

**Universidad de las Ciencias Informáticas**  
**Facultad 7**



**Título:** Aplicación Web para el Control de la Asistencia a  
Clases en la Facultad 7

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

**Autores:** Ketty Fraga Hernández  
Yanisleidys Alvarez Chamosa

**Tutores:** Ing. Lourdes Escalona Peral  
Ing. Alexander Fonseca Cardosa

**Ciudad de La Habana, Junio 2009**  
**“Año del 50 aniversario del triunfo de la Revolución”**



*La Revolución se lleva en el corazón, no en la boca para vivir de ella.*

*che*

## **DECLARACIÓN DE AUTORÍA**

Se declara que son los únicos autores de este trabajo y se autoriza a la Universidad de las Ciencias Informáticas, los derechos patrimoniales de la misma, con carácter exclusivo.

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

---

Ketty Fraga Hernández

---

Yanisleidys Alvarez  
Chamosa

---

Ing. Lourdes Escalona  
Peral

---

Ing. Alexander Fonseca  
Cardosa

## **DATOS DE CONTACTO**

### **Ing. Lourdes Escalona Peral**

Graduada de Ingeniería Informática curso 2003-2004, Holguín-CUJAE. Ha impartido las asignaturas de Ingeniería de Software 1 y 2, Metodologías de la Investigación Científica y Seminario de Tesis. Fue líder del proyecto Atención Primaria de Salud y del proyecto de Calidad de la Facultad 7.

Por tres años consecutivos desempeña la función de Jefa de Departamento de asignaturas de la especialidad. El trabajo sobre la estrategia para evaluar la Práctica profesional en la Facultad, en el cual fue coautora, obtuvo premio del rector en el año 2007. Presidenta de la comité técnico de la Facultad, así como responsable del comité de tesis por 2 años consecutivos. Resultó cuadro destacado durante el año 2008.

Correo-Electrónico: lescalonap@uci.cu

### **Ing. Alexander Fonseca Cardoso**

Graduado de Ingeniero en Ciencias Informáticas de la Universidad de las Ciencias Informáticas (UCI). Profesor de la Universidad de las Ciencias Informáticas, de la Disciplina de Matemática en la Facultad 7. Cuenta con 1 año de experiencia en la Educación Superior. Jefe de Módulo en el Área Temática de Atención Primaria de Salud (APS) de la Facultad 7.

Correo-Electrónico: afonseca@uci.cu

# Agradecimientos

---

A la Revolución Cubana y a nuestro Comandante Fidel, por proveernos de dicha oportunidad.  
A la Universidad de las Ciencias Informáticas, por la preparación recibida durante estos 5 años.

A nuestra familia, por brindarnos toda su fuerza, amor y apoyo durante toda esta etapa.

A nuestras compañeras de año, las cuales fueron nuestro sostén durante cada etapa difícil.

A nuestra tutora, por siempre estar disponible ante cualquier duda.

A todos los que contribuyeron al éxito de este trabajo de diploma en especial a los compañeros

Baquero y Jax.

Y a los que faltan y deberían estar, mas ahora escapan de nuestra memoria; a todos, gracias.

# Dedicatoria

---

A mi madre, por hacer todo lo que estuvo en sus manos y más para que yo pudiera estudiar en esta escuela. Para ella, quien merece toda la admiración y cariño infinito que brotan de mi corazón.

A mi prima Yeri, que me permitió seguir estudiando cuando mi madre ya no estuvo. A ella, por todo el amor y el cariño que me ha dado y por estar ahí siempre para mí.

A mi padre, mi hermana y a toda mi familia.

**Yanisleidys.**

A mi abuela Argentina, por haber hecho de mí la mujer que soy y por inculcar sus valores y principios.

A mi madre y hermano, por toda su paciencia durante todos estos años.

A Jorge, por todo su apoyo incondicional como padre, durante mi carrera.

A mi amiga Myrlan, por todo su apoyo y gran preocupación durante todo este tiempo.

A Tita, por ser más que una abuela para mí.

A Martucha, por ser madre y amiga al mismo tiempo.

A Jane, por ser mi hermana sin condiciones, siempre estuvo allí cuando más la necesité.

A mi compañera de tesis Yanisleidys, por ser paciente.

A mi pequeña familia, a mis grandes y verdaderos amigos, a todos aquellos de quienes aprendí algo en la vida, a todos aquellos de los que aún me falta por aprender.

A todos los que me permitieron hacer este sueño realidad.

**Ketty.**

## **Resumen**

Cada vez, alcanzan más auge las Aplicaciones de Control Docente en los centros educacionales. Las mismas constituyen un elemento fundamental, donde la principal atención, se centra en el perfeccionamiento de los procesos de generación de reportes e informes, los cuales ayudan a obtener un análisis de la información con exactitud para la toma de decisiones y el progreso de la calidad de la gestión universitaria.

La Gestión Académica en la Facultad 7, tiene como elemento fundamental el control de asistencia a clases. El cual no es satisfactorio, debido a que aún, los directivos de la facultad no obtienen diariamente información referente al estado diario de la misma y en caso de presentarse una situación de urgencia a evaluar, no tienen como tomar una decisión con seguridad. Esto trae consigo, la necesidad de desarrollar un sistema para llevar un control de asistencia a clases, que mejore este mecanismo. Además dicho control aún por parte de muchos profesores, se realiza por el método tradicional: papel y lápiz.

El desarrollo del trabajo tiene como objetivo, el diseño e implementación de una aplicación web, para la gestión de la asistencia a clases, que cumpla con las expectativas del usuario. Además de proveer información organizada y disponible al cliente. Para ello se hace uso de las siguientes tecnologías: herramienta Enterprise Architect, framework Symfony, gestor base de datos PostgreSQL y como herramienta de diseño Dreamweaver.

**Palabras clave:** Control docente, proceso, reporte, informe, gestión académica, asistencia a clases y aplicación web.

# TABLA DE CONTENIDO

---

## Tabla de Contenido

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	5
Sistemas de Gestión Académica.....	5
Antecedentes Internacionales .....	5
XesCampus .....	5
SoftAula.....	6
SIGA.....	6
Akademos.Net .....	6
SIU-Guaraní .....	6
GAUR .....	7
Ágora.....	7
GAUSS.....	7
Antecedentes Nacionales.....	7
SDI .....	7
Antecedentes en la universidad.....	7
GESTACAD.....	7
UCIMAT.....	7
Akademos.....	8
Tecnologías usadas .....	9
Herramienta de Diseño .....	9
Dreamweaver versión 8.0.....	9
Framework.....	11
Framework en PHP. Symfony versión 1.2 .....	11
Framework JavaScript. EXT js versión 3.0 .....	11
Lenguaje de Programación del lado del Cliente .....	12
JavaScript .....	12
Lenguajes de Programación del lado del Servidor .....	13
PHP versión 5.0 .....	13
Arquitectura .....	14
Lenguajes de Marcas.....	15
Lenguaje de Descripción de Servicios Web .....	16
Sistema Gestor de Base de Datos .....	17
PostgreSQL 8.3.....	17
Protocolo Ligero de Acceso a Directorios .....	18



# TABLA DE CONTENIDO

---

Servidores Web .....	19
Servidor HTTP Apache 2.2.....	19
Metodologías de Desarrollo de Software .....	20
Lenguaje de modelado 2.0 .....	20
Herramientas a utilizar en el Proceso de Desarrollo del Software .....	21
Tecnología de la herramienta CASE .....	21
Enterprise Architect 7.1 .....	21
Patrón de arquitectura .....	22
Modelo Vista Controlador .....	22
Conclusiones .....	23
<b>CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA .....</b>	<b>24</b>
Definición del mecanismo de control de asistencia.....	24
Objeto de informatización.....	25
Sistema informatizado existente.....	25
Información que se maneja .....	26
Modelo del Dominio.....	26
Estudiante: Persona dedicada a la lectura, puesta en práctica y aprehensión de conocimientos sobre alguna materia o arte. Que estudia como su ocupación principal.....	27
Diagrama de Clases del Modelo del Dominio .....	28
Reglas del Negocio .....	28
Modelo de Objeto .....	29
Trabajador del Negocio.....	29
Entidad del Negocio.....	29
Especificación de los requisitos de software.....	30
Requerimientos Funcionales.....	30
Requerimientos no Funcionales.....	31
Definición de los casos de uso y actores del sistema .....	34
Conclusiones.....	44
<b>CAPÍTULO 3: Análisis y Diseño del Sistema.....</b>	<b>45</b>
Análisis.....	45
Modelo de análisis .....	45
Diagramas de Clases de Análisis.....	46
Diseño.....	50
Análisis de posibles implementaciones, componentes o módulos ya existentes y que puedan ser reusados .....	51

# TABLA DE CONTENIDO

---

Estrategias de integración .....	52
Modelado mediante estereotipos Web .....	53
Diagramas de clases del diseño .....	54
Diagramas de interacción .....	58
Diagrama de Secuencia.....	59
Descripción de las clases del diseño .....	61
Conclusiones.....	62
<i>Capítulo 4: Implementación .....</i>	<i>63</i>
Descripción de las tablas .....	64
Diagrama de componentes.....	70
Diagrama de despliegue .....	74
Tratamiento de errores .....	76
Seguridad .....	77
Estrategias de codificación. Estándares y estilos a utilizar.....	78
Conclusiones.....	85
<i>CONCLUSIONES .....</i>	<i>87</i>
<i>RECOMENDACIONES.....</i>	<i>88</i>
<i>REFERENCIAS BIBLIOGRÁFICAS.....</i>	<i>89</i>
<i>BIBLIOGRAFÍA.....</i>	<i>91</i>
<i>ANEXOS.....</i>	<i>94</i>
<i>GLOSARIO DE TÉRMINOS .....</i>	<i>99</i>

# INTRODUCCIÓN

---

## **INTRODUCCIÓN**

La universidad cubana actual, ha desbordado sus límites gracias a la voluntad política del estado. La Universidad de las Ciencias Informáticas (UCI) es hoy en día, más que una universidad convencional, un emporio industrial. Es una comunidad digital de más de 14 mil personas que muestra cómo serían los servicios en una ciudad cubana del futuro. En el mundo no existen muchos ejemplos de tal imbricación fructífera entre universidad, industria y comunidad.

La UCI fue creada en el año 2002 en Ciudad de La Habana, cuenta con trece facultades que incluyen tres sedes regionales. Atendidas por 1 600 profesores, lo que constituye una importante fuerza para la expansión de las ciencias informáticas en Cuba. Actualmente, la sede central presenta una matrícula de 11 000 estudiantes procedentes de todas las provincias y municipios de país. Además, tiene tres sedes regionales en Artemisa, Ciego de Ávila y Granma, más dos sucursales para el desarrollo de proyecto en Holguín y Villa Clara. Fuera de la Isla, se activó desde el 2004 una filial universitaria en Caracas, donde un centenar de alumnos cubanos prosiguen sus investigaciones al tiempo que colaboran con programas sociales y productivos en este país.

La producción industrial del centro también es encomiable, ya que logró una gran presencia en mercados de programas informáticos relacionados con software de identidad, salud, educación, y filtraje de contenidos, entre otros. La universidad por su propia característica permite combinar armónicamente el estudio y el trabajo, garantiza que el estudiante aprenda en su entorno sociocultural y laboral. La propia flexibilidad, exige del estudiante esfuerzo y dedicación para que sea capaz de asumir de forma activa su aprendizaje. Esto le permite, la búsqueda de los conocimientos, enfrentar los problemas con independencia y emplear de forma adecuada los recursos personales, intelectuales, afectivos y motivacionales que dispone.

Un prototipo a seguir en cuanto a producción de software en la universidad es la Facultad 7, la cual se vincula en específico al desarrollo de software para la salud. La misma presenta actualmente una matrícula de 1 005 estudiantes, donde 460 de ellos están vinculados a proyectos productivos. La suma notable de estudiantes que presenta dicha facultad proporciona dificultades en el momento de llevar a cabo el control del proceso docente.

Dentro del proceso antes mencionado, el control de la asistencia docente es importante para que el estudiante alcance su potencial necesario en el momento de aprendizaje y preparación en general para el ejercicio futuro de su profesión. Es responsabilidad de cada uno asistir regularmente y con puntualidad a clases, producto a que las ausencias repetidas pueden causar dificultades académicas

# INTRODUCCIÓN

---

y aislamientos sociales. Para evitar los problemas expuestos antes se establece que es obligatorio cumplir con un mínimo de un 80% de asistencia a clases a lo largo del semestre.

En ocasiones los profesores reciben tardíamente la notificación de estudiantes enfermos o estudiantes que por alguna razón personal no pueden asistir a clases. Ciertamente en algunos casos, las justificaciones no llegan a tiempo; por lo tanto para no afectar la evaluación docente del mismo se hace imprescindible que el estudiante notifique que no asistirá ese día a clases. Si el estudiante tiene una condición crónica de salud que afecta su asistencia a la clases, debe contactar al profesor guía o al profesor de dicha asignatura para discutir esa situación particular.

En la Facultad 7, el control de la asistencia docente se realiza por muchos profesores utilizando el método tradicional, papel y lápiz. Para este control, ya sea por problemas de red en los locales donde son impartidas las clases, de la aplicación existente o por la no uniformidad de la misma, optan por la vía más cómoda: el papel.

En aras de llevar un registro satisfactorio de todo el proceso de control de asistencia docente, la facultad utilizó en el curso 2002-2003 el sistema **GESTACAD** un sistema desarrollado en la Universidad de Matanzas, se aplicó en la UCI durante los dos primeros cursos docentes, este de gran utilidad. Este permitía actualizar, mantener la información sobre estudiantes y profesores de la universidad y obtener determinados reportes del mismo. Sin embargo, el sistema no obedecía a la tónica del sistema docente que emplea la UCI, no tenía la posibilidad de contar con más de un Plan de Estudio y no daba un seguimiento en las evaluaciones de una asignatura.

Paralelamente a este, en el curso 2002-2003, comienza a prestar servicios **UCIMAT** hasta el curso 2003-2004, el cual permitía informatizar parte de la gestión académica de la universidad. Además gestionaba la información académica de los estudiantes y la información de los profesores que forman parte de la universidad. También, permitía consultar algunos reportes, pero sólo los que traía por defecto, de lo contrario, había que ir directamente a la base de datos. Posteriormente en el curso 2004-2005 se pasa a utilizar **Akademoss1** que también continúa en utilidad en el posterior curso 2005-2006 y 2006-2007.

El Sistema de Gestión Académica permite la gestión informatizada de los elementos que intervienen en la labor académica de un centro de estudios. También enfrenta los cambios que pudieran surgir adaptándose a las nuevas condiciones y formas de hacer con el menor costo. A partir de este sistema surge una nueva versión que tiene como nombre **Akademoss2** aplicada al curso 2007-2008 y el actual 2008-2009.

# INTRODUCCIÓN

---

Las aplicaciones utilizadas en la UCI, desde su comienzo hasta la actualidad, continúan presentando dificultades. Los reportes no admiten cambios, los estudiantes no pueden ver su estado de asistencia para tener un control propio, no se tiene un seguimiento del estudiante referente a su asistencia y disciplina, la interfaz no es amigable, no se logra un intercambio dinámico entre estudiante-profesor. Esto limita el control de los profesores y directivos en la evolución de los estudiantes. Además no admite registrar la tardanza de un estudiante.

La posibilidad de acceso diferente en la información no existe. Debido a que la estructura organizativa de la facultad, según los cargos asignados a cada profesor no se definen. Con el fin de lograr que tanto directivos como profesores, estén al margen de la situación actual docente dada su responsabilidad; se demuestra que los sistemas antes expuestos no superan las expectativas esperadas.

La situación problemática expuesta anteriormente, posibilitó formular el siguiente **problema a resolver** ¿Cómo facilitar el control de asistencia de los estudiantes de la Facultad 7 por parte de profesores, durante el proceso docente? Se definió como **objeto de la investigación** el proceso de gestión académica en la Universidad de las Ciencias Informáticas y como **campo de acción** el proceso de Control de la Asistencia a Clases en la Facultad 7.

Dirigido a resolver el problema, se determinó como **objetivo de la investigación**: desarrollar una aplicación web, que brinde prestaciones a los profesores de la Facultad 7 para el control de la asistencia a clases de los estudiantes. Para el desarrollo de la misma se hace uso de la tecnología XML Web Service y la Arquitectura Orientada a Servicios (SOA, por sus siglas en inglés) para aplicaciones distribuidas.

Para dar cumplimiento al objetivo expuesto se trazaron las siguientes **tareas de investigación** a desarrollar:

1. Realizar una valoración de los Sistemas de Gestión Académica existentes hasta el momento en el ámbito nacional, así como las tendencias tecnológicas actuales.
2. Describir las herramientas y tecnologías informáticas para la implementación de la Aplicación web para el Registro de Asistencia de la Facultad 7.
3. Validar los requisitos dados por el cliente a partir de la presentación del prototipo no funcional de la aplicación.
4. Modelar una aplicación y tener en cuenta los requisitos expuestos por el cliente.

# INTRODUCCIÓN

---

5. Modelar los Flujos de Trabajo de “Diseño”, “Implementación” y “Prueba” de la Aplicación Web para la Gestión de la Asistencia en la Facultad 7, y seguir el Proceso Unificado de Desarrollo (RUP, por sus siglas en inglés).
6. Implementar la Aplicación Web para la Gestión de la Asistencia en la Facultad 7, y hacer uso de tratamiento de errores o excepciones.
7. Validar la aplicación, y partir de las pruebas de aceptación de los clientes y la revisión por parte del grupo de calidad de la Facultad.

Una vez desarrollada la aplicación, se esperan los siguientes resultados:

- ◆ Facilitar a los profesores el control de la asistencia de los estudiantes a clases.
- ◆ Consultar reportes de acuerdo al cargo correspondiente, tales como:
  - ✓ Informe diario de asistencia a clases.
  - ✓ Cantidad de asistencia dado un estudiante. (%)
  - ✓ Listado de estudiantes con mayores dificultades de asistencia a clases.
  - ✓ Profesores con mayor incidencia de estudiantes ausentes a su clase.
  - ✓ Reporte de análisis realizados por profesores (colectivos pedagógicos, comisiones disciplinarias, profesor docente, entre otros) a los estudiantes con mayores dificultades.
- ◆ Permitir que los estudiantes puedan revisar su estado actual de asistencia a clases.
- ◆ Permitir al estudiante enviar alguna observación en caso de que tuviera que ausentarse a clases.
- ◆ Lograr que la comunicación estudiante – profesor fluya, para que tanto directivos como profesores tengan un control inmediato y puedan realizar valoraciones rápidas y objetivas, de la evolución de un determinado estudiante.
- ◆ Elevar la eficiencia y calidad del proceso de control de asistencia a clases.

Para lograr un trabajo satisfactorio, se ha estructurado la información expuesta en este documento en cuatro capítulos. **Capítulo 1 Fundamentación teórica**, centra al lector en el tema tratado en cuanto a tendencias, técnicas, tecnologías, metodologías y software usados para la solución del problema que se enfrenta y el estado del arte. **Capítulo 2 Características del sistema**, hace una descripción de la situación en que se enmarca el problema creando un entorno conceptual asociado a la información manipulada por el sistema, se define un acuerdo con el cliente sobre las funcionalidades y requerimientos deseados. **Capítulo 3 Diseño del sistema**, enfoca la modelación detallada y la construcción de la estructura de la aplicación. **Capítulo 4 Implementación**, describe cómo se despliegan las funcionalidades de la aplicación y se detalla la solución propuesta al cliente.

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

## **CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA**

### **Sistemas de Gestión Académica**

La Gestión Académica del Proceso Docente Educativo sustentada en las Tecnologías, es entendida como el proceso de toma de decisiones y acciones de carácter descentralizado, usual y múltiple. Esta se realiza por parte del personal docente apoyado en las tecnologías. En los ámbitos tecnológico, metodológico y de formación para la docencia, que se desarrolla a través de colaboración entre sujetos e instituciones, tiene como objetivo asegurar a través de sus funciones: planificación, organización y ejecución del desarrollo exitoso de las actividades propias de los procesos que la conforman.

Sin duda alguna, la Gestión Académica, en un centro educacional, constituye un elemento fundamental donde la principal atención, se centra en el perfeccionamiento de los procesos de generación de reportes e informes. Éstos ayudan a obtener un análisis de la información con exactitud para la toma de decisiones y el perfeccionamiento de la calidad de la gestión universitaria. La UCI en su trayectoria docente, ha hecho utilidad de tres sistemas de gestión académica. Los cuales aún no han superado todas las metas deseadas, se tiene en cuenta los requisitos pertinentes al control docente, y específicamente el control de asistencia a clases.

La Gestión Académica en la Facultad 7, va enfocada al control de asistencia a clases. El gran número de estudiantes involucrados, hacen poco satisfactorio este control. Aún los directivos de la Facultad 7, no pueden obtener diariamente información referente al estado diario de la misma. En caso de presentarse una situación de urgencia a evaluar, no tienen como tomar una decisión con seguridad, por lo que surge la necesidad en la Facultad 7 de desarrollar un sistema para llevar un control de asistencia a clases, que mejore este mecanismo.

Para el desarrollo del sistema antes mencionado, se llevó a cabo un análisis de los antecedentes de sistemas de gestión académica en distintos ámbitos.

### **Antecedentes Internacionales**

**XesCampus:** Aplicación de gestión académica desarrollada por el Departamento de Soluciones Informáticas para la Educación (SIE) de Unixest. Está dividida en varios módulos que abordan procesos característicos de gestión académica como planificación, matrícula, expediente, bolsas o títulos, entre otras. Entre las principales funciones del sistema se tienen: simplificar y organizar los trámites y procesos académicos; apoyar a las secretarías académicas en los procesos académicos; registrar y controlar las pre-matrículas, las matriculas, el pago de aranceles de los estudiantes, las

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

becas, las estadísticas, los convenios de la universidad; emitir resultados del proceso de admisión.

(1)

**SoftAula:** Sistema cliente - servidor destinado a gestionar y seguir la evolución de cada estudiante, todo ello con una gran cantidad de opciones, así como todo referente al profesor asignado a un grupo de estudiantes. Incorpora algunas características automáticas como la sincronización de hora, control de asistencia, programación de tareas, festividades, etc., pero su alto grado de modificación y sus elevadas posibilidades a la hora de personalizarlo lo hacen amoldarse a cualquier tipo de tareas. SoftAula Professional incluye funciones como un generador de horarios, gestión de calendarios, gestión de aulas, estudiantes y grupo docente, seguimiento académico, la gestión de estudiantes, profesores, cursos, gestión de matrículas y gestión contable. (2)

**SIGA:** Sistema Integrado de Gestión Académica (SIGA) que permite como herramienta confiable, la constatación y medición de procesos y acciones que realizan las Universidades. SIGA posibilita la generación de indicadores y reportes de análisis en Docencia de Pregrado; Postgrado, de Actividad Académica, de Investigación y Extensión. Está abierto permanentemente para que cada facultad complete manualmente su información y realice la actualización continua.

También existe la posibilidad de efectuar cargas masivas de información para agilizar el proceso y actualizar los antecedentes de cada Facultad o Instituto. Presenta varios módulos: Módulo General, Módulo Definición de Estudios, Módulo de Estudiantes, Módulo de Opciones de Estudiantes, dentro de este se encuentran, la Gestión de Tutorías, Mensajerías, Asistencias (faltas), Calificaciones (notas), otros módulos incluidos, Módulo de Profesores, Inventario, Ingresos-Gastos, Generador de informes, estadísticos y gráficos, Generador de Horarios y Diplomas. (3)

**Akademos.Net:** Aplicación específicamente diseñada para llevar un control completo de la organización académica de un centro. La aplicación de Akademos.Net incorpora varios módulos de gestión: asignaturas, estudiantes, tutores y tutorías, así como un módulo para la gestión de las actividades de extensión universitaria de cada centro. (4)

**SIU-Guaraní:** Sistema que administra la gestión académica de los estudiantes, desde que ingresan como aspirantes hasta que obtienen el diploma, de manera segura, íntegra y consistente. El sistema provee servicios para estudiantes, docentes, personal administrativo y autoridades del Sistema Universitario Nacional. Provee controles dinámicos que pueden ser ampliados por la unidad académica sin necesidad de modificar el sistema. SIU-Guaraní brinda servicios para autoridades, docentes, estudiantes, entre otros. (5)



# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

**GAUR:** Herramienta básica para la gestión de trámites administrativos y para la consulta de datos que conforman la vida académica del estudiantado. A través de GAUR el estudiante puede matricularse, solicitar el título, consultar sus notas, el expediente académico o la cuenta de correo, conocer los planes de estudio. Es una secretaría virtual donde se puede realizar cualquier trámite o consulta. Todo estudiante dispone de un acceso al servicio, un identificador que se comunica junto al resguardo de su primera matrícula, válido para toda la carrera. (6)

**Ágora:** Sistema de apoyo educativo. Presenta cuatro módulos: Gestión del módulo de matrícula, expediente, actas, y gestión económica. (7)

**GAUSS:** Sistema de Gestión Académica, que permite la gestión integral de la vida académica de los estudiantes. GAUSS es pionero en la utilización de procesos en autoservicio para los estudiantes. Sirve de apoyo a la gestión de todos los procedimientos administrativos relacionados con la vida académica de los estudiantes. Su ingreso en la Universidad hasta la finalización de sus estudios dentro de cualquier ciclo formativo. (8)

## Antecedentes Nacionales

**SDI:** Sistema docente integral, el cual está conformado actualmente por 8 módulos: Secretaría de Facultades, Matrícula, Estadística, Archivo, Histórico, Recuperación de Información, Pago de Estipendios, Planificación Docente y Postgrado. El sistema tiene como función registrar, procesar y crear mecanismos de recuperación de la información relacionada con el proceso docente, tanto de pregrado, como de postgrado, de forma automatizada.

## Antecedentes en la universidad

**GESTACAD:** Sistema de Gestión académica capaz de realizar diversas acciones. Además de brindar numerosos reportes como son: búsqueda de un estudiante, listado de estudiantes por grupo, reportes dinámicos de la información existente, reporte de notas por asignatura y grupo. Tabla con los resultados docentes de un grupo en un semestre, reporte de los resultados académicos de un estudiante en toda su carrera, actas de exámenes de las diferentes asignaturas, registro de características de un grupo de estudiantes, dar baja a un estudiante. (9)

**UCIMAT:** Sistema que permite consultar reportes, sólo los que trae por defecto. De lo contrario hay que ir directamente a la base de datos. También permite búsquedas de estudiantes dado un criterio. UCIMAT gestiona la información académica de los estudiantes universitarios y la información de los profesores que forman parte del proceso educativo. El sistema se concibe en módulos, los cuales se

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

diferencian en cuanto a: actualizaciones de datos y el sitio web, a través del cual se muestran las diversas salidas de la aplicación.

**Akademios:** Sistema capaz de realizar diversas acciones y brindar numerosos reportes como son: Matrícula (Gestión de estudiantes y estructuras), Estudiante (Herramienta que mantiene informado a los estudiantes sobre su desempeño académico). Registro (Control del desarrollo del proceso docente, notas y asistencia), Profesores (Gestión del claustro del centro), Plan de Estudio (Definición de los planes de estudio, Asignaturas, disciplinas, perfiles), Reportes (Diseño y generación de reportes personalizados).

Los sistemas de gestión académica antes mencionados a nivel internacional, nacional y en la universidad, presentan disímiles módulos. Permiten realizar diversas acciones y brindar numerosos reportes de gran utilidad para el centro estudiantil donde sean implantados. La utilización de alguno de ellos no sólo depende de sus cuantiosos módulos y reportes, va muy ligado al tipo de software que sea como en el caso de XesCampus. Este labora, en torno a una base de datos relacional a la que se accede mediante tecnologías estándar de Internet. Es un software propietario.

También depende del tiempo que pueda ser utilizado, como sería el sistema SoftAula. Sistema cliente – servidor, su familia se diseñó para una óptima integración con toda la gama de soluciones Microsoft Office. Muestra sus informes en Microsoft Word y Exporta sus datos a Excel, Access o HTML. Su mayor inconveniente es que cuenta con un límite de 30 días, luego de registrarse gratuitamente. Durante este tiempo se pueden hacer 20 registros por tabla. Entre los requerimientos indispensables para su funcionamiento está la presencia de una memoria RAM de 128 MB y procesador de 600 MHz ó superior.

El sistema SIGA cuenta con varias versiones. Algunas pueden ser usadas pero el centro donde se utiliza sólo puede tener un máximo de 1000 estudiantes matriculados, este se descarga o compra a través de Internet. Entre los requerimientos mínimos del sistema se encuentra, contar con una PC basado en procesador Pentium de 200 Mhz. SIU-Guaraní es un sistema que no se puede adaptar a las necesidades de otras facultades, gran dificultad para una personalización.

**Ágora** bajo una solución LAMP (Linux+Apache+MySQL+PHP) con el cual los profesores compartían documentos con sus alumnos, funciona, pero su desarrollo está parado, no tiene el servicio de Foro, Chat y Evaluación en Línea que tenían originalmente, aunque la estructura del mismo está lista para soportarlo.

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

**SDI** es un sistema que permite el control del proceso docente y se concibe para que funcione en red, posea una interfaz amigable y sea utilizado sobre el ambiente Windows como sistema operativo, el mismo no brinda los reportes que se desean obtener en la facultad y en su proceso de gestión académica no incluye el control docente. Akademos es el sistema que actualmente se utiliza en la universidad pero no tiene una utilidad continua, por parte de los profesores. Esto se debe, a que la interfaz del sistema no es amigable y los reportes del mismo no se puntualizan.

Después del estudio realizado de los sistemas de gestión académica existentes, tanto a nivel internacional como nacional, se definieron, un grupo de metodologías y herramientas para el posterior tratamiento de la aplicación a desarrollar.

## **Tecnologías usadas**

En la actualidad, las Ciencias Informáticas, se convierte en una rama de resultado clave en todas las esferas de la sociedad, desde las instituciones de comercio, hasta las de salud y educación utilizan cada día, más componentes y aplicaciones informáticas que apoyan la toma de decisiones. Los productos y servicios informáticos se hacen cada vez más cotizados, y las aplicaciones de escritorio disminuyen en por ciento en cuanto a las aplicaciones web con la utilización de servicios web. La herramienta a utilizar en la aplicación web a desplegar, para su diseño y desarrollo web profesional es, Dreamweaver.

