



Universidad de las Ciencias Informáticas

Facultad 2

Título: Sistema para la gestión de los proyectos productivos



Trabajo de Diploma

Para Optar Por El Título De Ingeniero en Ciencias Informáticas

Autores: Madelis Pérez Gil

Yanio García Vidal

Tutor: Ing. Anabel Parra Vázquez

Ciudad de la Habana, Junio 2008

“Año 50 de la Revolución”

“Hay una fuerza motriz más poderosa que el vapor, la electricidad y la energía atómica: la voluntad”

Albert Einstein

DECLARACIÓN DE AUTORÍA

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

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

Madelis Pérez Gil
Autor

Yanio García Vidal
Autor

Ing. Anabel Parra Vázquez
Tutor

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado Sistema para la gestión de los proyectos productivos, fue realizado en la Universidad de las Ciencias Informáticas. Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface

Totalmente

Parcialmente en un ____ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes (cuantificar):

Como resultado de la implantación de este trabajo se reportará un efecto económico que asciende a:

Y para que así conste, se firma la presente a los ____ días del mes de _____ del año _____.

Representante de la entidad

Cargo

Firma

Cuño

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título: Sistema para la gestión de los proyectos productivos.

Autores: Madelis Pérez Gil

Yanio García Vidal

El tutor del presente Trabajo de Diploma considera que durante su ejecución los estudiantes mostraron las cualidades que a continuación se detallan.

Por todo lo anteriormente expresado considero que los estudiantes están aptos para ejercer como Ingeniero en Ciencias Informáticas y propongo que se le otorgue al Trabajo de Diploma la calificación de _____.

Ing. Anabel Parra Vázquez

Fecha

Dedicatoria

A nuestros padres: porque en ellos no conocemos la más sencilla de las maldades, en ellos solo hemos visto amor y comprensión...

"A ustedes, va dedicado este sueño"

Agradecimientos Madelis

A mis padres, este trabajo es la recompensa de sus años de esfuerzo y dedicación... gracias por creer en mi, los amo.

A mi tutora Anabel por su inmenso esfuerzo y paciencia... ya llegamos al final, el resultado no te defraudará.

A Yayi por su paciencia para entenderme y por todo lo que me ha ayudado en estos 5 años.

A mi tete, mi viejita linda quien ha estado siempre a mi lado dándome apoyo.

A la memoria de mi abuelo Pipi ese ángel que ahora me bendice desde su reino, que siempre confió en mí, que me brindó su apoyo hasta el último día de su vida para que viera realizados mis sueños y aunque no esté presente se que le hubiese gustado disfrutar como yo de este inolvidable momento...

A mi hermana Cuca, por apoyarme en todo momento y demostrarme que la distancia nunca será un impedimento para el cariño que nos une.

A mi Laly por su apoyo desde siempre y su inmenso cariño, por hacerle la vida a mi mamá un poco mejor y por su preocupación hacia mí como si fuera su hija.

A mi amigo Modest esa persona magnífica y extraordinaria... gracias por enseñarme acerca del intento, de la energía y la magia que están atrapados en cada uno de nosotros, eres especial... te adoro

A mi novio por su comprensión y su paciencia en esta recta final de mi carrera, un beso bien grande... te quiero Yanse.

A toda mi familia, por estar al tanto siempre de que en mis estudios todo marchara bien...

A Iralys y la Ñek por ser mis hermanitas de la universidad, por tantas hora de estudio y trabajo juntas y por aguantar todas mis majaderías...

A Lisbet, la Negra, Halena, Yunai, Yen y Pedro que me ayudaron en todo momento y no me dejaron sola en esta tarea tan difícil.

A alguien que en todo momento estuvo a mi lado y con el que compartí todas mis angustias y alegrías. Gracias por tu apoyo y todos los magníficos momentos que me regalaste.

A mis compañeros de grupo (2503), por estar conmigo en estos cinco años, los voy a extrañar...sin ustedes todo este sueño no se hubiera hecho realidad.

Agradecimientos Yanyo

A mis padres, que saben comprenderme en todo momento y que siempre confiaron en mí para hacer realidad este sueño.

A nanín por su ayuda y su compañía, mi hermanita linda.

A Made por soportarme durante la elaboración de este trabajo.

A mi tutora Anabel por su dedicación y su ayuda desinteresada.

A mi novia por su comprensión y paciencia en esta recta final de mi carrera, un beso bien grande... te quiero mi niña.

A Lisbet, la Negra, Iralilla, Pedro, Yaili, Yunier, Ismari y Castillo que me ayudaron en todo momento de mi carrera y más en la confección de esta tesis.

A mis compañeros de grupo (2103), por estar conmigo en estos cinco años y compartir juntos momentos inolvidables.

A mi abuelita Dulce por su apoyo que nunca me faltó.

A la memoria de mi abuelita mama que siempre tuvo esperanzas en mí. Abuelita te extraño.

A toda mi familia, por estar prestando atención siempre a mis estudios todo terminó bien.

A Todos los que de una forma u otra contribuyeron a mi formación como ingeniero informático.

A Todos muchas gracias.

Agradecimientos Generales

Al claustro de profesores de la Facultad, en especial a **Yadilka** quien nos ayudó a que nuestros días en la Universidad fueran cada vez mejores, al **Yony**, a **Joel**, a **Félix** el mejor de los filósofos...

Gracias Profes por contribuir día tras día a nuestra formación como Ingenieros Informáticos.

A la Revolución y en especial a nuestro siempre Comandante en Jefe Fidel Castro Ruz por habernos dado todas las condiciones y posibilidades de llevar a cabo esta labor.

RESUMEN

La asignación de personal a empresas es una tarea que consume mucho tiempo y de dicha asignación depende el futuro desarrollo de la misma, producto de la competitividad existente en el mundo. Con el devenir del tiempo se han agregado características como aptitudes, actitudes y relaciones interpersonales que deben de cumplir las personas para ocupar dichos puestos.

Se considera que todos los factores implicados en la producción de software se encuentran estrechamente interrelacionados, ya sean las herramientas a utilizar, las metodologías de desarrollo, los posibles clientes y/o usuarios, los recursos humanos, entre otros. Independientemente de esta interrelación, hay que señalar que la gestión de miembros para un proyecto determinado es el eslabón fundamental y que no puede faltar nunca. Sin embargo, no se tiene en cuenta que las personas no son recursos humanos, son individuos vivos, con todo su derecho a ser diferentes. Los cuales tienen una serie de necesidades sociales, que tan solo las pueden satisfacer en el contexto de los grupos a los que pertenecen. Valdría la pena preguntarse entonces: ¿de qué depende la productividad de un grupo?, o tal vez ¿Cómo lograr mayor productividad de un grupo determinado? (1)

Por lo que se ha hecho necesario realizar un modelo para una mejor gestión de los estudiantes en los diferentes proyectos productivos en la Universidad de las Ciencias Informáticas, así como una mejor organización de la entrada de los mismos al proyecto garantizando un mayor rendimiento y mejores resultados.

Palabras claves: recursos humanos, gestión de procesos, proyectos productivos.

Índice.

INTRODUCCIÓN	1
CAPÍTULO 1: “FUNDAMENTACIÓN TEÓRICA”	5
1.1 INTRODUCCIÓN	5
1.2 SISTEMAS PARA LA GESTIÓN DEL PERSONAL.	5
1.3 SISTEMAS PARA LA GESTIÓN DEL PERSONAL EN EL MUNDO	6
1.4 SISTEMAS PARA LA GESTIÓN DEL PERSONAL EN CUBA	8
1.5 SISTEMAS PARA LA GESTIÓN DEL PERSONAL EN LA UCI	8
1.6 TENDENCIAS Y TECNOLOGÍAS ACTUALES.	8
1.7 METODOLOGÍAS Y LENGUAJES	8
1.7.1 <i>Proceso Unificado de Desarrollo de Software (RUP)</i>	8
1.7.2 <i>HyperText Marckup Language (HTML)</i>	10
1.7.3 <i>Cascading Style Sheets (CSS)</i>	10
1.7.4 <i>PHP: Hypertext Preprocessor</i>	11
1.7.5 <i>JavaScript</i>	12
1.7.6 <i>Lenguaje Unificado de Modelado (UML)</i>	12
1.8 HERRAMIENTAS Y TECNOLOGÍAS ACTUALES.	13
1.8.1 <i>GIMP</i>	13
1.8.2 <i>PgAdmin</i>	13
1.8.3 <i>Visual Paradigm - UML</i>	13
1.8.4 <i>Eclipse</i>	14
1.8.5 <i>AJAX</i>	14
1.9 PATRONES DE ARQUITECTURA	15
1.9.1 <i>Arquitectura Orientada a Servicios (SOA)</i>	15
1.9.2 <i>Modelo Cliente Servidor</i>	16
1.9.3 <i>Modelo Vista Controlador (MVC)</i>	16
1.10 PATRONES DE DISEÑO	17
1.10.1 <i>Patrones GRASP</i>	18
1.11 SERVICIOS WEB.....	19
1.12 SISTEMAS DE GESTIÓN DE BASE DE DATOS.	20
1.12.1 <i>PostgreSQL</i>	20

1.13	IDEF0.....	20
1.14	FRAMEWORK (SYMFONY)	21
1.15	CONCLUSIONES	22
CAPÍTULO 2: “CARACTERÍSTICAS DEL SISTEMA”		23
2.1	INTRODUCCIÓN	23
2.2	MODELO DEL NEGOCIO IDEF (INTEGRATION DEFINITION FUNCTION MODELING)	23
2.2.1	<i>Representación de los procesos con IDEF0.....</i>	23
2.2.2	<i>Descripción de los procesos del negocio.....</i>	24
2.3	REQUISITOS FUNCIONALES.....	25
2.4	REQUISITOS NO FUNCIONALES	27
2.5	PROPUESTA DE SISTEMA.....	30
2.6	MODELO DEL SISTEMA	31
2.6.1	<i>Actores y Casos de Uso del sistema</i>	31
2.6.2	<i>Diagramas de casos de uso del sistema</i>	32
2.7	BREVE DESCRIPCIÓN DE LOS CASOS DE USO DEL SISTEMA.....	34
2.8	DESCRIPCIÓN DE LOS CASOS DE USO DEL SISTEMA EXPANDIDOS	38
2.9	CONCLUSIONES	38
CAPÍTULO 3: “DISEÑO”		39
3.1	INTRODUCCIÓN	39
3.2	DISEÑO.....	39
3.2.1	<i>Diagramas de interacción.....</i>	39
3.2.2	<i>Diagramas de clases del Diseño.....</i>	39
3.3	DISEÑO DE LA BASE DE DATOS.....	48
3.3.1	<i>Modelo Lógico de Datos (diagrama de clases persistentes).</i>	48
3.3.2	<i>Modelo físico de Datos (modelo de datos).....</i>	49
3.4	CONCLUSIONES	50
CAPÍTULO 4: “IMPLEMENTACION”		51
4.1	INTRODUCCIÓN	51
4.2	DIAGRAMA DE DESPLIEGUE	51
4.3	DIAGRAMA DE COMPONENTES.....	52
4.3.1	<i>Descripción de los Componentes.</i>	53

4.4	CONCLUSIONES	54
CAPÍTULO 5: “ESTUDIO DE LA FACTIBILIDAD”		55
5.1	INTRODUCCIÓN.	55
5.2	ESTIMACIÓN BASADA EN ANÁLISIS DE PUNTOS DE CASOS DE USO.	55
5.3	BENEFICIOS TANGIBLES E INTANGIBLES.....	61
5.4	ANÁLISIS DE COSTOS Y BENEFICIOS.	61
5.5	CONCLUSIONES.	61
CONCLUSIONES.....		63
RECOMENDACIONES.....		64
REFERENCIAS BIBLIOGRÁFICAS		65
BIBLIOGRAFÍA		67
ANEXOS.....		69
GLOSARIO DE TÉRMINOS.....		96

INTRODUCCIÓN

El desarrollo del software ha venido evolucionando en estos 40 años, ya se puede hablar de modelos de calidad, de procesos y de técnicas de trabajo, para lograr mayor eficiencia en el desarrollo de un producto con calidad. No utilizar ningún método de trabajo siempre es una opción, no se puede decir que en sus inicios ya se contaba con una técnica de trabajo clara, pero el de cursar de los años se ha demostrado que evidentemente no es una metodología a seguir.

Casi siempre los equipos que no utilizan ninguna metodología que logran un software con calidad, se debe a que son muy buenos programadores. A este modo de producción se le ha llamado "programación heroica", porque todo el peso del resultado descansa sobre el "saber-hacer" de los programadores.

El éxito dependerá entonces de los conocimientos adquiridos de cada miembro del equipo. Esta forma de trabajo produce resultados, la cual puede resultar más o menos apropiada, pero para la Universidad de las Ciencias Informáticas (UCI) no resultaría viable un sistema de producción en el que no pudieran intervenir todos los estudiantes, que solo fuera para un grupo selecto de "buenos programadores". Por ello es necesario utilizar un sistema informático que nos ayude a establecer las metodologías, las responsabilidades, los recursos, las actividades, etc. (2)

Actualmente la gestión del personal que trabajará en los proyectos productivos de la UCI, específicamente los estudiantes, se realiza en dependencia del tipo de proyecto, es decir el nivel de conocimientos básicos que deben tener los mismos para poder formar parte. Muchas veces pasa que el tipo de proyecto no está en correspondencia con los gustos de los estudiantes, esto trae como consecuencia que pierdan el interés o que no rindan lo suficiente en el proyecto, y con ello atentar contra el éxito del mismo. A medida que el tiempo transcurre se va perfeccionando la gestión de los proyectos por lo que se tienen mejor definidas las etapas del desarrollo y con ello los roles que intervienen en el proceso. La experiencia ha demostrado que los conocimientos adquiridos en el proceso regular diurno muchas veces no son suficientes, por ello es necesario realizar un gran trabajo con los estudiantes para que una vez que formen parte de un equipo de desarrollo rindan lo mayor posible.

Todas las metodologías de desarrollo de software explican las cantidades de roles que deben ser asignados y las actividades que deben cumplir cada miembro del equipo, y en algunos casos los conocimientos mínimos que debe tener cada persona, pero no exponen como seleccionar este

personal o como lograr que adquieran los conocimientos y las habilidades para desempeñarse en dicho rol. (3)

Entonces hay que tener presente el aspecto cognitivo que tenga el individuo, es decir, la capacidad técnica, que pueden ser los conocimientos para realizar la tarea, la capacidad de realizarla y la experiencia sobre la materia, por otro lado está la voluntad, que no es más que la motivación de la persona, el compromiso que asumirá y la seguridad que tiene en sí para realizarla. Cada investigador define más o menos principios sobre la gestión de un proyecto, todo está en dependencia de la experiencia adquirida.

La gestión de un proyecto de software comienza por la planificación del proyecto determinando el ámbito del software, es decir describiendo el control y los datos a procesar, la función, el rendimiento, las restricciones, las interfaces y la fiabilidad. Una vez definido lo anterior se pasa a la estimación de los recursos requeridos, primeramente las herramientas de hardware y software, luego los componentes de software reutilizables y por último el recurso primario: el personal.

De ahí que, a partir de la situación problémica descrita con anterioridad surge el **problema** de: ¿Cómo solucionar el problema de la gestión del personal?

De aquí hacer énfasis que cuando se refiere a gestión se habla de lo relacionado con el estudio de las actitudes y aptitudes del estudiante mediante la realización de pruebas y test, además de asignar los estudiantes a proyectos productivos, mostrar los estudiantes vinculados a proyectos, etc.

Muchas empresas que desarrollan software en el mundo realizan una gestión del personal que trabajará en dicha empresa, ya sea luego del resultado de un profundo estudio o de manera empírica. Este personal es profesional de la materia con cierto nivel de madurez en el desarrollo de software.

La UCI es un centro docente-productor experimental, que tiene como una de sus misiones: producir software y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación.

Los estudiantes van adquiriendo un nivel de madurez a medida que pasan de año académico, hasta formarse como profesionales comprometidos con su Patria, altamente calificados en la rama de la informática. Este centro posee ciertas características que lo distinguen de cualquier otra institución en el mundo. Para realizar la gestión de los estudiantes que trabajarán en los proyectos productivos no existe un patrón claro definido, cada facultad lo realiza de manera particular siempre y cuando cumpla con las misiones de la universidad.

Cada investigador define a su forma principios sobre la gestión de un proyecto, todo está en dependencia de la experiencia adquirida. Es una disciplina relativamente joven aunque de importancia creciente en distintos entornos. Tiene como finalidad principal la planificación, el seguimiento y control de las actividades y de los recursos humanos y materiales que intervienen en el desarrollo de un sistema de información, y como consecuencia de este control es posible conocer en todo momento qué problemas se producen y resolverlos de manera inmediata (4) (5)

En el caso de la Facultad # 2 una vez definido la cantidad de estudiantes que deben ser miembros, se procede a buscar las notas de las asignaturas que tienen que ver con el rol que jugaría en el proyecto y seguidamente se realiza un aval de la Federación Estudiantil Universitaria (FEU) y la Unión de Jóvenes Comunistas (UJC) de la brigada y en caso satisfactorio el estudiante queda seleccionado.

Definiéndose en la investigación como **Objeto de Estudio** el proceso de informatización del sistema para la gestión del personal para los proyectos productivos.

Se reconoce además como **Campo de Acción**: el proceso de informatización de un sistema para la gestión del personal para los proyectos productivos de la Facultad 2 de la UCI.

Con el fin de solucionar el problema planteado se tiene como **Objetivo General**: Implementar un sistema informático para la gestión de los proyectos productivos en la Facultad 2.

Por lo que proponemos llevar a cabo el cumplimiento del objetivo desarrollando las siguientes **Tareas de la investigación**:

- Analizar las características del proceso de selección de los miembros de un proyecto productivo.
- Estudiar las tecnologías y tendencias actuales a aplicar en el desarrollo de la aplicación.
- Realizar un estudio de los beneficios que le aportaría a la facultad la existencia de esta aplicación.

La presente propuesta planteada contribuye a mejorar este proceso en la Facultad # 2 y de esta forma se garantiza una mejor organización en este aspecto. Se persigue además obtener un producto de software que responda a las necesidades de nuestra facultad.

El documento quedó estructurado de la siguiente forma:

Capítulo 1: Fundamentación Teórica: En este capítulo se realiza un estudio del estado del arte de los sistemas de control de medios, además se exponen los conceptos fundamentales relacionados con el tema de investigación y se describen los lenguajes, las herramientas y metodología a utilizar para el desarrollo de la aplicación.

Capítulo 2: Características del Sistema: En este capítulo se realiza la descripción del sistema a implementar, para esto se describe el problema y la situación problemática a resolver, el objeto de automatización, se representan los procesos del negocio mediante IDEF0, se especifican los requisitos funcionales y no funcionales y se define el modelo de casos de uso del sistema.

Capítulo 3: Diseño del sistema: En este capítulo se realiza el diseño del sistema, dentro de este se exponen algunos de los diagramas del diseño.

Capítulo 4: Implementación: Se desarrolla el flujo de trabajo Implementación, empleando para ello la metodología RUP, registrándose los diagramas relacionados con el mismo.

Capítulo 5: Estudio de la Factibilidad: Estudio de factibilidad económica realizado para este proyecto, en el que se determina si es factible o no el desarrollo del software propuesto, analizando los diferentes criterios que influyen en el cálculo del esfuerzo, tiempo de desarrollo y costo del proyecto.

CAPÍTULO 1: “FUNDAMENTACIÓN TEÓRICA”

1.1 Introducción

La tecnología en la actualidad se encuentra en constante evolución, por lo que se hace necesario tener un conocimiento avanzado y actualizado de esta a la hora de desarrollar aplicaciones informáticas; ya que muchas veces las herramientas usadas pueden quedar obsoletas, trayendo consigo una considerable disminución en la calidad de las mismas. Teniendo en cuenta lo antes planteado el presente capítulo está destinado a brindar una breve descripción de todo lo relacionado con los procesos asociados a la gestión de los proyectos productivos de la UCI enfatizando en las características y procesos que están estrechamente vinculados con el desarrollo del trabajo, se ofrecerá además una descripción de los mecanismos de comunicación empleados en el trabajo y como último aspecto se hará un breve estudio acerca del estado actual de desarrollo de las aplicaciones de control y gestión en el ámbito internacional, nacional y en la Universidad, además de un estudio de las tecnologías en las que se apoya el desarrollo de estos tipos de sistemas en función de un análisis de las tendencias actuales.

1.2 Sistemas para la gestión del personal.

“El individuo inteligente no es sinónimo de empresa inteligente”

Dado el perfeccionamiento empresarial que han tenido las empresas en el mundo, todas persiguen dentro de sus objetivos ser las mejores en su campo de acción. Para alcanzar esta meta, se debe tener muy en cuenta la gestión del personal que trabaje en ellas. Es por ello, que la selección de estos tiene un valor muy importante, y del cual depende el futuro desarrollo de la empresa. Por lo que se ha comprobado, que se hacen grandes esfuerzos para que este indicador se realice de forma óptima, siempre tratando de contratar a las personas más capacitadas.

La selección, formación, comunicación, relaciones laborales y seguridad en el trabajo así como la dirección estratégica de la empresa, se han convertido en una función clave que permiten identificar las necesidades específicas de la organización y ofrecer diagnósticos acertados y soluciones válidas.

Toda institución necesita profesionales cualificados en el área contable, ya que es la herramienta indispensable para valorar cuantitativa y cualitativamente la actividad de la misma. Esta función será tanto más efectiva si el profesional conoce profundamente el área en la cual estaría sujeto a pertenecer.

Un buena selección del personal y una buena organización en la institución sería tratado como una tarea fundamental que permitiría acondicionar, organizar y distribuir los frutos del pensamiento individual y colectivo con el fin de compartir y transmitir el conocimiento acumulado de capital humano a través de soportes electrónicos incluyendo así los sistemas de información, redes de computación electrónica, o incluso a través de otros soportes físicos como patentes, productos y procesos.

No basta invertir en personas, sistemas o clientes separadamente, se apoyan los unos en los otros, integrados en un sistema. El capital humano estaría compuesto por la experiencia, el poder de innovación, y la habilidad de los empleados para realizar las tareas cotidianas. Incluye también la cultura, los valores y la filosofía es por esto que la implementación de un sistema informático para una buena gestión del personal se dice que es y será de gran ayuda a este proceso.

1.3 Sistemas para la gestión del personal en el mundo



Figura 1.1 Logotipo de Base de Datos Integrada

Bdi (Base de Datos Integrada): Es una herramienta diseñada por Bendersky & Asoc para la selección, organización y sistematización de los procesos de Búsqueda de Personal. Es un sistema de bases de datos argentino que tiene la finalidad de organizar los curriculums de personas en búsqueda laboral y proveer la información a las organizaciones que necesiten incorporar personal, así como de los existentes en su empresa.

Los datos están disponibles para las empresas y organizaciones que necesiten buscar personal. Bdi puede realizar búsquedas complejas utilizando las herramientas más avanzadas de la tecnología de bases de datos.

Esta aplicación pueden utilizarla las consultoras en Selección de Personal, los departamentos de RRHH, las Asociaciones Profesionales, las Cámaras Empresariales, los Gremios, o cualquier organización que necesite organizar una Base de datos o Bolsa de Trabajo para el reclutamiento de personas con fines laborales.

RRHH (Recursos Humanos)

RRHH es una línea de productos de software argentino que brindan una solución integral y definitiva para las necesidades de las empresas que deseen optimizar los procesos de selección y búsqueda

de personal, para lograr rapidez, mayor efectividad, menores costos y mejorar la imagen empresarial.

¿Qué se logra con RRHH?

- Capturar curriculum vitae vía internet, email, o ingresarlos manualmente en la base de datos.
- Archivar ordenadamente información de sus empleados y verlos en un formato estándar.
- Realizar búsquedas por cualquier combinación de datos que hayan sido ingresados (formación, sexo, edad, etc.).
- Puede realizar el seguimiento de las entrevistas de los candidatos.

RRHH está conformado por dos productos que satisfacen distintas necesidades y que juntos conforman la herramienta más completa del mercado de Recursos Humanos.

Los productos que conforman el paquete son:

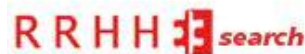


Figura 1.2 Logotipo de Recurso Humanos para Búsquedas

Un programa único y exclusivo de búsqueda inteligente y seguimiento de información que ningún otro producto iguala en el mercado.



Figura 1.3 Logotipo de Recurso Humanos para Web

Incluye todo lo necesario para que las personas interesadas en participar de búsquedas de empleo en su empresa ingresen los currículos a través del sitio en internet de su Organización.

Ventajas Concretas:

- Brinda información procesada en forma totalmente sistematizada y estandarizada.
- Ofrece infinitos parámetros o criterios de selección, color de ojos, estatura, con auto, con moto, sin experiencia.
- En segundos el programa brinda los resultados de una búsqueda compleja.
- La respuesta al cliente externo o interno es rápida en tiempo y minuciosa en cuanto a contenido.
- Cualquier persona dentro de la empresa puede acceder a los datos, con previa clave de autorización.
- Permite el seguimiento de las personas entrevistadas, con comentarios y evaluaciones, inclusive dentro de las distintas áreas de la empresa.
- Se evita la acumulación de papeles.

1.4 Sistemas para la gestión del personal en Cuba

Actualmente en Cuba, todo el sistema empresarial está en un proceso de perfeccionamiento del modelo de gestión por lo que establecer éste con una filosofía de mejoramiento continuo de los procesos es importante para dar respuesta a las exigencias del ambiente externo que rodea las organizaciones.

Con todo este marco de referencia, se puede observar que es necesario buscar competitividad transformando desde lo más profundo a la organización, modificando los valores, actividades y comportamientos de las personas que la forman para desarrollar una cultura competitiva de la calidad.

