



**Universidad de las Ciencias Informáticas.  
Facultad 3. Turismo y Negocio.**



**“Sistema automatizado para los procesos en el Laboratorio de Calidad.  
Módulo: Seguimiento de los recursos.”**

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE  
INGENIERO EN CIENCIAS INFORMÁTICAS.**

**Autoras**

Yadicza López Concepción.  
Keidys Martínez Barroso.

**Tutores**

Dra Ailyn Febles Estrada.  
Ing Ramsés Delgado Martínez.

Ciudad de La Habana, Cuba  
Junio de 2007.  
“Año 49 de la Revolución”

*“He ofendido a Dios y a la humanidad porque mi trabajo no tuvo la  
calidad que debía haber tenido”*

*Leonardo Da Vinci*

## DECLARACIÓN DE AUTORÍA

Declaramos que somos las únicas autoras de este trabajo y autorizamos al Laboratorio Central de Calidad de Software de la Universidad de las Ciencias Informáticas; así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

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

Yadicza López Concepción

\_\_\_\_\_

Firma del Autor

Keidys Martínez Barroso.

\_\_\_\_\_

Firma del Autor

Dra Ailyn Febles Estrada

\_\_\_\_\_

Firma del Tutor

Ing Ramsés Delgado Martínez

\_\_\_\_\_

Firma del Tutor

## OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título: **Sistema automatizado para los procesos en el Laboratorio de Calidad. Módulo: Seguimiento de los recursos.**

Autoras: **Yadicza López Concepción.**

**Keidys Martínez Barroso.**

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

**<Aquí el tutor debe expresar cualitativamente su opinión y medir (usando la escala: muy alta, alta, adecuada) entre otras las cualidades siguientes:**

- Independencia
- Originalidad
- Creatividad
- Laboriosidad
- Responsabilidad >

**< Además, debe evaluar la calidad científico-técnica del trabajo realizado (resultados y documento) y expresar su opinión sobre el valor de los resultados obtenidos (aplicación y beneficios) >**

Por todo lo anteriormente expresado considero que las estudiantes están aptas para ejercer como Ingenieras Informáticas; y propongo que se le otorgue al Trabajo de Diploma la calificación de **<nota>**. **<Además, si considera que los resultados poseen valor para ser publicados, debe expresarlo también>**

Dra Ailyn Febles Estrada

Ing Ramsés Delgado Martínez

\_\_\_\_\_  
Firma del Tutor

\_\_\_\_\_  
Firma del Tutor

\_\_\_\_\_  
Fecha

## OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado **Sistema Automatizado para los procesos en el Laboratorio de Calidad. Módulo: Seguimiento de los recursos**, fue realizado en la UCI. 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 <valor en MN o USD del efecto económico> Y para que así conste, se firma la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_

---

Representante de la entidad Cargo

\_\_\_\_\_  
Firma

\_\_\_\_\_  
Cuño

*Hay tesoros en la vida  
Imposibles de comprar,  
Imposibles de olvidar  
O simplemente apartar.*

*Es el amor de una madre  
El tesoro más inmenso,  
El que te ayuda a vivir,  
El que te brinda su aliento.*

*El que te hace sonreír  
En las noches de tristeza,  
El que te abre los brazos  
Con tesón y con firmeza.*

*Ese amor puro y sincero  
El de una madre de verdad  
Hoy no puedo tenerlo,  
Estas muerta en realidad.*

*No puedo sentir tus besos  
En las noches de dolor,  
No puedo oír tus consejos  
Porque no tengo este amor.*

*A ti, madre querida  
Hoy te quiero dedicar  
Con todo el amor del mundo  
Lo que me hace inspirar.*

*Desde el cielo donde reinas  
Me diste felicidad  
Protegiste mi camino  
Fuiste luz en la oscuridad.*

*Hoy, me guían tus recuerdos  
Tus palabras, al andar  
Tu mirada penetrante  
Y tu manera de actuar.*

*No sabes cuanto, mamita  
Te deseo aquí a mi lado  
Para poder entregarte  
Este sueño realizado.*

*Yadicza.*

*Tal vez sobren las palabras, se que lees mis pensamientos.  
Tal vez sabré recordarte cada día que pasa, aunque no te sienta, estás junto a mí.  
Tal vez no siempre caigan mis lágrimas al tratar de recordarte.  
Tal vez sobren mis suspiros que demuestran el dolor de mi alma.  
Pero lo que nunca podrá sobrarne madre mía, es amor para ti,  
y aunque poco te tuvo la vida junto a mi, no bastó para olvidarte...  
A ti te dedico este triunfo...*

*Keidys*

 *Agradecimientos compartidos.*

Llegue este profundo agradecimiento a nuestras familias por habernos guiado por el camino correcto y haber hecho posible que la escalada a esta cima no hubiese sido en vano.

Agradecemos a todos los profesores que de una forma u otra aportaron su grano de arena a la preparación que hemos adquirido en los años de estudiante.

A nuestros tutores, por apoyarnos y guiarnos para llegar al final.

Agradecemos de la misma manera a nuestros hermanos y amigos, y muy en particular a aquellos que transitaron estos últimos años junto a nosotras en esta honrosa misión, a la cual nos encomendamos para alcanzar nuestros anhelos.

A todos los que han sido nuestros compañeros de estudio, por la ayuda ofrecida y por haber tenido la suerte de compartir con ellos.

Las gracias también a nuestro comandante en jefe Fidel Castro Ruz, por darnos esta Revolución Socialista Cubana, y a esta última, por habernos permitido formarnos como profesionales para el bien de la sociedad.

En fin, para no olvidar a nadie, a todos aquellos que nos han apoyado durante este largo andar, a todos...,

Muchas Gracias.

...y gracias a la vida por habernos dado la oportunidad



 Yadicza.

A lo largo del camino la vida me ha puesto pruebas difíciles, situaciones en las que si no hubiera contado con el apoyo y cariño de personas especiales, no habría podido llegar al final. Hoy, a solo unos pasos de la meta, quiero expresar de todo corazón mi más sincero agradecimiento a esas personas que marcaron mi vida para siempre.

♥ *A mi nueva familia*, no bastará una vida para agradecerles todo lo que han hecho por mí; junto a ustedes conocí el amor de un hogar, la alegría de vivir y el impulso para seguir adelante. Gracias por su apoyo incondicional, por hacer suyos mis grandes momentos de tristezas y decisiones difíciles. Los admiro, y los amo por siempre.

♥ *A ti Martica*, lágrimas brotan al escribir estas líneas, jamás podré expresar con palabras todo lo que representas para mí. Has sido mi guía, mi amiga, mi confidente, has sabido abrirme tu corazón y acogerme como una hija. Gracias a ti, pude tener el amor y el apoyo de una madre en todos estos años tan difíciles.

♥ *A mis viejucos*, gracias por tantas horas de desvelo y oraciones, por sus mimos, sus caricias, su comprensión, por estar siempre presentes en los momentos más duros. Hoy, la nené de la casa los admira con respeto y les entrega este gran sueño hecho realidad, reciban este regalo porque ha sido por ustedes.

♥ *A mis hermanos del alma*, gracias por estar presentes en todos estos años, por dejarme entrar en su familia y acogerme como una hermana de corazón.

♥ *A mis papis Juan José y Elder*, ustedes con su ejemplo han sabido llevarme por este camino tan largo y complicado. Gracias a sus consejos y apoyo, cada cual en su momento ayudó a formarme como persona y a darme fuerzas para seguir adelante cuando todo parecía imposible.

♥ *A Magui*, estaré en deuda eternamente por todo lo que has hecho, gracias por ser como una madre y estar presente en los momentos decisivos.

♥ *A mis fieles amigas, Mabel y Yadira*, chicas, solo ustedes saben cuántas cosas pasaron para llegar hasta aquí, cuán lejano lo veíamos un día. Gracias por todos esos momentos maravillosos que pasamos justas, por estar junto a mí en aquellos años que marcaron mi vida. Solo me queda decirles que hoy, mi corazón estará abierto para ustedes y podrán contar conmigo para siempre.

♥ *A Leyi*, mi gran amiga de la universidad, gracias por soportarme todo este tiempo y por regalarme tantos momentos lindos en estos 5 años. Siempre estarás presente.

♥ *A mis compañeros de tesis*, gracias por darme la posibilidad de compartir con ustedes todo este tiempo, por los tantos momentos de alegrías, de incertidumbres, de situaciones complicadas que superamos juntos, sin ustedes no hubiera sido posible llegar al final. Nunca los olvidaré.

♥ *A mi Decana Sayda*, gracias por ser una madre para nosotros en todos estos años y guiarnos por el buen camino hasta llegar a la meta. Tus hijos nunca te fallarán.

♥ *A mi profe Saba*, gracias por guiarme en todos estos años, su apoyo, sus consejos y su disposición fueron indispensables para llegar a la meta. Hoy lo admiro con todo respeto y le entrego este gran logro. Nunca lo olvidaré.

♥ *A mis primos de la habana*, gracias por acogerme en sus casas aun sin haberme conocido antes, su apoyo, sus consejos, su disposición fueron indispensables para llegar al final, los quiero mucho.

♥ *A mi papá*, hoy después de tantos años, de todo corazón te doy las gracias por tu forma de ser y la manera que supiste actuar ante mis problemas. Gracias a ti aprendí a defenderme ante las situaciones de la vida, a seguir adelante y luchar mucho más fuerte por lograr este objetivo.

♥ A todas las personas que se han preocupado por mi y han contribuido en mi formación profesional y personal.

♥ *Al destino*, por darme el privilegio de haberlos conocido...

Gracias a todos, los llevo en el corazón.



En la vida siempre tenemos nuestros momentos de altas y bajas... en cada uno de ellos he tenido personas maravillosas a mi alrededor, personas a las que no tengo como agradecerles sus horas de infinito amor... hoy, de alguna manera, les hago saber que esos momentos no fueron en vano, pues ayudaron a convertirme en la persona que soy ...

♥ *A mis tíos Dito y Ondina:* a ustedes jamás podré agradecerles con palabras todo lo que han hecho por mi, me acogieron en su hogar y me brindaron su cariño como una hija de corazón, estuvieron presentes en momentos decisivos y siempre me han apoyado en todas mis situaciones y decisiones de la mejor manera, he conocido en ustedes nuevos padres para mí.

♥ *A mis tías Melba, Mari y Magali:* hoy la sobrina más grande o la hija mayor de ustedes alcanza el sueño por el cual tanto hemos luchado, sin su apoyo no habría sido realidad. Reciban hoy, de todo corazón, este regalo porque ha sido por ustedes.

♥ *A mis abuelos:* gracias por sus horas de desvelo, su comprensión, cariño, apoyo, por estar presentes en las decisiones más importantes de mi vida. Los llevo en el corazón.

♥ *A mi tío Chino:* gracias por tu apoyo en todos estos años, por tus palabras de alientos y tus consejos.

♥ *A mi papá:* gracias por darme la vida y por estar presente cuando más te necesito, hoy te doy un motivo más para estar orgulloso de mí.

♥ *A mi prima Ily:* gracias por tus preocupaciones, consejos, por estar siempre para mí en cualquier momento... hoy puedo decir que eres mi hermana mayor.

♥ *A mi primo Leonardo:* ¿Cómo está la escuela?... Estudia mucho... esas siempre fueron palabras tuyas, hoy puedo decirte que estudié lo suficiente para llegar hasta aquí... gracias por todo.

♥ *A mi familia en general:* cuando nacemos no nos dan la posibilidad de escoger nuestra familia, pero si volviera a nacer y me dieran esa oportunidad sin vacilar los escogería a cada uno de ustedes...

♥ *A mis compañeros de tesis:* gracias por los tantos momentos de alegrías, de incertidumbres, de situaciones complicadas que superamos juntos, sin ustedes no hubiera sido posible llegar al final. Nunca los olvidaré.

♥ *A mis amigos Madver, Daisy, Yeni:* además de mi familia los que siempre han estado sin titubear para mí han sido ustedes, para oír mis enredos y mis momentos de ser yo... gracias por poder contar con ustedes, los quiero muchísimo.

A todos aquellos que algún día se preocuparon por mí en estos años de intensa labor, aquellos que preguntaron alguna vez... ¿Keidys cómo te va todo?...

Lleguen a ellos mis más sinceros agradecimientos... gracias por estar ahí cuando más los necesitaba...

 *Resumen.*

La producción de software está estrechamente vinculada al control de la calidad en cada una de las etapas de su desarrollo. Con este objetivo, fue creado el Laboratorio Central de Calidad de Software en la UCI, siendo su principal tarea la realización de diferentes pruebas para determinar el grado de certificación de los productos desarrollados. Este proceso actualmente se realiza de forma manual, lo que dificulta el seguimiento de los recursos (humanos, medios computarizados y tiempo) empleados en la ejecución de dichas pruebas. La Herramienta Informática propuesta permite obtener un control sistemático del trabajo realizado por cada uno de los probadores involucrados en el proceso, posibilitando optimizar el empleo del tiempo utilizado y a partir de estas estimaciones, el Jefe de prueba puede emitir una evaluación que será seguida sistemáticamente; además, posibilita llevar un control real de los medios computarizados disponibles para la realización de las diferentes pruebas evitando que la asignación máquina-estudiante se realice de forma ineficiente. Esta aplicación genera diferentes reportes facilitando actualizar la información en el Laboratorio Central y responder a las solicitudes de certificación y cancelación en menor tiempo; la calidad, confiabilidad y rapidez de la información obtenida trae consigo un aumento de la productividad del trabajo y humanización del mismo. Se propone realizar la programación de la herramienta haciendo uso del Software Libre que en su mayoría se distribuye bajo la licencia pública general GNU/GPL.

 *Summary*

The production of software is closely linked to quality control at each step of development. With that objective, the Central Laboratory of Software Quality was created at the University of Informatics Sciences (UCI), with the main task of performing different tests to determine the level of certification of the developed products. Currently this process is performed manually; this practice creates difficulties in tracking the proper use of resources (human, computer and time) employed performing the aforementioned tests. The informatic tool proposed allows a systematic control of the work performed by each one of the students involved in this process, this creates the possibility of optimizing the time spent on it and following those estimations the quality assessor can propose an evaluation that will be followed systematically. This informatic tool also allows a control in real time of the computerized resources for the different tests preventing an inefficient assignment of computers to students. This application generates different reports which facilitates making current the information at the Central Laboratory and answer the requests of certification and cancellation in less time. The quality, trustworthiness, and speed of the obtained information bring with it an increase in the productivity of the work performed and a humanizing of it. This tool will be programmed using the free software that is distributed under license public general GNU/GPL.


**Índice de contenido.**

Introducción.....	1
Capítulo 1: Fundamentación Teórica.....	8
Introducción.....	8
1.1 La calidad de software.....	8
1.1.1 Tendencia internacional actual en la calidad de software.....	10
1.1.2 La calidad de software en Cuba.....	13
1.1.3 La calidad de software en la Universidad de las Ciencias Informáticas.....	15
1.2 Proceso de seguimiento y evaluación.....	15
1.3 Descripción de las tendencias y tecnologías actuales de software más importantes en el mundo.....	18
1.3.1 Tecnología .Net.....	18
1.3.2 ASP.NET.....	18
1.3.3 Hypertext Preprocesor (PHP).....	19
1.4 Lenguajes de programación Web.....	21
1.4.1 Java Server Pages (JSP).....	21
1.4.2 Java Script.....	21
1.4.3 Perl.....	22
1.4.4 PHP.....	22
1.4.5 Fundamentación del lenguaje a utilizar.....	25
1.5 Metodologías de desarrollo de software.....	26
1.5.1 Proceso unificado de desarrollo de software (RUP).....	26
1.5.2 Extreme Programing (XP).....	27
1.6 Lenguaje de Modelamiento Unificado (UML - Unified Modeling Language).....	27
1.6.1 UML como solución.....	28
1.6.2 Fundamentación del lenguaje de modelado, la metodología y herramienta Case, a utilizar.....	28
1.7 Diseño de aplicaciones Web.....	30
1.8 Arquitectura de aplicaciones.....	32
1.8.1 Arquitectura de dos capas.....	32
1.8.2 Arquitectura de tres capas.....	32
1.8.3 Fundamentación de la arquitectura a utilizar.....	32
1.8.4 Servidor de Aplicaciones.....	33
1.9 Diseño de Interfaz.....	34
1.10 Sistemas Gestores de Base de Datos (SGBD).....	35
1.10.1 SQL (Structure Query Language).....	35
1.10.2 MySQL.....	36
1.10.3 SQL Server.....	36
1.10.4 Fundamentación del sistema de Gestión de Base de Datos a utilizar.....	37
Conclusiones.....	37
Capítulo 2. Descripción del sistema propuesto.....	38
Introducción.....	38
2.1 Descripción de la situación problemática.....	38
2.1.1 Objetivos estratégicos de la organización y procesos del negocio que la soportan.....	39
2.1.2 Flujo actual de los procesos involucrados en el campo de acción.....	40
2.1.3 Análisis crítico de la ejecución actual de los procesos.....	40
2.1.4 Procesos objeto de automatización.....	41

2.1.5 Sistemas automatizados existentes vinculados al campo de acción.....	41
2.2 Descripción del negocio.....	44
2.3 Reglas del negocio.....	44
2.4 Actores del negocio.....	45
2.5 Diagrama de casos de uso del negocio.....	46
2.6 Trabajadores del negocio.....	46
2.7 Casos de uso del negocio.....	47
2.7.1 Caso de Uso Administrar Recursos.....	47
2.8 Definición de las entidades del negocio.....	48
2.9 Propuesta del sistema.....	49
2.10 Especificación de los requerimientos del software.....	50
2.10.1 Requisitos funcionales.....	50
2.10.2 Requisitos no funcionales.....	51
2.11 Actores del sistema.....	53
2.12 Modelo de casos de uso del sistema.....	54
2.13 Definición de los casos de uso del sistema.....	54
Conclusiones.....	59
Capítulo 3. Análisis y diseño del sistema.....	60
Introducción.....	60
3.1 Análisis.....	60
3.2 Modelo conceptual de clases de análisis.....	60
3.2.1 Diagrama de clases del análisis caso de uso “Gestionar datos de Jefe de prueba”.....	61
3.2.2 Diagrama de clases del análisis caso de uso “Gestionar datos de Probador”.....	62
3.2.3 Diagrama de clases del análisis caso de uso “Gestionar pc”.....	62
3.3 Diagramas de interacción.....	63
3.3.1 Diagrama de secuencia caso de uso “Gestionar datos de Jefe de prueba”. Sección: “Modificar Jefe de prueba”.....	63
3.3.2 Diagrama de secuencia caso de uso “Gestionar datos de probador”. Sección: “Insertar estudiante”.....	64
3.3.3 Diagrama de secuencia caso de uso “Gestionar pc”.....	65
3.4 Diseño.....	65
3.4.1 Diagrama de clases del diseño caso de uso “Gestionar datos de Jefe de prueba”.....	66
3.4.2 Diagrama de clases del diseño caso de uso “Gestionar datos de Probador”.....	67
3.4.3 Diagrama de clases del diseño caso de uso “Gestionar pc”.....	68
3.5 Diseño de la Base de Datos.....	68
3.5.1 Modelo entidad-relación.....	69
3.5.2 Descripción de las tablas de la base de datos. Módulo Seguimiento de los recursos.....	69
3.6 Principios de diseño.....	76
3.7 Tratamiento de errores.....	77
3.8 Seguridad.....	78
3.9 Concepción de la ayuda.....	79
Conclusiones.....	80
Capítulo 4. Implementación y prueba.....	81
Introducción.....	81
4.1 Implementación.....	81
4.1.1 Diagrama de despliegue.....	82
4.1.2 Diagrama de componentes.....	82
4.2 Prueba.....	87
Conclusiones.....	97
Conclusiones.....	98



Recomendaciones.....	99
Bibliografía.....	100
Anexos.....	XX
Anexo I: Cuaderno de trabajo del probador: documento donde se registran los datos de los trabajos realizados por el probador.....	XX
Anexo II: Planilla de control de pc: documento donde queda registrado el uso de cada pc.....	XXI
Anexo III: Descripción del caso de uso “Gestionar datos de Jefe de Prueba”.....	XXI
Flujos alternos Sección “Insertar un nuevo Jefe de prueba”.....	XXIII
Flujos alternos Sección “Buscar y visualizar datos de un Jefe de prueba”.....	XXIV
Anexo IV: Descripción del caso de uso “Gestionar datos del Probador”.....	XXVI
Flujos alternos Sección “Insertar datos de un nuevo Probador”.....	XXVIII
Flujos alternos Sección “Buscar y visualizar datos de Probador”.....	XXIX
Anexo V: Descripción del caso de uso “Evaluar Probador”.....	XXXI
Anexo VI: Descripción del caso de uso “Gestionar PC”.....	XXXII
Anexo VII: Descripción del caso de uso “Generar Reporte”.....	XXXIII
Anexo VIII: Descripción del caso de uso “Verificar cumplimiento de Métrica”.....	XXXVII
Anexo IX: Descripción del caso de uso “Gestionar cursos optativos”.....	XXXVIII
Anexo X: Descripción del caso de uso “Gestionar Usuarios”.....	XL
Anexo XI: Descripción del caso de uso “Autenticarse”.....	XLIV
Anexo XII: Base de datos General.....	XLVI
Glosario de Términos.....	XLVII

*Índice de figuras.*

<i>Figura 1.1 Esquema del funcionamiento de las páginas PHP.....</i>	<i>22</i>
<i>Figura 1.2 Esquema de representación del PHP.....</i>	<i>23</i>
<i>Figura 1.3 Representación del servicio Web.....</i>	<i>30</i>
<i>Figura 2.1 Diagrama de casos de uso del negocio.....</i>	<i>45</i>
<i>Figura 2.2 Diagrama de actividades del CU Administrar recursos.....</i>	<i>47</i>
<i>Figura 2.3 Diagrama de clases del modelo de objeto.....</i>	<i>48</i>
<i>Figura 2.4 Diagrama de casos de uso del sistema.....</i>	<i>53</i>
<i>Figura 3.1 Diagrama de clases del análisis CU “Gestionar datos de Jefe prueba”.....</i>	<i>60</i>
<i>Figura 3.2 Diagrama de clases del análisis CU “Gestionar datos de Probador”.....</i>	<i>61</i>
<i>Figura 3.3 Diagrama de clases del análisis CU “Gestionar pc”.....</i>	<i>61</i>
<i>Figura 3.4 Diagrama de secuencia CU “Modificar datos de Jefe de prueba”.....</i>	<i>62</i>
<i>Figura 3.5 Diagrama de secuencia CU “Gestionar datos del Probador”. Sección: “Insertar estudiante”.....</i>	<i>63</i>
<i>Figura 3.6 Diagrama de secuencia CU “Gestionar pc”.....</i>	<i>64</i>
<i>Figura 3.7 Diagrama de clases del diseño CU “Gestionar datos de Jefe de prueba”.....</i>	<i>65</i>
<i>Figura 3.8 Diagrama de clases del diseño CU “Gestionar datos de Probador”.....</i>	<i>66</i>
<i>Figura 3.9 Diagrama de clases del diseño CU “Gestionar pc”.....</i>	<i>67</i>
<i>Figura 3.10 Diagrama entidad de relación. Módulo: Seguimiento de los recursos.....</i>	<i>68</i>
<i>Figura 3.11 Mensaje de verificación. Eliminar usuario.....</i>	<i>76</i>
<i>Figura 3.12 Mensaje de validación. Campo Correo electrónico.....</i>	<i>76</i>
<i>Figura 3.13 Mensaje de validación. Campo Solapín.....</i>	<i>77</i>
<i>Figura 3.14 Mensaje de validación. Campo Nombre.....</i>	<i>77</i>
<i>Figura 3.15 Mensaje de verificación. Campo Contraseñas.....</i>	<i>78</i>
<i>Figura 3.16 Interfaz de autenticación.....</i>	<i>78</i>
<i>Figura 4.1 Diagrama de despliegue.....</i>	<i>81</i>
<i>Figura 4.2 Diagrama de componentes por paquetes.....</i>	<i>82</i>
<i>Figura 4.3 Diagrama de componentes. Paquete: Seguimiento de los recursos.....</i>	<i>83</i>
<i>Figura 4.4 Paquete: Seguimiento de los recursos. Vista Administrador.....</i>	<i>83</i>
<i>Figura 4.5 Paquete: Seguimiento de los recursos. Vista Especialista de calidad.....</i>	<i>84</i>
<i>Figura 4.6 Paquete: Seguimiento de los recursos. Vista Jefe de prueba.....</i>	<i>85</i>
<i>Figura 4.7 Diagrama de componentes. Paquete: Acceso a datos.....</i>	<i>86</i>

*Índice de tablas.*

<i>Tabla 2.1 Descripción de los actores del negocio.....</i>	<i>45</i>
<i>Tabla 2.2 Descripción de los trabajadores del negocio.....</i>	<i>46</i>
<i>Tabla 2.3 Actores del sistema.....</i>	<i>52</i>
<i>Tabla 2.4 Definición del CU “Gestionar datos de Jefe de prueba”.....</i>	<i>54</i>
<i>Tabla 2.5 Definición del CU “Gestionar datos de Probador”.....</i>	<i>54</i>
<i>Tabla 2.6 Definición del CU “Evaluar Probador”.....</i>	<i>55</i>
<i>Tabla 2.7 Definición del CU “Gestionar pc”.....</i>	<i>55</i>
<i>Tabla 2.8 Definición del CU “Generar reportes”.....</i>	<i>55</i>
<i>Tabla 2.9 Definición del CU “Verificar cumplimiento de métricas”.....</i>	<i>56</i>
<i>Tabla 2.10 Definición del CU “Gestionar cursos optativos”.....</i>	<i>56</i>
<i>Tabla 2.11 Definición del CU “Gestionar usuarios”.....</i>	<i>57</i>
<i>Tabla 2.12 Definición del CU “Autenticarse”.....</i>	<i>57</i>
<i>Tabla 3.1 Descripción de la tabla Probador.....</i>	<i>69</i>
<i>Tabla 3.2 Descripción de la tabla Estudiante.....</i>	<i>69</i>
<i>Tabla 3.3 Descripción de la tabla Trabajadores.....</i>	<i>70</i>
<i>Tabla 3.4 Descripción de la tabla Evaluación.....</i>	<i>70</i>
<i>Tabla 3.5 Descripción de la tabla Cursos_optativos_est.....</i>	<i>71</i>
<i>Tabla 3.6 Descripción de la tabla Cursos.....</i>	<i>72</i>
<i>Tabla 3.7 Descripción de la tabla Seguimiento_pc.....</i>	<i>72</i>
<i>Tabla 3.8 Descripción de la tabla Causas_maquina.....</i>	<i>73</i>
<i>Tabla 3.9 Descripción de la tabla Jefe_prueba.....</i>	<i>74</i>
<i>Tabla 3.10 Descripción de la tabla Cursos_optativos_prof.....</i>	<i>75</i>
<i>Tabla 4.1 Descripción del caso de prueba de integración.....</i>	<i>87</i>

 *Introducción.*  
■ ■ ■ ■ ■ ■ ■ ■

El desarrollo tecnológico a través de los años ha permitido la introducción de numerosos descubrimientos científicos. Para ello, el hombre ha ido desarrollando diferentes vías que permiten asimilar lo más rápido posible los nuevos conocimientos, obteniéndose avances significativos en esta dirección.

