar nombre de departamento docente.</p> <p>En caso de querer crear un nuevo departamento de invoca al servicio insertar_departamento. En caso de querer modificar el departamento se invoca el servicio actualizar_departamento y en caso de querer buscar un departamento se llama al servicio buscar_departamento finalizando así el CU.</p>
Referencias	RF_8 RF_8.1 RF_8.2 RF_8.3

Tabla2. 12 Descripción del CU sistema Gestionar Disciplinas

CU_9	Gestionar Disciplinas
Actor	Intranet facultad 1
Descripción	<p>Se inicia cuando el actor desea realizar una de las siguientes acciones: Adicionar una nueva disciplina o actualizar los datos de la disciplina.</p> <p>En caso de querer adicionar una nueva disciplina se invoca al servicio insertar_disciplina. En caso de querer modificar la disciplina se invoca el servicio actualizar_disciplina y en caso de desear buscar una o varias disciplinas se invoca el servicio buscar_disciplina_x_iddepartamento finalizando así el CU.</p>
Referencias	RF_9 RF_9.1 RF_9.2 RF_9.3

Tabla2. 13 Descripción del CU sistema Gestionar Asignaturas

CU_10	Gestionar Asignaturas
Actor	Intranet facultad 1
Descripción	<p>Se inicia cuando el actor desea realizar una de las siguientes acciones: Adicionar una nueva asignatura o actualizar los datos de una asignatura.</p> <p>En caso de querer adicionar una nueva asignatura se invoca al servicio insertar_asignatura. En caso de querer modificar la disciplina se invoca el servicio actualizar_asignatura y en caso de desear buscar una asignatura se invoca el servicio buscar_asignaturaXiddisciplina finalizando así el CU.</p>
	RF_10 RF_10.1 RF_10.2 RF_10.3

Tabla2. 14 Descripción del CU sistema Gestionar Grupos

CU_11	Gestionar Grupos
Actor	Intranet facultad 1
Descripción	<p>Se inicia cuando el actor desea realizar alguna de las siguientes acciones: Adicionar un nuevo grupo, actualizar grupo, eliminar grupo.</p> <p>En caso de querer adicionar un nuevo grupo se invoca al servicio insertar_grupo, en caso de querer modificar un grupo invoca el servicio actualizar_grupo, en caso de querer eliminar un grupo se invoca el servicio eliminar_grupo_general y en caso de querer buscar unos o varios grupos se invoca los servicios buscar_grupo o buscar_grupoXanno finalizando así el CU.</p>
Referencias	RF_11 RF_11.1 RF_11.2 RF_11.3 RF_11.4

Tabla2. 15 Descripción del CU sistema Gestionar Asignación de Grupos

CU_12	Gestionar Asignación de Grupos
Actor	Intranet facultad 1
Descripción	<p>Se inicia cuando el actor desea realizar alguna de las siguientes acciones: Asignarle a un profesor los grupos que atenderá en la facultad, Terminar asignación.</p> <p>En caso de querer Asignar los grupos a un profesor se invoca el servicio asignar_grupos. En caso de querer terminar la asignación se invoca el servicio terminar_asignacion finalizando así el CU.</p>
Referencias	RF_12 RF_12.1 RF_12.2

Tabla2. 16 Descripción del CU sistema Gestionar Postgrados

CU_13	Gestionar Postgrados
Actor	Intranet facultad 1
Descripción	<p>Se inicia cuando el actor desea realizar alguna de las siguientes acciones: Registrar datos de nuevo postgrado, actualizar datos de postgrado, registrar matricula del postgrado, registrar profesor que imparte postgrado o buscar el postgrado.</p> <p>En caso de querer Registrar los datos de un nuevo postgrado se invoca el servicio insertar_postgrado. En caso de querer actualizar los datos del postgrado se invoca el servicio actualizar_postgrado, en caso de querer buscar postgrados se invocan los servicios buscarpostgradoXestado o buscarpostgradoXnombre. Para registrar la matrícula del postgrado o registrar el profesor que imparte el postgrado se invoca a los servicios matricular_postgrado o registrar_profesor_postgrado finalizando así el CU.</p>
Referencias	RF_13 RF_13.1 RF_13.2 RF_13.3 RF_13.4 RF_13.5

Tabla2. 17 Descripción del CU sistema Generar Reportes

CU_14	Generar Reportes
Actor	Intranet facultad 1
Descripción	<p>Se inicia cuando el actor desea realizar alguna de las siguientes acciones: Obtener miembros de un departamento u Obtener personas por categoría.</p> <p>En caso de querer obtener los miembros de un departamento se invoca el servicio buscar_personas_x_departamento y en caso de querer obtener un listado de personas por categoría se invoca el servicio buscar_personas_x_categoria finalizando así el CU.</p>
Referencias	RF_14 RF_14.1 RF_14.2

Diagrama de casos de uso del sistema.

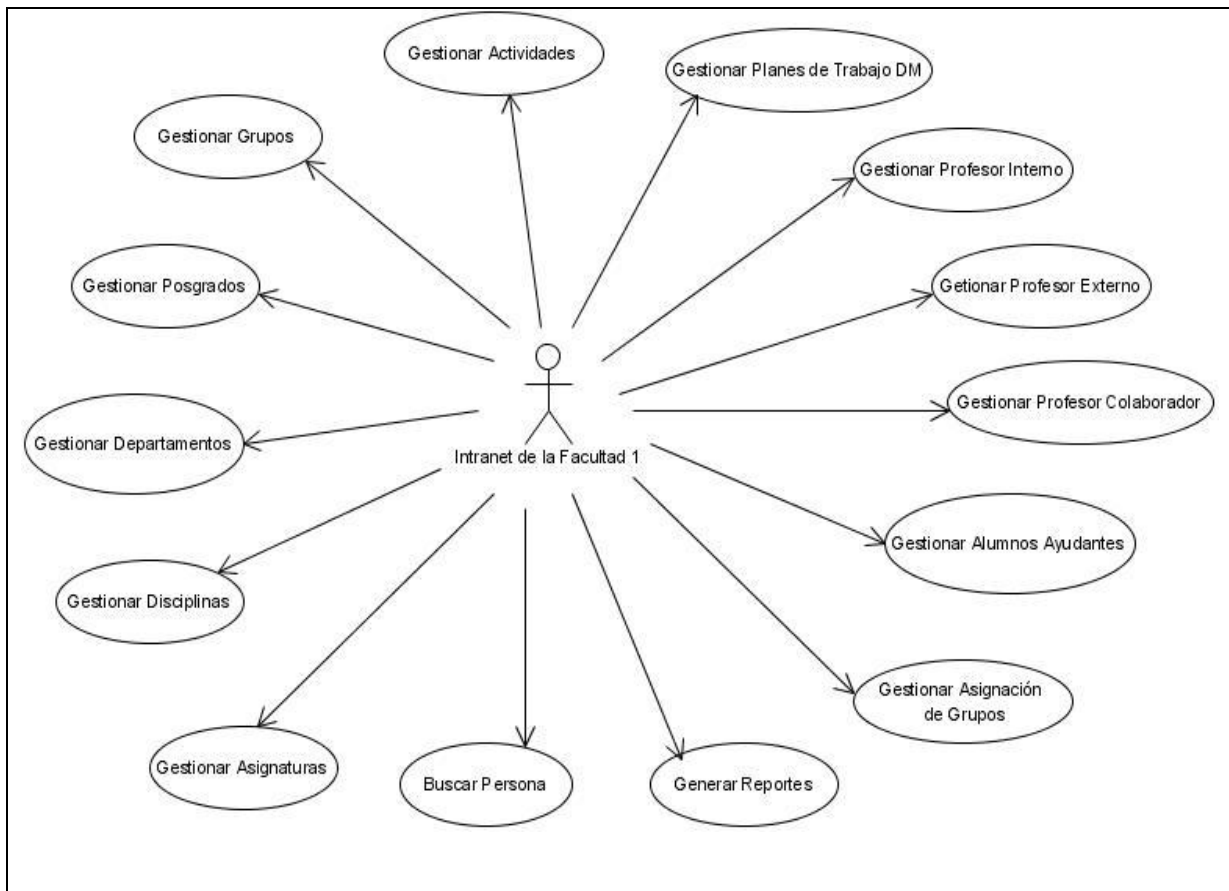


Figura 9 Diagrama de casos de uso del sistema

2.14.2- Casos de uso expandidos.

Las descripciones expandidas de los CU del sistema se encuentran en el **anexo II**.

2.15- Estudio de la Factibilidad.

Método a utilizar: Método de Estimación por Puntos de Casos de Uso.

Paso 1: Cálculo de Puntos de Casos de Uso sin ajustar.

Se calcula a través de la siguiente ecuación:

$$UUCP = UAW + UUCW$$

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.

Tipo de actor	Descripción	Factor de peso	Actores	Total
Simple	Sistema con sistema a través de interfaz de programación.	1	0	0
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			1	2

$$UAW = \sum \text{cant_actores} * \text{Factor de Peso}$$

$$UAW = 2$$

1.2 Factor de Peso de los Casos de Uso sin ajustar (UUCW)

Este valor se calcula mediante un análisis de la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Casos de Uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo, donde una transacción se entiende como una secuencia de actividades atómica, es decir, se efectúa la secuencia de actividades completa, o no se efectúa ninguna de las actividades de la secuencia. Los criterios se muestran en la siguiente tabla:

1.1 Factor de Peso de los Actores sin ajustar (UAW)

Este valor se calcula mediante un análisis de la cantidad de Actores presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Actores se establece teniendo en cuenta en primer lugar si se trata de una persona o de otro sistema, y en segundo lugar, la forma en la que el actor interactúa con el sistema.

Los criterios se muestran en la siguiente tabla:

Tipo de CU	Descripción	Peso	Cantidad de CU	Total
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	7	35
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	7	70
Complejo	El caso de uso tiene más de 8 transacciones.	15	0	0
Total			14	105

$$UUCW = \sum \text{cant_CU} * \text{Peso}$$

$$UUCW = (7*5) + (10*7)$$

$$UUCW = 35+70$$

$$UUCW = 105$$

Los Puntos de Casos de Uso sin ajustar resultan

$$UUCP = UAW + UUCW$$

$$UCCP = 2 + 105$$

$$UCCP = 107$$

Paso 2.

Cálculo de Puntos de Casos de Uso ajustados.

Se calcula mediante la siguiente ecuación:

$$UCP = UUCP \times TCF \times EF$$

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

2.1 Factor de complejidad técnica (TCF)

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.

Factor	Descripción	Peso	Valor asignado	Total
T1	Sistema distribuido	2	0	0
T2	Tiempo de respuesta	1	3	3
T3	Eficiencia del usuario final	1	1	1
T4	Funcionamiento Interno complejo	1	4	4
T5	El código debe ser reutilizable	1	3	3
T6	Facilidad de instalación	0,5	0	0
T7	Facilidad de uso	0,5	5	2,5
T8	Portabilidad	2	5	10
T9	Facilidad de cambio	1	5	5
T10	Concurrencia	1	3	3
T11	Incluye objetivos especiales de seguridad	1	3	3
T12	Provee acceso directo a terceras partes	1	0	0
T13	Se requieren facilidades especiales de entrenamiento de usuarios	1	1	1
Total				35.5

$$TCF = 0.6 + 0.01 \times \Sigma (\text{Peso}_i \times \text{Valor asignado}_i)$$

$$TCF = 0.6 + 0.01 * 35.5$$

$$TCF = 0.955$$

2.2 Factor de ambiente (EF)

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, es decir, se trata de un conjunto de factores que se cuantifican con valores de 0 a 5.

Factor	Descripción	Peso	Valor asignado	Total
E1	Familiaridad con el modelo de proyecto utilizado	1,5	3	4,5
E2	Experiencia en la aplicación	0,5	0	0
E3	Experiencia en la orientación a objeto.	1	3	3
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	3	-3
Total				21

$$EF = 1.4 - 0.03 \times \Sigma (\text{Peso}_i \times \text{Valor asignado}_i)$$

$$EF = 1.4 - 0.03 * 21$$

$$EF = 0.77$$

$$UCP = UUCP \times TCF \times EF$$

$$UCP = 107 * 0.955 * 0.77$$

$$UCP = 78.68245$$

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

$$E = UCP \times CF$$

Donde:

E: Esfuerzo estimado en horas hombres.

UCP: Punto de casos de usos ajustados.

CF: Factor de conversión.

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

En este caso se puede decir que:

CF = 20 Horas-Hombre / Punto de Casos de uso.

$$E = UCP \times CF$$

$$E = 78.68245 * 20$$

$$E = 1573.649 \text{ horas/hombres}$$

Paso 4.

Calcular esfuerzo de todo el proyecto.

Actividad	Porcentaje %	Horas-Hombres
Análisis	10	393.41225
Diseño	20	786.8245
Implementación	40	1573.649
Pruebas	15	590.118375
Sobrecarga (otras actividades)	15	590.118375
Total	100	3934.1225

Si **E_T = 3934.1225 horas-hombre** y se estima que cada mes tiene como promedio 300 horas laborables, eso daría un **E_T = 13.1 mes-hombre**.

Esto quiere decir que 1 persona puede realizar el problema analizado en 13 meses.

Conclusiones.

En este capítulo se realizó un análisis del funcionamiento del negocio a informatizar. Además se definieron los procesos a informatizar así como los actores, trabajadores y casos de uso del negocio, quedaron definidos los requisitos funcionales y no funcionales a través de los cuales se obtuvieron los casos de uso del sistema también obtenidos en este capítulo.

CAPÍTULO

III

3. Diseño del sistema propuesto.

Introducción.