## **Herramienta de Diseño**

### **Dreamweaver versión 8.0**

Este provee una poderosa combinación de herramientas de visualización de esquemas, funciones para el desarrollo de aplicaciones y soporte para edición de código, permite tanto a desarrolladores como a diseñadores, crear eficientemente aplicaciones y sitios web visualmente atractivos y basados en estándares. Dreamweaver está disponible como un producto individual y es la suite esencial utilizada para diseñar, desarrollar y mantener eficientes experiencias interactivas en línea.

Permite, que las mejores prácticas del desarrollo web estén al alcance de todos. Sean éstos, usuarios trabajando en su primer sitio web, o desarrollando aplicaciones web avanzadas con el Lenguaje Extensible de Marcas (XML según sus siglas en inglés, Extensible Markup Language) y Hojas de Estilo en Cascada (CSS según sus siglas en inglés, Cascading Style Sheets). Las nuevas herramientas visuales incluidas en Dreamweaver 8.0, simplifican la integración de contenido basado en XML, como la provisión de RSS a los sitios web y aplicaciones utilizando un flujo de trabajo basado en técnicas de arrastrar y colocar (drag and drop).

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

Otras ventajas incluyen un mejor sistema de suposición de código XML y Transformaciones del Leguaje de Hojas de Estilo Extensibles (XSLT según sus siglas en inglés, Extensible Stylesheet Lenguaje Transformations). También simplifica gratamente el trabajo con CSS. El nuevo panel unificado de CSS permite una forma más fácil de comprender un archivo CSS y como estilo en cascada se aplica al contenido. Este panel maneja y edita CSS en forma visual, reduce la necesidad de optar por el sistema de prueba y error basado en código.

El esquema de visualización CSS permite a los usuarios ver estructuras complejas basadas en CSS en forma más clara en la superficie de diseño de Dreamweaver. Esta herramienta permite que el trabajo con CSS sea más amigable para los nuevos usuarios y al mismo tiempo, otorgan una gran precisión y control para usuarios avanzados. Además ofrece un flujo de trabajo optimizado que reduce el tiempo requerido por los usuarios para completar tareas comunes.

Las nuevas herramientas de acercamiento y guías incorporadas en Dreamweaver permiten a los usuarios inspeccionar imágenes, tomar control sobre diseños complejos, con la precisión de un píxel y medir fácilmente las distancias entre objetos en las composiciones de diseño. El producto acelera aún más los procesos de desarrollo, a través de una nueva herramienta de colapso de código, para concentrarse en código específico y a través de una nueva barra de herramientas de codificación para acceder funciones comunes a esta materia. (10)

La gran base de este editor sobre otros es su gran poder de ampliación y personalización del mismo, puesto que este programa, tiene rutinas de JavaScript que hace que sea un programa muy fluido, donde programadores y editores web hagan extensiones para su programa y lo ponga a su gusto. Las versiones originales de la aplicación se utilizaban como simples editores WYSIWYG, sin embargo, versiones más recientes soportan otras tecnologías web como CSS, JavaScript y algunos frameworks del lado servidor.

Además es la herramienta de diseño de páginas web más avanzada, tal como se ha afirmado en muchos medios. Aunque sea un experto programador de HTML el usuario que lo maneje, siempre se encontra en este programa razones para utilizarlo, sobre todo en lo que a productividad se refiere. Cumple perfectamente el objetivo de diseñar páginas con aspecto profesional, y soporta gran cantidad de tecnologías, además es fácil de usar:

- ◆ Hojas de estilo y capas.
- ◆ JavaScript para crear efectos e interactividades.
- ◆ Inserción de archivos multimedia.

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

Además es un programa que se puede actualizar con componentes, que fabrica tanto Macromedia como otras compañías, para realizar otras acciones más avanzadas. (11)

En resumen, el programa es realmente satisfactorio, incluso el código generado es de buena calidad. El problema es que al ser tan avanzado, puede resultar un poco difícil su manejo para personas menos experimentadas en el diseño de web.

## **Framework**

Un framework, en el desarrollo del software es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

## **Framework en PHP. Symfony versión 1.2**

Symfony es un enorme conjunto de herramientas y utilidades que minimiza el desarrollo de la aplicación a desarrollar. También ha tomado las mejores ideas de muchos otros frameworks, incorporando ideas propias y el resultado es un framework elegante, estable, productivo y muy bien documentado. Emplea además el tradicional patrón de diseño Modelo Vista Controlador (MVC) para separar las distintas partes que forman una aplicación web. El modelo representa la información con la que trabaja en la aplicación y se encarga de acceder a los datos. (12)

## **Framework JavaScript. EXT js versión 3.0**

Este framework facilita la creación de las interfaces de usuario para la Aplicación Web del Control de Asistencia a Clases en la Facultad 7. El mismo se distribuye bajo licencia Open Source (gratis) y licencia comercial (de pago, pero con soporte y alguna funcionalidad adicional). EXT js simplifica y estandariza el código para todos los navegadores y pone algunos efectos a las páginas, además:

- ◆ Posibilidad de dividir la pantalla en paneles revisables (layouts).
- ◆ Posibilidad de incluir temas.
- ◆ Posibilita la inclusión de controles que soportan validación de datos.
- ◆ Posibilidad de asignar tooltips a los controles.
- ◆ Posibilidad de diseñar menús drop-down y contextuales.
- ◆ Permite agrupar controles en pestañas.

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

- ◆ Rejilla (grid).
- ◆ TreeView.
- ◆ ListView.
- ◆ Combos editables.
- ◆ Listas con Imágenes.
- ◆ Diálogos modales de ventanas (simuladas) y reales (divs).
- ◆ Ventanas.
- ◆ Posibilidad de crear nuevos controles y de extender la funcionalidad de los ya existentes.

Unido a la utilización de estos Framework se trabaja con lenguajes de programación del lado del cliente y del lado del servidor.

## Lenguaje de Programación del lado del Cliente

Los lenguajes de programación del lado del cliente son simplemente incluidos en el código HTML y pueden ser directamente "digeridos" por el navegador y no necesitan un pre tratamiento.

### JavaScript

Este lenguaje permite interactuar con el navegador de manera dinámica y eficaz, proporcionando dinamismo a las páginas web. También es un lenguaje de script multiplataforma orientado a objetos. Está diseñado para un fácil acoplamiento en otros productos y aplicaciones, tales como los navegadores Web. JavaScript puede ser conectado a los objetos de su ámbito para proveer un control programable sobre éstos. (13)

El núcleo de JavaScript puede se extiende para una variedad de propósitos complementándolo con objetos adicionales; por ejemplo:

- ◆ Del lado Cliente extiende el núcleo del lenguaje proporcionando objetos para el control del navegador (Navigator o cualquier Web browser) y su Modelo Objeto Documento [Document Object Model] (DOM por sus siglas en inglés). Por ejemplo, las extensiones del lado del cliente permiten a una aplicación ubicar elementos en un formulario HTML y responder a los eventos de usuario tales como el clic del mouse, entradas del formulario y navegación de páginas.
- ◆ Del lado Servidor extiende el núcleo del lenguaje proporcionando objetos relevantes para la ejecución de JavaScript en un servidor. Por ejemplo, las extensiones del lado del servidor permiten que una aplicación se comuniquen con una base de datos relacional, proporcionar continuidad de la información desde una invocación de la aplicación a otra o efectuar la manipulación de archivos en un servidor. (14)

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

## Lenguajes de Programación del lado del Servidor

Los lenguajes de lado servidor son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él.  
(15)

### PHP versión 5.0

Este lenguaje de programación del lado del servidor es gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas, a pesar de ser un programa establecido por la arquitectura de la facultad, se considera por sus características una buena opción para trabajar y desarrollar la aplicación. Además de permitir la conexión con el servidor de bases de datos PostgreSQL, a utilizar en el desarrollo de la aplicación.

### Ventajas:

- ◆ Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- ◆ Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- ◆ Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- ◆ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- ◆ Permite las técnicas de Programación Orientada a Objetos
- ◆ Biblioteca nativa de funciones sumamente amplia e incluida.
- ◆ No requiere definición de tipos de variables.
- ◆ Tiene manejo de excepciones (desde PHP5).

A la hora de emplear cualquier técnica de programación o desarrollo que permita escribir código ordenado, estructurado y manejable en la aplicación a desarrollar. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño, Modelo de Vista Controlador (MVC), se permite separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

La premisa de los servicios Web (como por ejemplo SOAP, XML y WSDL) forma el siguiente paso lógico en la evolución de la informática distribuida. Estos estándares siguen un ritmo acelerado durante los próximos años.

## **Arquitectura basada en componentes**

La Arquitectura Orientada a Servicios (SOA Service Oriented Architecture, por sus siglas en inglés), es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requisitos del negocio que permiten, la creación de la Aplicación para el Control de Asistencia a Clases en la Facultad 7, la cual a su vez brinda una forma estándar de exposición e invocación de servicios (comúnmente pero no exclusivamente servicios web), lo cual facilita la interacción entre diferentes sistemas propios o de terceros.

SOA define las siguientes capas de software:

**Aplicaciones básicas** - Sistemas desarrollados bajo cualquier arquitectura o tecnología, geográficamente dispersos y bajo cualquier figura de propiedad.

**De exposición de funcionalidades** - Donde las funcionalidades de la capa aplicativas son expuestas en forma de servicios (Servicios Web).

**De integración de servicios** - Facilitan el intercambio de datos entre elementos de la capa aplicativa orientada a procesos empresariales internos o en colaboración.

**De composición de procesos** - Que define el proceso en términos del negocio y sus necesidades, y que varía en función del negocio.

**De entrega** - donde los servicios son desplegados a los usuarios finales.

SOA proporciona una metodología y un marco de trabajo para documentar las capacidades de negocio y puede dar soporte a las actividades de integración y consolidación. (16)

## **Diferencias con otras arquitecturas**

Al contrario de las Arquitecturas Orientada a Objetos, las SOA's están formadas por servicios de aplicación débilmente acoplados y altamente interoperables. Para comunicarse entre sí, estos servicios se basan en una definición formal independiente de la plataforma subyacente y del lenguaje de programación. La definición de la interfaz encapsula (oculta) las particularidades de una implementación, lo que la hace independiente del fabricante, del lenguaje de programación o de la tecnología de desarrollo. Con esta arquitectura, se pretende que los componentes software desarrollados sean muy reusables, ya que la interfaz se rige por un estándar.

## **Beneficios:**



# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

Los beneficios que puede obtener una organización que adopte SOA son:

- ◆ Mejora en los tiempos de realización de cambios en procesos.
- ◆ Facilidad para evolucionar a modelos de negocios basados en tercerización.
- ◆ Facilidad para abordar modelos de negocios basados en colaboración con otros entes (socios, proveedores).
- ◆ Poder para reemplazar elementos de la capa aplicativa SOA sin problema en el proceso de negocio.
- ◆ Facilidad para la integración de tecnologías disímiles.

## Lenguajes de Marcas

Es una forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación. El lenguaje de marcas es el HTML, fundamento del WWW ( World Wide Web por sus siglas en ingles). Los lenguajes de marcas, se suelen diferenciar entre tres clases de lenguajes de marcado, aunque en la práctica pueden combinarse varias clases en un mismo documento, estas son: el marcado de presentación, el marcado de procedimientos y el marcado descriptivo o semántico.

XML es un Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. Se utiliza en su versión 2.0.

## Principales características:

- ◆ Es una arquitectura más abierta y extensible. No se necesita versiones para que puedan funcionar en futuros navegadores. Los identificadores pueden crearse de manera simple y ser adaptados en el acto en internet/intranet por medio de un validador de documentos (parser).
- ◆ Mayor consistencia, homogeneidad y amplitud de los identificadores descriptivos del documento con XML, en comparación a los atributos de la etiqueta <META> del HTML.
- ◆ Integración de los datos de las fuentes más dispares. Se puede hacer el intercambio de documentos entre las aplicaciones tanto en el propio PC como en una red local o extensa.
- ◆ Datos compuestos de múltiples aplicaciones. La extensibilidad y flexibilidad de este lenguaje permite agrupar una variedad amplia de aplicaciones, desde páginas web hasta bases de datos.
- ◆ Gestión y manipulación de los datos desde el propio cliente web.
- ◆ Los motores de búsqueda devuelven respuestas más adecuadas y precisas, ya que la codificación del contenido web en XML consigue que la estructura de la información resulte

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

más accesible.

- ◆ Se desarrollan de manera extensible las búsquedas personalizables y subjetivas para robot y agentes inteligentes. También conlleva a que los clientes web puedan ser más autónomos para desarrollar tareas que actualmente se ejecutan en el servidor.
- ◆ Se permite un comportamiento más estable y actualizable de las aplicaciones web, donde se incluyen enlaces bidireccionales y almacenados de forma externa (El famoso epígrafe "404 file not found" desaparece).
- ◆ El concepto de "hipertexto" se desarrolla ampliamente (permite denominación independiente de la ubicación, enlaces bidireccionales, enlaces que pueden especificarse y gestionarse desde fuera del documento, hiperenlaces múltiples, enlaces agrupados, atributos para los enlaces, etc. Creado a través del Lenguaje de enlaces extensible.
- ◆ Exportabilidad a otros formatos de publicación (papel, web, cd-rom). El documento maestro de la edición electrónica podría ser un documento XML que se integraría en el formato deseado de manera directa.

XSL es un acrónimo en inglés de Extensible Stylesheet Language (Lenguaje de hojas de estilo ampliable). Es un lenguaje diseñado para presentar datos XML en un formato legible. (17)

## **Lenguaje de Descripción de Servicios Web. WSDL**

Uno de los usos principales del XML es sin duda la creación de WSDL (Web Services Description Language por sus siglas en inglés). Los WSDL describen la interfaz pública a los servicios web en la aplicación. Están basados en XML y describen 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 ligán después al protocolo concreto de red y al formato del mensaje.

Permiten la descripción de los servicios web desplegados en la aplicación a desarrollar. Utiliza también para la localización y ubicación de estos servicios. Un documento WSDL no es más que un documento XML que describe ciertas características propias de un servicio web, así como su localización y aquellos parámetros y métodos que soporta. Un documento WSDL define un servicio web utilizando a tal fin elementos XML, como: (18)

- ◆ <portType> para las operaciones que proporciona el servicio web
- ◆ <message> para los mensajes que utiliza por el servicio web
- ◆ <types> para los tipos de datos que utiliza el servicio web
- ◆ <binding> para los protocolos de comunicaciones que utiliza el servicio web

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

## Sistema Gestor de Base de Datos (SGBD)

Los sistemas de gestión de base de datos (SGBD, Database management system, por sus siglas en ingles) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. El propósito general de los sistemas de gestión de base de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convierten en información relevante, para un buen manejo de datos.

Existen distintos objetivos que deben cumplir los SGBD:

- ◆ Abstracción de la información
- ◆ Independencia
- ◆ Consistencia
- ◆ Seguridad
- ◆ Integridad
- ◆ Respaldo
- ◆ Control de la concurrencia
- ◆ Manejo de Transacciones
- ◆ Tiempo de respuesta. (19)

## PostgreSQL 8.3

Este es un servidor de base de datos relacional orientada a objetos de software libre publicado bajo la licencia BSD, el cual se utiliza en la aplicación a desarrollar. Como muchos otros proyectos open source, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PostgreSQL Global Development Group (PGDG por sus siglas en inglés).

### Características:

#### ◆ Alta concurrencia.

Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos y cada usuario obtiene una visión consistente de lo último a lo que se le hizo commit (cometer). Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

#### ◆ Amplia variedad de tipos nativos.

PostgreSQL provee nativamente soporte para:

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

- ◆ Números depresión arbitraria.
- ◆ Texto de largo ilimitado.
- ◆ Figuras geométricas (con una variedad de funciones asociadas)
- ◆ Direcciones IP
- ◆ Bloques de direcciones estilo CIDR.
- ◆ Direcciones MAC.
- ◆ Arreglos.

Adicionalmente los usuarios pueden crear sus propios tipos de datos. PostgreSQL soporta funciones que retornan "filas", donde la salida puede tratarse como un conjunto de valores que pueden ser tratados igual a una fila retornada por una consulta. Las funciones pueden ser definidas para ejecutarse con los derechos del usuario ejecutor o con los derechos de un usuario que previamente se define. El concepto de funciones, en otros DBMS, son muchas veces referidas como "procedimientos almacenados", incluso provee las mismas funcionalidades y más a los usuarios del servidor de base de datos PostgreSQL.

## Protocolo Ligero de Acceso a Directorios

Protocolo Ligero de Acceso a Directorios (LDAP, Lightweight Directory Access Protocol, por sus siglas en inglés), es un protocolo que se utiliza a nivel de aplicación, este permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. LDAP también es considerado una base de datos (aunque su sistema de almacenamiento puede ser diferente) a la que pueden realizarse consultas.

Habitualmente, este almacena la información de autenticar (usuario y contraseña) y es utilizado para autenticarse aunque es posible almacenar otra información (datos de contacto del usuario, ubicación de diversos recursos de la red, permisos, certificados, etc).

En conclusión, LDAP es un protocolo de acceso unificado a un conjunto de información sobre una red. (20)

El LDAP se define en 4 modelos: (Ver Tabla 1.)

Modelo	Descripción
Información	Describe la estructura de la información almacenada en un directorio LDAP
De nombres	Describe cómo la información en un directorio LDAP está

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

	organizado e identificado
Funcional	Describe lo que las operaciones se pueden realizar sobre la información almacenada en un directorio LDAP
Seguridad	Describe cómo la información en un directorio LDAP puede ser protegida frente a accesos no autorizados

**Tabla 1. Modelo LDAP. Descripción.**

## **Características:**

- ◆ Soporta la Capa de autenticación y seguridad, la Seguridad de la capa de transporte y la Capa de conexión segura, entre otras mejoras. Muchos de los cambios en el protocolo son diseñados para hacer LDAP más seguro.
- ◆ Soporta la próxima generación del protocolo de Internet versión 6.
- ◆ Puede comunicar dentro de un sistema usando comunicación interproceso. Esto mejora la seguridad al eliminar la necesidad de comunicarse a través de la red.
- ◆ Mejora la forma en que los programadores se conectan para usar servidores de directorio LDAP.
- ◆ Provee compatibilidad completa con el formato de intercambio de datos, Data Interchange Format (LDIF) versión 1.
- ◆ Incluye un sistema de control de acceso actualizado, conjunto de hilos y herramientas mejoradas.
- ◆ Admite TCP/IP que es necesario para cualquier tipo de acceso a Internet.

LDAP es un protocolo abierto, las aplicaciones son independientes de plataforma de servidor, que alberga el directorio. (21)

## **Servidores Web**

Un servidor Web, es un programa que corre sobre el servidor que escucha las peticiones HTTP que le llegan y las satisface. De acuerdo al tipo de la petición, el servidor Web busca una página Web o bien ejecuta un programa en el servidor. De cualquier modo, siempre devuelve algún tipo de resultado HTML al cliente o navegador que realizó la petición. El mismo es fundamental en el desarrollo de las aplicaciones del lado del servidor, que se construyen, ya que se ejecutan en él.

## **Servidor HTTP Apache 2.2**

Este es un software libre, servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux), Windows, Macintosh y otras, que implementa el protocolo HTTP y la noción de sitio virtual. El

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

servidor Apache se desarrolla dentro del proyecto HTTP Server, que se utiliza. Apache tiene una amplia aceptación en la red, es el servidor HTTP más usado. Alcanzó su máxima cuota de mercado en 2005, empleado en el 70% de los sitios web en el mundo.

## **Ventajas:**

- ◆ Modular
- ◆ Open source
- ◆ Multi-plataforma
- ◆ Extensible
- ◆ Popular

## **Metodologías de Desarrollo de Software**

Una Metodología de Desarrollo de Software es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software.

Una metodología puede seguir uno o varios modelos de ciclo de vida, indicando en cada ciclo de vida qué es lo que hay que obtener a lo largo del desarrollo del proyecto, pero no cómo hacerlo, en otras palabras, la metodología indica cómo hay que obtener los distintos productos parciales y finales en el desarrollo de un software. La metodología que se utiliza para el desarrollo de la aplicación es RUP, que tiene como lenguaje de modelado UML.

## **Lenguaje de modelado 2.0**

Este Lenguaje Unificado de Modelado (UML, Unified Modeling Language por sus siglas en inglés), es el sistema de software que más se conoce y utiliza en la actualidad; está respaldado por el Object Management Group (OMG por sus siglas en inglés). Es el lenguaje gráfico que se utiliza para visualizar, especificar, construir y documentar la aplicación para controlar el proceso docente en la Facultad 7. Ofrece un estándar para describir un "plano" del sistema (modelo), incluye aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables, se utiliza en su versión 2.0.

Es importante resaltar que UML es un "lenguaje" para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional o RUP), pero no especifica en sí mismo qué

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

metodología o proceso usar. Por todo lo anterior expuesto es la herramienta a utilizar, a pesar de ser la designada por la facultad.

## Herramientas a utilizar en el Proceso de Desarrollo del Software

### Tecnología de la herramienta CASE

La tecnología CASE supone la automatización del desarrollo del software, contribuye a mejorar la calidad y la productividad en el desarrollo de sistemas de información. Para mejorar la calidad y la productividad de los sistemas de información a la hora de construir software se plantean los siguientes objetivos: (22)

- ◆ Permitir la aplicación práctica de metodologías estructuradas, las cuales al ser realizadas con una herramienta se consigue agilizar el trabajo.
- ◆ Facilitar la realización de prototipos y el desarrollo conjunto de aplicaciones.
- ◆ Simplificar el mantenimiento de los programas.
- ◆ Mejorar y estandarizar la documentación.
- ◆ Aumentar la portabilidad de las aplicaciones.
- ◆ Facilitar la reutilización de componentes software.
- ◆ Permitir un desarrollo y un refinamiento visual de las aplicaciones, mediante la utilización de gráficos.

### Enterprise Architect 7.1

Enterprise Architect, por sus siglas en inglés, es una herramienta flexible, completa y potente de modelado en UML bajo plataforma Windows. Provee lo más nuevo en desarrollo de sistemas, administración de proyectos y análisis de negocio. EA es una herramienta que abarca integralmente el ciclo de vida, cubre el desarrollo de software desde el relevamiento de los requerimientos, a través de las etapas de análisis, modelos de diseño, testing y finalmente el mantenimiento y re-uso.

Es utilizada para el desarrollo de varios tipos de software para un amplio rango de industrias, incluyen: bancos, desarrollo web, ingeniería, finanzas, medicina, investigación, educación, transporte, ventas, energía, ingeniería electrónica y muchas más. También es utilizado con efectividad para el entrenamiento en UML y arquitecturas de negocio en empresas de entrenamiento y universidades alrededor del mundo.

Enterprise Architect en su versión 7.0 se construye en base, al excepcional éxito de las versiones previas con un completo soporte para el estándar UML 2.0 como lo define el Grupo de Gestión de Objetos (OMG por sus siglas en inglés Object Management Group). Con EA, los modeladores tienen

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

todo el poder y la expresividad. EA ayuda a administrar la complejidad con herramientas para rastrear las dependencias, soporte para modelos muy grandes, control de versiones con proveedores CVS o SCC, Líneas Base por cada punto del tiempo, la utilidad de comparar, para seguir los cambios del modelo, interfaz intuitiva y de alto rendimiento con vista de proyecto como un "explorador". (23)

## Patrón de arquitectura

### Modelo Vista Controlador

El MVC es un patrón de arquitectura de software que separa los datos de la aplicación a desarrollar, la interfaz de usuario y la lógica de control en tres componentes distintos. Este patrón provee aplicaciones con un modelo de SGBD y la Lógica de Negocio para controlar y recibir los eventos de entrada desde la vista.

Descripción del patrón:

- ◆ **Modelo:** Esta es la representación específica de la información con la cual la aplicación de control de asistencia a clases opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos; por ejemplo, no permite comprar un número de unidades negativo, calculando si hoy es el cumpleaños del usuario o los totales, impuestos o importes en un carrito de la compra.
- ◆ **Vista:** Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- ◆ **Controlador:** Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista. (24)

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que sigue el control generalmente es el siguiente:

- ◆ El profesor y el estudiante interactuarán con la interfaz de usuario de alguna forma.
- ◆ El profesor recibe (por parte de los objetos de la interfaz-vista) la notificación de la inasistencia a clases del estudiante. El directivo gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler o callback).
- ◆ El profesor accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada.
- ◆ El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo. El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, el patrón de observador puede ser utilizado para proveer cierta indirección entre el modelo y la vista, permite al modelo notificar a los interesados de cualquier cambio. Un



# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. El controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice.

- ✓ Nota: En algunas implementaciones la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista.
- ◆ La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

## Conclusiones

En este capítulo, se hizo un estudio de los sistemas existentes de gestión académica a nivel internacional, nacional y en la universidad. Este estudio no aportó mucho al desarrollo del trabajo de diploma. Pues los sistemas, no gestionan la asistencia a clases y son software propietarios. Además de explicar detalladamente todas las tecnologías usadas que fueron definidas, por facultad.

## **CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA**

---

### **CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA**

Los sistemas de gestión de información son un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de un determinado centro. El objetivo de dichos sistemas se basa en que puedan operar de acuerdo a las necesidades impuestas por el centro al cual se establece. Las actividades básicas a realizar son: entrada, almacenamiento, procesamiento y salida de información. Con estas, se realiza una estructura de soporte para la creación y administración de contenidos por parte de los participantes principalmente en páginas web. Además permiten manejar de manera independiente el contenido.

La gestión de la información implica un proceso permanente e integral de planificación, investigación, acción y evaluación. Los temas que se abordan a continuación hacen referencia al objeto de estudio dada la descripción del mecanismo del negocio: control de asistencia docente. Con la realización del modelo del dominio, la especificación de los requisitos y definición de los casos de uso, se forman los cimientos de la Aplicación Web para el Control de la Asistencia a Clases en la Facultad 7.

#### **Definición del mecanismo de control de asistencia**

En el proceso docente educativo a lo largo de la historia ha quedado demostrado que el profesor es el protagonista principal de los cambios que hoy en día la sociedad necesita y exige de las universidades, para lograr la formación de un hombre integral, independiente, original, capaz de utilizar su potencial creador en función de su desarrollo individual y de la sociedad.

La UCI es una universidad que combina el estudio con el trabajo y a la vez prepara a los estudiantes para su futuro entorno laboral. He aquí la importancia de llevar un control eficiente de la asistencia docente. Este mecanismo de control de asistencia a clases, es soportado actualmente por el sistema de gestión académica, Akademos en su última versión 2.0. El sistema anterior, no es utilizado regularmente, esto se debe, a que el control de asistencia a clases es realizado por los profesores mediante el método tradicional.

Esto demuestra que los recursos tecnológicos expuestos por el centro, no satisfacen aún las necesidades fundamentales en el control de asistencia a clases, el cual es de vital importancia para que el estudiante se autoevalúe de acuerdo a su estado actual, además de brindar a directivos, profesores, entre otros cargos dado el nivel estructural de la facultad, la posibilidad de llevar un seguimiento profundo e inmediato referente a los estudiantes y así emitir un criterio objetivo respecto a él.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

---

El control de asistencia a clases es llevado a cabo de forma manual por la mayoría de los profesores de la Facultad 7, esto se debe a los problemas que presenta la aplicación Akademos referente a la interfaz no amigable y a la no uniformidad de la misma. Tanto los directivos, como profesores de la universidad pueden realizar notificaciones de la situación actual que presenta un estudiante o grupo determinado, tomando varios criterios de búsqueda que brinda el sistema de gestión académica presente en el centro.

Precisamente el control de asistencia a clases en el momento final de generar reportes a nivel de facultad, no es lo más objetivo, puesto que, los mismos no brindan a los directivos un registro completo a nivel de facultad, como por ejemplo: si el vicedecano de formación docente necesita el reporte, de un grupo en específico, la aplicación no se lo brinda, porque los reportes que se emiten son a nivel de años; también trae inconvenientes cuando se registra la asistencia, ya que permite introducir más de dos registros de un grupo con la misma fecha en menos de 90 minutos, tiempo de duración de un turno de clases, además no brinda la posibilidad de registrar la tardanza.

### **Objeto de informatización**

El control de asistencia a clases en la Facultad 7, es el mecanismo a informatizar, con la finalidad de que cada profesor de acuerdo al cargo que le corresponde, según la estructura organizativa de la facultad 7, tenga acceso a reportes actualizados. Además de permitir que los estudiantes puedan consultar su estado actual de asistencia y en caso de presentar algún problema en particular, poder notificar a sus profesores la ausencia, así se obtiene la interacción deseada actualmente entre alumno-profesor de manera fluida y rápida.

### **Sistema informatizado existente**

Akademos, sistema de gestión académica utilizado por la universidad para el control docente y dentro de este el control de la asistencia a clases. Este sistema es capaz de realizar diversas acciones y brindar numerosos reportes entre los que se encuentran:

- ◆ Matrícula
- ◆ Estudiante
- ◆ Registro
- ◆ Profesores
- ◆ Plan de Estudio
- ◆ Reportes

Akademos es un acercamiento a la solución que necesita la facultad, pero el mismo no cumple con las expectativas esperadas expuestas anteriormente en el mecanismo a informatizar.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

---

### Información que se maneja

**Matrícula:** Lleva a cabo la gestión de la información de los estudiantes referente al proceso de ingreso y continuidad de matrícula.

**Estudiante:** Mantiene informado a los estudiantes sobre su desempeño académico, pues cuenta con toda la información referente a las notas por asignatura de cada año cursado. Además de que brinda la posibilidad de poder consultar sus datos de matrícula y posibilita la búsqueda de otros estudiantes.

**Registro:** Establece un control en el desarrollo del proceso docente, notas y asistencia. Permite al profesor gestionar la información sobre la situación académica de sus estudiantes.

**Profesores:** Cuenta con toda la información para la gestión del claustro del centro.

**Plan de Estudio:** Definición de los planes de estudio. Asignaturas, disciplinas, perfiles.

**Reportes:** Diseño y generación de reportes personalizados.

### Modelo del Dominio

Un modelo del dominio no representa objetos de software; particiona y presenta los conceptos importantes relacionados con el dominio o realidad física. Los modelos de dominio pueden utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema. Es utilizado por el analista como un medio para comprender el negocio al cual el sistema va a servir. Con él se puede capturar un mayor grado de detalles, pero su objetivo es capturar lo necesario para comprender donde va a funcionar el sistema que se está diseñando.