La gestión de procesos, inicia la creación de un estilo de trabajo cuando nace la idea de desarrollar productos o servicios con una calidad competitiva pero no existe ningún sistema capaz de desarrollar esto aún, aunque se tiene pensado por la suma importancia y la organización que este traería consigo a cualquier institución.

1.5 Sistemas para la gestión del personal en la UCI

De este tipo de sistemas no se ha implementado ninguno que cumpla con los requisitos del software que se desarrollará.

1.6 Tendencias y tecnologías actuales.

La tendencia en este trabajo es mejorar el proceso de gestión de miembros de un equipo de proyecto productivo logrando su informatización, así como toda la gestión del personal que lleva implícito este proceso.

Para el desarrollo del sistema se realizó un estudio sobre las posibles herramientas a utilizar en su construcción. Teniendo en cuenta la tendencia actual y las novedades en este campo se hace la siguiente propuesta.

1.7 Metodologías y Lenguajes

1.7.1 Proceso Unificado de Desarrollo de Software (RUP).

El Proceso Unificado de Rational es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis,

implementación y documentación de sistemas orientados a objetos. RUP es en realidad un refinamiento realizado por Rational Software del más genérico Proceso Unificado.

El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento). Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. (6)

Son estas tres características las que distinguen precisamente a RUP de otras metodologías de desarrollo:

Proceso dirigido por Casos de Uso

En RUP los Casos de Uso no son sólo una herramienta para especificar los requisitos del sistema. También guían su diseño, implementación y prueba. Los Casos de Uso constituyen un elemento integrador y una guía del trabajo. Los Casos de Uso no sólo inician el proceso de desarrollo sino que proporcionan un hilo conductor, permitiendo establecer trazabilidad entre los artefactos que son generados en las diferentes actividades del proceso de desarrollo.

Proceso centrado en la arquitectura

La arquitectura involucra los aspectos estáticos y dinámicos más significativos del sistema, está relacionada con la toma de decisiones que indican cómo tiene que ser construido el sistema y ayuda a determinar en qué orden. Además la definición de la arquitectura debe tomar en consideración elementos de calidad del sistema, rendimiento, reutilización y capacidad de evolución por lo que debe ser flexible durante todo el proceso de desarrollo.

Proceso iterativo e incremental

El equilibrio correcto entre los Casos de Uso y la arquitectura es algo muy parecido al equilibrio de la forma y la función en el desarrollo del producto, lo cual se consigue con el tiempo. Para esto, la estrategia que se propone en RUP es tener un proceso iterativo e incremental en donde el trabajo se divide en partes más pequeñas o mini proyectos. Permitiendo que el equilibrio entre Casos de Uso y arquitectura se vaya logrando durante cada mini proyecto, así durante todo el proceso de desarrollo. Cada mini proyecto se puede ver como una iteración (un recorrido más o menos completo a lo largo de todos los flujos de trabajo fundamentales) del cual se obtiene un incremento que produce un crecimiento en el producto. Al finalizar se realiza una integración de los resultados con lo obtenido de las iteraciones anteriores.

Además contiene un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. (7)

1.7.2 HyperText Marckup Language (HTML).

HTML define varios aspectos de una página web, incluyendo títulos, negritas, itálicas, imágenes, párrafos y conexiones de hipertexto a otros recursos, es por ello que puede ser comparado con el procesamiento de palabras. El texto de un archivo de un procesador de palabras puede ser formateado de varias formas.

HTML es estandarizado y multiplataforma. En efecto, un documento que sea preparado utilizando marcas HTML puede ser leído utilizando una variedad de navegadores Web, como Netscape, Explorer o Linux. El navegador interpreta las marcas HTML en un archivo y presenta la información formateada como una página Web.

1.7.3 Cascading Style Sheets (CSS).

Las hojas de estilo en cascada (Cascading Style Sheets, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML).

Se dice que es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre el estilo y formato de sus documentos.

Ventajas de utilizar CSS:

- Control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo.
- Los Navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio web, con lo que aumenta considerablemente la accesibilidad. Por ejemplo, personas con deficiencias visuales pueden configurar su propia hoja de estilo para aumentar el tamaño del texto o remarcar más los enlaces.

1.7.4 PHP: Hypertext Preprocessor

PHP es un lenguaje de script (o de guiones), diseñado para, entre otras cosas, aumentar, incrementar el dinamismo de las páginas Web. Originalmente se trataba de un conjunto de macros concebidas para ayudar en el mantenimiento de páginas Web. (8)

Desde entonces, sus características han ido creciendo hasta convertirse en un lenguaje de programación completo, capaz de manejar entornos que integran grandes bases de datos.

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

Tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como UNIX (y de ese tipo, como Linux), Windows y Mac OS X, y puede interactuar con los servidores web más populares.

Incluye algunos usos importantes de los cuales se hace referencia a continuación:

- Programación de páginas web dinámicas, habitualmente en combinación con el motor de base datos MySQL, aunque cuenta con soporte nativo para otros motores, incluyendo el estándar ODBC, lo que amplía en gran medida sus posibilidades de conexión.
- Creación de aplicaciones gráficas independientes del navegador, por medio de la combinación de PHP y Qt/GTK+, lo que permite desarrollar aplicaciones de escritorio en los sistemas operativos en los que está soportado.

Por qué escogimos PHP?

1. PHP no soporta directamente punteros, como el C, de forma que no existen los problemas de depuración provocados por estos.
2. Se pueden hacer grandes cosas con pocas líneas de código. Lo que hace que merezca la pena aprenderlo.
3. El código php es mucho más legible que el de PERL, todo el que haya programado PERL podrá corroborar esta afirmación.
4. Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad.

Resumiendo, el PHP corre en 7 plataformas, funciona en 11 tipos de servidores, ofrece soporte sobre unas 20 Bases de Datos y contiene unas 40 extensiones estables sin contar las que se están experimentando, además de que está instalado en el 34,02% de los servidores de Internet, seguido por ASP en el 21,41%. (9)

1.7.5 JavaScript

Java Script es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas Web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. Al contrario que Java, Java Script no es un lenguaje orientado a objetos propiamente dicho, ya que no dispone de herencia, es más bien un lenguaje basado en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad. (10)

Todos los navegadores interpretan el código Java Script integrado dentro de las páginas Web. Para interactuar con una página Web se provee al lenguaje Java Script de una implementación del Document Object Model (DOM).

Se venía utilizando en páginas Web HTML, para realizar tareas y operaciones en el marco de la aplicación únicamente cliente, sin acceso a funciones del servidor. Java Script se ejecuta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML.

1.7.6 Lenguaje Unificado de Modelado (UML).

UML (Unified Modeling Language) es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software.

UML establece la base para un estándar en el dominio del análisis y el diseño orientados a objetos, fundado en una amplia base de experiencia de los usuarios. Es necesario recalcar que UML es una notación y no un proceso/método, es decir, es una herramienta útil para representar los modelos del sistema en desarrollo, más no ofrece ningún tipo de guía o criterios acerca de cómo obtener esos modelos. (11)

Permite a los creadores de sistemas generar diseños que capturen sus ideas en una forma convencional y fácil de comprender para comunicarlas a otras personas que estén involucradas en el proceso de desarrollo de los sistemas, esto se lleva a cabo mediante un conjunto de símbolos y diagramas.

El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas y proporciona un estándar que permite al analista de sistemas generar un anteproyecto de varias facetas que sean comprensibles para los clientes, desarrolladores y todos aquellos que estén involucrados en el proceso de desarrollo. Un modelo UML indica que es lo que supuestamente hará el sistema pero no como lo hará. (12)

1.8 Herramientas y Tecnologías actuales.

1.8.1 GIMP.

GIMP (GNU Image Manipulation Program) es un programa de edición de imágenes, tanto dibujos como fotografías. Es un programa libre y gratuito, englobado en el proyecto GNU y disponible bajo la licencia Licencia pública general de GNU. Existen versiones totalmente funcionales para Windows, para Mac OS X y para otros sistemas operativos, haciéndolo el programa multiplataforma. GIMP existe en español además de en la lengua de desarrollo original, el inglés, y también en otros lenguajes. Muy similar a la Adobe Photoshop.

1.8.2 PgAdmin

Es desarrollado por una comunidad de los expertos de PostgreSQL de varias partes del mundo y está disponible en más de una docena de idiomas. Se trata de una herramienta para la administración de bases de datos PostgreSQL. Su uso se puede extender hacia las plataformas de Linux, de FreeBSD, de Solaris, del Mac OSX y de Windows.

1.8.3 Visual Paradigm - UML

Visual Paradigm para UML es una de las herramientas UML CASE del mercado, considerada como muy completa y fácil de usar, con soporte multiplataforma y que proporciona excelentes facilidades de interoperabilidad con otras aplicaciones. Fue creada para el ciclo vital completo del desarrollo del software, permitiendo la captura de requisitos, análisis, diseño e implementación. (13)

Visual Paradigm-UML también proporciona características tales como generación del código, ingeniería inversa y generación de informes. Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de clases. Permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML al instante, creando de manera simple toda la documentación.

Está diseñado para usuarios interesados en sistemas de software de gran escala con el uso del acercamiento orientado a objetos, además apoya los estándares más recientes de las notaciones de Java y de UML. Incorpora el soporte para trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros. Es un producto de calidad que soporta aplicaciones Web y es muy fácil de instalar y de utilizar.

Además el diseño es centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.

Esta herramienta usa un lenguaje estándar común para todo el equipo de desarrollo que facilita la comunicación, disponibilidad de múltiples versiones, para cada necesidad.

Entre las principales ventajas que proporciona el VP-UML podemos citar:

- Integración completa con Microsoft Office.
- Es multiplataforma y muy útil para la generación de código fuente en PHP.
- Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de clases.
- Permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML al momento, creando de forma simple toda la documentación.
- Incorpora el soporte para trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros de equipo.

1.8.4 Eclipse

Eclipse es un entorno de desarrollo integrado de código abierto independiente de una plataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. (14)

Las versiones actuales de Eclipse disponen de las siguientes características:

- Editor de texto
- Resaltado de sintaxis
- Compilación en tiempo real
- Pruebas unitarias con JUnit
- Control de versiones con CVS

Asimismo, a través de "plugins" libremente disponibles es posible añadir otras funcionalidades.

1.8.5 AJAX

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una tecnología de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Éstas se ejecutan en el cliente, es decir, en el navegador de los usuarios y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma. (15)

Ventajas del AJAX

- Recuperación asíncrona de datos, el usuario no tienen que esperar después de una petición.
- Acercamiento de la metáfora de escritorio a la Web.
- No requiere plugins.
- Se reduce el tamaño de la información intercambiada.
- Interactividad: Las aplicaciones AJAX se ejecutan en la máquina del usuario, manipulando la página actual dentro de sus navegadores usando métodos de Document Object Model. Puede ser usado para multitud de tareas como actualizar o eliminar registros, expandir formularios web; todo sin tener la necesidad de recargar toda la página de HTML cada vez que se realiza un cambio.
- Portabilidad: Las aplicaciones construidas con AJAX utilizan características bien documentadas presentes en todos los navegadores importantes en la mayoría de las plataformas existentes.

1.9 Patrones de Arquitectura

1.9.1 Arquitectura Orientada a Servicios (SOA)

La Arquitectura Orientada a Servicios o SOA es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requerimientos de software del usuario. 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. En un ambiente SOA, los nodos de la red hacen disponibles sus recursos a otros participantes en la red como servicios independientes a los que tienen acceso de un modo estandarizado. (16)

Ventajas

1. Reducción del tiempo de desarrollo: Los servicios SOA son fácilmente reusables y pueden ser ensamblados rápidamente. Gracias a la reusabilidad los tiempos de desarrollo de las aplicaciones se reducen drásticamente. Puesto que generalmente en las empresas la mayor parte de procesos de negocio ya estaban implementados, es mucho más rápido reutilizarlos que volver a crearlos.
2. Más ágil: Debido al débil acoplamiento entre las interfaces de los distintos servicios se consiguen aplicaciones más ágiles debido al entorno en continuo cambio en el que están inmersas las empresas.
3. Mayor calidad en los servicios y reducción de riesgos: La reutilización de servicios es segura y fiable puesto que ya han sido probados a través de numerosos clientes que acceden a

ellos y en muchos casos mejorados. Con lo que se aumenta la calidad del servicio y se reducen los riesgos.

1.9.2 Modelo Cliente Servidor.

Se dice que la arquitectura Cliente/Servidor es la integración distribuida de un sistema en red, con los recursos, medios y aplicaciones que, definidos modularmente en los servidores, administran, ejecutan y atienden las solicitudes de los clientes; todos interrelacionados física y lógicamente, compartiendo datos, procesos e información.

Se establece así un enlace de comunicación transparente entre los elementos que conforman la estructura. Entre las principales características de la arquitectura Cliente/Servidor, se pueden destacar las siguientes:

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

Ventajas de la arquitectura cliente-servidor:

- El servidor no necesita potencia de procesamiento, parte del proceso se reparte con los clientes.
- Se reduce el tráfico de red considerablemente. Idealmente, el cliente se conecta al servidor cuando es estrictamente necesario, obtiene los datos que necesita y cierra la conexión dejando la red libre.

PostgreSQL usa una arquitectura proceso-por-usuario cliente/servidor.

1.9.3 Modelo Vista Controlador (MVC).

El patrón Modelo-Vista-Controlador separa el modelamiento del dominio, la presentación, y las acciones basadas en las entradas hechas por el usuario en tres clases fundamentales:

Modelo: Administra y maneja el comportamiento y los datos del dominio de aplicación, da respuestas a peticiones de información sobre el estado de la aplicación (normalmente desde la Vista), y responde con instrucciones de cambio de estado (usualmente desde el controlador) a la vista.

Vista: Gestiona lo relacionado con mostrar la información al usuario.

Controlador: El controlador interpreta los eventos que son lanzados por la entrada estándar del usuario (normalmente mouse y teclado), informando de los mismos al modelo y/o la vista para que se ejecuten los cambios apropiadamente.

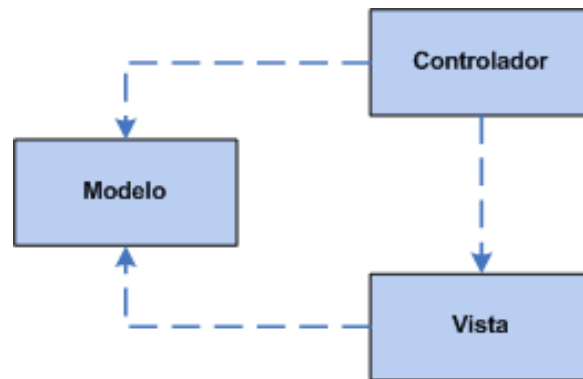


Figura 1.4 Esquema del patrón

1.10 Patrones de Diseño

Un patrón es un modelo que podemos seguir para realizar algo. Los patrones surgen de la experiencia de seres humanos de tratar de lograr ciertos objetivos. Estos capturan la experiencia existente y probada para promover buenas prácticas. (17)

Se dice que además es una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características:

- Se debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores.
- Debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

Cada patrón describe un problema que ocurre infinidad de veces en nuestro entorno, así como la solución al mismo, de tal modo que podemos utilizar esta solución más adelante sin tener que volver a pensarla.

Los patrones de diseño pretenden:

- Proporcionar catálogos de elementos reusables en el diseño de sistemas software.
- Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.
- Formalizar un vocabulario común entre diseñadores.
- Estandarizar el modo en que se realiza el diseño.

- Facilitar el aprendizaje de las nuevas generaciones de diseñadores condensando conocimiento ya existente.

Específicamente se utilizaron en la realización de este trabajo los siguientes patrones:

1.10.1 Patrones GRASP.

GRASP es el acrónimo de General Responsibility Assignment Software Patterns. Una de las cosas más complicadas en Orientación a Objeto consiste en elegir las clases adecuadas y decidir como estas clases deben interactuar. Incluso cuando utilizamos metodologías rápidas como programación extrema (extreme programming) y centramos el proceso en el desarrollo continuo, es inevitable elegir cuidadosamente las responsabilidades de cada clase en la primera codificación y fundamentalmente, en la refactorización (continua) del programa.

Bajo Acoplamiento.

1. Debe haber pocas dependencias entre las clases. Si todas las clases dependen de todas ¿cuántos software podemos extraer de un modo independiente y reutilizarlos en otro proyecto?
2. Para determinar el nivel de acoplamiento de clases, son muy buenos los diagramas de colaboración de UML.
3. Uno de los principales síntomas de un mal diseño y alto acoplamiento es una herencia muy profunda. Siempre hay que considerar las ventajas de la delegación respecto de la herencia.

Alta Cohesión.

Cada elemento del diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable. Ejemplos de una baja cohesión son clases que hacen demasiadas cosas. En todas las metodologías se considera la refactorización. Uno de los elementos a refactorizar son las clases saturadas de métodos. Ejemplos de buen diseño se producen cuando se crean los denominados "paquetes de servicio" o clases agrupadas por funcionalidades que son fácilmente reutilizables (bien por uso directo o por herencia).

Experto

1. La responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados (atributos). Una clase, contiene toda la información necesaria para realizar la labor que tiene encomendada. Hay que tener en cuenta que esto es aplicable mientras estemos considerando los mismos aspectos del sistema:
2. Lógica de negocio.
3. Persistencia a la base de datos.
4. Interfaz de usuario.

Creador

Guía la asignación de responsabilidades relacionadas con la creación de objetos (una tarea muy común). La intención básica del patrón es encontrar un creador que necesite conectarse al objeto creado en alguna situación. Teniendo como ventaja bajo acoplamiento logrando mayor mantenimiento y reutilización.

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

1.11 Servicios Web

Un Servicio Web es una colección de protocolos y estándares que sirve para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios Web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. (19)

El Servicio Web es un componente de software que se basa en las siguientes tecnologías:

1. Un formato que describa la interfaz del componente (sus métodos y atributos) basado en XML. Por lo general este formato es el WSDL
2. Un protocolo de aplicación basado en mensajes y que permite que una aplicación interactúe (use, instancia, llame, ejecute) al servicio. Por lo general este protocolo es SOAP (Simple Object Access Protocol), Protocolo Simple de Acceso a Datos.

Un servicio Web puede ser usado internamente por una aplicación o ser publicado en Internet. Estos servicios permiten la ejecución de sus funcionalidades sin importar la plataforma, sistema operativo, o lenguaje en el cual estén implementados. Gracias a los servicios Web podemos hacer que sistemas heterogéneos trabajen conjuntamente como una sola aplicación computacional. Estos servicios constituyen una potente herramienta para el desarrollo de aplicaciones distribuidas.

1.12 Sistemas de Gestión de Base de Datos.

1.12.1 PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos Objeto-Relacionales (ORDBMS) libre, liberado bajo la licencia BSD (Berkeley Software Distribution). Es una alternativa a otros sistemas de bases de datos de código abierto (como MySQL, Firebird y MaxDB), así como sistemas propietarios como Oracle y SQLServer. El mismo ha sido desarrollado de varias formas desde 1977.

Este sistema gestor posee una serie de características positivas respecto a otros, por ejemplo: (20)

- Posee una gran escalabilidad: Es capaz de ajustarse al número de procesadores y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta (en algunos benchmarks se dice que ha llegado a soportar el triple de carga de lo que soporta MySQL).
- Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos.
- Aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, transacciones, optimización de consultas, herencia, y arreglos.
- Tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL.

1.13 IDEF0 (Modelación de Funciones)

La traducción literal de las siglas IDEF0 es Integration Definition for Function Modeling (Definición de la integración para la modelización de las funciones). Consiste en una serie de normas que definen la metodología para la representación de funciones.

Representa de manera estructurada y jerárquica las actividades que conforman un sistema o empresa y los objetos o datos que soportan la interacción de esas actividades.

IDEF0 posee las siguientes características:

- Es capaz de representar gráficamente una amplia variedad de negocios y de otros tipos de operaciones de la empresa a cualquier nivel del detalle.
- Es muy coherente y simple, pero a la vez riguroso y exacto.
- Puede ser generado fácilmente por una gran variedad de herramientas gráficas en computadores.

Usos de IDEF0:

- Facilitar el análisis para la identificación de puntos de mejora.
- Modelar una gran variedad de sistemas, tanto automatizados como no automatizados.
- Comunicar reglas y procesos de negocios.

1.14 Framework (Symfony)

Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. (21)

Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft.

Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows.

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares)
- Independiente del sistema gestor de bases de datos
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.

Además automatiza la mayoría de elementos comunes de los proyectos web, como por ejemplo :

- La capa de internacionalización que incluye Symfony permite la traducción de los datos y de la interfaz, así como la adaptación local de los contenidos.
- Los formularios incluyen validación automatizada y relleno automático de datos ("repopulation"), lo que asegura la obtención de datos correctos y mejora la experiencia de usuario.
- Los datos incluyen mecanismos de escape que permiten una mejor protección contra los ataques producidos por datos corruptos.
- Las interacciones con Ajax son muy fáciles de implementar mediante los helpers que permiten encapsular los efectos Java Script compatibles con todos los navegadores en una única línea de código.

Es un framework maduro, bien documentado, recomendable para aplicaciones Web complejas con mucha lógica de negocio pues permite el mantenimiento y las ampliaciones futuras de la aplicación, con un código ligero, legible y efectivo.

1.15 Conclusiones

En este capítulo se exponen las condiciones y problemas que rodean el objeto de estudio y a través de los conceptos y definiciones planteadas. Se evidencia la necesidad de implementar un software que permita la informatización para la gestión de los miembros para un proyecto productivo en la facultad 2.

Para desarrollar el sistema se hace uso de la tecnología para la programación de páginas dinámicas el lenguaje PHP5 y con soporte de base de datos en PostgreSQL. La metodología utilizada para llevar a cabo el proceso de desarrollo es RUP, la cual está basada en la orientación a objetos y el modelamiento visual usando UML, lo cual permite incorporar al proceso de desarrollo de software un mejor control de los requerimientos y cambios. Se fundamentó la elección de las herramientas, el uso de patrones de diseño y de arquitectura, framework que se utilizarán a lo largo del desarrollo de la aplicación.

CAPÍTULO 2: “CARACTERÍSTICAS DEL SISTEMA”

2.1 Introducción

Antes de comenzar a desarrollar un sistema es necesario comprender la organización bajo estudio y los procesos que en ella tienen lugar, a fin de lograr una mejor comprensión del problema a resolver y el común entendimiento entre clientes y desarrolladores; para lo cual se realiza la modelación del negocio.

El modelo del negocio posibilita obtener una visión más clara del proceso en cuestión.

En este capítulo se describe la propuesta de solución para la implementación del sistema. Se presenta el modelo IDEF0, donde se representan las funcionalidades requeridas por el sistema, lo que permitirá comprender el contexto en que se emplaza el mismo.

Se plantean los requisitos funcionales de la aplicación a desarrollar, y se modela la misma teniendo en cuenta los casos de uso y actores que interactúan con ellos.

2.2 Modelo del Negocio IDEF (Integration Definition Function Modeling)

Para el modelado del negocio se empleó IDEF0; pues con su utilización se pueden representar de una forma estructurada y sencilla los principales procesos existentes en el mismo. Trayendo esto consigo un mayor entendimiento del cliente de todo el proceso de negocio.

2.2.1 Representación de los procesos con IDEF0

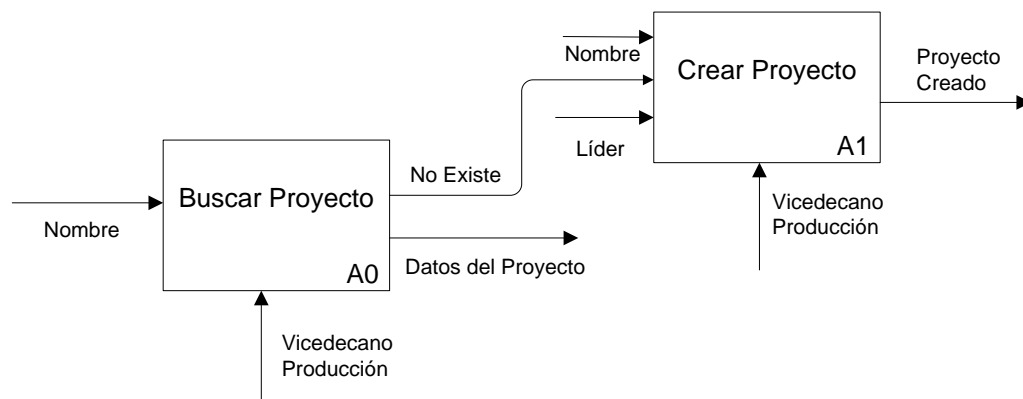


Figura 2.1 Proceso Crear Proyecto

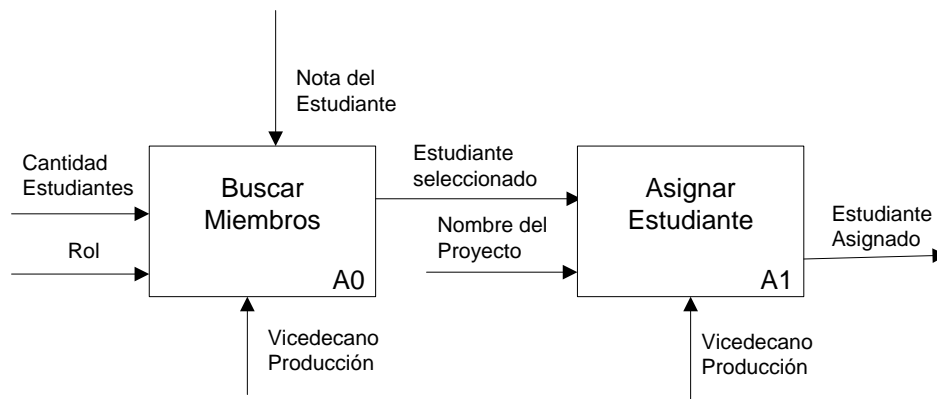


Figura 2.2 Proceso Asignar Estudiante.