Este capítulo está dirigido fundamentalmente al diseño del sistema propuesto. Se representan los diagramas de secuencia, los diagramas de clases del diseño, las descripciones de las clases identificadas y el diseño de la base de datos mediante el modelo de datos, con la descripción de cada una de las tablas.

3.1- 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 con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar.

3.2- Diagramas de interacción.

Los diagramas de interacción modelan los aspectos dinámicos del sistema, mostrando las interacciones entre objetos mediante transferencia de mensajes entre ellos. Dentro de estos diagramas están los de secuencia y los de colaboración. En el presente trabajo se emplearon los **diagramas de secuencia** teniendo en cuenta que muestran las interacciones entre objetos, ordenadas en secuencia temporal durante un escenario concreto. Los mismos pueden ser encontrados en el **anexo III**.

3.3- Diagrama de clases del diseño.

Con el objetivo de comprender mejor el diagrama de clases del diseño las mismas han sido divididas en 4 paquetes atendiendo a las funcionalidades que representan. Dichos paquetes a su vez han sido divididos en otros subpaquetes para una mejor organización y comprensión del mismo, así como para lograr un mayor desacoplamiento.

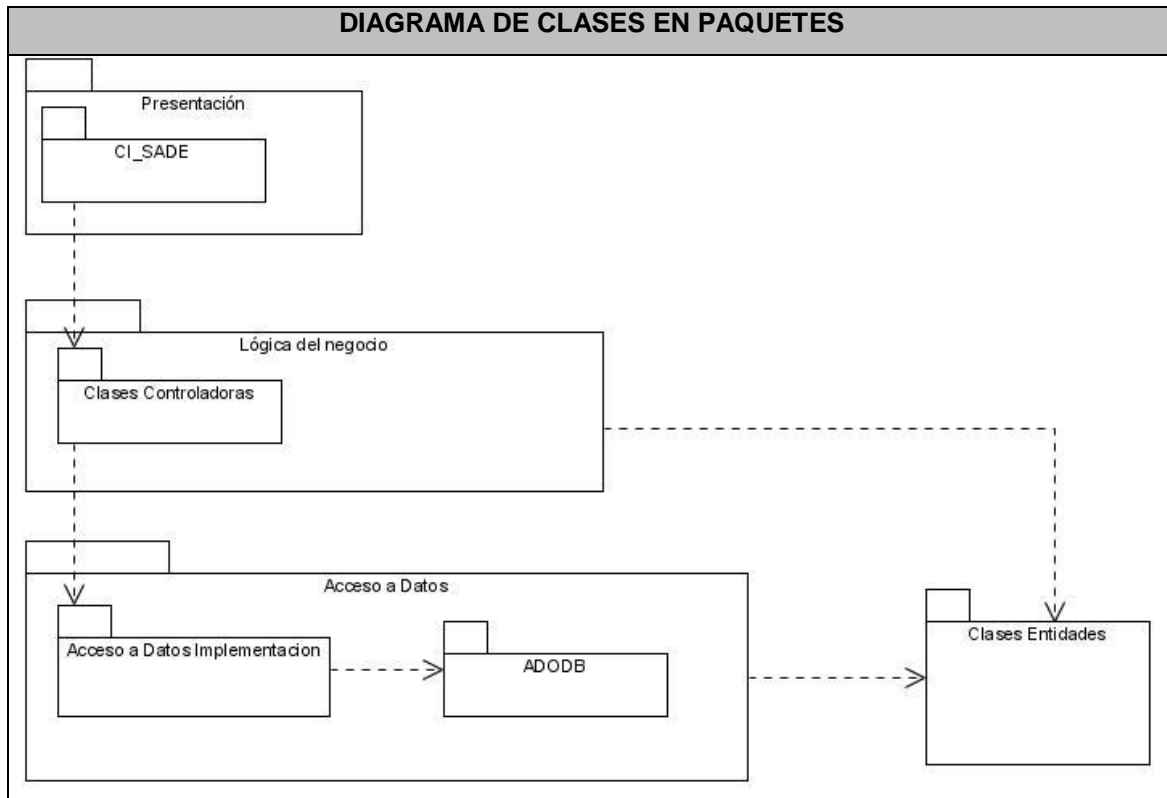


Figura 10 Diagrama de Clases en Paquetes

3.4- Descripción de cada uno de los paquetes.

El paquete **Acceso Datos** contiene las clases que hacen posible la persistencia y recuperación de objetos. Está dividido en dos subpaquetes: el subpaquete **Acceso a Datos de la Implementación**, que contiene las clases encargadas de acceder a la base de datos para manipular la persistencia de las entidades, y el subpaquete **ADO**, utilizado por el paquete anterior, donde se encuentran un conjunto de clases que permiten conectarse a diferentes tipos de base de datos. El Paquete **Acceso a Datos** en general permite a la aplicación abstraerse del origen de los datos y de la lógica de su persistencia, logrando un bajo acoplamiento entre sus componentes.

El paquete **Lógica del Negocio** contiene la lógica de negocio del servicio web. Contiene un paquete **Clases Controladoras** de las entidades relacionadas en el negocio. Estas clases permiten diversas funciones como la validación de datos.

El paquete **Entidades** contiene clases que no tienen comportamiento, sólo propiedades y son representaciones de entidades reales del dominio. La mayoría son clases persistentes que son accedidas por las clases de los paquetes **Lógica del Negocio** y **AccesoDatos**. Por último, el paquete **Presentación** contiene la clase interfaz del servicio web, donde

están registrados todos los servicios que se exponen en WSDL (lenguaje de descripción del servicio web).

Diagramas de clases por paquetes.

Los diagramas de clases correspondientes a cada paquete del diagrama general representado anteriormente se encuentran en el **anexo IV**.

3.5- Descripción de las clases del diseño.

Tabla3. 1 Descripción de la Clase Interfaz. CI_SADE.

Nombre : CI_SADE	
Tipo de clase: Interfaz	
Atributo	Tipo
----	-----
Responsabilidades:	
insertar_departamento (Nombre:xsd:string, key:xsd:string)	
Descripción: Servicio encargado de insertar un nuevo departamento dado los datos enviados.	
buscar_departamento (Nombre:xsd:string)	
Descripción: Servicio encargado de devolver uno o varios departamentos en un arreglo dado el nombre del departamento.	
buscardepartamentoXid(id_dpto:xsd:int)	
Descripción: Servicio encargado de devolver datos de un departamento dado su id.	
get_all_departamentos()	
Descripción: Servicio encargado de devolver todos los departamentos	
actualizar_departamento(Nombre:xsd:string, id:xsd:int, key:xsd:string)	
Descripción: Servicio encargado de actualizar los datos de un departamento dado los datos enviados.	
insertar_disciplina(Nombre:xsd:string, id:xsd:int, key:xsd:string)	
Descripción: Servicio encargado de insertar una nueva disciplina dado los datos enviados.	
buscar_disciplina_x_iddepartamento(id_departamento:xsd:int)	
Descripción: Servicio encargado de devolver los datos de las disciplinas dado el id del departamento.	
buscar_disciplina_x_iddisciplina(id_disciplina:xsd:int)	
Descripción: Servicio encargado de devolver los datos de una disciplina dado su ID.	
actualizar_disciplina(Nombre:xsd:string, id_dpto:xsd:int, id_disciplina:xsd:int, key:xsd:string)	
Descripción: Servicio encargado de actualizar datos de una disciplina.	
insertar_asignatura(NombreAsignatura:xsd:string,id_Departamento:xsd:int,id_disciplina:xsd:int, anno:xsd:string, key:xsd:string)	
Descripción: Servicio encargado de insertar una nueva asignatura dado los datos enviados.	
buscar_asignaturaXiddisciplina(id_disciplina:xsd:int)	
Descripción: Servicio encargado de buscar las asignatura dado el id de la disciplina.	
buscar_asignaturaXidasignatura(id_asignatura:xsd:int)	
Descripción: Servicio encargado de devolver los datos de una asignatura dado su id.	
actualizar_asignatura(NombreAsignatura:xsd:string, id_Departamento:xsd:int, id_disciplina:xsd:int, anno:xsd:string, id_asignatura:xsd:int, key:xsd:string)	

Descripción: Servicio encargado de actualizar los datos de una asignatura.
buscar_persona(nombre:xsd:string, id_dpto:xsd:int)
Descripción: Servicio encargado de devolver los datos de las personas dado id del departamento y nombre.
buscarsepersonaXci(ci:xsd:string)
Descripción: Servicio encargado de devolver los datos de una persona dado su CI.
buscarprofesorXci(ci:xsd:string)
Descripción: Servicio encargado de devolver los datos de un profesor su dado su CI.
buscarprofesorXdepartamento(id_departamento:xsd:int)
Descripción: Servicio encargado de devolver los datos de los profesores de un departamento dado id del departamento.
insertar_profesor_interno_general(id_Departamento:xsd:int, MilitanciaUJC:xsd:string, id_Tipo:xsd:int, NombreApellidos:xsd:string, CI:xsd:string, Solapin:xsd:string, Correo:xsd:string, id_disciplina:xsd:int, id_asignatura:xsd:int, Categoriacientifica:xsd:string, Categoriadocente:xsd:string, MilitanciaPCC:xsd:string, apto:xsd:int, Telefono_uci:xsd:int, LaboratorioProd:xsd:int, telefono_lab:xsd:int, Provincia:xsd:string, Municipio:xsd:string, Entre_Calle:xsd:string,Numero_Casa:xsd:int,Reparto:xsd:string,Edificio:xsd:int, Apartamento:xsd:int, Calle:xsd:string, key:xsd:string)
Descripción: Servicio encargado de insertar un profesor interno dado los datos enviados. Escribe información en las tablas persona, profesor, profesor interno, dirección.
actualizar_profesor_interno_general (id_Departamento:xsd:int, MilitanciaUJC:xsd:string, id_Tipo:xsd:int, NombreApellidos:xsd:string, CI:xsd:string, Solapin:xsd:string, Correo:xsd:string, id_disciplina:xsd:int, id_asignatura:xsd:int, Categoriacientifica:xsd:string, Categoriadocente:xsd:string, MilitanciaPCC:xsd:string, apto:xsd:int, Telefono_uci:xsd:int, LaboratorioProd:xsd:int, telefono_lab:xsd:int, id_direccion:xsd:int ,Provincia:xsd:string, Municipio:xsd:string, Entre_Calle:xsd:string, Numero_Casa:xsd:int, Reparto:xsd:string, Edificio:xsd:int, Apartamento:xsd:int, Calle:xsd:string,key:xsd:string)
Descripción: Servicio encargado de actualizar un profesor interno dado los datos enviados. Actualiza información en las tablas persona, profesor, profesor interno, dirección.
eliminar_profesor_interno_general(ci:xsd:string, key:xsd:string)
Descripción: Servicio encargado de eliminar un profesor interno dado su CI. Elimina los datos de las tablas persona, profesor, profesor interno, dirección y grupo_profesor.
buscarprofesorinternoXci(ci:xsd:string)
Descripción: Servicio encargado de devolver los datos de un profesor interno su dado su CI.
buscardireccionXci(ci:xsd:string)
Descripción: Servicio encargado de devolver los datos de una direccion dirección dado el CI de la persona.
insertar_profesor_externo_general(id_Departamento:xsd:int, MilitanciaUJC:xsd:string, id_Tipo:xsd:int, NombreApellidos:xsd:string, CI:xsd:string, Solapin:xsd:string, Correo:xsd:string, id_disciplina:xsd:int, id_asignatura:xsd:int, Categoriacientifica:xsd:string, Categoriadocente:xsd:string, MilitanciaPCC:xsd:string, institución:xsd:string, Telf_Institucion:xsd:int ,Provincia:xsd:string, Municipio:xsd:string, Entre_Calle:xsd:string, Numero_Casa:xsd:int, Reparto:xsd:string, Edificio:xsd:int, Apartamento:xsd:int, Calle:xsd:string, key:xsd:string)
Descripción: Servicio encargado de insertar un profesor externo dado los datos enviados. Escribe información en las tablas persona, profesor, profesor externo y dirección.
actualizar_profesor_externo_general(id_Departamento:xsd:int, MilitanciaUJC:xsd:string, id_Tipo:xsd:int, NombreApellidos:xsd:string, CI:xsd:string, Solapin:xsd:string, Correo:xsd:string, id_disciplina:xsd:int, id_asignatura:xsd:int, Categoriacientifica:xsd:string, Categoriadocente:xsd:string, MilitanciaPCC:xsd:string, institución:xsd:string, Telf_Institucion:xsd:int, id_direccion:xsd:int ,Provincia:xsd:string, Municipio:xsd:string, Entre_Calle:xsd:string, Numero_Casa:xsd:int, Reparto:xsd:string, Edificio:xsd:int, Apartamento:xsd:int, Calle:xsd:string, key:xsd:string)
Descripción: Servicio encargado de actualizar los datos de un profesor externo dado los datos enviados. Actualiza información en las tablas persona, profesor, profesor externo y dirección.