En los tiempos modernos, donde la tecnología es la base de la competitividad y del poder político, militar y económico, los esfuerzos por aplicar los resultados científicos son cada vez mayores. En la actualidad, cuando el mundo se enfrenta a procesos de cambios paradigmáticos, caracterizados por la globalización de la economía, la industria de software se ha convertido en una iniciativa de gran valor para el desarrollo económico. Actualmente, una de las empresas más rentables del mundo es la de software, situación impensable hace 40 años.

En ese sentido, el desarrollo de software constituye un sector de capital importancia y se encuentra en el centro de todas las transformaciones mundialmente; sobre todo si se considera que los grandes temas del momento, como lo son la economía digital, la evolución de las empresas y la administración del conocimiento, se resuelven informáticamente.

La industria del software interviene en todos los procesos que habilitan a la "nueva economía". Entre los países más avanzados en este campo se encuentran Brasil, la India, Irlanda, EEUU y Canadá.

Cuba, a pesar de la situación actual y las trabas ocasionadas por el bloqueo económico impuesto por EEUU, ha decidido insertarse en este mercado por lo que en la última década se ha sometido a una revolución digital con el objetivo de informatizar la sociedad; además, se llevan a cabo importantes programas que han contribuido a fomentar una cultura informática, lo cual posibilita el acceso a los conocimientos más actualizados a nivel mundial. La creación de los Joven Club de computación y de instituciones especializadas en la producción de software, la implementación de modernos laboratorios en todas las escuelas y enseñanzas, así como la construcción de diversos centros dedicados a la formación de profesionales, constituyen fuentes de vital importancia para lograr este empeño.

Hasta el momento, la industria cubana de software ha demostrado resultados satisfactorios, existiendo diferentes empresas dedicadas a este fin, ejemplo de esto son Softel, Casa Sigta, Desoft, Centersoft y como una de las ideas más nuevas la creación de la Universidad de las Ciencias Informáticas.

El recurso más importante en esta industria lo constituyen los técnicos y profesionales de carreras relacionadas con la informática que requieren, además de la especialización técnica, contar con una amplia visión para la creatividad y la innovación, elementos básicos en la producción de software.

La Universidad de las Ciencias Informáticas es una universidad de nuevo tipo, surgida al calor de la Batalla de Ideas en septiembre del 2002. Entre sus principales retos se encuentran la formación académica de miles de jóvenes de diferentes partes del país como futuros profesionales en las ciencias informáticas y contribuir al desarrollo de la economía cubana, situando en uno de sus principales renglones la producción de software.

Esta Universidad ha tomado como concepción la integración docencia–producción con el fin de crear constantemente nuevas habilidades y fortalecer la experiencia en sus educandos, lo cual ha permitido que, a pesar de sus pocos años de fundada, haya obtenido resultados meritorios en la realización de proyectos de gran envergadura con clientes extranjeros.

La exportación de software creados en la UCI muestra hoy resultados satisfactorios, actualmente existen más de 200 proyectos dirigidos a la producción de software distribuidos en las 10 facultades, donde se encuentran vinculados estudiantes de todos los años.

La producción de software está estrechamente vinculada al control de la calidad en cada una de las etapas de su desarrollo, este proceso comienza desde el levantamiento de requisitos y se hace sistemático hasta el punto final, donde el cliente queda satisfecho y el producto obtiene un grado de certificación en correspondencia con las normas internacionales.

A lo largo del tiempo el concepto de calidad ha adquirido un carácter multidimensional, debido a que diversos autores lo han enfocado desde diferentes puntos de vistas: Deming, como *el grado predecible de uniformidad y conformidad a un bajo costo que se ajuste a las necesidades del mercado*; Feigenbaum, como *el conjunto total de las características del producto de marketing, ingeniería, fabricación y mantenimiento a través del cual el producto en uso satisfará las expectativas del cliente* y Jurán, como *la idoneidad o aptitud para el uso*.

En todas estas definiciones se valoran elementos comunes como la satisfacción de necesidades o perspectivas del cliente y la existencia de rasgos o requisitos para satisfacerlas, lo cual ha decursado por varios estadios, comenzando con la verificación y terminando en la calidad total.

La tendencia internacional actual está fundamentada en la aplicación y certificación sobre la base de las normas ISO 9000 que suponen un lenguaje común, adoptado ya por un elevado número de países.

De ahí, que se pueda considerar como el concepto más actual de calidad el definido por la ISO 9000:2000, considerándola como *grado en el que un conjunto de características inherentes cumplen con los requisitos*.

Además de la aplicación de las normas ISO 9000, para lograr una efectiva gestión de la calidad en determinados sectores, es necesario compatibilizarlas con otras normas específicas adecuadas al tipo de actividad que desarrollan. Tal es el caso de las empresas de la industria del software donde se usan metodologías tan extendidas como el CMM, CMMI, IEEE y la ISO SPICE, entre otros modelos.

Incursionar en otros mercados requiere contar con un aval sólido que demuestre la calidad del producto. Es entonces cuando entran en juego los procesos de certificación basados en las normas internacionales ISO 9000-2000.

En el mundo existen diferentes laboratorios de certificación de software. Ejemplo de esto son el Inqalab, de España, así como el ubicado en la universidad de Carleton, Canadá y en la ciudad de Córdoba, Argentina.

Con el objetivo de controlar la calidad de los productos en el proceso productivo, en el año 2005 fue creado el Laboratorio Central de Calidad de Software en la UCI, siendo su principal tarea la realización de diferentes pruebas para determinar el grado de certificación del producto. Para ello es necesaria la participación de un grupo de estudiantes, quienes son asignados a la realización del trabajo en dependencia de sus horarios docentes.

Con la observación de la práctica diaria, el intercambio con especialistas y estudiantes relacionados con el tema y la experiencia adquirida en el desempeño pre-profesional se pudieron constatar algunas tendencias y regularidades que conllevaron a la determinación de la problemática a investigar. Como principal insuficiencia detectada se encuentra:

- El proceso del control de la calidad de software actualmente se realiza de forma manual, lo que dificulta el seguimiento del empleo correcto de los recursos en la realización de las diferentes pruebas a los productos desarrollados en este centro.

Por la no existencia de un sistema capaz de gestionar con la operatividad requerida todo lo referente al seguimiento de recursos se han determinado las siguientes **tendencias**:

1. No se tiene un control sistemático del trabajo realizado por cada uno de los estudiantes.
2. Existen limitaciones en el empleo del tiempo real por parte del estudiante para realizar las diferentes pruebas de software por las dificultades que se presentan con la planificación del mismo.
3. No se tiene un control del estado real de los medios computarizados disponibles para realizar el trabajo.
4. La asignación máquina–estudiante no se realiza de forma eficiente.
5. No se cumple con el tiempo asignado para la revisión de la calidad del producto desarrollado.
6. No existen reportes que muestren el seguimiento del trabajo realizado en cada una de las iteraciones en la revisión del producto.

A partir de lo antes expuesto se plantea el siguiente **problema científico**:

¿Cómo darle un seguimiento eficiente a los recursos empleados en el proceso de revisión de calidad de los productos desarrollados en la UCI?

Determinándose como **objeto de estudio** el proceso de revisión y control de la calidad de los productos desarrollados en la UCI y como **campo de acción** el seguimiento de los recursos humanos, medios computarizados y tiempo en la realización de las pruebas de software.

El **principal objetivo** es: Elaborar un sistema automatizado que permita darle un seguimiento eficiente a los recursos empleados en el proceso de revisión de calidad a los productos desarrollados en la UCI.

De él se derivan los siguientes **objetivos específicos**:

1. Caracterizar el seguimiento de los recursos empleados en el proceso de revisión de calidad de los productos desarrollados en la UCI.
2. Diseñar una aplicación informática para dar un seguimiento eficiente a los recursos empleados en el proceso de revisión de calidad de los productos desarrollados en la UCI.

3. Implementar una aplicación Web para dar un seguimiento eficiente a los recursos empleados en el proceso de revisión de calidad de los productos desarrollados en la UCI.
4. Realizar las pruebas de integración con los diferentes módulos por los que está compuesta la herramienta que se propone.

A partir de entonces se plantea la siguiente **hipótesis**: Con la creación de un sistema automatizado basado en la utilización de MySQL como gestor de bases de datos y un intérprete como PHP en el diseño e implementación de páginas Web dinámicas es posible darle un seguimiento eficiente a los recursos empleados en el proceso de revisión de calidad de los productos desarrollados en la UCI.

Los pasos a seguir en el proceso de investigación están centrados en las siguientes **tareas científicas**:

1. Fundamentación teórica del proceso de revisión de calidad de los productos desarrollados en la UCI.
2. Caracterización del seguimiento a los recursos empleados en el proceso de revisión de calidad.
3. Diseño de un sistema automatizado para dar seguimiento a los recursos empleados en el proceso de revisión de calidad.
4. Implementación de una aplicación Web que facilite el seguimiento eficiente a los recursos empleados en el proceso de revisión de calidad.

Para hacer más confiable esta investigación se utilizaron **métodos teóricos y empíricos**. La selección de los mismos está en función del objeto a investigar.

### **Nivel Teórico**

Histórico y lógico: para comprender el movimiento histórico por el que ha atravesado el objeto y reproducir en el plano teórico las cuestiones esenciales en el proceso de revisión de calidad de los productos desarrollados.

Análisis y síntesis: se realiza para la obtención de la información sobre el seguimiento a los recursos empleados en el proceso de revisión de calidad de software y así poder elaborar los fundamentos teóricos de la investigación.



Inducción–Deducción: para establecer conjeturas a partir del conocimiento del estado actual en el seguimiento a los recursos empleados en el proceso de revisión de la calidad del software y realizar inferencias basadas en el estudio bibliográfico, hasta hacer generalizaciones.

### ***Nivel Empírico.***

Observación científica: para constatar el estado actual de la problemática mediante la observación al proceso de seguimiento de los recursos en el Laboratorio Central de Calidad de Software.

El presente trabajo permite realizar de forma automatizada, con mayor brevedad y eficiencia el seguimiento de los recursos empleados en el proceso de certificación de software, posibilitando un mayor control de los medios computarizados, el tiempo y los recursos humanos. Además, con el empleo de este software se obtienen diferentes tipos de reportes en dependencia de las necesidades de los usuarios.

El documento está estructurado de la siguiente manera:

### **Capítulo 1. Fundamentación teórica.**

Este capítulo comprende el estado del arte sobre el tema tratado, hace una breve explicación del problema, las tendencias actuales que existen sobre la Calidad de Software y el seguimiento de los recursos a nivel nacional e internacional. Se describe la justificación de las tecnologías y metodologías utilizadas para la solución del problema.

### **Capítulo 2. Características del sistema.**

En este capítulo se hace una definición del objeto de estudio del problema, se plantean los objetivos estratégicos de la organización y procesos del negocio que los soportan, se realiza la descripción de los procesos que serán objeto de automatización y de la propuesta del sistema, se define el Modelo de negocio, especificación de los requisitos del sistema y casos de uso.

### **Capítulo 3. Análisis y diseño del sistema.**

En el capítulo se plantean las definiciones del Modelo de Análisis, Modelo de clases del análisis, además se realizan los diferentes diagramas de diseño, así como el Modelo entidad relación y las descripciones de las tablas de la base de datos y de las diferentes clases utilizadas.

**Capítulo 4. Implementación y prueba.**

En el capítulo se plantean las definiciones de los diferentes modelos utilizados en la implementación para la solución del problema y se procede a la construcción de los requisitos. Se realiza el Modelo de despliegue y el Modelo de componentes.

 *Capítulo 1: Fundamentación Teórica.***Introducción.**

En el presente capítulo se describen los conceptos fundamentales asociados a la calidad de software y al proceso de seguimiento y evaluación en un proyecto u organización. Se realiza un estudio comparativo entre las principales tendencias y tecnologías actuales con el objetivo de fundamentar la más recomendable para la realización de la propuesta. Por otra parte, se fundamenta el uso del lenguaje de programación y el gestor de base de datos utilizados exponiendo las características que los hacen distintivos.

**1.1 La calidad de software.**

La creciente necesidad de realizar sistemas complejos en cortos periodos de tiempo, con menores esfuerzos tanto humanos como económicos, ha favorecido el crecimiento de una cultura que aboga por realizar los productos cada vez más con mayor eficiencia. Esto ha provocado que en los últimos tiempos el término de calidad sea frecuentemente utilizado y aparezca en multitud de contextos, pues con él se busca despertar en quien lo escucha una sensación positiva, transmitiendo la idea de que algo es mejor, es decir, la idea de excelencia.

Actualmente existen diferentes criterios sobre calidad. A lo largo del tiempo muchos han sido los que, de alguna forma u otra, han aportado nuevas ideas a esta rama, por lo que este concepto está matizado por diferentes puntos de vista. Algunos de los más conocidos y utilizados son:

- “Es el conjunto de características de un producto que satisfacen las necesidades de los clientes y, en consecuencia, hacen satisfactorio al producto” (Aiteco 2006).
- “Conjunto de propiedades y características de un producto o servicio que le confieren su aptitud para satisfacer unas necesidades explícitas o implícitas” (ISO 84)
- “Grado en el que un conjunto de características inherentes cumple con los requisitos.” (ISO 2000)

Al analizar estos conceptos se pueden apreciar dos elementos de gran importancia a tener en cuenta: el grupo de rasgos o caracteres de un producto (necesariamente suficientes) y que a su vez satisfagan intereses o necesidades. Es por ello que todas las empresas productoras están haciendo lo posible por brindarles a sus clientes productos de máxima calidad. La producción de software no está exenta de esto. En la actualidad existen ya algunas definiciones de calidad del software, tales como:

- La calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario".(IEEE 1990)
- “Concordancia del software producido con los requerimientos explícitamente establecidos, con los estándares de desarrollo prefijados y con los requerimientos implícitos no establecidos formalmente, que desea el usuario” (S. 2005)

Después de hacer un análisis valorativo, las autoras consideran que la calidad de software es uno de los requisitos más importantes en el ciclo de vida de un producto. Esta comienza desde la primera fase de desarrollo y se hace sistemática hasta el punto donde se obtiene un conjunto de cualidades que lo hagan eficiente y a la vez responda a las necesidades específicas de los clientes.

La obtención de un producto con calidad implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software que permitan uniformar la filosofía de trabajo, en aras de lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad. La política establecida debe estar sustentada sobre tres principios básicos: tecnológico, administrativo y ergonómico.

- El principio tecnológico define las técnicas a utilizar en el proceso de desarrollo del software.
- El principio administrativo contempla las funciones de planificación y control del desarrollo del software, así como la organización del ambiente o centro de ingeniería de software.
- El principio ergonómico define la interfaz entre el usuario y el ambiente automatizado.

Para controlar la calidad del software es necesario, ante todo, definir los parámetros, indicadores o criterios de medición, pues como plantea Tom De Marco, "usted no puede controlar lo que no se puede medir".

Las cualidades para medir la calidad del software son definidas por innumerables autores, los cuales las denominan y agrupan de formas diferentes. Por ejemplo, John Wiley define métricas de calidad y criterios, donde cada métrica se obtiene a partir de combinaciones de los diferentes criterios. La Metodología para la evaluación de la calidad de los medios de programas de la CIC, de Rusia, define indicadores de calidad estructurados en cuatro niveles jerárquicos: factor, criterio, métrica y elemento de evaluación, donde cada nivel inferior contiene los indicadores que conforman el nivel precedente. Otros autores identifican la calidad con el nivel de complejidad del software y definen dos categorías de métricas: de complejidad de programa o código, y de complejidad de sistema o estructura.

Todos los autores coinciden en que el software posee determinados índices medibles que son las bases para la calidad, el control y el perfeccionamiento de la productividad.

Una vez seleccionados los índices de calidad, se debe establecer el proceso de control, el cual requiere los siguientes pasos:

- Definir el software que va a ser controlado: clasificación por tipo, esfera de aplicación, complejidad, entre otros, de acuerdo con los estándares establecidos para el desarrollo del software.
- Seleccionar una medida que pueda ser aplicada al objeto de control. Para cada clase de software es necesario definir los indicadores y sus magnitudes.
- Crear o determinar los métodos de valoración de los indicadores: métodos manuales como cuestionarios o encuestas estándares para la medición de criterios periciales y herramientas automatizadas para medir los criterios de cálculo.
- Definir las regulaciones organizativas para realizar el control: quiénes participan en el control de la calidad, cuándo se realiza, qué documentos deben ser revisados y elaborados.

### **1.1.1 Tendencia internacional actual en la calidad de software.**

Hoy en día, las compañías que producen software deben buscar continuamente alternativas que les permitan mejorar la calidad de sus productos para seguir compitiendo en un escenario cada vez más globalizado y agresivo.

La industria tiene una fuerte necesidad de optimización de todos los aspectos del desarrollo de productos. A nivel mundial, la calidad del proceso utilizado para desarrollar un determinado producto

impacta fuertemente en la calidad final. En consecuencia, el mejoramiento de los procesos de desarrollo forma parte de la estrategia competitiva.

Con la evaluación de calidad una empresa busca diferenciarse, dar un valor agregado. Y más allá de las diferentes prácticas que se desarrollan en la industria del software para trabajar con el exterior, la calidad es una exigencia que facilita la comunicación entre compañías que han alcanzado el mismo nivel de madurez.

Existen diferentes normas que posibilitan medir la calidad de los productos permitiendo obtener un lenguaje común adoptado por un gran número de países. Dentro de ellas se pueden destacar:

- **CMM - CMMI** es un modelo de calidad del software que clasifica las empresas en niveles de madurez. Estos niveles sirven para conocer la madurez de los procesos que se realizan para producir software.(Joaquin 2005)

#### **Niveles CMM - CMMI**

**Inicial o Nivel 1 CMM - CMMI.** Este es el nivel donde están insertadas todas las empresas que no tienen procesos.

**Repetible o Nivel 2 CMM - CMMI.** Quiere decir que el éxito de los resultados obtenidos se puede repetir. La principal diferencia entre este nivel y el anterior es que el proyecto es gestionado y controlado durante el desarrollo del mismo. El desarrollo no es opaco y se puede saber el estado del proyecto en todo momento.

Los procesos que hay que implantar para alcanzar este nivel son:

- Gestión de requisitos
- Planificación de proyectos
- Seguimiento y control de proyectos
- Gestión de proveedores
- Aseguramiento de la calidad
- Gestión de la configuración

**Definido o Nivel 3 CMM - CMMI.** Alcanzar este nivel significa que la forma de desarrollar proyectos (gestión e ingeniería) está definida, es decir, que está establecida, documentada y que existen métricas (obtención de datos objetivos) para la consecución de objetivos concretos.

Los procesos que hay que implantar para alcanzar este nivel son:

- Desarrollo de requisitos

- Solución Técnica
- Integración del producto
- Verificación
- Validación
- Desarrollo y mejora de los procesos de la organización
- Definición de los procesos de la organización
- Planificación de la formación
- Gestión de riesgos
- Análisis y resolución de toma de decisiones

La mayoría de las empresas que llegan al nivel 3, se detienen aquí, ya que es un nivel que proporciona muchos beneficios y no ven la necesidad de ir más allá porque tienen cubierta la mayoría de sus necesidades.

**Cuantitativamente Gestionado o Nivel 4 CMM - CMMI.** Los proyectos usan objetivos medibles para alcanzar las necesidades de los clientes y la organización. Se usan métricas para gestionar la organización.

Los procesos que hay que implantar para alcanzar este nivel son:

- Gestión cuantitativa de proyectos
- Mejora de los procesos de la organización

**Optimizado o Nivel 5 CMM - CMMI.** Los procesos de los proyectos y de la organización están orientados a la mejora de las actividades. Mejoras incrementales e innovadoras de los procesos que mediante métricas son identificadas, evaluadas y puestas en práctica.

Los procesos que hay que implantar para alcanzar este nivel son:

- Innovación organizacional
- Análisis y resolución de las causas

Normalmente las empresas que intentan alcanzar los niveles 4 y 5 lo realizan simultáneamente ya que están muy relacionados.

La implantación de un modelo de estas características es un proceso largo y costoso que puede consumir varios años de esfuerzo.

- **ISO 9000**

Con el objetivo de estandarizar los sistemas de calidad de las diferentes empresas y sectores se publican las normas ISO 9000, que son un conjunto de normas editadas y revisadas periódicamente por la Organización Internacional de Normalización (ISO) sobre la garantía de calidad de los procesos.(ISO 2000)

La tendencia internacional actual está fundamentada por la aplicación y certificación de sus productos sobre la base de esta norma por las siguientes características:

- Publicada en el año 1987. Adoptada por más de 90 países.
- Cuenta con directrices para la gestión del sistema de calidad y modelos de garantía para la empresa.
- Las directrices son genéricas y aplicables a cualquier sector.
- Es un marco de trabajo para la mejora continua.

Con la versión surgida en el año 2000 de la familia de normas ISO 9000, se introdujo el enfoque de procesos y los tres modelos (ISO 9001, ISO 9002 e ISO 9003) se unieron en el modelo ISO 9001:2000 aplicable a cualquier organización. Además la norma ISO 8402 se sustituyó por la ISO 9000: Vocabulario y la ISO 9004 se convirtió en el modelo para la mejora del desempeño.(9001 2000)

En la actualidad, las empresas productoras de software, están viendo la necesidad de que sus productos salgan al mercado con la calidad requerida; por este motivo surgen los laboratorios de calidad de software, creados con el objetivo de medir el grado de calidad de un producto.