El modelo del dominio que se representa, describe cómo se desarrolla el Mecanismo de control de asistencia docente en la Facultad 7. Este mecanismo no es más, que la Técnica de funcionamiento que es ejercida por cada profesor para controlar la asistencia de sus estudiantes y la determinación de las causas de inasistencia a clases. Tiene como finalidad, lograr una buena formación docente en los estudiantes. Los parámetros del mismo, son informados al iniciar su curso escolar. (Ver Fig. 1)

**Reporte:** Conjunto de datos en relación a la representación de un contenido en un formato determinado. En él se encuentran:

**Informe diario de asistencia:** Descripción escrita que permite caracterizar el estado actual de la asistencia de la facultad. Entre ellos se encuentran:

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

---

- **Informe diario de asistencia total:** Descripción escrita que permite caracterizar el estado actual de la asistencia en el día de la facultad.
- **Informe diario de asistencia parcial:** Descripción escrita que permite caracterizar el estado actual de la asistencia en el día, respecto a un año, grupo o estudiante.

**Registro:** Documento que presenta resultados que se obtienen o proporcionan evidencia de actividades desempeñadas. Establece un control en el desarrollo del proceso docente, en cuanto a la asistencia a clases. Permite al profesor obtener la información sobre la situación académica de sus estudiantes.

**Estudiante:** Persona dedicada a la lectura, puesta en práctica y aprehensión de conocimientos sobre alguna materia o arte. Que estudia como su ocupación principal.

**Profesor:** Persona que se dedica a la enseñanza y máximo responsable de tomar la asistencia a los estudiantes y dar seguimiento a los mismos.

**Facultad:** Entidad que forma parte de una universidad y está conformada por estudiantes y profesores.

**Cargo:** Responsabilidad asignada a los profesores para lograr un mejor funcionamiento de la facultad. Entre los que se encuentran:

- **Directivo:** Miembro de la dirección de la facultad.
- **Profesor guía:** Persona que se dedica a la enseñanza y lleva el control y la responsabilidad de un grupo de estudiantes en la facultad.
- **Decano:** Persona que preside una facultad universitaria, aunque no sea el miembro más antiguo de ella.
- **Vicedecano de Formación Docente:** Persona que preside de segunda autoridad en una universidad o facultad en la rama de la docencia.
- **Jefe de Departamento:** Persona encargada de llevar el control de todos los profesores por disciplina y asignaturas, así como los estudiantes que reciben clases de las mismas.
- **Jefe de Disciplina:** Persona encargada de llevar el control de todos los profesores por disciplina, así como los estudiantes que reciben clases de las mismas.
- **Jefe de Asignatura:** Persona encargada de llevar el control de todos los profesores por asignaturas, así como los estudiantes que reciben clases de las mismas.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

- **Coordinador de Año:** Persona encargada de ordenar, controlar y poner en estado de normalidad todo referente a un año correspondiente en la facultad. Además, debe regular cualquier situación no esperada.

**Secretaria Docente de Asistencia:** Persona encargada de llevar la organización de los documentos de los directivos del centro.

**Grupo:** Pluralidad de individuos que se relacionan entre sí, con un cierto grado de interdependencia, que dirigen su esfuerzo a la consecución de un objetivo común con la convicción de que juntos pueden alcanzar este objetivo mejor que en forma individual. En este caso se refiere a un grupo de estudiantes.

**Indisciplina:** Se consideran actos de indisciplina todas las acciones, palabras, actitudes, gestos y reacciones que contrarían las normas disciplinarias vigentes en el centro de enseñanza, o que representan atentados contra la moral, la autoridad, el orden, el espíritu y las tradiciones de la institución.

### Diagrama de Clases del Modelo del Dominio

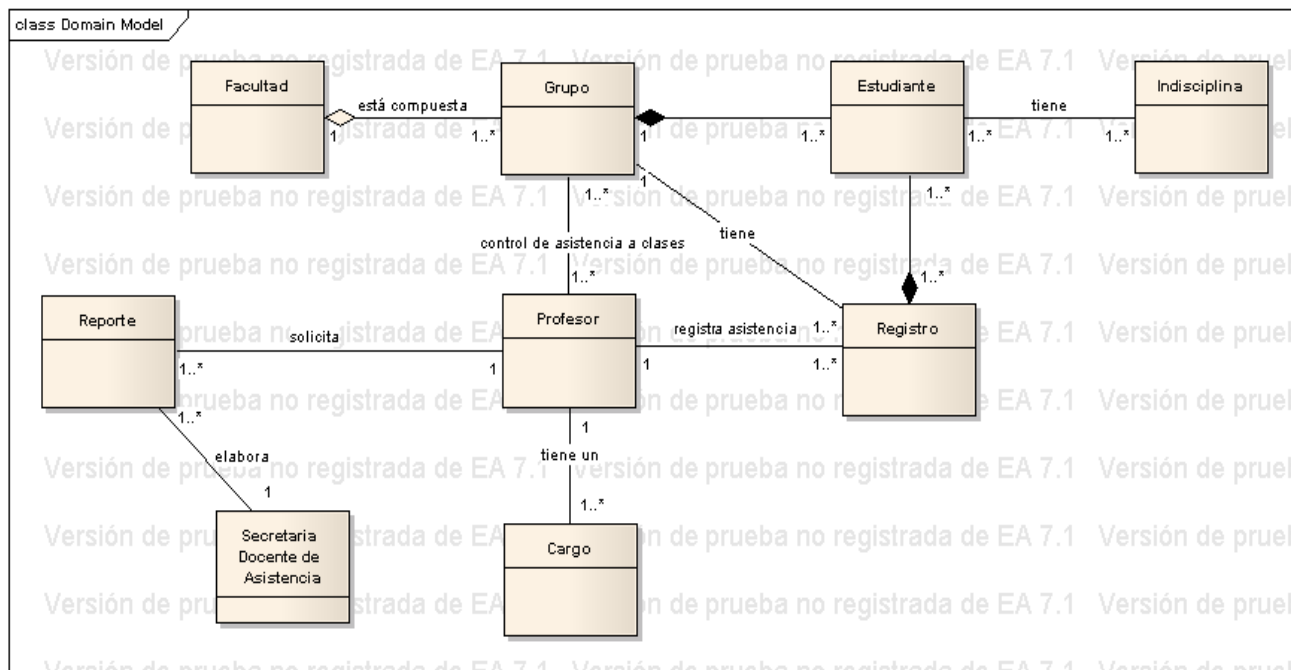


Fig. 1 Diagrama de Clases del Modelo del Dominio.

### Reglas del Negocio

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

---

Durante el desarrollo del negocio se determinó establecer un conjunto de reglas, estas deben tomarse en cuenta al llevar a cabo la informatización de los procesos del negocio, esto permite que el sistema elaborado se ajuste y garantice el cumplimiento de las necesidades del cliente.

1. De los estudiantes, se debe tener toda la información relacionada con las asignaturas que le corresponde recibir durante el semestre del año que cursa.
2. Se considera ausencia justificada a la no presencia en clases de aquellos estudiantes que adelantaron asignaturas, que se encuentran de cuartelaría, que solicitaron pase no reglamentado, a los estudiantes enfermos o de certificado.
3. Se debe contar con toda la información acerca del claustro de profesores de la facultad, además de definir los cargos que ocupa.
4. Tener una restricción de acceso a información según los cargos del profesor que quiera acceder a la misma.
5. Cuando un estudiante tiene 20% de inasistencia a clases se debe notificar al profesor guía, coordinador de año y al Decano de la Facultad.
6. El profesor puede registrar la asistencia en un mismo día si transcurre un plazo de al menos 60 minutos.

### **Modelo de Objeto**

El modelo de objetos del negocio no es más que un modelo de objetos que describe cómo colaboran los trabajadores y las entidades del negocio dentro del flujo de trabajo del proceso de negocio (Ver Fig. 2). A continuación se exponen quienes serían los trabajadores del negocio y las entidades del mismo.

### **Trabajador del Negocio:**

Persona o sistema que desempeña un rol dentro de las del Negocio y realiza las actividades en este caso del mecanismo de control de asistencia docente. Estos trabajadores (Profesor y Secretaria docente) dentro de la frontera del negocio serán los futuros usuarios de la aplicación a desarrollar para el Control de la Asistencia a Clases en la Facultad 7. (Ver Tabla2.)(Ver Fig. 2)

### **Entidad del Negocio:**

Documento o sistema del cual se consulta o se toma información significativa, que es manipulada por los actores y trabajadores del negocio. Las entidades del negocio de la Aplicación Web para el Control de la Asistencia a Clases en la Facultad 7, representa las entidades reales: Informe de Asistencia y Registro. Estas entidades son la base para compartir documentos entre los profesores y directivos de la facultad y éstas son utilizadas para llevar el mecanismo de control de asistencia del Negocio. (Ver Fig. 2)

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Trabajador del Negocio	Justificación
Profesor	Encargado de registrar y controlar la asistencia docente a clases. Si desempeña algún cargo en específico, realiza las actividades correspondientes al mismo, tomando las medidas pertinentes, para lograr de manera satisfactoria la formación de los futuros ingenieros.
Secretaria Docente	Encargada de almacenar y organizar toda la información referente a la docencia de los estudiantes, de acuerdo a las necesidades de los directivos del centro.

Tabla 2. Trabajadores del Negocio. Descripción.

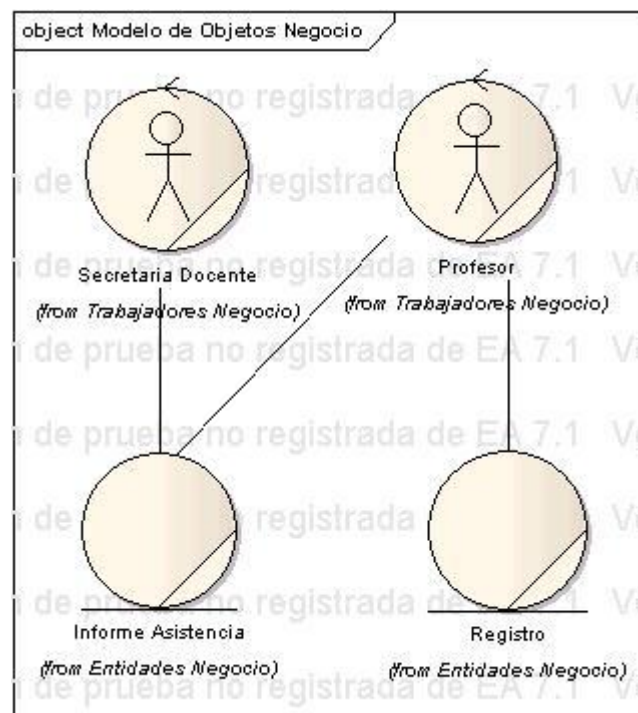


Fig. 2 Diagrama del Modelo de Objeto del Negocio

### Especificación de los requisitos de software

#### Requerimientos Funcionales:

RF1: Controlar el acceso a la información correspondiente a cada usuario.



## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

---

1.1: Definir niveles de acceso.

RF2: Gestionar la asistencia a clases.

2.1: Registrar la asistencia.

2.2: Modificar la asistencia.

RF3: Generar reportes.

3.1: Informe de asistencia.

3.1.1 – Informe de asistencia diario en % de la Facultad o por año, grupo y estudiante.

3.1.2 – Informe de asistencia semanal en % de la Facultad o por año, grupo y estudiante.

3.1.3 – Informe de asistencia mensual en % de en la Facultad o por año, grupo y estudiante.

3.2 Listado de estudiantes con mayores incidencias en la asistencia a clases.

3.3: Profesores con mayor incidencia de estudiantes ausentes a su clase.

3.4: Informe con los análisis realizados a un estudiante.

RF4: Enviar justificación de inasistencia a clases.

RF5: Registrar sanciones de estudiantes.

RF6: Consumir servicios de las aplicaciones que están integradas al sistema, como son el Sistema de Horario Docente, el Sistema de Gestión de Pase no Reglamentado F7 y el Sistema de Cuartelería.

### **Requerimientos no Funcionales:**

RNF 1: Usabilidad.

1.1: Documentar adecuadamente la aplicación. Esto proporciona que cada usuario pueda satisfactoriamente utilizar las funcionalidades que brinda la misma.

1.2: Adiestrar al personal. El personal que hace uso del sistema, debe poseer la capacitación requerida al nivel que se establece.

1.3: Definir nivel de acceso. El sistema sólo puede ser usado por los estudiantes y profesores que integran la Facultad 7.

1.4: Tener en cuenta la experiencia del usuario, la aplicación debe ser fácil de usar.

RNF 2: Fiabilidad.

2.1: Disponer en un 45.83 % de horas al día de prestación de servicio que equivale a 8 horas diarias.

2.2: Acceder para dar mantenimiento al sistema externamente al horario de disponibilidad del sistema.

2.3: Brindar prestaciones cuando esté en modo de funcionamiento con problemas, sólo si las funcionalidades de registrar y modificar asistencia continúan brindando sus servicios.

2.4: Definir el tiempo medio entre fallos del sistema, que solo es de 0.5 horas.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

---

- 2.5: Definir el tiempo medio de reparación del sistema luego de un falló, que es sólo de 0.416 horas.
- 2.6: Especificar la salida del sistema, el usuario debe dirigirse a la parte superior de la interfaz que se observa, a la opción **Salir** y confirmar la salida.
- 2.7: Admitir como máximo de errores 100/MLC.
- 2.8: Categorizar los errores. Categoría definida: menores, significativos y críticos. La primera categoría es asignada cuando se produzca un error producto al funcionamiento en la red o la electricidad del local donde se imparte la clase. La segunda se asigna cuando existan problemas en el nivel de acceso a la información y la tercera cuando haya pérdida total de la información a registrar o esté inhabilitado alguna funcionalidad del sistema.

### RNF 3: Eficiencia.

- 3.1: Permitir la visualización de los Reportes de asistencia en un tiempo de respuesta menor o igual a 15 segundos.
- 3.2: Obtener la Información Estadística de la asistencia a clases o cualquier otro tipo de información referente a la evolución de un estudiante de la facultad. Esta funcionalidad envía respuestas SOAP que no deben exceder los 50 Kbyte en el tiempo de respuesta al usuario.
- 3.3: Implementar sobre la tecnología Web las prestaciones, para facilitar su uso a través de la red.

### RNF 4: Soporte.

- 4.1 Utilizar como medio de almacenamiento de la información una base de datos soportada sobre el SGBD PostgreSQL.
- 4.2: Integrar el sistema de manera que se mantenga la colaboración entre todos los subsistemas que lo componen.

### RNF 5: Restricciones de diseño e implementación.

- 5.1: Definir la lógica de presentación que constituye una capa independiente de la lógica de negocio, centrando su función en la interfaz de usuario y validaciones simples de los datos de entrada.
- 5.2: Validar el proceso de la captación de datos para evitar entradas inadecuadas.
- 5.3: Utilizar PostgreSQL como Sistema de Gestión de Base de Datos.
- 5.4: Utilizar como framework Symfony.

### RNF 6: Interfaz.

- 6.1: Interfaces de usuario
  - 6.1.1: Realizar interfaces con un diseño sencillo que contenga pocos gráficos, para así acelerar las respuestas del sistema al usuario y brindarle mayores posibilidades de adaptación al sistema.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

---

6.1.2: Los estilos CSS usados, son los que define el portal de la Facultad.

6.2: Interfaces Hardware.

6.2.1: El usuario requiere una máquina con un procesador 600 megahertz (MHz) como mínimo y se recomienda 800 (MHz).

6.2.2: Memoria RAM como mínimo 128 megabytes (MB) y se recomienda 256 MB.

6.2.3: Requiere una capacidad en el disco duro de 20GB como mínimo.

6.2.4: Monitor VGA o superior.

6.2.5: Ratón Microsoft o compatible.

6.2.6: Impresora local o de red para imprimir los reportes solicitados.

6.3: Interfaces Software.

6.3.1: Sistema operativo Windows con interfaz gráfica y soporte para red.

6.3.2: Navegador Web Mozilla 1.5 o Internet Explorer 5.0 ó superior.

6.3.3: Se utiliza el LDAP como protocolo de ligero acceso a directorios.

RNF 7: Requisitos Legales, de Derecho de Autor y otros.

7.1 Establecer derechos. Este sistema de Control de Asistencia para la Secretaría Docente de la Facultad 7 pertenece a la UCI, específicamente a la Secretaría Docente de la Facultad 7.

RNF 8: Estándares Aplicables.

8.1 Definir un estándar de codificación para lograr la reutilización de código en el equipo de trabajo y un satisfactorio entendimiento del código desarrollado.

RNF 9: Portabilidad.

9.1: Utilizar componentes que puedan tener funcionamiento en sistemas operativos como Microsoft® Windows® 98, Windows 98 Second Edition o superior y/o GNU-Linux.

RNF 10: Seguridad.

10.1: Garantizar que la información sea editada únicamente por el personal autorizado. Que contenga verificación sobre acciones irreversibles (eliminaciones).

10.2: Emplear métodos para asegurar la integridad de los datos.

10.3: Contar con varios niveles de acceso para permitir el trabajo organizado con el sistema.

10.4: Guardar la información de los estudiantes de modo confidencial, por lo que sólo puede ser consultada por el personal autorizado y esta prohibida su divulgación.

10.5: Realizar la comunicación usando un protocolo seguro. (Https).

RNF 11: Requisitos de Disponibilidad.

11.1: Garantizar el acceso a la información y que los dispositivos o mecanismos utilizados para lograr la seguridad no ocultan o retrasan a los usuarios para obtener los datos deseados en un momento dado.

11.2: Establecer el funcionamiento del sistema las 8 horas del día en que se impacten clases.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

---

RNF 12: Rendimiento.

12.1: El tiempo de respuesta de cada una de las páginas debe ser menor de un minuto, excepto aquellas que requieren más tiempo.

12.2: Se implementa sobre tecnología web, facilitando su uso en la red.

12.3: Permitir la visualización de los Reportes de asistencia en un tiempo de respuesta menor o igual a 15 segundos.

### Definición de los casos de uso y actores del sistema

◆ Definición de los actores.

Actores	Justificación
Profesor	Puede ver el estado de asistencia de los grupos a los que le imparte clase, y es el único encargado de registrar o modificar la asistencia de los mismos. Obtiene reportes tales como: informes de asistencia diario, semanal o mensual, % de estudiantes ausentes a su clase, listado de profesores con mayor incidencia de estudiantes ausentes a clases, listado de estudiantes con mayor dificultad en asistencia a clases, de acuerdo al rol que se le ha otorgado.
Estudiante	Es el encargado de enviarle al profesor, la justificación del por qué no asiste a clases. Puede ver su estado actual de asistencia.
Jefe de Comisión Disciplinaria	Es el encargado de registrar las sanciones realizadas a estudiantes que presentan problemas de asistencia a clases. Además en el rol de profesor de la facultad, puede ejecutar todas las tareas que un profesor realiza.
Usuario	Generaliza el rol de autenticación al sistema y puede obtener reportes según el nivel de acceso que tenga, dígame estudiante, profesor, profesor guía, jefe de año, jefe de asignatura, jefe de disciplina, jefe de

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	departamento, jefe de comisión disciplinaria, vicedecano de formación, secretaria.
Sistema de Gestión Académica (versión 2)	Brinda un grupo de servicios que pueden ser reutilizados. Mediante él se pueden obtener los estudiantes dados un profesor o un grupo. Posibilita la búsqueda de profesores dado un grupo o estudiante, la búsqueda de estudiantes de una facultad, entre muchos otros servicios.
Registro de Identidad UCI (versión 3)	Brinda un grupo de servicios que pueden ser reutilizados. Mediante el se pueden obtener una persona dado el usuario y contraseña.
Sistema de Gestión de Pase no Reglamentado F7	Permite gestionar la solicitud de pase por parte de los estudiantes. Mediante él se puede obtener los estudiantes que se encuentran de pase, y por tal motivo no pueden asistir a clases.
Sistema de Horario Docente	Permite gestionar las afectaciones de los profesores, para poder elaborar el horario docente. Mediante el se gestiona el cargo del profesor.
Sistema de Cuartelería	Permite gestionar los estudiantes que están de cuartelería. A través de él se puede obtener los estudiantes que se encuentran de cuartelería y por tal motivo están justificados para no asistir a clase.

Tabla 3. Definición de actores del sistema.

**Casos de Uso del Sistema:** Autenticar Usuario, Enviar justificación, Generar reportes de Asistencia, Gestionar Asistencia, Registrar Sanción.

<b>CUS- 1</b>	Autenticar Usuario
<b>Actor</b>	(Usuario)
<b>Descripción</b>	Permite, una vez que el usuario introduzca los datos

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	que se le piden para acceder a la aplicación, que éstos sean verificados y admitan que se le den los permisos al usuario, según su rol y se le habilite la entrada a la aplicación con las operaciones que puede realizar. Este caso de uso una vez que se integren todos los sistemas al portal de la facultad, se hace a través del mismo.
<b>Referencia</b>	RF1

Tabla 4. Descripción del CU Autenticar Usuario.

<b>CU- 2</b>	Enviar justificación
<b>Actor</b>	(Estudiante)
<b>Descripción</b>	Permite al estudiante, enviar al profesor que imparte la asignatura, el motivo por el cual no asistirá a clases. Posibilita de esta manera que el profesor al llegar al turno y registre la asistencia, sepa ya la causa, por la cual el estudiante no asistirá a clase, y lo notifique como justificado o no.
<b>Referencia</b>	RF4

Tabla 5. Descripción del CU Enviar Justificación.

<b>CU- 3</b>	Gestionar asistencia.
<b>Actor</b>	(Profesor)
<b>Descripción</b>	Permite que el profesor pueda registrar la asistencia o una vez registrada la pueda modificar si se transcurre el tiempo que se requiere.
<b>Referencia</b>	RF2

Tabla 6. Descripción del CU Gestionar Asistencia.

<b>CU- 4</b>	Registrar sanción
<b>Actor</b>	(Jefe de Comisión Disciplinaria)
<b>Descripción</b>	Permite al profesor (Jefe de comisión disciplinaria) dejar archivado las sanciones que se le realizan a un estudiante determinado, para que luego el resto de los profesores puedan tener acceso a ella de ser

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

---

	necesario y al mismo tiempo llevar un seguimiento del estudiante. Estas no se pueden modificar y los posibles tipos de sanción se seleccionan a través de un nomenclador.
<b>Referencia</b>	RF5

**Tabla 7. Descripción del CU Registrar Sanción.**

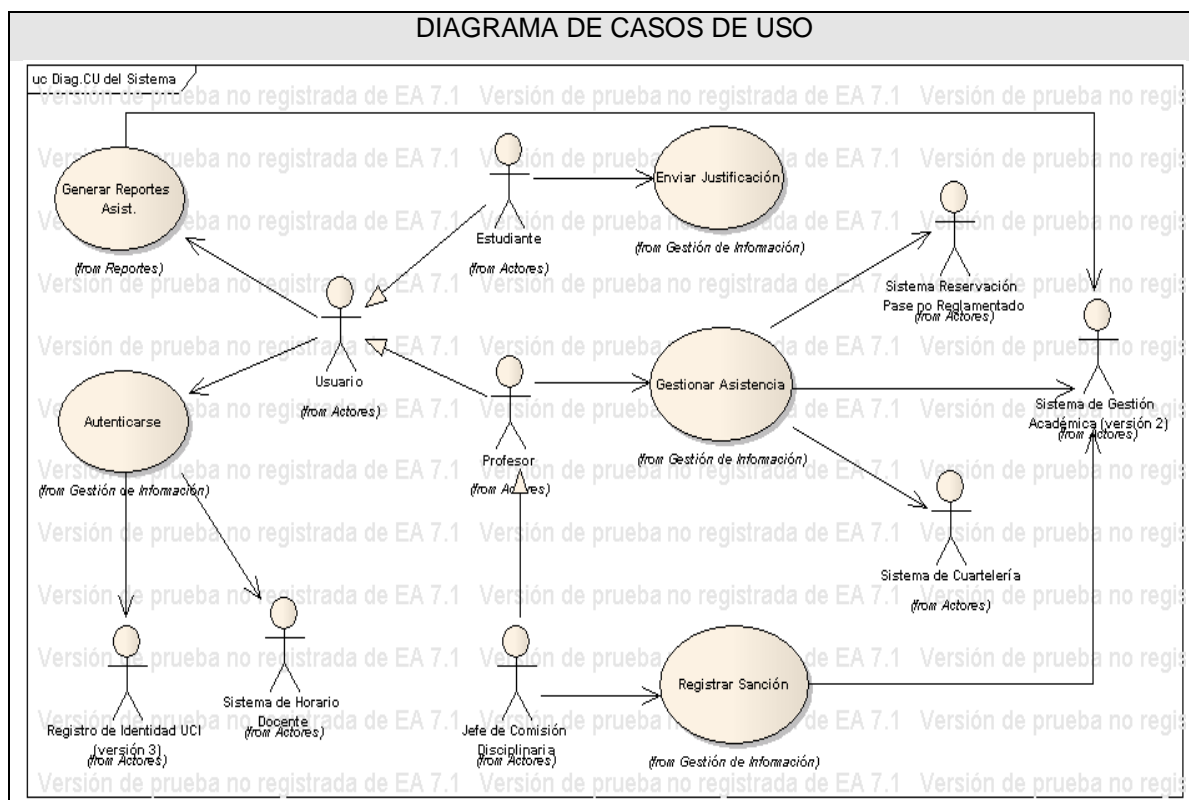
<b>CU- 5</b>	Generar Reportes de Asistencia
<b>Actor</b>	Usuario (Estudiante o Profesor)
<b>Descripción</b>	Posibilita la obtención de todos los reportes que desea el usuario, ya sea profesor o estudiante, según el rol que desempeñan, y los permisos de acceso a información que tengan. Reportes tales como: informe de asistencia diario, semanal o mensual, listado de estudiantes con mayor dificultad en cuanto a asistencia, profesores con mayor incidencia de estudiantes ausentes, chequear estado actual de asistencia, entre otros.
<b>Referencia</b>	RF3

**Tabla 8. Descripción del CU Generar Reportes de Asistencia.**

### Diagrama de casos de uso

Los Casos de Uso no son más que lo que hace el sistema desde el punto de vista del usuario. Es decir, describen un uso del sistema y cómo este interactúa con el usuario. Cuando empiezas a tener un número considerable de casos de uso, no resulta nada fácil situarlos y relacionarlos. Entonces empiezas a tener la necesidad de una visión general del asunto, es ahí cuando los diagramas de casos de uso son de gran utilidad. Con esto consigues una visión general de cómo los diferentes actores interactúan con los distintos casos de uso.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA



**Fig. 3 Diagrama de Caso de Usos del Sistema.**

### Casos de Uso Expandidos

Caso de uso	
CU-1	Autenticar Usuario
<b>Propósito</b>	Permitir autenticarse.
<b>Actor:</b> Usuario	
<b>Resumen:</b> El caso de uso se inicia cuando el usuario desea acceder a la aplicación. Para autorizar su entrada a la aplicación el sistema le solicita los datos de usuario y contraseña. Estos son verificados para darle los permisos según el rol que desempeña cada usuario. El caso de uso finaliza cuando se le habilita la entrada y los permisos al usuario a la aplicación.	
<b>Referencias</b>	RF1
<b>Flujo de sucesos</b>	
Acción del actor	Respuesta del sistema
1. El usuario entra Usuario y Contraseña.	2. Es sistema encripta la contraseña. 3. Busca el usuario y compara la contraseña. 4. En caso de ser correcto se verifica el cargo que ocupa el usuario en la facultad. 5. Una vez que se confirma el cargo se le



## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	<p>asignan los permisos.</p> <p>6. Muestra la interfaz del usuario y con las operaciones a las que tiene acceso.</p>
<b>Flujo Alternativo</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	5. En caso de no existir se envía un mensaje de error.

Tabla 9. Descripción del CU Expandido Autenticar Usuario.

Caso de uso	
CU-2	Enviar justificación
<b>Propósito</b>	Permitir que el estudiante introduzca un comentario y seleccione a los profesores que le desea enviar el motivo por el cual no asistirá a clases.
<b>Actor:</b> Estudiante	
<b>Resumen:</b> El caso de uso se inicia cuando el estudiante que no asistirá a clases, selecciona la opción Enviar Justificación del menú. Esta le permite escribir un comentario sobre el motivo por el cual se ausentará. De esta manera cuando el profesor llegue al turno y registre la asistencia, sabrá la causa, por la cual el estudiante no asistirá a clase, y lo notifica justificado o no. El caso de uso finaliza cuando el estudiante envía la justificación.	
<b>Referencias</b>	RF4
<b>Precondiciones</b>	Estudiante ya autenticado.
<b>Poscondiciones</b>	Se envía la justificación al registro.
<b>Flujo de sucesos</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El Estudiante selecciona la opción: Enviar Justificación, que le brinda la aplicación en el menú.	<p>2. El sistema muestra una interfaz para que introduzca la fecha en la cual no asistirá (Fecha Inicio y Fecha Fin) en caso de que se ausente más de un día.</p> <p>3. Le permite que seleccione también el tipo de Justificación.</p>

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	4. Posibilita que introduzca un comentario.
5. Introduce la fecha en la que se ausentará. 6. Selecciona el tipo de justificación. 7. Escribe la justificación y luego elige la opción (Aceptar o Cancelar).	8. Valida los datos. 9. Se envía la justificación al registro.
<b>Flujo Alternativo</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	8. No acepta al dar cancelar regresa a la página de inicio del estudiante. 9. Lanza un mensaje de error. Error al salvar los datos.

Tabla 10. Descripción del CU Expandido Enviar Justificación.