eliminar_profesor_externo_general(ci:xsd:string, key:xsd:string)
Descripción: Servicio encargado de eliminar un profesor interno dado el CI. Elimina los datos de las tablas persona, profesor, profesor externo, dirección y grupo_profesor.
buscarprofesorexternoXci(ci:xsd:string)
Descripción: Servicio encargado de buscar los datos de un profesor externo dado CI.
insertar_profesor_colaborador_general(id_Departamento:xsd:int, MilitanciaUJC:xsd:string, id_Tipo:xsd:int, NombreApellidos:xsd:string, CI:xsd:string, Solapin:xsd:string, Correo:xsd:string, id_disciplina:xsd:int, id_asignatura:xsd:int, Categoriacientifica:xsd:string, Categoriadocente:xsd:string, MilitanciaPCC:xsd:string, AreaTrabajo:xsd:string ,Provincia:xsd:string, Municipio:xsd:string, Entre_Calle:xsd:string, Numero_Casa:xsd:int, Reparto:xsd:string, Edificio:xsd:int, Apartamento:xsd:int, Calle:xsd:string, key:xsd:string)
Descripción: Servicio encargado de insertar un profesor colaborador de dado los datos enviados. Escribe información en las tablas persona, profesor, profesor colaborador y dirección.
actualizar_profesor_colaborador_general(id_Departamento:xsd:int, MilitanciaUJC:xsd:string, id_Tipo:xsd:int, NombreApellidos:xsd:string, CI:xsd:string, Solapin:xsd:string, Correo:xsd:string, id_disciplina:xsd:int, id_asignatura:xsd:int, Categoriacientifica:xsd:string, Categoriadocente:xsd:string, MilitanciaPCC:xsd:string, AreaTrabajo:ar xsd:string, id_direccion:xsd:int , Provincia:xsd:string, Municipio:xsd:string, Entre_Calle:xsd:string, Numero_Casa:xsd:int, Reparto:xsd:string, Edificio:xsd:int, Apartamento:xsd:int, Calle:xsd:string, key:xsd:string)
Descripción: Servicio encargado de actualizar los datos de un profesor colaborador dado los datos enviados. Actualiza información en las tablas persona, profesor, profesor colaborador y dirección.
eliminar_profesor_colaboraor_general(ci:xsd:string, key:xsd:string)
Descripción: Servicio encargado de eliminar un profesor colaborador dado los datos enviados. Elimina los datos de las tablas persona, profesor, profesor colaborador, dirección y grupo_profesor.
buscarprofesorcolaboradorXci(ci:xsd:string)
Descripción: Servicio encargado de devolver los datos de un profesor colaborador dado su CI.
insertar_alumno_ayudante_general(id_Departamento:xsd:int, MilitanciaUJC:xsd:string, id_Tipo:xsd:int, NombreApellidos:xsd:string, CI:xsd:string, Solapin:xsd:string, Correo:xsd:string, id_disciplina:xsd:int, id_asignatura:xsd:int, Telefono:xsd:int, Apartamento:xsd:int, ProfesorTutor:xsd:string ,Provincia:xsd:string, Municipio:xsd:string, Entre_Calle:xsd:string, Numero_Casa:xsd:int, Reparto:xsd:string, Edificio:xsd:int, Apartamento:xsd:int, Calle:xsd:string, key:xsd:string)
Descripción: Servicio encargado de insertar un alumno ayudante dado los datos enviados. Escribe información en las tablas persona, alumno_ayudante y dirección.
actualizar_alumno_ayudante_general(id_Departamento:xsd:int, MilitanciaUJC:xsd:string, id_Tipo:xsd:int, NombreApellidos:xsd:string, CI:xsd:string, Solapin:xsd:string, Correo:xsd:string, id_disciplina:xsd:int, id_asignatura:xsd:int, Telefono:xsd:int, Apartamentoo:xsd:int, ProfesorTutor:xsd:string,id_direccion:xsd:int Provincia:xsd:string, Municipio:xsd:string, Entre_Calle:xsd:string, Numero_Casa:xsd:int, Reparto:xsd:string, Edificio:xsd:int, Apartamento:xsd:int, Calle:xsd:string, key:xsd:string)
Descripción: Servicio encargado de actualizar un alumno ayudante dado los datos enviados. Actualiza información en las tablas persona, alumno_ayudante y dirección.
eliminar_alumno_ayudante_general(ci:xsd:string, key:xsd:string)
Descripción: Servicio encargado de eliminar un alumno ayudante dado los datos enviados. Elimina información de las tablas persona, alumno_ayudante, dirección y grupo_profesor.
buscaralumnoayudanteXci(ci:xsd:string)
Descripción: Servicio encargado de devolver los datos de un alumno ayudante dado su CI.
insertar_Actividad_docente_metodologica(semestre:xsd:string,responsable:xsd:string, actividad:xsd:string,aseguramientos:xsd:string,participantes:xsd:string,id_Departamento:xsd:int, arreglo_meses:tns:Arreglo_meses,key:xsd:string)
Descripción: Servicio encargado de adicionar una nueva actividad al plan de trabajo DM de un departamento.
actualizar_Actividad_docente_metodologica(id_actividad:xsd:int,semestre:xsd:string, responsable:xsd:string,actividad:xsd:string,aseguramientos:xsd:string,participantes:xsd:string, id_Departamento:xsd:int, arreglo_meses:tns:Arreglo_meses,key:xsd:string)

Descripción: Servicio encargado de actualizar los datos de una actividad del plan de trabajo DM del un departamento.
buscar_actividadesDM(id_dpto :xsd:int, semestre:xsd:string)
Descripción: Servicio encargado de devolver los datos de las actividades DM dado el id del departamento y el semestre en que fue planificada.
buscar_actividadesDM_x_id(id_actividad :xsd:int)
Descripción: Servicio encargado de devolver los datos de una actividad DM dado su ID.
obtenerMesesActividad(id_actividad :xsd:int)
Descripción: Servicio encargado de obtener meses en los cuales se planificó una actividad DM dado el id de la actividad.
insertar_actividad(actividad:xsd:string,mes:xsd:int,lugar:xsd:string,id_Departamento:xsd:int, fecha:xsd:string, responsable:xsd:string,key:xsd:string)
Descripción: Servicio encargado de adicionar una actividad al plan de trabajo de un departamento.
buscar_actividad(id_dpto:xsd:int, mes:xsd:int)
Descripción: Servicio encargado de devolver los datos de las actividades del plan de trabajo dado el id del departamento y el mes en que fueron planificadas.
buscar_actividad_x_id(id_actividad:xsd:int)
Descripción: Servicio encargado de devolver los datos de una actividad del plan de trabajo dado el id de la actividad.
actualizar_actividad(id_actividad:xsd:int,actividad:xsd:string,mes:xsd:int,lugar:xsd:string, id_Departamento:xsd:int, fecha:xsd:string, responsable:xsd:string,key:xsd:string)
Descripción: Servicio encargado de actualizar los datos de una actividad del plan de trabajo del departamento.
registrar_asistencia(id_actividad:xsd:int, arreglo_personas:tns:Arreglo_registro, key:xsd:string)
Descripción: Servicio encargado insertar un registro de asistencia en la BD. El registro se compone por: id de la actividad y CI de la persona. Recibe por parámetro un arreglo con todos los CI de las personas que asistieron a la actividad.
registrosXidactividad(id_actividad:xsd:int)
Descripción: Servicio encargado de devolver todos los registros de asistencia a una actividad dado id_actividad.
actualizar_registro_asistencia(id_actividad:xsd:int, arreglo_personas:tns:Arreglo_registro, key:xsd:string)
Descripción: Servicio encargado de actualizar el registro de la asistencia a una actividad.
insertar_grupo(anno:xsd:string, Nombre_Grupo:xsd:string, key:xsd:string)
Descripción: Servicio encargado de insertar un nuevo grupo dado los datos enviados.
buscar_grupo(nombre:xsd:string)
Descripción: Servicio encargado de devolver los datos de los grupos dado el nombre.
buscar_grupoXanno(anno:xsd:string)
Descripción: Servicio encargado de devolver los datos de los grupos dado el año.
buscar_grupoXid(id_grupo:xsd:int)
Descripción: Servicio encargado de devolver los datos de un grupo dado su id.
actualizar_grupo(id_grupo:xsd:int, anno:xsd:string,Nombre_Grupo:xsd:string,key:xsd:string)
Descripción: Servicio encargado de actualizar los datos de un grupo dado los datos enviados.
eliminar_grupo_general(id_grupo:xsd:int, key:xsd:string)
Descripción: Servicio encargado de eliminar un grupo. Elimina también de la tabla grupo_profesor las asignaciones donde se halle el grupo a eliminar.
asignar_grupos(CI:xsd:string,arreglo_grupos:tns:Arreglo_grupoprofesor,key:xsd:string)
Descripción: Servicio encargado de asignar los grupos que atenderá una persona. Recibe por parámetro el CI de la persona y un arreglo con los identificadores de los grupos que atenderá.
buscarGruposAsignadosXci(ci:xsd:string)
Descripción: Servicio encargado de buscar los grupos que atiende una persona dado el CI.
terminar_asignacion(CI:xsd:string, arreglo_grupos:tns:Arreglo_grupoprofesor, key:xsd:string)
Descripción: Servicio encargado de terminar la asignación profesor-grupo dado id grupo y CI. Recibe por parámetro el CI de la persona y un arreglo con los identificadores de los grupos que ya no atenderá.

insertar_postgrado(nestado:xsd:string, Fecha_inicio:xsd:string, Fecha_Fin:xsd:string, CursoEScolar:xsd:string, Semestre:xsd:string, ombrePostgrado:xsd:string, horarios:xsd:string, key:xsd:string)
Descripción: Servicio encargado de insertar un nuevo postgrado.
buscarpostgradoXestado(estado:xsd:string)
Descripción: Servicio encargado de devolver los datos de los postgrados dado estado.
buscarpostgradoXnombre(nombre:xsd:string)
Descripción: Servicio encargado de devolver los datos de los postgrados dado nombre.
buscarpostgradoXid(id_postgrado:xsd:int)
Descripción: Servicio encargado de devolver los datos de un postgrado dado su id.
actualizar_postgrado(id_Postgrado:xsd:int, nestado:xsd:string, Fecha_inicio:xsd:string, Fecha_Fin:xsd:string, CursoEScolar:xsd:string, Semestre:xsd:string, NombrePostgrado:xsd:string, Horarios:xsd:string, key:xsd:string)
Descripción: Servicio encargado de actualizar los datos de un postgrado.
matricular_postgrado(CI:xsd:string, id_Postgrado:xsd:int, nestado:xsd:string, key:xsd:string)
Descripción: Servicio registrar la matrícula de un postgrado. Recibe por parámetro el id del postgrado y el CI del profesor que se está matriculando.
registrar_profesor_postgrado(CI:xsd:string, id_Postgrado:xsd:int, nestado:xsd:string, key:xsd:string)
Descripción: Servicio encargado de registrar el profesor que impartirá un postgrado. Recibe por parámetro el id del postgrado y el CI del profesor que impartirá el postgrado.
buscar_personas_x_categoria(id_dpto:xsd:int, categoría:xsd:int)
Descripción: Servicio encargado de devolver los datos de las personas dado el id del departamento y el tipo de persona (profesor externo, interno, colaborador o alumno ayudante).
buscar_personas_x_departamento(id_dpto:xsd:int)
Descripción: Servicio encargado de devolver los datos de personas dado id del departamento.
ObtenerPersonalIdentificacion(solapin:xsd:string, ci:xsd:string)
Descripción: Servicio que devuelve los datos de las personas obtenidos mediante el servicio de identificación de la Universidad. Recibe por parámetros un CI o una Solapín.

Tabla 3. 2 Descripción de la Clase Entidad: ce_actividades.

Nombre: ce_actividades	
Tipo de clase: Entidad	
Atributo	Tipo:
Id_actividad	Var
Nombre	Var
Fecha	Var
Responsable	Var
Obj_departamento	Var
Responsabilidades:	
get_idActividad()	
Descripción: Responsable de devolver el id de la actividad.	
get_name()	
Descripción: Responsable de devolver el nombre de la actividad.	
get_fecha()	
Descripción: Responsable de devolver la fecha en que se planifica la actividad.	
get_responsable()	
Descripción: Responsable de devolver el responsable de la actividad.	
get_objDepartamento()	
Descripción: Responsable de devolver el id del departamento en el cual se planifica la actividad.	

Tabla 3. 3 Descripción de la Clase Entidad: ce_alumno_ayudante

Nombre: ce_alumno_ayudante	
Tipo de clase: Entidad	
Atributo	Tipo
Teléfono	Var
Apartamento	Var
ProfesorTutor	Var
CI	Var
Responsabilidades	
get_CI()	
Descripción: Responsable de devolver el carnet de identidad del alumno ayudante.	
get_telefono()	
Descripción: Responsable de devolver el teléfono del apartamento.	
get_apartamento()	
Descripción: Responsable de devolver el apartamento del alumno ayudante.	
get_profesorTutor()	
Descripción: Responsable de devolver el profesor Tutor del alumno ayudante.	

Tabla 3. 4 Descripción de la Clase Entidad: ce_asignatura.

Nombre: ce_asignatura	
Tipo de clase: Entidad	
Atributo	Tipo
Id	Var
Nombre	Var
Anno	Var
obj_departamento	Var
Obj_disciplina	Var
Responsabilidades	
get_anno()	
Descripción: Responsable de devolver el año en que se imparte la asignatura.	
get_objDisciplina()	
Descripción: Responsable de devolver el id de la disciplina a la cual pertenece la asignatura.	
get_objDepartamento()	
Descripción: Responsable de devolver el id del departamento al cual pertenece la asignatura.	
get_id()	
Descripción: Responsable de devolver el id de la asignatura.	
get_nombre()	
Descripción: Responsable de devolver el nombre de la asignatura.	

Tabla 3. 5 Descripción de la Clase Entidad: ce_asistencia_actividades

Nombre: ce_asistencia_actividades	
Tipo de clase: Entidad	
Atributo	Tipo

id_actividad	Var
Asistente	Var
Responsabilidades	
get_idactividad()	
Descripción: Responsable de devolver el id de la actividad.	
get_asistente()	
Descripción: Responsable de devolver el CI de la persona que asistió a la actividad.	