A nivel mundial existen diferentes laboratorios de este tipo, tales como: el Inqalab, de España; el ubicado en la Universidad de Carleton, Canadá; así como el LabCS, de la Universidad Tecnológica Nacional Facultad Regional Mendoza, el INTI-Rosario y el laboratorio en la Universidad Austral ubicados todos en Argentina.

### **1.1.2 La calidad de software en Cuba.**

La industria cubana del software tiene pocos años de fundada. En la década de los 90, existían unas pocas empresas que desarrollaban algunos pequeños programas informáticos con el fin de solucionar un problema interno. Con el perfeccionamiento constante y creciente de las nuevas tecnologías de la información y las comunicaciones, Cuba se vio en la necesidad de insertarse en este mercado; para lograr ese objetivo se desarrolló un conjunto de acciones y se tomaron nuevas medidas.



Como resultado, fueron creadas varias empresas dedicadas a la producción de software como Desoft, Segurmática, Casa Sigta, Centersoft, Softel, Citmatel, Tecnomática, entre otras.

A pesar de todos los esfuerzos realizados por estas empresas la calidad de software aún es un tema por madurar. Actualmente el país cuenta con pocos especialistas en la materia y en la mayoría de estos centros no tienen los procesos bien definidos, además no existe una política genérica que pueda ser tomada como línea base para desarrollar este proceso.

Con el surgimiento de la Universidad de las Ciencias Informáticas se abrió una nueva brecha en el desarrollo del software unido al proceso de control de la calidad. Actualmente Cuba está trabajando en la creación de un Centro Nacional de Certificación de Software integrado por las universidades, empresas y centros de investigación, para trabajar en aras de lograr la mejora continua de los procesos. Entre los objetivos principales de este centro se destacan:

- Convertirse en un tercero confiable que permita darle seguridad al cliente en el producto que está adquiriendo.
- Contribuir al desarrollo de la industria de software en el país incrementando su competitividad a través de la difusión, la mejora continua y el conocimiento de la tecnología de la información.
- Convertirse en un filtro de calidad para los software comprados con vistas a ser implantados en las organizaciones del país.
- Potenciar la inclusión de Cuba, con éxito, en el mercado internacional del Software.

Son varias las acciones estratégicas que se llevan a cabo para fortalecer esta industria, algunas de las cuales se mencionan a continuación:

- Desarrollo de un programa efectivo de mejora de procesos de software, basado en el modelo de calidad CMMI.
- Acreditación de especialistas con certificados internacionales.
- Crear una Red Temática de Calidad y Mejora de Procesos.
- Creación de SPINs (Software Process Improvement Networks) con el objetivo de difundir el estado del arte y de la práctica, tanto de la ingeniería de software como de la calidad en la misma.
- Creación de el ESI Center (Centro de Excelencia para la Industria del Software) del Caribe en Cuba, con el objetivo de contribuir al desarrollo de la industria, incrementando su

competitividad a través de la difusión, la mejora continua y el conocimiento en tecnologías de la información. (Melchor 2007)

### **1.1.3 La calidad de software en la Universidad de las Ciencias Informáticas.**

La Universidad de las Ciencias Informáticas, como productora de software, se vio en la necesidad de activar un mecanismo para asegurar la calidad de los productos desarrollados. Desde sus primeros años de fundada comenzaron los primeros intentos hacia este objetivo; un grupo de compañeros realizó varias acciones que sentaron las bases y posibilitaron que un conjunto de especialistas en el tema siguiera sus ideas, propiciando la creación del Laboratorio Central de Calidad de Software en el año 2005.

En este laboratorio comenzaron a realizarse diferentes pruebas a los productos terminados con el objetivo de obtener un grado de certificación de acuerdo con las normas internacionales, para esto fue necesaria la participación de un pequeño grupo de estudiantes de las diferentes facultades.

La creciente necesidad de obtener cada vez más, productos de mayor relevancia y en el menor tiempo, posibilitó que fuera desarrollándose una cultura encaminada al conocimiento de todos los procesos que intervienen en la calidad de software.

Actualmente todas las facultades cuentan con un grupo de estudiantes dedicados a garantizar la calidad de los productos en todas las fases de su desarrollo permitiendo que este proceso se realice de forma continua y sistemática, además se están desarrollando un conjunto de trabajos científicos relacionados con esta temática permitiendo elevar el nivel de conocimientos e interés sobre el tema.

### **1.2 Proceso de seguimiento y evaluación.**

El seguimiento es un proceso continuo y constituye un aspecto esencial integrado en el desarrollo de todo proyecto desde sus inicios. Posibilita a los desarrolladores determinar el progreso de las actividades e identificar y anticipar los problemas, permitiéndoles así tomar las medidas precisas para evitarlos o resolverlos, haciendo los ajustes necesarios en los objetivos y actividades, lo que posibilita optimizar al máximo el tiempo y los recursos empleados, así como evitar atrasos y entregas tardías lo cual puede traducirse en una mala calidad del producto.

El seguimiento consiste en el análisis y recopilación sistemáticos de información a medida que avanza un proyecto. Su objetivo es mejorar la eficiencia y efectividad de un proyecto y organización. Se basa en metas establecidas y actividades planificadas en las distintas fases del trabajo. Ayuda a que se siga una línea de trabajo, y además, permite a la administración conocer cuándo algo no está funcionando. Si se lleva a cabo adecuadamente, es una herramienta de incalculable valor para una buena administración y proporciona la base para la evaluación permitiendo determinar si los recursos disponibles son suficientes y están bien administrados, si la capacidad de trabajo es eficiente y adecuada y si se está haciendo lo que está planificado.

Este proceso puede considerarse como indispensable para la realización de cualquier actividad, su elemento más representativo es la retroalimentación, entendida como la información que resulta de comparar los objetivos y las metas de cada eje estratégico con los resultados obtenidos al momento de la evaluación, lo que permite conocer la brecha recorrida y el camino por recorrer entre el estado actual del sistema y la visión al futuro del mismo, las acciones que conviene mantener y las rectificaciones que sea necesario realizar.

En la actualidad existen diferentes conceptos que abordan el tema, ejemplo de esto son:

- El seguimiento es el proceso mediante el cual se recopilan sistemáticamente y con cierta regularidad los datos referidos al desarrollo de un programa a lo largo del tiempo.
- El seguimiento es el proceso de recopilación de información para controlar los resultados con arreglo a determinados criterios.

Las autoras asumen el primer concepto por ser el que más se relaciona con los procesos involucrados en el Laboratorio de Calidad. Se destaca el valor de la frecuencia del tiempo como elemento significativo en el desarrollo del trabajo.

La evaluación consiste en la comparación de los impactos reales del proyecto con los planes estratégicos acordados. Está enfocada hacia lo que se había establecido hacer, lo que se ha conseguido y cómo se ha conseguido. Puede ser:

- formativa: tiene lugar durante la vida de un proyecto u organización con la intención de mejorar la estrategia o el modo de funcionar del proyecto y organización.
- conclusiva: Se obtiene aprendizaje a partir de un proyecto completado o una organización que ya no está en funcionamiento.

El seguimiento y la evaluación constituyen dos conjuntos de actividades organizativas que están muy estrechamente relacionadas. Comparten la misma orientación, hacia un aprendizaje a partir de aquello que se hace y cómo se hace, concentrándose en:

- Eficacia.
- Efectividad.
- Impacto.

La **eficacia** informa sobre la adecuada aportación en el trabajo en cuanto a producción. Podría tratarse de aportación en cuanto a dinero, tiempo, personal y equipamiento, entre otros. Cuando se dirige un proyecto y se está interesado por las posibilidades de su reproducción o su aplicación a escala, entonces la eficacia resulta de gran importancia.

La **efectividad** mide los logros obtenidos por un programa o proyecto de desarrollo en relación con aquellos objetivos específicos que se habían establecido.

El **impacto** informa sobre la influencia causada en la situación del problema que se intentaba afrontar. Antes de tomar la decisión de una ampliación o una reproducción del proyecto en otro ámbito, es necesario estar seguro de que tiene sentido aquello que se está haciendo con relación al impacto que se quiere lograr.

A través del seguimiento y la evaluación se puede:

- Revisar el progreso.
- Identificar problemas en la planificación y/o en la puesta en práctica.
- Encontrar posibles soluciones a estos problemas.
- Hacer ajustes de modo que permita tener más probabilidades de generar un cambio.
- Plantear cuestiones sobre suposiciones o estrategias.

El objetivo principal del seguimiento y evaluación para una organización o proyecto debe ser la observación de su propio trabajo con relación a los objetivos: si está haciendo impacto; si está trabajando eficazmente; y de este modo, aprender a trabajar de mejor manera.

### 1.3 Descripción de las tendencias y tecnologías actuales de software más importantes en el mundo.

#### 1.3.1 Tecnología .Net

El Framework .NET es una infraestructura sobre la que se reúne todo un conjunto de lenguajes y servicios que simplifican el desarrollo de aplicaciones. Mediante esta herramienta se ofrece un entorno de ejecución distribuido, además soporta múltiples lenguajes de programación y es posible desarrollar cualquier tipo de aplicación con cualquiera de ellos: C# (C Sharp), Visual Basic, C++, J# (Java #), Jscript, también se han adaptado lenguajes existentes, como Perl, Python, Fortran, Cobol, entre otros. (Patterson, Brian et al. 2003)

La idea central detrás de la plataforma .NET es la de servicio. Más concretamente software como servicio, y cómo construir o instalar esos servicios para que se puedan acceder y consumir mediante Internet. La plataforma .NET permite usar Internet y su capacidad de distribución para que los usuarios accedan desde cualquier dispositivo, en cualquier sistema operativo y lugar a la funcionalidad que los servicios Web proveen. (Paúl and Lilia 2006)

#### 1.3.2 ASP.NET

ASP.NET es un marco de trabajo de programación generado en Common Language Runtime que puede utilizarse en un servidor para generar eficaces aplicaciones Web. ASP.NET ofrece varias ventajas importantes acerca de los modelos de programación Web, como por ejemplo:

- **Eficiencia**

Desde el principio, uno de los objetivos más importantes del diseño de .NET ha sido su gran rendimiento y nivelación. Para que .NET tenga éxito, las empresas deben estar capacitadas para migrar sus aplicaciones y no sufrir de un rendimiento deficiente debido a la forma en que CLR ejecuta el código. Para asegurarse un óptimo rendimiento, el CLR compila, en algún punto, todos los códigos de aplicaciones en códigos naturales de máquina. Esta conversión puede hacerse, o bien en el momento en que se ejecuta la aplicación (método por método), o cuando se instala la aplicación por primera vez. Este proceso hará uso automáticamente de todas las características del microprocesador, disponibles en diferentes plataformas, algo que las aplicaciones tradicionales de

Windows nunca podrían hacer, a menos que se cargasen distintos binarios para distintas plataformas. (Desarrollo 2003)

- **Soporte de Lenguajes.**

Esta es una de las novedades más importantes que vienen de la mano de ASP.NET. La posibilidad de escribir códigos diferentes como: VisualBasic.Net (VB) y C#, el nuevo lenguaje creado por Microsoft con la intención de aprovechar la potencia del C++ y combinarlo con las facilidades que brinda a la programación en Internet un lenguaje como Java.

- **Contenido y Código, por separado.**

Muchos desarrolladores de sitios Web han tenido que lidiar con el inconveniente de tener que crear la interfaz de usuario y el código ASP, todo junto. Esta mezcla de imágenes, botones y tablas en código HTML con pedazos de código en VBScript o Jscript llegaba a ser algo muy molesto para el desarrollador. ASP.NET soluciona este problema, utilizando un criterio similar al que utiliza Visual Basic, es decir, separa la interfaz de usuario del código. La mayor desventaja que presenta este lenguaje es que solo se puede implementar en los Servidores Web de su desarrollador: Microsoft.

### **1.3.3 Hypertext Preprocesor (PHP)**

La tecnología PHP fue desarrollada en 1994 por Rasmus Lerdorf. La primera versión disponible para el público fue conocida como Personal Home Page Tools (PHP Tools) en 1995. Inicialmente esta versión era un simple conjunto de scripts de Perl para controlar los accesos a su trabajo online e incluía herramientas básicas, como contadores, libro de visitas, entre otras. En este mismo año el intérprete de PHP fue reescrito y se relanzó con una versión llamada PHP/FI versión 2, la cual era capaz de comunicarse con bases de datos y permitía a los usuarios desarrollar sencillas aplicaciones dinámicas. Esta versión se mantuvo para páginas personales y como intérprete de formularios, incluía algunas de las funcionalidades básicas del PHP actual. Tenía variables como las de Perl, interpretación automática de variables de formulario y sintaxis embebida en HTML.(Martín 2004). En 1997, pasó a ser un proyecto Open Source (Código Abierto) para que cualquiera pudiese utilizarlo, así como arreglar errores y mejorar el código.

Con el transcurso del tiempo la tecnología PHP fue desarrollándose y en consecuencia a esto cada día aparecían nuevas versiones con características y funcionalidades mejores a la anterior. Ejemplo

de esto son: el PHP/FI 2.0, la cual fue liberada oficialmente en 1997 y sucedida en breve tiempo por las primeras versiones de PHP 3.0.

Esta versión fue considerada como la primera que se parecía al PHP utilizado en la actualidad. Fue creado por Andi Gutmans y Zeev Zuraski en 1997 reescribiéndolo completamente, después de haber encontrado algunas desventajas para desarrollar aplicaciones comerciales con el PHP/FI 2.0

Una de las características más significativas de PHP 3.0 era su gran extensibilidad. Además de proveer a los usuarios finales de una sólida infraestructura para muchísimas bases de datos, protocolos y APIs, las características de extensibilidad de PHP 3.0 atrajeron a docenas de desarrolladores a unirse y enviar nuevos módulos de extensión. Sin duda, ésta fue la clave del enorme éxito de PHP 3.0. Otras características claves introducidas en PHP 3.0 fueron el soporte de sintaxis orientado a objetos y una sintaxis de lenguaje mucho más consistente y potente. (Jim 2003)

A finales de la década de los 90 surge el nuevo motor, apodado "Motor Zend" (comprimido de sus apellidos, Zeev y Andi), el cual alcanzó satisfacer los objetivos de diseño que no se lograban en la versión anterior, propiciándose que se introdujera por primera vez a mediados de 1999 PHP 4.0, basado en este motor, y acoplado con un gran rango de nuevas características adicionales, fue oficialmente liberado en mayo de 2000, casi dos años después que su predecesor, PHP 3.0. Además de la mejora de ejecución de esta versión, PHP 4.0 incluía otras características claves como el soporte para la mayoría de los servidores Web, sesiones HTTP, buffers de salida, formas más seguras de controlar las entradas de usuario y muchas nuevas construcciones de lenguaje. (Martín 2004)

La principal novedad de PHP4 fue en el intérprete del lenguaje PHP. Gracias a Zend, se lograron aumentos de entre 5 y 10 veces en la velocidad de ejecución de páginas PHP. Este nuevo rendimiento lo puso por delante de ASP, la tecnología de Microsoft, por lo que le permitió competir directamente con ella. Otra nueva característica de PHP4 es que se puede compilar para soportar hebras, algo que no era posible en PHP3. La principal ventaja que se puede obtener de esta funcionalidad es el uso de librerías que utilicen hebras. (Henry 2006)

Actualmente PHP es uno de los lenguajes de creación de páginas Web dinámicas más utilizados del mercado. La quinta versión cuenta con innumerables mejoras que consolidan su éxito. Se ofrece la posibilidad de hacer programas orientados a objetos, lectura de archivos XML de forma sencilla, utilización de la base de datos ligera SQLite o la implementación de servicios Web. Este se diferencia

en varias partes: La primera sección se centra en los fundamentos para el desarrollo de programas, la segunda parte muestra los avances que han surgido en PHP 5, como la programación orientada a objetos, la conexión con las bases de datos MySQL y SQLite, la lectura de archivos XML, el envío de correo electrónico o la utilización del protocolo FTP, y la última división, orientada a usuarios avanzados, cuenta con la descripción de técnicas actuales de desarrollo e incorpora librerías que ayudan a la implementación de archivos RSS, servicios Web XML-RPC, generación de gráficos y plantillas con Smarty.

#### **1.4 Lenguajes de programación Web.**

Actualmente existen diferentes lenguajes del lado del servidor, entre los que más se destacan se encuentran algunos como PERL, ASP, PHP, Java, JSP, los módulos CGIs e ISAPIs, etc. Estos se caracterizan por desarrollar la lógica de negocio dentro del Servidor, además de ser los encargados del acceso a Bases de Datos, tratamiento de la información, etc. Del lado del cliente se encuentran principalmente el Java Script y el Visual Basic Script, que son los encargados de aportar dinamismo a la aplicación en los navegadores.

##### **1.4.1 Java Server Pages (JSP).**

JSP es un acrónimo de Java Server Pages, que en español significa “Páginas de Servidor Java”. Es una tecnología orientada a crear páginas Web con programación en Java, permitiendo que se ejecuten en variados servidores Web, de múltiples plataformas. Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java. (Web 2005)

##### **1.4.2 Java Script.**

Java Script es un lenguaje de scripts desarrollado por Netscape para incrementar las funcionalidades del lenguaje HTML. Se utiliza embebido en el código HTML, entre las tags <script> y </script>. (Java 2005)

Entre sus principales características resaltan:

1. Es un lenguaje interpretado, es decir, no requiere compilación.



2. El navegador del usuario se encarga de interpretar las sentencias Java Script contenidas en una página HTML y ejecutarlas adecuadamente.
3. Java Script es un lenguaje orientado a objetos.
4. El modelo de objetos de Java Script está reducido y simplificado, pero incluye los elementos necesarios para que los Scripts puedan acceder a la información de una página y puedan actuar sobre la interfaz del navegador.
5. Cuando un usuario pincha sobre un enlace o mueve el puntero sobre una imagen, se produce un evento. Mediante Java Script se pueden desarrollar Scripts que ejecuten acciones en respuesta a estos eventos. En este caso actúa como un lenguaje orientado a eventos. (Java 2005)

#### **1.4.3 Perl.**

Es un lenguaje de programación muy utilizado para construir aplicaciones CGI para la Web. Perl es un acrónimo de Practical Extracting and Reporting Language, es un lenguaje de programación muy práctico para extraer información de archivos de texto y generar informes a partir del contenido de los ficheros. (Desarrollo 2003)

Es un lenguaje libre de uso, por lo que es accesible a cualquier tipo de usuario. Antes estaba muy asociado a la plataforma Unix, pero en la actualidad está disponible en otros sistemas operativos como Windows. Perl es un lenguaje de programación interpretado, al igual que muchos otros lenguajes de Internet como Java Script o ASP.

#### **1.4.4 PHP.**

Es un lenguaje para programar scripts del lado del servidor, rápido, con una gran librería de funciones y mucha documentación que se incrustan dentro del código HTML. Este lenguaje es gratuito y multiplataforma puesto que existe un módulo de PHP para casi cualquier servidor Web, esto hace que cualquier sistema pueda ser compatible con el lenguaje y significa una ventaja importante, ya que permite portar el sitio desarrollado en PHP de un sistema a otro de manera fácil y rápida. Además posibilita configurar el servidor de modo que se acepten o rechacen diferentes usos, lo que puede hacer al lenguaje más seguro dependiendo de las necesidades de cada cual.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor Web, justo antes de que se envíe la página al cliente a través de Internet. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. Este solamente recibe una página con el código HTML resultante de la ejecución de la página PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores.



**Figura 1.1. Esquema del funcionamiento de las páginas PHP.** (Albert 2005)

Este lenguaje de programación está preparado para realizar muchos tipos de aplicaciones Web gracias a la extensa librería de funciones con la que está dotado. La librería de funciones cubre desde cálculos matemáticos complejos, hasta tratamiento de conexiones de red.

Existe una gran comunidad de desarrolladores y programadores que continuamente implementan mejoras en su código y están dispuestos a prestar ayuda a los usuarios seguidores de este lenguaje. (José 2003)

El funcionamiento del PHP se puede describir a través de los pasos siguientes:

1. Escribir páginas PHP.
2. Guardar la página en el servidor Web.
3. Un navegador solicita una página al servidor.
4. El servidor interpreta el código PHP.
5. El servidor envía el resultado como código HTML.



**Figura 1.2. Esquema de representación del PHP.**

En ningún caso se envía código PHP al navegador, por lo que todas las operaciones realizadas son transparentes al usuario; el código PHP es ejecutado en el servidor y el resultado enviado al

navegador. El resultado es normalmente una página HTML, por lo que al usuario le parecerá que está visitando una página HTML que cualquier navegador puede interpretar.

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que el navegador lo soporte, es independiente del navegador, pero sin embargo para que sus páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP.

PHP corre sobre 7 plataformas, funciona con 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:

- Es software libre y abierto, lo que implica menos costes y servidores más baratos que otras alternativas.
- Es muy rápido. Su integración con la base de datos MySQL y el servidor Apache, le permite constituirse como una de las alternativas más atractivas del mercado.
- Su sintaxis está inspirada en C, ligeramente modificada para adaptarlo al entorno en el que trabaja, de modo que si se está familiarizado con esta sintaxis, resultará muy fácil aprender PHP. (PHP 2005)

#### 1.4.5 Fundamentación del lenguaje a utilizar.

Se han dado características particulares de los diferentes lenguajes de programación; para fundamentar la elección que se realizó se hará una breve comparación teniendo en cuenta particularidades que influyen en el entorno de trabajo donde se va a desarrollar la propuesta.

- **Características multiplataformas:** Menos el ASP, que es solamente soportado por la plataforma Windows, los demás lenguajes están soportados en múltiples plataformas.
- **Velocidad de ejecución:** La velocidad es mayor en PHP, seguidos por PERL y JSP.
- **Disponibilidad de recursos:** Actualmente los más utilizados en la Internet son el PHP y el JSP, por lo que existen una gran cantidad de artículos y códigos de ejemplos. PHP tiene una de las comunidades más grandes en Internet, al igual que la de Java.
- **Familiaridad con el lenguaje:** En la universidad los lenguajes más utilizados por los programadores son el .NET y el PHP.

Teniendo en cuenta lo expuesto, el PHP es el lenguaje por el que se optará, pues es el más adecuado para implementar la propuesta de la aplicación que se desea desarrollar, ya que es multiplataforma, tiene mayor velocidad de ejecución y existe amplia documentación en el mundo relacionada con él.

### 1.5 Metodologías de desarrollo de software.

Muchas veces al realizarse el diseño de un software se hace de manera rígida, con los requerimientos que el cliente solicitó, de tal forma que cuando en la etapa final (etapa de prueba) se solicita un cambio se hace muy difícil realizarlo, pues altera muchas cosas que no se habían previsto, y es justo éste, uno de los factores que ocasiona un atraso en el proyecto.

En un proyecto de desarrollo de software la metodología define Quién debe hacer Qué, Cuándo y Cómo debe hacerlo.

#### 1.5.1 Proceso unificado de desarrollo de software (RUP).

RUP es un proceso de desarrollo de software, o sea, un conjunto de actividades necesarias para transformar los requisitos del usuario en un sistema de software, y es además un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.

Está basado en componentes, lo que quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces. Este además utiliza el Lenguaje Unificado de Modelado (UML).

El ciclo de vida de RUP se caracteriza por:

- **Dirigido por casos de uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo, ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso. El usuario no solo representa a una persona sino que comprende además a otros sistemas. (S. 2005). Los casos de uso constituyen el hilo conductor que orienta las actividades en su desarrollo.

- **Centrado en la arquitectura:** El concepto de arquitectura incluye los aspectos estáticos y dinámicos más significativos del sistema. La arquitectura surge de las necesidades de la empresa como la perciben los usuarios y los inversores, y se refleja en los casos de uso. Esta además, se ve influida por otros factores como la plataforma en que debe funcionar el software, arquitectura hardware, sistema operativo, etc. En resumen, es una vista del diseño completo con las características más importantes resaltadas, dejando los detalles de lado.

- **Iterativo e incremental:** RUP se desarrolla mediante iteraciones, comenzando por los casos de uso relevantes desde el punto de vista de la arquitectura. Propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros.