Caso de uso	
CU-3	Gestionar asistencia
<b>Propósito</b>	Permitir registrar la asistencia y modificarla en caso de que existan tardanzas.
<b>Actores:</b> Profesor	
<b>Resumen:</b> El caso de uso se inicia cuando el profesor selecciona del menú la opción Registro de Asistencia. Este puede registrar o modificar la asistencia, en caso de que existan tardanzas (llegada tarde), de uno de los grupos al cual le imparte clases. El caso de uso finaliza una vez que sea registrada o modificada la asistencia por el profesor.	
<b>Referencias</b>	RF2
<b>Precondiciones</b>	El profesor tiene que haberse autenticado.
<b>Poscondiciones</b>	Queda actualizado el registro de asistencia del estudiante.
<b>Flujo de sucesos</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El profesor selecciona del menú	2. El sistema muestra una interfaz para que

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

la opción Registro de Asistencia.	<p>seleccione la fecha en la que va a registrar la asistencia.</p> <p>3. Luego brinda la posibilidad de seleccionar el grupo al que le va a registrar la asistencia en particular.</p>
<p>4. Selecciona la fecha en la que desea registrar la asistencia.</p> <p>5. Selecciona el grupo al que le va a registrar la asistencia.</p>	<p>6. El sistema le muestra una pantalla con el registro donde aparece todos los estudiantes presentes y en caso de que el estudiante este ausente y haya enviado su justificación o el mismo se encuentre de pase o de cuartelero, aparece en el campo de tipo de justificación un comentario con el motivo de porque no asistió a clases.</p>
<p>7. Registra la asistencia, se pone, presente, ausente, tarde o justificado.</p> <p>8. Acepta para agregar los datos o Cancela la operación.</p>	<p>9. El sistema inserta el nuevo registro de asistencia del grupo.</p> <p>10. Muestra un mensaje: Los datos fueron guardados correctamente.</p>
<b>Flujo Alternativo</b>	
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>
	<p>6. Si en la fecha seleccionada ya fue registrada la asistencia el sistema le muestra el registro para que el profesor modifique los datos en caso de ser deseado.</p>
<p>7. Modifica los datos deseados.</p> <p>8. Acepta o Cancela la operación.</p>	<p>9. El sistema inserta el registro de asistencia del grupo que fue modificado.</p> <p>10. Muestra un mensaje: Los datos fueron guardados correctamente.</p>
	<p>9. Si el profesor Cancela la operación se regresa a la página principal del profesor.</p>

**Tabla 11. Descripción del CU Expandido Gestionar Asistencia.**

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Caso de uso	
CU-4	Registrar sanción
<b>Propósito</b>	Permitir registrar las sanciones de los estudiantes.
<b>Actores:</b> Jefe de comisión disciplinaria	
<b>Resumen:</b> El caso de uso se inicia cuando el Jefe de comisión disciplinaria, una vez realizado el análisis, accede a la aplicación para dejar registrado y documentado las medidas que se tomaron con el estudiante. Finaliza luego de que el Jefe de comisión disciplinaria introduzca los datos en el sistema y este los archiva.	
<b>Referencias</b>	RF5
<b>Precondiciones</b>	El Jefe de comisión disciplinaria tiene que haberse autenticado.
<b>Poscondiciones</b>	Se archiva la sanción realizada.
Flujo de sucesos	
Acción del actor	Respuesta del sistema
1. El Jefe de comisión disciplinaria selecciona en el menú, Registrar Indisciplina.	2. El sistema le muestra una interfaz para introducir los datos del estudiante. 3. Permite seleccionar el grupo del estudiante. 4. Luego posibilita seleccionar el estudiante. 5. Seleccionar la fecha. 6. Seleccionar el tipo de indisciplina. 7. Posibilita introducir una breve descripción.
8. Introduce los datos necesarios para enviar el acta de indisciplina. 9. Acepta o Cancela la operación.	10. Valida los datos de entrada. 11. Archiva los datos de la indisciplina cometida. 12. Muestra un mensaje: los datos fueron guardados correctamente.
Flujos Alternativo	
Acción del actor	Respuesta del sistema
	11. Se muestra un mensaje de error. Error al salvar los datos. 10. Si decide cancelar el envío de la sanción, esta no se archiva y se regresa a la página del Jefe de Comisión Disciplinaria.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

--	--

Tabla 12. Descripción del CU Expandido Registrar Sanción.

Caso de uso	
CU-5	Generar reportes de asistencia
Propósito	Permitir que los profesores tengan acceso a reportes de asistencia actualizados en el momento que se necesiten, así como permitir que el estudiante pueda saber su estado actual de asistencia.
Actor: Usuario	
<b>Resumen:</b> El caso de uso inicia cuando el usuario, en dependencia del rol que desempeña, y del nivel de acceso a la información que tenga solicite un reporte determinado. En el caso de un profesor comienza cuando selecciona del menú la opción Reportes y finaliza cuando recibe el reporte seleccionado. En el caso del estudiante comienza cuando selecciona del menú la opción Estado Asistencia y finaliza una vez que se obtiene el reporte.	
Referencias	RF3
Precondiciones	El usuario tiene que haberse autenticado.
Poscondiciones	El usuario recibe el informe solicitado.
Flujo de sucesos	
Acción del actor	Respuesta del sistema
1. El usuario selecciona la opción de menú Reportes.	2. Muestra una interfaz para que seleccione el tipo de reporte que desea. <ul style="list-style-type: none"> <li>a) Estudiantes</li> <li>b) Grupo</li> <li>c) Año</li> <li>d) General</li> <li>e) Lista de estudiantes con mayor dificultad en asistencia.</li> <li>f) Profesores con mayor incidencia de estudiantes ausentes a clases.</li> <li>g) Informe con los análisis realizados a un estudiante.</li> <li>h) Porcentaje de estudiantes ausentes a</li> </ul>

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	clases.
3. El usuario selecciona el tipo de reporte.	4. Muestra los campos que debe llenar de acuerdo al tipo de reporte para poder recibir el mismo.
5. Introduce los criterios de búsqueda. (Estudiante, Grupo, Asignatura, Desde (Fecha de inicio) y Hasta (Fecha de fin)). 6. Acepta o Cancela la operación.	7. Valida los datos de entrada. 8. Muestra el reporte solicitado.
<b>Flujos Alternativo</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	8. Muestra un mensaje de error. 7. Si el usuario decide cancelar la operación, no aparece el reporte y se regresa a la página del usuario.

**Tabla 13. Descripción del CU Expandido Generar Reportes de Asistencia.**

### Conclusiones

El desarrollo de este capítulo permitió obtener, una mayor comprensión de las características del sistema propuesto. Así como de las restricciones que deben existir para dar cumplimiento a los requerimientos expuestos por el cliente. En su elaboración, se obtuvo el Modelo de Dominio donde se explica de forma detallada toda la información que se maneja y el mecanismo de control de asistencia docente llevado a cabo en la Facultad 7. Con la descripción detallada de los casos de uso, y los trabajadores del sistema, se logró un mayor entendimiento de la solución propuesta para el desarrollo de la aplicación web.

### **CAPÍTULO 3: Análisis y Diseño del Sistema**

La informática es un campo muy dinámico, constantemente aparecen nuevas tecnologías, esto implica cambios en los lenguajes de programación, cambios en la forma de utilizar las bibliotecas de funciones, cambios en los componentes del sistema operativo en el que se ejecuta, entre muchos otros. Es aquí donde radica la gran importancia del flujo de trabajo análisis y diseño, ya que el objetivo de este flujo de trabajo es traducir los requisitos a una especificación que describe cómo implementar el sistema.

En la construcción del sistema anteriormente propuesto, primeramente, se modela el análisis que consiste en obtener un enfoque del sistema, mediante la elaboración de diferentes artefactos contenidos en este flujo, como son los diagramas de clases del análisis con la definición de las mismas. Esto trae consigo un mejor entendimiento del sistema propuesto y sirve como base para el desarrollo del diseño. Luego se pasa al modelado del diseño que no por ser el segundo paso, deja de ser importante, todo lo contrario.

El diseño, se modela para que el sistema soporte y de cumplimiento a todos los requisitos funcionales y no funcionales expuestos por el cliente, de esta manera se contribuye para obtener una arquitectura sólida para una futura implementación del software. En este se incluyen los artefactos que se generan en el flujo de trabajo de diseño, la definición de los elementos que en él se encuentran, los diagramas de clases y la descripción de las mismas, todo esto mediante la realización de los casos de usos definidos en el capítulo anterior. Se describen los principios del diseño y los diagramas de iteración.

#### **Análisis**

##### **Modelo de análisis**

El análisis consiste en obtener una visión un poco más detallada del sistema, de modo que sólo se interesa por los requisitos funcionales, para refinarlos y estructurarlos. Esto proporciona una estructura centrada en el mantenimiento en aspectos como la flexibilidad ante los cambios y la reutilización de componentes, útil además para el comienzo de las actividades de diseño e implementación. Para describir un modelo de análisis se utiliza el lenguaje de los desarrolladores al razonar sobre los funcionamientos internos del sistema.

El objetivo del análisis, se centra en comprender perfectamente los requisitos del software y no precisar cómo se implementa el producto. En el modelo del análisis se refinan los requisitos, no se toman en cuenta el lenguaje de programación a usar en la construcción, la plataforma en la que se

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

ejecuta la aplicación, los componentes reutilizables de otras aplicaciones, entre otras características que afectan al sistema.

### Diagramas de Clases de Análisis

El diagrama de clases del análisis se realiza para cada caso de uso del sistema y muestra las clases participantes en dicho caso de uso y sus relaciones. En los diagramas de clases se identifican tres tipos de clases: Interfaz, Controladora y Entidad (Ver Tabla 14).

A continuación sólo se muestran los diagramas de los CU más significativos para la arquitectura. Ver [Anexo I](#).


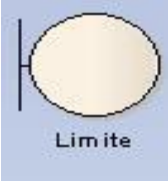

Representación	Nombre	Características
 Entidad	Entidad	Modelan información que posee larga vida y que es a menudo persistente.
 Limite	Interfaz	Modelan la interacción entre el sistema y sus actores.
 Control	Control	Coordinan la realización de uno o unos pocos casos de uso, regulando las actividades de los objetos que implementan la funcionalidad del caso de uso.

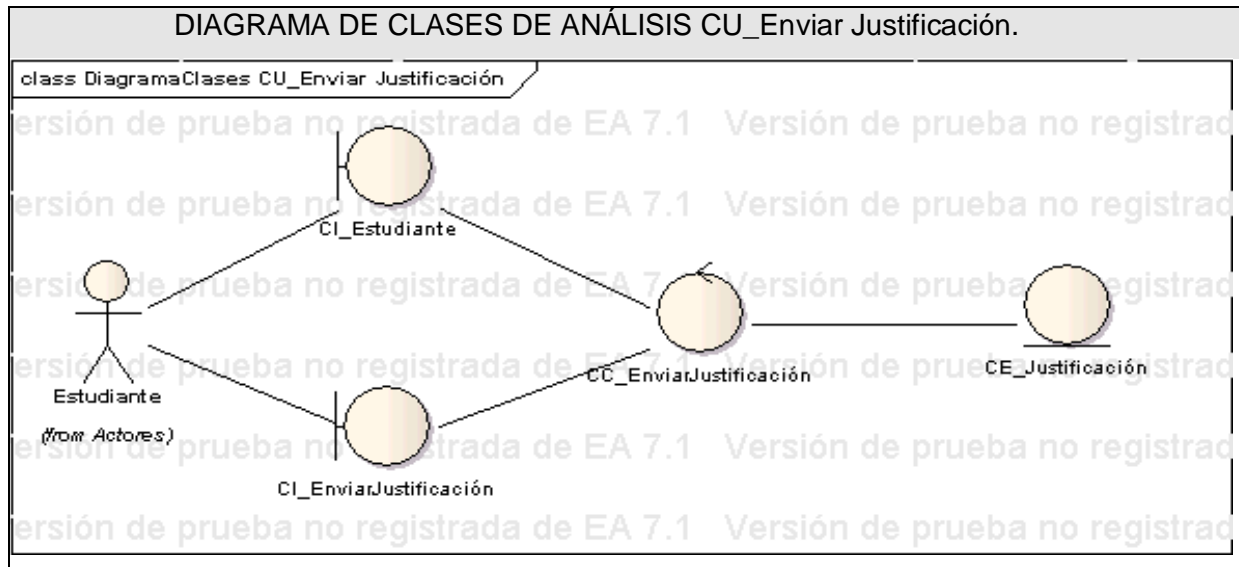
Tabla 14. Representación de los estereotipos a usar en el Diagrama de Clases del Análisis.

### Diagrama de clases de análisis del caso de uso: Enviar Justificación

Una vez autenticado el usuario le aparece la interfaz de acuerdo al rol o cargo que ocupa en la facultad y en este caso se tiene la clase CI\_Estudiante, la cual muestra las opciones que tienen los mismos. Aquí el estudiante selecciona la opción de enviar justificación. A través de la clase controladora CC\_EnviarJustificación que coordina las actividades de este caso de uso, se muestra la interfaz CI\_EnviarJustificación que permite introducir los datos para enviar la justificación al registro. Luego a través CC\_EnviarJustificación se inserta en la clase entidad CE\_Justificación que contiene las justificaciones que envían los estudiantes cuando no asisten a clases. (Ver Fig. 4)



## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA



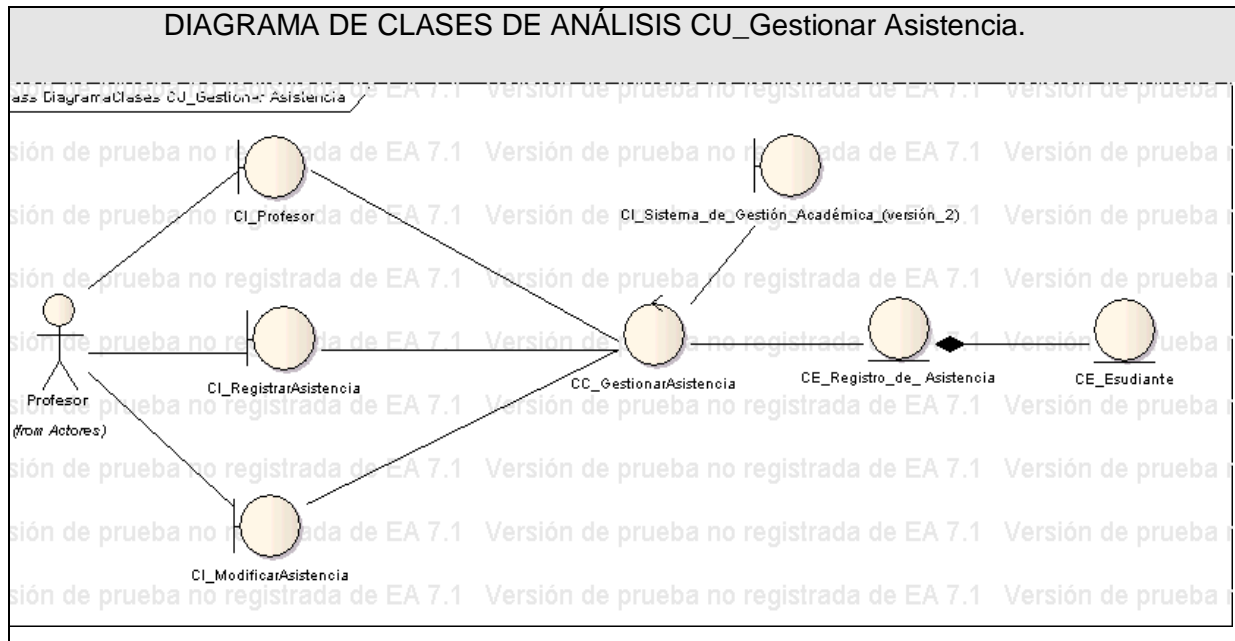
**Fig. 4 Diagrama de Clases del Análisis del Paquete Gestión de Información. Realización del CU: Enviar Justificación.**

### Diagrama de clases de análisis del caso de uso: Gestionar Asistencia

Una vez que el usuario se autentica le sale la interfaz **CI\_Profesor** que muestra las opciones de los profesores. Desde aquí el profesor selecciona la opción registrar o modificar la asistencia. A través de la clase **CC\_GestionarAsistencia**, la cual es la encargada de coordinar las actividades de registrar y modificar la asistencia, se muestra la interfaz **CI\_RegistrarAsistencia** que busca y muestra los datos del grupo para registrar la asistencia o **CI\_ModificarAsistencia** busca y muestra los datos del registro de asistencia del grupo para modificarla.

La búsqueda de los estudiantes dado el grupo se hace mediante la clase interfaz **CI\_Sistema\_de\_Gestión\_Académica (versión\_2)**, que no es más que un Sistema del cual se consumen servicios. Luego de ser registrada o modificada la asistencia se guarda en la clase entidad **CE\_Registro\_de\_Asistencia** que contiene la información sobre el registro de asistencia. (Ver Fig. 5)

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA



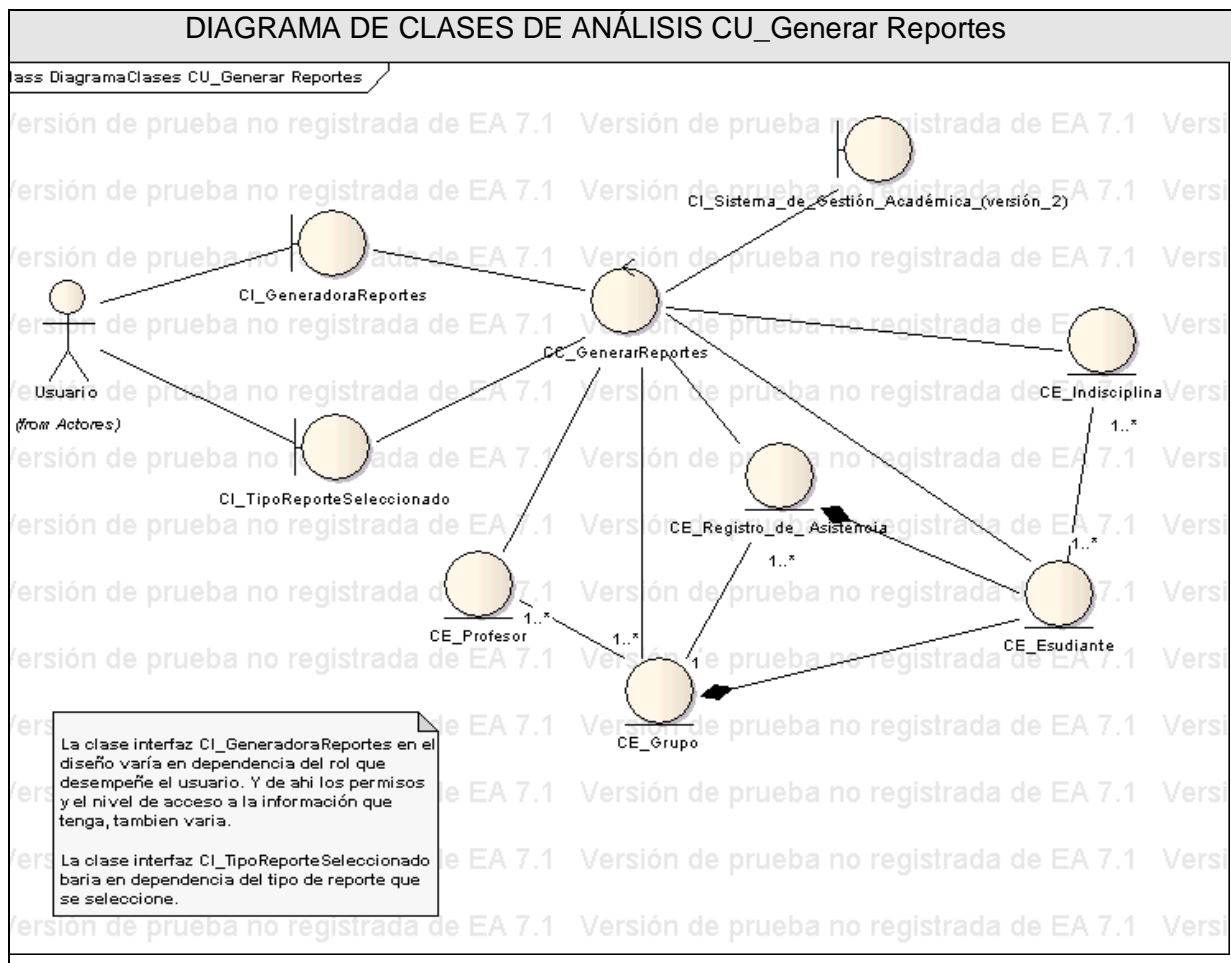
**Fig. 5 Diagrama de Clases del Análisis del Paquete Gestión de Información. Realización del CU: Gestionar Asistencia.**

### Diagrama de Clases de Análisis del caso de uso: Generar Reportes

El usuario selecciona en la clase interfaz CI\_GeneradoraReportes que permite acceder a los reportes la opción de reportes. Esta a través de la clase controladora CC\_GenerarReportes que es la que coordina las actividades de generación de reportes, según el tipo de reporte, le muestra la interfaz CI\_TipoReporteSeleccionado que sería la que le muestra los datos que se deben introducir para poder obtener el reporte deseado. Para cargar el nomenclador de estudiantes dado un grupo, que se puede pedir en algún tipo de reporte, se utiliza la interfaz CI\_Sistema\_de\_Gestión\_Académica (versión\_2)

La CI\_Sistema\_de\_Gestión\_Académica representa un sistema del cual se consumen servicios. Para darle respuesta al usuario del reporte deseado se consultan las clases entidades CE\_Grupo que contiene la información de los grupos, CE\_Registro\_de\_Asistencia que contiene la asistencia diaria de los estudiantes, CE\_Estudiante que contiene la información de estudiantes de cada grupo, CE\_Indisciplina que contiene la información de las sanciones que se le hacen a un estudiante y CE\_Profesor que contiene la información del profesor. (Ver Fig. 6)

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA



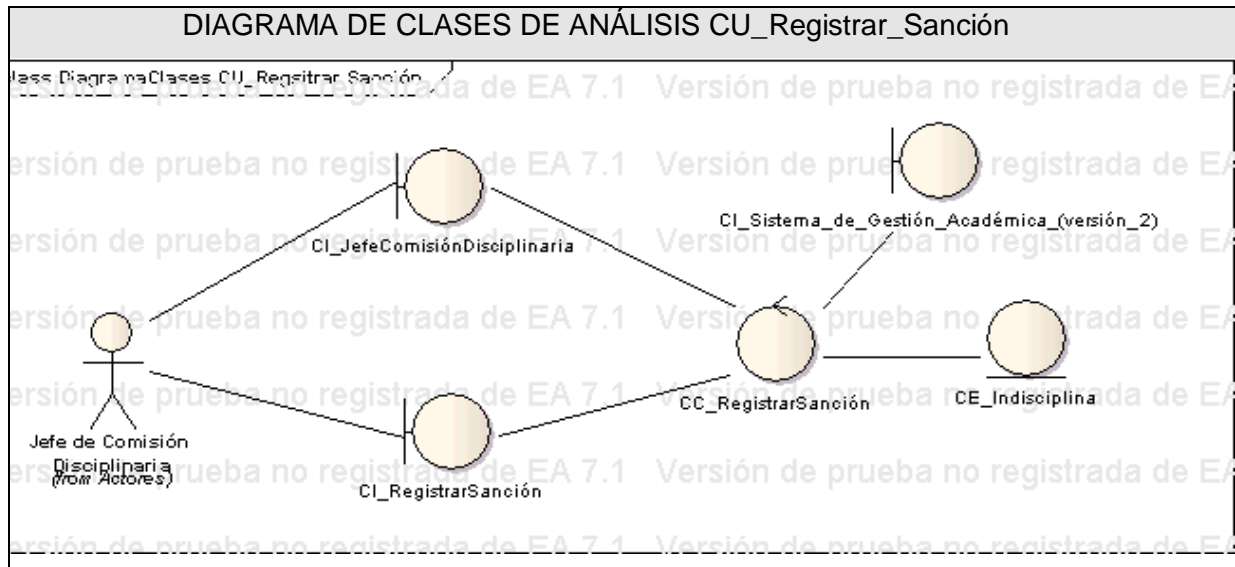
**Fig. 6 Diagrama de Clases del Análisis del Paquete Reportes. Realización del CU: Generar Reportes de Asistencia.**

### Diagrama de Clases de Análisis del caso de uso: Registrar Sanción

El caso de uso Registrar Sanción comienza después que el usuario se autentifique y acceda a la clase interfaz CI\_JefeComisiónDisciplinaria que es la que le muestra las opciones del jefe de comisión disciplinaria, único autorizado a registrar una sanción. Mediante la clase controladora CC\_RegistrarSanción se coordinan las actividades del registro de sanciones realizadas a los estudiantes y se muestra la clase interfaz CI\_RegistrarSanción que es donde se muestran los campos para llenar el acta.

Utilizando la clase interfaz CI\_Sistema\_de\_Gestión\_Académica que representa un sistema del cual se consumen servicios, se llena el nomenclador con los estudiantes dado un grupo. Luego de insertar todos los datos se registra la sanción en la clase entidad CE\_Indisciplina, esta contiene todos los datos referentes a las indisciplinas cometidas por un estudiante. (Ver Fig. 7)

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA



**Fig. 7 Diagrama de Clases del Análisis del Paquete Reportes. Realización del CU: Registrar Sanción.**

### Diseño

El modelo del diseño, es un modelo de objetos que describe la realización física de los casos de uso, centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar y constituye una entrada principal en la actividad de implementación. (25)

Para el desarrollo del Modelo de Diseño es necesario establecer una descomposición del mismo en subsistemas, con sus interfaces y las dependencias. Esta representación es de gran importancia para la arquitectura, ya que los subsistemas y sus interfaces constituyen la estructura fundamental del producto de software. Se utilizó el patrón de arquitectura MVC, que separa los datos de la interfaz de usuario, y la lógica del negocio, quedando conformado de la siguiente manera: (Ver Fig. 8)

**Modelo:** Esta es la representación específica de la información con la cual la aplicación de control de asistencia a clases opera. Representa las tablas de la Base de Datos, la cual puede ser, procesada por los Reportes a través de la Capa de Negocio. En esta se encuentra una carpeta con las tablas de la Base de Dato Externa de la aplicación, otra con las tablas que genera el Symfony por defecto y que se utilizan en el desarrollo de la aplicación y una con las clases persistentes.

**Vista:** las funcionalidades de esta consisten en intercambiar información con los actores, por lo tanto es la única capa que interactúa directamente con el usuario. Contiene los ficheros mediante los cuales son invocados los métodos de la Capa de Negocio. En esta se encuentran la clase interfaces, los formularios, y las bibliotecas utilizadas durante el diseño de la aplicación y una carpeta CP de

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

integración que en ella se encuentran las clase interfaces que se utilizan para representar la integración con los sistemas externos.

**Controlador:** Establece la comunicación entre el modelo y la vista, encargada de recibir y responder cada petición de los usuarios. Los ficheros que la conforman reciben las solicitudes de los clientes, se comunican con el modelo, actualizando o recuperando información, emite una respuesta. Constituye la parte del sistema donde se establecen todas las reglas de negocio que deben cumplirse. En esta se encuentran la clase sf\_frontend y ControlDocenteActions.

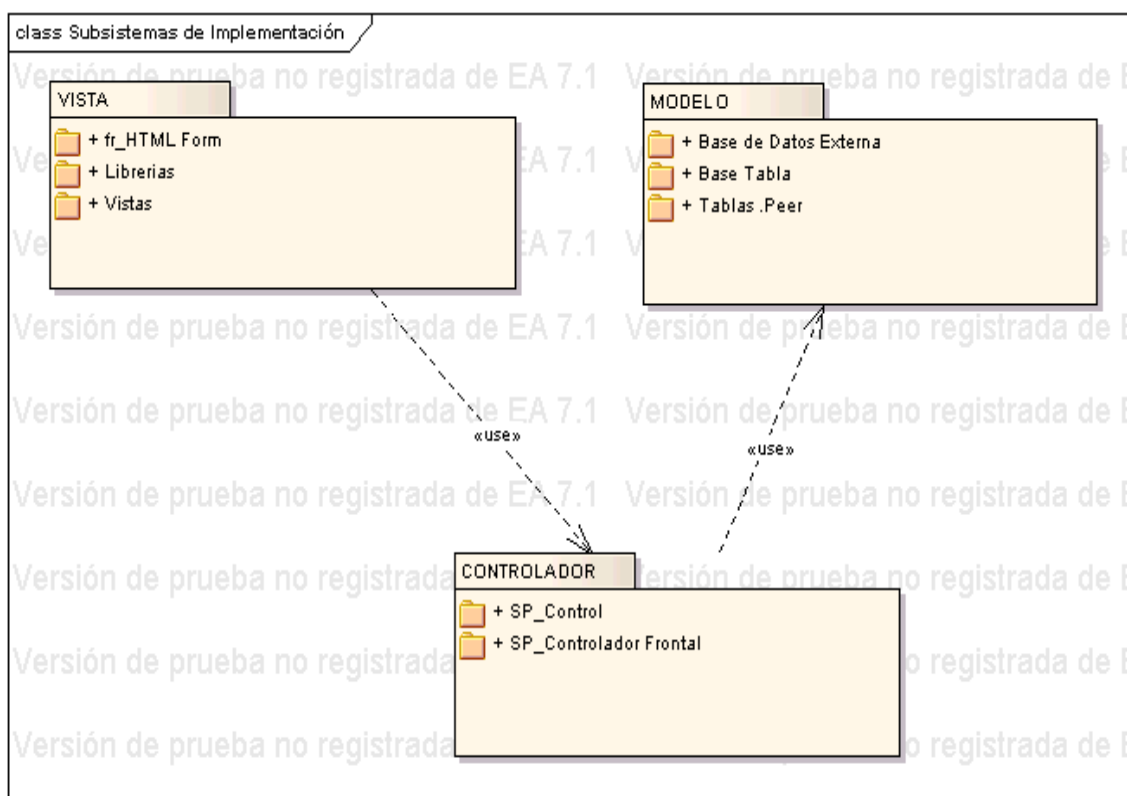


Fig. 8 Organización en paquetes de los Subsistemas de Servicios.

### Análisis de posibles implementaciones, componentes o módulos ya existentes y que puedan ser reusados

La integración de un sistema con otros componentes o módulos ya existentes, es un proceso indispensable para el correcto funcionamiento de las aplicaciones, debido a que se elimina la duplicidad de información almacenada en las bases de datos correspondientes a dichos sistemas externos. En el caso específico, de la Aplicación Web para el Control de la Asistencia a Clases en la Facultad 7, se necesita consumir servicios del Sistema de Gestión Académica Akademos, solicitar información del Registro de Identidad de la UCI, del Sistema de Gestión de Pase no Reglamentado F7, del Sistema de Cuartelería y del Sistema de Horario Docente.