2.2.2 Descripción de los procesos del negocio

Buscar Proyecto: El proceso se inicia cuando el vicedecano de producción entra el nombre para buscar el proyecto correspondiente a la búsqueda especificada. Como resultado de ésta búsqueda pueden existir proyectos ya existentes teniendo como salida los datos del proyecto y en el caso que no haya sido registrado o sea que no exista se procede al paso de crearlo.

Crear Proyecto: El proceso se inicia cuando el proyecto no ha sido registrado o creado, luego el vicedecano de producción introduce los datos necesarios para registrarlo que en este caso es el nombre del proyecto y el líder proporcionando como salida el proyecto creado.

Buscar Miembros: El proceso se inicia cuando el Vicedecano de producción decide cual sería el rol que necesita según un determinado proyecto en este caso: arquitecto, diseñador de la base de datos, diseñador gráfico, programador y analista, esta clasificación es dada teniendo en cuenta los principales roles que tienen los proyectos existentes en la facultad. De acuerdo al rol que se necesite se valora la nota del estudiante en correspondencia con la asignatura principal que corresponda con el rol, ejemplo: si se busca un programador el estudiante debe tener 5 puntos como nota en la asignatura de programación.

En caso que el estudiante ya esté en otro proyecto pues no se asigna a ninguno y se excluye de la selección y en caso contrario pues el proyecto recibirá la asignación con los estudiantes correspondientes y entonces se modifica el listado de proyecto actualizándose con los nuevos miembros, siendo entonces la salida del proceso una asignación de estudiante a un proyecto.

Asignar Estudiantes a Proyecto: El proceso comienza con la entrada de los estudiantes que resultaron seleccionados en la búsqueda realizada presentando como salida el listado de estudiantes asignados a proyecto.

2.3 Requisitos Funcionales

“El paso desde la determinación de las necesidades del cliente hasta la implementación no es trivial. En primer lugar las necesidades del cliente no son fáciles de discernir. Después debemos ser capaces de diseñar una implementación funcional que se ajuste a esas necesidades”.

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

R1. Autenticar Usuario.

- 1.1. Pedir nombre de usuario y contraseña para entrar al sistema.
- 1.2. Mandar directamente al usuario registrado a las opciones principales con que cuenta su privilegio.

R2. Cargar Configuración.

- 2.1. Verificar en la BD el privilegio del usuario autenticado:
- 2.2. Mostrar la interfaz con las opciones del menú que tiene acceso el tipo de usuario autenticado.
 - 2.2.1. Si el usuario autenticado es estudiante se habilitan las opciones siguientes:
 - 2.2.1.1 Realizar Test.
 - 2.2.1.2 Realizar Prueba.
 - 2.2.2. Si el usuario autenticado es Vicedecano Producción se habilitan las opciones siguientes:
 - 2.2.2.1 Asignar Estudiantes.
 - 2.2.2.2 Mostrar Resultado.
 - 2.2.2.3 Mostrar Miembros.
 - 2.2.2.4 Gestionar Proyecto.
 - 2.2.2.5 Gestionar Rol.
 - 2.2.2.6 Gestionar Pregunta.
 - 2.2.2.7 Gestionar Lenguaje.

R3. Realizar Test.

- 3.1. Realizar el Test donde se le mide las actitudes y aptitudes del estudiante.

R4. Realizar Prueba

- 4.1. Realizar prueba para programador.
- 4.2. Realizar prueba para diseñador de la base de datos.
- 4.3. Realizar prueba para diseñador de la interfaz de usuario.
- 4.4. Realizar prueba para arquitecto.
- 4.5. Realizar prueba para analista.

R5. Buscar Miembros para un proyecto.

5.1. Criterios para la búsqueda:

- 5.1.1. Rol (Diseñador de interfaz de Usuario, Diseñador de Base de Datos, Programador, Arquitecto y Analista).

5.2. Resultado del Test.

5.3. Resultado de la prueba.

5.4. Notas de la asignatura en correspondencia con el rol que se pida.

- Programador (5 en P1, P2, P3, P4.)
- Diseñador de Base de Datos (5 en BD).
- Arquitecto (5 en P1, P2, P3, P4, BD, ISW1, ISW2).
- Analista (5 ISW 1, ISW2).
- Diseñador Gráfico (Cursos Optativo

5.5. Promedio General (más de 4 puntos)

5.6. Si estuvo vinculado a proyecto o no y en caso de ser afirmativo que rol desempeñó.

5.7. Resultados de la búsqueda.

5.7.1. Nombre y Apellidos

5.7.2. Grupo

5.7.3. Rol

R6. Gestionar Proyecto.

- 6.1. Crear Proyecto.
- 6.2. Listar Proyecto.
- 6.3. Eliminar Proyecto.

R7. Gestionar Pregunta.

- 7.1. Adicionar Pregunta.
- 7.2. Listar Pregunta.
- 7.3. Eliminar Pregunta.

R8. Gestionar Rol.

- 8.1. Adicionar Rol.
- 8.2. Eliminar Rol.

R9. Gestionar Lenguaje

- 9.1. Adicionar Lenguaje.
- 9.2. Eliminar Lenguaje.

R10. Asignar estudiantes a un proyecto.

- 10.1. Asignar Manual
- 10.2. Asignar Automático

R11. Mostrar Miembros.

- 11.1. Mostrar por rol
- 11.2. Mostrar por proyecto.

R12. Mostrar Resultado de Prueba.**2.4 Requisitos no Funcionales**

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Están vinculados a requerimientos funcionales, es decir una vez se conozca lo que el sistema debe hacer podemos determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser.

Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto, pues si se conoce que el mismo cumple con la toda la funcionalidad requerida, las propiedades no funcionales, como cuán usable, seguro, conveniente y agradable, pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.

A continuación se muestran los requerimientos no funcionales:

Interfaz externa:

Interfaz con un diseño sencillo que contenga pocos gráficos, con vista a acelerar la velocidad de respuesta hacia el usuario debido a la complejidad de los procesos llevados a cabo en el terminal. La interfaz debe limitarse a presentar las funcionalidades netamente de la estación logrando la concentración del usuario en las tareas que esté realizando. Su diseño gráfico debe ser acorde con las pautas de diseño de la Intranet universitaria, adaptada para una resolución de 1024x768.

Usabilidad.

El sistema ha sido diseñado de tal forma que permite a los usuarios tener control en todo momento del sitio web, evitando para eso la sobrecarga de información y para cada proceso una muestra a través de mensaje de las acciones hechas por los usuarios. Los colores usados permiten el acceso a usuarios con problemas de distinción de color, se reduce el tiempo de latencia, además, requiriendo un mínimo proceso de aprendizaje.

Rendimiento.

Teniendo en cuenta que el sistema servirá como base para la toma de decisiones muy importantes para la universidad, la aplicación debe ser eficiente, precisa y rápida, con un tiempo de respuesta en una transacción o traspaso de información mínimo de 2 segundos.

Soporte

El producto debe recibir mantenimiento ante cualquier fallo que ocurra. El sistema es de fácil instalación. El sistema debe ser integrado, de manera que se mantenga la colaboración entre todos los subsistemas que lo componen. La base de datos que utilizará el sistema como medio de almacenamiento de la información estará soportada sobre un gestor de bases de datos PostgreSQL, permitiéndole interactuar con otros sistemas estableciendo vías de compatibilidad.

Se logra una solidez de los datos realizando mantenimientos automatizados en la base de datos, orientados a la actualización y salvadas de seguridad, a horas del día donde haya la menor cantidad de usuarios conectados.

Portabilidad.

El producto es usado bajo los S.O Windows, Linux y Unix.

El producto corre sobre una plataforma Web, codificada en "PHP5".

Seguridad

Este es quizás el tipo de requerimiento más difícil, que provocará los mayores riesgos si no se maneja correctamente. La seguridad puede ser tratada en tres aspectos diferentes:

- **Confidencialidad:** La información manejada por el sistema está protegida de acceso no autorizado y divulgación.
- **Integridad:** la información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes, de la misma forma será considerada igual a la fuente o autoridad de los datos. Pueden incluir también mecanismos de chequeo de integridad y realización de auditorías.
- **Disponibilidad:** Significa que los usuarios autorizados se les garantizará el acceso a la información y que los dispositivos o mecanismos utilizados para lograr la seguridad no ocultarán o retrasarán a los usuarios para obtener los datos deseados en un momento dado.

La seguridad de un sistema no solo tiene en cuenta la seguridad del sistema propiamente dicho sino, además, el ambiente en el que se usará el sistema. Por lo que se tiene que contemplar la seguridad física del lugar donde se usa la aplicación, los controles administrativos que se establecen de acceso al sistema y las regulaciones legales que afecta o determina el uso del sistema y que serán tenidas en cuenta si se incumple. El acceso al sistema será restringido donde cada usuarios solo tendrá los mínimos permisos para realizar sus acciones y tendrá un módulo de administración para crear o eliminar los usuarios, con sus roles, modificar los permisos que estos tienen y crear o eliminar los almacenes en el sistema.

Legales

El sistema se basa en un estándar que se rige por normas internacionales y cumple con las normas y leyes establecidas en nuestro país. La plataforma escogida para el desarrollo de la aplicación, está basada en la licencia GNU/GPL.

Confiabilidad.

Es decisiva la precisa ejecución de la aplicación para la actualización de la base de datos. Esta característica es fundamental debido que la precisión de los datos que se reciben y en función de su valor garantizan que el artículo este de alta o baja.

En caso de alguna falla que pudiera provocar la detención de la aplicación se debe proceder a su inmediata reparación y realización nuevamente de todos los chequeos que se realizaron en ese tiempo.

Hardware.

Para la terminal cliente:

- Tarjeta de red.

Para el servidor:

- Con 120 Gb de disco duro.
- Tarjeta de red.
- 2 Gb de memoria RAM.

Software

Para el funcionamiento del sistema en el servidor es necesario:

- Servidor de Aplicaciones (Apache Web Server 2.0 o superior)
- Sistema Gestor de Bases de Datos (PostgreSQL 8.2 o superior)
- Sistema Operativo: GNU/Linux (se recomienda Debian)

Para el funcionamiento en las terminales clientes es necesario:

- Navegador Web: Internet Explorer v 6.0 o Mozilla 2.0 o Superior.

2.5 Propuesta de Sistema

Con el objetivo de mejorar las dificultades que existen con el proceso de gestión de los miembros que forman o formarán un proyecto productivo en la facultad 2 se decidió realizar una aplicación web, que se encargue de controlar todo este proceso, la misma tendrá una interfaz amigable, simulando una aplicación de escritorio que permite una fácil interacción con el cliente.

Esta aplicación debe ser capaz de brindar diferentes servicios como son: En caso del estudiante se le brinda la posibilidad de realizar un test donde se le midan las aptitudes y actitudes para ingresar a un proyecto además de una prueba por cada rol, obligatoriamente tiene que realizar una prueba pero se le brinda la posibilidad para su mejor preparación que si quiere puede realizar más de una y en caso de que ya esté vinculado a proyecto pues puede repetir cuantas veces quiera una prueba pues estas siempre no va a ser las mismas.

Además el vicedecano de producción puede ver el resultado y la prueba de un determinado estudiante en todas las que este realice, así como gestionar preguntas que consiste en adicionar una pregunta a una determinada prueba, eliminarla, o mostrarla, gestionar rol y lenguaje que permite adicionar y eliminar estos respectivamente, gestionar proyecto incluyendo aquí las operaciones de crear, listar o eliminar un proyecto, asignar estudiantes a los mismos.

2.6 Modelo del Sistema

2.6.1 Actores y Casos de Uso del sistema

Considerando que la definición de Actor del Sistema relaciona todo aquello que intercambie información con la aplicación y que puede representar el rol que juega una o varias personas, un equipo o un sistema automatizado, además de intercambiar información con él o ser recipientes pasivos de información, se definen los siguientes actores:

Actores del sistema

Actores	Justificación
Usuario	Representa a todos los usuarios que tienen acceso a la aplicación, o sea el Vicedecano Producción y el estudiante.
Estudiante	Representa a todos lo estudiantes de la facultad que tendrán acceso al sistema para realizar un test y pruebas.
Vicedecano Producción	Es la persona encargada de realizar las principales funcionalidades del Sistema, brindándole así los diferentes servicios para realizar la gestión de los proyectos productivos

Tabla 2.1: Descripción textual de los Actores del Sistema

2.6.2 Diagramas de casos de uso del sistema

A continuación se muestra el diagrama de casos de uso del sistema.

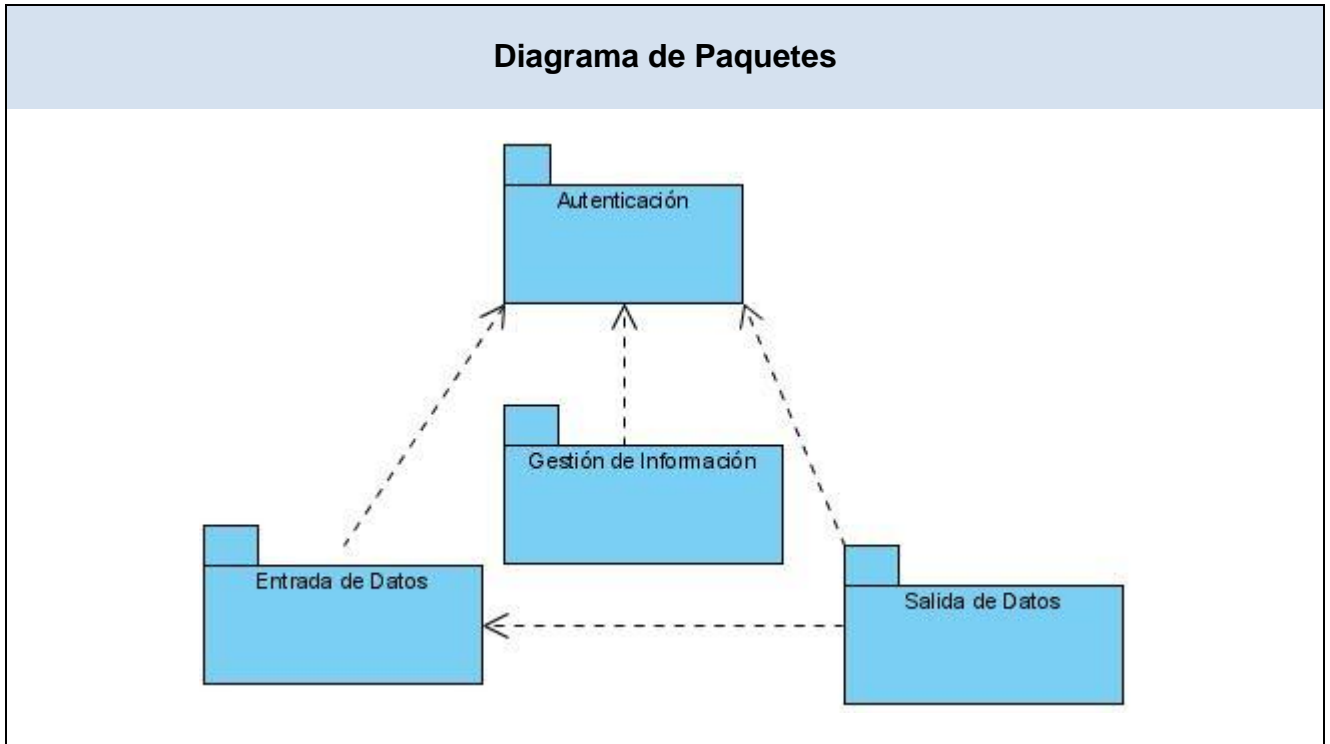


Figura 2.3: Diagrama de Paquetes

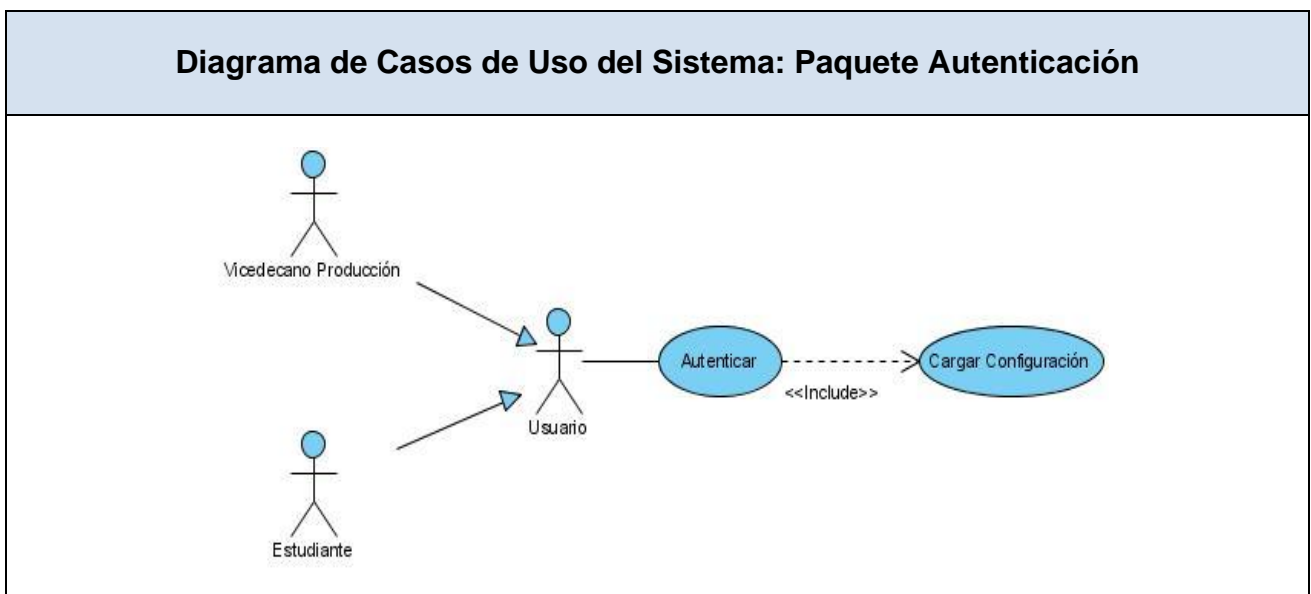


Figura 2.4: Diagrama de casos de uso del sistema Paquete Autenticación

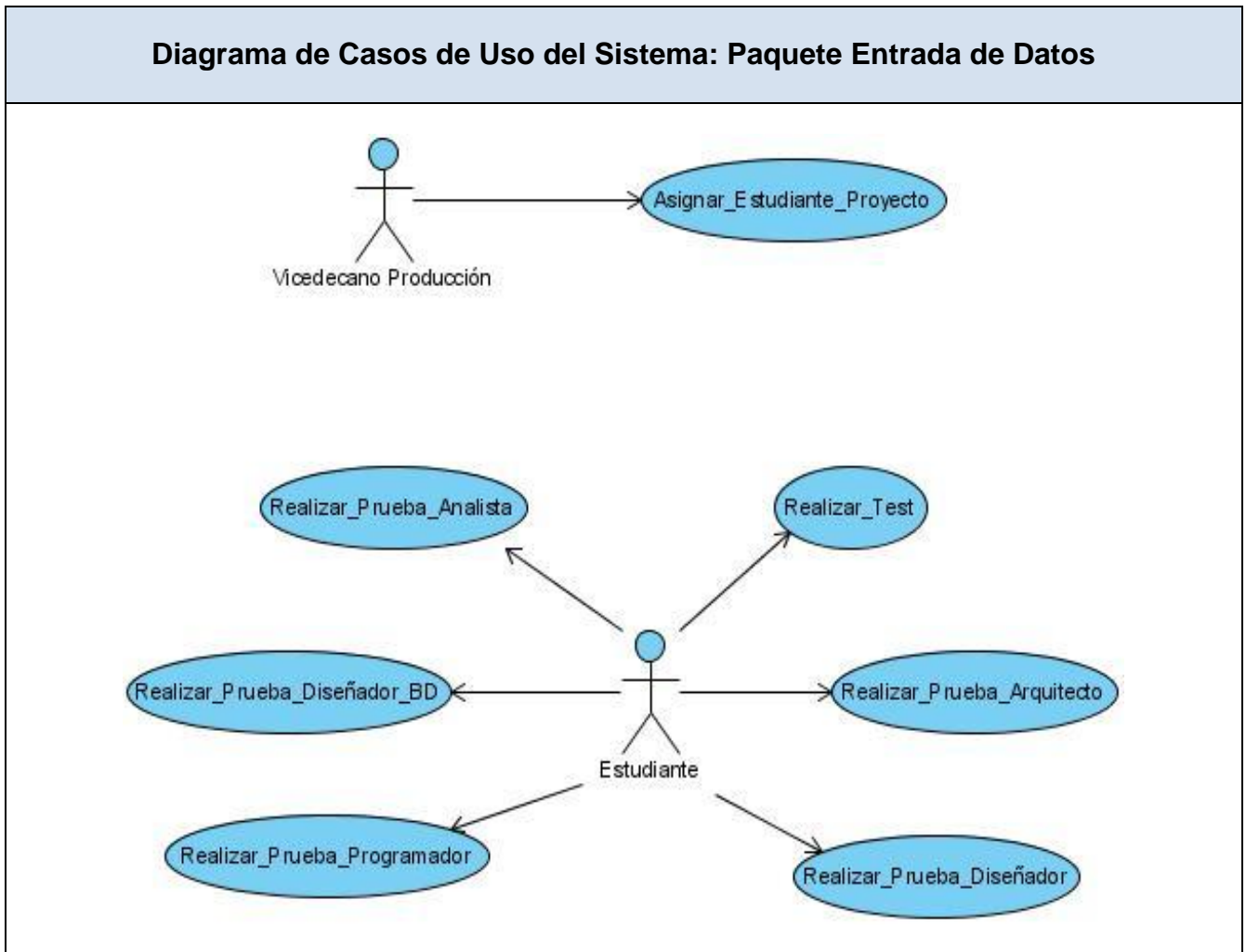


Figura 2.5: Diagrama de casos de uso del sistema Paquete Entrada de Datos

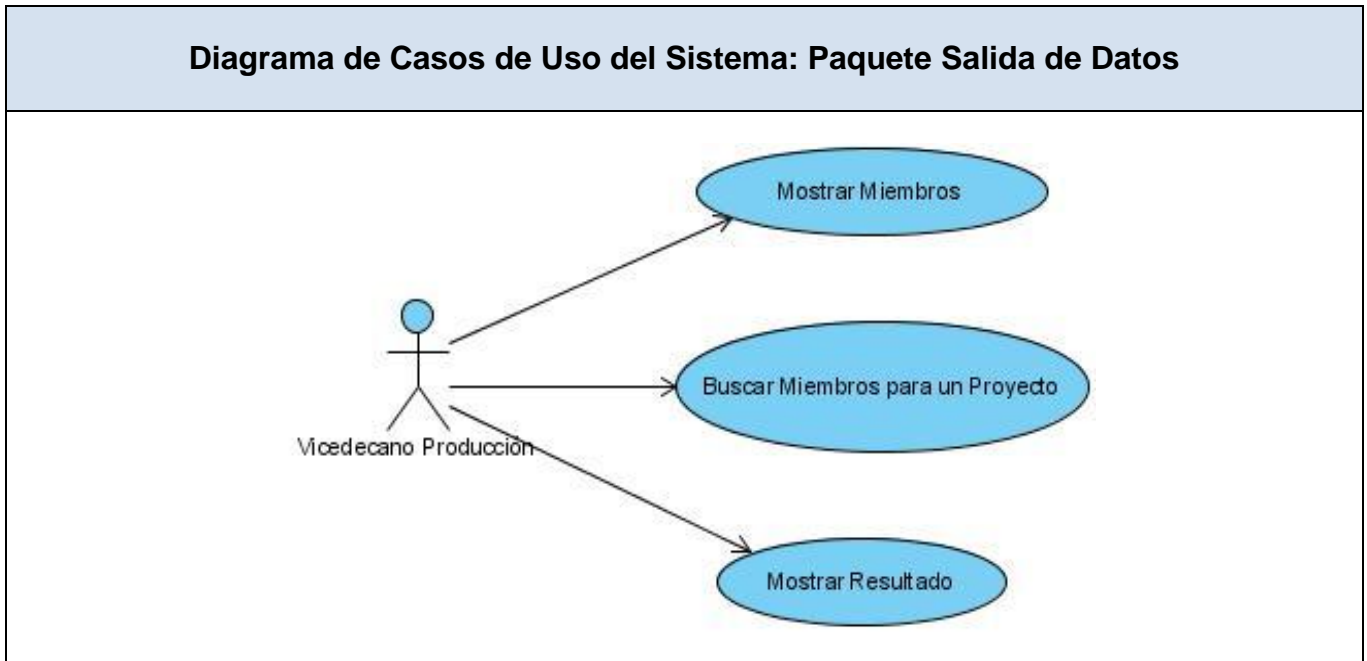


Figura 2.6: Diagrama de casos de uso del sistema Paquete Salida de Datos

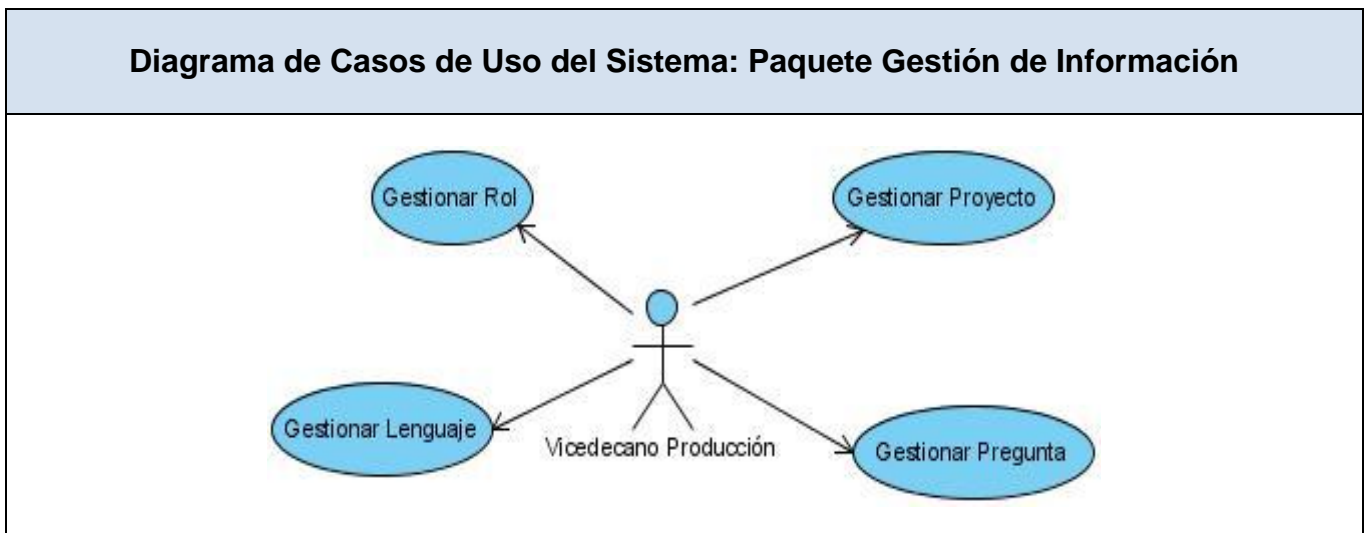


Figura 2.7: Diagrama de casos de uso del sistema Gestión de Información

2.7 Breve descripción de los casos de uso del sistema

CU-1	Autenticar Usuario
Actor	Usuario (Inicia)
Descripción	El caso de uso se inicia cuando el usuario (que puede ser vicedecano o el estudiante) introduce los datos para autenticarse, se verifica que los datos introducidos son correctos. En caso contrario no se permite la autenticación del usuario.
Referencia	R1

Tabla 2.2: Descripción breve del CU Autenticar Usuario

CU-2	Cargar Configuración
Actor	Usuario (Inicia)
Descripción	El caso de uso es una inclusión del CU Autenticar Usuario, este se inicia cuando es llamado, carga los elementos de configuración definidos para el usuario autenticado y obtiene de la BD los privilegios a los que tiene acceso.
Referencia	R1,R2

Tabla 2.3: Descripción breve del CU Cargar Configuración

CU-3	Gestionar Proyecto
Actor	Vicedecano Producción (Inicia)
Descripción	El caso de inicio se inicia cuando se necesita hacer uno de las 3 opciones que permite el CU Gestionar Proyecto: Crear Proyecto, listar proyectos y eliminar proyecto.
Referencia	R1,R2, R6

Tabla 2.4: Descripción breve del CU Gestionar Proyecto.