### 1.5.2 Extreme Programming (XP).

Es una de las metodologías de desarrollo de software más exitosas en la actualidad, utilizada para proyectos de corto plazo y pequeños equipos de trabajo. Consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. Se basa principalmente en pruebas unitarias, refabricación y programación en pares.

Entre sus principales características se destacan:

- La comunicación entre los usuarios y los desarrolladores.
- La simplicidad al desarrollar y codificar los módulos del sistema.
- La retroalimentación concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.

### 1.6 Lenguaje de Modelamiento Unificado (UML - Unified Modeling Language).

El desarrollo del Unified Modeling Lenguaje, (UML) comenzó en octubre de 1994, cuando Grady Booch y Jim Rumbaugh en la Rational Software Corporation empezaron a trabajar para unificar el Booch (Metodología de Grady Booch) y la OMT (ObjectModeling Techniques). Un proyecto versión 0.8 del Método Unificado (UML), como se llamó desde sus inicios, salió al público en octubre de 1995.

En esa fecha, Jacobson se anexó a la compañía y aportó su esfuerzo al nuevo modelo, uniendo el OOSE (Object Oriented Software Engineering) al UML. (Héctor 2005).

El Lenguaje de Modelamiento Unificado (UML - Unified Modeling Language) es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. UML entrega una forma de modelar cosas conceptuales como lo son procesos de negocio y funciones de sistema, además de cosas concretas como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables. (Patricio and Nancy 2005)

### **1.6.1 UML como solución.**

UML surge como respuesta al primer problema reseñado para contar con un lenguaje estándar y poder escribir planos de software. Muchos han creído ver el UML como solución para todos sus problemas, sin saber en muchos casos de lo que se trataba en realidad. UML es una notación, es decir, una serie de reglas y recomendaciones para representar modelos. Permite documentar y especificar los elementos creados mediante un lenguaje común describiendo modelos. La elección del lenguaje para modelar depende de muchos factores. Para muchos, UML es un lenguaje a escoger cuando se van a modelar sistemas de software. La necesidad de modelar aplicaciones Web viene dado por lo siguiente:

¿Cómo expresar la lógica de negocios que se ejecutará en los componentes Web específicos, separados del resto de la aplicación?

Las aplicaciones Web al igual que otros sistemas de software son representadas típicamente con un conjunto de modelos: el modelo de casos de uso, el modelo de implementación, el modelo de distribución, etc. Un modelo adicional usado exclusivamente para los sistemas Web es el mapa del sitio, el cual no es más que una abstracción de las páginas Web y las rutas de navegación a través del sistema. (Héctor 2005)

### **1.6.2 Fundamentación del lenguaje de modelado, la metodología y herramienta Case, a utilizar.**

La decisión de utilizar UML como notación para el desarrollo del software se debe a que se ha convertido en un estándar que tiene las siguientes características:

- Permite modelar sistemas utilizando técnicas orientadas a objetos (OO).
- Permite especificar todas las decisiones de análisis y diseño, construyéndose así modelos precisos, no ambiguos y completos.
- Puede conectarse con lenguajes de programación (Ingeniería directa e inversa).
- Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, etc.).
- Es un lenguaje muy expresivo que cubre todas las vistas necesarias para desarrollar y luego desplegar los sistemas.
- Existe un equilibrio entre expresividad y simplicidad, pues no es difícil de aprender ni de utilizar.
- UML es independiente del proceso, aunque para utilizarlo óptimamente se debería emplear en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental.
- La metodología seleccionada utiliza UML.

Se utilizará como metodología de desarrollo de software RUP por constituir un marco de trabajo genérico que puede ser adaptado a una gran diversidad de sistemas de software independientemente del tamaño del proyecto, el tipo de organización y los diferentes niveles de aptitud. Además, agrupa las actividades en grupos lógicos definiéndose 9 flujos principales de trabajo. Los 6 primeros son conocidos como flujos de ingeniería y los tres últimos, como de apoyo.

Permite de forma práctica dividir el trabajo en partes más pequeñas o miniproyectos. Cada miniproyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en los flujos de trabajo y los incrementos al crecimiento del producto. Cada iteración se realiza de forma planificada.

Plantea la culminación de cada fase con la obtención de hitos, los cuales para lograr alcanzarlos hay que construir determinados artefactos definidos dentro de los flujos de trabajo involucrados.

Al finalizar un ciclo se obtiene una nueva versión del sistema; cada versión es un producto terminado que incluye los requisitos, casos de uso, especificaciones no funcionales y casos de prueba, así como el modelo de la arquitectura y el modelo visual -artefactos modelados con el Lenguaje Unificado de



Modelado. Todos estos artefactos constituyen una representación del producto que es necesaria para que los desarrolladores puedan llevar a cabo su siguiente ciclo de vida.

Para modelar el desarrollo del proyecto se utilizará como herramienta Case el Visual Paradigm por las siguientes características:

- Es una herramienta que sirve para realizar modelado UML siguiendo el estándar UML 2.1.
- Presenta características gráficas muy cómodas que facilitan la realización de los diagramas de modelado.
- Permite realizar la ingeniería inversa para diferentes tecnologías como Java, .NET, XML e Hibernate.
- Posibilita la exportación de imágenes jpg, png y svg.
- Permite la generación de código del lenguaje seleccionado.

### **1.7 Diseño de aplicaciones Web.**

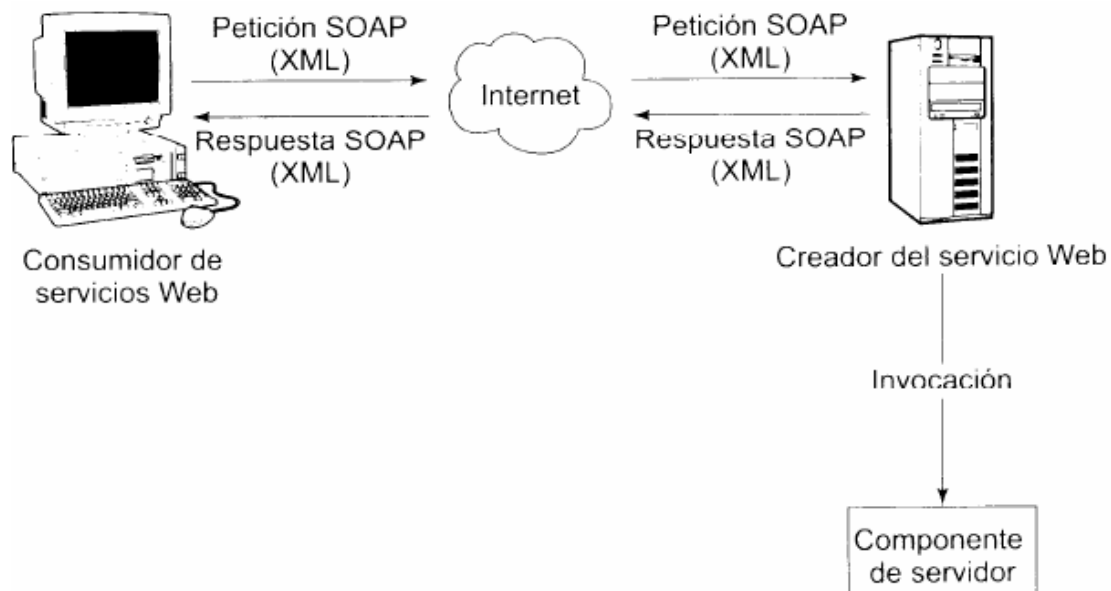
Un servicio Web es simplemente un componente de servidor que puede ser invocado en Internet. Este componente de servidor normalmente realiza un servicio fundamental de negocios como la autenticación del usuario, la validación de tarjetas de crédito, el cálculo del precio de un seguro de derivados, entre otros.

Los servicios Web permiten a las aplicaciones invocar servicios de negocios mediante un mecanismo basado en estándares (usando xml y http). En la invocación real de un servicio Web toman parte tres tecnologías: el protocolo de conexión, el formato de mensaje y el mecanismo de invocación.

El protocolo de conexión es el mecanismo de transporte que se emplea para establecer la comunicación entre el cliente y el servidor. Por lo general, suele ser http, protocolo de Internet basado en TCP/IP. El formato de mensaje, es el formato que se emplea para invocar un servicio Web. Un servicio Web puede ser invocado mediante http puro o con un mensaje xml en formato específico llamado Protocolo de Acceso simple a objetos (SOAP).

La tercera tecnología que controla cómo se llama a los componentes de servidor, no está especificada por los estándares del servicio Web. El programador que crea el servicio Web elige la tecnología usada para llamar al código de negocio en el servidor. (Enrique. 2005)

Las dos partes de un servicio Web pueden describirse como el creador (cliente) y el consumidor (servidor). El creador desarrolla el componente de servidor y muestra este servicio a quien corresponda. Mostrar un servicio Web significa publicar la URL que los usuarios necesitan para invocar al servicio Web. El consumidor puede usar el servicio expuesto enviando un mensaje de petición SOAP a la URL publicada. Al recibir una petición SOAP escrita en xml, el componente de servidor tras el servicio Web es invocado en el servidor creador. Los resultados de esta invocación toman el formato de un mensaje de respuesta SOAP y se envían de vuelta al consumidor.



**Figura 1.3. Representación del servicio Web.**

El sistema operativo del cliente o del servidor, el lenguaje de programación usado para escribir el componente de servidor o el mecanismo utilizado para invocar el componente del servidor no tienen importancia para un servicio Web. Un cliente escrito en C# que se ejecuta en Windows XP, por ejemplo, puede invocar un servicio Web escrito en java ejecutándose en Sun Solaris. De hecho un cliente no tiene forma de saber qué tecnologías se usan para mostrar el servicio Web.

Los servicios Web eliminan la necesidad de decidir el sistema operativo que se ejecuta en el servidor al integrar dos aplicaciones Web. Al usar servicios Web puede ensamblar aplicaciones usando componentes de otros servidores. Por ejemplo, puede usar un servicio Web de una compañía de

tarjetas para validar tarjetas de crédito; un servicio Web de una compañía distribuidora para determinar los gastos de envío y así sucesivamente. (Patterson, Brian et al. 2003)

## **1.8 Arquitectura de aplicaciones.**

### **1.8.1 Arquitectura de dos capas.**

La arquitectura tradicional de cliente/servidor también es conocida como arquitectura de dos capas. Requiere una interfaz de usuario que se instala y corre en una PC o estación de trabajo y que envía solicitudes a un servidor para ejecutar operaciones complejas. Por ejemplo, una estación de trabajo utilizada como cliente puede correr una aplicación de interfaz de usuario que interroga a un servidor central de bases de datos.

En la mayoría de los casos, el cliente requiere recursos y servicios que el servidor le proporciona. Ambos procesos pueden existir en una máquina o en máquinas separadas que se comunican a través de una red. La aplicación lógica es dividida entre las aplicaciones del cliente y la base de datos. (José 2005)

### **1.8.2 Arquitectura de tres capas.**

La arquitectura de tres capas se refiere a un diseño reciente que introduce una capa intermedia al proceso. Cada capa es un proceso separado y bien definido corriendo en plataformas separadas. La parte funcional de la arquitectura de tres capas generalmente es conocida como la capa intermedia o el servidor de aplicaciones. En éste ocurren la mayoría de los procesos. (Enrique. 2005)

### **1.8.3 Fundamentación de la arquitectura a utilizar.**

Después de efectuado un análisis comparativo a partir de las ventajas y desventajas de ambas arquitecturas, las autoras asumen la arquitectura de tres capas por las siguientes razones:

- Las llamadas de la interfaz del usuario en la estación de trabajo, al servidor de capa intermedia, son más flexibles que en el diseño de dos capas, ya que la estación sólo necesita transferir parámetros a la capa intermedia.

- Con la arquitectura de tres capas, la interfaz del cliente no es requerida para comprender o comunicarse con el receptor de los datos. Por lo tanto, esa estructura de los datos puede ser modificada sin cambiar la interfaz del usuario en la PC (Personal Computer).
- El código de la capa intermedia puede ser reutilizado por múltiples aplicaciones si está diseñado en formato modular. Esto puede reducir los esfuerzos de desarrollo y mantenimiento, así como los costos de migración.
- La separación de roles en tres capas, hace más fácil reemplazar o modificar una capa sin afectar a los módulos restantes.
- Separando la interfaz del usuario de la aplicación, libera de gran procesamiento a la estación de trabajo y permite que las actualizaciones de la aplicación sean centralizadas en el servidor de aplicaciones. (James 2005)

#### **1.8.4 Servidor de Aplicaciones.**

El servidor de aplicaciones fue introducido como parte del diseño de tres capas. Es relativamente nuevo y aún no está bien definido. Las empresas del mundo entero están esforzándose para producir su propia versión de lo que creen que es un servidor de aplicaciones. La definición más común de un servidor de aplicaciones es la de software corriendo en una capa intermedia entre un cliente pequeño y una base de datos. Generalmente se acepta que un servidor de aplicaciones maneja todas las transacciones lógicas y de conectividad que históricamente compartían el cliente y el servidor en un diseño cliente/servidor. (James 2005)

La propuesta se desarrollará utilizando el servidor HTTP Apache, por las siguientes características:

- Constituye parte de la amplia gama de software libre, proporcionado código abierto para plataformas Unix (BSD, GNU/Linux, etc), Windows entre otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.
- Constituye el complemento perfecto para la creación de aplicaciones Web dinámicas con PHP y MySQL.
- Presenta una gran sencillez en el manejo. (Luke and Laura 2003)
- A pesar de haber sido criticado por la falta de una interfaz gráfica que ayude en su configuración brinda la posibilidad de obtener mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido.

- Funciona sobre muchas plataformas (diferentes versiones de Unix, Linux, Vms, Win32, OS2).

### 1.9 Diseño de Interfaz.

El diseño de la interfaz necesita una herramienta para editar páginas Web y otra para el tratamiento de imágenes. Actualmente existen en el mundo diversos tipos de ellas. Para la realización de la propuesta las autoras tomaron como base la utilización de las siguientes teniendo en cuenta las características planteadas a continuación.

- Dreamweaver MX 2004: por ser uno de los editores visuales profesionales más importantes para la creación y administración de aplicaciones Web que combina facilidad y potencialidad en un entorno de desarrollo integrado para los sitios Web ColdFusion, HTML, XHTML, ASP, ASP.NET, JSP y PHP.

Con esta herramienta se puede desarrollar cualquier sitio Web personal con las mismas características de un sitio profesional y utilizar casi todos los recursos de la Web, así como realizar aplicaciones que se ejecuten en un servidor y vinculaciones dinámicas de datos; además de contar con un soporte para aplicaciones PHP y utilización de bases MySQL.

También cuenta con un amplio soporte para la creación y utilización de CSS (Cascading Style Sheets), propiciando un diseño fácil y óptimo. Tiene además un espacio de trabajo integrado, compartido con Macromedia Flash MX, Fireworks MX y Free Hand, que incluye ventanas de documentos con fichas, grupos de paneles acoplables, barras de herramientas personalizables y exploración integrada de archivos.

Se descubren en este producto, los beneficios de los estándares emergentes y las nuevas tecnologías Web con el soporte para XML, servicios Web y el amplio cumplimiento de accesibilidad para rehacer sitios ya existentes y crear aplicaciones de nueva generación.

- El Adobe Photoshop: por ser una de las principales herramientas para el tratamiento de imágenes, posee características distintivas que aportan nuevas soluciones a este proceso. Las continuas mejoras han hecho de este programa uno de los más profesionales para la edición y retoque fotográfico. Tiene un enfoque dirigido hacia los gráficos para la Web, y posee una total integración con su avanzada herramienta de producción Web: Adobe ImageReady.

### 1.10 Sistemas Gestores de Base de Datos (SGBD)

Una Base de Datos (BD) es un conjunto de datos interrelacionados, almacenados con carácter más o menos permanente en la computadora, puede ser considerada una colección de datos variables en el tiempo. (Date 2003)

Un Sistema Gestor de Base de Datos (SGBD) 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. Su objetivo fundamental consiste en suministrar al usuario las herramientas que le permitan manipular, en términos abstractos, los datos, o sea, de forma que no le sea necesario conocer el modo de almacenamiento de los mismos en la computadora, ni el método de acceso empleado. Entre sus principales funciones se encuentran:

- Independencia de los datos y los programas de aplicación.
- Minimización de la redundancia.
- Integración y sincronización de las bases de datos.
- Integridad de los datos.
- Seguridad y protección de los datos.
- Facilidad de manipulación de la información.
- Control centralizado.

La información es representada a través de tuplas, las cuales describen al fenómeno, proceso o ente de la realidad objetiva que se está analizando y se representan a través de tablas. (Date 2003)

#### 1.10.1 SQL (Structure Query Language).

SQL es un lenguaje de consulta estructurado. Entre sus características más importantes resaltan:

- Es una forma estándar de consulta de datos específicos.
- Permite extraer y manipular datos de una base de datos.
- Usado para todas las funciones de bases de datos, incluyendo administración.
- Creación de esquemas y datos recuperables.
- Puede ser usado de forma implícita dentro de una aplicación.

Existen SGBD que utilizan el SQL para realizar el tratamiento de los datos almacenado como son MySQL y SQL Server. (Héctor 2005)

### **1.10.2 MySQL.**

MySQL es un Sistema de Gestión de Base de Datos de código abierto, basado en el modelo relacional. Es muy rápido, fiable y fácil de usar. Fue desarrollado inicialmente para manejar grandes bases de datos mucho más rápido que las soluciones existentes y ha sido usado exitosamente por muchos años en ambientes de producción de alta demanda. A través de constante desarrollo, este SGBD ofrece hoy una rica variedad de funciones. Su conectividad, velocidad y seguridad lo hacen altamente satisfactorio para el acceso a bases de datos en Internet. Una de sus características más importantes es que consume muy pocos recursos, tanto de CPU como de memoria.

La ventaja más grande de MySQL es sin duda alguna su sencillez. Permite rápidamente crear bases de datos con muchas de las características y ventajas del lenguaje SQL. Unido a esto se encuentra el soporte que existe para controladores nativos en muchos lenguajes de programación. Esto último crea la posibilidad de conectar datos con interfaces Web, aplicaciones específicas o sistemas de información de forma centralizada y veloz.

### **1.10.3 SQL Server.**

Microsoft SQL Server, propietario de Microsoft, pertenece a la familia de los sistemas de administración de base de datos, operando en una arquitectura cliente/servidor de gran rendimiento. Su desarrollo fue orientado para hacer posible el manejo de grandes volúmenes de información, y un elevado número de transacciones. SQL Server es una aplicación completa que realiza toda la gestión relacionada con los datos. El servidor sólo tiene que enviarle una cadena de caracteres (la sentencia SQL) y esperar a que le devuelvan los datos.

SQL Server permite la creación de procedimientos almacenados, los cuales consisten en instrucciones SQL que se almacenan dentro de una base de datos de SQL Server, realizados en lenguaje SQL; se trata de procedimientos que se guardan semicompilados en el servidor y que pueden ser invocados desde el cliente. Se ejecutan más rápido que instrucciones SQL independientes. (Date 2003)

SQL Server puede manejar perfectamente bases de datos de TeraBytes con millones de registros y funciona sin problemas con miles de conexiones simultáneas a los datos, sólo depende de la potencia del hardware del equipo en el que esté instalado y solamente corre sobre Windows NT- 2000 Server.

#### 1.10.4 Fundamentación del sistema de Gestión de Base de Datos a utilizar.

Tanto el SQL Server como el MySQL operan en una arquitectura cliente/servidor, de tal manera que este sólo tiene que enviarle una cadena de caracteres (la sentencia SQL) y esperar la devolución de los datos.

Luego de analizadas las características y facilidades de los SGBD presentados, y de la herramienta a desarrollar, las autoras deciden usar el MySQL como SGBD, por las siguientes razones:

- No se necesitará de un manejo complejo de la información.
- El PHP maneja más fácil al MySQL que al SQL Server, debido a la gran cantidad de funciones que tiene explícitas.
- El MySQL es multiplataforma.
- El MySQL no tiene precio en el mercado, se adquiere libremente.

#### Conclusiones.

En este capítulo se abordó un estado del arte referente a la calidad de software, así como diferentes conceptos relacionados con el seguimiento de los recursos, se analizaron las tendencias y tecnologías actuales, ofreciendo las definiciones necesarias para comprender el porqué de las tecnologías que serán utilizadas en el desarrollo de la aplicación. Se fundamentó la metodología a utilizar, UML. Se hizo un estudio de las aplicaciones Web y el modelo cliente servidor, del lenguaje de programación Web y el gestor de base de datos, decidiéndose el uso de PHP como lenguaje de programación y MySQL como gestor de base de datos por las potencialidades que ofrecen, así como por formar parte del grupo de software de código abierto, solución por la que abogan, La Universidad de las Ciencias Informáticas en particular, y el país en general, debido al bajo costo de mantenimiento. Además se habló de las herramientas de desarrollo a utilizar para la concepción del sistema, explicando su gran potencialidad y facilidades que brinda a la hora de desarrollar la aplicación.





## Capítulo 2. Descripción del sistema propuesto.



### Introducción.

En el presente capítulo se explica cómo es el proceso de seguimiento de los diferentes recursos en el Laboratorio Central de Calidad, se identifican las necesidades de los usuarios y se describe el objeto a automatizar, también se realiza un estudio comparativo con otros sistemas que tienen los mismos objetivos que el que se propone y se fundamenta el por qué ninguno de ellos puede ser utilizado para resolver la situación problemática.

Se presenta también la propuesta del sistema y se especifican los requerimientos funcionales y no funcionales. Se realiza la definición de los casos de uso así como los actores que intervienen en ellos

### 2.1 Descripción de la situación problemática.

El Laboratorio Central de Calidad de Software de la UCI tiene como principal objetivo la realización de las diferentes pruebas a los productos terminados posibilitándole obtener un grado de certificación de acuerdo con las normas internacionales. Este servicio es solicitado por las diferentes facultades cuando cumplen con las condiciones necesarias para comenzar el proceso.

Para poder brindar un servicio eficiente y con calidad es muy importante mantener la información actualizada del Laboratorio y poder así atender las solicitudes de certificación en el menor tiempo posible. Este laboratorio tiene poco tiempo de creado por lo que la mayoría de los procesos involucrados se realizan de forma manual.

Cuando una facultad solicita el servicio de certificación, el tiempo de respuesta en la mayoría de los casos, no satisface a las necesidades de los mismos. El servicio debe brindarse con inmediatez porque el cliente lo necesita a tiempo para poder cumplir con su cronograma de actividades y evitar que se atrase la entrega del producto.

Para la realización de las diferentes pruebas es necesario contar con un conjunto de recursos (humanos, tecnológicos y tiempo) a los cuales se le debe realizar un proceso de planificación y seguimiento garantizando así el correcto empleo y distribución de ellos.

Actualmente no se cuenta con un control sistemático del trabajo realizado por cada una de las personas involucradas, lo que dificulta obtener un conjunto de informaciones así como optimizar el empleo del tiempo utilizado para la realización de las diferentes pruebas, además no existe un mecanismo que permita realizar un conjunto de evaluaciones de acuerdo con su desempeño profesional.

En el Laboratorio no se cuenta con un mecanismo que permita realizar la asignación de recursos de forma eficiente lo que trae como consecuencia que la entrega del producto se atrase por dificultades de tiempo y distribución de medios básicos.

Cuando los directivos de la Universidad solicitan conocer el estado de las certificaciones se hace muy engorroso posibilitarles la información con niveles de detalles en un tiempo relativamente corto debido a que no existe ningún mecanismo que permita agrupar la documentación de los proyectos y clasificarla de forma rápida de acuerdo con las solicitudes.

### **2.1.1 Objetivos estratégicos de la organización y procesos del negocio que la soportan.**

La UCI constituye un centro de estudios de nuevo tipo surgido en septiembre del 2002 como uno de los mayores proyectos de la Revolución. Entre sus principales objetivos se encuentran la formación académica de miles de jóvenes de diferentes partes del país y contribuir al desarrollo de la industria del software con la idea de convertir a la informática en una poderosa fuerza científica, económica y política para Cuba.

Para lograr estos objetivos es preciso desarrollar con calidad los productos informáticos y que a su vez sean avalados por las normas internacionales. Para esto es necesario contar con una entidad preparada tecnológicamente y con personal capacitado que sea el encargado de probar esta calidad y emitir un grado de certificación que los avale para poderse insertar en el mercado internacional.