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

---

### ◆ **Sistema de Gestión Académica: Akademos (versión 2)**

Brinda un grupo de servicios que pueden ser rehusados. Mediante él, se pueden obtener dado un área todo sus estudiantes, a partir del id del grupo obtener el grupo, dado un área obtener los grupos de esa área, entre otros. Se consumen para el desarrollo de la aplicación el servicio de dado un grupo obtener los estudiantes.

### ◆ **Registro de Identidad UCI (versión 3)**

Ofrece varios servicios para realizar búsquedas de una persona en particular, dígame: estudiante o profeso; a partir del solapín, carnet de identidad o el identificador de la persona. Se utiliza para el servicio de autenticación de la aplicación en desarrollo.

### ◆ **Sistema de Gestión de Pase no Reglamentado F7**

Permite gestionar la solicitud de pase por parte de los estudiantes. Mediante él se puede obtener los estudiantes que se encuentran de pase, y por tal motivo no pueden asistir a clases. Se utiliza para obtener todos los estudiantes que están de pase y ponerlos como justificado en el registro del profesor sin necesidad de que envíen la justificación.

### ◆ **Sistema de Cuartelería**

Permite gestionar los estudiantes que están de cuartelería. A través de él se puede obtener los estudiantes que se encuentran de cuartelería y por tal motivo están justificados para no asistir a clase. Se utiliza para obtener todos los estudiantes que están de cuarteros y ponerlos como justificado en el registro del profesor sin necesidad de que envíen la justificación.

### ◆ **Sistema de Horario Docente**

Permite gestionar las afectaciones de los profesores, para poder elaborar el horario docente. Mediante el se gestiona el cargo del profesor. Se utiliza como parte de la seguridad de la aplicación, a la hora de definir los roles de cada usuario para limitar el acceso a la información.

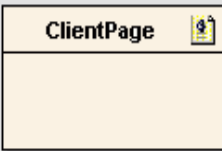
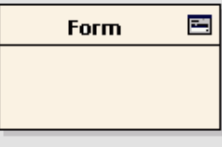
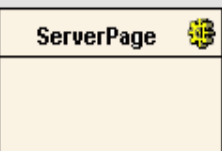
### **Estrategias de integración**

Para lograr la integración con estos sistemas, se utilizó la arquitectura Basada en componentes y se usó Servicios Web para el consumo de servicios.

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

### Modelado mediante estereotipos Web

Para la realización de los Diagramas de Clases del Diseño se utiliza la extensión de UML para el modelado de aplicaciones Web. Esta extensión presenta como elementos significativos a tres clases UML: Server Page, Client Page y Form empleadas para el código servidor, código cliente y formularios respectivamente, permite además representar ficheros contenedores de sentencias script. A continuación se muestra una tabla con las posibles relaciones entre clase, a la hora de desarrollar los diagramas del diseño de la aplicación. (Ver Tabla 15)

Estereotipos	Hasta Desde	Client Page	Form	Server Page
	Client Page	<b>&lt;&lt;link&gt;&gt;, &lt;&lt;redirect&gt;&gt;</b>	<b>aggregation</b>	<b>&lt;&lt;link&gt;&gt;, &lt;&lt;redirect&gt;&gt;</b>
Representan páginas que son mostradas por el navegador web y pueden ser una combinación de algún o algunos lenguajes de marcado.				
	Form	<b>aggregated by</b>		<b>&lt;&lt;submit&gt;&gt;</b>
Representa una colección de campos de entrada que forman parte con una página del lado cliente (Client Page).				
	Server Page	<b>&lt;&lt;build&gt;&gt;, &lt;&lt;redirect&gt;&gt;</b>		<b>&lt;&lt;redirect&gt;&gt;,&lt;&lt;include&gt;&gt;</b>
Representa una página Web que tiene scripts ejecutados por el servidor. Estos scripts interactúan con los recursos que se encuentran al alcance del servidor.				

**Tabla 15. Representación de los estereotipos a usar en el Diagrama de Clases del Diseño.**

La ServerPage es la responsable de construir o generar el resultado XHTML que conforma a la ClientPage mediante la relación <<build>>. Esta página cliente está compuesta por Form (formularios) de ahí que la relación entre ambas sea de agregación. Los formularios se encargan de enviar los datos a la ServerPage para ser procesados mediante la relación <<submit>>. Entre páginas clientes pueden existir vínculos (<<link>>) o redireccionamientos (<<redirect>>). Una página cliente es construida por una sola página servidora.

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

---

La servidora a su vez, puede completar su funcionamiento si incluye código existente en otra página de este mismo tipo, utilizando la relación de inclusión (<<include>>), que a pesar de no ser propia de la extensión de UML, es considerada para representar las relaciones existentes en el modelo. En este modelo del diseño, los casos de uso son realizados por las clases del diseño y sus objetos. A continuación se muestran los diagramas de clases del diseño y diagramas de interacción para la realización de los casos de usos descritos en el capítulo anterior.

### **Diagramas de clases del diseño**

Describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. Contiene información como: Clases, asociaciones y atributos; Interfaces, con sus operaciones y constantes; Métodos; Información sobre los tipos de los atributos; Navegabilidad; Dependencias.

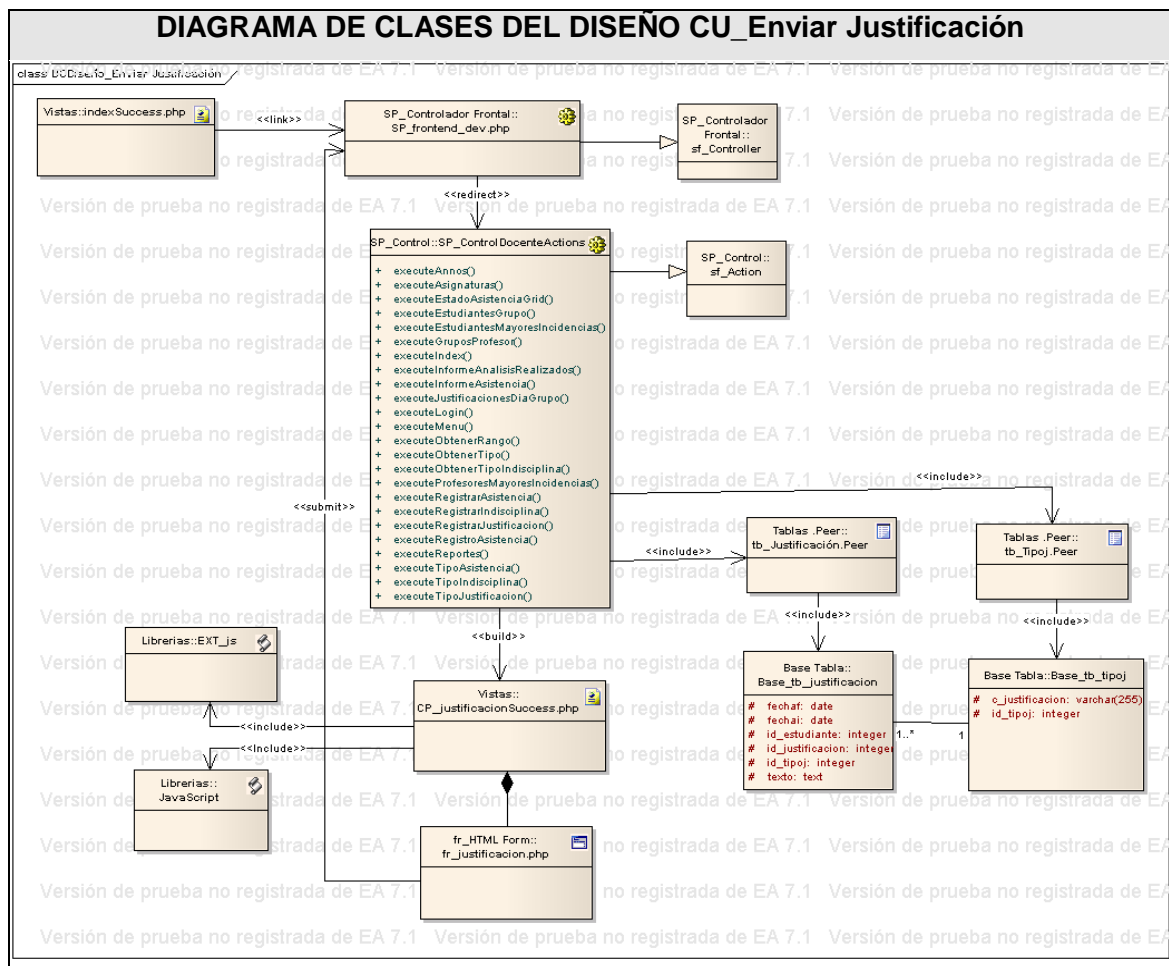
### **Diagrama de clases del diseño del caso de uso: Enviar Justificación**

En el diagrama se tiene la clase principal indexSuccess.php que es la página que se le muestra al usuario de acuerdo al cargo que ocupa una vez autenticado. Esta le hace un link al SP\_frontend\_dev.php que es el único punto de entrada a la aplicación y que hereda de una clase sf\_Controller que trae Symfony. Esta carga la configuración y determina la acción a ejecutarse, redireccionando al SP\_ControlDocenteActions que hereda de sf\_Actions y que contiene todas las operaciones que se realizan en el módulo de Control Docente.

SP\_ControlDocenteActions es la encargada de construir la CP\_justificacionSuccess.php que está compuesta por un formulario para la entrada de los datos. Además es la encargada de hacer las consultas a la base de datos por medio de las tablas base y .Peer que crea symfony por defecto para comunicarse con la base de datos externa. (Ver Fig. 9)



## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA



**Fig. 9 Diagrama de Clases del Diseño del Caso de Uso, Enviar Justificación. Paquete Gestión de Información.**

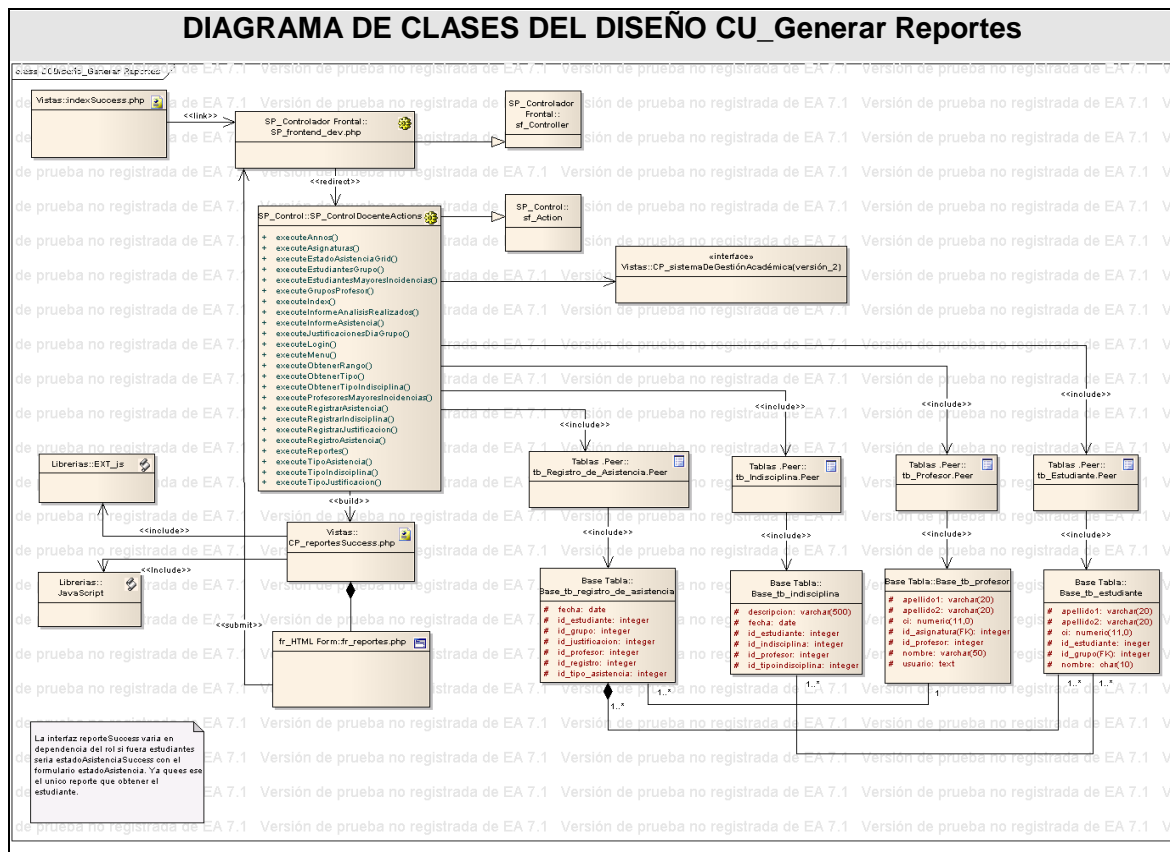
### Diagrama de clases del diseño del caso de uso: Generar Reportes

En el diagrama se tiene la clase principal indexSuccess.php que es la página que se le muestra al usuario de acuerdo al cargo que ocupa una vez autenticado. Esta le hace un link al SP\_frontend\_dev.php. Luego redirecciona la acción al SP\_ControlDocenteActions que contiene todas las operaciones que se realizan en el módulo de Control Docente. La SP\_ControlDocenteActions es la encargada de construir la CP\_reportesSuccess.php que está compuesta por un formulario para la entra de los datos.

Además es la responsable de hacer las consultas a la base de dato por medio de las tablas base y .Peer que crea symfony por defecto para comunicarse con la base de dato externa. La interfaz

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

CP\_reportesSuccess.php puede variar en dependencia del rol que tiene el usuario, si es estudiantes sería CP\_estadoAsistenciaSuccess.php. (Ver Fig. 10)



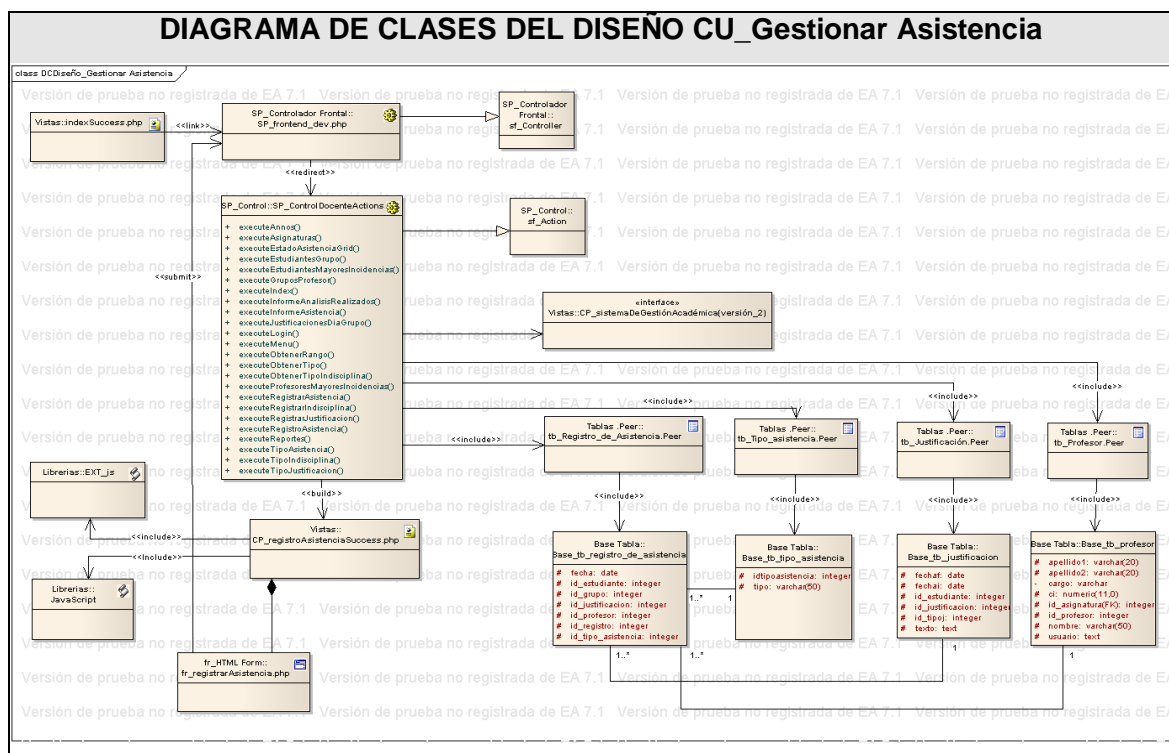
**Fig. 10 Diagrama de Clases del Diseño del Caso de Uso, Generar reportes de Asistencia. Paquete Reportes.**

### Diagrama de clases del diseño del caso de uso: Gestionar Asistencia

En el diagrama se tiene la clase principal indexSuccess.php que es la página que se le muestra al usuario de acuerdo al cargo que ocupa una vez autenticado. Esta le hace un link al SP\_frontend\_dev.php. Luego redirecciona la acción al SP\_ControlDocenteActions que contiene todas las operaciones que se realizan en el módulo de Control Docente. La SP\_ControlDocenteActions es la encargada de construir la CP\_registroAsistenciaSuccess.php que está compuesta por un formulario para la entrada de los datos.

Además es la responsable de hacer las consultas a la base de datos por medio de las tablas base y .Peer que crea symfony por defecto para comunicarse con la base de datos externa. (Ver Fig. 11)

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA



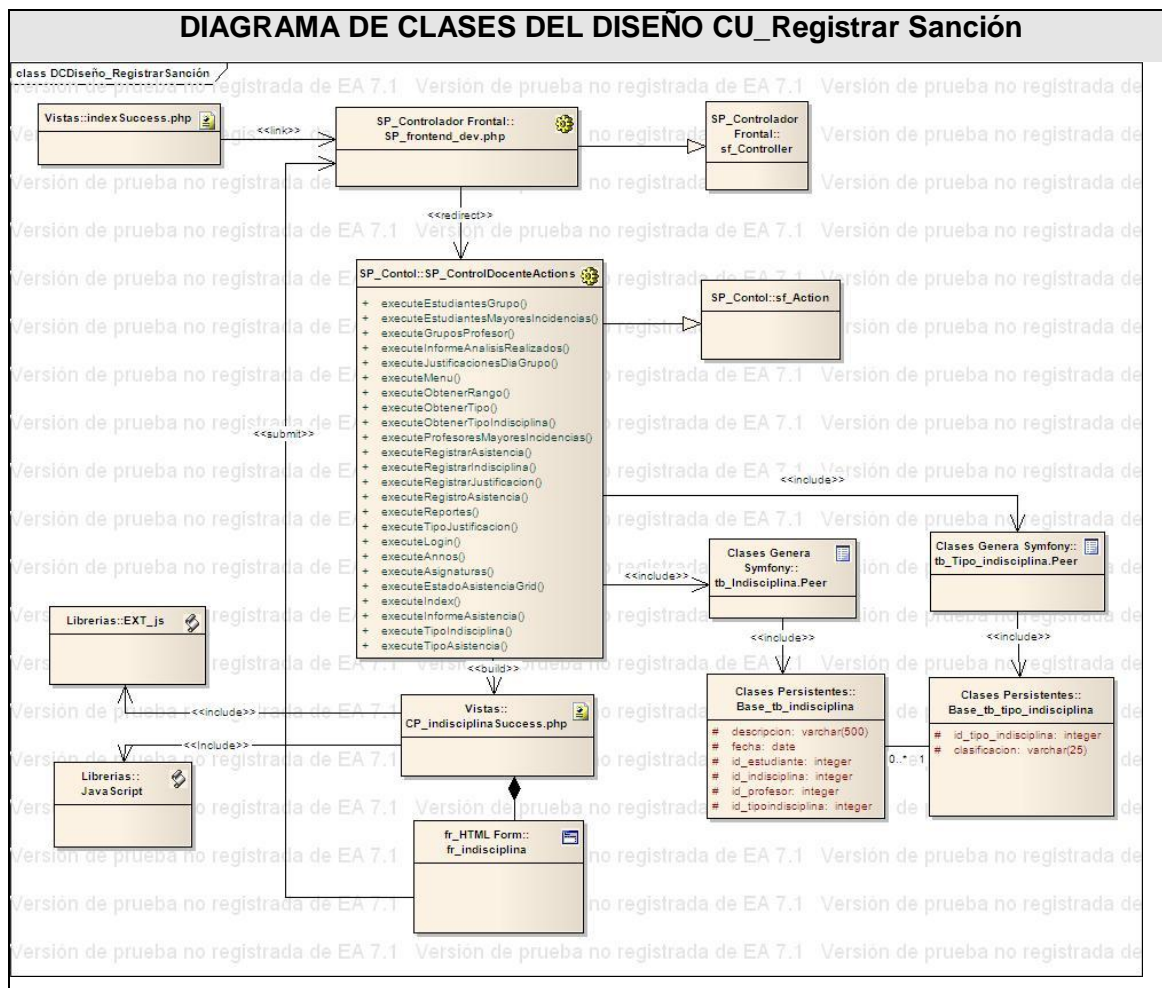
**Fig. 11 Diagrama de Clases del Diseño del Caso de Uso, Gestionar Asistencia. Paquete Gestión de Información.**

### Diagrama de clases del diseño del caso de uso: Registrar Sanción

En el diagrama se tiene la clase principal `indexSuccess.php` que es la página que se le muestra al usuario de acuerdo al cargo que ocupa una vez autenticado. Esta le hace un link al `SP_frontend_dev.php`. Luego redirecciona la acción al `SP_ControlDocenteActions` que contiene todas las operaciones que se realizan en el módulo de Control Docente. La `SP_ControlDocenteActions` es la encargada de construir la `CP_indisciplinaSuccess.php` que está compuesta por un formulario para la entrada de los datos.

Además es la responsable de hacer las consultas a la base de dato por medio de las tablas base y .Peer que crea symfony por defecto para comunicarse con la base de dato externa. (Ver Fig. 12)

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA



**Fig. 12 Diagrama de Clases del Diseño del Caso de Uso, Registrar Sanción. Paquete Gestión de Información.**

### Diagramas de interacción

Entre los artefactos que brinda UML para expresar las iteraciones entre los objetos dándole cumplimiento a los requerimientos del sistema, se encuentran los Diagramas de Interacción. Estos se clasifican en dos tipos: Diagramas de Secuencia y Diagramas de Colaboración. Permiten modelar aspectos dinámicos del sistema. Se utilizan para realizar una traza de la ejecución de un escenario y a cada uno de estos le corresponde un diagrama de interacción.

Una interacción es un conjunto de objetos y sus relaciones, incluye los mensajes ordenados temporalmente, mediante los cuales pueden establecer comunicación. Un diagrama de colaboración destaca la organización estructural de los objetos que envían y reciben mensajes.

- ◆ Un diagrama de interacción contiene:
  - ✓ Objetos.
  - ✓ Enlaces.
  - ✓ Mensajes.

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

- ◆ Un mensaje de interacción puede contener:
  - ✓ Notas.
  - ✓ Restricciones.

### Diagrama de Secuencia

El diagrama de secuencia es uno de los diagramas más efectivos para modelar interacción entre objetos en un sistema. Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada método de la clase. Mientras que el diagrama de casos de uso permite el modelado de una vista del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluye los objetos y clases que se usan para implementar el escenario, y mensajes pasados entre los objetos.

Los diagramas de secuencias se usan continuamente como modelos explicativos para los escenarios de los casos de uso. Muestra los objetos que intervienen en el escenario con líneas discontinuas verticales, y los mensajes pasados entre los objetos como flechas horizontales. Existen dos tipos de mensajes: síncronos y asíncronos.

- ◆ Los mensajes síncronos se corresponden con llamadas a métodos del objeto que recibe el mensaje. El objeto que envía el mensaje queda bloqueado hasta que termina la llamada.
- ◆ Los mensajes asíncronos terminan inmediatamente, y crean un nuevo hilo de ejecución dentro de la secuencia.

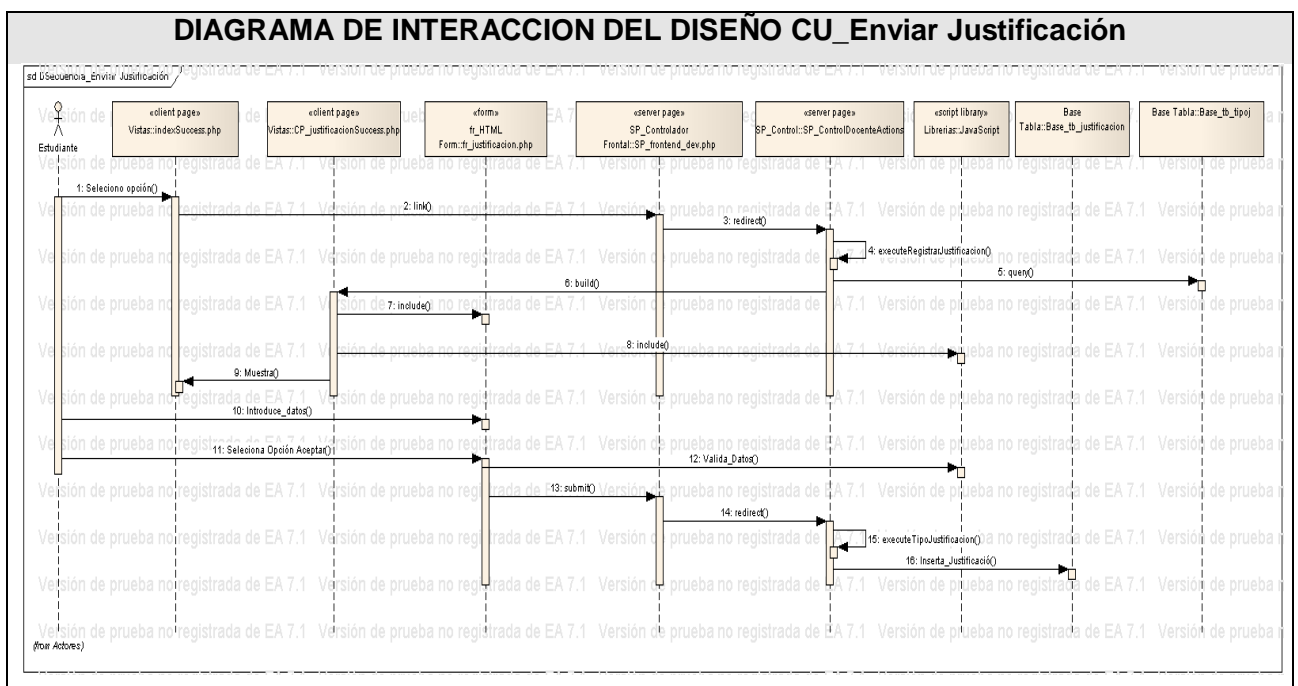


Fig. 13 Diagrama de Interacción del Diseño CU Enviar Justificación.

# CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

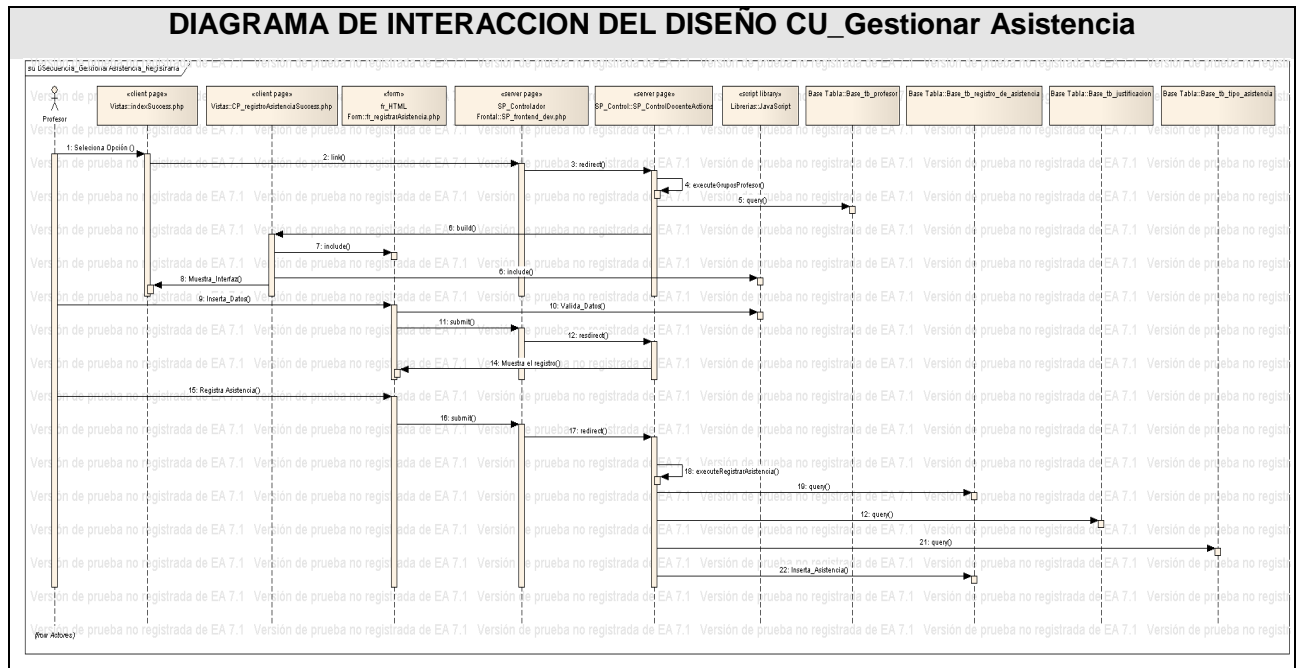


Fig. 14 Diagrama de Interacción del Diseño CU Gestionar Asistencia. Registrar.