CU-4	Gestionar Rol
Actor	Vicedecano Producción (Inicia)
Descripción	El caso de inicio se inicia cuando se necesita hacer uno de las 2 opciones que permite el CU Gestionar Rol: Adicionar Rol y Eliminar .
Referencia	R1, R2, R8

Tabla 2.5: Descripción breve del CU Gestionar Rol

CU-5	Gestionar Lenguaje
Actor	Vicedecano Producción (Inicia)
Descripción	El caso de inicio se inicia cuando se necesita hacer uno de las 2 opciones que permite el CU Gestionar Lenguaje: Adicionar Lenguaje y Eliminarlo.
Referencia	R1, R2, R9

Tabla 2.6: Descripción breve del CU Gestionar Lenguaje

CU-6	Gestionar Pregunta
Actor	Vicedecano Producción (Inicia)
Descripción	El caso de inicio se inicia cuando se necesita hacer uno de las 3 opciones que permite el CU Gestionar Pregunta: Adicionar pregunta, Listar Preguntas y Eliminar Pregunta.
Referencia	R1, R2, R7

Tabla 2.7: Descripción breve del CU Gestionar Pregunta

CU-7	Asignar estudiantes a Proyecto
Actor	Vicedecano Producción (Inicia)
Descripción	El caso de uso permite asignarle a los proyectos de la facultad los estudiantes seleccionados. El vicedecano busca el proyecto y el estudiante que quiere asignar respectivamente. En la base de datos se comprueba que el estudiante no esté asignado a ningún otro proyecto y entonces se asigna al proyecto correspondiente de forma manual.
Referencia	R1, R2, R10

Tabla 2.8: Descripción breve del CU Asignar Estudiante

CU-8	Realizar Test
Actor	Estudiante (Inicia)
Descripción	El caso de uso se inicia cuando el estudiante realiza un test que posibilita recoger información acerca de los conocimientos que

	domine en cuanto a las diferentes herramientas a utilizar en el desarrollo de un proyecto determinado así como sus aptitudes para el mismo.
Referencia	R1, R2, R3

Tabla 2.9: Descripción breve del CU Realizar Test.

CU-9	Realizar Prueba (Roles de: arquitecto, analista, diseñador gráfico, diseñador de BD y programador).
Actor	Estudiante (Inicia)
Descripción	El caso de uso se inicia cuando el estudiante realiza una prueba de un determinado rol donde se evalúa los conocimientos necesarios que debe dominar para ingresar a un proyecto con este rol.
Referencia	R1, R2, R4

Tabla 2.10: Descripción breve del CU Realizar Prueba

CU-10	Mostrar resultado
Actor	Vicedecano Producción (Inicia)
Descripción	El vicedecano de producción selecciona la opción “Mostrar Resultado” y escoge el grupo al que pertenece el estudiante(s) de quien quiere ver el resultado y una vez seleccionado(s) presiona el botón mostrar. Se obtiene de la BD el listado de los resultados que cumplan con los criterios de búsqueda especificados y se muestra el mismo.
Referencia	R1, R2, R12

Tabla 2.11: Descripción breve del CU Mostrar Resultado.

CU-11	Mostrar Miembros
Actor	Vicedecano Producción (Inicia)
Descripción	El vicedecano de producción selecciona la opción “Mostrar Miembros” y se muestra una interfaz con los criterios de búsqueda (Por Rol o por Proyecto). Se obtiene de la BD el listado de los

	resultados que cumplan con los criterios de búsqueda especificados mostrándose el mismo.
Referencia	R1, R2, R11

Tabla 2.12: Descripción breve del CU Mostrar Miembros.

2.8 Descripción de los casos de uso del sistema expandidos

Ver Anexo No.1

2.9 Conclusiones

Para la realización de un software con calidad es necesario tener un conocimiento basto de los principales procesos que intervienen en la investigación y además una amplia visión del desarrollo del proyecto. Por lo cual en este capítulo se reflejan los principales procesos del negocio, los requerimientos funcionales y no funcionales que representan las características fundamentales que requiere la aplicación a desarrollar, la propuesta del software y el diagrama de casos de uso del sistema con la descripción detallada de cada uno de ellos.

CAPÍTULO 3: “DISEÑO”

3.1 Introducción

Tras la definición y descripción, en el anterior capítulo, de las funcionalidades deseadas y necesarias del sistema propuesto; se hace necesario definir cómo se desarrollará.

Este capítulo tiene el objetivo de plantear la concepción general del diseño del sistema propuesto y cómo se implementa éste. Así, se presentan los diagramas de clases Web que detallan la interacción de las distintas páginas; se estructura la información que se desea persista a través del diseño de la base de datos.

3.2 Diseño

3.2.1 Diagramas de interacción.

Ver Anexo 2.

3.2.2 Diagramas de clases del Diseño

Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia, todo el código que irá creando las páginas, así como el contenido dinámico de estas una vez que estén en el navegador del cliente. En el caso de las aplicaciones Web, el diagrama de clases representa las colaboraciones que ocurren entre las páginas, donde cada página lógica puede ser representada como una clase, es muy importante pues estos son los artefactos que se necesitan modelar para que el desarrollador los implemente y obtener así el producto final con la calidad requerida.

Al tratar de utilizar el diagrama de clases tradicional para modelar aplicaciones Web surgen varios problemas, por lo cual los especialistas del Rational plantearon la creación de una extensión al modelo de análisis y diseño que permitiera representar el nivel de abstracción adecuado y la relación con los restantes artefactos de UML.