Tabla 3. 6 Descripción de la Clase Entidad: ce_departamento.

Nombre: ce_departamento	
Tipo de clase: Entidad	
Atributo	Tipo
Name	Var
Id	Var
Responsabilidades	
get_id()	
Descripción: Responsable de devolver el id del departamento.	
get_name()	
Descripción: Responsable de devolver el nombre del departamento.	

Tabla 3. 7 Descripción de la Clase Entidad: ce_direccion

Nombre: ce_direccion	
Tipo de clase: Entidad	
Atributo	Tipo
Provincia	Var
municipio	Var
entreCalle	Var
numeroCasa	Var
reparto	Var
edificio	Var
apartamento	Var
calle	Var
obj_persona	Var
id_direccion	Var
Responsabilidades	
get_provincia()	
Descripción: Responsable de devolver el nombre de la provincia	
get_municipio()	
Descripción: Responsable de devolver el nombre del municipio.	
get_entrecalle()	
Descripción: Responsable de devolver el nombre de la entre calle.	

get_numeroCasa()
Descripción: Responsable de devolver el numero de la casa.
get_reparto()
Descripción: Responsable de devolver el nombre del reparto.
get_edificio()
Descripción: Responsable de devolver el número del edificio.
get_apartamento()
Descripción: Responsable de devolver el número del apartamento
get_calle()
Descripción: Responsable de devolver el nombre de la calle.
get_objPersona()
Descripción: Responsable de devolver el CI de la persona.
get_id()
Descripción: Responsable de devolver el id de la dirección.

Tabla3. 8 Descripción de la Clase Entidad: ce_disciplina

Nombre: ce_disciplina	
Tipo de clase: Entidad	
Atributo	Tipo
nombre	Var
id	Var
obj_departamento	Var
Responsabilidades	
get_id()	
Descripción: Responsable de devolver el id de la disciplina.	
get_nombre()	
Descripción: Responsable de devolver el nombre de la disciplina.	
get_objDepartamento()	
Descripción: Responsable de devolver el id del departamento al cual pertenece la disciplina.	

Tabla3. 9 Descripción de la Clase Entidad: ce_persona

Nombre: ce_persona	
Tipo de clase: Entidad	
Atributo	Tipo
obj_departamento	Var
obj_tipoPersona	Var
MilitanciaUJC	Var
NombreApellidos	Var
CI	Var
Solapin	Var
Correo	Var
obj_disciplina	Var
obj_asignatura	Var

Responsabilidades
get_objDisciplina()
Descripción: Responsable de devolver el id de la disciplina a la cual pertenece la persona.
get_objAsignatura()
Descripción: Responsable de devolver el id de la asignatura que imparte la persona.
get_objDepartamento()
Descripción: Responsable de devolver el id del departamento al cual pertenece la persona.
get_militUJC()
Descripción: Responsable de devolver la militancia de la UJC.
get_name()
Descripción: Responsable de devolver el nombre de la persona.
get_CI()
Descripción: Responsable de devolver el carnet de identidad de la persona.
get_Correo()
Descripción: Responsable de devolver el correo de la persona.
get_solapin()
Descripción: Responsable de devolver el solapin.
get_objTipoPersona()
Descripción: Responsable de devolver el tipo de persona.

Tabla3. 10 Descripción de la Clase Entidad: ce_grupo_profesor

Nombre: ce_grupo_profesor	
Tipo de clase: Entidad	
Atributo	Tipo
id_grupo	Var
id_persona	Var
Responsabilidades	
get_idgrupo()	
Descripción: Responsable de devolver el id del grupo.	
get_idpersona()	
Descripción: Responsable de devolver el id de la persona (CI).	

Tabla3. 11 Descripción de la Clase Entidad: ce_grupos

Nombre: ce_grupos	
Tipo de clase: Entidad	
Atributo	Tipo
Nombregrupo	Var
id	Var
anno	Var
Responsabilidades	
get_id()	
Descripción: Responsable de devolver el id del grupo.	
get_Nombregrupo()	
Descripción: Responsable de devolver el nombre del grupo.	
get_anno()	
Descripción: Responsable de devolver el año del grupo.	

Tabla3. 12 Descripción de la Clase Entidad: ce_meses

Nombre: ce_meses	
Tipo de clase: Entidad	
Atributo	Tipo
id_actividad	Var
nombre	Var
Responsabilidades	
get_actividad()	
Descripción: Responsable de devolver el id de la actividad DM.	
get_nombre()	
Descripción: Responsable de devolver el nombre del mes.	

Tabla3. 13 Clase Entidad: ce_plan_actividades

Nombre: ce_plan_actividades	
Tipo de clase: Entidad	
Atributo	Tipo
id_actividad	Var
mes	Var
lugar	Var
fecha	Var
responsable	Var
actividad	Var
obj_departamento	Var
Responsabilidades	
get_actividad()	
Descripción: Responsable de devolver la descripción de la actividad.	
get_idactividad()	
Descripción: Responsable de devolver el id de la actividad.	
get_mes()	
Descripción: Responsable de devolver el mes en que se planifica la actividad.	
get_lugar()	
Descripción: Responsable de devolver el lugar donde se planifica la actividad.	
get_fecha()	
Descripción: Responsable de devolver la fecha en que se planifica a actividad.	
get_responsable()	
Descripción: Responsable de devolver el responsable de la actividad.	
get_objdepartamento()	
Descripción: Responsable de devolver el id del departamento en el que se planifica la actividad.	

Tabla3. 14 Descripción de la Clase Entidad: ce_planDM

Nombre: ce_planDM	
Tipo de clase: Entidad	
Atributo	Tipo

Id	Var
semestre	Var
actividad	Var
responsable	Var
aseguramiento	Var
participantes	Var
obj_departamento	Var
Responsabilidades	
get_id()	
Descripción: Responsable de devolver el id de la actividad DM.	
get_semestre()	
Descripción: Responsable de devolver el semestre en que se planifica la actividad DM.	
get_actividad()	
Descripción: Responsable de devolver la descripción de la actividad DM.	
get_responsable()	
Descripción: Responsable de devolver el responsable de la actividad DM.	
get_objdepartamento()	
Descripción: Responsable de devolver el id del departamento en el cual se planifica la actividad DM.	
get_aseguramiento()	
Descripción: Responsable de devolver la descripción de los aseguramientos para la actividad DM.	
get_participantes()	
Descripción: Responsable de devolver los participantes.	

Tabla3. 15 Descripción de la Clase Entidad: ce_postgrado

Nombre: ce_postgrado	
Tipo de clase: Entidad	
Atributo	Tipo
id_postgrado	Var
fecha_inicio	Var
fecha_fin	Var
curso_escolar	Var
semestre	Var
nombre_postgrado	Var
Nestado	Var
horarios	Var
Responsabilidades	
get_horario()	
Descripción: Responsable de devolver una descripción del horario del postgrado.	
get_idPostgrado()	
Descripción: Responsable de devolver el id del postgrado.	
get_fechaInicio()	
Descripción: Responsable de devolver la fecha de inicio del postgrado.	
get_fechaFin()	
Descripción: Responsable de devolver la fecha de fin del postgrado.	
get_cursoescolar()	

Descripción: Responsable de devolver el curso escolar.
get_semestre()
Descripción: Responsable de devolver el semestre en que se planifica el postgrado.
get_nombrepostgrado()
Descripción: Responsable de devolver el nombre del postgrado.
get_nestado()
Descripción: Responsable de devolver el estado del postgrado (iniciado, impartándose, terminado).

Tabla3. 16 Descripción de la Clase Entidad: ce_profesor

Nombre: ce_profesor	
Tipo de clase: Entidad	
Atributo	Tipo
Categoria_cientifica	Var
Categoria_docente	Var
MilitanciaPCC	Var
CI	Var
Responsabilidades	
get_CI()	
Descripción: Responsable de devolver el CI del profesor.	
get_categoriaCientif()	
Descripción: Responsable de devolver la categoría científica del profesor.	
get_CategoriaDocente()	
Descripción: Responsable de devolver la categoría docente del profesor.	
get_militanciaPCC()	
Descripción: Responsable de devolver la militancia del PCC.	

Tabla3. 17 Descripción de la Clase Entidad: ce_profesorColaborador

Nombre: ce_profesorColaborador	
Tipo de clase: Entidad	
Atributo	Tipo
areaTrabajo	Var
CI	Var
Responsabilidades	
get_CI()	
Descripción: Responsable de devolver el CI del profesor colaborador.	
get_AreaTrabajo()	
Descripción: Responsable de devolver el área de trabajo del profesor colaborador.	

Tabla3. 18 Descripción de la Clase Entidad: ce_profesor_externo

Nombre: ce_profesor_externo	
Tipo de clase: Entidad	
Atributo	Tipo
Institucion	Var

telf_institucion	Var
CI	Var
Responsabilidades	
get_CI()	
Descripción: Responsable de devolver el CI.	
get_institucion()	
Descripción: Responsable de devolver la institución que proviene el profesor externo.	
get_telfInstitucion()	
Descripción: Responsable de devolver el teléfono de la institución que proviene el profesor externo.	

Tabla3. 19 Descripción de la Clase Entidad: ce_profesorInterno

Nombre: ce_profesorInterno	
Tipo de clase: Entidad	
Atributo	Tipo
Apto	Var
telefonoApto	Var
laboratorioP	Var
telf_labP	Var
CI	Var
Responsabilidades	
get_CI()	
Descripción: Responsable de devolver el CI del profesor externo.	
get_apto()	
Descripción: Responsable de devolver el apartamento del profesor interno.	
get_telfApto()	
Descripción: Responsable de devolver el teléfono del apartamento.	
get_labP()	
Descripción: Responsable de devolver el laboratorio de producción del profesor. interno	
get_telefLabP()	
Descripción: Responsable de devolver el teléfono del laboratorio de producción.	

Tabla3. 20 Descripción de la Clase Entidad: ce_profesorPostgrado

Nombre: ce_profesorPostgrado	
Tipo de clase: Entidad	
Atributo	Tipo
id_persona	Var
id_postgrado	Var
estado	Var
Responsabilidades	
get_idpersona()	
Descripción: Responsable de devolver id_persona.	
get_idpostgrado()	
Descripción: Responsable de devolver el id del postgrado.	
get_estado()	
Descripción: Responsable de devolver el estado de la relación (si imparte o recibe el postgrado).	

Tabla3. 21 Descripción de la Clase Controladora: DAO_asignatura

Nombre: DAO_asignatura	
Tipo de clase: controladora	
Atributo	Tipo
-----	-----
Responsabilidades	
insertar_asignatura(asignatura:var)	
Descripción: Responsable de insertar los datos de una asignatura en la BD.	
buscar_X_id(id:var)	
Descripción: Responsable de devolver los datos de una asignatura dado su id.	
buscar_asignatura(id_disciplina:var)	
Descripción: Responsable de devolver todos los datos de las asignaturas dado el id de la disciplina.	
actualizar_asignatura(asignatura:var)	
Descripción: Responsable de actualizar los datos de una asignatura en la BD.	

Tabla3. 22 Descripción de la Clase Controladora: DAO_Departamentos

Nombre: DAO_Departamentos	
Tipo de clase: controladora	
Atributo	Tipo
-----	-----
Responsabilidades	
insert_departamento(departamento:var)	
Descripción: Responsable de insertar los datos de un departamento en la BD.	
buscar_dpto(id_dpto:var)	
Descripción: Responsable de buscar y devolver los datos de un departamento dado su id.	
buscar_X_nombre(nombre:var)	
Descripción: Responsable de buscar y devolver los datos de uno o varios departamentos dado nombre de departamento.	
actualizar_departamento(ce_departamento :var,departamento:var)	
Descripción: Responsable de actualizar los datos de un departamento en la BD.	

Tabla3. 23 Descripción de la Clase Controladora: DAO_direccion

Nombre: DAO_direccion	
Tipo de clase: controladora	
Atributo	Tipo
-----	-----
Responsabilidades	
insertar_direccion(dirección:var)	
Descripción: Responsable de insertar los datos de una dirección en la BD.	
actualizar_direccion(dirección: var)	
Descripción: Responsable de actualizar los datos de una dirección en la BD.	
buscar_direccion(ci:var)	
Descripción: Responsable de buscar y devolver los datos de una dirección dado el CI de la persona.	
eliminar_direccion(ci:var)	
Descripción: Responsable de eliminar los datos de una dirección dado el CI de la persona	

Tabla3. 24 Descripción de la Clase Controladora: DAO_disciplina

Nombre: DAO_disciplina	
Tipo de clase: controladora	
Atributo	Tipo
-----	-----
Responsabilidades	
insertar_disciplina(disciplina:var)	
Descripción: Responsable de insertar los datos de una disciplina en la BD.	
buscarDisciplina_X_iddpto(id_departamento:var)	
Descripción: Responsable de buscar y devolver los datos de una o varias disciplinas dado el id del departamento.	
buscarDisciplina_X_id(id:var)	
Descripción: Responsable de buscar y devolver los datos de una disciplina dado su id.	
actualizar_disciplina(disciplina:var)	
Descripción: Responsable de actualizar los datos de una disciplina en la BD.	