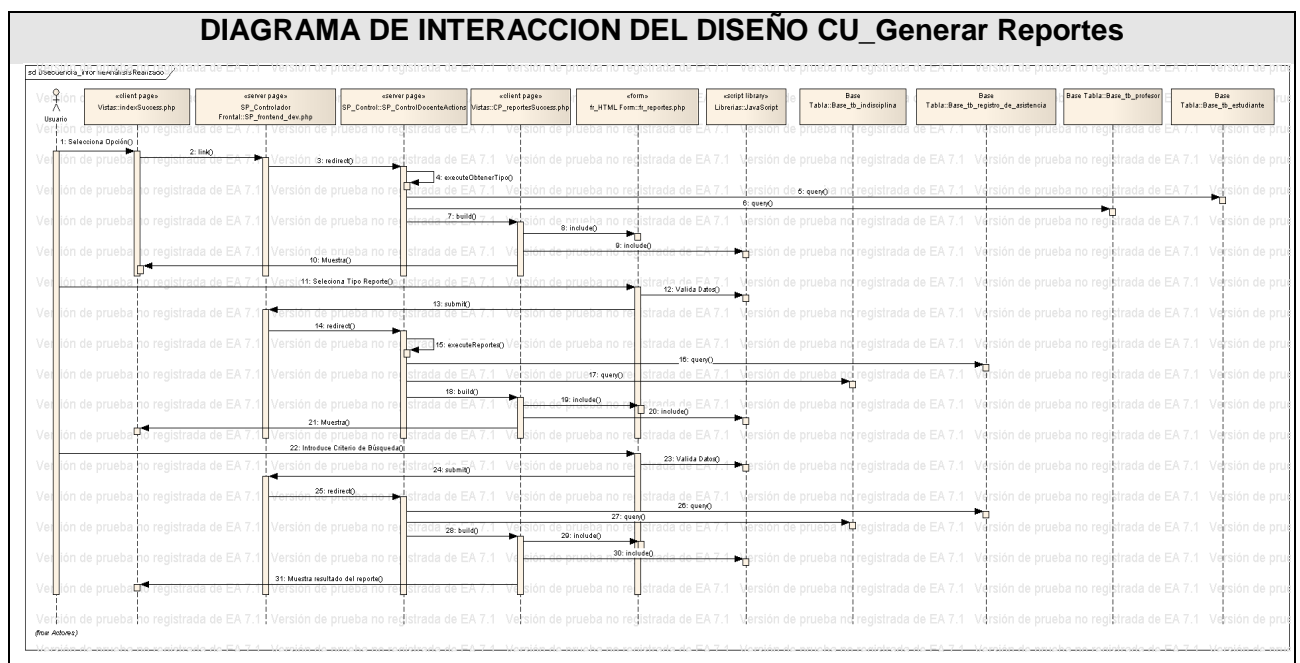


Fig. 15 Diagrama de Interacción del Diseño CU Generar Reportes.

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

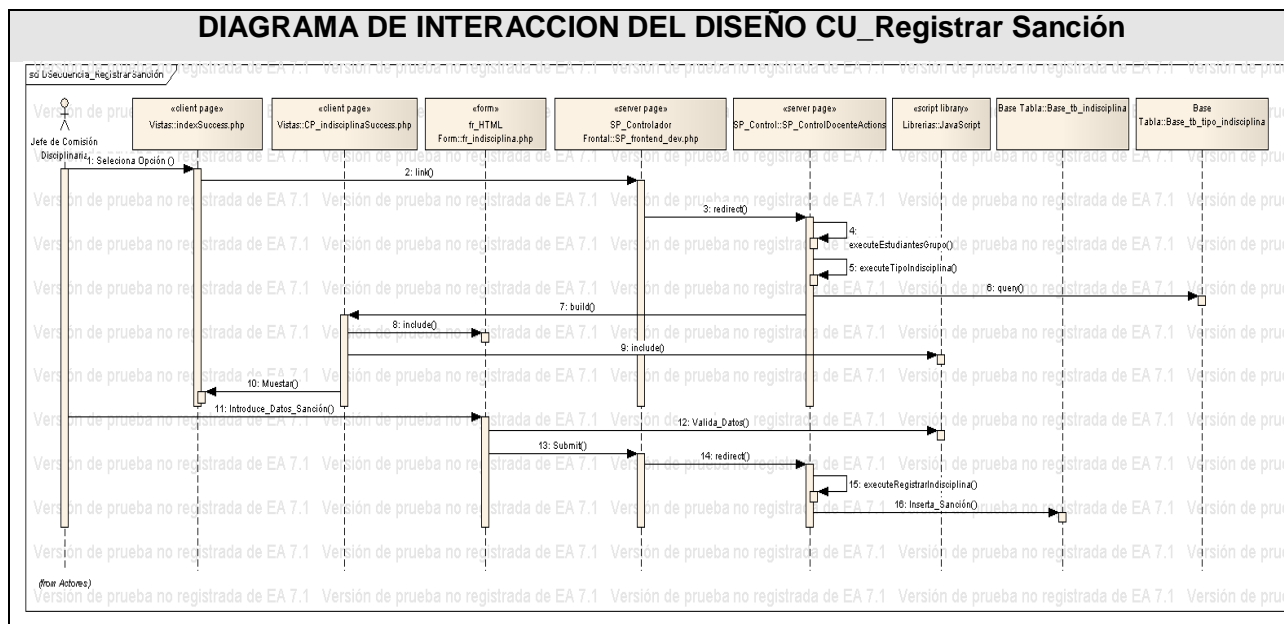


Fig. 16 Diagrama de Interacción del Diseño CU Registrar Sanción.

### Descripción de las clases del diseño

<b>Nombre:</b> frontend_dev	
<b>Tipo de clase:</b> Controladora	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	Controlar.
<b>Descripción:</b>	Contiene el código que liga la lógica de negocio con la presentación. Es el único punto de entrada a la aplicación. Carga la configuración y determina la acción a ejecutarse.

Tabla 16. Descripción de la clase del diseño, frontend\_dev.

<b>Nombre:</b> ControlDocenteActions	
<b>Tipo de clase:</b> Controladora	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	Acción
<b>Descripción:</b>	Es la clase que almacena todas las acciones del módulo.

Tabla 17. Descripción de la clase del diseño, contDoc.



## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

---

### Conclusiones

En este capítulo se obtuvo el Modelo de Análisis del sistema propuesto. Se realizaron diferentes artefactos mediante los cuales es posible adquirir una visión detallada de las responsabilidades de cada Clase del Análisis para el correcto funcionamiento del sistema a desarrollar. Los artefactos desarrollados fueron los Diagramas de Clases del Análisis y la Descripción de las Clases.

Además se obtuvo el Modelo de Diseño. Los artefactos generados durante la modelación del diseño, definen la entrada, a las actividades que se realizan en el Flujo de Trabajo de Implementación, estos son los Diagramas de Clases del Diseño, la Descripción de las Clases y los Diagramas de Secuencia del Diseño.



### **Capítulo 4: Implementación**

El Modelo de Implementación comprende un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes se encuentran datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos. (26)

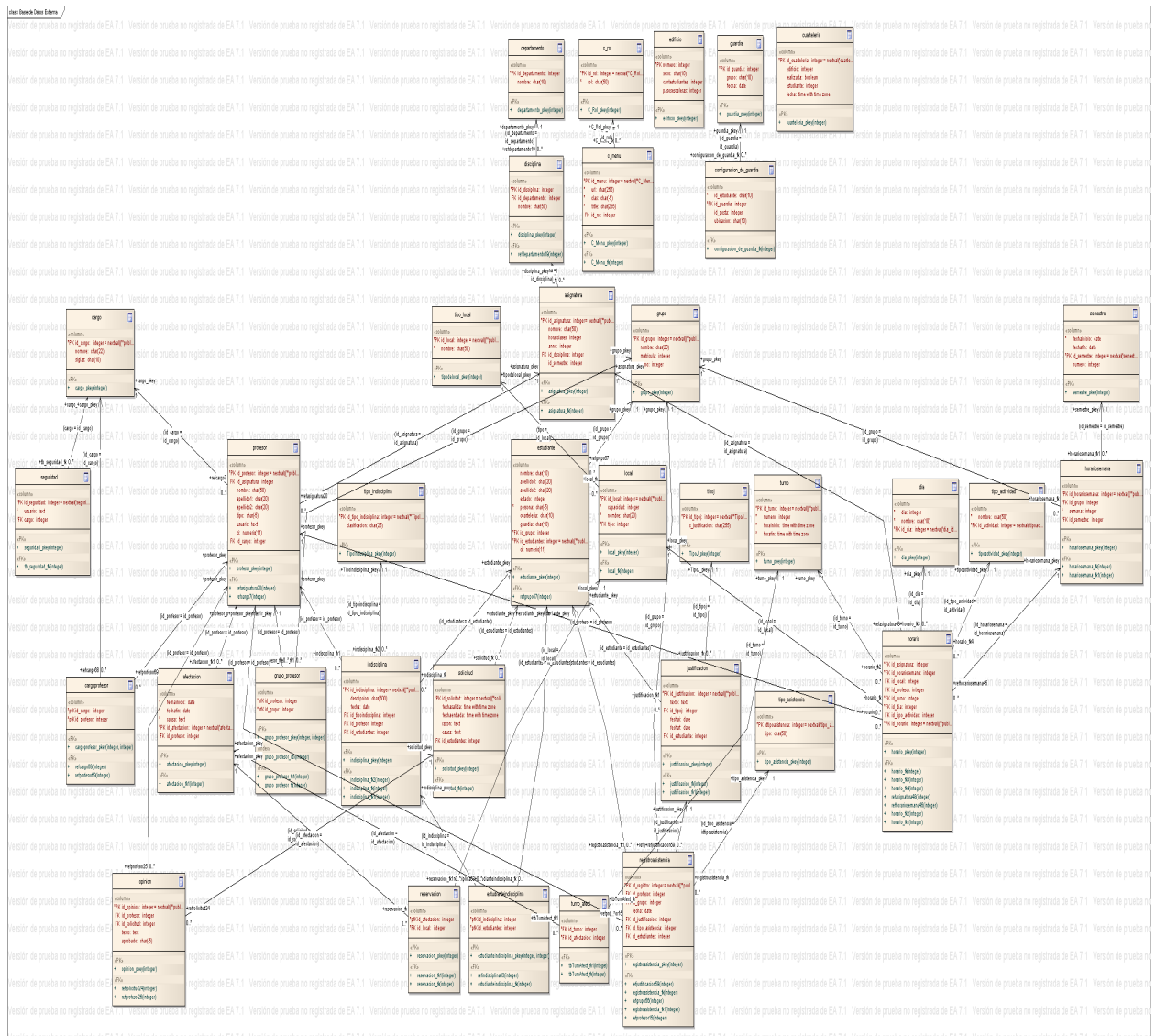
El presente capítulo constituye la secuencia lógica del Diseño del Sistema, a través del mismo se implementa las clases y subsistemas encontrados durante el Flujo de Trabajo anterior en términos de componentes. Se describen las tablas del modelo de datos que se realizó específicamente para posibilitar la obtención de los Reportes Estadísticos. Como parte del Modelo de Implementación, se obtiene además el Diagrama de Despliegue, que indica cómo la solución implementada esta distribuida físicamente.

#### **Modelo de datos**

Los modelos de datos aportan la base conceptual para diseñar aplicaciones que hacen un uso intensivo de datos, así como la base formal para las herramientas y técnicas empleadas en el desarrollo y uso de sistemas de información. Un modelo de datos no es más que un conjunto de conceptos, reglas y convenciones que permiten describir y manipular determinados datos que se desean almacenar en una base de datos. Al producto del modelo de datos se le llama esquema (descripción de la estructura de la base de datos) y a los datos almacenados en ese momento, ocurrencia del esquema.

## CAPÍTULO 4: IMPLEMENTACIÓN

## Modelo de dato lógico



**Fig. 17 Modelo de dato lógico.**

El modelo de datos lógico representa todas las tablas de la Base de Datos externa de la aplicación, que es común para un grupo de aplicaciones. A continuación se describen las tablas que se utilizan para el desarrollo del sistema propuesto.

### Descripción de las tablas

<b>Nombre: asignatura</b>		
<b>Descripción:</b> Esta tabla contiene los atributos de las diferentes asignaturas así como el semestre y el año al que pertenece cada una.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_asignatura	INTEGER	Representa el identificador de la asignatura.

## CAPÍTULO 4: IMPLEMENTACIÓN

id_disciplina	INTEGER	Representa el identificador de la disciplina.
id_semestre	INTEGER	Representa el identificador del semestre al cual pertenece la asignatura.
nombre	VARCHAR(50)	Representa el nombre de la asignatura.
horasclases	INTEGER	Representa la cantidad de horas clases que tiene la asignatura.
anno	INTEGER	Representa el año al cual se le imparte la asignatura.

Tabla 18. Descripción de la tabla asignatura.

Nombre: cargo		
<b>Descripción:</b> Esta tabla contiene los atributos de los diferentes cargos que puede ocupar un profesor determinado en la facultad.		
Atributo	Tipo	Descripción
id_cargo	INTEGER	Representa el identificador del cargo.
nombre	CHAR(10)	Representa el nombre del cargo.
siglas	VARCHAR(10)	Representa las siglas por las cuales se identifica el cargo.

Tabla 19. Descripción de la tabla cargo.

Nombre: cargoprofesor		
<b>Descripción:</b> Esta tabla contiene los atributos de identificación de las tablas de profesor y de cargo.		
Atributo	Tipo	Descripción
id_cargo	INTEGER	Representa el identificador del cargo.
id_profesor	INTEGER	Representa el identificador del profesor.

Tabla 20. Descripción de la tabla cargoprofesor.

Nombre: departamento		
<b>Descripción:</b> Esta tabla contiene todos los atributos de los diferentes departamentos que tiene la facultad.		
Atributo	Tipo	Descripción
id_departamento	INTEGER	Representa el identificador del departamento.
nombre	CHAR(10)	Representa el nombre que tiene el departamento.

Tabla 21. Descripción de la tabla departamento.

## CAPÍTULO 4: IMPLEMENTACIÓN

Nombre: disciplina		
<b>Descripción:</b> Esta tabla contiene todos los atributos de las disciplinas que presentan la carrera, así como el departamento al que pertenece.		
Atributo	Tipo	Descripción
id_disciplina	INTEGER	Representa el identificador de la disciplina.
id_departamento	INTEGER	Representa el identificador del departamento al que pertenece la disciplina.
nombre	VARCHAR(50)	Representa el nombre de la disciplina.

Tabla 22. Descripción de la tabla Disciplina.

Nombre: estudiante		
<b>Descripción:</b> Esta tabla contiene todos los atributos de cada estudiante de la facultad.		
Atributo	Tipo	Descripción
id_estudiante	INTEGER	Representa el identificador del estudiante.
nombre	CHAR(10)	Representa el nombre del estudiante.
apellido 1	VARCHAR (20)	Representa el primer apellido del estudiante.
apellido 2	VARCHAR (20)	Representa el segundo apellido del estudiante.
id_grupo	INTEGER	Representa el identificador del grupo al que pertenece el estudiante.
ci	NUMERIC(11,0)	Representa el carnet de identidad del estudiante.

Tabla 23. Descripción de la tabla estudiante.

Nombre: estudianteindisciplina		
<b>Descripción:</b> Esta tabla contiene los atributos identificación de las tablas de estudiante e indisciplina.		
Atributo	Tipo	Descripción
id_estudiante	INTEGER	Representa el identificador del estudiante.
id_indisciplina	INTEGER	Representa el identificador de la indisciplina.

Tabla 24. Descripción de la tabla estudianteindisciplina.

Nombre: grupo		
<b>Descripción:</b> Esta tabla contiene todos los atributos de cada uno de los grupos de la facultad, así como el año al que pertenece.		
Atributo	Tipo	Descripción

## CAPÍTULO 4: IMPLEMENTACIÓN

id_grupo	INTEGER	Representa el identificador del grupo.
nombre	VARCHAR (20)	Representa el nombre del grupo.
matrícula	INTEGER	Representa la cantidad de estudiantes que tiene el grupo.
anno	INTEGER	Representa el año que cursa el grupo.

Tabla 25. Descripción de la tabla grupo.

Nombre: grupoprofesor		
<b>Descripción:</b> Esta tabla contiene los atributos identificación de las tablas grupo y profesor.		
Atributo	Tipo	Descripción
id_grupo	INTEGER	Representa el identificador del grupo.
id_profesor	INTEGER	Representa el identificador del profesor.

Tabla 26. Descripción de la tabla grupoprofesor.

Nombre: indisciplina		
<b>Descripción:</b> Esta tabla contiene los atributos de las indisciplinas que comenten los estudiantes y que se archivan.		
Atributo	Tipo	Descripción
id_indisciplina	INTEGER	Representa el identificador de la Indisciplina.
descripcion	VARCHAR(500)	Representa la medida tomada.
fecha	DATE	Representa la fecha de la justificación.
id_estudiante	INTEGER	Representa el identificador del estudiante.
id_tipoindisciplina	INTEGER	Representa el identificador del tipo de indisciplina.
id_profesor	INTEGER	Representa el identificador del profesor.

Tabla 27. Descripción de la tabla indisciplina.

Nombre: justificacion		
<b>Descripción:</b> Esta tabla contiene todos los atributos de las justificaciones que envían los estudiantes a sus profesores.		
Atributo	Tipo	Descripción
id_justificacion	INTEGER	Representa el identificador de la justificación.
texto	TEXT	Representa la justificación que dio el estudiante para no asistir a clases.
id_tipoj	INTEGER	Representa el identificador del tipo de

## CAPÍTULO 4: IMPLEMENTACIÓN

		justificación.
fechai	DATE	Representa la fecha de inicio de la inasistencia que tiene el estudiante, en caso de que falte varios días seguidos por una misma causa.
fechaf	DATE	Representa la fecha de fin de la inasistencia que tiene el estudiante, en caso de que falte varios días seguidos por una misma causa.
id_estudiante	INTEGER	Representa el identificador del estudiante que está enviando la justificación.

Tabla 28. Descripción de la tabla justificación.

Nombre: profesor		
<b>Descripción:</b> Esta tabla contiene todos los atributos de cada uno de los profesores de la facultad, así como la asignatura y el cargo que ocupa.		
Atributo	Tipo	Descripción
id_profesor	INTEGER	Representa el identificador del profesor.
id_asignatura	INTEGER	Representa el identificador de la asignatura que imparte el profesor.
nombre	VARCHAR(50)	Representa el nombre del profesor.
apellido 1	VARCHAR(20)	Representa el primer apellido del profesor.
apellido 2	VARCHAR(20)	Representa el segundo apellido del profesor.
usuario	TEXT	Representa el usuario del profesor para poder comunicarse con el por correo o jabber.
ci	NUMERIC(11,0)	Representa el carnet de identidad del profesor.

Tabla 29. Descripción de la tabla profesor.

Nombre: registroasistencia		
<b>Descripción:</b> Esta tabla contiene los atributos del registro de asistencia que utiliza cada profesor para registrar la asistencia de cada uno de los grupos a los que le imparte clase.		
Atributo	Tipo	Descripción
id_registro	INTEGER	Representa el identificador del registro.
id_profesor	INTEGER	Representa el identificador del profesor que tiene el registro.
id_grupo	INTEGER	Representa el identificador del grupo al que pertenece el registro.

## CAPÍTULO 4: IMPLEMENTACIÓN

id_estudiante	INTEGER	Representa el identificador de los estudiantes que pertenecen al grupo de ese registro.
fecha	DATE	Representa la fecha de cada uno de los turnos en los que se registra la asistencia.
id_tipo_asistencia	INTEGER	Representa el identificador del tipo de asistencia que es.
id_justificacion	INTEGER	Representa el identificador de la justificación enviada por el estudiante.

Tabla 30. Descripción de la tabla registroasistencia.

<b>Nombre: menu</b>		
<b>Descripción:</b> Esta tabla contiene los atributos del menú que salen en cada página con las opciones de acuerdo al cardo que ocupa cada usuario.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_menu	INTEGER	Representa el identificador del menú.
url	VARCHAR(255)	Representa la dirección donde se encuentra la página que se mostrara.
clas	VARCHAR	Representa la clase.
title	VARCHAR(255)	Representa el titulo.
id_rol	INTEGER	Representa el identificador del cargo que ocupa el usuario.

Tabla 31. Descripción de la tabla menu.

<b>Nombre: tipo_asistencia</b>		
<b>Descripción:</b> Esta tabla contiene los atributos del tipo de asistencia que es registrada en el registro de asistencia en el momento que el profesor recoja la asistencia.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
idtipoasistencia	INTEGER	Representa el identificador del tipo de asistencia.
tipo	VARCHAR(50)	Representa la clasificación del tipo de asistencia.

Tabla 32. Descripción de la tabla tipo\_asistencia.

<b>Nombre: tipo_indisciplina</b>		
<b>Descripción:</b> Esta tabla contiene los atributos del tipo de indisciplina que comete un estudiante.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_tipo_indisciplina	INTEGER	Representa el identificador del tipo de indisciplina.

## CAPÍTULO 4: IMPLEMENTACIÓN

clasificacion	VARCHAR (25)	Representa la clasificación del tipo de indisciplina cometida por el estudiante.
---------------	--------------	--

Tabla 33. Descripción de la tabla tipo\_indisciplina.

<b>Nombre: tipoj</b>		
<b>Descripción:</b> Esta tabla contiene los atributos del tipo de justificación que va a enviar el estudiante al profesor cuando no asista a clases.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_tipoj	INTEGER	Representa el identificador del tipo de justificación.
c_justificacion	VARCHAR(255)	Representa el tipo de justificación.

Tabla 34. Descripción de la tabla tipoj.

### Diagrama de componentes

Un diagrama de componentes representa la separación de un sistema de software en componentes físicos (archivos, cabeceras, módulos, paquetes) y muestra las dependencias entre estos componentes. Estos son utilizados para modelar la vista estática de un sistema. Muestra la organización y las dependencias entre un conjunto de componentes. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema.

En estos apartados se sitúan librerías, tablas archivos, ejecutables y documentos que forman parte del sistema. Uno de los usos principales es que puede servir para ver cuales componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema. Lo que distingue a un diagrama de componentes de otros tipos de diagramas es su contenido. Normalmente contienen componentes, interfaces y relaciones entre ellos. Y como todos los diagramas, también puede contener paquetes utilizados para agrupar elementos del modelo.

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean éstos componentes de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo.

Dado que los diagramas de componentes muestran los componentes software que constituyen una parte reusable, sus interfaces, y sus interrelaciones, en muchos aspectos se puede considerar que



## CAPÍTULO 4: IMPLEMENTACIÓN

---

un diagrama de componentes es un diagrama de clases a gran escala. Cada componente en el diagrama debe ser documentado con un diagrama de componentes más detallado, un diagrama de clases o un diagrama de casos de uso.

Un paquete en un diagrama de componentes representa una división física del sistema. Los paquetes se organizan en una jerarquía de capas donde cada capa tiene una interfaz bien definida. Un ejemplo típico de una jerarquía en capas de este tipo es: Interfaz de usuario; Paquetes específicos de la aplicación; Paquetes reusables; Mecanismos claves; y Paquetes hardware y del sistema operativo. Un diagrama de componentes se representa como un grafo de componentes software unidos por medio de relaciones de dependencia (generalmente de compilación). Puede mostrar también que un componente software contiene una interfaz, es decir, la soporta. (27)

La estructura de paquetes del diagrama de componentes de la Aplicación Web para el Control de la Asistencia a Clases en la Facultad 7 queda de la siguiente manera: (Ver Fig. 18)

- ◆ Un paquete Vista donde se encuentran todas las bibliotecas utilizadas para el diseño, el layout de las páginas y el paquete con el nombre del módulo, que contiene todas las interfaces y formularios utilizados en la aplicación. (Ver Fig. 20)
- ◆ Un paquete Controlador en el que se encuentra el `forntend_dev` que no es más que el controlador frontal de la aplicación, único punto de entrada a la misma y un paquete con el nombre del módulo que contiene la clase `ControlDocenteActions` que es la tiene todas las acciones a realizar en la aplicación. (Ver Fig. 21)
- ◆ Un paquete Modelo en el que se encuentran las tablas Base y `.Peer` que genera symfony por defecto para acceder a los datos de la base de dato. (Ver Fig. 19)

## CAPÍTULO 4: IMPLEMENTACIÓN

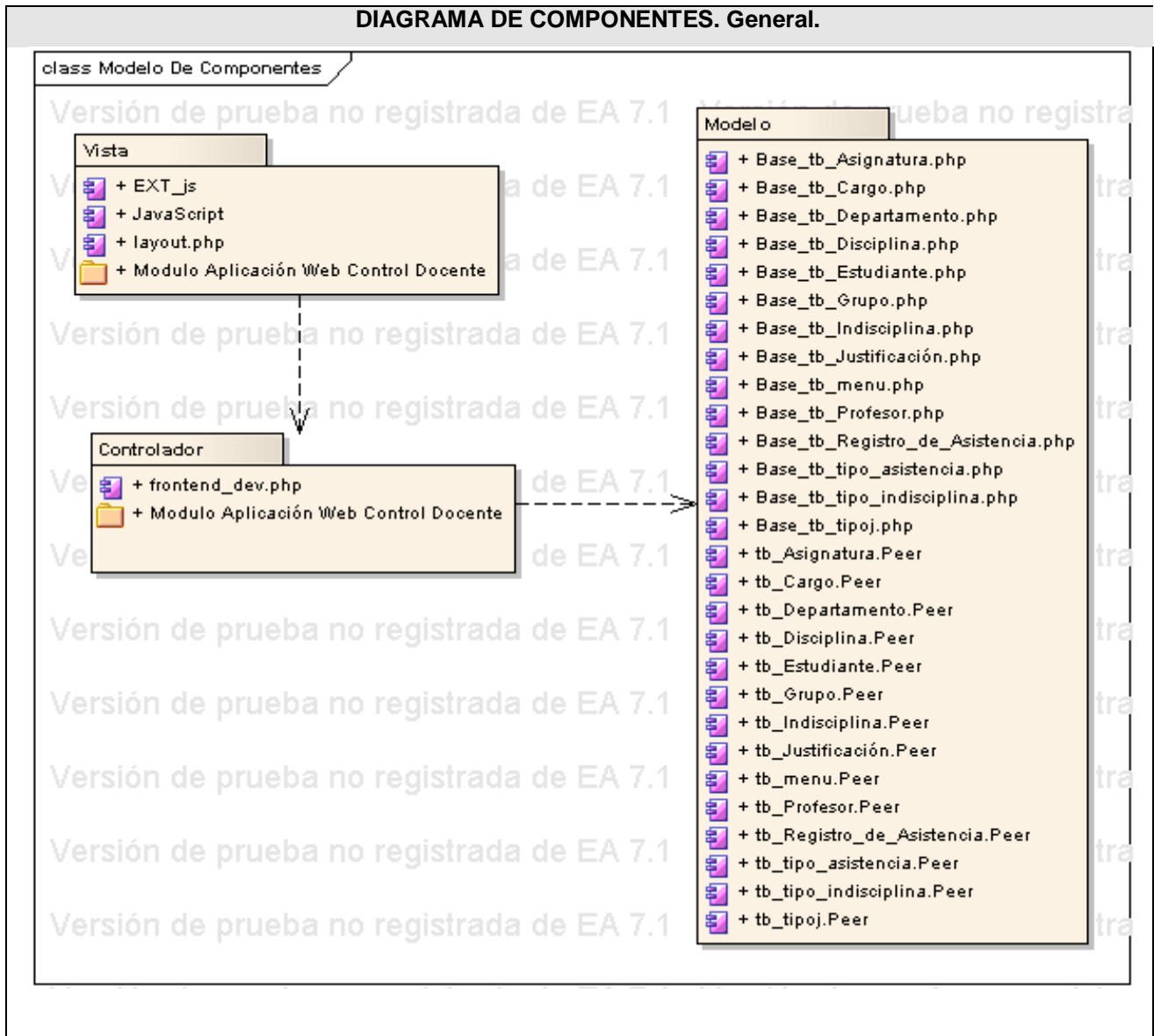


Fig. 18 Diagrama de Componentes General.

## CAPÍTULO 4: IMPLEMENTACIÓN

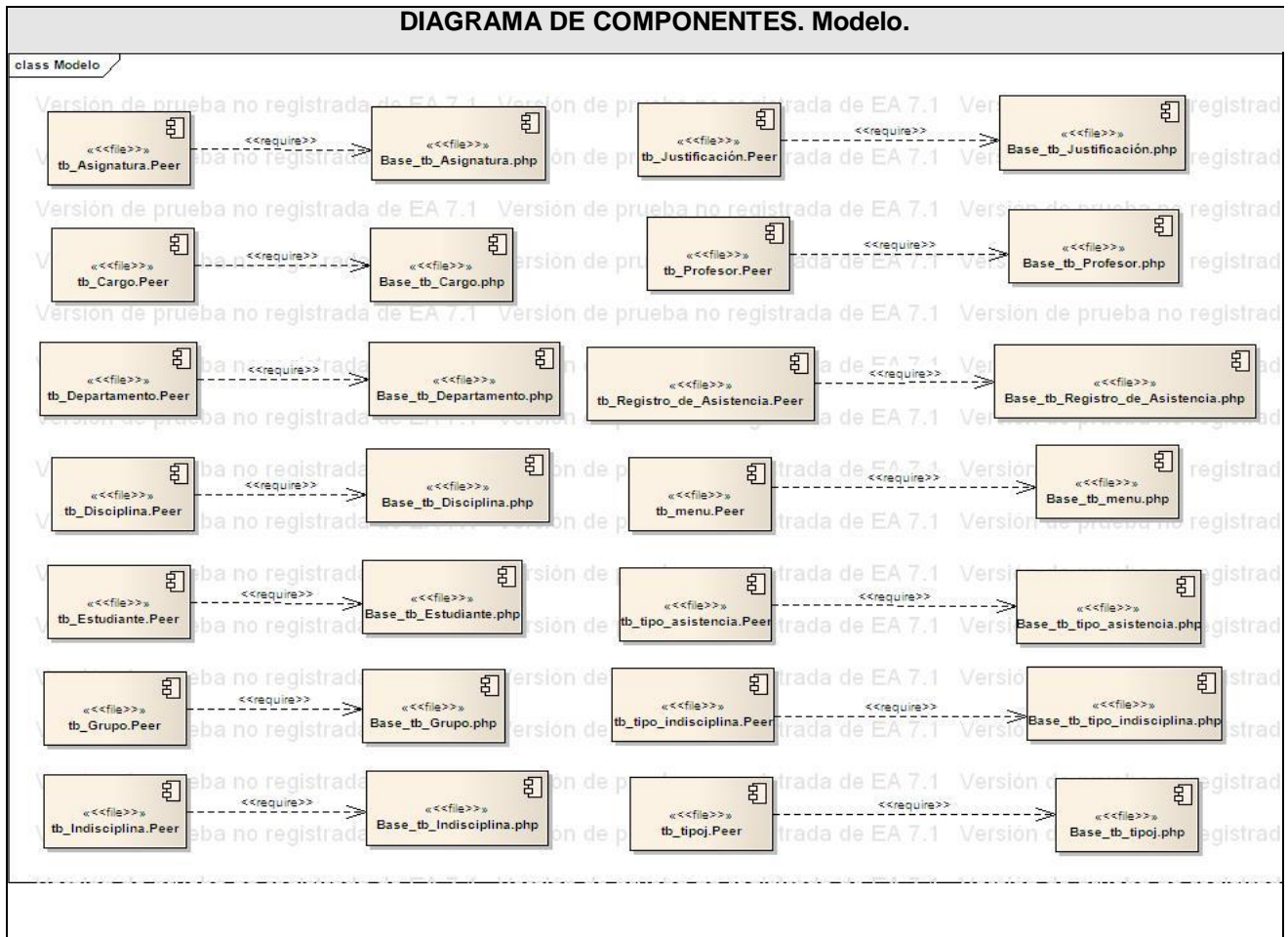


Fig. 19 Diagrama de Componentes del Modelo.