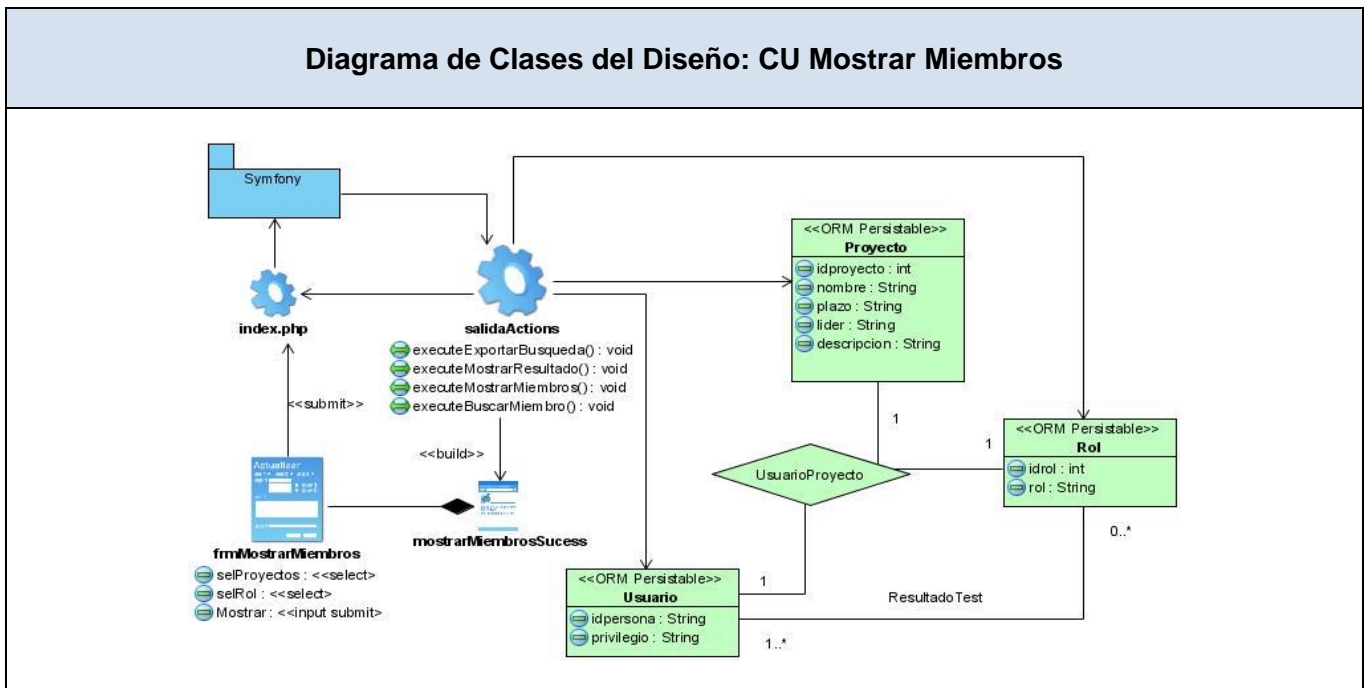


Figura 3.1: Diagrama de clases del diseño CU Mostrar Miembros

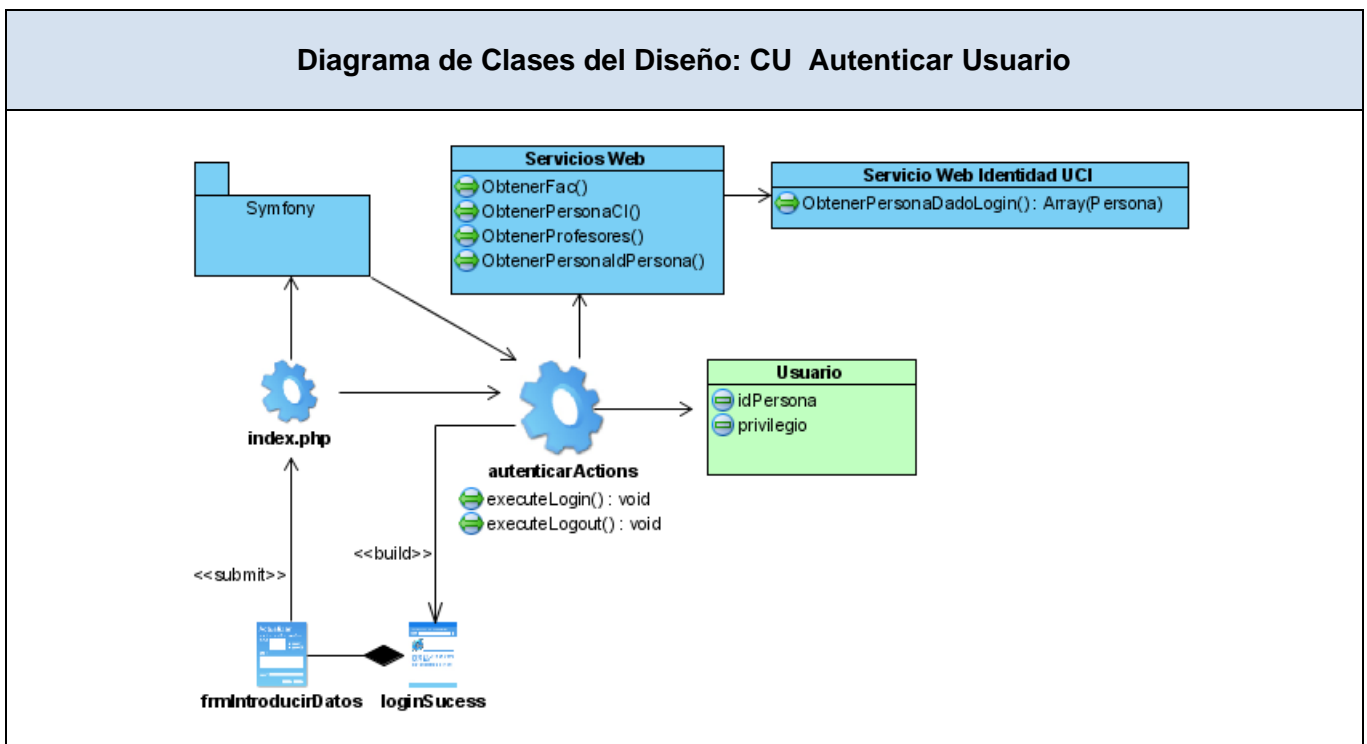


Figura 3.2: Diagrama de clases del diseño CU Autenticar Usuario

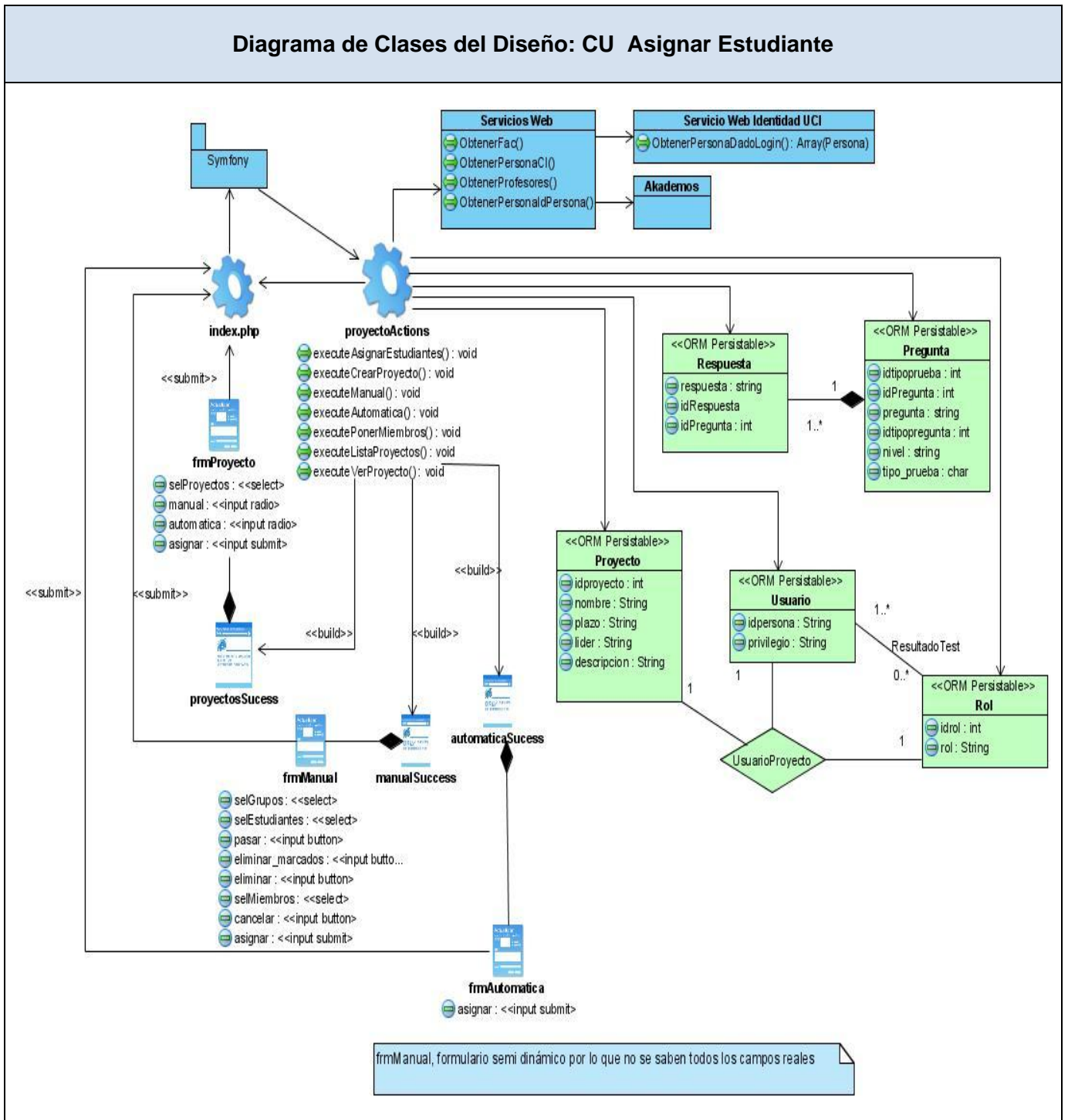


Figura 3.3: Diagrama de clases del diseño CU Asignar Estudiante

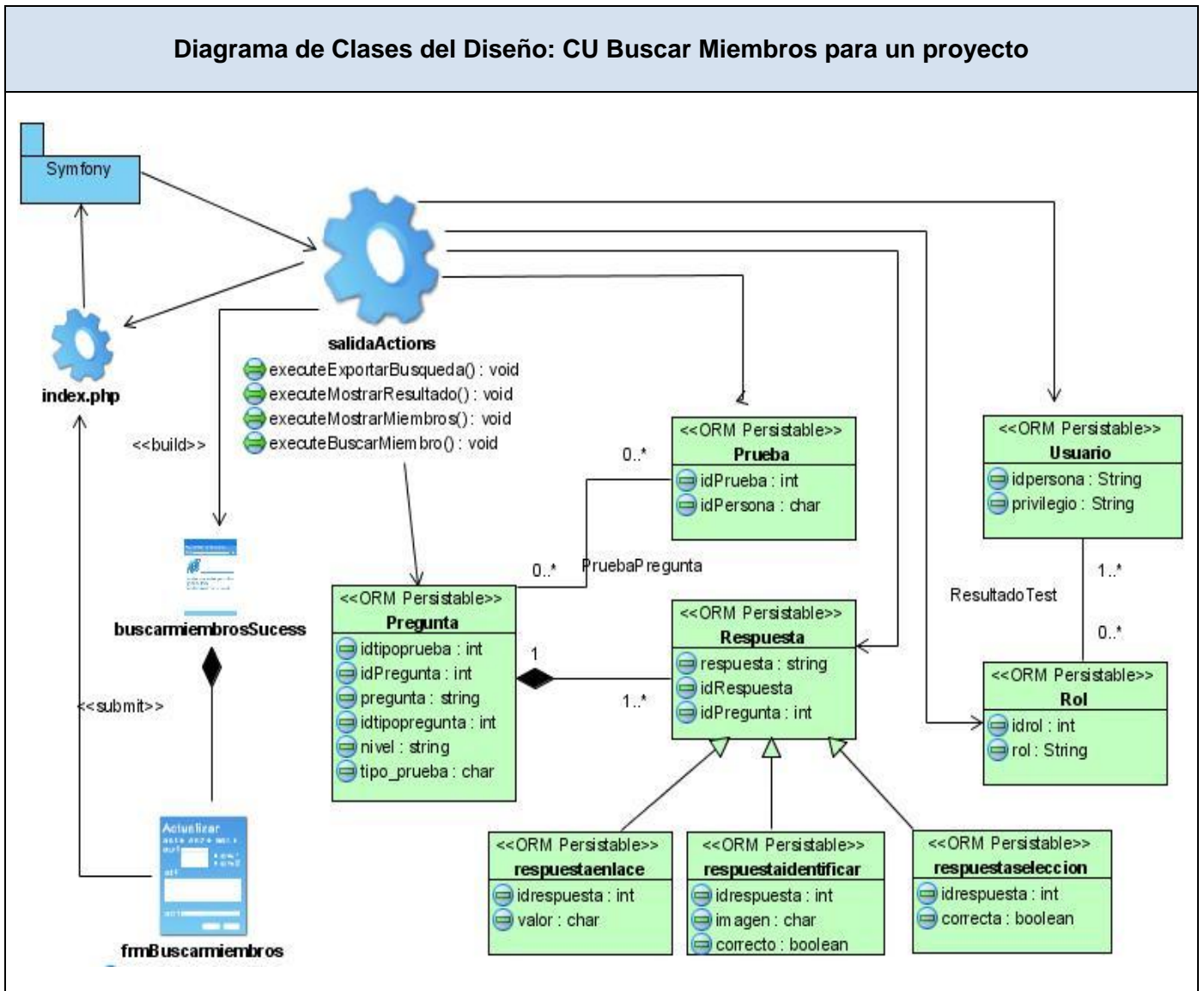


Figura 3.4: Diagrama de clases del diseño CU Buscar Miembros para un Proyecto

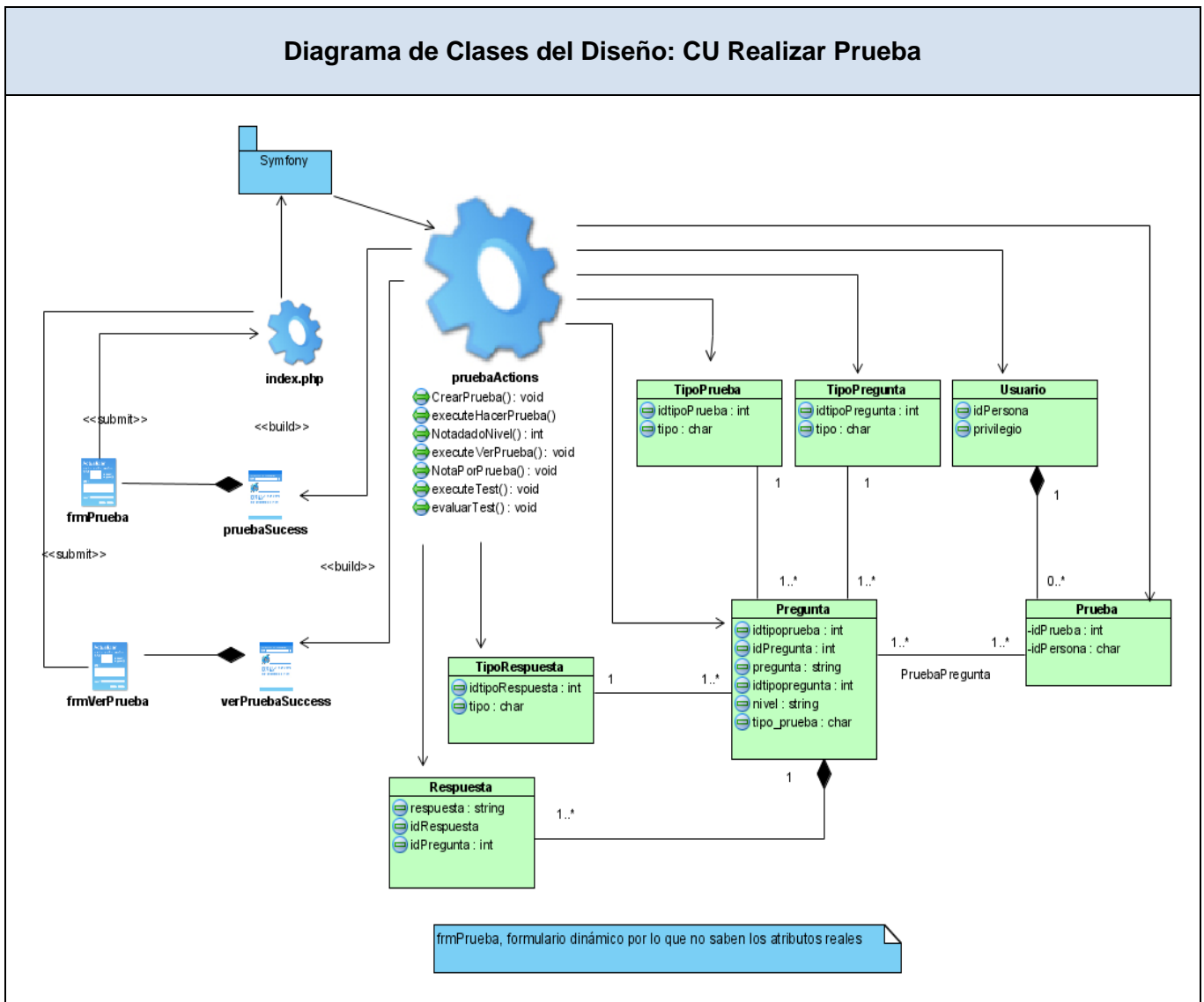


Figura 3.5: Diagrama de clases del diseño CU Realizar Prueba

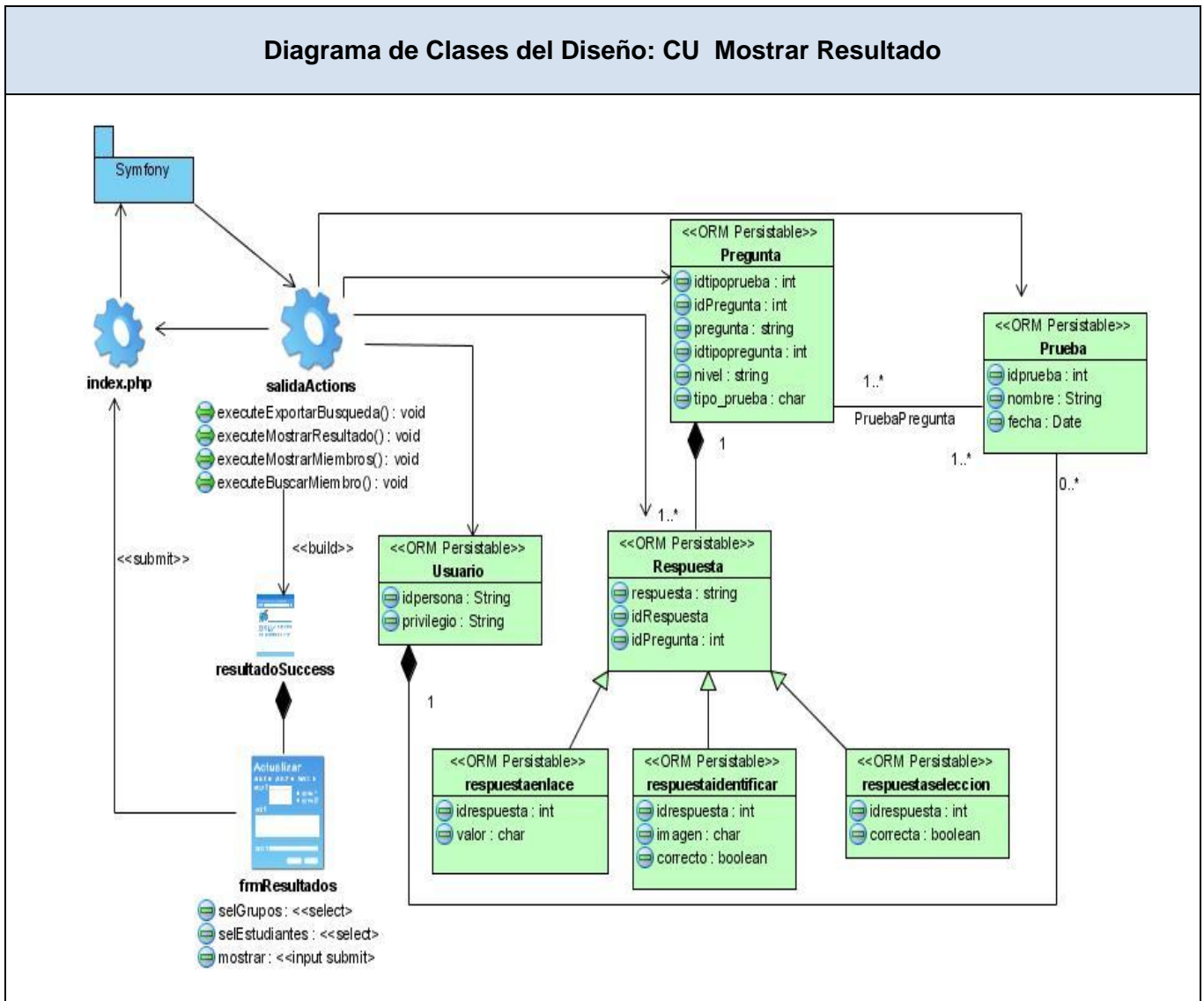


Figura 3.6: Diagrama de clases del diseño CU Mostrar Resultado.

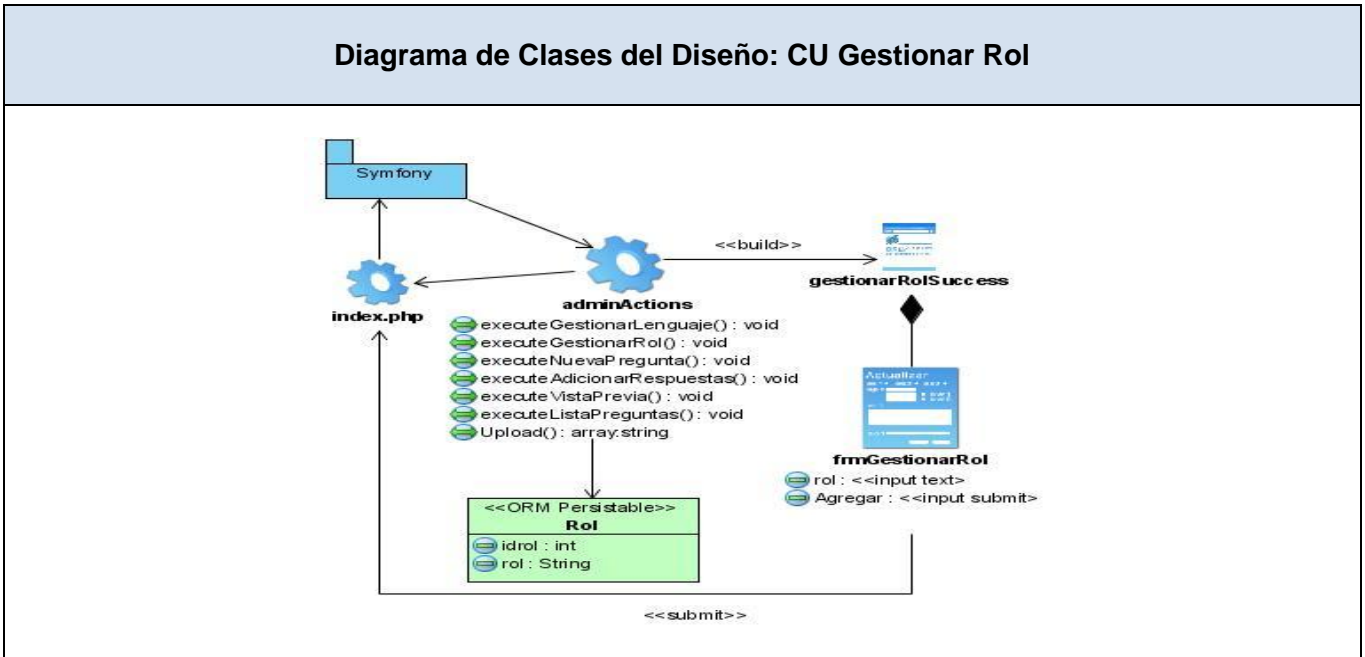


Figura 3.7: Diagrama de clases del diseño CU Gestionar Rol

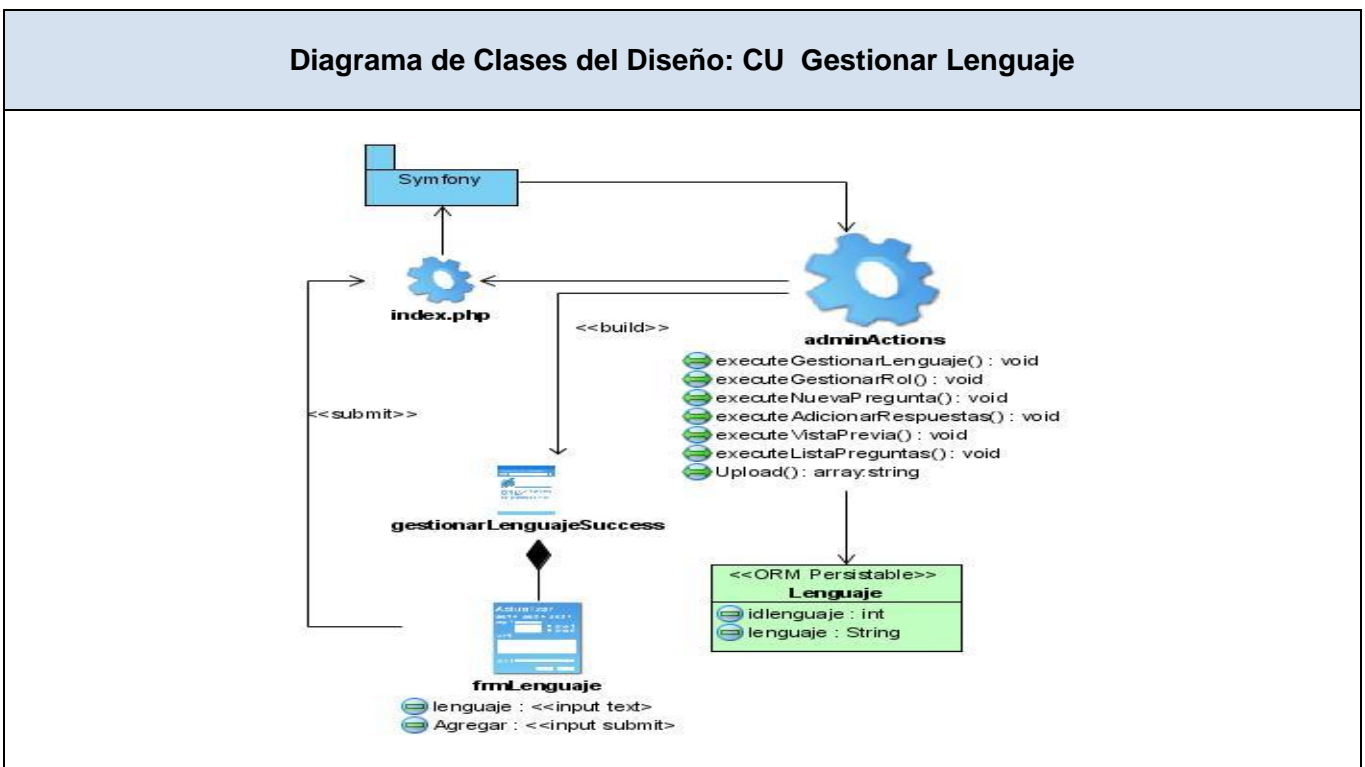


Figura 3.8: Diagrama de clases del diseño CU Gestionar Lenguaje

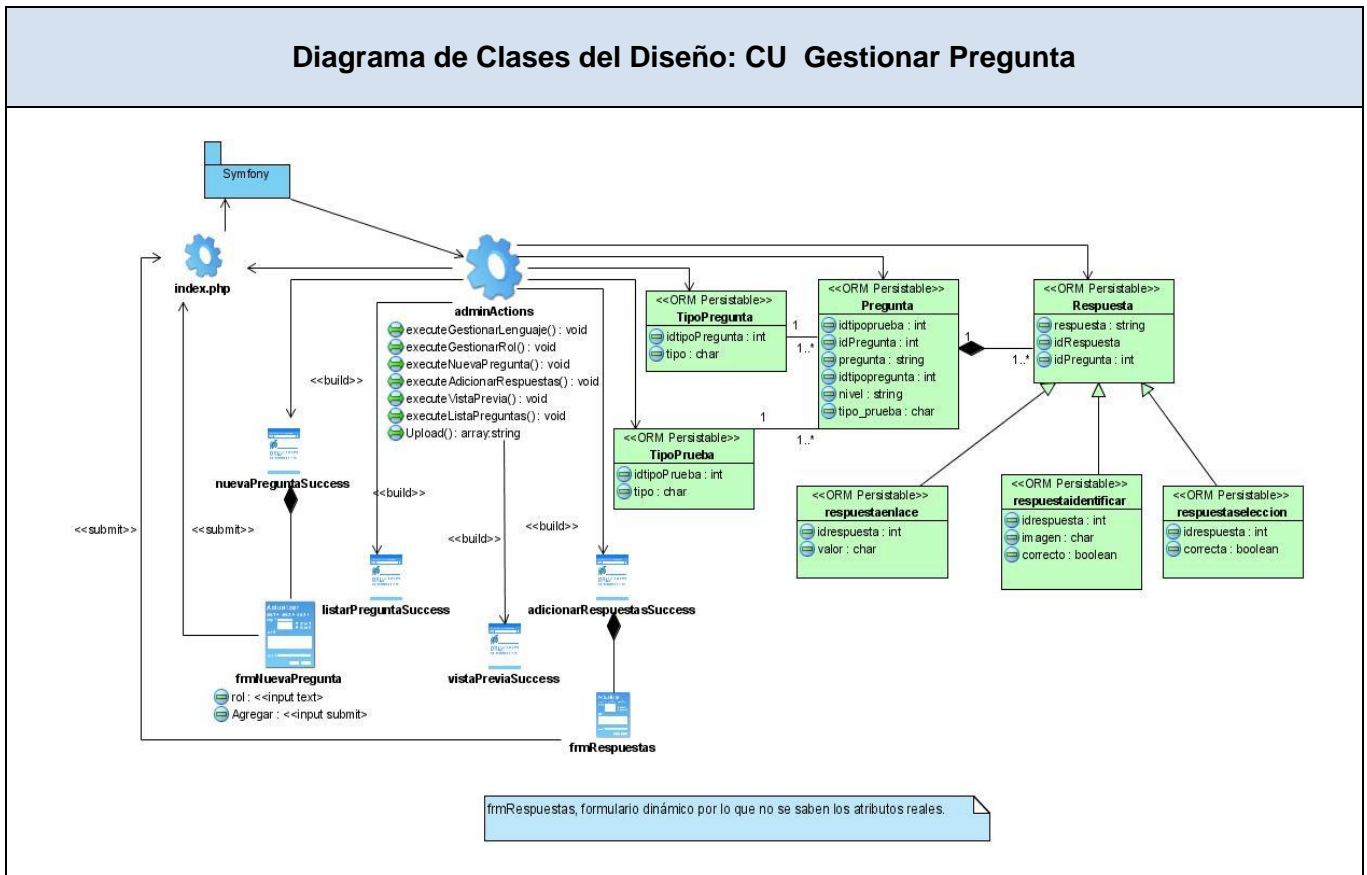


Figura 3.9: Diagrama de clases del diseño CU Gestionar Pregunta

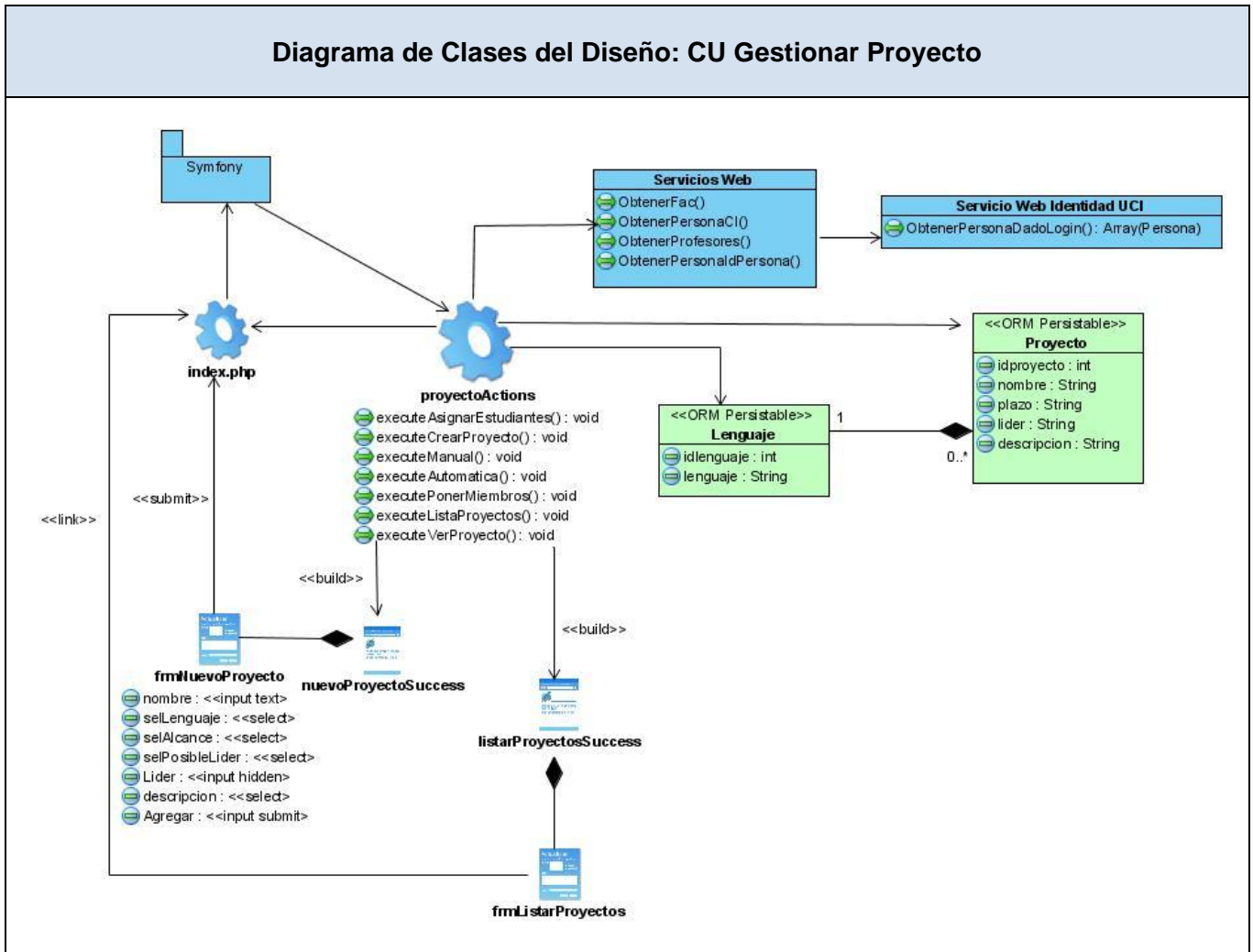


Figura 3.10: Diagrama de clases del diseño CU Gestionar Proyecto

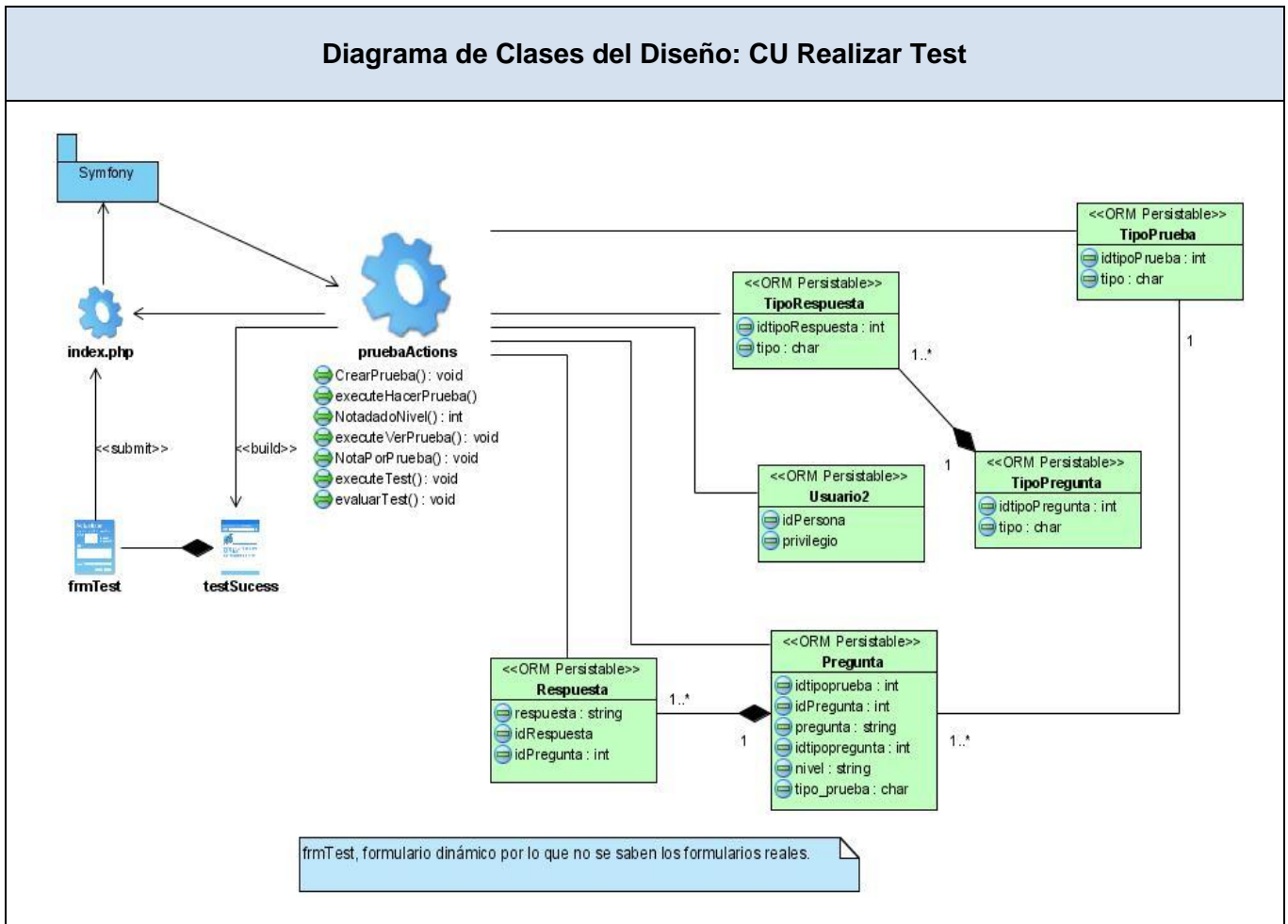


Figura 3.11: Diagrama de clases del diseño CU Realizar Test

3.3 Diseño de la base de datos

3.3.1 Modelo Lógico de Datos (diagrama de clases persistentes).

Las clases persistentes son las clases que necesitan ser capaz de guardar su estado en un medio permanente, la necesidad de guardar su estado esta dado por el almacenamiento físico permanente de la información de la clase, para la copia de seguridad en caso del fracaso del sistema, o para el intercambio de información.

A continuación se muestra el diagrama de clases persistentes.

CAPÍTULO 4: “IMPLEMENTACION”

4.1 Introducción

En el presente capítulo se representan en términos de componentes los elementos del modelo del diseño definidos en el capítulo anterior. A continuación se modela un diagrama de despliegue y un diagrama de componentes, con los componentes principales que constituyen esta aplicación. Se realiza una breve descripción de cada componente, especificando su propósito y contenido.

4.2 Diagrama de Despliegue

Los Diagramas de Despliegue se usan para modelar la configuración de los elementos de procesado en tiempo de ejecución y de los componentes, procesos y objetos de software que viven en ellos. En el diagrama 'Despliegue', se modelan nodos físicos y las asociaciones de comunicación que existen entre ellos. Para cada nodo, se puede indicar qué instancias de componentes viven o corren (se ejecutan) en el nodo.