Tabla3. 25 Descripción de la Clase Controladora: dao_grupo_profesor

Nombre: dao_grupo_profesor	
Tipo de clase: controladora	
Atributo	Tipo
-----	-----
Responsabilidades	
insertar_relacion(relación:var)	
Descripción: Responsable de insertar una nueva asignación grupo-profesor. Inserta id_persona e id_grupo creándose así una nueva relación.	
eliminar_relacion(id_grupo:var)	
Descripción: Responsable de eliminar una relación grupo-profesor de la BD dado id_grupo.	
eliminar_relacion1(id_persona:var)	
Descripción: Responsable de eliminar una relación grupo-profesor de la BD dado id_persona.	
eliminar_relacion2(id_persona:var,id_grupo:var)	
Descripción: Responsable de eliminar una relación grupo-profesor de la BD dado id_grupo y id_persona.	
validar(id_grupo:var,id_persona:var)	
Descripción: Responsable de validar que no se asigne un grupo dos veces a un profesor dado id_grupo y id_persona.	
buscar_X_ci(ci:var)	
Descripción: Responsable de buscar y devolver todas la relaciones donde halle una persona dado el CI de la persona.	

Tabla3. 26 Descripción de la Clase Controladora: dao_grupos

Nombre: dao_grupos	
Tipo de clase: controladora	
Atributo	Tipo
-----	-----
Responsabilidades	
insertar_grupo(grupo:var)	
Descripción: Responsable de insertar los datos de un grupo en la BD.	

actualizar_grupo(grupo:var)
Descripción: Responsable de actualizar los datos de un grupo en la BD.
eliminar_grupo(id:var)
Descripción: Responsable de eliminar los datos de un grupo dado id grupo.
eliminar_grupo_general(id_grupo:var)
Descripción: Responsable de eliminar un grupo y todas sus relaciones de la tabla grupo_profesor dado el id del grupo.
buscar_X_nombre(nombre:var)
Descripción: Responsable de buscar y devolver los datos de uno o varios grupos dado el nombre.
buscar_X_anno(anno:var)
Descripción: Responsable de buscar y devolver los datos de uno o varios grupos dado el año.
buscar_X_id(id:var)
Descripción: Responsable de buscar y devolver los datos de un grupo dado el id del grupo.

Tabla3. 27 Descripción de la Clase Controladora: DAO_persona

Nombre: DAO_persona	
Tipo de clase: controladora	
Atributo	Tipo
-----	-----
Responsabilidades	
insert_profesorInterno(prof_interno:var)	
Descripción: Responsable de insertar los datos de un profesor interno en la BD.	
insert_profesorExterno(prof_externo:var)	
Descripción: Responsable de insertar los datos de un profesor externo en la BD.	
insert_profesorColaborador(prof_colaborador:var)	
Descripción: Responsable de insertar los datos de un profesor colaborador en la BD.	
insert_AA(AA:var)	
Descripción: Responsable de insertar los datos de un alumno ayudante en la BD.	
insert_profesor(prof:var)	
Descripción: Responsable de insertar los datos de un profesor en la BD.	
insertar_persona(persona:var)	
Descripción: Responsable de insertar los datos de una persona en la BD.	
buscar_persona_X_Solapin(solapín:var)	
Descripción: Responsable de buscar y devolver los datos de una persona dado el solapín.	
buscar_persona_X_CI(ci:var)	
Descripción: Responsable de buscar y devolver los datos de una persona dado el CI.	
personas_X_departamento(id_dpto:var)	
Descripción: Responsable de buscar y devolver los datos de una o varias personas dado el id del departamento.	
buscar_profesor(ci:var)	
Descripción: Responsable de buscar y devolver los datos de un profesor dado el CI.	
buscar_profesorColaborador(ci:var)	
Descripción: Responsable de buscar y devolver los datos de un profesor colaborador dado el CI.	
buscar_profesorInterno(ci:var)	
Descripción: Responsable de buscar y devolver los datos de un profesor interno dado el CI.	
buscar_profesorExterno(ci:var)	
Descripción: Responsable de buscar y devolver los datos de un profesor externo dado CI.	
buscar_profesorAA(ci:var)	
Descripción: Responsable de buscar y devolver los datos de un alumno ayudante dado CI.	
actualizar_persona(persona:var)	
Descripción: Responsable de actualizar los datos de una persona en la BD.	
actualizar_profesor(profesor:var)	
Descripción: Responsable de actualizar los datos de un profesor en la BD.	

actualizar_profesorExterno(prof_externo:var)
Descripción: Responsable de actualizar los datos de un profesor externo en la BD.
actualizar_profesorInterno(prof_interno:var)
Descripción: Responsable de actualizar los datos de un profesor interno en la BD.
actualizar_profesorColaborador(prof_colaborador:var)
Descripción: Responsable de actualizar los datos de un profesor colaborador en la BD.
actualizar_AlumnoAyudante(AA:var)
Descripción: Responsable de actualizar los datos de un alumno ayudante en la BD.
existe_persona(persona:var)
Descripción: Responsable de comprobar si existe una persona en la BD. Si se encuentra retorna true si no retorna false.
insertar_profesor_interno_general(prof:var,prof_interno:var,dirección:var)
Descripción: Responsable de insertar un profesor interno pasándole como parámetro los datos especificados. Recibe como parámetros objetos de tipo persona, profesor, profesor interno y dirección. Escribe en varias tablas.
actualizar_profesor_interno_general(prof:var, prof_interno:var, dirección:var)
Descripción: Responsable de actualizar los datos de un profesor interno pasándole como parámetro los datos especificados. Recibe como parámetros objetos de tipo persona, profesor, profesor interno y dirección. Actualiza datos en varias tablas.
eliminar_profesor_interno_general(ci:var)
Descripción: Responsable de eliminar los datos de un profesor interno pasándole como parámetro el carnet de identidad. Elimina los datos de la persona de las tablas persona, profesor, profesor interno, dirección y grupo_profesor.
insertar_profesor_externo_general(persona:var,prof:var,prof_externo:var,direccion:var)
Descripción: Responsable de insertar los datos de un profesor externo pasándole como parámetro los datos especificados. Recibe como parámetros objetos de tipo persona, profesor, profesor externo y dirección. Escribe en varias tablas.
actualizar_profesor_externo_general(persona:var,prof:var,prof_externo:var,direccion:var)
Descripción: Responsable de actualizar los datos de un profesor externo pasándole como parámetro los datos especificados. Recibe como parámetros objetos de tipo persona, profesor, profesor externo y dirección. Actualiza datos en varias tablas.
eliminar_profesor_externo_general(ci:var)
Descripción: Responsable de eliminar los datos de un profesor externo pasándole como parámetro el carnet de identidad. Elimina los datos de la persona de las tablas persona, profesor, profesor externo, dirección y grupo_profesor.
insertar_profesor_colaborador_general(persona:var,prof:var,prof_colaborador:var,dirección:var)
Descripción: Responsable de insertar los datos de un profesor colaborador pasándole como parámetro los datos especificados. Recibe como parámetros objetos de tipo persona, profesor, profesor colaborador y dirección. Escribe en varias tablas.
actualizar_profesor_colaborador_general(persona:var,prof:var,prof_colaborador:var,dirección:var)
Descripción: Responsable de actualizar los datos de un profesor colaborador pasándole como parámetro los datos especificados. Recibe como parámetros objetos de tipo persona, profesor, profesor colaborador y dirección. Actualiza datos en varias tablas.
eliminar_profesor_colaborador_general(ci:var)
Descripción: Responsable de eliminar los datos de un profesor colaborador pasándole como parámetro el carnet de identidad. Elimina los datos de la persona de las tablas persona, profesor, profesor colaborador, dirección y grupo_profesor.
insertar_alumno_ayudante_general(ce_persona:var,persona:var,AA:var,dirección:var)
Descripción: Responsable de insertar los datos de un alumno ayudante pasándole como parámetro los datos especificados. Recibe como parámetros objetos de tipo persona, alumno ayudante y dirección. Escribe en varias tablas.
actualizar_alumno_ayudante_general(ce_persona:var,persona:var,AA:var,dirección:var)
Descripción: Responsable de actualizar los datos de un alumno ayudante. Recibe como parámetros objetos de tipo persona, alumno ayudante y dirección. Escribe en varias tablas.
eliminar_alumno_ayudante_general(ci:var)
Descripción: Responsable de eliminar un alumno ayudante pasándole como parámetro el carnet de

identidad.
buscar_persona_x_categoria(id_dpto:var,categoría:var)
Descripción: Responsable de buscar y devolver los datos de las personas según por categoría pasándole por parámetros el id del departamento y la categoría
profesoresXdepartamento(dpto:var)
Descripción: Responsable de buscar y devolver datos de los profesores de un departamento dado el departamento como parámetro.

Tabla3. 28 Descripción de la Clase Controladora: dao_planTrabajo

Nombre: dao_planTrabajo	
Tipo de clase: controladora	
Atributo	Tipo
-----	-----
Responsabilidades	
insertar_plandm(plandm:var)	
Descripción: Responsable de insertar los datos de una actividad del plan de trabajo DM en la BD.	
update_plandm(plandm:var)	
Descripción: Responsable de actualizar los datos de una actividad del plan de trabajo DM en la BD.	
eliminar_plandm(id_actividad:var)	
Descripción: Responsable de eliminar una actividad del plan de trabajo docente metodológico dado el id de la actividad.	
buscar_actividadesDM_X_dpto(id_dpto:var)	
Descripción: Responsable de buscar y devolver los datos de las actividades docentes metodológicas dado el id del departamento en el cual se planifican.	
buscar_actividadesDM_X_id(id:var)	
Descripción: Responsable de insertar los datos de una actividad DM dado el id de la actividad.	
buscar_actividadesDM(id_dpto:var,semestre:var)	
Descripción: Responsable de devolver los datos de las actividades DM dado id departamento y semestre.	
insertar_actividad(plan_Actividades:var)	
Descripción: Responsable de insertar los datos de una actividad del plan de actividades en la BD.	
actualizar_actividad(plan_Actividades:var)	
Descripción: Responsable de actualizar los datos de una actividad del plan de actividades en la BD.	
eliminar_actividad(id_Actividad:var)	
Descripción: Responsable de eliminar los datos de una actividad dado el id de la actividad.	
buscar_actividad_X_dpto(id_dpto:var, mes:var)	
Descripción: Responsable de buscar y devolver los datos de una o varias actividades dado el id del departamento y el mes.	
insertar_registro_asistencia(registro:var)	
Descripción Responsable de insertar los datos del registro de asistencia a actividades (id de la actividad y id de la persona).	
eliminar_registro(id_actividad:var)	
Descripción: Responsable de eliminar los registros de asistencia dado id de la actividad como parámetro.	
valida_registro(id_actividad:var,asistente:var)	
Descripción: Responsable de verificar si una persona ya tiene la asistencia registrada para una actividad dado el id de la actividad y el id de la persona.	
buscar_registro_X_idactividad(id_actividad:var)	
Descripción: Responsable de buscar y devolver registros de asistencia dado el id de actividad.	
insertar_mes(mes:var)	
Descripción: Responsable de insertar en la BD los meses que se planifica una actividad DM. Inserta id de la actividad y el nombre del mes o meses en que se planifica.	
buscarmeses_X_idactividad(id_actividad:var)	

Descripción: Responsable de buscar y devolver los meses en que se planificó una actividad DM dado el id de la actividad.
eliminar_meses_X_idactividad(id_actividad:var)
Descripción: Responsable de eliminar los meses de una actividad DM dado el id de la actividad.

Tabla3. 29 Descripción de la Clase Controladora: dao_postgrados

Nombre: dao_postgrados	
Tipo de clase: controladora	
Atributo	Tipo
-----	-----
Responsabilidades	
insertar_postgrado(postgrado:var)	
Descripción: Responsable de insertar los datos de un postgrado en la BD.	
actualizar_postgrado(postgrado:var)	
Descripción: Responsable de actualizar los datos de un postgrado en la BD.	
buscarpostgrado_X_nombre(nombre:var)	
Descripción: Responsable de buscar y devolver los datos de uno o varios postgrados dado el nombre.	
buscarpostgrado_X_estado(estado:var)	
Descripción: Responsable de buscar y devolver los datos de uno o varios postgrados dado el estado.	
buscarpostgrado_X_idpostgrado(id_postgrado:var)	
Descripción: Responsable de buscar y devolver los datos de un postgrado dado su id.	
insertar_profesor_postgrado(matricula:var)	
Descripción: Responsable de insertar en la tabla profesor_postgrado los datos de una nueva matricula.	
valida_matricula(id_persona:var,id_postgrado:var)	
Descripción: Responsable de validar que una persona no matricule dos veces en el mismo postgrado dado el id del postgrado y el id de la persona (CI).	
valida_profesor(id_postgrado:var)	
Descripción: Responsable de validar que un postgrado no tenga dos profesores asignados.	

Tabla3. 30 Descripción de la Clase Controladora: DAO_select

Nombre: DAO_select	
Tipo de clase: controladora	
Atributo	Tipo
-----	-----
Responsabilidades	
get_all_departamentos()	
Descripción: Responsable de devolver los datos de todos los departamentos.	
get_all_ProfesorColaborador()	
Descripción: Responsable de devolver los datos de todos los profesores colaboradores.	
get_all_profesorExterno()	
Descripción: Responsable de devolver los datos de todos los profesores externos.	
get_all_profesorInterno()	
Descripción: Responsable de devolver los datos de todos los profesores internos.	
get_all_AA()	
Descripción Responsable de devolver los datos de todos los Alumnos Ayudantes.	
get_all_personas()	
Descripción: Responsable de devolver los datos de todas las personas.	
get_all_direcciones()	
Descripción Responsable de devolver los datos de todas las Direcciones.	

get_all_grupos()
Descripción: Responsable de devolver los datos de todos los grupos.
get_all_profesores()
Descripción: Responsable de devolver los datos de todos los profesores.
get_all_grupos_profesor()
Descripción: Responsable de devolver los datos de todos los grupos que tienen profesor.