## CAPÍTULO 4: IMPLEMENTACIÓN

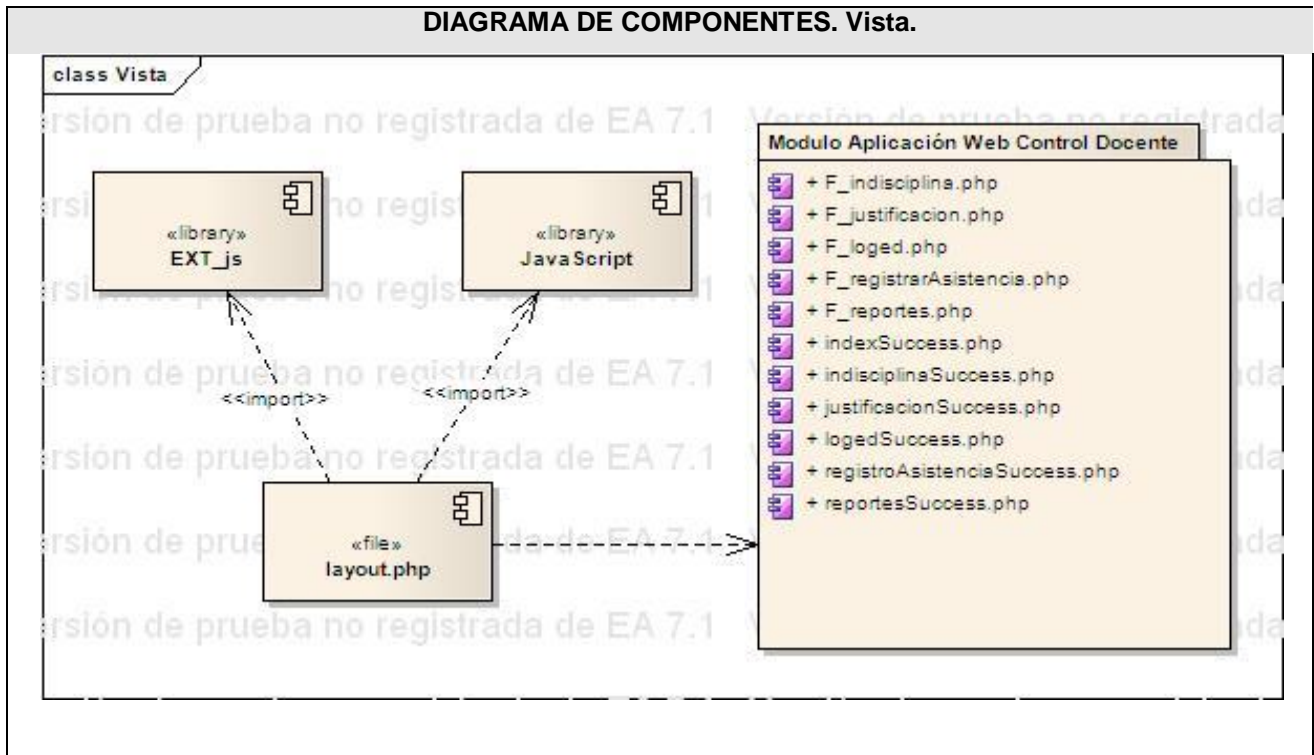


Fig. 20 Diagrama de Componentes de la Vista.



Fig. 21 Diagrama de Componentes del Controlador.

### Diagrama de despliegue

El Modelo Físico de Despliegue, provee un modelo detallado de la forma en la que los componentes se despliegan a lo largo de la infraestructura del sistema. En este se detallan las capacidades de red, las especificaciones del servidor, los requisitos de hardware y otra información relacionada al despliegue del sistema propuesto. Un diagrama de despliegue, muestra las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos).

## CAPÍTULO 4: IMPLEMENTACIÓN

Además, se puede ver, como es un grafo de nodos unidos por conexiones de comunicación. Un nodo, puede contener instancias de componentes software, objetos, procesos, etc. Esta es una unidad de computación de algún tipo. Las instancias de componentes software pueden estar unidas por relaciones de dependencia.

Para la estructura de la posible distribución del sistema se utilizó un nodo PC\_Cliente que representa la computadora del usuario. Esta se conecta mediante el protocolo de Transferencia de Hipertexto (http por sus siglas en inglés, HyperText Transfer Protocol) al nodo Servidor WEB y de Aplicación. Desde el servidor se puede acceder al nodo Servidor de BD mediante el protocolo diseñado para facilitar la reutilización de código de bases de datos, la capa de abstracción de bases de datos (PDO, PHP Data Objects, por sus siglas en inglés).

Se consumen los servicios del Servidor de Aplicación Registro Identidad UCI y el Servidor de Aplicación Akademos, mediante el Protocolo Simple de Acceso a Objetos (SOAP por sus siglas en inglés, Simple Object Access Protocol), este define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. Además se tiene una impresora a la que se accede mediante un puerto Bus Universal en Serie (USB por sus siglas en inglés, Universal Serial Bus) o Conductor Universal en Serie (CUS), abreviado comúnmente USB, es un puerto que sirve para conectar periféricos a una computadora. (Ver Fig. 22)

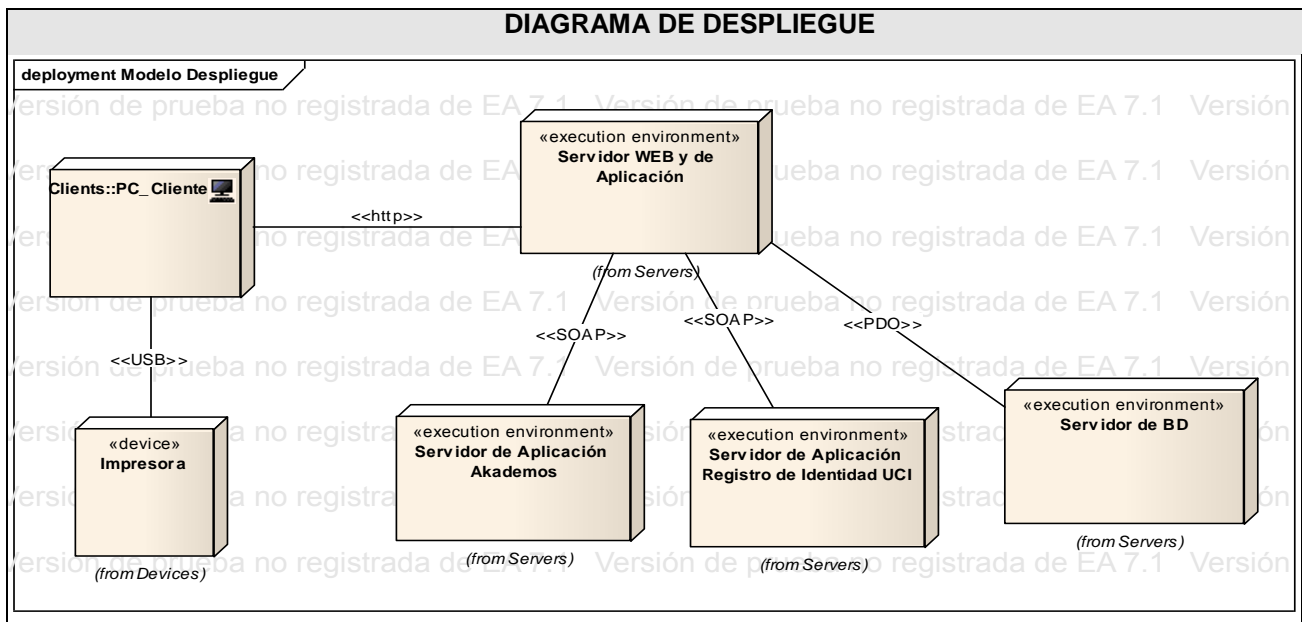


Fig. 22 Diagrama de Despliegue del Sistema Control Docente.

## CAPÍTULO 4: IMPLEMENTACIÓN

---

### Tratamiento de errores

Desde los inicios de los lenguajes de programación, la gestión de errores es uno de los asuntos más difíciles. Es tan complicado diseñar un buen esquema de gestión de errores, que muchos lenguajes simplemente lo ignoran, delegando el problema en los diseñadores de la librería, que lo resuelven a medias, de forma que puede funcionar en muchas situaciones, pero se pueden eludir, normalmente ignorándolos.

En la validación de los datos de la acción, los parámetros de la petición son una tarea repetitiva y tediosa. Symfony incluye un sistema de validación, utilizando métodos de la clase acción. Se ve, un ejemplo: cuando un usuario hace una petición a la Acción, Symfony siempre busca primero un método llamado `validateMiAccion ()`. Si lo encuentra, Symfony ejecuta ese método. El valor de retorno de esta validación, determina el siguiente método que se ejecuta: si devuelve `True`, entonces se ejecuta el método `executeMiAccion ()`; en otro caso, se ejecuta `handleErrorMiAccion ()`. En el caso de que `handleErrorMiAccion ()` no exista, Symfony busca un método genérico llamado `handleError ()`. Si tampoco existe, simplemente devuelve el valor `sfView: ERROR` para producir la plantilla `miAccionError.php`. (28)

La clave para un correcto funcionamiento de la validación es respetar la convención de nombres para los métodos de la acción:

- ◆ **`validateNombreAccion`** es el método de validación, que devuelve `true` o `false`. Se trata del primer método buscado cuando se solicita la acción `NombreAccion`. Si no existe, la acción se ejecuta directamente.
- ◆ **`handleErrorNombreAccion`** es el método llamado cuando el método de validación falla. Si no existe, entonces se muestra la plantilla `Error`.
- ◆ **`executeNombreAccion`** es el método de la acción. Debe existir para todas las acciones.

Se puede incluir cualquier código en el método `validate ()`. La única condición es que devuelva un valor `true` o `false`. Como es un método de la clase `sfActions`, tiene acceso a los objetos `sfRequest` y `sfUser`, que pueden ser realmente útiles para validación de los datos de la petición y del contexto. Se pueden utilizar este mecanismo para implementar la validación de los formularios (esto es, controlar los valores introducidos por el usuario en un formulario antes de procesarlo), pero se trata de una tarea muy repetitiva para la que Symfony proporciona herramientas automatizadas.

## CAPÍTULO 4: IMPLEMENTACIÓN

---

### Seguridad

La posibilidad de ejecutar una acción puede ser restringida a usuarios con ciertos privilegios. Las herramientas proporcionadas por Symfony para este propósito permiten la creación de aplicaciones seguras, en las que los usuarios necesitan estar autenticados antes de acceder a alguna característica o partes de la aplicación. Añadir esta seguridad a una aplicación requeridos pasos: declarar los requerimientos de seguridad para cada acción y autenticar a los usuarios con privilegios para que puedan acceder estas acciones seguras.

### Restricción de Acceso

Antes de ser ejecutada, cada acción pasa por un filtro especial que verifica si el usuario actual tiene privilegios de acceder a la acción requerida. En Symfony, los privilegios están compuestos por dos partes:

- ◆ Las acciones seguras requieren que los usuarios estén autenticados.
- ◆ Las credenciales son privilegios de seguridad agrupados bajo un nombre y que permiten organizar la seguridad en grupos. Para restringir el acceso a una acción se crea y se edita un archivo de configuración YAML llamado `security.yml` en el directorio `config/` del módulo. En este archivo, se pueden especificar los requerimientos de seguridad que los usuarios deben satisfacer para cada acción o para todas las acciones.

Las acciones no incluyen restricciones de seguridad por defecto, así que cuando no existe el archivo `security.yml` o no se indica ninguna acción en ese archivo, todas las acciones son accesibles por todos los usuarios. Si existe un archivo `security.yml`, Symfony busca por el nombre de la acción y si existe, verifica que se satisfagan los requerimientos de seguridad. Lo que sucede cuando un usuario trata de acceder una acción restringida depende de sus credenciales:

- ◆ Si el usuario está autenticado y tiene las credenciales apropiadas, entonces la acción se ejecuta.
- ◆ Si el usuario no está autenticado, es redireccionado a la acción de login.
- ◆ Si el usuario está autenticado, pero no posee las credenciales apropiadas, se rige a la acción segura por defecto.

Las páginas login y secure son bastante simples, por lo que seguramente es necesario personalizarlas. Se puede configurar qué acciones se ejecutan en caso de no disponer de suficientes privilegios en el archivo `settings.yml` de la aplicación cambiando el valor de las propiedades.



## CAPÍTULO 4: IMPLEMENTACIÓN

---

### Otorgando Acceso

Para obtener acceso a áreas restringidas, los usuarios necesitan estar autenticados y/o poseer ciertas credenciales. Extiende los privilegios del usuario mediante llamadas a métodos del objeto `sfUser`. El estado autenticado se establece con el método `setAuthenticated()` y se puede comprobar con el método `isAuthenticated()`.

Las credenciales son un poco más complejas de tratar, ya que se pueden verificar, agregar, quitar y borrar. Si el usuario tiene la credencial "parámetro", entonces ese usuario puede acceder a las acciones para las cuales el archivo `security.yml` requiere esa credencial. Las credenciales se pueden utilizar también para mostrar contenido autenticado en una plantilla. Y para el estado de autenticación, las credenciales normalmente se dan a los usuarios durante el proceso de login. Este es el motivo por el que el objeto `sfUser` normalmente se extiende para añadir métodos de login y de logout, de forma que se pueda establecer el estado de seguridad del usuario de forma centralizada.

### Credenciales Complejas

La sintaxis YAML utilizada en el archivo `security.yml` permite restringir el acceso a usuarios que tienen una combinación de credenciales, usando asociaciones de tipo AND y OR. Con estas combinaciones, se pueden definir flujos de trabajo y sistemas de manejo de privilegios muy complejos como por ejemplo, un sistema de gestión de contenidos (CMS) cuya parte de gestión sea accesible solo a usuarios con credencial `admin`, donde los artículos pueden ser editados solo por usuarios con credenciales de `editor` y publicados solo por aquellos que tienen credencial de `Publisher`.

### Estrategias de codificación. Estándares y estilos a utilizar

Crear un estándar de codificación, tiene como objetivo, que los desarrolladores se rijan por el mismo a la hora de implementar. Este estándar sirve, para identificar de forma sencilla el objetivo y las funcionalidades que brinda cada una de las clases, funciones y demás componentes de Software dada su nomenclatura, es necesario que esto se pueda identificar a simple vista. Además debe servir de guía para los implementadores que tienen que continuar el desarrollo de las aplicaciones luego.

### Reglas Generales

- ◆ Cuando se incluyan abreviaturas en mayúsculas no se debe incluir su nombre completo, sino que se utilice el primer nombre en mayúscula y el resto en minúsculas.

Ejemplo:

`GetHtmlStatistic` //Correcto

`GetHTMLStatistic` //Incorrecto



## CAPÍTULO 4: IMPLEMENTACIÓN

---

### Disposición del idioma

La implementación de la aplicación está basada en la utilización de un Framework que en su totalidad está codificado en inglés, es imposible desligar la implementación completamente de este idioma, pero también es complicado programar las aplicaciones completamente en inglés, ya que la experiencia de los desarrolladores en estos momentos, no lo permite.

Por lo tanto, es necesario crear una vinculación entre ambos idiomas, permitir que algunas nomenclaturas estén tanto en inglés como en español, incluso en ambos idiomas simultáneamente, ya que se hace muy difícil desligar la utilización de algunos prefijos y sufijos muy convencionales que están actualmente en inglés como son “Set”, “Get”, “Is”, “Max”, etc.

Hasta este punto, se decide a elección de los desarrolladores utilizar la nomenclatura idiomática más conveniente para nombrar cada uno de los elementos que se utilicen en el código, siempre y cuando cumpla con las demás restricciones de nomenclatura expresadas en el documento y se deje con claridad el propósito y la funcionalidad de cada elemento, ya sea en español o inglés.

### Componentes del Framework

#### Aplicaciones

- ◆ La aplicación debe tener nombres que dejen reflejado claramente cuál es el propósito de la misma, ya sea en una palabra o en siglas.
  - ✓ En caso de ser mediante siglas se ponen todas en mayúsculas.
- ◆ Se debe evitar mientras sea posible la utilización de palabras compuestas o la utilización de varias palabras, en caso de que sea palabras compuestas se utiliza la notación CamelCase.

#### Módulos

- ◆ Deben referirse a los nombres de tablas en caso de que se trate de un módulo generado por el CRUD.
- ◆ En caso de que sean módulos del negocio de la aplicación debe cumplir con las mismas reglas de codificación de los nombres de la aplicación.

#### Acciones

Symfony trae su propia nomenclatura para las clases de las acciones y sus funciones, pero no especifica claramente cuál es el nombre que se le debe poner a cada una de las funcionalidades del modelo que son accedidas por el usuario [dígase acciones].

## CAPÍTULO 4: IMPLEMENTACIÓN

---

- ◆ Dentro de las especificaciones del Framework está que, cada una de estas acciones debe comenzar con la palabra `execute`.
- ◆ Todos los nombres de acciones deben estar en la nomenclatura `CamelCase` comenzando por la palabra `execute`.
- ◆ En caso de sean acciones referentes a un módulo de CRUD de una tabla deben ser nombres específicos como `executeNuevo`, `executeEditar`, etc.
- ◆ Los nombres de las acciones deben especificar con la menor cantidad de palabras cual es el objetivo de la acción, de ser posible estar en infinitivo. Debe especificar bien claro cuál es la acción que se pretende ejecutar con la acción pero sin especificar los parámetros que recibe.

Ejemplo:

Correcto: `usuario/editar`

Incorrecto: `usuario/editar_dado_id`

### Nombre de las clases

- ◆ Los nombres de las clases deben estar expresados en notación `UpperCamelCase`.
- ◆ No se debe utilizar guiones bajos en su nombre “\_”.
- ◆ Se debe expresar con claridad cuál es el alcance y la responsabilidad de la clase.
- ◆ Los nombres de las clases no deben estar atados a las clases de las que se deriva, cada clase debe tener un significado por ella misma, no en dependencia de la clase de la que deriva.
- ◆ En los nombres compuestos por más de tres palabras se debe revisarse el diseño, no sea que se le estén dando a la clase más responsabilidades de las que realmente tiene.

### Clases en distintos ámbitos

Las nomenclaturas de las clases que se describe a continuación no deben entrar en conflicto con lo antes expresado para todas las clases de forma general.

### Modelo

Las clases del modelo generadas por Propel deben cumplir con la nomenclatura que propone el Framework para las clases de este tipo, el mismo nombre de la tabla que está referenciada, basándose en la utilización de la notación `UpperCamelCase` y sustitución de los guiones bajos de los nombres de las tablas por palabras compuestas.

Se mantiene el uso de los sufijos “Peer” para las clases que se encargan de estas funciones en el modelo.

## CAPÍTULO 4: IMPLEMENTACIÓN

---

### Negocio

Las clases del negocio deben cumplir con claridad las reglas de nomenclatura de las clases.

### Librerías

Generalmente las librerías utilizadas por terceros establecen un namespace para identificarlas y evitar que entren en conflictos con otras clases implementadas por los desarrolladores, en caso de desarrollar librerías propias se le debe identificar con un prefijo que sirva para diferenciar las clases de dicha librería de las demás clases del sistema. El prefijo puede estar dado por unas pocas letras o una palabra completa.

### Plugins

Las clases agrupadas dentro de los plugins cumplen con la misma nomenclatura que las aplicaciones del proyecto.

### Nombres de los archivos de las clases

Los nombres de los archivos de las clases deben estar compuestos por el nombre de la clase y a continuación de un punto y la palabra “class” y la extensión del archivo “.php”.

Ejemplo:

MiClase.class.php

### Helpers

En caso de que el helper sea un archivo de funciones el mismo debe estar nombrado con el sufijo “Helper” ajustándose a las notaciones del Framework. Debe expresar con claridad en no más de tres palabras cual es el alcance de los Helpers agrupados en el archivo.

En caso de que se trate de una clase el archivo debe contar con la nomenclatura de una clase contando además con el sufijo “Helper”.

### Estructura de las clases

Las clases se estructuran por niveles de la siguiente forma:

1. Variables Globales.
2. Constantes Globales.
3. Declaración de la clase.
4. Variables.
  - a. Miembros.
  - b. Referenciadas.
5. Constantes.

## CAPÍTULO 4: IMPLEMENTACIÓN

---

6. Constructores.
  - a. Sin parámetro.
  - b. Con parámetros.
  - c. Destruyores.
7. Funciones públicas.
  - a. Estáticas
  - b. No estáticas.
  - c. Redefinidas.
8. Funciones protegidas.
  - a. IDEM.
9. Funciones privadas.
  - a. IDEM.

### Nombres de las funciones

Todas las funciones definidas por los desarrolladores deben seguir la nomenclatura UpperCamelCase, a no ser que para cierto ámbito se especifiquen características específicas.

Además deben cumplir con las siguientes disposiciones de forma general:

1. Los nombres de las funciones deben dejar reflejado claramente cuál es la acción que realiza el mismo.

Ejemplo:

ManejarErrorBD, DescargarArchivoFTP, etc.

2. Se debe apoyar en la utilización de sufijos que ayuden a identificar el resultado final de la ejecución de un método:

Max: devuelve el máximo de una operación

Contar: Devuelve la cantidad de valores de una operación

3. Se debe apoyar en las prefijos para expresar la acción que realiza sobre un elemento determinado:

Set: para cambiar el valor de una variable.

Get: Para devolver un valor determinado.

Is: Devuelve un valor booleano de si un elemento cumple con una condición dada.

## CAPÍTULO 4: IMPLEMENTACIÓN

---

### Funciones en diferentes ámbitos

#### Modelo

Cualquier operación relacionada con una tabla de la base de datos debe situarse en el modelo de la clase correspondiente a dicha tabla.

- ◆ Deben expresar claramente el resultado devuelto.
- ◆ No se debe reflejar en el nombre de la función los parámetros que recibe a no ser que sea de relevancia para expresar el resultado de la función.

Ejemplo:

retireveByPK

#### Globales

Las funciones definidas de forma global dentro de un archivo y que no pertenezcan a clases en específico siguen las siguientes reglas de codificación:

- ◆ Son escritas en minúsculas
- ◆ Las palabras se separan por guiones bajos “\_”.

#### Helpers

Las funciones de los Helpers que cumplan con las características de las funciones globales siguen la nomenclatura de codificación para este tipo de funciones.

#### Plugins

Las funciones pertenecientes a los plugins siguen los mismos estándares de codificación de las funciones pertenecientes a las clases que correspondan dentro del plugin.

#### Visibilidad

No se debe hacer uso indiscriminado de funciones públicas, las funciones que solamente sean utilizadas por las clases que las contienen deben ser declaradas privadas, en las que clases que se esperen que sean extendidas, estas se declaran protegidas.

#### Variables

- ◆ Los nombres de las variables deben expresar el contenido de la misma.
- ◆ Pueden estar referidas en singular o plural.
- ◆ Se definen al principio de las estructuras donde son utilizadas.

Ejemplo:

Archivo, clase, función, IF, SWITCH, WHILE, etc.

## CAPÍTULO 4: IMPLEMENTACIÓN

---

- ◆ En caso de que no se le asigne un valor inicial se deben inicializar con un valor que indique el tipo de dato más general al que debe pertenecer.
  - a. Los tipos de datos cadena son definidos con comillas dobles ("");
  - b. Los tipos de datos de caracteres se definen con comillas simples (');
  - c. En caso de que se espere almacenar tipos de datos diversos no se inicializa.

Ejemplo:

```
$numero_bajas = 0;  
$nombre_acotado = "";  
$arreglo_temp = array();  
$objeto = new Clase;  
$mixta;
```

- ◆ De esta nomenclatura se exceptúan las variables generadas por el Framework.

### Prefijos

La utilización de prefijos se agrupa según en consecuencia de que las variables pertenezcan a varios grupos de los que se describen a continuación, comenzando por el nivel de visibilidad y luego por las otras características según el orden en que se nombran.

El siguiente grupo de variables utilizan nomenclatura CamelCase.

### Globales:

1. Comienzan con el prefijo "g".

### Atributos:

1. Comienzan con el prefijo "m".

### Parámetros de funciones:

1. Comienzan con el prefijo "p".
2. Están separados por una coma y un espacio.
3. En caso de ser objetos de una clase se le debe pasar el tipo de dato.

Ejemplo:

```
public function executeIndex(sfWebRequest $request)  
{  
    //TODO  
}
```

4. Los parámetros opcionales (o por defecto) se ubican después de los obligatorios en la declaración de las funciones.

## CAPÍTULO 4: IMPLEMENTACIÓN

---

### Referencias:

1. Comienzan con el prefijo “r”.
2. El “&” se ubica al lado del tipo de dato y no del nombre de la variable.

Ejemplo:

```
var&      $mrVariable;
```

### Funciones:

1. Se separan por guiones bajos y se escriben en minúsculas.

Ejemplo:

```
$propiedades_html;
```

```
$nombre_archivo;
```

- 1.1 Clases.
  - 1.2 Funciones.
  - 1.3 Constantes.
  - 1.4 Bucles.
  - 1.5 Mensajes.
  - 1.7 etc.
2. Nomenclatura de clases que contienen Formularios y campos de formularios.
  3. Nomenclatura de funciones JS.
    - 3.1 Agrupación de las funciones en distintos archivos.
    - 3.2 Nombre de los archivos.
  4. Nomenclaturas de clases de estilos.
    - 4.1 Agrupación de los distintos estilos en diferentes archivos.
    - 4.2 Nombre de los archivos.
    - 4.3 Nomenclaturas de los selectores.
    - 4.4 Tipos de selectores.

### Conclusiones

En el capítulo se completó el ciclo de vida del software, la etapa de Implementación. Esta es el centro durante las iteraciones de construcción. También se lleva a cabo el trabajo de implementación durante la fase de Elaboración, para crear la línea base ejecutable de la arquitectura. Durante la fase de Transición, para tratar defectos tardíos; debido: a que el modelo de implementación denota la implementación actual del sistema en términos de componentes y subsistemas de implementación. Por lo que es significativo, mantener el modelo de implementación a lo largo de todo el ciclo de vida del software.





# CONCLUSIONES

---

## **CONCLUSIONES**

Al culminar el presente trabajo de diploma, se ha cumplido con el objetivo y las tareas propuestas, por lo que se han obtenido los siguientes resultados:

- ✓ Se desarrolló la Aplicación Web para el Control de la Asistencia a Clases en la Facultad 7.
- ✓ Se implementó una aplicación en PHP mediante el framework Symfony para mejorar el proceso de control docente.
- ✓ Se puede acceder a la aplicación desde cualquier lugar debido a que es una aplicación web.
- ✓ Se estructuró la posible distribución del sistema.

Se obtuvo como producto final, una Aplicación Web para el Control de la Asistencia a Clases en la Facultad 7, la misma permite: la gestión de la asistencia a clases, la generación de reportes, el envío de justificaciones al ausentarse a clases y el registro de indisciplinas de los estudiantes. Esto proporciona, el control inmediato y objetivo de las evaluaciones, así como la eficiencia y calidad del control docente en la facultad.

### **RECOMENDACIONES**

Se recomienda:

- ✓ Continuar el desarrollo del sistema, agregándole funcionalidades en dependencia de nuevos procesos que se definan en la Facultad 7, como por ejemplo permitir que el estudiante pueda tener acceso a sus reportes de indisciplina.
- ✓ Extender el uso de la aplicación a todas las facultades de la universidad.