Actualmente la entidad encargada de esto es el Laboratorio Central de Calidad de Software por lo que es de suma importancia desarrollar todos los procesos involucrados de forma eficiente para atender con mayor rapidez las demandas de las facultades. Entre sus principales procesos se destacan el proceso de aseguramiento de la calidad, las pruebas de aceptación, liberación, revisión y auditoría.

Dentro de estos procesos, la planificación y el seguimiento de los recursos utilizados juegan un papel importante pues son los encargados de mantener actualizada toda la información manejada en ese laboratorio.

### 2.1.2 Flujo actual de los procesos involucrados en el campo de acción.

Los procesos que intervienen en el **seguimiento de los recursos** empleados para la realización de las diferentes pruebas en el Laboratorio Central de Calidad de Software de la UCI se desarrollan de la siguiente manera:

- Recursos humanos.

Cuando un estudiante es seleccionado para pertenecer al grupo de calidad se convierte en probador y el asesor debe tener conocimiento del trabajo realizado por él; para ello se utiliza la planilla “Cuaderno de trabajo” propuesta por Watts Humphrey en su libro “Introducción al Proceso Software Personal (PSP)” (Ver Anexo I). En la medida que el estudiante realiza las diferentes pruebas, el jefe de equipo llena los datos de acuerdo con los parámetros de la planilla de forma manual, obteniéndose al final del proceso un histórico del trabajo realizado por cada probador, con esta información el jefe de equipo elabora un resumen del desempeño laboral de los probadores y mantiene actualizado al asesor.

- Recursos tecnológicos.

Para el control de los recursos tecnológicos es utilizada la planilla “Control de PC” (Ver Anexo II). Esta planilla fue definida por los directivos del laboratorio con el objetivo de conocer el estado real de cada PC, así como su tiempo de uso diario. Este proceso se realiza de forma manual, cada persona es la encargada de llenar sus datos cuando se incorpora al trabajo.

- Tiempo.

Actualmente no existe ningún mecanismo definido específicamente para controlar el tiempo utilizado en la realización de las diferentes pruebas, solo es posible conocer el tiempo que es utilizada una PC guiándose por los resultados obtenidos a partir de la planilla utilizada para el control de los recursos tecnológicos (Anexo II).

### 2.1.3 Análisis crítico de la ejecución actual de los procesos.

Los procesos descritos en el epígrafe anterior actualmente se realizan de forma manual, todo esto trae como consecuencia que las revisiones de los productos se atrasen por no contar con la información actualizada y detallada.

Los mecanismos utilizados para el control sistemático de los recursos humanos y los medios tecnológicos no son eficientes, pues no brindan toda la información necesaria para garantizar un buen seguimiento de ellos; además, las planillas utilizadas para estos procesos no son confiables pues al

estar al alcance de todos pueden ser alteradas y no mostrar los datos verdaderos. No existe ningún mecanismo que permita conocer el tiempo real que un probador utiliza para la realización de las diferentes pruebas

#### **2.1.4 Procesos objeto de automatización.**

Serán objeto de automatización todos los procesos que intervienen en el seguimiento de los recursos (humanos, tecnológicos y tiempo.) empleados en la revisión de la calidad de los productos software desarrollados en la Universidad de las Ciencias Informáticas.

#### **2.1.5 Sistemas automatizados existentes vinculados al campo de acción.**

Existen diferentes herramientas que permiten realizar el seguimiento de diversos proyectos y los recursos utilizados en ellos. Cada una cuenta con características específicas que las hacen distintivas. Entre las más conocidas actualmente se encuentran:

- **Software de seguimiento y control de tareas.**

Es un software de oficina diseñado para mantener el control y el estado de las tareas y documentos que circulan por una oficina desde que se originan hasta que se cierran. Además, permite mantener el control de los asuntos y documentos que llegan así como los que son delegados a un determinado tipo de usuario, manteniendo informado al personal de las fechas de inicio, vencimiento y terminación que estos guardan.

Entre sus características más importantes se encuentran:

- ✓ Cuenta con un módulo de seguridad para restringir accesos al sistema.
- ✓ Maneja un número ilimitado de tareas.
- ✓ Permite la clasificación de los asuntos mostrados por cualquier columna de la ventana.
- ✓ Se pueden incluir documentos anexos a cualquier tarea.
- ✓ Cuenta con una bitácora de movimientos que permite revisar el estado de todas las tareas registradas.
- ✓ Está diseñado para usarse en Internet, su interfaz es intuitiva y fácil de usar. (Software 2006)

Por ser una herramienta de oficina que solamente permite realizar el control y seguimiento de los documentos y las tareas realizadas en un determinado periodo de tiempo, además de ser un software

propietario, las autoras consideran que no puede ser empleado en el seguimiento de los recursos utilizados en el proceso de revisión de la calidad a los productos desarrollados en la UCI.

- **Generación, control y seguimiento de recursos propios. (GCSRP)**

El GCSRP es una plataforma colaborativa que permite acceder a las distintas fuentes de información relacionada con la generación de ingresos propios, buscando la integración de las actividades y procesos administrativos a los sustantivos de la entidad, con una visión de calidad y servicio.

Este software permite ofrecer el conocimiento de los ingresos en una empresa desde sus mismas fuentes generadoras, además posibilita la comparación en distintos períodos de tiempo, establecer estrategias y dar seguimiento a las mismas, a través de un monitoreo constante.

Entre sus características más importantes se destaca que está desarrollado en software libre lo que implica una reducción considerable de costos, una alta confidencialidad y gran capacidad de adaptación. (Magdalena and Jorge 2006)

Por todas las características antes expuestas esta herramienta puede considerarse como una aplicación robusta pero al igual que la anteriormente analizada solo puede ser utilizada en determinadas áreas de la economía, no puede ser adaptada con las características necesarias para llevar el control del proceso de seguimiento de los diferentes recursos empleados en el laboratorio de calidad de software.

- **Sistema de seguimiento de recursos SMART TAG.**

El sistema Varis Smart Tag es un software sofisticado y de alta tecnología que resulta fácil de instalar y cuenta con diferentes funcionalidades como:

- ✓ Una interfaz gráfica para el usuario, clara y de fácil manipulación.
- ✓ Ubicación de recursos en tiempo real sobre una mina.
- ✓ Especificación de tipos de recursos.
- ✓ Tres niveles de seguridad, predeterminados para distintos tipos de usuarios.
- ✓ Generación automática de reportes.

Es un sistema de identificación por frecuencia de radio de rango largo que se conecta a la red Ethernet de una mina para seguimiento en tiempo real de recursos, como vehículos y minerales.

Esta herramienta cuenta con varias características técnicas que la ubican entre los mejores software de seguimiento de recursos en su área. Está implementada con tecnología de punta permitiendo obtener información detallada sobre la ubicación del equipo y personal que se encuentra en un determinado lugar. (Varis 2006)

Este sistema está diseñado para resolver un determinado problema en un área específica y forma parte de la gran gama de software propietarios existentes en el mundo, por estas características las autoras consideran que no puede ser utilizado en el laboratorio de calidad de software.

- **Microsoft Office Project.**

La familia de productos Microsoft Office Project incluye Project Standard, Project Professional, Project Server y Project Web Access. Como parte integral de Microsoft Office System. El Project se ha diseñado para ser flexible al cubrir las necesidades de administración de trabajo y usuarios, tanto si se trata de administrar proyectos de forma independiente o en un equipo, departamento u organización. (Enterprise 2006)

Entre sus principales características de encuentran:

- ✓ Gestiona y notifica una cantera de proyectos.
- ✓ Establece los estándares y las prácticas correctas para cada proyecto.
- ✓ Centraliza la administración de los recursos basándose en la destreza y disponibilidad de cada empleado.
- ✓ Permite una comunicación y colaboración eficaz entre los distintos equipos de trabajo.

Es una herramienta de gran importancia para los líderes de equipos de desarrollo, pues les permite programar y organizar sus recursos y tareas, a fin de generar proyectos a tiempo y conforme a su presupuesto. (Microsoft 2007). También es considerada como una de las más amplias y adaptables a cualquier tipo de proyecto, permitiendo una gestión excelente en todas las etapas de su desarrollo.

Mediante ella es posible realizar el seguimiento a grandes rasgos, de los recursos empleados en un determinado proyecto, pero no permite llevar un control estricto de recursos específicos, como por ejemplo:

- ✓ No es posible realizar un control automático del tiempo utilizado para desarrollar actividades específicas.
- ✓ No permite generar reportes automáticos en dependencia de las necesidades del cliente.

- ✓ No permite realizar búsquedas avanzadas de un determinado recurso estableciendo comparaciones entre diferentes períodos de trabajo.
- ✓ No permite llevar un control sistemático de determinadas evaluaciones realizadas a diferentes tipos de personas, así como establecer comparaciones entre ellas y generar una estadística automática donde pueda ser posible observar el desenvolvimiento de alguien en específico en una determinada actividad.

Por todas estas situaciones antes planteadas a pesar de ser el Project una aplicación robusta con una gran cantidad de opciones y facilidades para la administración de proyectos, las autoras consideran que no puede ser utilizado para llevar el control sistemático y específico de los recursos empleados en el proceso de revisión de calidad de software realizado en el Laboratorio Central de Calidad de la Universidad de las Ciencias Informáticas.

## **2.2 Descripción del negocio.**

El modelo de negocio que se va a describir, responde a un conjunto de actividades realizadas dentro de los procesos que se desarrollan en el Laboratorio Central de Calidad de Software de la UCI. Inicialmente el solicitante (vicedecanos de producción, Jefes de pruebas, desarrolladores) realiza la solicitud de revisión de sus productos a la dirección de calidad de la UCI. Posteriormente se procede a la revisión del software que se divide en tres procesos fundamentales: la revisión de documentación, planificación de las pruebas y ejecución de las mismas.

En el primer proceso se verifica que la documentación del proyecto cumpla con los lineamientos básicos que exige la dirección de calidad. De no cumplir se le informa al solicitante la negación de la revisión del software; en caso contrario se comienza con la planificación y estimación de las pruebas y luego se realiza la ejecución de las mismas incluyendo la generación de reportes de las no conformidades obtenidas. Paralelamente a esto se efectúa la administración de los recursos.

Los directivos de la universidad (rector, vicerrectores, directivos de calidad, decanos, vicedecanos) pueden solicitar reportes en dependencia de sus intereses.

## **2.3 Reglas del negocio.**

Las reglas de negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio (Pressman 1998).

En el sistema que se propone se deben considerar las siguientes reglas:

1. Las únicas personas autorizadas a solicitar la revisión de los productos informáticos son los directivos de las facultades (decano, vicedecano, jefe de proyecto) y los directivos de la universidad (rector, vicerrectores).
2. Para poder realizarse la solicitud de revisión de un software es necesario contar con una versión del producto, incluyendo su documentación.
3. Para poder realizar la revisión del software es necesario efectuar la revisión de la documentación como primer proceso. Los procesos de planificar y ejecutar pruebas no necesariamente se realizan en un orden lógico. El proceso de administrar recursos se realiza paralelo a los dos últimos.
4. Los valores de las métricas utilizadas para la planificación de las pruebas son redefinidos en dependencia de los resultados obtenidos a partir del seguimiento sistemático a los probadores en la realización de las diferentes pruebas.
5. La documentación necesaria para la realización de las diferentes pruebas será colocada en una estructura de directorios previamente definida, a la que tendrán acceso todos los involucrados en el proceso, por lo que cualquier integrante del grupo de calidad podrá acceder a los recursos que necesite para llevar a cabo su trabajo. De esta forma se facilita la coordinación y la comunicación entre los diferentes equipos de trabajo, a la vez que se agiliza el proceso.
6. Para poder realizar un seguimiento a los recursos tecnológicos tienen que haber sido planificados.

#### **2.4 Actores del negocio.**

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados (Pressman 1998).



Actores del negocio	Justificación
Solicitante	Asesor de calidad de la facultad o vicedecano de producción que solicite la revisión de sus productos.
Directivos	Personas en la UCI como rector, vicerrectores, directivos de calidad, decanos, vicedecanos que estén interesados en realizar solicitudes de reportes.

Tabla 2.1. Descripción de los actores del negocio

### 2.5 Diagrama de casos de uso del negocio.

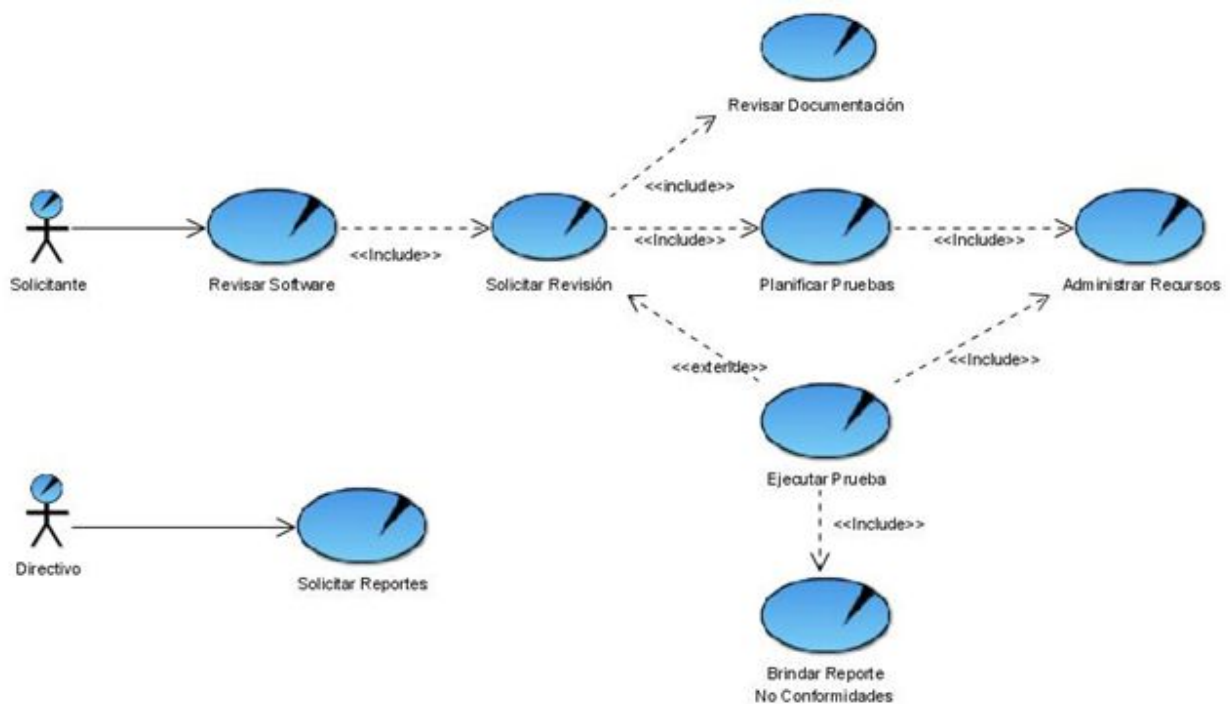


Figura 2.1. Diagrama de casos de uso del negocio.

### 2.6 Trabajadores del negocio.

Los trabajadores del negocio representan personas o sistemas (Software) dentro del negocio que son los encargados de realizar las diferentes actividades comprendidas dentro de un caso de uso. Estos trabajadores están dentro de la frontera del negocio y son los que en un futuro se convertirán en usuarios del sistema que se quiere construir (Pressman 1998).

Trabajadores del negocio	Justificación
Jefe de prueba	Encargado en cada facultad de planificar y controlar los recursos en el Laboratorio de calidad, gestiona las no conformidades y realiza los reportes de seguimiento de los recursos.
Especialista de calidad	Encargado de insertar las pruebas a realizar y las métricas.
Probador	Persona que realiza las pruebas de software.
Desarrollador	Ve las no conformidades emitidas y las responde.

**Tabla 2.2. Descripción de los trabajadores del negocio.**

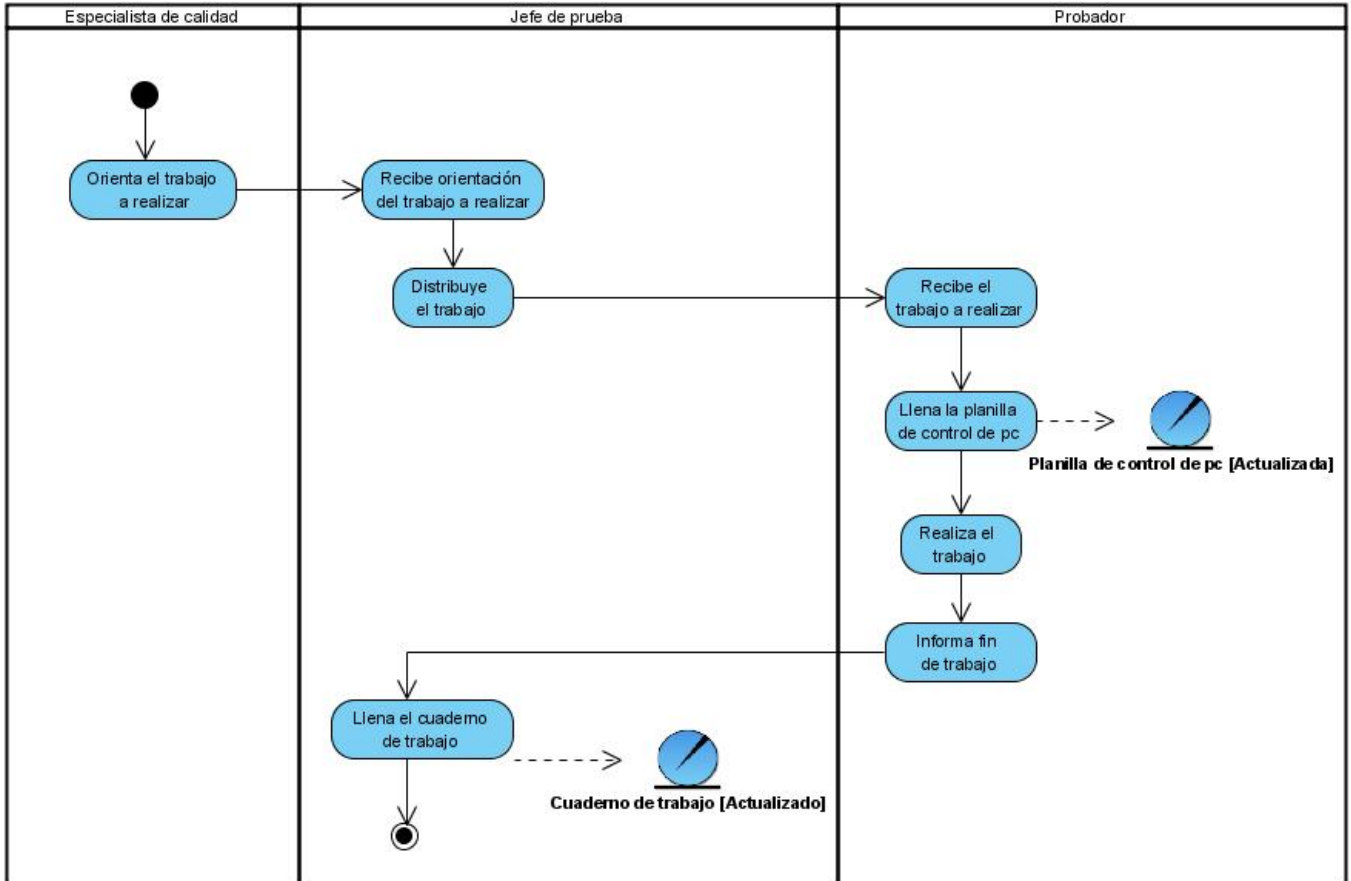
### 2.7 Casos de uso del negocio.

Un caso de uso del negocio representa los procesos del negocio, por lo que corresponde con una secuencia de acciones con un orden lógico que produce un resultado observable para los actores del mismo (Pressman 1998).

Este negocio se encuentra dividido en tres módulos fundamentales. A continuación se muestra la descripción de los casos de uso relacionados con el seguimiento de recursos

#### 2.7.1 Caso de Uso Administrar Recursos.

En este caso de uso se realizará un conjunto de actividades necesarias para controlar el rendimiento y utilización reales de los recursos planificados para la revisión de software, permitiendo obtener datos concretos de los proyectos realizados así como de los recursos humanos y tecnológicos involucrados en cada una de las pruebas de software realizadas. A continuación se muestra el diagrama de actividades asociado a este caso de uso.

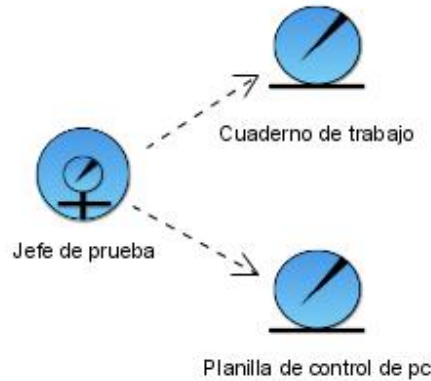


**Figura 2.2. Diagrama de actividades del caso de uso Administrar recursos.**

### 2.8 Definición de las entidades del negocio.

Las entidades del negocio representan un contenedor de información, algo físico que se utilice en el proceso del negocio y que sirva para obtener o actualizar información. En este negocio las entidades son las siguientes:

- Cuaderno de trabajo: Para conocer las funcionalidades de este documento consulte (Anexo I).
- Planilla de control de pc: Para conocer las funcionalidades de este documento consulte (Anexo II).



**Figura 2.3. Diagrama de clases del modelo de objetos.**

### 2.9 Propuesta del sistema.

La herramienta que se propone desarrollar automatiza los procesos que se llevan a cabo en el Laboratorio Central de Calidad de Software de la UCI. Para una mayor eficiencia en la implementación se organizó en tres módulos fundamentales.

El módulo de Planificación es el encargado de planificar y estimar los recursos. Para realizar esto, previamente deben introducirse en el sistema, los tipos de pruebas y las métricas asociadas a cada uno. Una vez adicionadas las planificaciones, pueden consultarse mediante reportes solicitados por el personal autorizado. Estos reportes dan la posibilidad de exportarse e imprimirse y al igual que las pruebas y las métricas pueden ser gestionados.

En el módulo Gestión de las No Conformidades se gestionan todas las no conformidades encontradas durante el proceso de revisión. Para realizar esto, previamente debe tenerse toda la documentación necesaria en dependencia del tipo de prueba que se realizará y las condiciones requeridas para el proceso de prueba. Las no conformidades serán insertadas en el sistema y estas se podrán modificar, eliminar y se le darán respuesta por parte del equipo de desarrollo. Este módulo permite además, asignar el trabajo a los probadores y visualizar el estado de los elementos de prueba y el de los proyectos. A partir de toda esta información el personal autorizado podrá obtener reportes teniendo la posibilidad de exportarlos e imprimirlos.

El módulo de Seguimiento es el encargado de administrar todos los recursos utilizados para la realización de las diferentes pruebas, permitiendo conocer el estado real de estos y a la vez brindar un

conjunto de reportes que posibilitan mantener actualizados a los directivos de la Universidad. Como resultado de este proceso se obtiene un histórico con toda la información relacionada con los diferentes tipos de recursos.

Esta herramienta se implementará sobre la tecnología PHP, aprovechando las amplias ventajas que el mismo ofrece y que fueron descritas en el capítulo 1. Además se utilizará como gestor de base de datos el MySQL por ser compatible con el lenguaje seleccionado y brindar diferentes funcionalidades descritas también en el capítulo 1.

El sistema constará de políticas de seguridad, otorgando a cada usuario los derechos que le corresponden. Existen 5 tipos de usuarios: administradores, especialistas de calidad, jefes de prueba, probadores y desarrolladores. Los usuarios deberán autenticarse antes de realizar alguna acción.

## **2.10 Especificación de los requerimientos del software.**

### **2.10.1 Requisitos funcionales**

El sistema que se propone construir para automatizar los procesos que se desarrollan en el Laboratorio Central de Calidad, debe permitir:

**RF1.** Gestionar datos de Jefe de prueba.

- RF 1.1 Insertar datos de un nuevo Jefe de prueba.
- RF 1.2 Buscar y visualizar datos de un Jefe de prueba.
- RF 1.3 Modificar datos de un Jefe de prueba.
- RF 1.4 Eliminar datos de Jefe de prueba.
- RF 1.5 Exportar datos de Jefe de prueba.
- RF 1.6 Imprimir datos de Jefe de prueba.