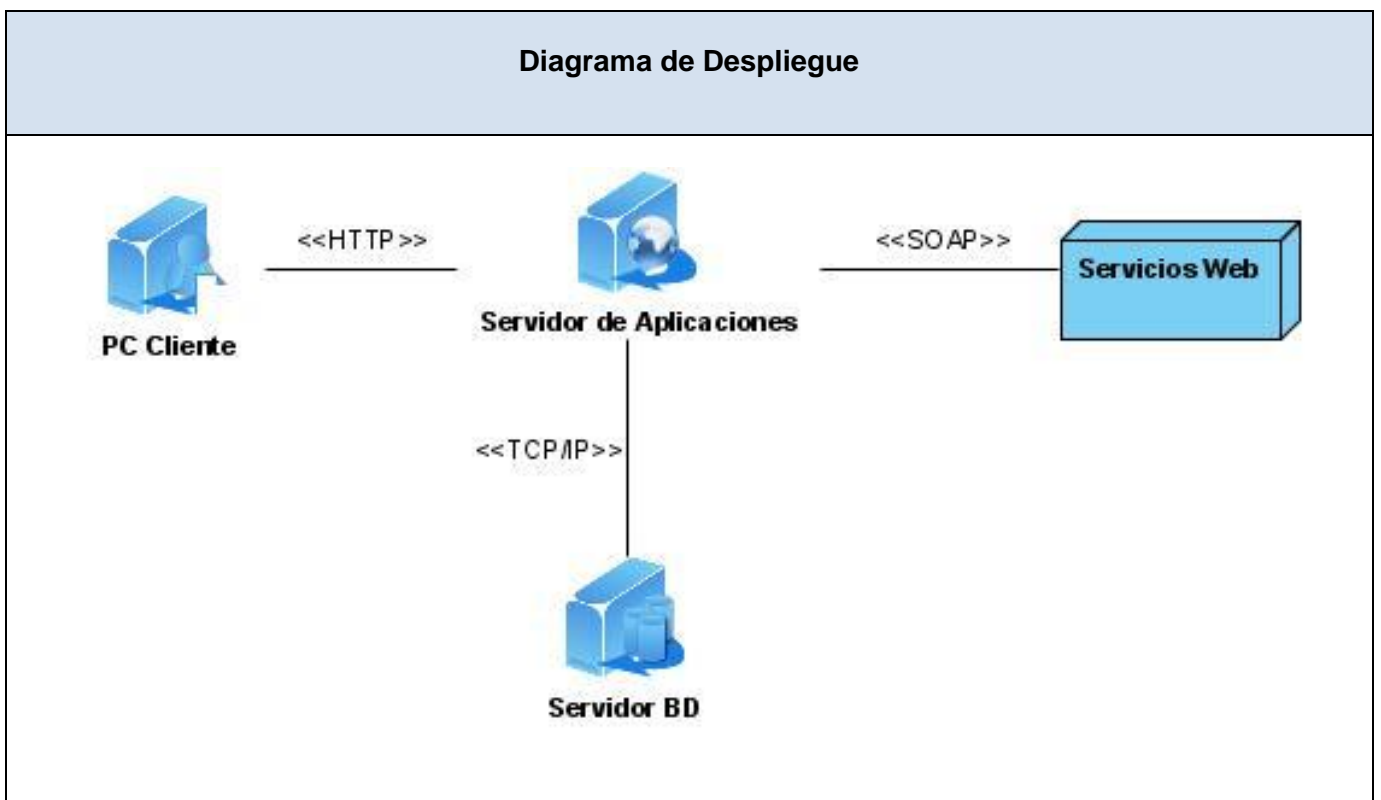


Figura 4.1: Diagrama Despliegue

4.3 Diagrama de Componentes.

Los diagramas de componentes describen los elementos físicos del sistema y su relación, muestran las dependencias lógicas entre componentes software, sean éstos componentes fuentes, binarios o ejecutables. El diagrama de componente hace parte de la vista física de un sistema, la cual modela la estructura de implementación de la aplicación por sí misma, su organización en componentes y su despliegue en nodos de ejecución. Esta vista proporciona la oportunidad de establecer correspondencias entre las clases y los componentes de implementación y nodos. La vista de implementación se representa con los diagramas de componentes.

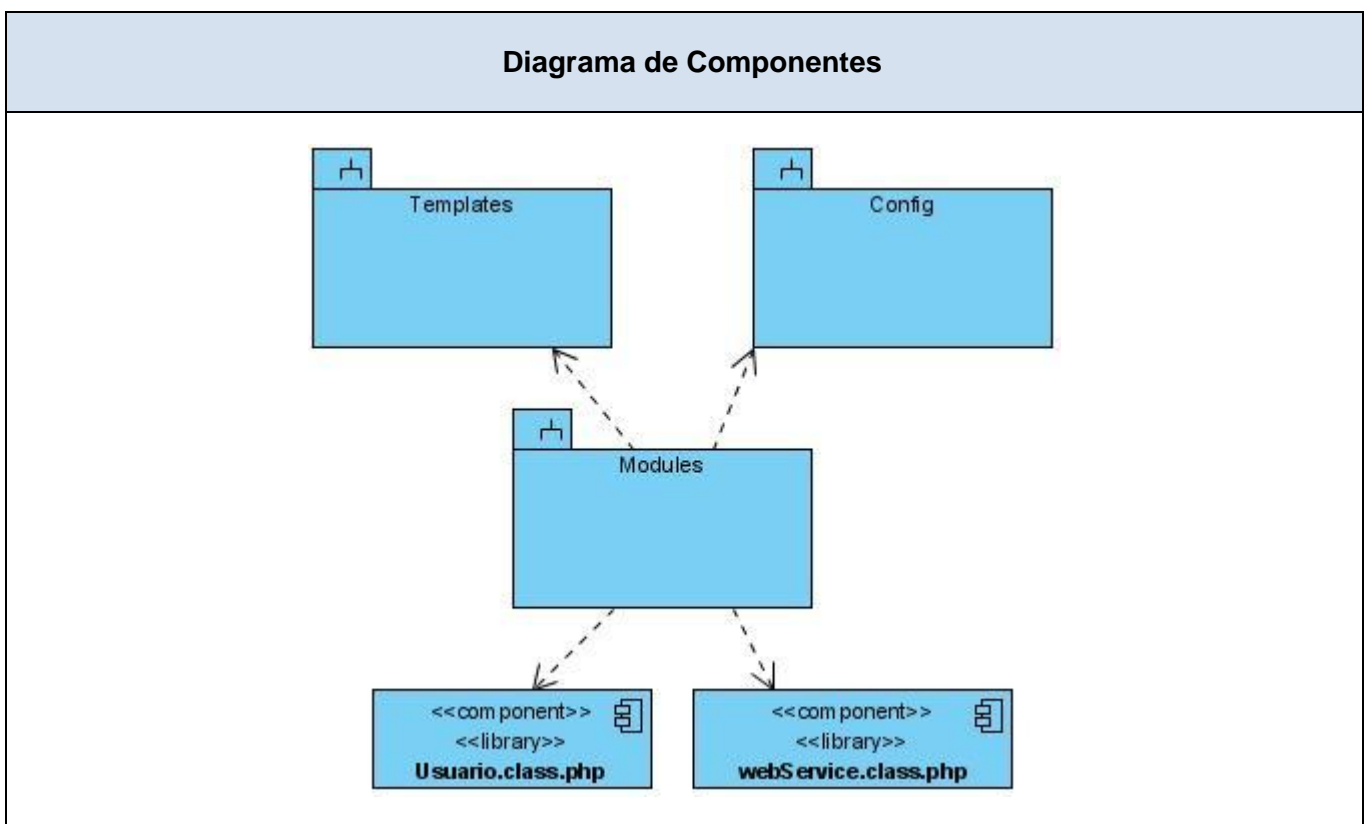


Figura 4.2: Diagrama Componentes

4.3.1 Descripción de los Componentes.

a. Subsistema Templates

Contenido:

- menu.php.
- vicedecano.php
- layout.php
- layout-login.php
- layoutAjax.php
- layout1.php
- usuario.php

b. Subsistema Config

Contenido:

- autoload.yml.
- config.php
- filters.yml
- logging.yml
- security.yml
- factories.yml

c. Subsistema Modules

Contenido:

- autenticar
- proyecto
- salida
- admin
- común
- prueba

Todos estos subsistemas contienen dentro los subsistemas mencionados a continuación:

- config
- templates

- actions
- lib
- valídate

d. Subsistema lib

Contiene los ficheros

- webService.class.php
- Usuario.class.php

4.4Conclusiones

Con la culminación de este capítulo se han obtenido los diferentes diagramas que dieron lugar a la obtención del modelo de implementación final. El modelo de despliegue, describe las configuraciones sobre las cuales deberá implementarse el sistema y el diagrama de componentes que ilustra los componentes de software que se usan para construir el sistema .en el cual se describe su organización y dependencia entre nodos físicos en los que funcionará a aplicación.

CAPÍTULO 5: “ESTUDIO DE LA FACTIBILIDAD”

5.1 Introducción.

Para la realización de un proyecto es de suma importancia el análisis del costo y los beneficios que reportará. Como resultado de este análisis se obtiene el tiempo de desarrollo en meses, costo y la cantidad de personas que se necesitan para desarrollar el proyecto.

Para la estimación del tamaño de un sistema a partir de sus requerimientos, entre las técnicas más difundidas se encuentran el Análisis de Puntos de Función, COCOMO II y una variante más reciente Análisis de Puntos de Casos de Uso.

En este capítulo se describe la estimación de costos del sistema propuesto y sus beneficios empleando esta última.

5.2 Estimación basada en análisis de Puntos de Casos de Uso.

La estimación mediante el análisis de Puntos de Casos de Uso es un método propuesto originalmente por Gustav Karner de Objectory AB, y posteriormente refinado por muchos otros autores.

Se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores. A continuación, se detallan los pasos a seguir para la aplicación de éste método.

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

Se calcula a partir de la siguiente ecuación:

$$\text{UUCP} = \text{UAW} + \text{UUCW}$$

Donde:

UUCP: Puntos de Casos de Uso sin ajustar

UAW: Factor de Peso de los Actores sin ajustar

UUCW: Factor de Peso de los Casos de Uso sin ajustar

Actor de Peso de los Actores sin ajustar (UAW)

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

Para calcular UAW

Tipo	Descripción	Peso	Cant*peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API, Application Programming Interface)	1	0*1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto	2	1*2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica	2	2*2
Total			6

Tabla 5.1: Peso de los Actores sin ajustar.

UAW=6.

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

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

Para calcular UUCW

Tipo	Descripción	Peso	Cant * peso
Simple	El Caso de Uso contiene de 1 a 3 transacciones	5	16*5
Medio	El Caso de Uso contiene de 4 a 7 transacciones	10	2*10
Complejo	El Caso de Uso contiene más de 8 transacciones	15	0*15
Total			100

Tabla 5.2: Peso de los Casos de Uso sin ajustar.

UUCW = 100.

UUCP = UAW + UUCW = 6+ 100 = 106

2. Cálculo de Puntos de Casos de Uso ajustados

Una vez que se tienen los Puntos de Casos de Uso sin ajustar, se debe ajustar éste valor mediante la siguiente ecuación:

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

Donde:

UCP: Puntos de Casos de Uso ajustados

UUCP: Puntos de Casos de Uso sin ajustar

TCF: Factor de complejidad técnica

EF: Factor de ambiente

Factor de complejidad técnica (TCF)

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante. En la siguiente tabla se muestra el significado y el peso de cada uno de éstos factores:

Factor	Descripción	Peso	Valor	Σ (Peso _i * Valor _i)
T1	Sistema distribuido	2	3	6
T2	Tiempo de respuesta	1	4	4
T3	Eficiencia del usuario final	1	4	4
T4	Procesamiento interno complejo	1	1	1
T5	El código debe ser reutilizable	1	5	5
T6	Facilidad de instalación	0.5	5	2.5
T7	Facilidad de uso	0.5	4	2.0
T8	Portabilidad	2	4	8
T9	Facilidad de cambio	1	4	4
T10	Concurrencia	1	5	5
T11	Incluye objetivos especiales de seguridad	1	3	3
T12	Provee acceso directo a terceras partes	1	2	2

T13	Se requieren facilidades especiales de entrenamiento a los usuarios.	1	2	2
Total				48.5

Tabla 5.3: Factor de complejidad técnica.

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

$$TCF = 0.6 + 0.01 \times 48.5 = 1.085$$

Factor de ambiente (EF)

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del Factor de ambiente. El cálculo del mismo es similar al cálculo del Factor de complejidad técnica, es decir, se trata de un conjunto de factores que se cuantifican con valores de 0 a 5.

Para Calcular EF.

Factor	Descripción	Peso	Valor	$\Sigma (\text{Peso}_i * \text{Valor}_i)$
E1	Familiaridad con el modelo de proyecto utilizado	1.5	0	0
E2	Experiencia en la aplicación	0.5	0	0
E3	Experiencia en orientación a objetos	1	3	3
E4	Capacidad del analista líder	0.5	4	2.0
E5	Motivación	1	5	5
E6	Estabilidad de los requerimientos	2	3	6
E7	Personal part-time	-1	3	-3
E8	Dificultad del lenguaje de programación	-1	3	-3
Total				10

Tabla 5.4: Factor de ambiente.

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

$$EF = 1.4 - 0.03 * 10 = 1.1$$

$$UUCP = 64.$$

$$TCF = 1.085.$$

$$EF = 1.1$$

$$UCP = UUCP * TCF * EF = 106 * 1.085 * 1.1 = 126.511$$

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

Para la fase de implementación

El esfuerzo en horas-hombre viene dado por:

$$E = UCP * CF$$

Donde:

E: Esfuerzo estimado en horas-hombre

UCP: Puntos de Casos de Uso ajustados

CF: Factor de conversión

Este método proporciona una estimación del esfuerzo en horas-hombre contemplando sólo el desarrollo de la funcionalidad especificada en los casos de uso.

Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por debajo del valor medio (3), para los factores E1 a E6.

Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por encima del valor medio (3), para los factores E7 y E8.

CF = 20 horas-hombre (si Total_{EF} ≤ 2)

CF = 28 horas-hombre (si Total_{EF} = 3 ó Total_{EF} = 4)

CF = abandonar o cambiar proyecto (si Total_{EF} ≥ 5)

Total_{EF} = Cant EF < 3 (entre E1 –E6) + Cant EF > 3 (entre E7, E8)

Total_{EF} = 2 +

Total_{EF} = 2

CF = 20 horas-hombre (porque Total_{EF} = 2)

UCP = 126.511

CF = 20 horas-hombre.

$$E = UCP * CF = 126.511 * 20 \text{ horas-hombre} = 2530.22 \sim 2530$$

4. Calcular esfuerzo de todo el proyecto

Actividad	% esfuerzo	Valor esfuerzo
Análisis	10%	632 horas-hombre
Diseño	20%	1265 horas-hombre
Implementación	40%	2530 horas-hombre
Prueba	15%	949 horas-hombre
Sobrecarga	15%	949 horas-hombre
Total	100%	6325 horas-hombre

Tabla 5.5: Esfuerzo de todo el proyecto.

$E_T = 6325$ horas-hombre y se estima que cada mes tiene como promedio 240 horas laborables eso daría un $E_T = 26.35$ mes-hombre.

Se estima que una persona podrá realizar el problema analizado en 26 meses y medio aproximadamente.

Costo del Proyecto.

Se asume como salario promedio mensual \$100.00

CH: Cantidad de hombres.

Tiempo: Tiempo total del proyecto.

CH = 2 hombres

CHM = 2 * Salario Promedio

CHM = 200.00 \$/mes

Costo = **CHM** * **ET** / **CH**

Costo = 200.00 * 26.35 / 2

Costo = \$2635

Tiempo = **ET** / **CH**

Tiempo = 26.35 / 2

Tiempo = 13.18

De los resultados obtenidos se estima que con dos hombres trabajando en el proyecto, el mismo se desarrolla en 13 meses y medio aproximadamente, con un costo total estimado en \$ 2635.

5.3 Beneficios tangibles e intangibles.

El sistema de gestión para miembros para un proyecto productivo no es un producto con fines comerciales, su principal objetivo es resolver los problemas que existen con el desarrollo de este proceso en la facultad.

El beneficio fundamental del sistema es contar con una aplicación Web flexible, dinámica y de interfaz agradable que permita llevar a cabo todos los procesos inmersos en el objetivo principal del trabajo de una forma eficiente y rápida.

Por tanto, los beneficios inmediatos son intangibles:

1. Fácil y rápido acceso a la gestión de miembros de un o para un proyecto.
2. Optimo funcionamiento de la aplicación.
3. Implementada bajo la nueva arquitectura concebida en el centro SOA.

5.4 Análisis de costos y beneficios.

Los beneficios pueden ser económicos y de orden social, estos últimos son de tanta importancia como los primeros. El sistema que se propone está dirigido a la Universidad de las Ciencias Informáticas específicamente a la facultad viendo la posibilidad de que sea valorado en el resto de las facultades, para la satisfacción del personal por tanto su mayor beneficio es de orden social. Contribuyendo así a aumentar la eficiencia y a disminuir el tiempo de respuesta del sistema y obtener estos resultados con mayor rapidez y certeza.

La tecnología utilizada para el desarrollo del sistema es totalmente libre, por tanto no hay que incurrir en gastos en el pago de licencias de uso. El sistema es portable por lo que un cambio de plataforma para la implantación del mismo es viable y factible, y no hay que incurrir en muchos cambios.

Analizando el costo del proyecto, los numerosos beneficios que reporta, detallados con anterioridad, se puede concluir que su implementación es realmente factible.

5.5 Conclusiones.

En este capítulo se describió el estudio de factibilidad correspondiente al sistema propuesto, utilizando la técnica de estimación por Puntos de Caso de Uso, la cual resulta muy efectiva para estimar el esfuerzo requerido en el desarrollo de los primeros Casos de Uso de un sistema, si se sigue una aproximación iterativa como el Proceso Unificado de Rational. En éste tipo de

aproximación, los primeros Casos de Uso a desarrollar son los que ejercitan la mayor parte de la arquitectura del software y los que a su vez ayudan a mitigar los riesgos más significativos. Teniendo en cuenta el costo estimado y los beneficios que reportará al ser implantado dicha proposición, al contribuir a mejorar el proceso gestión de miembros de y para un proyecto productivo se llega a la conclusión de que es factible implementar la herramienta propuesta.

CONCLUSIONES

Llegado este punto se espera que el documento haya servido para la comprensión teórica de la situación problemática existente y su solución, así como el desarrollo de las diferentes etapas de la aplicación usando la metodología RUP.

El desarrollo de este trabajo de tesis está orientado a la concepción de una herramienta informática para la gestión de los proyectos productivos. El valor fundamental del sistema se expresa en la contribución a simplificar el trabajo y la demora que produce el procesamiento manual de la información y mejorar la gestión de las actividades en esta esfera.

Se alcanzó, satisfactoriamente, el objetivo propuesto: desarrollar una solución flexible y única de software que dé soporte a los procesos relacionados con la gestión de los proyectos productivos específicamente en la facultad 2; reafirmando así la utilidad y validez de emplear las tecnologías informáticas para apoyar las labores que se desarrollan en cualquier tipo de esfera.

Se ha demostrado la eficacia de los lenguajes y tecnologías utilizadas para el desarrollo del sistema.

Se realizó una base de datos, donde se almacena toda la información necesaria que se genera de los procesos relacionados con nuestro objetivo general, para de esta forma garantizar la veracidad y centralización de la misma. Se realizó el análisis, diseño e implementación del sistema.

RECOMENDACIONES

Se recomienda:

- Poner a prueba el sistema durante un período de tiempo significativo, para comprobar su desempeño y que las funcionalidades del sistema se correspondan con la actividad que se está gestionando.
- Continuar el estudio con el objetivo de añadir nuevas funcionalidades. Proponer, tras corroborar un desempeño exitoso, la utilización y generalización de este sistema en los diferentes lugares que se lleva este proceso en la Universidad.

REFERENCIAS BIBLIOGRÁFICAS

1. **T., Alhama B. R. y Martínez N.** Gestión de recursos humanos. *Pensamiento determinista vs Pensamiento integrador*. [En línea] [Citado el: 3 de Diciembre de 2007.]
http://www.nodo50.org/cubasigloXXI/economia/alhama_300606.pdf .
2. **O., Fernández T. A. y Valdés P.** *La gestión del proyecto productivo SIS1.IP y la influencia de la producción en la docencia*. 2006.
3. **E, Bichi.** *El analista “en persona”*. 2004.
4. **Roberts A., Wallace W.** Gestión de proyecto. [En línea]
<http://ebsamericas.com/PDF/DemoGestioneProyectos.pdf>
5. **J., Palacio.** Gestión y procesos en empresas de software. [En línea] [Citado el: 4 de Diciembre de 2007.] http://www.navegapolis.net/files/articulos/gestion_y_procesos.pdf .
6. Metodologías de Desarrollo de Software. [En línea]
<http://www.scribd.com/doc/2050925/metodologias-de-desarrollo-software>
7. Metodologías de Desarrollo de Software. [En línea]
<http://www.scribd.com/doc/2050925/metodologias-de-desarrollo-software>
8. Metodologías RUP y XP . *[PROCESOS DE DESAROLLO]*. . [En línea]
<http://jackopc.blogspot.com/2007/05/metodologias-rup-y-xp-procesos-de.html>
9. **PHP, Programación en.** Programación en PHP. [En línea] [Citado el: 20 de Enero de 2008.] http://es.wikibooks.org/wiki/Programaci%C3%B3n_en_PHP
10. Lenguaje PHP. [En línea]
<http://www.virtual.unal.edu.co/cursos/sedes/manizales/4060029/lecciones/cap11-2.html>
11. Java Script. [En línea] <http://www.webestilo.com/javascript/>
12. **Orallo, Enrique Hernández.** El Lenguaje Unificado de Modelado (UML). [En línea]
<http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>
13. Software y programas. *Uso de una Herramienta de Modelado*. [En línea]
<http://softwareyprogramas.blogspot.com>
14. **Vizcaíno, Aurora, García, Felix Oscar and Caballero, Ismael.** Una Herramienta CASE para ADOO: Visual Paradigm. [En línea] <http://www.visual-paradigm.com/product/vpuml>

15. PDT: Eclipse + PHP. [En línea] . [En línea] <http://www.desarrolloweb.com/articulos/pdt-eclipse-php.html>
16. Encarta. Aplicación (informática). [En línea] <http://mx.encarta.msn.com>
17. Estructura SOA Alcance SOA . [En línea]
<http://www.tecnologiahechapalabra.com/datos/soluciones/negocio/articulo.asp?i=961>
18. *Conferencia 7 de Ingeniería del Software Patrones de Diseño*. s.l. : UCI, 2007.
19. La Oficina del W3C en España. *Guía Breve de Servicios Web*. [En línea]
20. **Valdés, Damián Pérez**. Maestros del Web. *Qué son las bases de datos?* [En línea]
21. **Fabien Potencier, François Zaninotto**. *Symfony, la guía definitiva*. [En línea]
www.librosweb.es


BIBLIOGRAFÍA

1. **Álvarez Romero, Eduardo and Pueyo, Daniel. 2004-2005.** Integration Definition For Funcion Modeling. [Online] 2004-2005. [Cited: Diciembre 12, 2007.] <http://dmi.uib.es/~burguera/download/IDEF0trabajo.doc>
2. **Alvarez, Miguel Angel.** Aptana Studio. [Online] [Cited: Diciembre 7, 2007.] <http://www.desarrolloweb.com/articulos/aptana-studio.html>
3. **Apache. 1995-2006.** Versión 2.2 de la documentación del Servidor de HTTP Apache. [Online] 1995-2006. [Cited: Enero 20, 2008.] <http://www.apache2.org/>
4. **Artiles Visual, Leticia, et al.** PERFECCIONAMIENTO DE LA GESTION UNIVERSITARIA EN EL ISCMH: Proyecto Tocaroro. [Online] [Cited: Enero 8, 2008.] http://www.ucmh.sld.cu/rhab/vol5_num1/rhcm10106.htm
5. **Departamento de Control de Calidad y Auditoría Informática. 2001.** Sistemas en Arquitectura Cliente/Servidor. [Subdirección de Sistemas] s.l.: Dirección General de Servicios de Cómputo Académico, 2001.
6. **DesarrolloWeb.** Qué es PHP. [Online] [Cited: Noviembre 17, 2007.] <http://www.desarrolloweb.com/articulos/392.php>
7. **Edinet. 2002.** Qué es XML? [Online] 2002. [Cited: Diciembre 12, 2007.] <http://www.edinet.com/sabia2.asp>
8. **Eguíluz Pérez, Javier. 2007.** Libros Web. Intoducción a JavaScript. [Online] 2007. [Cited: Diciembre 5, 2007.] <http://www.librosweb.es/javascript>
9. —. **2007.** Libros Web. Introducción a CSS. [Online] 2007. [Cited: Enero 20, 2008.] <http://www.librosweb.es/css>
10. —. **2007.** Libros Web. Introducción a XHTML. [Online] 2007. [Cited: Enero 15, 2008.] <http://www.librosweb.es/xhtml>
11. —. **2007.** Libros Web. Introducción a AJAX. [Online] 2007. [Cited: Enero 8, 2008.] <http://www.librosweb.es/ajax>
12. **Genbeta. 2006.** Blog de Software Genbeta. Aptana, IDE para aplicaciones Ajax. [Online] 7 26, 2006. [Cited: Diciembre 16, 2007.] <http://www.genbeta.com/2006/07/26-aptana-ide-para-aplicaciones-ajax>
13. **GNU Image Manipulation Program. 2001.** GIMP. [Online] 2001. [Cited: Diciembre 10, 2007.]
14. **Grupo de Electrónica para el Turtismo. 2006.** GET. Productos. [Online] 2006. [Cited: Diciembre 15, 2007.] http://www.get.tur.cu/prod_info.php?view=zun

15. **HTMLPOINT.com. 1997-2006.** ¿Qué es Apache? [Online] 1997-2006. [Cited: Diciembre 10, 2007.] <http://www.htmlpoint.com/faq/apache/01.htm>
16. **Lago, Ramiro. 2007.** Patrón "Modelo-Vista-Controlador". [Online] 2007. [Cited: Diciembre 10, 2007.] <http://www.proactiva-calidad.com/java/patrones/mvc.html>
17. **pgAdmin PostgreSQL Tools . pgAdmin.** [Online] [Cited: Diciembre 10, 2008.] <http://www.pgadmin.org/>
18. **Pitschke, Jurguen. 2002-2008.** Visual Paradigm Suite. [Online] 2002-2008. [Cited: Enero 20, 2008.] <http://www.visual-paradigm.eu/content/view/12/1/lang.en/>
19. **ProgramacionWeb.net . 2003-2008.** MVC - Modelo Vista Controlador. [Online] 2003-2008. [Cited: Enero 9, 2008.] <http://www.programacionweb.net/articulos/articulo/?num=505>
20. **Rumbaugh, James, Jacobson, Ivar and Booch, Grady. 2000.** El Lenguaje Unificado de Modelado. [Online] 2000. [Cited: Enero 10, 2008.] <http://bibliodoc.uci.cu/pdf/reg00060.pdf>
21. **San Félix, Alvaro del Castillo. 2000.** El servidor de web Apache: Introducción práctica. [Online] 2000. [Cited: Enero 8, 2008.] <http://acsblog.es/articulos/trunk/LinuxActual/Apache/html/index.html>
22. **Staff&Line España. 2000.** Business Asset Management. EasyVista Business Asset: el programa informático de gestión de los activos estratégicos. [Online] 2000. [Cited: Diciembre 8, 2007.] <http://www.staffandline.es/EasyVista-Business-Asset.html>
23. **Visual Paradigm.** Visual Paradigm for UML. [Online] [Cited: Diciembre 10, 2007.] <http://www.visual-paradigm.com/product/vpuml/>
24. **Vizcaíno, Aurora, García, Felix Óscar and Caballero, Ismael.** Una Herramienta CASE para ADOO: Visual Paradigm. [Online] [Cited: Noviembre 27, 2007.] http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/LabTr1_VP.pdf
25. **Welling, Luke and Thomson, Laura.** Desarrollo Web con PHP y MySQL. [Online] [Cited: Enero 8, 2008.] <http://bibliodoc.uci.cu/pdf/reg02819.pdf>
25. **World Wide Web Consortium. 1994-2005.** Guía Breve de Tecnologías XML . [Online] 1994-2005. [Cited: Noviembre 25, 2007.] <http://www.w3c.es/divulgacion/guiasbreves/TecnologiasXML>
27. **Worsley, John and Drake, Joshua. 2001.** Características de PostgreSQL. [Online] 2001. [Cited: Noviembre 29, 2007.] <http://www.sobl.org/traduccion/practical-postgres/node19.html>
28. **— . 2001.** ¿Qué es PostgreSQL? [Online] 2001. [Cited: Noviembre 29, 2007.] <http://www.sobl.org/traduccion/practical-postgres/node12.html>
29. **XperimentoS.** Los mejores 12 FrameWorks JavaScript. [Online] [Cited: Noviembre 25, 2007.] <http://www.xperimentos.com/2007/09/04/los-mejores-12-frameworks-javascript>

ANEXOS

.Anexo N1: Descripción de los casos de uso del sistema.