# REFERENCIAS BIBLIOGRÁFICAS

---

## REFERENCIAS BIBLIOGRÁFICAS

- [1] Paco Meseguer. La nueva gestión académica, 2003. [Disponible en: [http://www.microsoft.com/spain/enterprise/perspectivas/numero\\_7/educacion.msp](http://www.microsoft.com/spain/enterprise/perspectivas/numero_7/educacion.msp)]
- [2] Copyright © -2009 | Softaula, SL [Disponible en: [http://www.softaula.com/es/emp/lista\\_clientes.asp](http://www.softaula.com/es/emp/lista_clientes.asp)]
- [3] Facultad Regional de Buenos Aires. SIGA [Disponible en: <http://siga.frba.utn.edu.ar/>]
- [4] Departamento de Informática del Centro de la UNED de Barbastro. 2008-2009. [Disponible en: <http://www.barbastro.unedaragon.org/proyectos/default.aspx> ]
- [5] SIU-Guaraní.
- [6] Servicios informáticos al Alumnado. [Disponible en: <http://www.ehu.es/si/formacion/servicios/servicios-alumnos.html> ]
- [7] Asociación Peruana De software Libre. Noticia. [Disponible en: <http://www.apesol.org/news/100> ]
- [8] Universidad de Burgos. GAUSS - Formulario de Alta. [Disponible en: [http://www.ubu.es/ubu/cm/ubu/tkContent?idContent=98274&locale=es\\_ES](http://www.ubu.es/ubu/cm/ubu/tkContent?idContent=98274&locale=es_ES) ]
- [9] Lic. Yanoski Calderín Delgado, Dr. Julio A. Telot González (Tutor). GESTACAD. SISTEMA PARA LA GESTIÓN ACADÉMICA. Universidad Matanzas “Camilo Cienfuegos”. Facultad De Informática, CUBA.
- [10] Inc. (Nasdaq: MACR). Herramienta de desarrollo web líder de la industria se focaliza en la eficiencia, innovación e integración. 8 de Agosto de 2005 [Disponible en: <http://www.macromedia.com/software/dreamweaver/>.]
- [11] Diseñe, desarrolle y realice el mantenimiento de aplicaciones y sitios Web basados en estándares. [Disponible en: <http://www.adobe.com/es/products/dreamweaver/> ]
- [12] Fabien Potencier y François Zaninotto. Symfony 1.0, la guía definitiva. Publicada por la editorial Apress (ISBN-13: 978-1590597866). [Disponible en: [http://www.librosweb.es/symfony\\_1\\_0/](http://www.librosweb.es/symfony_1_0/) ]
- [13] Manual Javascript.2006. [Disponible en: <http://www.webestilo.com/javascript/> ]
- [14] Guía JavaScript 1.5. Forma parte de crozilla.com. 2008. [Disponible en: [https://developer.mozilla.org/es/Gu%C3%ADa\\_JavaScript\\_1.5/Concepto\\_de\\_JavaScript](https://developer.mozilla.org/es/Gu%C3%ADa_JavaScript_1.5/Concepto_de_JavaScript) ]
- [15] Aníbal de la Torre. Lenguajes del lado servidor o cliente. 2006. [Disponible en: [http://www.adelat.org/media/docum/nuke\\_publico/lenguajes\\_del\\_lado\\_servidor\\_o\\_cliente.html](http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html) ]
- [16] Arquitectura orientada a servicios. [Disponible en: [http://es.wikipedia.org/wiki/Arquitectura\\_orientada\\_a\\_servicios](http://es.wikipedia.org/wiki/Arquitectura_orientada_a_servicios) ]
- [17] [JAMS 98] K. Jamsa, S. Lalani y S. Weakley Programación para la Web, Ed. McGraw-Hill, 1998 [Disponible en: <http:// triana.escet.urjc.es/apliweb/AW-Tema2.pdf> ]
- [18] Carles Mateu i Piñol. Los manuales titulados Introducción a las aplicaciones web, Instalación del servidor web y Servicios web han sido extraídos, con algunas modificaciones, del manual "Desarrollo

## REFERENCIAS BIBLIOGRÁFICAS

---

- de aplicaciones web", publicado por la UOC dentro de su máster de software libre. [Disponible en: [http://www.cibernetia.com/manuales/servicios\\_web/4\\_wsdl.php](http://www.cibernetia.com/manuales/servicios_web/4_wsdl.php) ]
- [19] Centro de Soporte. Base de Conocimiento. Sistema de gestión de base de datos. 2006. [Disponible en: [https://www.ibercom.com/soporte/index.php?\\_m=knowledgebase&\\_a=viewarticle&kbarticleid=85](https://www.ibercom.com/soporte/index.php?_m=knowledgebase&_a=viewarticle&kbarticleid=85) ]
- [20] Los cuatro modelos de LDAP. [Disponible en: <http://es.tech-faq.com/ldap-lightweight-directory-access-protocol.shtml> ]
- [21] Manual de implementación de Red Hat Enterprise Linux 10.1. Razones por las cuales usar LDAP. [Disponible en [http://www.redhat.com/docs/manuals/enterprise/RHEL-5-manual/es-ES/Deployment\\_Guide/index.html](http://www.redhat.com/docs/manuals/enterprise/RHEL-5-manual/es-ES/Deployment_Guide/index.html) ]
- [22] Herramientas Case. [Disponible en: <http://www.monografias.com/trabajos14/herramicase/herramicase.shtml>]
- [23] Enterprise Architect - Herramienta de diseño UML. [Disponible en: <http://www.sparxsystems.com.ar/products/ea.html> ]
- [24] Patrón de arquitectura de software MVC modelo vista controlador. 2007. [Disponible en: <http://sabuesoweb.com/blog/2007/08/28/patron-de-arquitectura-de-software-mvc-modelo-vista-controlador/> ]
- [25] Jacobson, Ivar; Booch, Grady; Rumbaugh, James. "El Proceso Unificado de Desarrollo de Software". Addison Wesley, 2000.
- [26] Informática, Ministerio del Poder Popular para las Telecomunicaciones y la. MeRinde. MeRinde. [En línea] [http://merinde.rinde.gob.ve/index.php?option=com\\_content&task=view&id=96&Itemid=29](http://merinde.rinde.gob.ve/index.php?option=com_content&task=view&id=96&Itemid=29)
- [27] Vilas, Ana Fernandez. Diagrama de Componentes . 2001. [En línea] <http://tvdi.det.uvigo.es/~avilas/UML/node49.html>
- [28] François Zaninotto y Fabien Potencier. Symfony 1.1, la guía definitiva. [Disponible en: <http://www.scribd.com/doc/12977613/Symfony-1-0-Guia-Definitiva> ].

# BIBLIOGRAFÍA

---

## **BIBLIOGRAFÍA**

[En línea]. - <http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/>.

Base de Conocimiento. Sistema de gestión de base de datos [En línea]. –

[https://www.ibercom.com/soporte/index.php?\\_m=knowledgebase&\\_a=viewarticle&kbarticleid=85](https://www.ibercom.com/soporte/index.php?_m=knowledgebase&_a=viewarticle&kbarticleid=85).

propel [En línea]. - <http://propel.phpdb.org/trac/>.

ADOBE [En línea]. - <http://www.macromedia.com/la/>.

ADOBE DREAMWEAVER CS4 [En línea]. - <http://www.adobe.com/products/dreamweaver/>.

Aires Facultad Regional de Buenos SIGA [En línea]. - <http://siga.frba.utn.edu.ar>.

Arquitectura orientada a servicios. [En línea]. –

[http://es.wikipedia.org/wiki/Arquitectura\\_orientada\\_a\\_servicios](http://es.wikipedia.org/wiki/Arquitectura_orientada_a_servicios)

ASAMBLE3D [En línea]. - <http://www.ensemble3d.com/jscript/index.html>

Barbastro Departamento de Informática del Centro de la UNED de [En línea]. –

<http://www.barbastro.uned.aragon.org/proyectos/default.aspx>

Burgos Universidad de GAUSS - Formulario de Alta. [En línea]. –

[http://www.ubu.es/ubu/cm/ubu/tkContent?idContent=98274&locale=es\\_ES](http://www.ubu.es/ubu/cm/ubu/tkContent?idContent=98274&locale=es_ES)

danielpecos.com [En línea]. - [http://www.netpecos.org/docs/mysql\\_postgres/index.html](http://www.netpecos.org/docs/mysql_postgres/index.html) .

Departamento de Informática (ATC, CCIA, LSI) [En línea]. –

<http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node20.html>

desarrolloweb.com [En línea]. - <http://www.desarrolloweb.com/articulos/1581.php>

Diseño, desarrolle y realice el mantenimiento de aplicaciones y sitios Web basados en estándares.

[En línea]. - <http://www.adobe.com/es/products/dreamweaver/>

doc.ubuntu-es [En línea]. - <http://doc.ubuntu-es.org/LDAP>

Enterprise Architect - Herramienta de diseño UML. [En línea]. –

<http://www.sparxsystems.com.ar/products/ea.html>

Feedorss's Weblog [En línea]. - <http://feedorss.wordpress.com/2008/02/15/internacionalizacion-y-localizacion-en-symfony/>

FORMACIÓN DE USUARIOS EN LA GESTIÓN DE SECRETARÍA CON EL SISTEMA ÁGORA [En línea]. - [http://www.forpas.us.es/cursos/ficha\\_curso.asp?curso=02-15](http://www.forpas.us.es/cursos/ficha_curso.asp?curso=02-15)

Guía JavaScript 1.5 [En línea]. –

[https://developer.mozilla.org/es/Gu%C3%ADa\\_JavaScript\\_1.5/Concepto\\_de\\_JavaScript](https://developer.mozilla.org/es/Gu%C3%ADa_JavaScript_1.5/Concepto_de_JavaScript)

Herramientas Case [En línea]. –

<http://www.monografias.com/trabajos14/herramicase/herramicase.shtml>

HTMLPOINT.com [En línea]. - [http://www.htmlpoint.com/javascript/corso/js\\_02.htm](http://www.htmlpoint.com/javascript/corso/js_02.htm)

IvanEx [En línea]. - <http://ivanex.wikidot.com/patron-arquitectura>

# BIBLIOGRAFÍA

---

- Jacobson Ivar, Booch Grady y Rumbaugh James. "El Proceso Unificado de Desarrollo de Software". Addison Wesley, 2000 [En línea].
- K. Jamsa S. Lalani y S. Weakley Programación para la Web [En línea]. – <http:// triana.escet.urjc.es/apliweb/AW-Tema2.pdf>
- Libre Asociación Peruana De software [En línea]. - <http://www.apesol.org/news/100>.
- librosweb.es [En línea]. - [http://www.librosweb.es/symfony\\_1\\_0/capitulo9/helpers\\_de\\_enlaces.html](http://www.librosweb.es/symfony_1_0/capitulo9/helpers_de_enlaces.html).
- Lic. Yanoski Calderín Delgado Dr. Julio A. Telot González GESTACAD. SISTEMA PARA LA GESTIÓN ACADÉMICA. Universidad Matanzas "Camilo Cienfuegos" [Informe].
- Los cuatro modelos de LDAP [En línea]. - <http://es.tech-faq.com/ldap-lightweight-directory-access-protocol.shtml>
- MACR) Inc. (Nasdaq: Herramienta de desarrollo web líder de la industria se focaliza en la eficiencia, innovación e integración [Informe].
- MACR) Inc. (Nasdaq: Herramienta de desarrollo web líder de la industria se focaliza en la eficiencia, innovación e integración. [En línea]. - <http://www.macromedia.com/software/dreamweaver/>
- Manual de implementación de Red Hat Enterprise Linux 10.1. Razones por las cuales usar LDAP. [En línea]. - [http://www.redhat.com/docs/manuals/enterprise/RHEL-5-manual/es-ES/Deployment\\_Guide/index.html](http://www.redhat.com/docs/manuals/enterprise/RHEL-5-manual/es-ES/Deployment_Guide/index.html)
- Manual Javascript [En línea]. - <http://www.webestilo.com/javascript>.
- MeRinde. Ministerio del Poder Popular para las Telecomunicaciones y la Informática [Artículo].
- Meseguer Paco La nueva gestión académica [En línea]. – [http://www.microsoft.com/spain/enterprise/perspectivas/numero\\_7/educacion.msp](http://www.microsoft.com/spain/enterprise/perspectivas/numero_7/educacion.msp).
- Mozilla [En línea]. – [https://developer.mozilla.org/es/Gu%C3%ADa\\_JavaScript\\_1.5/Concepto\\_de\\_JavaScript](https://developer.mozilla.org/es/Gu%C3%ADa_JavaScript_1.5/Concepto_de_JavaScript).
- Patrón de arquitectura de software MVC modelo vista controlador [En línea]. – <http://sabuesoweb.com/blog/2007/08/28/patron-de-arquitectura-de-software-mvc-modelo-vista-controlador/>.
- Piñol. Carles Mateu i Los manuales titulados Introducción a las aplicaciones web, Instalación del servidor web y Servicios web han sido extraídos, con algunas modificaciones, del manual "Desarrollo de aplicaciones web", publicado por la UOC dentro de su m [En línea]. – [http://www.cibernetia.com/manuales/servicios\\_web/4\\_wsd.php](http://www.cibernetia.com/manuales/servicios_web/4_wsd.php).
- Potencier François Zaninotto y Fabien Symfony 1.1, la guía definitiva [En línea]. – <http://www.scribd.com/doc/12977613/Symfony-1-0-Guia-Definitiva>.
- Servicios informáticos al Alumnado [En línea]. - <http://www.ehu.es/si/formacion/servicios/servicios-alumnos.html>.
- SIU-Guaraní [Informe].
- Softaula [En línea]. - [http://www.softaula.com/es/emp/lista\\_clientes.asp](http://www.softaula.com/es/emp/lista_clientes.asp).

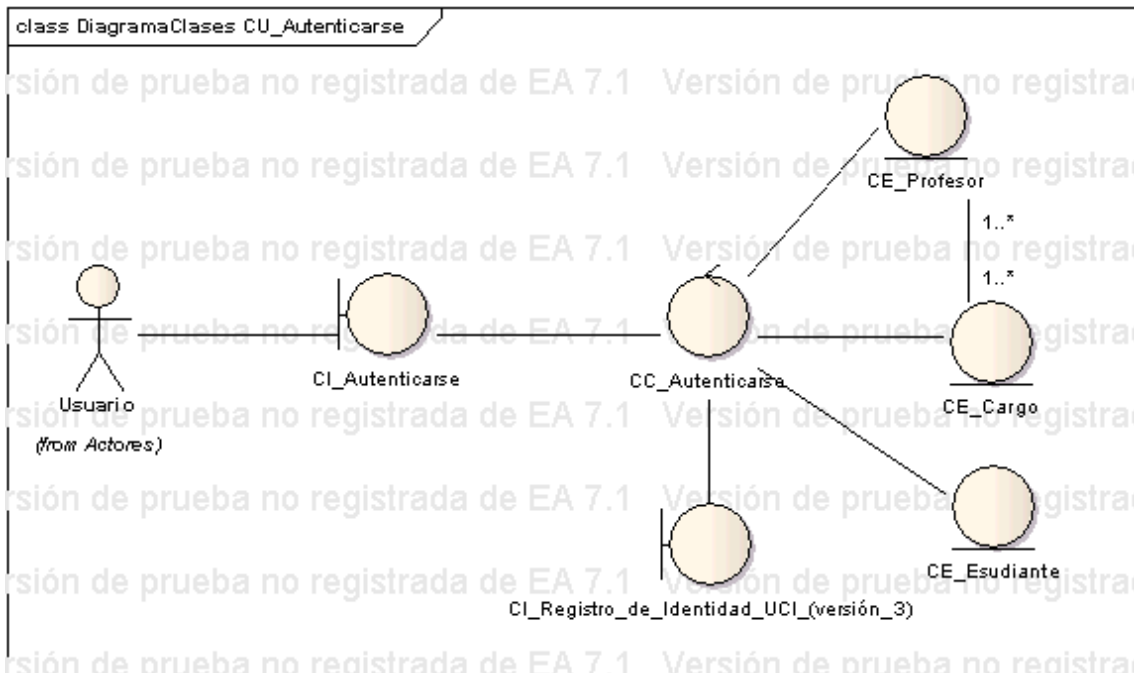
## BIBLIOGRAFÍA

---

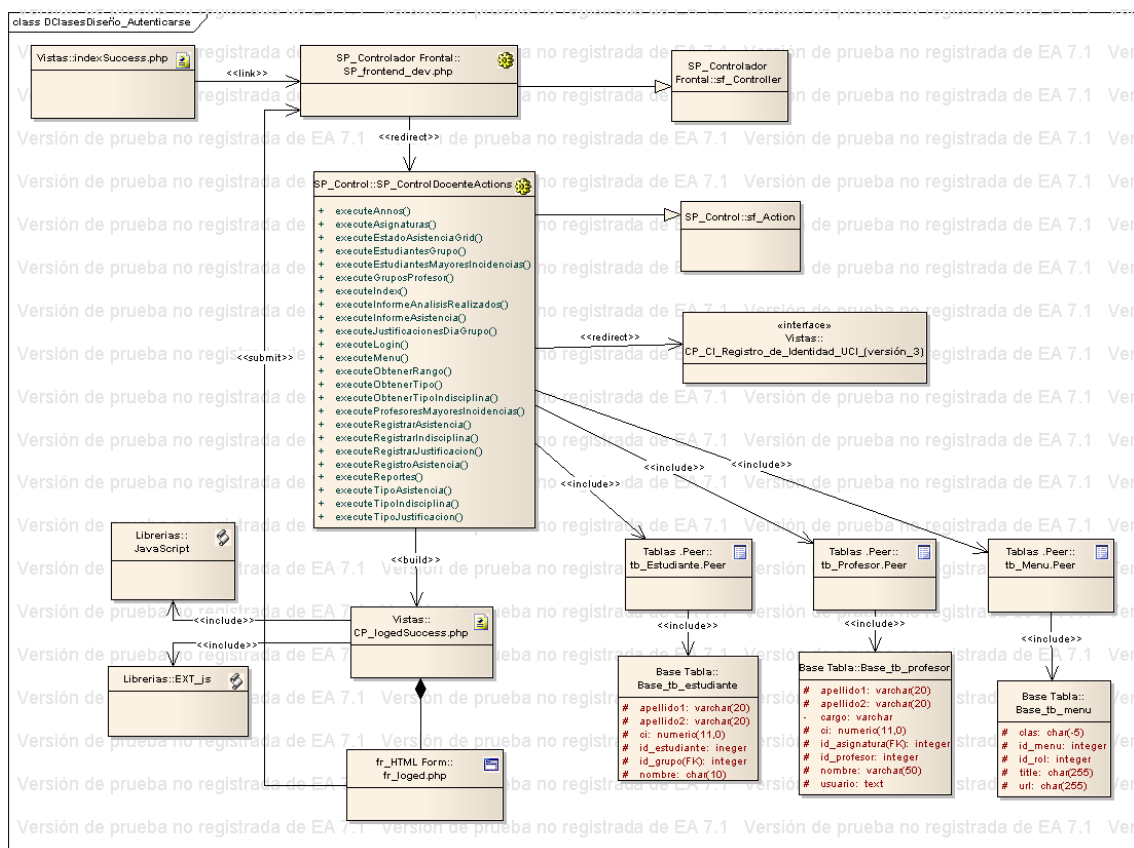
- SOURCEFORGE.NET [En línea]. - <http://sourceforge.net/projects/agorapucp>.
- SPARX SYSTEMS [En línea]. - <http://www.sparxsystems.com.ar/products/ea.html> .
- The Apache Software Foundation [En línea]. - <http://www.apache.org>.
- Torre. Aníbal de la Lenguajes del lado servidor o cliente. [En línea]. –  
[http://www.adelat.org/media/docum/nuke\\_publico/lenguajes\\_del\\_lado\\_servidor\\_o\\_cliente.html](http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html)
- UNED Barbastro [En línea]. - <http://www.barbastro.unedaron.org/proyectos/default.aspx>.
- Vilas Ana Fernandez Diagrama de Componentes [En línea]. –  
<http://tvdι.det.uvigo.es/~avilas/UML/node49.html>.
- Zaninotto. Fabien Potencier y François Symfony 1.0, la guía definitiva [En línea]. –  
[http://www.librosweb.es/symfony\\_1\\_0/](http://www.librosweb.es/symfony_1_0/)

## ANEXOS

**Anexo I: Diagrama de Clases del Análisis del caso de uso: Autenticarse.**

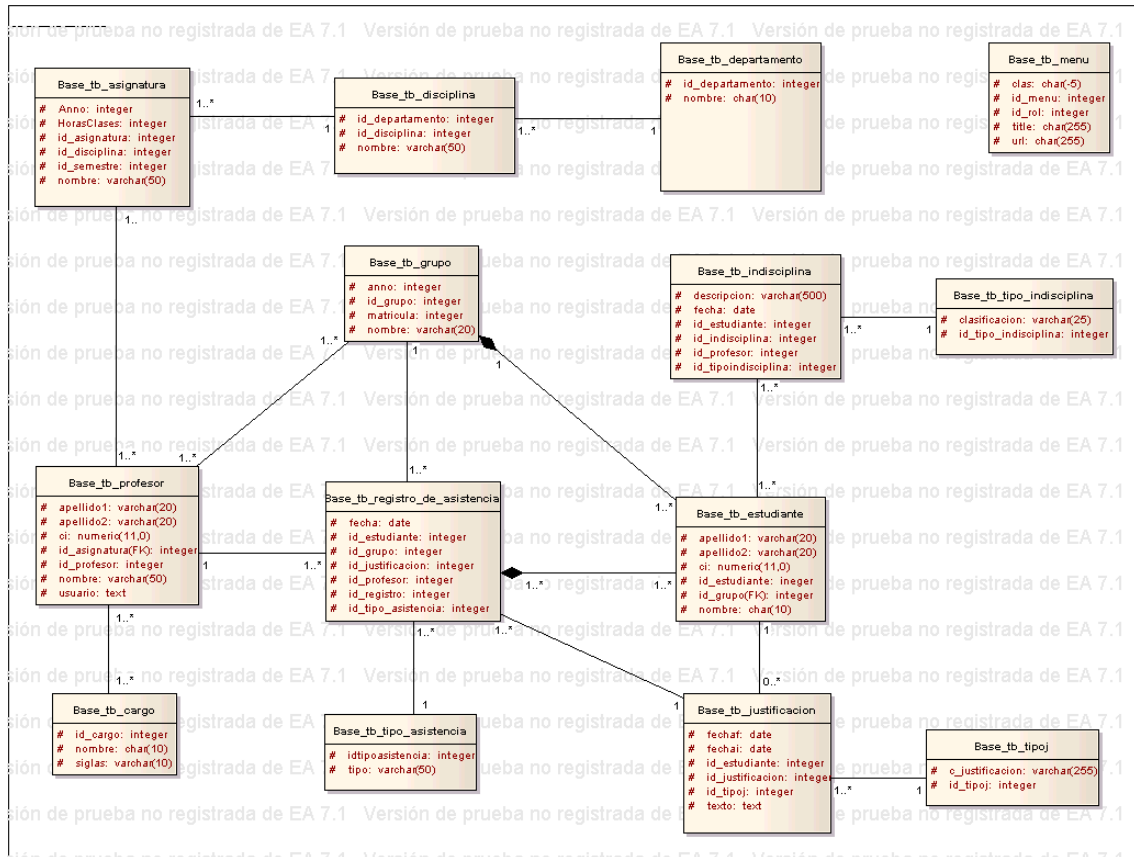


**Anexo II: Diagrama de Clases del Diseño del caso de uso: Autenticarse.**





## Anexo III: Diagrama de Clases Persistentes.



## Anexo IV: Interfaz de Autenticarse.

# Sistema de Control Docente para la Facultad 7

### Sistema de Control Docente

El sistema Control Docente, brinda prestaciones para que cada profesor de acuerdo al cargo que le corresponde, según la estructura organizativa de una facultad, tenga acceso a reportes actualizados. Además de permitir que los estudiantes puedan consultar su estado actual de asistencia y en caso de presentar algún problema particular poder notificar a sus profesores la ausencia, obteniendo la interacción deseada actualmente entre alumno-profesor de manera fluida y rápida.

### Login

## ANEXOS

### Anexo V: Interfaz de Página de Estudiante.

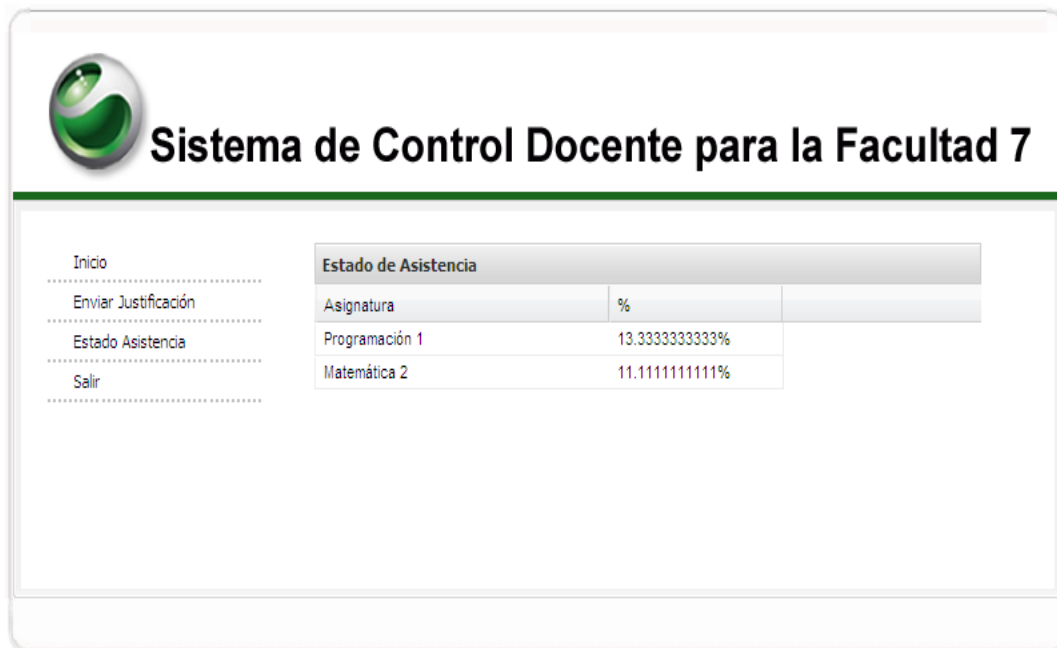
The screenshot shows the student interface of the 'Sistema de Control Docente para la Facultad 7'. It features a green and white logo on the top left. Below the logo, the title 'Sistema de Control Docente para la Facultad 7' is displayed. On the left side, there is a vertical menu with the following options: 'Inicio', 'Enviar Justificación', 'Estado Asistencia', and 'Salir'. The main content area is titled 'Sistema de Control Docente' and contains a paragraph of text: 'Estudiante: El sistema de gestión de la información de la asistencia a clases, Control Docente, brinda a su disposición, información que le permitirá consultar su estado actual de asistencia, proporcionándole un control propio de sí mismo.'

### Anexo VI: Interfaz de Enviar Justificación.

The screenshot shows the 'Enviar Justificación' (Send Justification) interface of the 'Sistema de Control Docente para la Facultad 7'. It features the same green and white logo and title as the previous interface. The left menu is identical. The main content area is titled 'Enviar Justificación' and contains the following fields: 'Fecha Inicio' (06/01/2009), 'Fecha Fin' (06/22/2009), 'Justificación' (Enfermo), and 'Comentario' (Diarreal). Below these fields are two buttons: 'Aceptar' (Accept) and 'Cancelar' (Cancel).

### Anexo VII: Interfaz de Estado de Asistencia.

## ANEXOS



**Sistema de Control Docente para la Facultad 7**

Inicio  
Enviar Justificación  
Estado Asistencia  
Salir

Estado de Asistencia	
Asignatura	%
Programación 1	13.3333333333%
Matemática 2	11.1111111111%

Anexo VIII: Interfaz de Registrar Asistencia.



**Sistema de Control Docente para la Facultad 7**

Reportes  
Registrar Indisciplina  
Registro de Asistencia  
Salir

Registro de asistencia

Fecha: 06/15/2009 Grupo: 07506

Aceptar Cancelar

Rigistro Asistencia		
Guardar Modificaciones		
Estudiante	Asistencia	justificado
francis	Presente	
yake	Tarde	
osvaldo	Ausente	
ketty	Ausente	
yosnier	Presente	

Anexo IX: Interfaz de Registrar Indisciplina.

## ANEXOS



### Sistema de Control Docente para la Facultad 7

Reportes

Registrar Indisciplina

Registro de Asistencia

Salir

**Registrar Indisciplina**

Grupo: 07506

Estudiantes: pepe chamosaoriente

Fecha: 06/16/2009

Tipo de Indisciplina: Menos Grave

Descripción: Falta de respeto

Anexo X: Interfaz de Reportes.



### Sistema de Control Docente para la Facultad 7

Reportes

Registrar Indisciplina

Registro de Asistencia

Salir

**Reportes**

Tipo de Reporte: Estudiante

Grupo: 07506

Estudiantes: francis dostres

Asignatura: Programaci&oacute;n 1

Desde: 06/01/2009

Hasta: 06/22/2009

Identificador	Estado
francis dos tres	6.66666666667%

## **GLOSARIO DE TÉRMINOS**

**CamelCase:** Consiste en escribir frases o palabras compuestas, eliminar los espacios intermedios y poner en mayúscula la primera letra de cada palabra excepto la primera letra de la frase.

**Componente de Software:** Es todo aquel recurso desarrollado para un fin concreto y que puede formar solo o junto con otro/s, un entorno funcional requerido por cualquier proceso predefinido. Son independientes entre ellos, y tienen su propia estructura e implementación. Si fueran propensos a la degradación debieran diseñarse con métodos internos propios de refresco y actualización. Son partes intangibles (que no se pueden tocar) de una computadora el cual lee los datos del hardware introduciéndolos en el ordenador.

**CU (Caso de uso):** En ingeniería del software, un caso de uso es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software. En otras palabras, un caso de uso es una secuencia de interacciones que se desarrollan entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.

**Despliegue:** Es un método sistemático, para garantizar que las propiedades, características y especificaciones de un producto, así como la selección y desarrollo de equipos, métodos y controles de proceso, estén orientados a las demandas del cliente o del mercado.

**Emporio:** Ciudad o lugar notable por el florecimiento del comercio, las ciencias, las artes, etc. Gran almacén comercial.

**Gestión:** Conjunto de trámites que se llevan a cabo para resolver un asunto.

**Hardware:** Conjunto de elementos materiales que constituyen el soporte físico de un ordenador.

**Helpers:** Son funciones de PHP que devuelven código HTML y que se utilizan en las plantillas. A veces, solamente se utilizan para ahorrar tiempo, agrupando en una sola instrucción pequeños trozos de código utilizados habitualmente en las plantillas.

**Imbricación:** Superposición parcial de objetos iguales, imitando la disposición de las escamas en los peces.

# GLOSARIO DE TÉRMINOS

---

**Mecanismo:** Estructura interna que hace funcionar algo. Modo de funcionamiento o desarrollo.

**MLC:** Miles de líneas de código.

**Módulo:** Es un software que agrupa un conjunto de subprogramas y estructuras de datos.

**Nomenclatura:** Conjunto o sistema de nombres o términos empleados en alguna disciplina, oficio o por alguna comunidad, etc. Conjunto de las palabras técnicas de una ciencia o facultad.

**Software:** Término genérico que se aplica a los componentes no físicos de un sistema informático, como p. ej. Los programas, sistemas operativos, etc., que permiten a este ejecutar sus tareas.

**Versión:** Modo de referir un mismo suceso. Narración o descripción distinta de un mismo hecho.

**Parámetro:** Valor numérico o dato fijo que se considera en el estudio o análisis de una cuestión.

**Plugins:** Son pequeños fragmentos de software que interactúan con el navegador para proporcionar algunas funciones que en la mayoría de los casos son muy específicas. Ejemplos típicos de plugins son los usados para mostrar los distintos formatos gráficos o para reproducir ficheros multimedia. Los plugins son ligeramente diferentes de las Extensiones, que modifican o se añaden a funcionalidades ya existentes.

**Requerimiento:** Necesidad o solicitud.

**UpperCamelCase:** Consiste en escribir frases o palabras compuestas, eliminar los espacios intermedios y poner en mayúscula la primera letra de cada palabra, incluye la primera letra de la frase.

**Variable:** Magnitud que puede tener un valor cualquiera de los comprendidos en un conjunto.

## GLOSARIO DE TÉRMINOS

---