**RF2.** Gestionar datos de Probador.

- RF 2.1 Insertar datos de nuevo Probador.
- RF 2.2 Buscar y visualizar datos de Probador.
- RF 2.3 Modificar datos de Probador.
- RF 2.4 Eliminar datos de Probador.
- RF 2.5 Exportar datos de Probador.
- RF 2.6 Imprimir datos de Probador.

**RF3.** Evaluar Probador.



**RF4.** Gestionar pc.

RF 4.1 Insertar cantidad de pc que no se utilizan en una fecha determinada.

RF 4.2 Seleccionar causa de no utilización de pc.

RF 4.3 Adicionar nueva causa.

**RF5.** Gestionar reportes.

RF 5.1 Generar reporte

RF 5.2 Exportar reporte.

RF 5.3 Imprimir reporte.

**RF6.** Verificar cumplimiento de métrica.

RF 6.1 Buscar métrica por tipo de prueba.

RF 6.2 Buscar trabajo por tipo de prueba.

**RF7.** Gestionar cursos optativos.

RF 7.1 Adicionar Curso optativo.

RF 7.2 Asignar curso optativo.

**RF8.** Gestionar usuario.

RF 8.1 Crear un nuevo usuario.

RF 8.2 Editar datos de usuario.

RF 8.3 Buscar y visualizar datos de usuario.

RF 8.4 Asignar roles a usuarios.

RF 8.5 Restringir roles a usuarios.

RF 8.6 Eliminar usuario.

**RF9.** Autenticarse.

**2.10.2 Requisitos no funcionales.**

1. Apariencia o interfaz externa.

El sistema deberá poseer una interfaz Web sencilla, amigable, lo más atractiva y clara posible para el usuario, además su funcionamiento debe ser de fácil comprensión.

2. Usabilidad.

El sistema solo podrá ser utilizado por aquellas personas que de una u otra forma se encuentren relacionadas con el control y aseguramiento de la calidad en los proyectos productivos de la Universidad de las Ciencias Informáticas.

3. Rendimiento.

Debido a que se trata de una aplicación cliente/servidor debe ser eficiente, con capacidad adecuada de procesamiento y cálculo; además, como requiere de un tiempo de respuesta relativamente pequeño.

4. Portabilidad.

El sistema debe ser multiplataforma.

5. Seguridad.

Para poder acceder al sistema el usuario deberá estar registrado en la aplicación. Los usuarios serán creados en dependencia del rol que desempeñen. Se debe garantizar que solo posean acceso a la información con derecho a ver o manipular.

6. Software.

En el lado del Cliente debe existir un navegador que soporte Java Script.

En el lado del Servidor debe estar instalado: PHP 5.0 o superior y gestor de base de datos MySQL 4.1.8 o superior. Es necesario tener instalado el Adobe Reader para la impresión de los Reportes.

7. Hardware.

Para el desarrollo y ejecución de esta aplicación se necesita conexión a la red local, por lo que se requiere tarjeta de red. Además es necesario contar con una impresora para poder reproducir los diferentes tipos de reportes.

8. Restricciones en el diseño y la implementación.

Para la construcción del sistema deberá utilizarse PHP como lenguaje de programación y MySQL como gestor de base de datos.

### 2.11 Actores del sistema.

Los actores representan terceros fuera del sistema que interactúan con él, pueden representar el rol que juegan una o varias personas, un equipo o un sistema automatizado.

Actores	Justificación
Especialista de calidad	Gestiona los datos de Jefes de prueba, gestiona reportes, verifica las métricas.
Jefe de prueba	Gestiona los datos de Probadores, evalúa a estos, gestiona reportes, las pc sin utilizar y verifica las métricas.
Administrador	Es el encargado de administrar las cuentas de los usuarios y el sistema en general.
Usuario	Se autentifica al entrar en el sistema.
Gestor de reportes	Este actor es una generalización de Jefe de prueba y Especialista de calidad.

**Tabla 2.3. Actores del sistema.**



2.12 Modelo de casos de uso del sistema.

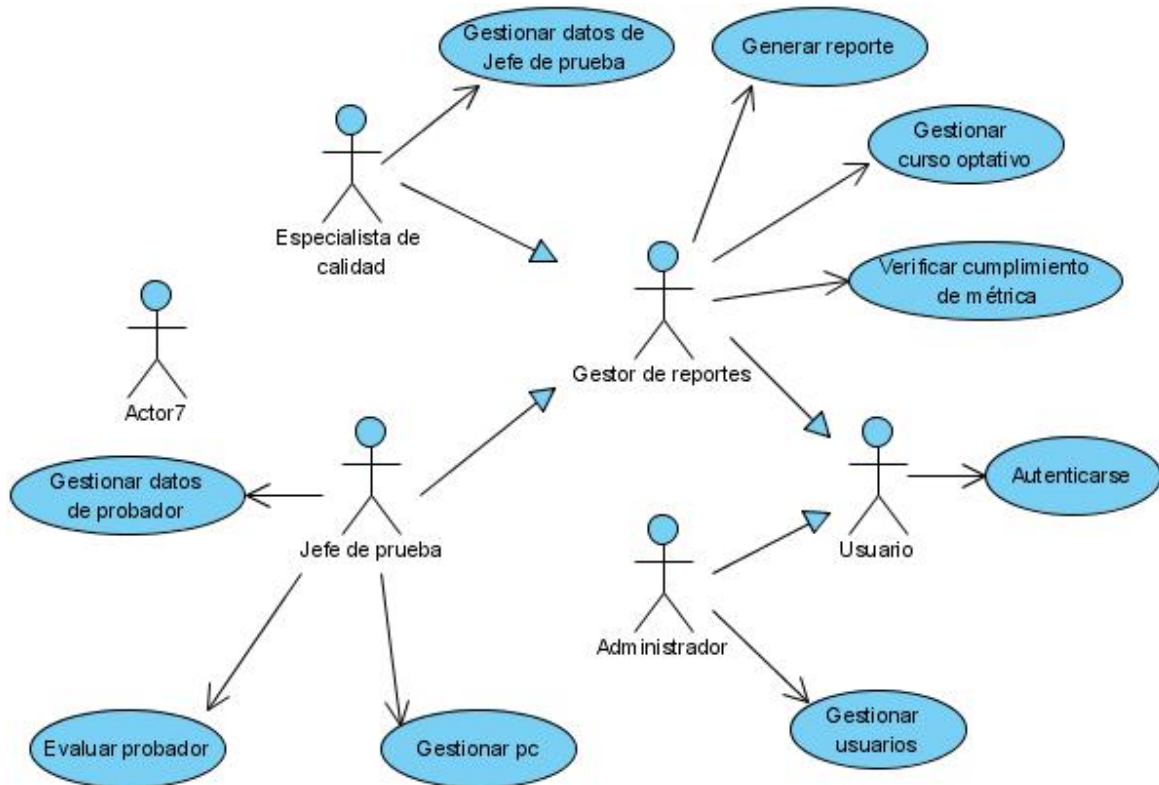


Figura 2.4. Diagrama de casos de uso del sistema.

2.13 Definición de los casos de uso del sistema.

<b>CU - 1</b>	Gestionar datos de Jefe de prueba
<b>Actor</b>	Especialista de calidad
<b>Descripción</b>	<p>Este caso de uso inicia cuando el especialista de calidad va a realizar alguna de las siguientes operaciones relacionadas con el Jefe de prueba:</p> <ul style="list-style-type: none"> <li>• Insertar datos de un nuevo Jefe de prueba.</li> <li>• Buscar y visualizar datos de un Jefe de prueba.</li> <li>• Modificar datos de un Jefe de prueba.</li> </ul>

	<ul style="list-style-type: none"> <li>• Eliminar datos de Jefe de prueba.</li> <li>• Exportar datos de Jefe de prueba.</li> <li>• Imprimir datos de Jefe de prueba.</li> </ul> <p>El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se emite el resultado de la operación solicitada.</p>
<b>Referencia</b>	RF 1

**Tabla 2.4. Definición del caso de uso “Gestionar datos de Jefe de prueba”.**

Para ver la descripción del caso de uso “Gestionar datos de Jefe de prueba” dirijase a: (Anexo III).

<b>CU - 2</b>	Gestionar datos de Probador
<b>Actor</b>	Jefe de prueba
<b>Descripción</b>	<p>Este caso de uso inicia cuando el Jefe de prueba va a realizar alguna de las siguientes operaciones relacionadas con el Probador:</p> <ul style="list-style-type: none"> <li>• Insertar datos de un nuevo Probador.</li> <li>• Buscar y visualizar datos de un Probador.</li> <li>• Modificar datos de un Probador.</li> <li>• Eliminar datos de Probador.</li> <li>• Exportar datos de Probador.</li> <li>• Imprimir datos de Probador.</li> </ul> <p>El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se emite el resultado de la operación solicitada.</p>
<b>Referencia</b>	RF 2

**Tabla 2.5. Definición del caso de uso “Gestionar datos de Probador”.**

Para ver la descripción del caso de uso “Gestionar datos de Probador” dirijase a: (Anexo IV).

<b>CU - 3</b>	Evaluar Probador
<b>Actor</b>	Jefe de prueba
<b>Descripción</b>	<p>Este caso de uso se inicia cuando el Jefe de prueba revisa el trabajo realizado por el Probador, según la calidad del trabajo desempeñado por el Probador el Jefe de prueba lo evalúa de:</p> <ul style="list-style-type: none"> <li>• 0</li> <li>• 1</li> <li>• 2</li> <li>• 3</li> <li>• 4</li> <li>• 5</li> </ul> <p>El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se emite el resultado de la operación solicitada.</p>
<b>Referencia</b>	RF 3

**Tabla 2.6. Definición del caso de uso “Evaluar Probador”.**

Para ver la descripción del caso de uso “Evaluar Probador” diríjase a: (Anexo V).

<b>CU - 4</b>	Gestionar PC
<b>Actor</b>	Jefe de prueba
<b>Descripción</b>	Brindar la posibilidad al Jefe de prueba insertar el número de pc que nos se utilizaron y la causa por la que cada pc no se utilizó.
<b>Referencia</b>	RF 4

**Tabla 2.7. Definición del caso de uso “Gestionar PC”.**

Para ver la descripción del caso de uso “Gestionar PC” diríjase a: (Anexo VI).

<b>CU - 5</b>	Generar Reporte
<b>Actor</b>	Gestor de reportes

<b>Descripción</b>	Este caso de usos comienza cuando el Gestor de reportes desea generar un reporte. A cada reporte que se genere se le da la posibilidad de exportarlo e imprimirlo.  El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se emite el resultado de la operación solicitada.
<b>Referencia</b>	RF 5

**Tabla 2.8. Definición del caso de uso “Generar Reporte”.**

Para ver la descripción del caso de uso “Generar Reporte” diríjase a: (Anexo VII).

<b>CU - 6</b>	Verificar cumplimiento de métrica
<b>Actor</b>	Gestor de reportes
<b>Descripción</b>	Este caso de uso se inicia cuando el Gestor de reporte desea ver el cumplimiento de las métricas planificadas.  El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se emite el resultado de la operación solicitada.
<b>Referencia</b>	RF 6

**Tabla 2.9. Definición del caso de uso “Verificar cumplimiento de métrica”.**

Para ver la descripción del caso de uso “Verificar cumplimiento de métrica” diríjase a: (Anexo VIII).

<b>CU - 7</b>	Gestionar cursos optativos
<b>Actor</b>	Gestor de reportes
<b>Descripción</b>	Este caso de usos comienza cuando el Gestor de reportes desea adicionar o asignar un curso optativo.  El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se emite el resultado de la operación solicitada.
<b>Referencia</b>	RF 7

**Tabla 2.10. Definición del caso de uso “Gestionar cursos optativos”.**

Para ver la descripción del caso de uso “Gestionar cursos optativos” diríjase a: (Anexo IX).

<b>CU - 8</b>	Gestionar Usuarios
<b>Actor</b>	Administrador
<b>Descripción</b>	<p>El caso de uso comienza cuando el Administrador va a realizar alguna de las siguientes operaciones:</p> <ul style="list-style-type: none"> <li>• Crear un nuevo usuario.</li> <li>• Editar datos de usuario.</li> <li>• Buscar y visualizar datos de usuario.</li> <li>• Asignar roles a usuarios.</li> <li>• Restringir roles a usuarios.</li> <li>• Eliminar usuario.</li> </ul> <p>El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se emite el resultado de la operación solicitada.</p>
<b>Referencia</b>	RF 8

**Tabla 2.11. Definición del caso de uso “Gestionar Usuarios”.**

Para ver la descripción del caso de uso “Gestionar Usuarios” diríjase a: (Anexo X).

<b>CU - 9</b>	Autenticarse
<b>Actor</b>	Usuario
<b>Descripción</b>	<p>El Caso de Uso se inicia cuando el usuario introduce los datos que se le piden para acceder a la aplicación, estos se verifican y finaliza dándole los permisos y habilitándole la entrada.</p>
<b>Referencia</b>	RF 8

**Tabla 2.12. Definición del caso de uso “Autenticarse”.**

Para ver la descripción del caso de uso “Autenticarse” diríjase a: (Anexo XI).

### **Conclusiones.**

El seguimiento de los recursos en el Laboratorio de Calidad de la UCI fue ampliamente abordado en el transcurso de este capítulo. El establecimiento de una adecuada correspondencia entre las insuficiencias declaradas, la descripción de los procesos del negocio actual y la propuesta de solución, permitió la realización del análisis y el diseño del software que elevará la calidad de los procesos de revisión de los productos desarrollados.



## Capítulo 3. Análisis y diseño del sistema.



### Introducción.

En este capítulo se muestran los resultados obtenidos en el desarrollo de los procesos de análisis y diseño del sistema, así como los diagramas que fueron necesarios utilizar para obtener una mayor claridad a la hora de elaborar la solución que se propone y para lograr una implementación acorde a las necesidades de los usuarios.

### 3.1 Análisis.

Durante el análisis se analizan los requisitos que se describieron en la captura de requisitos, refinándolos y estructurándolos. El objetivo de hacerlo es conseguir una comprensión más precisa de los requisitos y una descripción de los mismos que sea fácil de mantener y que ayude a estructurar el sistema entero, incluyendo su arquitectura. El modelo de análisis está escrito en el lenguaje del desarrollador, es la vista interna del sistema. El análisis se basa en un modelo de objetos conceptual, que es llamado modelo de análisis. (Pressman 1998)

### 3.2 Modelo conceptual de clases de análisis.

El modelo conceptual se realizó utilizando el diagrama de clases de la notación UML de forma simplificada. Se utilizan las clases y asociaciones preliminares entre estas, la multiplicidad o cardinalidad para cada asociación y los nombres para las clases y las asociaciones.

Una clase de análisis representa una abstracción de una o varias clases y/o subsistemas del diseño del sistema. Las clases de análisis se basan en tres estereotipos básicos: interfaz, entidad y control.

Clases de Interfaz: se utilizan para modelar la interacción entre el sistema y sus actores (usuarios y sistemas externos). Modelan las partes del sistema que dependen de sus actores. Representan

abstracciones de ventanas, formularios, paneles, interfaces de comunicación, interfaces de impresora, sensores y terminales.

Clases de Entidad: se utiliza para modelar información que posee una vida larga y que es a menudo persistente.

Clases controladoras: Representan coordinación, secuencia, transacciones, y control de otros objetos y se usa con frecuencia para encapsular el control de un caso de uso en concreto, modelan los aspectos dinámicos del sistema. (Pressman 1998)

A continuación se describen los diagramas de clases del análisis de los casos más representativos para la realización de la solución propuesta.

### 3.2.1 Diagrama de clases del análisis caso de uso “Gestionar datos de Jefe de prueba”.

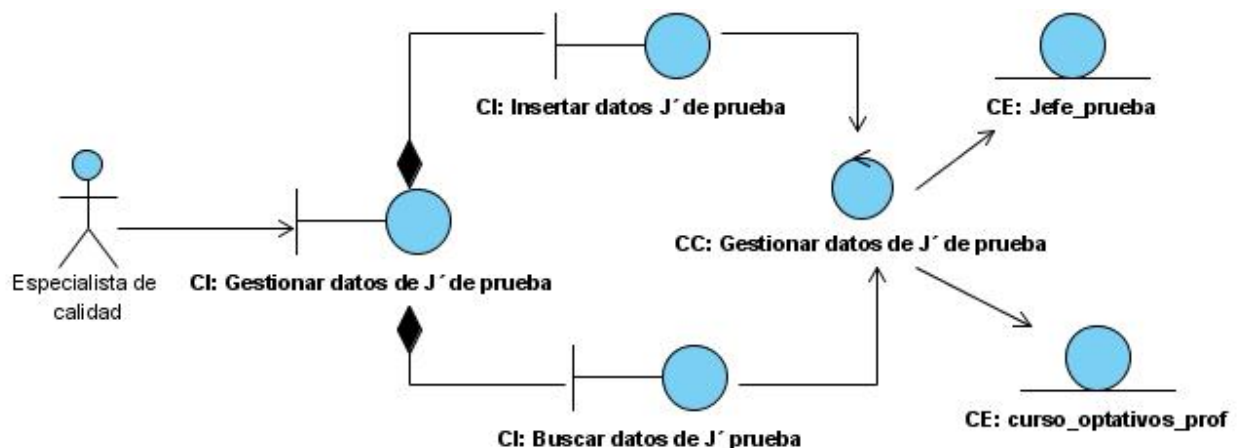
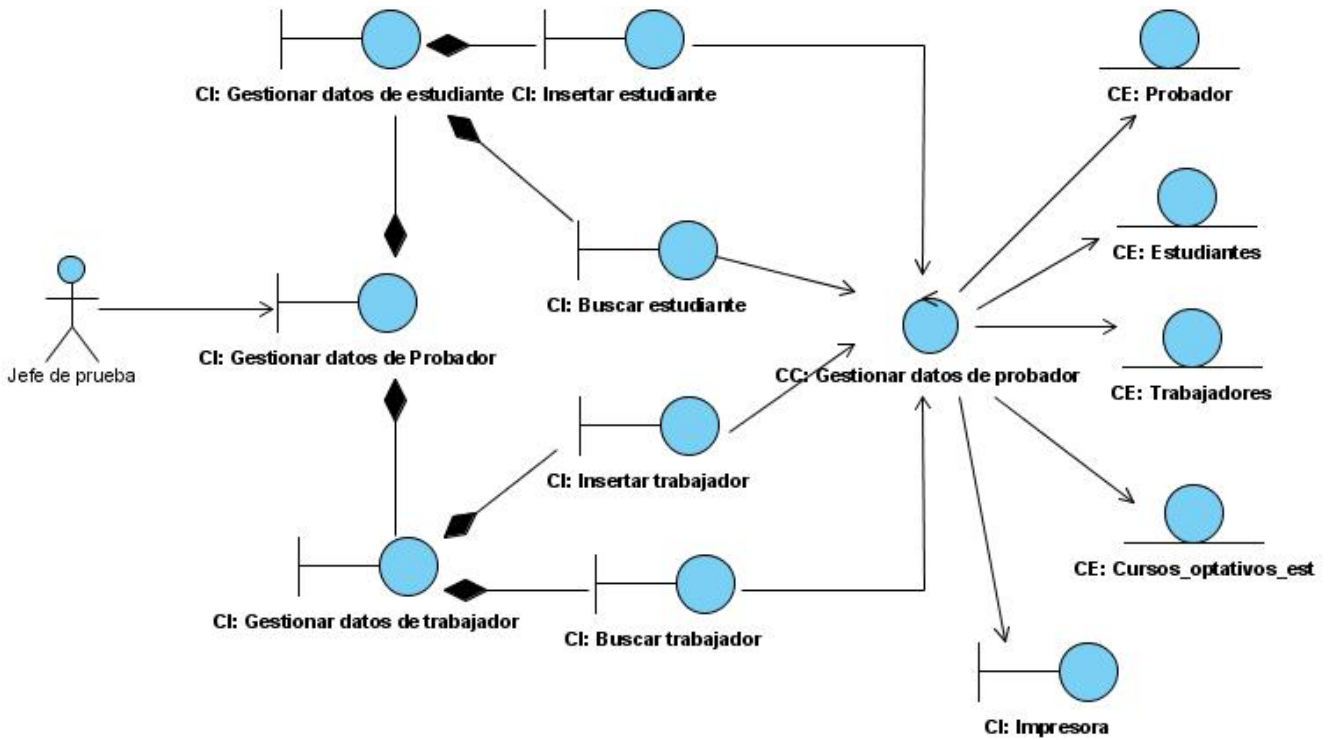
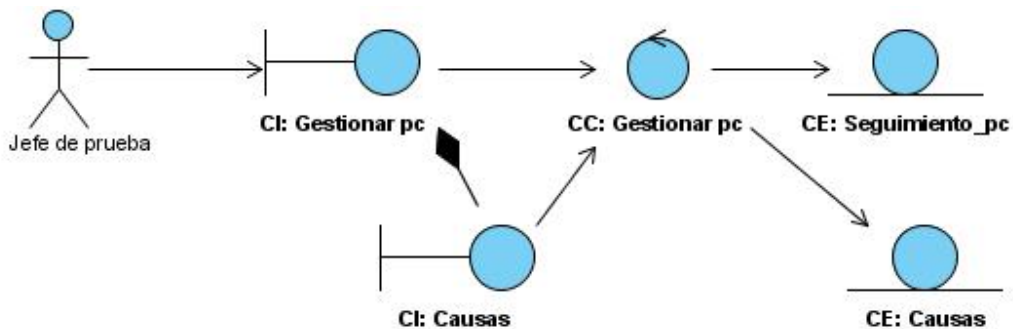


Figura 3.1. Diagrama de clases del análisis caso de uso “Gestionar datos de Jefe de prueba”.



**3.2.2 Diagrama de clases del análisis caso de uso “Gestionar datos de Probador”.**

**Figura 3.2. Diagrama de clases del análisis caso de uso “Gestionar datos de Probador”.**
**3.2.3 Diagrama de clases del análisis caso de uso “Gestionar pc”.**

**Figura 3.3. Diagrama de clases del análisis caso de uso “Gestionar pc”.**

### 3.3 Diagramas de interacción.

Describen secuencia de acciones o colaboración entre objetos y son representados por Diagramas de secuencia o de colaboración.

Diagramas de secuencia: Se muestra la interacción entre objetos mediante transferencia de mensajes entre objetos o subsistemas.

Diagramas de colaboración: Se muestra la interacción entre objeto creando enlaces entre ellos y añadiendo mensajes a esos enlaces. (Pressman 1998)

A continuación se muestran los diagramas de secuencia de los casos de usos más significativos.