Nombre del caso de uso	Autenticar Usuario
Actores	Usuario(inicia)
Propósito	Garantizar que en todo momento se conozca quién está accediendo al sistema.
<p>Resumen</p> <p>El caso de uso se inicia cuando el usuario envía al sistema sus datos (nombre de usuario y contraseña). Se le informa si los datos fueron correctos o no y termina el caso de uso.</p>	
Referencia	R1.
Precondición	No aplica
Poscondición	Usuario autenticado
Caso de uso asociado	
Curso normal de eventos	
Acción del actor	Respuesta del sistema
	1. Muestra los controles necesarios para que el usuario introduzca su usuario y contraseña.
2. Introduce los datos requeridos y acepta la operación.	3. Valida la Identidad del usuario y le informa que ya está autenticado. 4. Invoca al CU Cargar Configuración.
	

Cursos alternativos	
Acción del actor	Respuesta del sistema
	Si los datos enviados no son válidos el sistema muestra un mensaje de error.

Tabla 2.1 Descripción del caso de uso Autenticar Usuario.

Nombre del caso de uso	Cargar Configuración
Actores	Usuario (Inicia)
Propósito	Cargar los elementos de configuración definidos para el usuario autenticado.
Resumen: Obtiene de la BD los privilegios a los que tiene acceso el usuario autenticado.	
Referencia	R1,R2
Precondición	Usuario autenticado
Poscondición	Aplicación lista
Caso de uso asociado	
Curso normal de eventos	
Acción del actor	Respuesta del sistema
	1. Comprueba en la BD el privilegio del usuario autenticado. Si el usuario autenticado es vicedecano de producción ir a la sección "Configuración del Vicedecano Producción". Si el usuario autenticado es estudiante ir a la sección "Configuración del Estudiante".
Sección "Configuración del Vicedecano Producción"	
	2. Obtiene las opciones a las que tiene acceso el "Vicedecano Producción". Las opciones a las que tiene acceso son: Asignar estudiante a proyecto. Mostrar Resultado de Test Mostrar Miembros. Buscar Miembros. Gestionar Proyecto.

	Gestionar Rol Gestionar Pregunta. Gestionar Lenguaje.
	3. Muestra una interfaz con las opciones a las que tiene acceso.
Sección "Configuración del Estudiante"	
	2. Obtiene las opciones a las que tiene acceso el "Estudiante" Las opciones a las que tiene acceso son: Realizar Test. Realizar Prueba de Arquitecto. Realizar Prueba de Diseñador. Realizar Prueba de Programador Realizar Prueba de Diseñador de la Base de Datos. Realizar Prueba de Analista.
	3. Muestra una interfaz con las opciones a las que tiene acceso.

Tabla 3 Descripción del caso de uso Cargar Configuración.

Nombre del caso de uso	Asignar estudiantes a Proyecto
Actores	Vicedecano Producción. (inicia)
Propósito	Asignar un estudiante a un proyecto determinado.
Resumen: El vicedecano selecciona la forma de asignación manual o automática, busca el proyecto donde se asignará y los estudiantes que quiere asignar respectivamente. En la base de datos se comprueba que el estudiante no esté asignado a ningún otro proyecto y entonces se asigna al proyecto correspondiente.	
Referencia	R1, R2, R10
Precondición	Vicedecano Autenticado
Poscondición	Estudiante asignado a proyecto
Caso de uso asociado	
Curso normal de eventos	
Acción del actor	Respuesta del sistema
1. Selecciona la forma de asignar(Manual o	

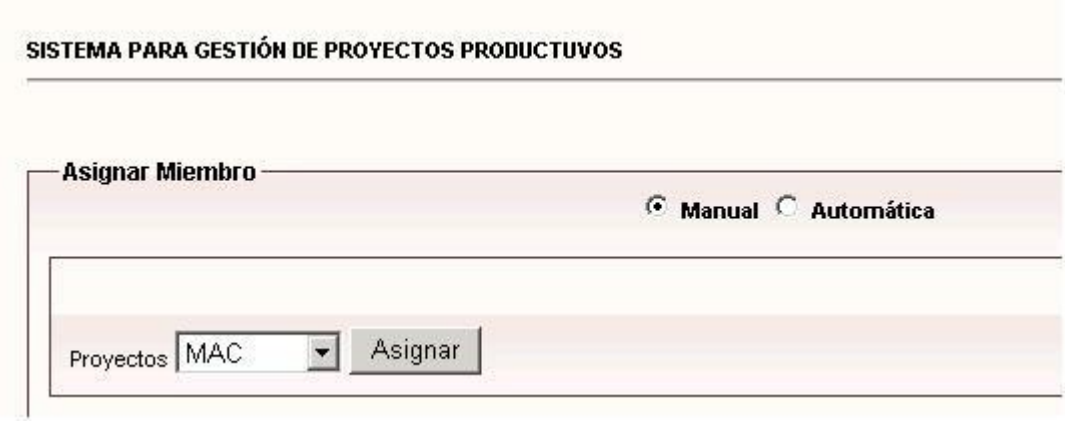
Automática)	
2-Busca proyecto que quiere asignar.	3-Busca los miembros del proyecto seleccionado y sus roles correspondiente.
4- Selecciona los estudiantes a asignar, así como el rol que desempeñará en el proyecto.	5. Comprueba que el estudiante no esté asignado a ningún otro proyecto.
6- Presiona el botón Asignar.	7-. El estudiante queda asignado al proyecto que le corresponde.
	
Cursos alternativos	
Acción del actor	Respuesta del sistema
	Si algunos de los estudiantes están asignados a otro proyecto el sistema no lo asigna al este proyecto.

Tabla 5 Descripción del caso de uso Asignar estudiante a Proyecto.

Nombre del caso de uso	Realizar Test
Actores	Estudiante(inicia)
Propósito	Realizar un test y enviarlo al sistema
Resumen El caso de uso se inicia cuando el estudiante realiza un test que posibilita recoger información acerca de los conocimientos que domine en cuanto a las diferentes herramientas a utilizar en el	

desarrollo de un proyecto determinado así como sus aptitudes para el mismo.	
Referencia	R1,R2,R3
Precondición	Estudiante autenticado
Poscondición	El test se ha realizado
Caso de uso asociado	
<p>1. ¿Estuvo vinculado a algún proyecto de producción en años anteriores? _____ No <input type="radio"/> Si <input type="radio"/></p> <p>2. ¿Qué Rol te gustaría desempeñar en un proceso productivo? _____</p> <p><input type="radio"/> Analista</p> <p><input type="radio"/> Arquitecto</p> <p><input type="radio"/> Diseñador BD</p> <p><input type="radio"/> Diseñador Gráfico</p> <p><input type="radio"/> Programador</p> <p>3. Si trabajara en un equipo profesional me gustaría ser: _____</p> <p><input type="radio"/> El que obliga a que el trabajo avance funcionalmente (se construya).</p> <p><input type="radio"/> El que critica y evalúa el trabajo de los demás.</p> <p><input type="radio"/> El que se le ocurren las mejores ideas para modelar "como" va a trabajar el sistema.</p>	
Curso normal de eventos	
Acción del actor	Respuesta del sistema
1. Realiza el test y presiona el botón Enviar.	
	2. Almacena los resultados del Test.
Cursos alternativos	
Acción del actor	Respuesta del sistema
	2.1 Si existe alguna pregunta por responder el sistema muestra un mensaje de error.

Tabla 6. Descripción del caso de uso Realizar Test.


Nombre del caso de uso	Realizar Prueba de Arquitecto
Actores	Estudiante(inicia)
Propósito	Realizar una prueba de arquitecto y enviarla al sistema.
<p>Resumen</p> <p>El caso de uso se inicia cuando el estudiante realiza una prueba del rol de arquitecto donde se evalúa los conocimientos necesarios que debe dominar para ingresar a un proyecto con este rol.</p>	
Referencia	R1,R2,R4
Precondición	Usuario Autenticado
Poscondición	La prueba ha sido realizada.
Caso de uso asociado	
Curso normal de eventos	
Acción del actor	Respuesta del sistema
1. Realiza la prueba y presiona el botón Enviar.	
	2. Almacena y muestra al usuario los resultados de la prueba.
 <p>SISTEMA PARA GESTIÓN DE PROYECTOS PRODUCTIVOS</p> <ul style="list-style-type: none"> Prueba Arquitecto Prueba Analista Prueba Diseñador BD Prueba Diseñador IU Prueba Programador 	

Tabla 8. Descripción del caso de uso Realizar Prueba de Arquitecto.

Nombre del caso de uso	Realizar Prueba de Programador
Actores	Estudiante(inicia)


Propósito	Realizar una prueba de programador y enviarla al sistema.
Resumen El caso de uso se inicia cuando el estudiante realiza una prueba del rol de programador donde se evalúa los conocimientos necesarios que debe dominar para ingresar a un proyecto con este rol.	
Referencia	R1,R2,R4
Precondición	Usuario Autenticado
Poscondición	La prueba ha sido realizada.
Caso de uso asociado	
Curso normal de eventos	
Acción del actor	Respuesta del sistema
1. Realiza la prueba y presiona el botón Enviar.	2. Almacena y muestra al usuario los resultados de la prueba.
	

Tabla 9. Descripción del caso de uso Realizar Prueba de Programador.

Nombre del caso de uso	Realizar Prueba de Diseñador de Interfaz de Usuario
Actores	Estudiante(inicia)
Propósito	Realizar una prueba de diseñador de interfaz gráfica y enviarla al sistema.
Resumen El caso de uso se inicia cuando el estudiante realiza una prueba del rol de diseñador de interfaz de	


usuario donde se evalúa los conocimientos necesarios que debe dominar para ingresar a un proyecto con este rol.	
Referencia	R1,R2,R4
Precondición	Usuario Autenticado
Poscondición	La prueba ha sido realizada.
Caso de uso asociado	
Curso normal de eventos	
Acción del actor	Respuesta del sistema
1. Realiza la prueba y presiona el botón Enviar.	2. Almacena y muestra al usuario los resultados de la prueba.
	

Tabla 10. Descripción del caso de uso Realizar Prueba de Diseñador Gráfico.


Nombre del caso de uso	Realizar Prueba de Diseñador de Base de Datos
Actores	Estudiante(inicia)
Propósito	Realizar una prueba de diseñador de la Base de Datos y enviarla al sistema.
<p>Resumen</p> <p>El caso de uso se inicia cuando el estudiante realiza una prueba del rol de diseñador de Base de Datos donde se evalúa los conocimientos necesarios que debe dominar para ingresar a un proyecto con este rol.</p>	
Referencia	R1,R2,R4
Precondición	Usuario Autenticado
Poscondición	La prueba ha sido realizada.
Caso de uso asociado	
Curso normal de eventos	
Acción del actor	Respuesta del sistema
1. Realiza la prueba y presiona el botón Enviar.	2. Almacena y muestra al usuario los resultados de la prueba.
	

Tabla 11. Descripción del caso de uso Realizar Prueba de Diseñador de Base de Datos

Nombre del caso de uso	Realizar Prueba de Analista
Actores	Estudiante(inicia)
Propósito	Realizar una prueba de analista y enviarla al sistema.


Resumen	
El caso de uso se inicia cuando el estudiante realiza una prueba analista donde se evalúa los conocimientos necesarios que debe dominar para ingresar a un proyecto con este rol.	
Referencia	R1,R2,R4
Precondición	Usuario Autenticado
Poscondición	La prueba ha sido realizada.
Caso de uso asociado	
Curso normal de eventos	
Acción del actor	Respuesta del sistema
1. Realiza la prueba y presiona el botón Enviar.	2. Almacena y muestra al usuario los resultados de la prueba.
	

Tabla 12. Descripción del caso de uso Realizar Prueba de Analista

Nombre del caso de uso	Buscar miembros para un proyecto.
Actores	Vicedecano Producción (Inicia)
Propósito	Ofrecer los servicios necesarios para la búsqueda de estudiantes para ingresar a un proyecto productivo.
<p>Resumen:</p> <p>El caso de uso se inicia cuando el vicedecano de producción necesita buscar los miembros que formarán parte de un determinado proyecto productivo. De acuerdo al rol (programador, analista, diseñador de la interfaz gráfica, diseñador de la base de dato y arquitecto) que necesite el proyecto hace la petición de búsqueda siguiendo los criterios establecidos o sea introduciendo el rol y la cantidad que necesita.</p>	
Referencia	R1,R2,R5
Precondición	Vicedecano autenticado.
Poscondición	Los resultados han sido devueltos.
Caso de uso asociado	
Curso normal de eventos	
Acción del actor	Respuesta del sistema
<p>1. Realiza una petición de búsqueda entrando previamente los datos necesarios.</p> <ul style="list-style-type: none"> • Rol • Cantidad 	
	<p>2. El sistema recibe los datos y envía estos al Servicio Web.</p> <p>3. El Servicio Web valida los datos y realiza la búsqueda.</p> <p>4. El sistema recoge el resultado y visualiza la información.</p> <p>5. El Servicio Web muestra el resultado de la búsqueda.</p>

SISTEMA PARA GESTIÓN DE PROYECTOS PRODUCTIVOS

Buscar Miembro

Rol Cantidad

Nombre	Grupo	Rol

Cursos alternativos

Acción del actor	Respuesta del sistema
	5.1 Si no existe datos en el sistema muestra un mensaje de error.

Tabla 13. Descripción del caso de uso Buscar Miembros para un Proyecto.

Nombre del caso de uso	Mostrar Miembros
Actores	Vicedecano Producción (Inicia).
Propósito	Mostrar el listado de los estudiantes que cumplan con los criterios de búsqueda introducidos por el vicedecano de producción.
Resumen	
El vicedecano de producción selecciona la opción “Mostrar Miembros” y se muestra una interfaz con los criterios de búsqueda (Por Rol o por Proyecto). Se obtiene de la BD el listado de los resultados que cumplan con los criterios de búsqueda especificados mostrándose el mismo.	
Referencia	R1,R2,R11
Precondición	Vicedecano Producción Autenticado
Poscondición	El listado de los estudiantes ha sido mostrado.
Caso de uso asociado	
Curso normal de eventos	
Acción del actor	Respuesta del sistema

1. Introduce los criterios de búsqueda (Rol o proyecto).	
	<p>2. Muestra una interfaz y muestra la pestaña “Mostrar Miembros”</p> <p>3. Obtiene de la BD los nombres de los estudiantes que ya existen registrados y cumplen con los criterios de búsqueda introducidos.</p> <p>4. Muestra una interfaz solicitando los siguientes criterios de búsqueda:</p> <ul style="list-style-type: none"> • Por Rol • Por proyecto
5. Selecciona la opción “Mostrar Miembros”	
	6. Obtiene de la BD el listado de los estudiantes usando los criterios de búsqueda introducidos.
	<p>7. Muestra el listado de los estudiantes obtenidos como resultado de la búsqueda con hipervínculos a ver detalles.</p> <p>Los datos a mostrar por cada estudiante son:</p> <ul style="list-style-type: none"> • Nombre y Apellidos • Grupo • Proyecto • Rol

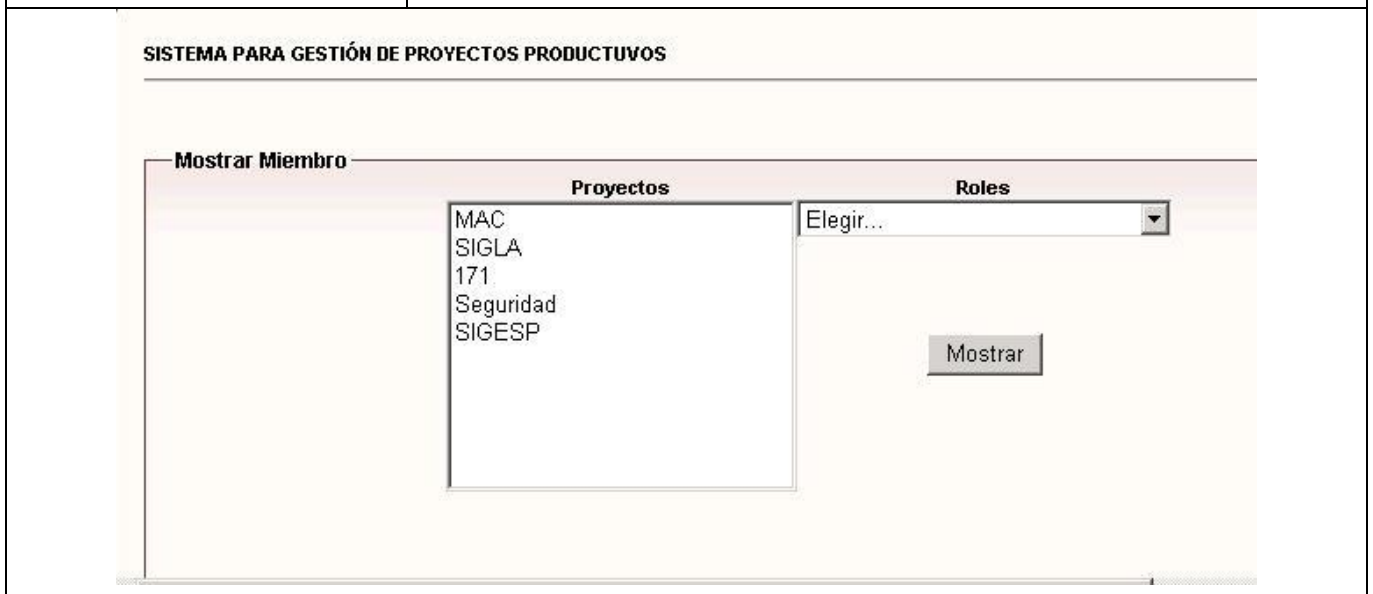


Tabla 14. Descripción del caso de uso Mostrar Miembros.

Nombre del caso de uso	Mostrar Resultado
Actores	Vicedecano Producción (Inicia).
Propósito	Mostrar los resultados de las pruebas hechas por cada rol.
Resumen El vicedecano de producción selecciona la opción “Mostrar Resultado” y escoge el grupo al que pertenece el estudiante(s) de quien quiere ver el resultado y una vez seleccionado(s) presiona el botón mostrar. Se obtiene de la BD el listado de los resultados que cumplan con los criterios de búsqueda especificados y se muestra el mismo.	
Referencia	R1,R2,R12
Precondición	Vicedecano Producción Autenticado
Poscondición	Listado de resultados mostrado.
Caso de uso asociado	
Curso normal de eventos	
Acción del actor	Respuesta del sistema
1. Selecciona la opción “Mostrar Resultado”.	
2. Selecciona el estudiante(s) del cual quiere saber los resultados.	
	3. Comprueba que se hayan introducido los datos para la búsqueda.
4. Selecciona la opción “Mostrar”.	
	5. Obtiene de la BD los resultados usando los criterios introducidos. Los datos a obtener por cada resultado son: <ul style="list-style-type: none"> • Nombre y Apellidos • Pruebas que realizó de cada rol en caso de haberlas hecho

Nombre	Arquitecto	Analista	Diseñador BD
Madelis Pérez Gil		2008-05-31 2008-05-31 2008-06-01 2008-06-04	2008-06-01
Cursos alternativos			
Acción del actor	Respuesta del sistema		
	3.1 De no escoger ningún criterio de búsqueda se muestra un mensaje indicando que debe seleccionar al menos un estudiante.		

Tabla 15. Descripción del caso de uso Mostrar Resultado de Prueba

Nombre del caso de uso	Gestionar Proyecto
Actores	Vicedecano Producción (Inicia)
Propósito	Permitir realizar las operaciones de listar, eliminar y crear proyecto.
Resumen: El caso de uso se inicia cuando el vicedecano de producción desea realizar las operaciones de listar, eliminar y crear proyecto.	
Referencia	R1,R2,R6
Precondición	Vicedecano Producción autenticado.
Poscondición	No aplica
Caso de uso asociado	
Curso normal de eventos	
Acción del actor	Respuesta del sistema
1. Selecciona una de las siguientes opciones: <ul style="list-style-type: none"> • Crear Proyecto. • Listar Proyecto. • Eliminar Proyecto. 	2. El sistema según la opción seleccionada por el Vicedecano Producción realiza la operación siguiente: <ul style="list-style-type: none"> • Si el VP selecciona Crear Proyecto ir a sección Crear Proyecto. • Si el VP selecciona Listar Proyecto ir a sección Listar Proyecto.

	<ul style="list-style-type: none"> • Si el VP selecciona Eliminar Proyecto ir a sección Eliminar Proyecto.
Sección 1 "Crear Proyecto"	
1. Selecciona la opción de Crear Proyecto.	2. Muestra la interfaz que permite crear un proyecto.
3. Introduce los datos necesarios para crear el proyecto y presiona el botón crear: <ul style="list-style-type: none"> • Nombre • Alcance • Lenguaje • Líder • Descripción 	4. Verifica que no exista ningún campo vacío. Ni que el nombre del proyecto se encuentre en la BD.
	5. Guarda en la base de datos los datos del proyecto y muestra el proyecto creado junto con los demás existentes en la BD.

Datos del Proyecto

nombre:

alcance:

lenguaje:

Posible Lider

listado:

ADONIS RODRIGUEZ FERNANDEZ

ADRIAN MARANJE AGRAMONTE

AILIN ALARCON FERRA

AIMEL CASTELLÓN BELTRÁN

ALAIN PEREZ BALART

ALBERTO ARCE MARTINEZ

ALBERTO FERNANDEZ OLIVA

ALBERTO HERNANDEZ MEDEROS

ALEJANDRO RODRIGUEZ REYES

ALIENNIS MERCEDES GONZALEZ HURTAC



Sección "Listar Proyecto"	
1. Solicita la opción Listar Proyectos	2. Muestra una interfaz con todos lo proyectos existentes en la BD.

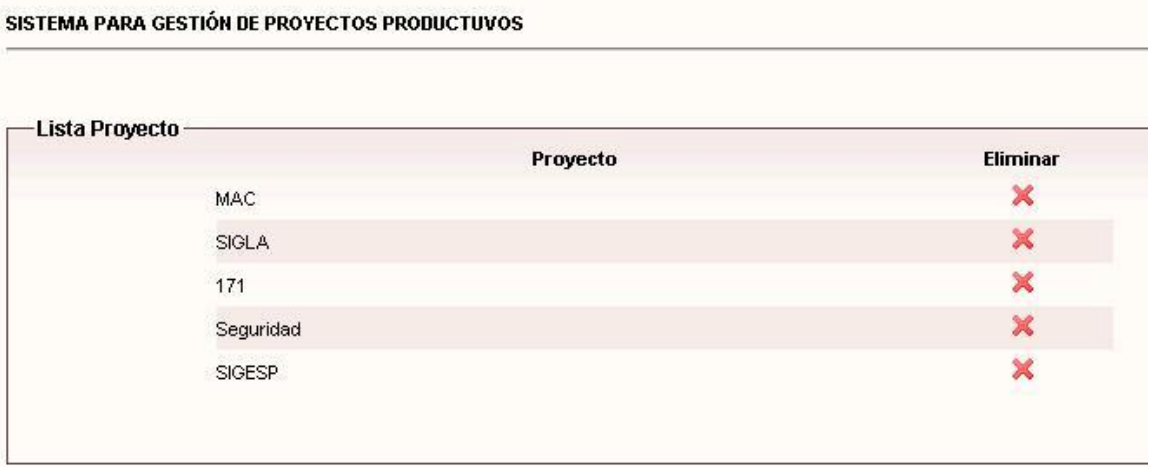
Sección "Eliminar Proyecto"	
1. Solicita la opción Eliminar Proyecto	2. El sistema pregunta si realmente desea eliminar el proyecto y muestra dos opciones C (OK) y D (Cancel).
3. Selecciona una de las dos opciones mostradas.	4. Si selecciona: <ul style="list-style-type: none"> • C (OK) el sistema elimina el proyecto de la base de datos. • D (Cancel) el sistema no elimina el proyecto en la base de datos y permanece en la misma página.
	
Cursos alternativos Sección "Crear Proyecto"	
Acción del actor	Respuesta del sistema
	4.1 El sistema emite un mensaje de error si existe algún campo vacío.