Tabla3. 31 Descripción de la Clase Controladora: cc_alumno_ayudante

Nombre: cc_alumno_ayudante	
Tipo de clase: Controladora	
Atributo	Tipo
-----	-----
Responsabilidades	
check_insert(alumno_Ayudante:var,persona:var,dirección:var)	
Descripción: Responsable de validar que al insertar los datos del alumno ayudante sean correctos.	
insertar_alumno_ayudante_general(ce_persona:var,persona:var,AA:var,dirección:var)	
Descripción: Responsable de insertar un alumno ayudante pasándole como parámetro los datos especificados.	
actualizar_alumno_ayudante_general(ce_persona:var,persona:var,AA:var,dirección:var)	
Descripción: Responsable de actualizar un alumno ayudante pasándole como parámetro los datos especificados.	
eliminar_alumno_ayudante_general(ci:var)	
Descripción: Responsable de eliminar un alumno ayudante pasándole como parámetro el CI.	
check_update(alumno_Ayudante:var,persona:var,dirección:var)	
Descripción: Responsable de validar que al actualizar los datos de un alumno ayudante los datos sean correctos.	

Tabla3. 32 Descripción de la Clase Controladora: cc_asignatura

Nombre : cc_asignatura	
Tipo de clase: Controladora	
Atributo	Tipo
-----	-----
Responsabilidades	
insertar_asignatura(asignatura:var)	
Descripción: Responsable de insertar una asignatura con los datos especificados.	
check_insert(asignatura:var)	
Descripción: Responsable de validar que al insertar los datos de la asignatura sean correctos.	
buscar_asignaturaXiddisciplina(id_disciplina:var)	
Descripción: Responsable de buscar y devolver los datos de una o varias asignaturas dado el id de la disciplina.	
buscar_X_id(id:var)	
Descripción: Responsable de buscar y devolver los datos de una asignatura dado su id.	
actualizar_asignatura(asignatura:var)	
Descripción: Responsable de actualizar los datos de una asignatura con los parámetros especificados.	

Tabla3. 33 Descripción de la Clase Controladora: cc_departamento

Nombre : cc_departamento	
Tipo de clase: controladora	
Atributo	Tipo
-----	-----
Responsabilidades	
insertar_departamento(departamento:var)	
Descripción: Responsable de insertar un departamento con los datos especificados como parámetros.	
check_insert(departamento:var)	
Descripción: Responsable de validar que al insertar los datos del departamento sean correctos.	
actualizar_departamento(ce_departamento:var,departamento:var)	
Descripción: Responsable de actualizar los datos de un departamento con los parámetros especificados.	
buscar_X_nombre(nombre:var)	
Descripción: Responsable de buscar y devolver los datos de uno o varios departamentos dado el nombre del departamento.	
buscar_dpto(id_dpto:var)	
Descripción: Responsable de buscar y devolver los datos de un departamento dado su id.	
check_búsqueda(nombre:var)	
Descripción: Responsable de validar que los datos de la búsqueda sean correctos.	

Tabla3. 34 Descripción de la Clase Controladora: cc_direccion

Nombre : cc_direccion	
Tipo de clase: Controladora	
Atributo	Tipo
-----	-----
Responsabilidades	
insertar_direccion(direccion:var)	
Descripción: Responsable de insertar una dirección si los datos enviados son correctos.	
check_insert(direccion:var)	
Descripción: Responsable de validar que al insertar los datos de la dirección sean correctos.	
actualizar_direccion(dirección:var)	
Descripción: Responsable de actualizar una dirección si los datos enviados son correctos.	
buscar_direccion(ci:var)	
Descripción: Responsable de buscar y devolver los datos de una direccion dado por parámetro el carnet de identidad.	

Tabla3. 35 Descripción de la Clase Controladora: cc_disciplina

Nombre : cc_disciplina	
Tipo de clase: Controladora	
Atributo	Tipo
-----	-----
Responsabilidades	
insertar_disciplina(disciplina:var)	
Descripción: Responsable de insertar una disciplina si los datos enviados son correctos.	
check_insert(disciplina:var)	

Descripción: Responsable de validar que al insertar los datos de las disciplinas sean correctos. buscarDisciplina_X_iddpto(id_departamento:var)
Descripción: Responsable de buscar y devolver los datos de las disciplinas dado id departamento como parámetro. buscarDisciplina_X_id(id:var)
Descripción: Responsable de buscar y devolver los datos de una disciplina dado su id. actualizar_disciplina(disciplina:var)
Descripción: Responsable de actualizar una disciplina si los datos enviados son correctos.

Tabla3. 36 Descripción de la Clase Controladora: cc_grupos

Nombre : cc_grupos	
Tipo de clase: Controladora	
Atributo	Tipo
-----	-----
Responsabilidades	
insertar_grupo(grupo:var)	
Descripción: Responsable de insertar un grupo si los datos enviados son correctos.	
check_insert(grupo:var)	
Descripción: Responsable de validar que al insertar los datos de los grupos sean correctos.	
actualizar_grupo(grupo:var)	
Descripción: Responsable de actualizar un grupo si los datos enviados son correctos.	
buscar_X_nombre(nombre:var)	
Descripción: Responsable de buscar y devolver los datos de uno o varios grupos dado NombreDisciplina.	
buscar_X_anno(anno:var)	
Descripción: Responsable de buscar y devolver los datos de uno o varios grupos dado el año.	
buscar_X_id(id:var)	
Descripción: Responsable de buscar y devolver los datos de un grupo dado su id.	
eliminar_grupo_general(id_grupo:var)	
Descripción: Responsable de eliminar un grupo.	

Tabla3. 37 Descripción de la Clase Controladora: cc_persona

Nombre : cc_persona	
Tipo de clase: Controladora	
Atributo	Tipo
-----	-----
Responsabilidades	
insertar_persona(persona:var)	
Descripción: Responsable de insertar una persona si los datos enviados son correctos.	
check_búsqueda(nombre:var)	
Descripción: Responsable de validar que los datos para la búsqueda sean correctos.	
actualizar_persona(persona:var)	
Descripción: Responsable de actualizar los datos de una persona si los datos enviados son correctos.	
buscar_persona(nombre:var)	
Descripción: Responsable de buscar y devolver los datos de una o varias personas dado nombre de persona.	
buscar_profesor(ci:var)	
Descripción: Responsable de buscar y devolver los datos de un profesor dado su CI.	

buscar_profesorColaborador(ci:var)
Descripción: Responsable de buscar y devolver los datos de un profesor colaborador dado su CI.
buscar_profesorInterno(ci:var)
Descripción: Responsable de buscar y devolver los datos de un profesor interno dado su CI.
buscar_profesorExterno(ci:var)
Descripción: Responsable de buscar y devolver los datos de un profesor externo dado su CI.
buscar_profesorAA(ci:var)
Descripción: Responsable de buscar y devolver los datos de un alumno ayudante dado su CI.
buscar_persona_X_CI(ci:var)
Descripción: Responsable de buscar y devolver los datos de una persona dado su CI.
profesoresXdepartamento(dpto:var)
Descripción: Responsable de buscar y devolver los datos de uno o varios profesores dado el id del departamento como parámetro.
personas_X_departamento(id_dpto:var)
Descripción: Responsable de buscar y devolver los datos de una o varias personas dado el id del departamento.

Tabla3. 38 Descripción de la Clase Controladora: cc_profesor

Nombre : cc_profesor	
Tipo de clase: Controladora	
Atributo	Tipo
-----	-----
Responsabilidades	
profesores_x_categoriacientifica(categoría:var)	
Descripción: Responsable de buscar y devolver los datos de uno o varios profesores dada la categoría científica.	
profesores_x_categoriadocente(cdocente:var)	
Descripción: Responsable de buscar y devolver los datos de uno o varios profesores dada la categoría docente.	

Tabla3. 39 Descripción de la Clase Controladora: cc_profesor_colaborador

Nombre : cc_direccion	
Tipo de clase: Controladora	
Atributo	Tipo
-----	-----
Responsabilidades	
check_insert(profesor_colaborador:var,persona:var,profesor:var,dirección:var)	
Descripción: Responsable de validar que al insertar los datos del profesor colaborador sean correctos.	
check_update(profesor_colaborador:var,persona:var,profesor:var,dirección:var)	
Descripción: Responsable de validar que al actualizar los datos del profesor colaborador sean correctos.	
insertar_profesor_colaborador_general(persona:var,prof:var,prof_colaborador:var,dirección:var)	
Descripción: Responsable de insertar los datos de un profesor colaborador con los parámetros especificados, si estos son correctos.	
actualizar_profesor_colaborador_general(persona:var,prof:var,prof_colaborador:var,dirección:var)	
Descripción: Responsable de actualizar los datos de un profesor colaborador con los parámetros especificados, si estos son correctos.	
eliminar_profesor_colaborador_general(ci:var)	
Descripción: Responsable de eliminar los datos de un profesor colaborador dado su CI.	

Tabla3. 40 Descripción de la Clase Controladora: cc_profesor_externo

Nombre : cc_profesor_externo	
Tipo de clase: Controladora	
Atributo	Tipo
-----	-----
Responsabilidades	
check_insert (profesor_externo:var,persona:var,profesor:var,dirección:var)	
Descripción: Responsable de validar que al insertar los datos del profesor externo sean correctos.	
check_update (profesor_externo:var,persona:var,profesor:var,dirección:var)	
Descripción: Responsable de validar que al actualizar los datos del profesor externo sean correctos.	
insertar_profesor_externo_general(persona:var,prof:var,prof_externo:var,direccion:var)	
Descripción: Responsable de insertar los datos de un profesor externo con los parámetros especificados, si estos son correctos.	
actualizar_profesor_externo_general(persona:var,prof:var,prof_externo:var,direccion:var)	
Descripción: Responsable de actualizar los datos de un profesor externo con los parámetros especificados, si estos son correctos.	
eliminar_profesor_externo_general(ci)	
Descripción: Responsable de eliminar los datos de un profesor externo dado el CI.	

Tabla3. 41 Descripción de la Clase Controladora: cc_profesorInterno

Nombre : cc_profesorInterno	
Tipo de clase: Controladora	
Atributo	Tipo
-----	-----
Responsabilidades	
check_insert(profesor_interno:var,persona:var,profesor:var,dirección:var)	
Descripción: Responsable de validar que al insertar los datos del profesor interno sean correctos.	
check_update (profesor_interno:var,persona:var,profesor:var,dirección:var)	
Descripción: Responsable de validar que al actualizar los datos del profesor interno sean correctos.	
insertar_profesor_interno_general(persona:var,prof:var,prof_interno:var,direccion:var)	
Descripción: Responsable de insertar los datos de un profesor interno con los parámetros especificados, si estos son correctos.	
actualizar_profesor_interno_general(persona:var,prof:var,prof_interno:var,direccion:var)	
Descripción: Responsable de actualizar los datos de un profesor interno con los parámetros especificados, si estos son correctos.	
eliminar_profesor_interno_general(ci:var)	
Descripción: Responsable de eliminar los datos de un profesor interno dado el carnet de identidad.	

Tabla3. 42 Descripción de la Clase Controladora: cc_planesTrabajo

Nombre : cc_planesTrabajo	
Tipo de clase: Controladora	
Atributo	Tipo
-----	-----
Responsabilidades	
insertar_planDM(planDM:var)	
Descripción: Responsable de insertar una actividad del plan de trabajo docente metodológico si los datos enviados son correctos.	

check_insert(planDM:var)
Descripción: Responsable de validar que al insertar o actualizar los datos de la actividad DM sean correctos.
actualizar_actividad_dm(planDM:var)
Descripción: Responsable de actualizar los datos de una actividad DM del plan de trabajo docente metodológico si los datos enviados son correctos.
eliminar_actividad_dm(id:var)
Descripción: Responsable de eliminar una actividad del plan de trabajo docente metodológico dado el id de la actividad.
buscar_actividadesDM_X_id(id:var)
Descripción: Responsable de buscar y devolver los datos de una actividad del plan de trabajo docente metodológico dado el id de la actividad.
buscar_actividadesDM(id_dpto:var, semestre:var)
Descripción: Responsable de buscar y devolver los datos de una o varias actividades del plan de trabajo docente metodológico dado el id del departamento y el semestre en que se planificaron.
insertar_meses(mes:var)
Descripción: Responsable de insertar los datos de los meses en que se planifica una actividad DM.
buscarmeses_X_idactividad(id_actividad:var)
Descripción: Responsable de buscar y devolver los meses en que se planifica una actividad DM dado el id de la actividad.
eliminarmeses_X_idactividad(id_actividad:var)
Descripción: Responsable de eliminar de la BD los meses en que se planificó una actividad DM dado el id de la actividad.
insertar_plan_actividades(plan_actividades:var)
Descripción: Responsable de insertar una actividad del plan de actividades de un departamento si los datos enviados son correctos.
actualizar_actividad(plan_Actividades:var)
Descripción: Responsable de actualizar los datos de una actividad del plan de actividades de un departamento si los datos enviados son correctos.
eliminar_actividad(id_Actividad:var)
Descripción: Responsable de eliminar los datos de una actividad del plan de actividades de un departamento.
buscar_actividad_X_iddpto(id_dpto;var,mes:var)
Descripción: Responsable de buscar y devolver los datos de una o varias actividades dado el id del departamento y el mes en que se planificaron.
buscar_actividad_X_idactividad(id_actividad:var)
Descripción: Responsable de buscar y devolver los datos de una actividad dado el id de la actividad.
insertar_registro_asistencia(registro:var)
Descripción: Responsable de insertar registros de asistencia en la BD dado el id de la actividad y el CI de la persona.
existe_registro(id_actividad:var,asistente:var)
Descripción: Responsable de comprobar si un registro de asistencia existe en la BD.
registroXidactividad(id_actividad:var)
Descripción: Responsable de buscar y devolver los registros de asistencia de una actividad dado el id de la actividad.
eliminar_registro(id_actividad:var)
Descripción: Responsable de eliminar todos los registros de asistencia de una actividad dado el id de la actividad.
check_insert_planDM(planDM:var)
Descripción: Responsable de validar que al insertar los datos de una actividad DM sean correctos.
check_insert_planActividades(ce_plan_actividades:var,plan_actividades:var)
Descripción: Responsable de validar que al insertar los datos de la actividad sean correctos.