#### 3.3.1 Diagrama de secuencia caso de uso “Gestionar datos de Jefe de prueba”. Sección: “Modificar Jefe de prueba”.

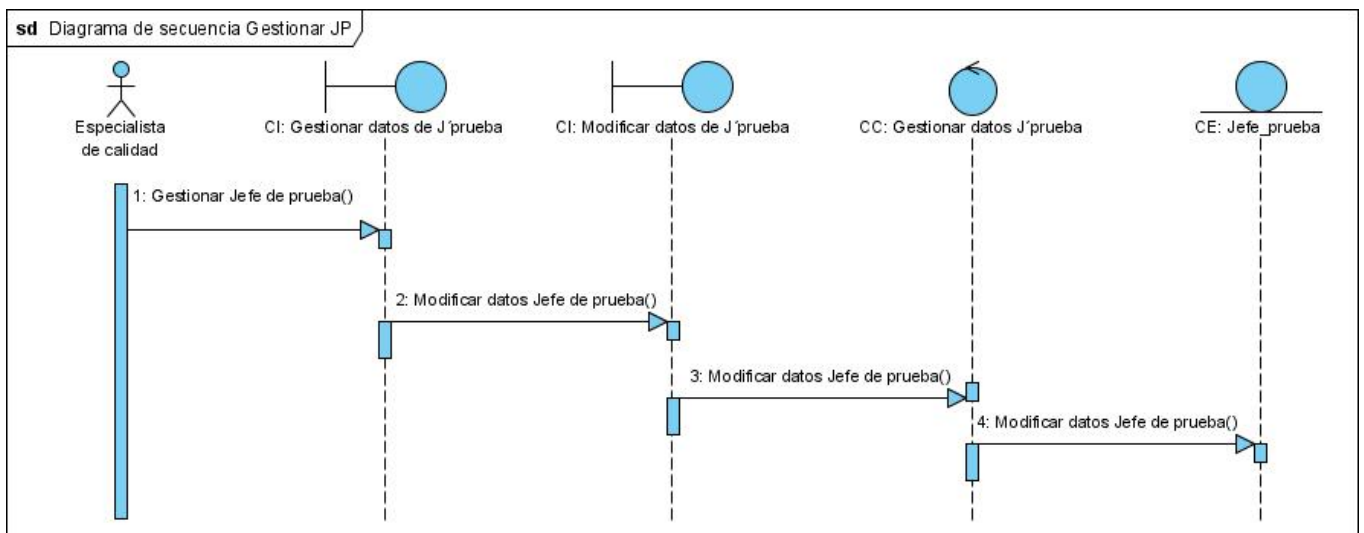
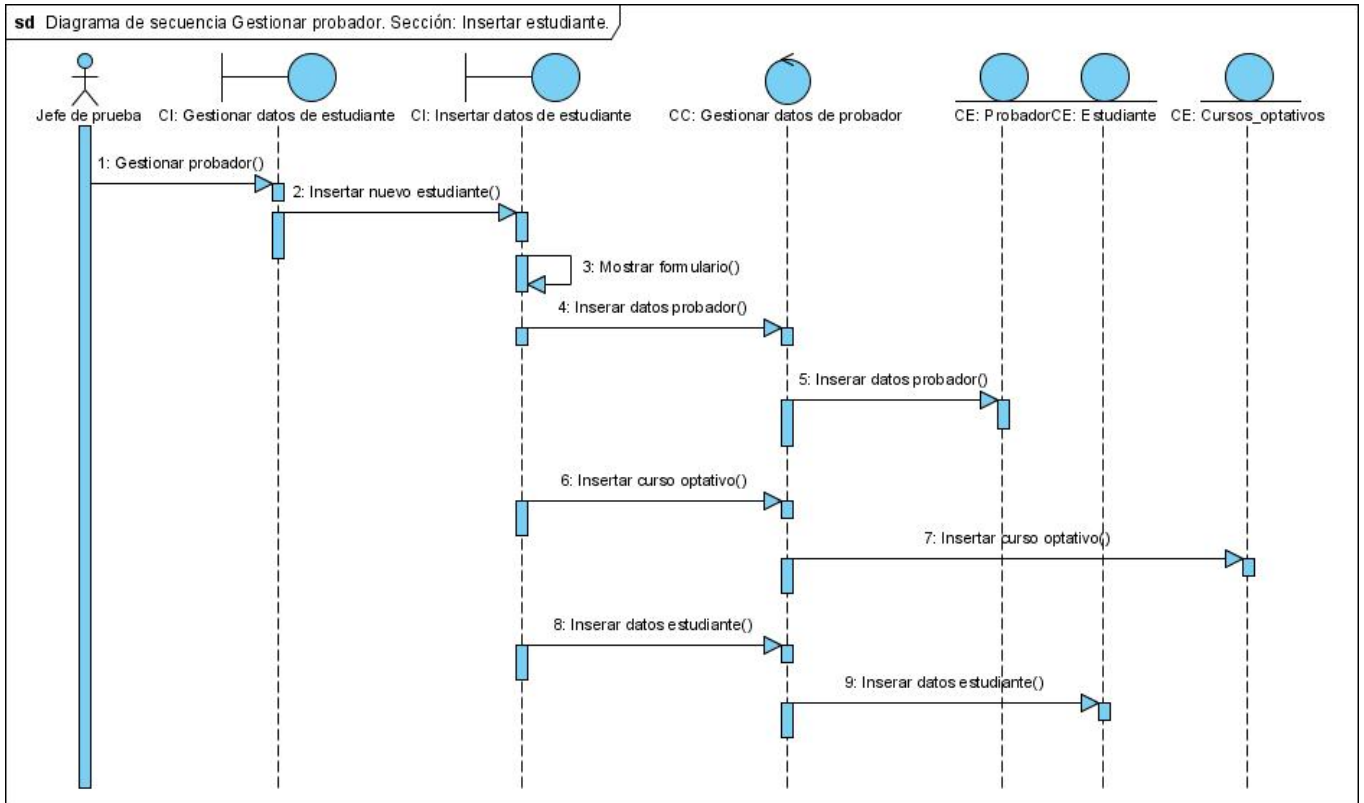
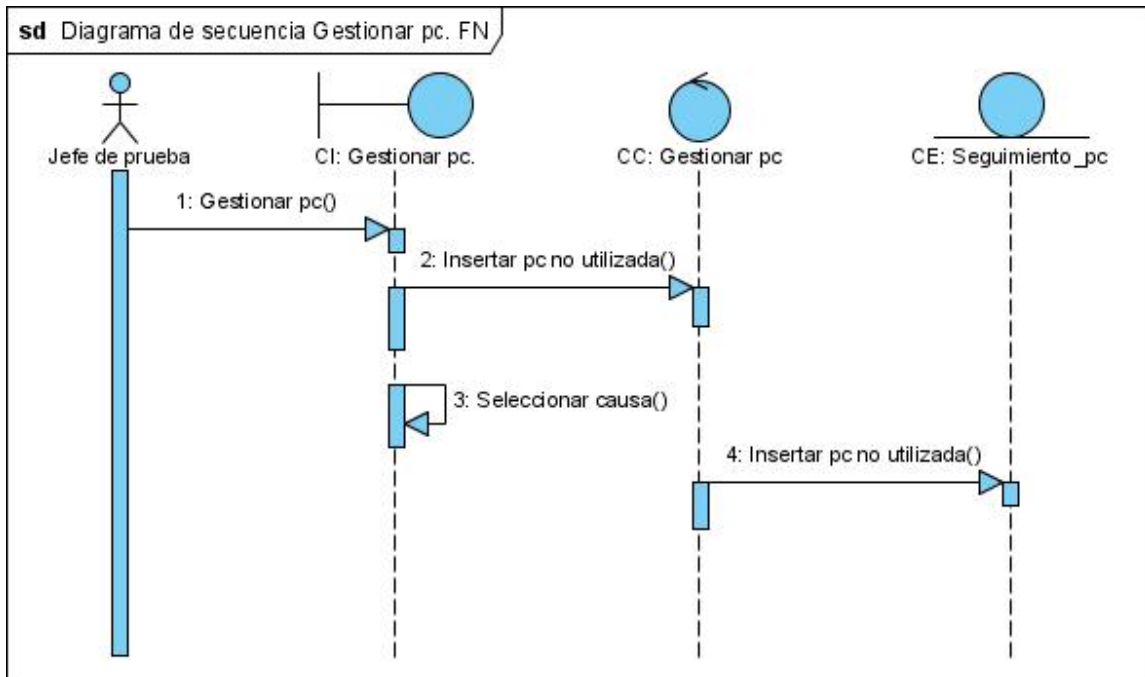


Figura 3.4. Diagrama de secuencia caso de uso “Gestionar datos de Jefe de prueba”. Sección: “Modificar Jefe de prueba”.

**3.3.2 Diagrama de secuencia caso de uso “Gestionar datos de probador”. Sección: “Insertar estudiante”.**


**Figura 3.5. Diagrama de secuencia caso de uso “Gestionar datos de probador”. Sección: “Insertar estudiante”.**

**3.3.3 Diagrama de secuencia caso de uso “Gestionar pc”.**

**Figura 3.6. Diagrama de secuencia caso de uso “Gestionar pc”.**
**3.4 Diseño.**

El propósito del diseño es modelar el sistema y encontrar su forma para que soporte todos los requisitos. Es un modelo físico que crea una entrada apropiada y un punto de partida para la implementación. (Date 2003)

El diagrama del diseño Web del sistema (diagrama de clases para diseño orientado a objetos) se obtiene como resultado del refinamiento del modelo conceptual y se basa fundamentalmente en los diagramas de interacción. Es una abstracción a la implementación del sistema.

A continuación se muestran los diagramas de clases del diseño del sistema a implementar.

### 3.4.1 Diagrama de clases del diseño caso de uso “Gestionar datos de Jefe de prueba”.

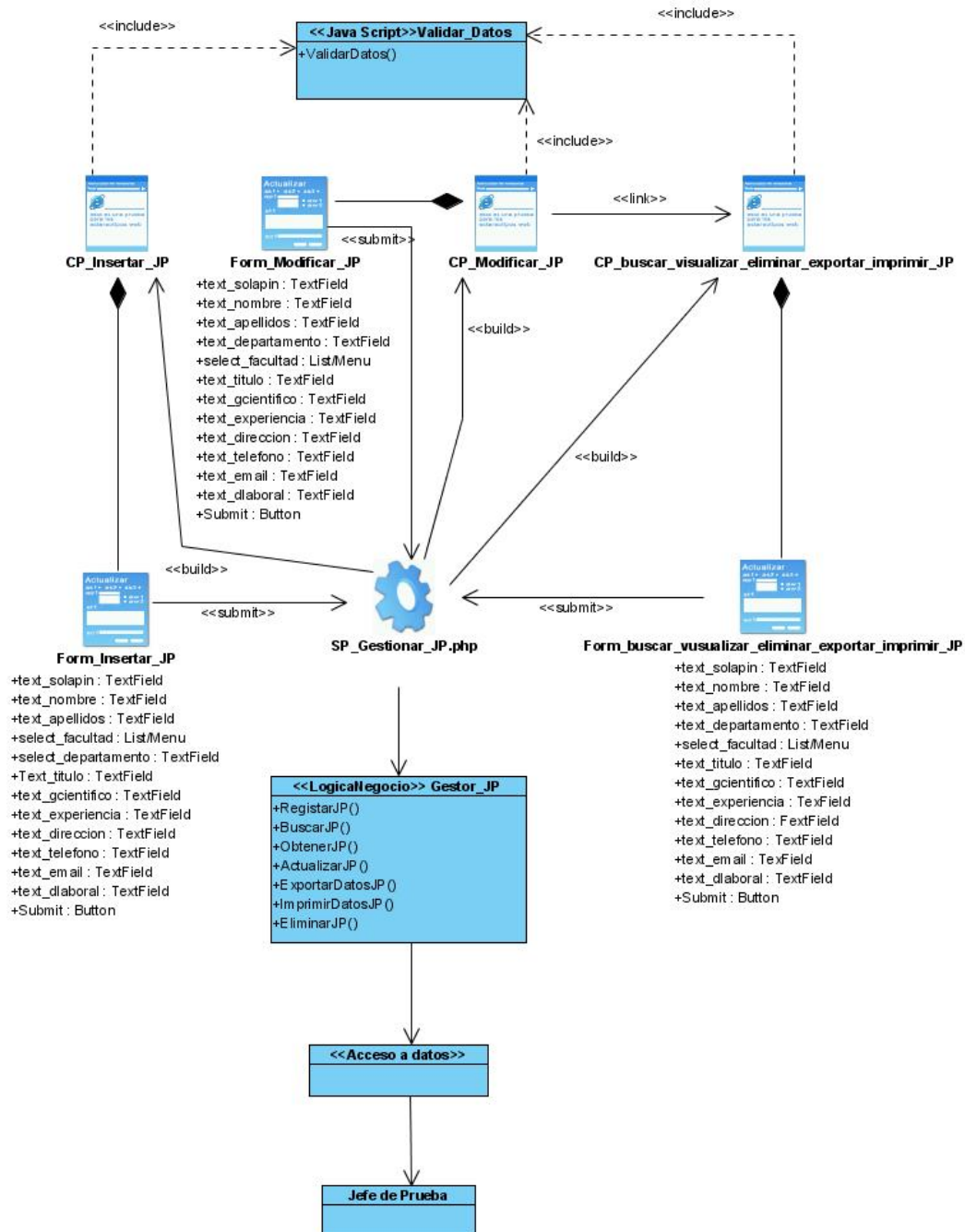
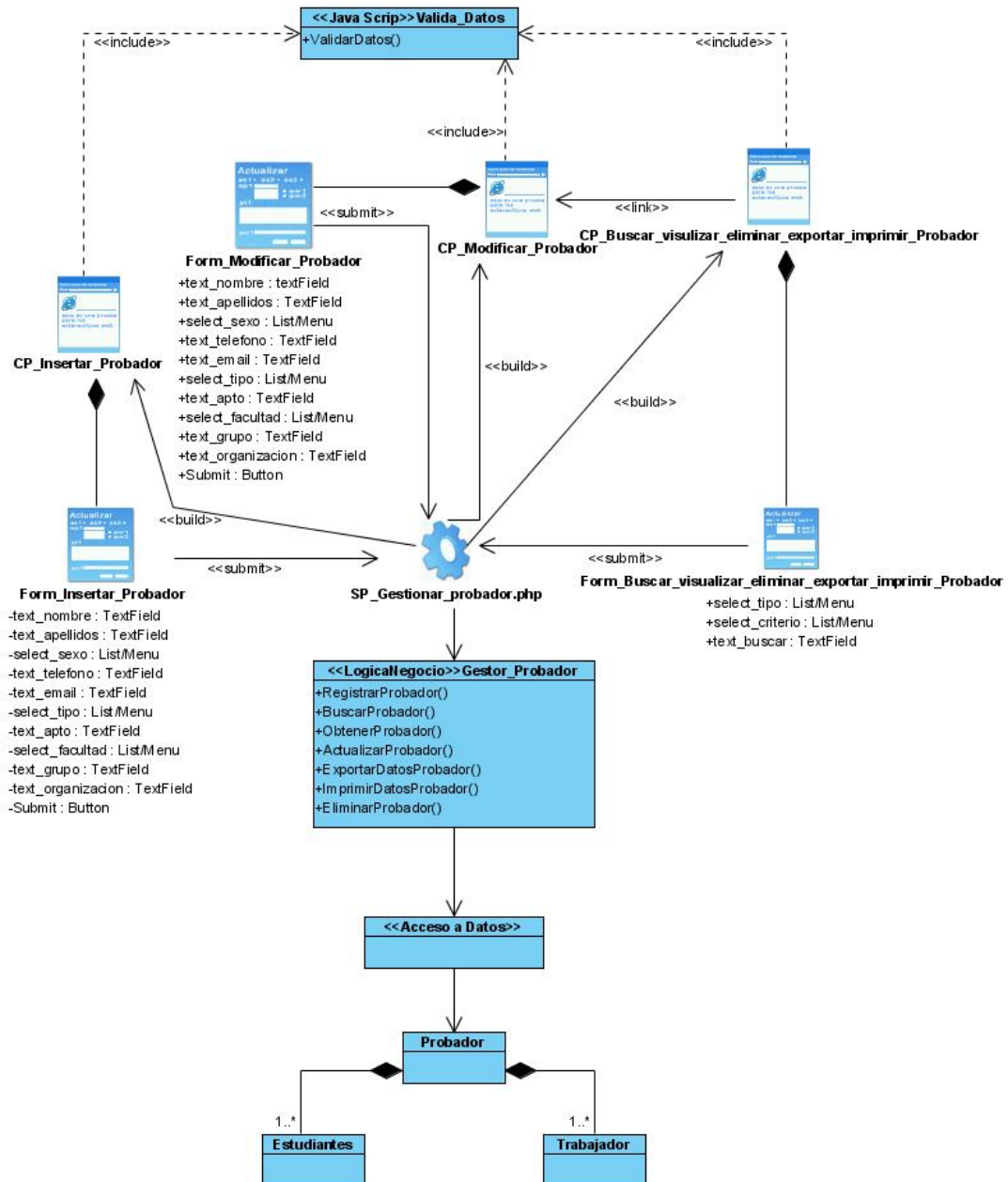
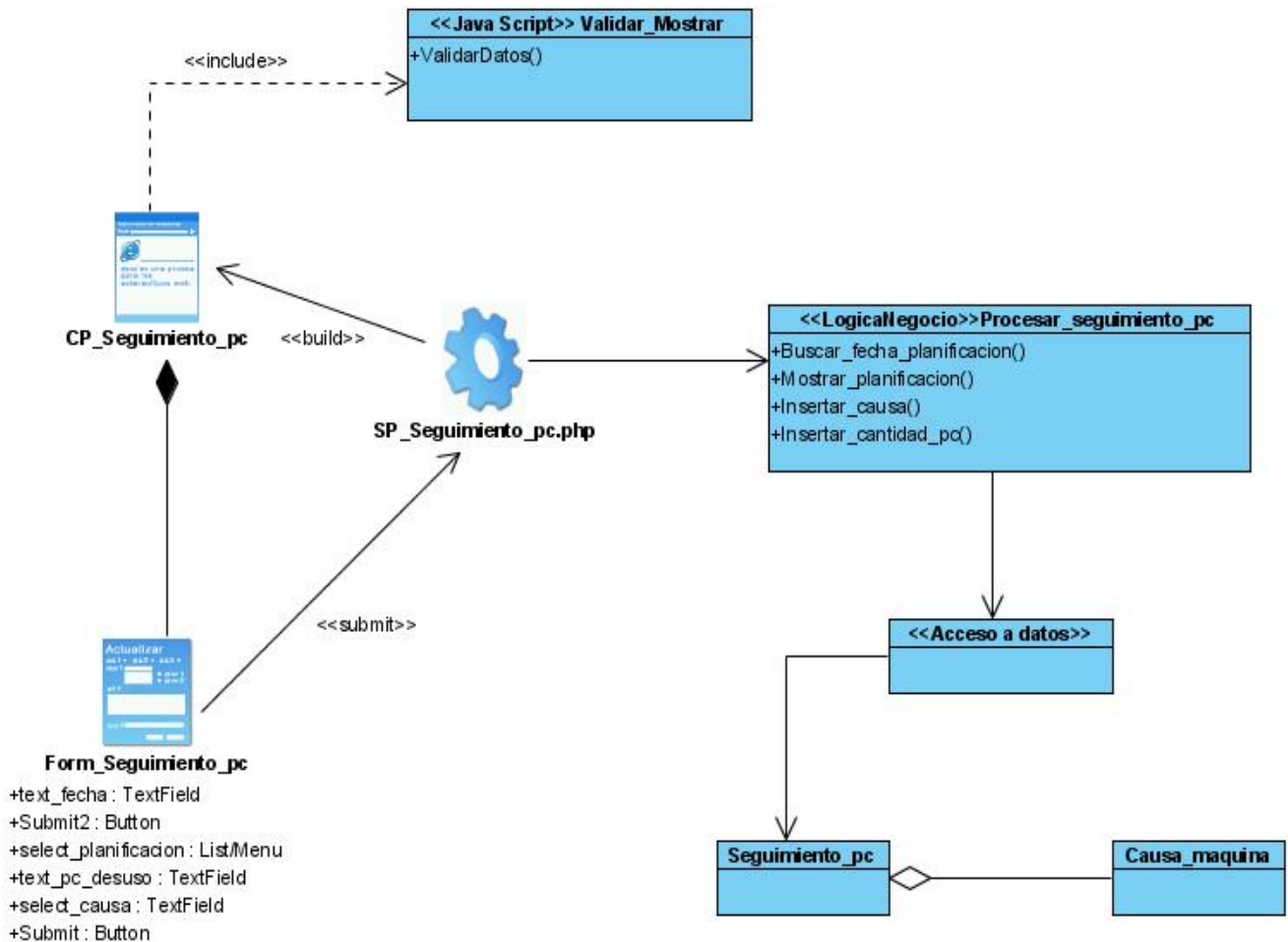
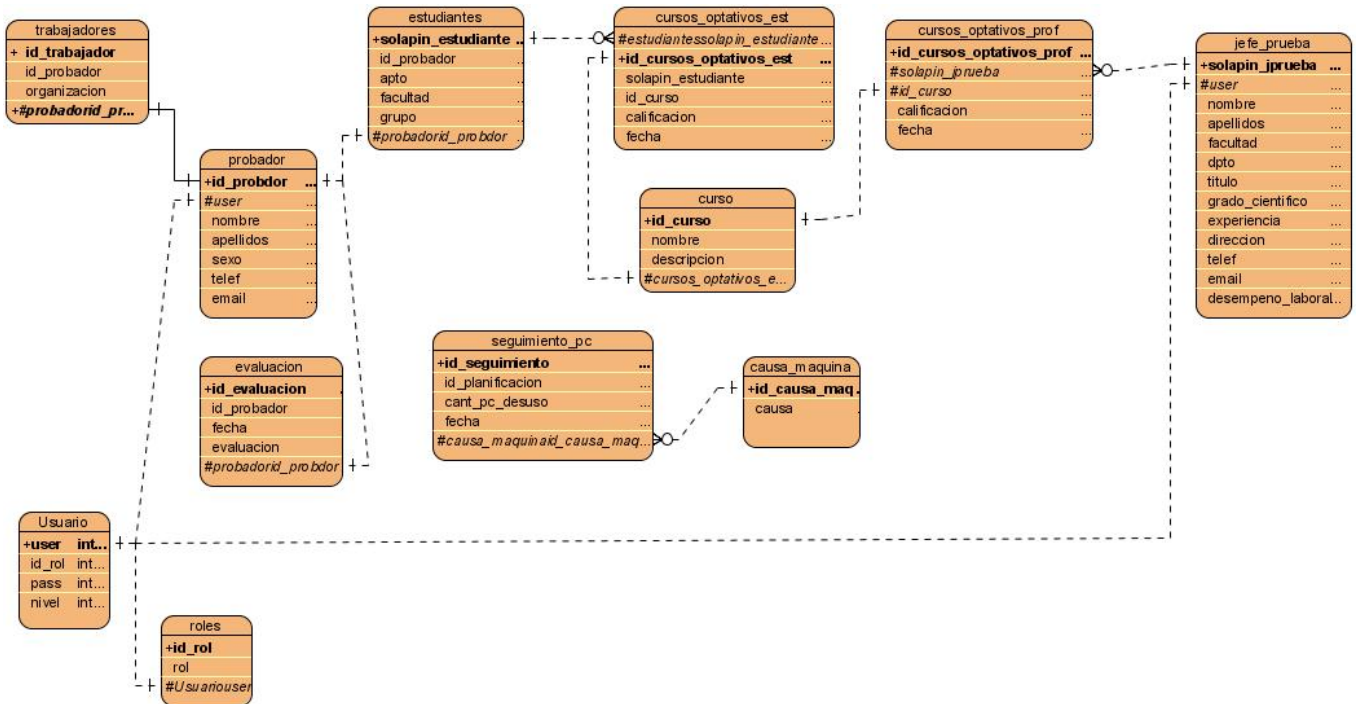


Figura 3.7. Diagrama de clases del diseño caso de uso “Gestionar datos de Jefe de prueba”.

**3.4.2 Diagrama de clases del diseño caso de uso “Gestionar datos de Probador”.**

**Figura 3.8. Diagrama de clases del diseño caso de uso “Gestionar datos de Probador”.**

**3.4.3 Diagrama de clases del diseño caso de uso “Gestionar pc”.**

**Figura 3.9. Diagrama de clases del diseño caso de uso “Gestionar pc”.**
**3.5 Diseño de la Base de Datos.**

Este diagrama de entidad de relación, conforma solo la parte del módulo de seguimiento de los recursos, como se puede observar en las tablas hay llaves foráneas que pertenecen a otras tablas de la herramienta en general, de las cuales el sistema obtiene y ofrece información.

**3.5.1 Modelo entidad-relación.**

**Figura 3.10. Diagrama de entidad relación. Módulo seguimiento de los recursos.**

Para ver el diseño de la Base de datos completa remítase a: Anexo XII.

**3.5.2 Descripción de las tablas de la base de datos. Módulo Seguimiento de los recursos.**

Nombre: probador		
<b>Descripción:</b> En esta tabla se almacenan los datos que coinciden de los estudiantes y los trabajadores, los probadores del sistema.		
Atributo	Tipo	Descripción
Id_probador	Integer	Este campo es una llave primaria y almacena el identificador que distingue al Probador.
user	String	Este campo es una llave extranjera y almacena el nombre de usuario relacionado con los probadores.



nombre	String	Este campo almacena el nombre del Probador.
apellidos	String	Este campo almacena los apellidos del Probador.
sexo	String	Este campo almacena el sexo del Probador.
telef	String	Este campo almacena el teléfono del Probador.
email	String	Este campo almacena el correo electrónico del Probador.