Tabla 16 Descripción del caso de uso Gestionar Proyecto.

Nombre del caso de uso	Gestionar Rol
Actores	Vicedecano Producción (Inicia)
Propósito	Permitir realizar las operaciones de adicionar un nuevo rol y eliminarlo.
Resumen: El caso de uso se inicia cuando el vicedecano de producción desea realizar las	

operaciones de adicionar un nuevo rol y eliminarlo.	
Referencia	R1,R2,R8
Precondición	El Vicedecano Producción autenticado.
Poscondición	No aplica
Caso de uso asociado	
Curso normal de eventos	
Acción del actor	Respuesta del sistema
<p>1 El Vicedecano Producción selecciona una de las siguientes opciones:</p> <ul style="list-style-type: none"> • Adicionar Rol. • Eliminar Rol. 	<p>2.El sistema según la opción seleccionada por el Vicedecano Producción realiza la operación siguiente:</p> <ul style="list-style-type: none"> • Si el VP selecciona Adicionar Rol ir a sección Adicionar Rol. • Si el VP selecciona Eliminar Rol ir a sección Eliminar Rol.
Sección 1 “Adicionar Rol”	
1. Selecciona la opción de Adicionar Rol	2. Muestra la interfaz que permite Adicionar Rol.
3. Introduce los datos necesarios para Adicionar Rol y presiona el botón Agregar: <ul style="list-style-type: none"> • Rol 	4. Verifica no existan campos vacíos.
	5. Guarda en la base de datos los datos del rol y muestra el rol creado junto con los demás existentes en la BD.
Sección “Eliminar Rol”	
1. Solicita la opción Eliminar Rol	2. Pregunta si realmente desea eliminar el rol y muestra dos opciones C (OK) y D (Cancel).
3. Selecciona una de las dos opciones mostradas.	4. Si selecciona: <ul style="list-style-type: none"> • C (OK) el sistema elimina el rol de la base de datos. • D (Cancel) el sistema no elimina el rol en la base de datos y permanece en la misma página.

<p>SISTEMA PARA GESTIÓN DE PROYECTOS PRODUCTIVOS</p> <hr/> <p>Adicionar Nuevo Rol</p> <p>rol <input type="text"/> <input type="button" value="Agregar"/></p> <table border="1"> <thead> <tr> <th>Rol</th> <th>Eliminar</th> </tr> </thead> <tbody> <tr> <td>Analista</td> <td style="text-align: center;">✗</td> </tr> <tr> <td>Arquitecto</td> <td style="text-align: center;">✗</td> </tr> <tr> <td>Diseñador de Base de Datos</td> <td style="text-align: center;">✗</td> </tr> <tr> <td>Programador</td> <td style="text-align: center;">✗</td> </tr> <tr> <td>Diseñador de Interfaz de Usuario</td> <td style="text-align: center;">✗</td> </tr> </tbody> </table>		Rol	Eliminar	Analista	✗	Arquitecto	✗	Diseñador de Base de Datos	✗	Programador	✗	Diseñador de Interfaz de Usuario	✗
Rol	Eliminar												
Analista	✗												
Arquitecto	✗												
Diseñador de Base de Datos	✗												
Programador	✗												
Diseñador de Interfaz de Usuario	✗												
Cursos alternativos Sección "Adicionar Rol"													
Acción del actor	Respuesta del sistema												
	3.1 Emite un mensaje de error si existe algún campo vacío.												

Tabla17 Descripción del caso de uso Gestionar Rol.

Nombre del caso de uso	Gestionar Lenguaje
Actores	Vicedecano Producción (Inicia)
Propósito	Permitir realizar las operaciones de adicionar un nuevo lenguaje y eliminarlo.
Resumen: El caso de uso se inicia cuando el vicedecano de producción desea realizar las operaciones de adicionar un nuevo lenguaje y eliminarlo.	
Referencia	R1,R2,R9
Precondición	Vicedecano Producción autenticado.
Poscondición	No aplica
Caso de uso asociado	
Curso normal de eventos	
Acción del actor	Respuesta del sistema
1 Selecciona una de las siguientes opciones:	2. Según la opción seleccionada por el Vicedecano Producción realiza la operación

<ul style="list-style-type: none"> • Adicionar Lenguaje. • Eliminar Lenguaje. 	<p>siguiente:</p> <ul style="list-style-type: none"> • Si el VP selecciona Adicionar Lenguaje ir a sección Adicionar Lenguaje. • Si el VP selecciona Eliminar Lenguaje ir a sección Eliminar Lenguaje.
<p>Sección 1 “Adicionar Lenguaje”</p>	
<p>1. Selecciona la opción de Adicionar Lenguaje.</p>	<p>2. Muestra la interfaz que permite Adicionar Lenguaje.</p>
<p>3- Introduce los datos necesarios para Adicionar Lenguaje y presiona el botón Agregar:</p> <ul style="list-style-type: none"> • Lenguaje 	<p>4- Verifica no existan campos vacíos.</p>
	<p>5. Guarda en la base de datos los datos del lenguaje y muestra el lenguaje creado junto con los demás existentes en la BD.</p>
<p>Sección “Eliminar Lenguaje ”</p>	
<p>1- Solicita la opción Eliminar Lenguaje</p>	<p>2. Pregunta si realmente desea eliminar el lenguaje y muestra dos opciones C (OK) y D (Cancel).</p>
<p>3. Selecciona una de las dos opciones mostradas.</p>	<p>4. Si selecciona:</p> <ul style="list-style-type: none"> • C (OK) el sistema elimina el lenguaje de la base de datos. • D (Cancel) el sistema no elimina el lenguaje en la base de datos y permanece en la misma página.

SISTEMA PARA GESTIÓN DE PROYECTOS PRODUCTIVOS															
<p>Adicionar Nuevo Lenguaje</p> <p>lenguaje <input type="text"/> <input type="button" value="Agregar"/></p> <table border="1"> <thead> <tr> <th>Lenguaje</th> <th>Eliminar</th> </tr> </thead> <tbody> <tr> <td>PHP</td> <td>X</td> </tr> <tr> <td>C#</td> <td>X</td> </tr> <tr> <td>C++</td> <td>X</td> </tr> <tr> <td>Java</td> <td>X</td> </tr> <tr> <td>ASP</td> <td>X</td> </tr> <tr> <td>Python</td> <td>X</td> </tr> </tbody> </table>		Lenguaje	Eliminar	PHP	X	C#	X	C++	X	Java	X	ASP	X	Python	X
Lenguaje	Eliminar														
PHP	X														
C#	X														
C++	X														
Java	X														
ASP	X														
Python	X														
Cursos alternativos Sección "Adicionar Lenguaje "															
Acción del actor	Respuesta del sistema														
	4.1 El sistema emite un mensaje de error si existe algún campo vacío.														

Tabla 18 Descripción del caso de uso Gestionar Lenguaje.

Nombre del caso de uso	Gestionar Pregunta
Actores	Vicedecano Producción (Inicia)
Propósito	Permitir realizar las operaciones de adicionar, listar y eliminar pregunta
Resumen: El caso de uso se inicia cuando el vicedecano de producción desea realizar las operaciones de adicionar, listar y eliminar pregunta a un tipo de prueba determinado.	
Referencia	R1,R2,R7
Precondición	Vicedecano Producción autenticado.
Poscondición	No aplica
Caso de uso asociado	
Curso normal de eventos	
Acción del actor	Respuesta del sistema

<p>1- Selecciona una de las siguientes opciones:</p> <ul style="list-style-type: none"> • Adicionar Pregunta. • Listar Pregunta. • Eliminar Pregunta. 	<p>2- Según la opción seleccionada por el Vicedecano Producción realiza la operación siguiente:</p> <ul style="list-style-type: none"> • Si el VP selecciona Adicionar Pregunta ir a sección Adicionar Pregunta. • Si el VP selecciona Listar Pregunta ir a sección Listar Pregunta. • Si el VP selecciona Eliminar Pregunta ir a sección Eliminar Pregunta.
<p>Sección 1 “Adicionar Pregunta”</p>	
<p>1. Selecciona la opción de Adicionar Pregunta</p>	<p>2. Muestra la interfaz que permite Adicionar Pregunta.</p>
<p>3- Introduce los datos necesarios para Adicionar Pregunta y presiona el botón Continuar:</p> <ul style="list-style-type: none"> • Texto o Enunciado • Tipo Pregunta • Tipo Prueba • Número de Incisos • Nivel 	<p>4- Verifica que no exista ningún campo vacío.</p>
	<p>5- Guarda en la base de datos los datos de la pregunta y muestra otra interfaz para introducir la respuesta a la pregunta.</p>
<p>6- Presiona el botón Adicionar Preguntas y Respuestas.</p>	<p>7- Muestra la pregunta confeccionada.</p>









	en la misma página.
<p>Diga Verdadero (V) o Falso (F) según corresponda:  </p>	
<p>Una buena gestión de tiempo implica:  </p>	
<p>Los modelos de calidad como CMM e ISO se implementan en el nivel de:  </p>	
<p>Diga Verdadero (V) o Falso (F):  </p>	
Cursos alternativos Sección "Adicionar Pregunta"	
Acción del actor	Respuesta del sistema
	El sistema emite un mensaje de error si existe algún campo vacío.

Tabla 19 Descripción del caso de uso Gestionar Lenguaje.

Anexo N2: Diagrama de Interacción.

Diagrama de Interacción (Secuencia): Framework Symfony

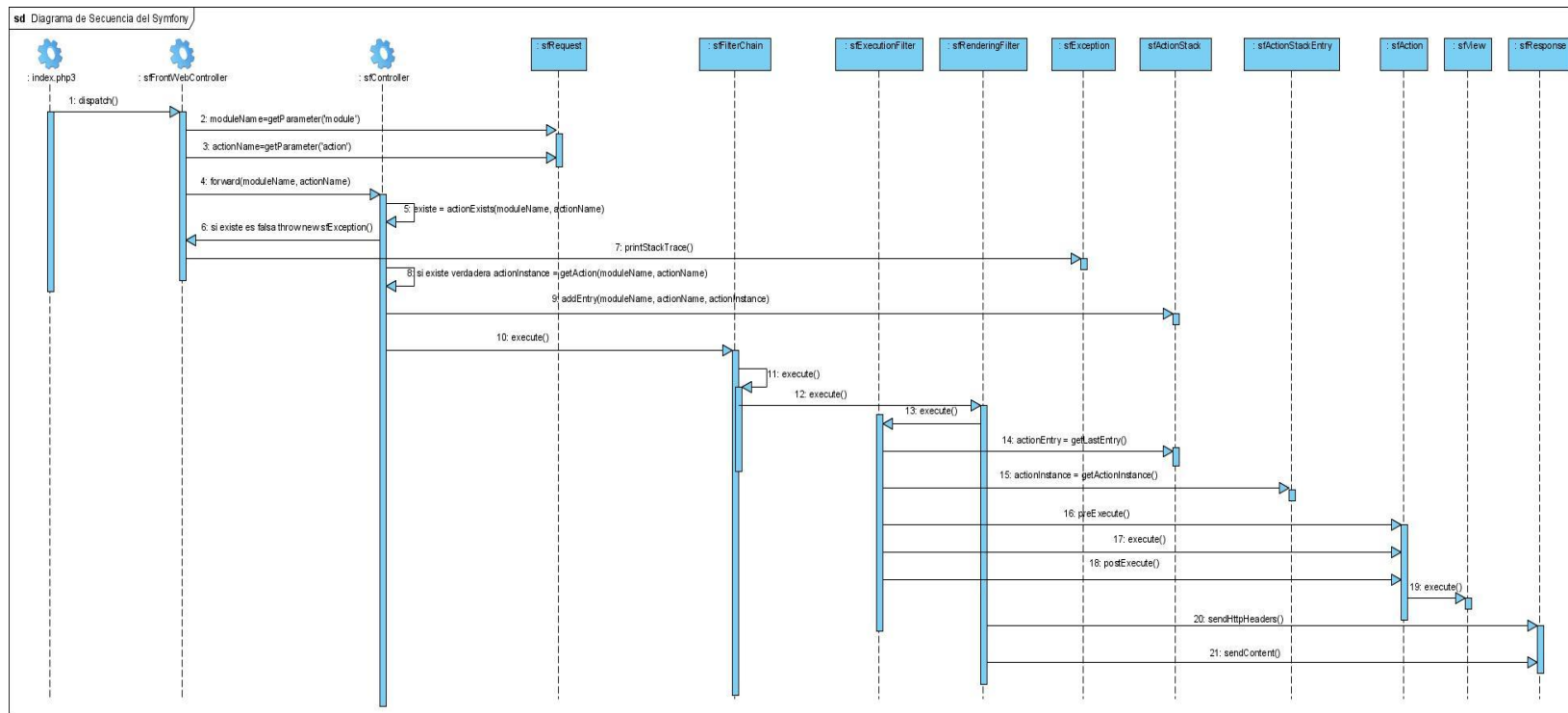


Diagrama de Interacción (Secuencia): Asignar Estudiante.

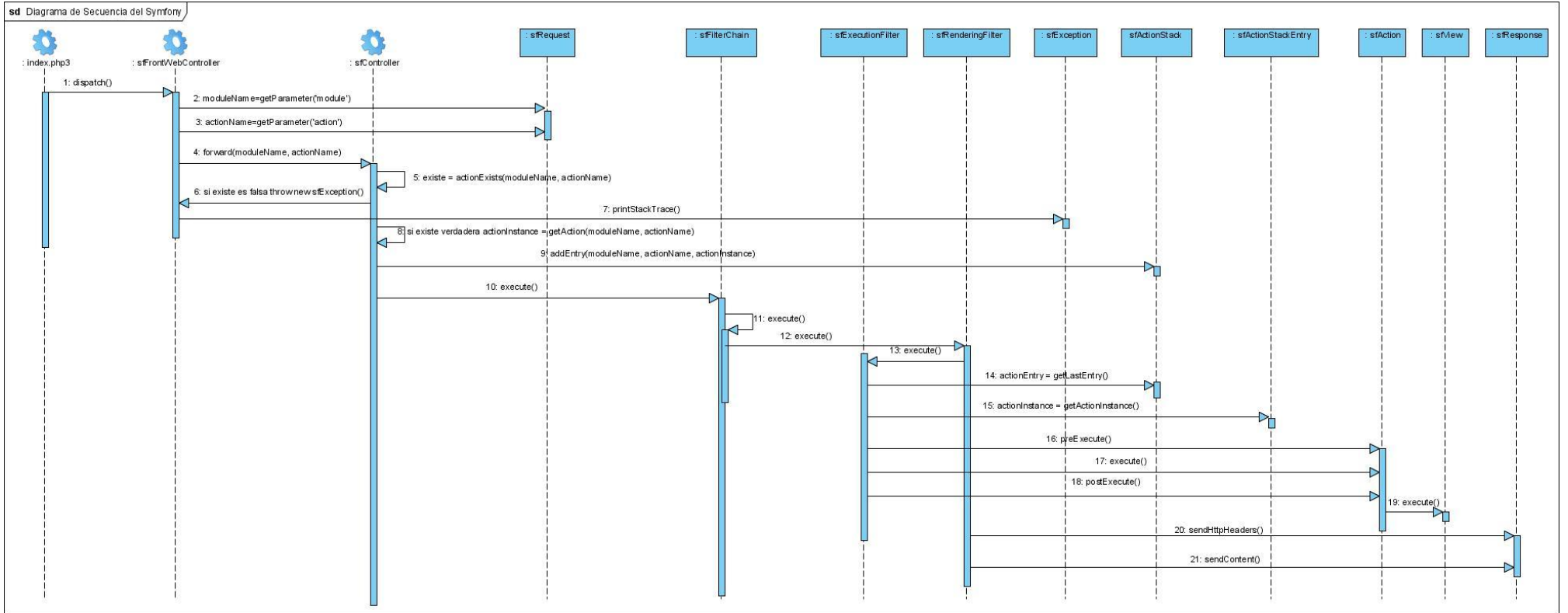
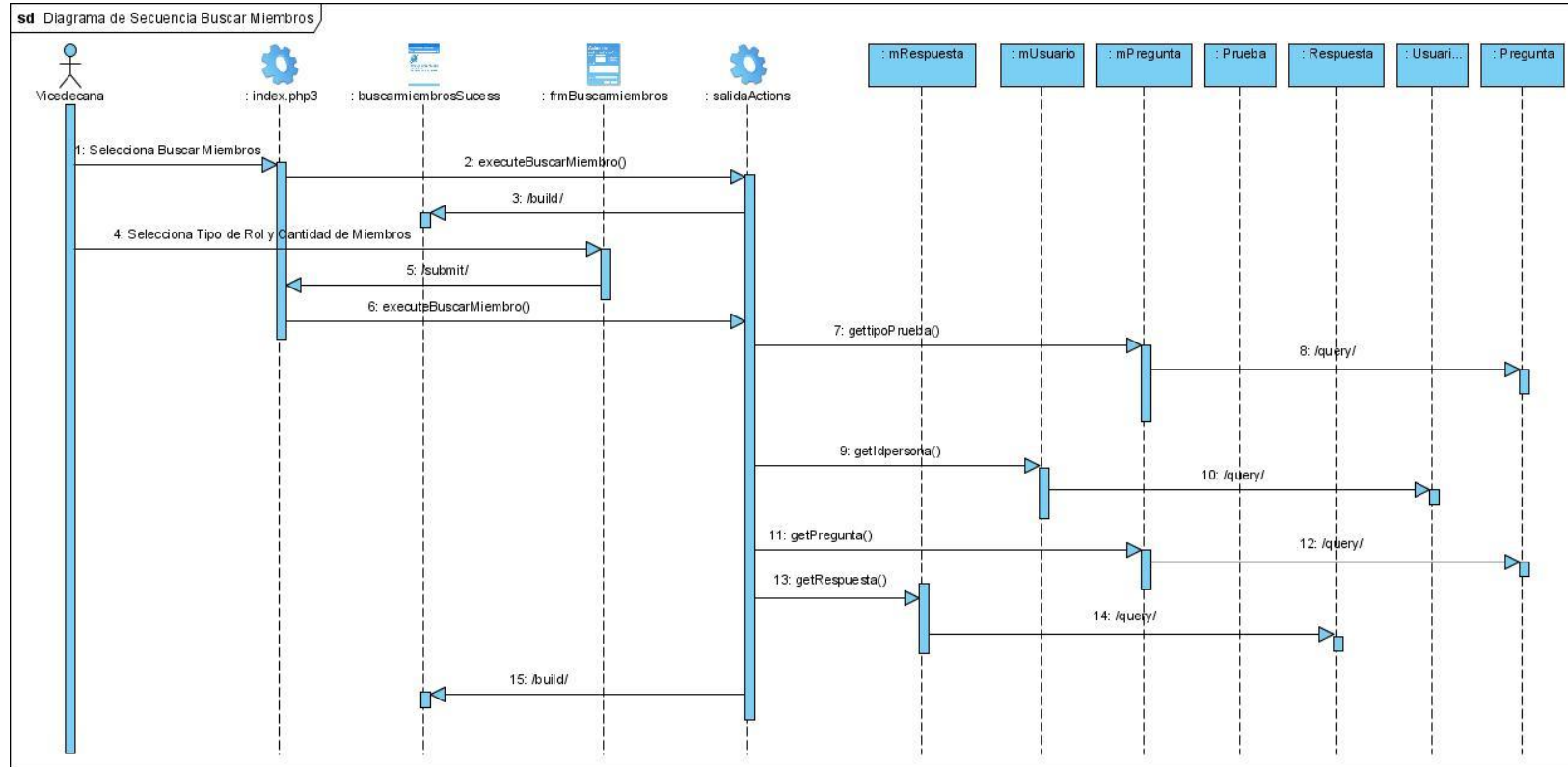


Diagrama de Interacción (Secuencia): Asignar Estudiante.



GLOSARIO DE TÉRMINOS

- **Administrador:** es la persona que tiene privilegios para determinadas funcionalidades del sistema.
- **APACHE:** es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/1.1.
- **AJAX:** Asynchronous JavaScript And XML.
- **ASP:** Active Server Pages. Es una tecnología del lado servidor de Microsoft para páginas web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Server (IIS). Con ASP se pueden combinar páginas HTML, scripts y objetos COM. Con el objetivo de crear aplicaciones potentes. Se caracterizan por su fácil desarrollo y mantenimiento.
- **Arquitectura Cliente/Servidor:** es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en elementos independientes que cooperan entre sí para intercambiar información, servicios o recursos.
- **Benchmark:** es una técnica utilizada para medir el rendimiento de un sistema o componente de un sistema.
- **Cliente:** Persona, organización o grupo de personas que encarga la construcción de un sistema, ya sea empezando desde cero o mediante el refinamiento de versiones sucesivas.
- **CUS:** Caso de uso del sistema.
- **DHTML:** Dynamic HTML.
- **DOM:** Document Object Model (Modelo de Objetos de documentos).
- **Email:** Correo Electrónico
- **FrameWork:** Estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un FrameWork 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.
- **GIMP:** GNU Image Manipulation Program (programa para manipular imágenes)
- **Herramientas CASE:** (Computer Aided Software Engineering): Herramientas CASE: Herramientas utilizadas para el desarrollo de proyectos de Ingeniería de Software.
- **Hardware:** Componentes electrónicos, tarjetas, periféricos y equipo que conforman un sistema de computación; se distinguen de los programas (software) porque son tangibles.
- **HTML:** HyperText Markup Language. Lenguaje usado para escribir documentos para servidores World Wide Web. Es una aplicación de la ISO Standard 8879:1986. Es un lenguaje de marcas. Los lenguajes de marcas no son equivalentes a los lenguajes de programación aunque se definan

igualmente como "lenguajes". Son sistemas complejos de descripción de información, normalmente documentos, que se pueden controlar desde cualquier editor ASCII.

- **HTTP:** HyperText Transfer Protocol. Protocolo de Transferencia de Hipertextos. Modo de comunicación para solicitar páginas Web.
- **IDEFO:** Integration Definition for Function Modeling (Definición de la integración para la modelización de las funciones)
- **Interfaz de usuario:** Interfaz a través de la cual un usuario interactúa con un sistema.
- **Internet:** Sistema de redes de computación ligadas entre sí, con alcance mundial, que facilita servicios de comunicación de datos como registro remoto, transferencia de archivos, correo electrónico y grupos de noticias. Internet es una forma de conectar las redes de computación existentes que amplía en gran medida el alcance de cada sistema participante.
- **JSP:** Java Server Pages. Es la tecnología para generar páginas web de forma dinámica en el servidor, desarrollado por Sun Microsystems, basado en scripts que utilizan una variante del lenguaje java. La tecnología JSP, o de JavaServer Pages, es una tecnología Java que permite a los programadores generar dinámicamente HTML, XML o algún otro tipo de página web. Esta tecnología permite al código Java y a algunas acciones predefinidas ser embebidas en el contenido estático.
- **Linux:** Es el nombre de un núcleo, pero se suele denominar con este nombre a un sistema operativo de libre distribución software libre (y de código abierto), donde el código fuente está disponible públicamente y cualquier persona, con los conocimientos informáticos adecuados, puede libremente estudiarlo, usarlo, modificarlo y redistribuirlo.
- **Macromedia Dreamweaver MX:** Herramienta para el desarrollo de aplicaciones Web de Macromedia. Combina en un único entorno de desarrollo accesible y potente las reconocidas herramientas de presentación visual de Dreamweaver, las características de rápido desarrollo de aplicaciones Web de Dreamweaver UltraDev y ColdFusion Studio, y el extenso soporte de edición de código de HomeSite. Ofrece una completa solución abierta para las tecnologías Web y estándares de hoy, incluyendo la accesibilidad y servicios Web.
- **Microsoft:** Compañía que manufactura los sistemas de operación DOS y Windows.
- **MySQL:** Es un sistema de gestión de bases de datos relacional que cuentan con todas las características de un motor de BD comercial: transacciones atómicas, triggers, replicación, llaves foráneas entre otras. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar.
- **MVC:** Modelo Vista Controlador.
- **ORDBMS:** Objeto-Relacionales

- **PHP:** PHP: Hypertext Preprocessor. Es un ambiente script del lado del servidor que permite crear y ejecutar aplicaciones Web dinámicas e interactivas. Con PHP se pueden combinar páginas HTML y scripts. Con el objetivo de crear aplicaciones potentes.
- **Plugin:** Un **plugin** o *componente enchufable* (o **plug-in** -en inglés "enchufar"-, también conocido como **addin**, **add-in**, **addon** o **add-on**) es una aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica, generalmente muy específica.
- **PostgreSQL:** es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) libre.
- **RIA:** (Rich Internet Applications): Son aplicaciones web que tienen las características y la funcionalidad de aplicaciones de escritorio tradicionales.
- **RUP:** Rational Unified Process (Proceso Unificado de desarrollo). Metodología para el desarrollo de Software.
- **Software:** Programas de sistema, utilerías o aplicaciones expresados en un lenguaje de máquina.
- **Sitio Web:** Es un conjunto de páginas web, típicamente comunes a un dominio de Internet o subdominio en la World Wide Web en Internet.
- **SGBD:** Sistema de Gestión de Bases de Datos. Es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez.
- **UCI:** Universidad de las Ciencias Informáticas.
- **UML:** Unified Modeling Language. Es una notación estándar para modelar objetos del mundo real como primer paso en el desarrollo de programas orientados a objetos. Es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software.
- **VP:** Vicedecano Producción.
- **WEB** (WWW): Red de documentos HTML intercomunicados y distribuidos entre servidores del mundo entero.
- **XML:** Extensible Markup Language. Es un lenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium. Orientado principalmente al almacenamiento, procesamiento y transmisión de mensajes.