Tabla3. 43 Descripción de la Clase Controladora: cc_postgrados

Nombre : cc_postgrados	
Tipo de clase: Controladora	
Atributo	Tipo
-----	-----
Responsabilidades	
insertar_postgrado(postgrado:var)	
Descripción: Responsable de insertar un postgrado si los datos enviados son correctos.	
insertar_profesor_postgrado(matricula:var)	
Descripción: Responsable de insertar en la BD un nuevo registro de matrícula. Se compone del id del postgrado y el id del profesor que matricula.	
actualizar_postgrado(postgrado:var)	
Descripción: Responsable de actualizar los datos de un postgrado si los datos enviados son correctos.	
buscarpostgrado_X_nombre(nombre:var)	
Descripción: Responsable de buscar y devolver los datos de uno o varios postgrado dado NombrePostgrado.	
buscarpostgrado_X_estado(estado:var)	
Descripción: Responsable de buscar y devolver los datos de uno o varios postgrados dado el estado (iniciado, impartándose, terminado)	
buscarpostgrado_X_idpostgrado(id_postgrado:var)	
Descripción: Responsable de buscar y devolver los datos de un postgrado dado su id.	
check_insert(postgrado:var)	
Descripción: Responsable de validar que al insertar los datos del postgrado sean correctos.	
check_update(ce_postgrado:var,postgrado:var)	
Descripción: Responsable de validar que al actualizar los datos del postgrado sean correctos.	
valida_matricula(id_persona:var,id_postgrado:var)	
Descripción: Responsable de validar que no se repita un registro en la tabla profesor_postgrado.	
valida_profesor(id_postgrado:var)	
Descripción: Responsable de validar que un postgrado no tenga dos profesores asignados.	

Tabla3. 44 Descripción de la Clase Controladora: cc_grupoprofesor

Nombre : cc_grupoprofesor	
Tipo de clase: Controladora	
Atributo	Tipo
-----	-----
Responsabilidades	
insertar_relacion(relación:var)	
Descripción: Responsable de insertar una dirección si los datos enviados son correctos.	
Descripción: Responsable de insertar asignaciones grupo-profesor si los datos enviados son correctos	
validar(id_grupo:var,id_persona:var)	
Descripción: Responsable de comprobar si un registro existe en grupo_profesor.	
buscar_X_ci(ci:var)	
Descripción: Responsable de buscar y devolver las relaciones grupo-profesor dado CI.	
eliminar_relacion(id_grupo:var)	
Descripción: Responsable de eliminar relaciones grupo-profesor dado id_grupo.	
eliminar_relacion1(id_persona:var)	
Descripción: Responsable de eliminar relaciones grupo-profesor dado CI.	
eliminar_relacion2(id_persona:,id_grupo:var)	
Descripción: Responsable de eliminar relaciones grupo-profesor dado id_grupo, CI.	

3.6- Diseño de la Base de Datos:

Una Base de Datos es un conjunto de información de forma correlativa y organizada de la cual posteriormente se puede extraer uno o varios datos necesarios. Uno de los modelos más usados hoy en día es el relacional, con el que se puede, como bien dice su nombre, relacionar unos datos con otros para facilitar el trabajo al programador y también a la hora de realizar una búsqueda. Para la construcción de la base de datos se tuvo en cuenta que fuese de este tipo y se empleó el modelo de datos teniendo en cuenta que constituye la representación física de la misma.

El modelo de datos se encontrará en el **anexo V**.

3.7- Descripción de las tablas de la Base de Datos

Tabla 3. 45 Descripción de la tabla dDisciplina:

Nombre: dDsiciplina		
Descripción: Tabla que almacena los datos de las disciplinas.		
Atributo	Tipo	Descripción:
id_disciplina	integer (0)	Identificador de la tabla.
NombreDisciplina	nvarchar(50)	Nombre de la disciplina que se registra.
Id_Departamento	integer (0)	Identificador de la tabla dDepartamento (Donde se almacenan los datos de los departamentos).

Tabla 3. 46 Descripción de la tabla dAsignatura

Nombre: dAsignatura		
Descripción: Tabla que almacena los datos de las asignaturas.		
Atributo	Tipo	Descripción:
id_asignatura	integer (0)	Identificador de la tabla.
NombreAsignatura	nvarchar(50)	Nombre de la asignatura que se registra.
anno	nvarchar(50)	Anno de la asignatura que se registra.
id_Departamento	integer (0)	Identificador de la tabla dDepartamento (Donde se almacenan los datos de los departamentos).
id_disciplina	integer(0)	Identificador de la tabla dDisciplina (donde se almacenan los datos de las disciplinas)

Tabla3. 47 Descripción de la tabla dDepartamento

Nombre: dDepartamento		
Descripción: Tabla que almacena los datos de los departamentos.		
Atributo	Tipo	Descripción:
id_Departamento	integer (0)	Identificador de la tabla.
Nombre	nvarchar(50)	Nombre del departamento que se registra.

Tabla3. 48 Descripción de la tabla DMeses

Nombre: DMeses		
Descripción: Tabla que almacena los meses de las actividades.		
Atributo	Tipo	Descripción:
nombre_mes	nvarchar(50)	Nombre del mes que se registra.
Id_actividad	integer (0)	Identificador de la tabla PlanTrabajoDM (Donde se almacenan los datos de las actividades del plan del de trabajo docente metodológico).

Tabla3. 49 Descripción de la tabla PlanTrabajoDM

Nombre: PlanTrabajoDM		
Descripción: Tabla que almacena los datos de las actividades DM.		
Atributo	Tipo	Descripción:
id_actividad	integer (0)	Identificador de la tabla.
semestre	nvarchar(50)	Semestre de la actividad DM.
responsable	nvarchar(100)	Responsable de la actividad.
id_Departamento	integer (0)	Identificador de la tabla dDepartamento (Donde se almacenan los datos de los departamentos).
actividad	nvarchar(500)	Descripción de la actividad DM.
aseguramientos	nvarchar(500)	Descripción de los aseguramientos de la actividad DM.
participantes	nvarchar(500)	Participantes en la actividad DM.

Tabla3. 50 Descripción de la tabla dasistencia

Nombre: dasistencia		
Descripción: Tabla que almacena los registros de asistencia a las actividades.		
Atributo	Tipo	Descripción:
id_actividad	integer (0)	Identificador de la tabla.
asistentes	nvarchar(50)	Identificador de la persona.

Tabla3. 51 Descripción de la tabla PlanActividades

Nombre: PlanActividades		
Descripción: Tabla que almacena los datos de las actividades del plan de actividades.		
Atributo	Tipo	Descripción:
id_actividad	integer (0)	Identificador de la tabla.
id_Departamento	integer (0)	Identificador de la tabla dDepartamento (Donde se almacenan los datos de los departamentos).
actividad	nvarchar(500)	Descripción de la actividad.
mes	nvarchar(50)	Mes de la actividad.
lugar	nvarchar(100)	Lugar de la actividad.
fecha_cumplimento	datetime	Fecha de cumplimiento de la actividad.
responsable	Char(15)	Responsable de la actividad.

Tabla3. 52 Descripción de la tabla dDireccion

Nombre: dDireccion		
Descripción: Tabla que almacena los datos de las direcciones.		
Atributo	Tipo	Descripción:
id_direccion	integer (0)	Identificador de la tabla.
CI	char(11)	Identificador de la tabla dPersona(donde se almacenan los datos de todas las personas)
Provincia	nvarchar(40)	Provincia que se registra.
Municipio	nvarchar(40)	Municipio que se registra.
Entre_Calle	nvarchar(40)	Entre calle que se registra.
Numero_Casa	integer (0)	Numero de la casa que se registra.
Reparto	nvarchar(40)	Reparto que se registra.

Edificio	integer (0)	Edificio que se registra.
Apartamento	integer (0)	Apartamento que se registra.
calle	nvarchar(40)	Calle que se registra.

Tabla3. 53 Descripción de la tabla dPersona

Nombre: dPersona		
Descripción: Tabla que almacena todas los datos de las personas.		
Atributo	Tipo	Descripción:
CI	char (11)	Identificador de la tabla dPersona(CI)
id_Departamento	integer (0)	Identificador de la tabla dDepartamento (Donde se almacenan los datos de los departamentos).
id_disciplina	integer(0)	Identificador de la tabla dDisciplina (donde se almacenan los datos de las disciplinas)
id_asignatura	integer(0)	Identificador de la tabla dasignatura (donde se almacenan los datos de las asignaturas)
MilitanciaUJC	nvarchar(10)	Militancia de la UJC.
NombreApellidos	nvarchar(100)	Nombre y apellidos de la persona.
Solapin	nvarchar(40)	Solapin de la persona.
Correo	nvarchar(40)	Correo de la persona.

Tabla3. 54 Descripción de la tabla dgrupoProfesor

Nombre: dgrupoProfesor		
Descripción: Tabla que almacena el id del grupo y el id del profesor (Asignación entre grupo y profesor.)		
Atributo	Tipo	Descripción:
id_grupo	integer (0)	Identificador de la tabla dGrupos
CI	char (11)	Identificador de la tabla dPersona (donde se almacenan los datos de las personas).

Tabla3. 55 Descripción de la tabla dGrupos

Nombre: dGrupos		
Descripción: Tabla que almacena todos los grupos que.		
Atributo	Tipo	Descripción:

id_grupo	integer (0)	Identificador de la tabla.
anno	nvarchar(50)	Anno del grupo.
Nombre_Grupo	nvarchar(50)	Nombre del grupo.

Tabla3. 56 Descripción de la tabla dProfesorPostgrado

Nombre: dProfesorPostgrado		
Descripción: Tabla que almacena el id del postgrado, el id del profesor y el estado (Relación entre postgrado y profesor al realizar la matrícula).		
Atributo	Tipo	Descripción:
id_Postgrado	integer (0)	Identificador de la tabla dPostgrados (donde se almacenan los datos de los postgrados).
CI	char (11)	Identificador de la tabla dPersona (donde se almacenan los datos de las personas).
nestado	nvarchar(20)	Estado de la relación (si imparte o recibe el postgrado).

Tabla3. 57 Descripción de la tabla dProfesor

Nombre: dProfesor		
Descripción: Tabla que almacena los profesores.		
Atributo	Tipo	Descripción:
CI	char (11)	Identificador de la tabla dPersona(donde se almacenan los datos de todas las personas)
Categoriacientifica	nvarchar(50)	Categoría Científica que posee el profesor.
Categoriadocente	varchar(50)	Categoría docente que posee el profesor.
MilitanciaPCC	varchar(10)	Si es militante del PCC o no.

Tabla3. 58 Descripción de la tabla dPostgrados

Nombre: dPostgrados		
Descripción: Tabla que almacenan los datos de los postgrados.		
Atributo	Tipo	Descripción:
id_Postgrado	integer (0)	Identificador de la tabla.
horarios	nvarchar(200)	Descripción del horario del postgrado.
nestado	nvarchar(20)	Estado de los postgrados.
CursoEScolar	nvarchar(10)	Curso escolar en que se realiza el postgrado.
NombrePostgrado	nvarchar(10)	Nombre del postgrado.
Fecha_inicio	DateTime	Fecha de inicio del postgrado.

Fecha_Fin	DateTime	Fecha de fin del postgrado.
Semestre	nvarchar(20)	Semestre en que se realiza el postgrado.

Tabla3. 59 Descripción de la tabla dprofesor_externo

Nombre: dprofesor_externo		
Descripción: Tabla que almacena los profesores externos.		
Atributo	Tipo	Descripción:
CI	char (11)	Identificador de la tabla dPersona(donde se almacenan los datos de todas las personas)
institucion	varchar(50)	Institución a la que pertenece el profesor externo.
Telf_Institucion	Integer(0)	Teléfono de la institución a la que pertenece el profesor externo.

Tabla3. 60 Descripción de la tabla dProfesorColaborador

Nombre: dProfesorColaborador		
Descripción: Tabla que almacena los profesores colaboradores que se registran.		
Atributo	Tipo	Descripción:
CI	char (11)	Identificador de la tabla dPersona(donde se almacenan los datos de todas las personas)
AreaTrabajo	varchar(40)	Área de trabajo del profesor colaborador.

Tabla3. 61 Descripción de la tabla AlumnoAyudante

Nombre: AlumnoAyudante		
Descripción: Tabla que almacena los alumnos ayudantes.		
Atributo	Tipo	Descripción:
CI	char (11)	Identificador de la tabla dPersona(donde se almacenan los datos de todas las personas)
Teléfono	Integer(0)	Teléfono del apartamento.
ProfesorTutor	Nvarchar(50)	Profesor tutor del alumno ayudante.
Apartamento	Integer(0)	Apartamento del alumno ayudante.

Tabla3. 62 Descripción de la tabla dProfesor_Interno

Nombre: dProfesor_Interno		
Descripción: Tabla que almacena los datos de los profesores internos.		
Atributo	Tipo	Descripción:
CI	char (11)	Identificador de la tabla dPersona(donde se almacenan los datos de todas las personas)
Telefono_Apto	Integer(0)	Teléfono del apartamento.
apto	Integer(0)	Apartamento del profesor interno.
LaboratorioProd	Integer(0)	Laboratorio de producción del profesor interno.
telefono_lab	Integer(0)	Teléfono del laboratorio de producción del profesor interno.

Conclusiones

En este capítulo se definió el diseño del sistema propuesto. Se representaron los diagramas de clases del diseño, las descripciones de las clases identificadas, los diagramas de secuencia y el diseño de la base de datos mediante el modelo de datos, con las descripciones de cada una de las tablas.

CAPÍTULO

IV

4. Implementación del sistema propuesto.

Introducción.

En el presente capítulo se explica cómo está estructurado físicamente el sistema implementado mediante los modelos de despliegue y de componentes, con los diagramas correspondientes estructurados por paquetes con sus respectivas descripciones.

4.1- Modelo de despliegue

El modelo de despliegue contiene los nodos que conforman la topología de hardware sobre la cual se ejecuta el sistema software implementado (servicio web). En la siguiente figura se muestra el Diagrama de despliegue del sistema desarrollado. El mismo cuenta con dos nodos: uno de ellos es un servidor "Servicio Web " donde se va a encontrar toda la lógica del negocio, o sea el servicio web como tal y en otro nodo el servidor de la Base de Datos.

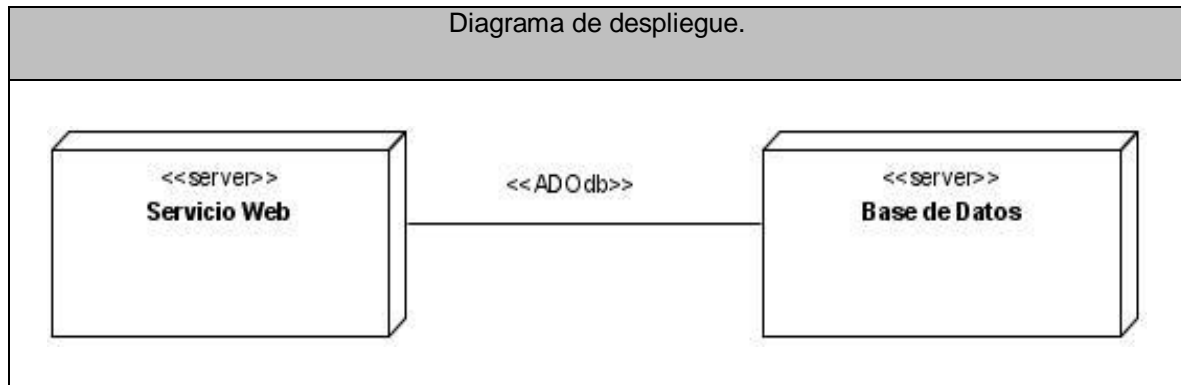


Figura 111 Diagrama de Despliegue

4.2- Modelo Implementación

El modelo de implementación describe cómo se implementan los elementos del modelo de diseño en términos de componentes. Describe también como se organizan y se relacionan unos con otros, definiendo un componente como el empaquetamiento físico de los elementos de un modelo, como es el caso de las clases del modelo de diseño.

4.3- Diagrama de Componentes en Paquetes

Los Diagramas de Componentes representan la separación del software en componentes físicos (archivos, cabeceras, módulos, paquetes, etc.). Se usan para modelar la estructura del software, incluyendo las dependencias entre los componentes de software, los componentes de código binario, y los componentes ejecutables. A continuación se representa el diagrama de componentes estructurado por paquetes correspondientes al sistema implementado.

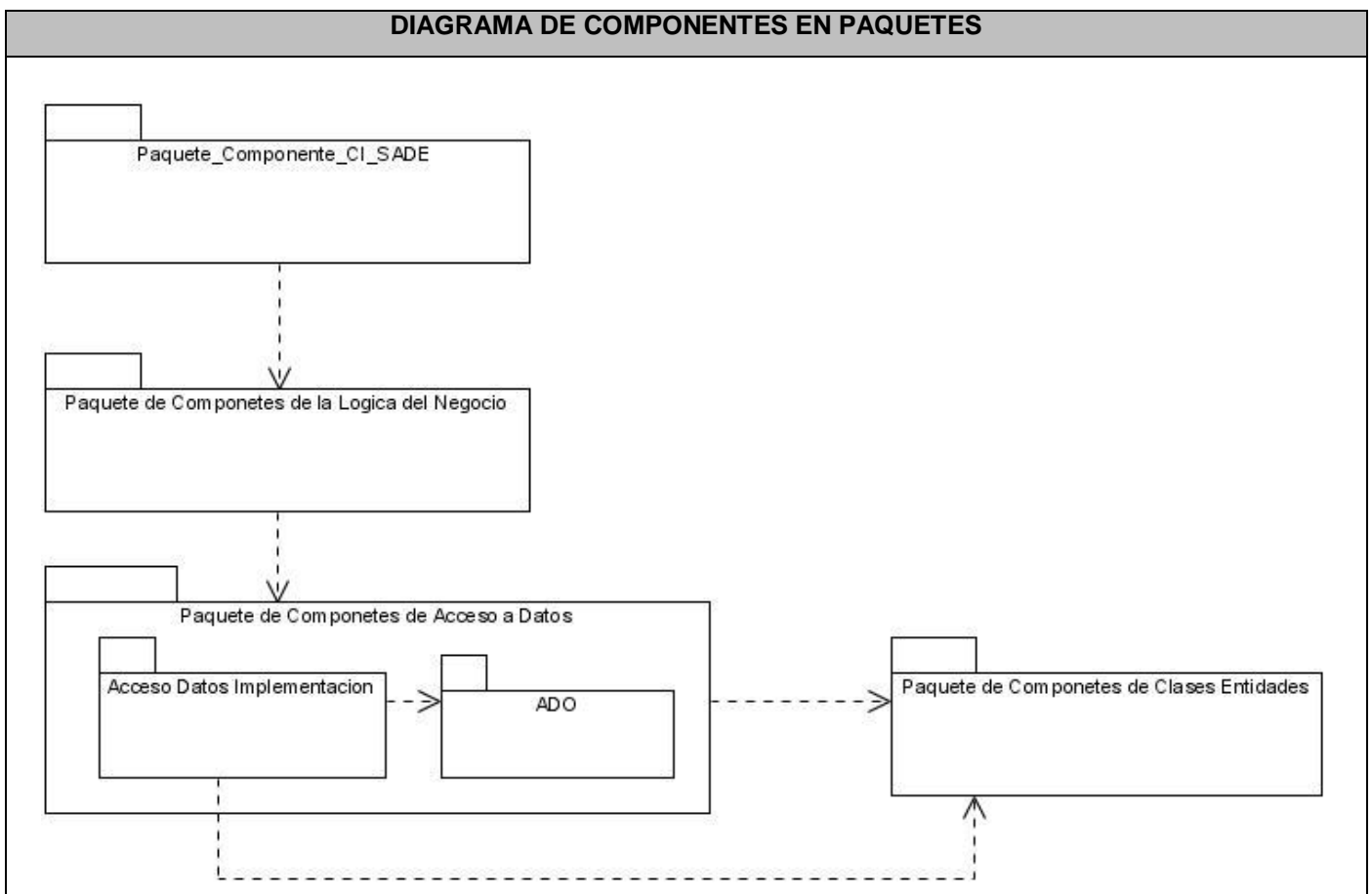


Figura 12 Diagrama de Componentes en Paquetes

4.3.1- Diagramas de Componentes separados por paquetes

Los diagramas de componentes correspondientes a cada uno de los paquetes expuestos anteriormente podrán ser encontrados en el **Anexo VI**. Con el objetivo de lograr una mayor limpieza y rapidez al encontrar las clases, se implementó cada una en un fichero separado, con el mismo nombre, por tanto, cada componente representado en estos diagramas tiene el mismo nombre de la clase que contiene.

4.3.2- Descripción de cada uno de los Paquetes:

- **Paquete_Componente_CI_SADE:** En este paquete se encuentran los componentes que contienen las implementaciones de las clases interfaz del Servicio Web, donde se exponen todas las funcionalidades que brinda el mismo, así como la librería NuSOAP que posibilitó su creación.
- **Paquete de Componentes de clases Entidades:** En este paquete se encuentran los componentes que contienen las implementaciones de las clases que representan las entidades del sistema.
- **Paquete_Componente de las clases controladoras:** En este paquete se encuentran los componentes que contienen las implementaciones de clases controladoras de entidades.
- **Paquete de Acceso a Datos Implementación:** En este paquete se encuentran los componentes que contienen las implementaciones de clases que se ocupan de la persistencia de las entidades en la base de datos.
- **Paquete de Acceso a Datos:** En este paquete se encuentran los componentes que contienen las implementaciones de clases y librerías que permiten conectarnos a diferentes tipos de bases de datos.

Conclusiones.

En este capítulo se explicó cómo está estructurado físicamente el sistema implementado mediante los modelos de despliegue y de componentes con los diagramas correspondientes estructurados por paquetes.

Conclusiones:

1. Con el desarrollo de la presente investigación se demostró la necesidad de desarrollar e implementar un sistema para la gestión de varios de los procesos que se desarrollan en los departamentos docentes de la facultad 1.
2. Los sistemas similares que actualmente existen en Cuba o el mundo no dan soporte ni cumplen con los requisitos que se solicitan.
3. Se implementó un servicio web que brinda un conjunto de funcionalidades que dan solución al problema planteado. Mediante el mismo podrán ser informatizados varios de los procesos de los departamentos docentes de la facultad 1.

Recomendaciones:

- 1- Se recomienda que la presente investigación sea tomada como material de consulta para profesionales que se enfrenten al desarrollo de servicios web.
- 2- Poner el sistema en funcionamiento en los departamentos docentes de la Facultad 1.
- 3- Extender su funcionamiento al resto de las facultades de la universidad.
- 4- Adicionarle nuevas funcionalidades al sistema de manera que pueda ser utilizado no solo en la UCI, sino por cualquier centro de estudios superiores.

Referencias bibliográficas.

[1] World Wide Web. Publicado febrero 2008, última actualización febrero 2008. Disponible en http://es.wikipedia.org/wiki/World_Wide_Web

[2] Software libre. Publicado febrero 2008, última actualización febrero 2008. Disponible en http://es.wikipedia.org/wiki/C%C3%B3digo_libre

[3] Fernández, Ivette. Estudio sobre la unificación de conceptos en Ciencias de la información, Bibliotecología y Archivología. La Habana, 1996. Disponible en http://www.uh.cu/facultades/fcom/portal/interes_glosa_terminos.htm

[4] José María Salinero. Organismo autónomo para el desarrollo local. 2003. Disponible en:
<http://www.oadl.dip-caceres.org/GuiaLUCES/es/Contenidos/Vocabulario.htm>

[5] Conceptos claves acerca de la salud .Prof. Dr. Edgardo A. Marecos.6-10-2001. Disponible en http://med.unne.edu.ar/revista/revista108/con_claves_salud.html

[6] CABALLERO, I. Práctica de Ingeniería de Software 3 [Consultado el: febrero de 2008]. Disponible en:
http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/LabTr1_VP.pdf.

[7] Servicios Web con PHP (NuSOAP) [Consultado el: febrero de 2008]. Disponible en:
<http://www.nociondigital.com/webmasters/php-tutorial-servicios-web-con-php-nusoap-detalle-168.html>.

Bibliografía

1. Proceso Unificado de Rational. publicado el: febrero 2007 de última actualización: febrero 2007. Disponible en: http://es.wikipedia.org/wiki/Rational_Unified_Process.
2. Metodología de Desarrollo de Software (MDS) última actualización: 2005. [Consultado el: febrero de 2008]. Disponible en: <http://www.reynox.com.ar/sap/metodologia.php>.
3. LARMAN, C. UML y Patrones. Introducción al análisis y diseño orientado a objetos. Ciudad de la Habana: 2004. Vol. 1,
4. RUMBAUGH, J. J., I. Y BOOCH, G. El proceso unificado de desarrollo de software. 2000. vol. 1,
5. LARMAN, C. UML y Patrones. Introducción al análisis y diseño orientado a objetos. Ciudad de la Habana: vol. 2,
6. PRESSMAN, R. S. Ingeniería del Software. Un enfoque práctico. Ciudad de la Habana: 2005. Vol.1,
7. Web services con PHP. Última actualización: marzo, 2004. [Consultado el: abril de 2008]. Disponible en: <http://www.desarrolloweb.com/manuales/61/>.
8. Introducción a los Web Services en PHP. publicado el: febrero 2007 de última actualización: febrero 2007. Disponible en: <http://www.desarrolloweb.com/articulos/1852.php>
9. WSDL para la documentación de Servicios Web. vol. 2007, Disponible en: <http://www.desarrolloweb.com/articulos/1581.php>.
10. Lenguajes de Programación para la Web. publicado el: febrero 2008 de última actualización: febrero 2008. Disponible en: <http://www.tejedoresdelweb.com/307/article-1883.html>.

11. Proceso Unificado de Rational. publicado el: febrero 2008 de última actualización: febrero 2008. Disponible en: http://es.wikipedia.org/wiki/Rational_Unified_Process.

12. Módulo Control de Acceso del proyecto Intranet del Centro Rector de Universidad para Todos, Madelyn Saumell Reyes, Miguel Jaeger Rodríguez Lazo, Junio, 2007.