**Tabla 3.1. Descripción de la tabla Probador.**

<b>Nombre: estudiantes</b>		
<b>Descripción:</b> En esta tabla se almacenan los datos específicos de los estudiantes.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
solapin_estudiante	Integer	Este campo es una llave primaria y es donde se almacena el número del solapín del estudiante.
Id_probador	Integer	Este campo es una llave extranjera y almacena el identificador que distingue al Probador.
apto	String	Este campo almacena el número del apartamento UCI donde reside el estudiante.
facultad	String	Este campo almacena el número de la facultad a la que pertenece el estudiante.
grupo	String	Este campo almacena el grupo al que pertenece el estudiante.

**Tabla 3.2. Descripción de la tabla Estudiante.**

<b>Nombre: trabajadores</b>		
<b>Descripción:</b> En esta tabla se almacenan los datos específicos de los trabajadores.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_trabajador	Integer	Este campo es una llave primaria y almacena el identificador de cada uno de los trabajadores.
Id_probador	Integer	Este campo es una llave extranjera y almacena el identificador que distingue al Probador.
organizacion	String	Este campo almacena el nombre de la organización a la que pertenece el trabajador.

**Tabla 3.3. Descripción de la tabla Trabajadores.**

<b>Nombre: evaluación</b>		
<b>Descripción:</b> En esta tabla se almacenan los datos de las evaluaciones realizadas a los probadores.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_evaluacion	Integer	Este campo es una llave primaria y almacena el identificador de la evaluación de cada uno de los estudiantes.
Id_probador	Integer	Este campo es una llave extranjera y almacena el identificador que distingue al Probador.
fecha	Date	Este campo almacena la fecha de la evaluación.
evaluacion	String	Este campo almacena el valor de la

		evaluación realizada al estudiante durante su turno de trabajo.
--	--	---

**Tabla 3.4. Descripción de la tabla Evaluación.**

<b>Nombre: cursos_optativos_est</b>		
<b>Descripción:</b> En esta tabla se almacenan los datos específicos de los cursos optativos recibidos por los estudiantes.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_curso_optativo_est	Integer	Este campo es una llave primaria y almacena el identificador del curso optativo asociado a un estudiante.
solapin_estudiante	Integer	Este campo es una llave extranjera y es donde se almacena el número del solapín del estudiante.
id_curso	Integer	Este campo es una llave extranjera y almacena el identificador del curso que fue recibido por cualquiera de las personas involucradas en el laboratorio, ya sea estudiante o trabajador.
calificacion	Integer	Este campo almacena la calificación obtenida de un estudiante en un curso optativo.
fecha	Date	Este campo almacena la fecha en la que un estudiante recibió un curso optativo.

**Tabla 3.5. Descripción de la tabla Cursos\_optativos\_est.**

<b>Nombre: cursos</b>		
<b>Descripción:</b> En esta tabla se almacenan los datos de todos los cursos optativos.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_curso	Integer	Este campo es una llave primaria y almacena el identificador del curso que fue recibido por cualquiera de las personas involucradas en el laboratorio, ya sea estudiante o trabajador.
nombre	String	Este campo es una llave primaria y almacena el nombre de los cursos optativos.
descripcion	String	Este campo almacena la descripción de los cursos optativos.

**Tabla 3.6. Descripción de la tabla Cursos.**

<b>Nombre: seguimiento_pc</b>		
<b>Descripción:</b> En esta tabla se almacenan los datos de las pc sin utilizar.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_seguimiento	Integer	Este campo es una llave primaria y almacena el identificador de seguimiento.
Id_planificacion	Integer	Este campo es una llave extranjera y almacena el identificador de las planificaciones.
cant_pc_desuso	Integer	Este campo almacena el número de pc que no se usan.
fecha	String	Este campo almacena la fecha en que la pc no se utilizó.

**Tabla 3.7. Descripción de la tabla Seguimiento\_pc.**

**Nombre: causa\_maquina**

**Descripción:** En esta tabla se almacenan las diferentes causas por las que las pc no se pueden utilizar.

Atributo	Tipo	Descripción
Id_causa_maq	Integer	Este campo es una llave primaria y almacena el identificador de la causa por la que no se utiliza una pc.
Id_seguimiento	Integer	Este campo es una llave extranjera y almacena el identificador de seguimiento.
causas	String	Este campo almacena las diferentes causas por las que las pc no se pueden utilizar.

**Tabla 3.8. Descripción de la tabla Causa\_maquina.**

**Nombre: jefe\_prueba**

**Descripción:** En esta tabla se almacenan los datos específicos de los Jefes de prueba.

Atributo	Tipo	Descripción
solapin_jprueba	Integer	Este campo es una llave primaria y es donde se almacena el número del solapín del Jefe de prueba.
user	String	Este campo es una llave extranjera y almacena el nombre de usuario relacionado con el Jefe de prueba.
nombre	String	Este campo almacena el nombre del Jefe de prueba.
apellidos	String	Este campo almacena los apellidos del Jefe de prueba.
facultad	String	Este campo almacena el número de la facultad a la que pertenece el Jefe de

		prueba.
dpto	String	Este campo almacena el departamento al que pertenece el Jefe de prueba.
titulo	String	Este campo almacena el título de graduación de cada Jefe de prueba
grado_cientifico	String	Este campo almacena el grado científico de cada Jefe de prueba.
experiencia	Integer	Este campo almacena los años de experiencia en el tema.
direccion	String	Este campo almacena la dirección del Jefe de prueba.
telef	String	Este campo almacena el teléfono del Jefe de prueba.
email	String	Este campo almacena el correo electrónico del Jefe de prueba.
desempeno_laboral	String	Este campo almacena los cursos que ha pasado cada Jefe de prueba.

**Tabla 3.9. Descripción de la tabla Jefe\_prueba.**

<b>Nombre: cursos_optativos_prof</b>		
<b>Descripción:</b> En esta tabla se almacenan los datos de los cursos optativos de los jefes de pruebas.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_cursos_optativos_prof	Integer	Este campo es una llave primaria y almacena el identificador de los cursos recibidos por cada Jefe de prueba.
Solapin_jprueba	Integer	Este campo es una llave extranjera y almacena el número del solapín del Jefe de

		prueba.
Id_curso	Integer	Este campo es una llave extranjera y almacena el identificador de los cursos recibidos por cada Jefe de prueba.
Calificación	Integer	Este campo almacena la calificación obtenida de un Jefe de prueba en un curso optativo.
Fecha	Date	Este campo almacena la fecha en la que un Jefe de prueba recibió un curso optativo.

**Tabla 3.10. Descripción de la tabla Cursos\_optativos\_prof.**

### 3.6 Principios de diseño.

Para garantizar un vínculo entre el usuario y el sistema realizado se ha hecho un esfuerzo por garantizar la máxima claridad posible en la interfaz procurando que sea intuitiva y que garantice un empleo óptimo del sistema y una comodidad en el trabajo de los usuarios finales.

Se trabajó con formularios Web, para lograr una mayor visualización de la información, se utilizó un mismo color en todas las páginas y un mismo tipo y tamaño de letra con textos claros, el color predominante es el azul.

El envío de la información debe ser lo mas rápida posible por lo que no se utilizaron muchas imágenes ni funciones que atenten contra esto y que posibiliten también una navegación rápida y eficiente.

Cada página está representada por un título acorde a su funcionalidad. Los botones principales están adornados con iconos que sirven de referencia a la función de los mismos. A continuación se presentan algunos de estos íconos:



Modificar usuario



Insertar usuario



Cambiar claves de usuarios



Eliminar usuario

### 3.7 Tratamiento de errores.

Se contará con un sistema de tratamiento de errores para disminuir la posibilidad de cometerlos. Para esto se contará con la validación de la información introducida en el sistema por la validación de los formularios utilizando funciones Java Scripts.

Mediante la interfaz Web se evitará que el usuario asuma un papel activo en la introducción de la información, para esto se contará con cuadros de opción, menú de selección lo cual facilitará la entrada de datos. La información que requiera ser adicionada por el usuario se validará mediante funciones que garanticen que sea válida y que el cuadro de texto no esté vacío si es obligatorio llenarlo. Si hay un error en la información le saldrá al usuario un mensaje en pantalla indicándole el error, al oprimir "Aceptar" el mensaje desaparece y el usuario podrá seguir introduciendo los datos en el formulario.

También se validarán las opciones correspondientes a la extracción o modificación de datos del servidor de base datos. Si se desea eliminar algún elemento de la BD se preguntará al usuario si está seguro de realizar dicha acción, al igual que cuando desee modificar alguna información, antes de actualizarla se le preguntará si desea realizarla o no. Así se logra que se realicen las operaciones que se desean y que se rectifique al cometer un error.

A continuación se muestran algunos de estos mensajes:



**Figura 3.11. Mensaje de verificación. Eliminar usuario.**





**Figura 3.12. Mensaje de validación. Campo Correo electrónico.**



**Figura 3.13. Mensaje de validación. Campo Solapín.**



**Figura 3.14. Mensaje de validación. Campo Nombre.**

### **3.8 Seguridad.**

La seguridad en el sitio está implementada a través del servidor de base de datos MySQL y el uso de variables de sesión para restringir el acceso de los usuarios a determinadas páginas. Así como también un Sub\_Módulo de autenticación que verifica que el usuario existe en la aplicación y que los datos que introduce sean correctos, en caso de no ser así no tendrá acceso a la aplicación.

Para garantizar la seguridad de la información se crearon varios niveles de seguridad, definidos como tipos de usuario, que pueden ser: Especialista de Calidad, Jefe de Prueba, Probador, Desarrollador y Administrador. Este último es el encargado del buen funcionamiento del sistema por lo que tendrá derecho al control total del mismo. Los demás usuarios no tendrán acceso a la información restringida para ellos, para esto se trabaja con variables de sesión de forma tal que siempre se sabe qué usuario intenta visitar dichas páginas y estas se muestran solo para aquellos que pueden tener acceso a ellas. A continuación se muestran algunos de estos mensajes:



**Figura 3.15. Mensaje de verificación. Campo Contraseñas.**

Autenticarse	
Usuario :	<input type="text"/>
Contraseña:	<input type="password"/>
Error: Password ó Usuario no existe	
<input type="button" value="Entrar"/>	

**Figura 3.16. Interfaz de autenticación.**

### 3.9 Concepción de la ayuda.

El sistema consta de una ayuda que permite a los usuarios del mismo conocer aspectos de su funcionamiento, el vínculo para acceder a la ayuda se encuentra en la plantilla del sistema por lo que

se podrá consultar desde cualquier interfaz, en ella se explica de forma clara y amena todas las funcionalidades del sistema por niveles de permiso.

### **Conclusiones.**

En el capítulo se realizaron las definiciones del Modelo de Análisis, Modelo de clases de análisis así como los diagramas de análisis. Y se realizaron los diferentes diagramas de diseño, diagramas de secuencia, se describieron las clases utilizadas en el diseño, así como el modelo entidad relación de la base de datos y las descripciones de las tablas de la base de datos. También se explicó la definición del diseño que se aplicó, la forma de tratar los errores, se definió la seguridad del sistema y la interfaz.



## *Capítulo 4. Implementación y prueba.*



### **Introducción.**

En el capítulo se muestran las definiciones de los diferentes modelos utilizados en la implementación para la solución del problema y se procede a la construcción de los modelos especificados en los requisitos. Se realiza el modelo de despliegue y el modelo de componentes. El sistema se implementa en términos de componentes: ficheros de código fuente, scripts, ficheros de código binario, ejecutables, entre otros. El modelo de implementación parte del diseño.

### **4.1 Implementación.**

Con el flujo de trabajo Implementación se le está dando comienzo a la fase de Construcción, el propósito de esta fase es dejar listo un producto software en su versión operativa inicial (versión beta), el cual debe tener la calidad adecuada para su aplicación y cumplir con los requisitos de software establecidos en el Capítulo II.

Al final de la fase de Elaboración se ha llevado el producto software al estado de una línea base de la arquitectura ejecutable. Las fases previas han reducido los riesgos críticos y significativos a niveles rutinarios que pueden ser gestionados durante el plan de construcción. Durante esta fase, el equipo del proyecto estableció los fundamentos de los elementos arquitectónicamente significativos de los modelos de diseño y despliegue.

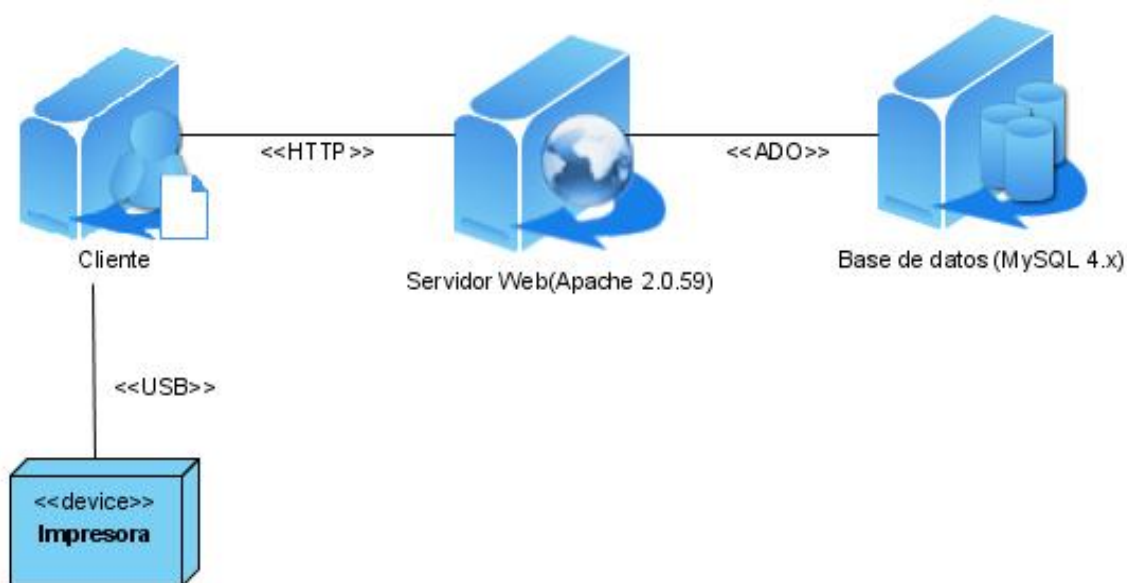
Los objetivos principales del flujo de trabajo Implementación son:

- Definir la organización del código en términos de Subsistemas de Implementación organizados en capas.
- Implementar los elementos de diseño en términos de elementos de implementación (ficheros fuentes, binarios, ejecutables y otros).
- Probar los componentes desarrollados independientemente como unidades.

- Integrar los resultados producidos por desarrolladores independientes o equipos en un sistema ejecutable.

#### 4.1.1 Diagrama de despliegue.

En el diagrama de despliegue se indica la situación física de los componentes lógicos desarrollados, es decir, se sitúa el software en el hardware que lo contiene; cada hardware se representa como un nodo.



**Figura 4.1. Diagrama de despliegue.**

#### 4.1.2 Diagrama de componentes.

Se representa como un grafo de componentes software unidos por medio de relaciones de dependencia, pudiendo mostrarse las interfases que estos soporten.

Es un diagrama que muestra un conjunto de elementos del modelo tales como componentes, subsistemas de implementación y sus relaciones.

Se utiliza para modelar la vista estática de un sistema. Muestra la organización y las dependencias lógicas entre un conjunto de componentes software, los cuales pueden ser código fuente, librerías, binarios o ejecutables.

A continuación se muestra el diagrama de componentes del sistema organizado en paquetes.

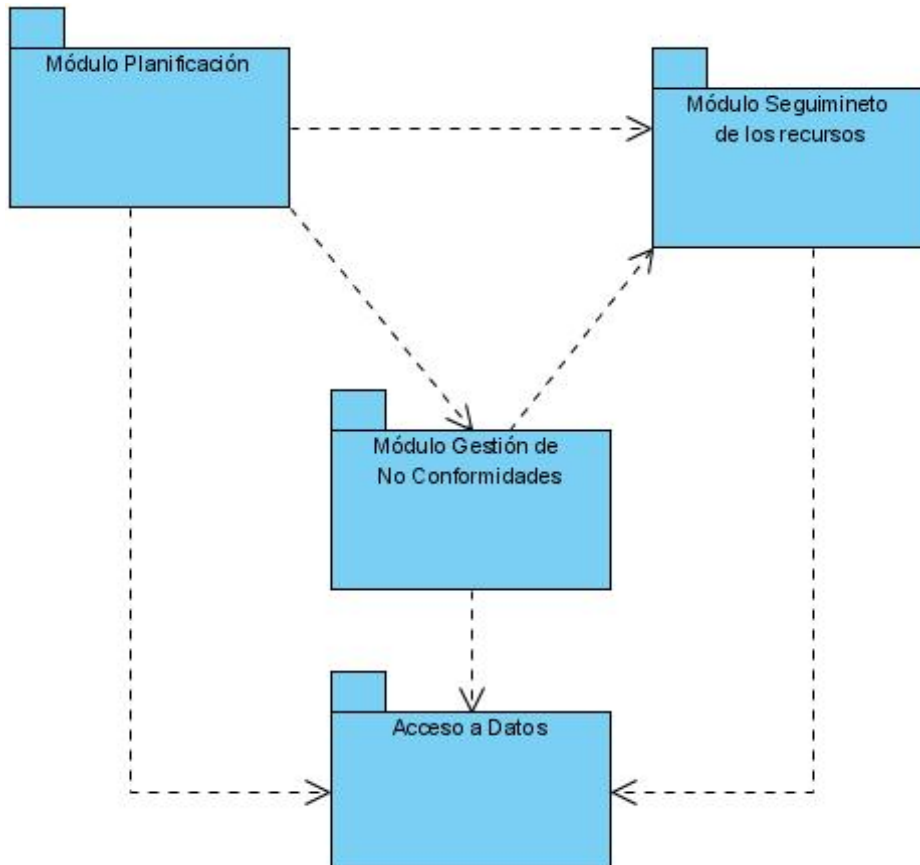
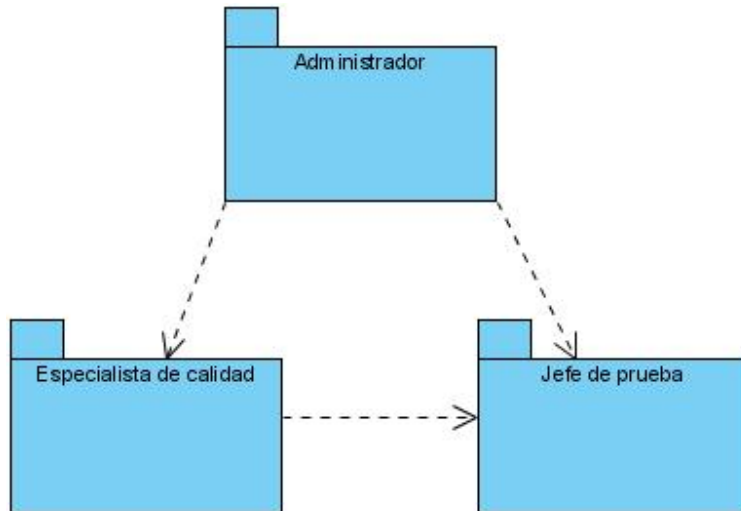
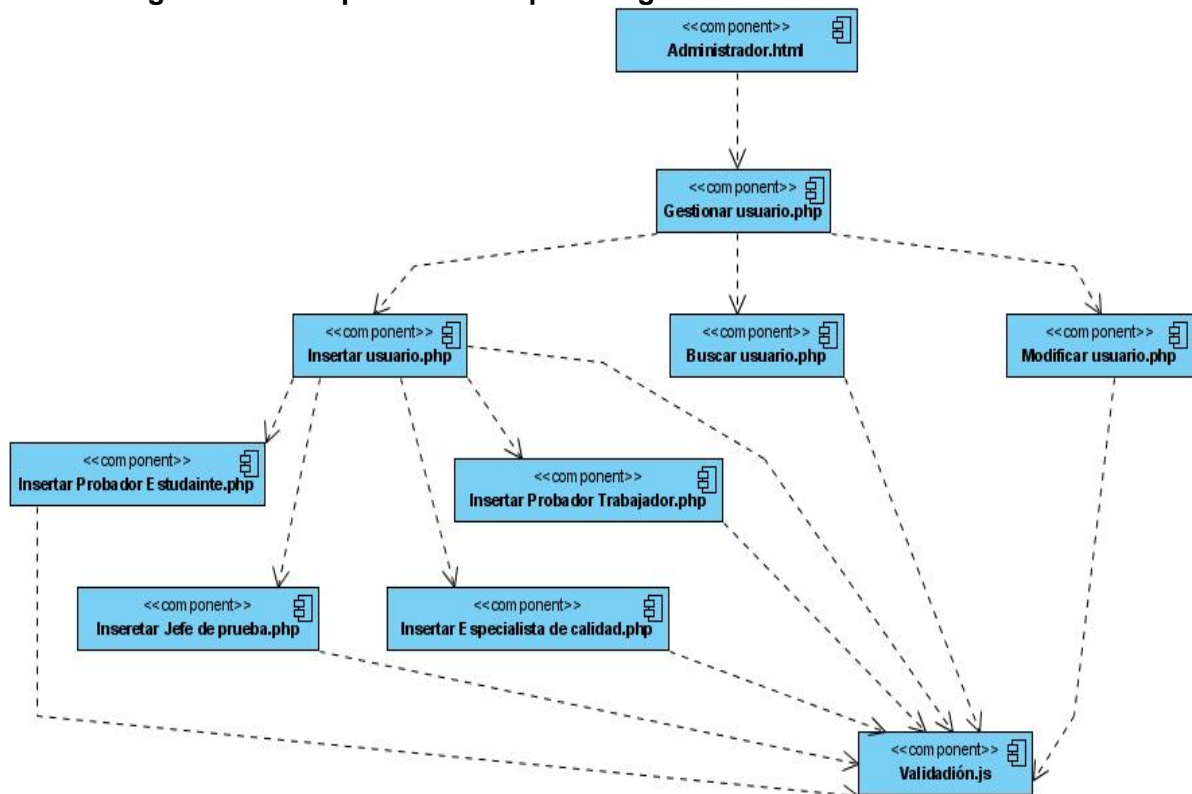


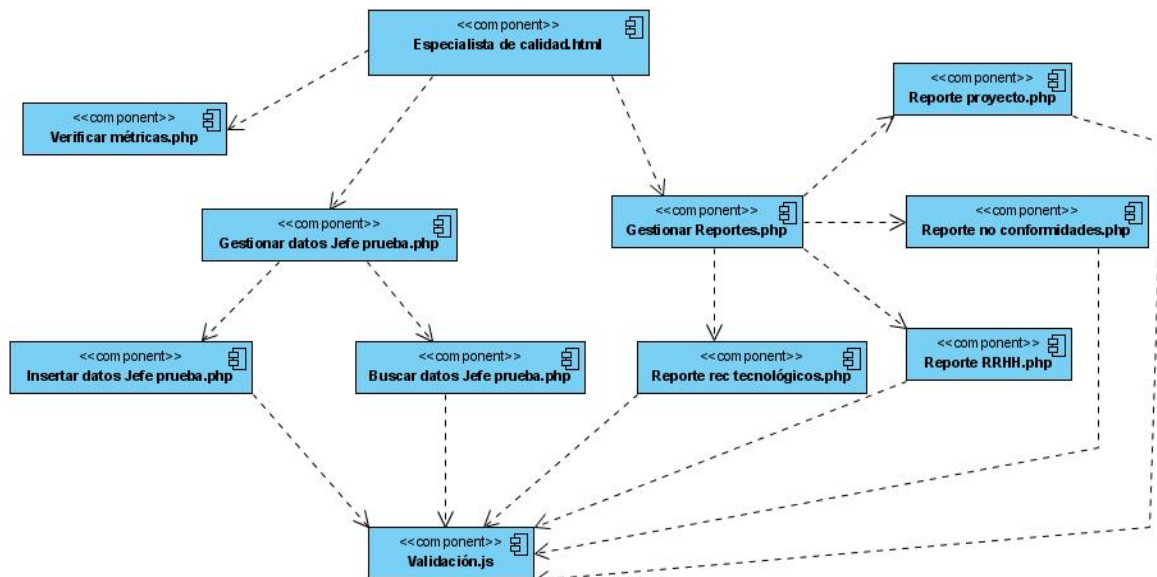
Figura 4.2. Diagrama de componente por paquetes.



**Figura 4.3. Diagrama de componentes. Paquete Seguimiento de los recursos.**

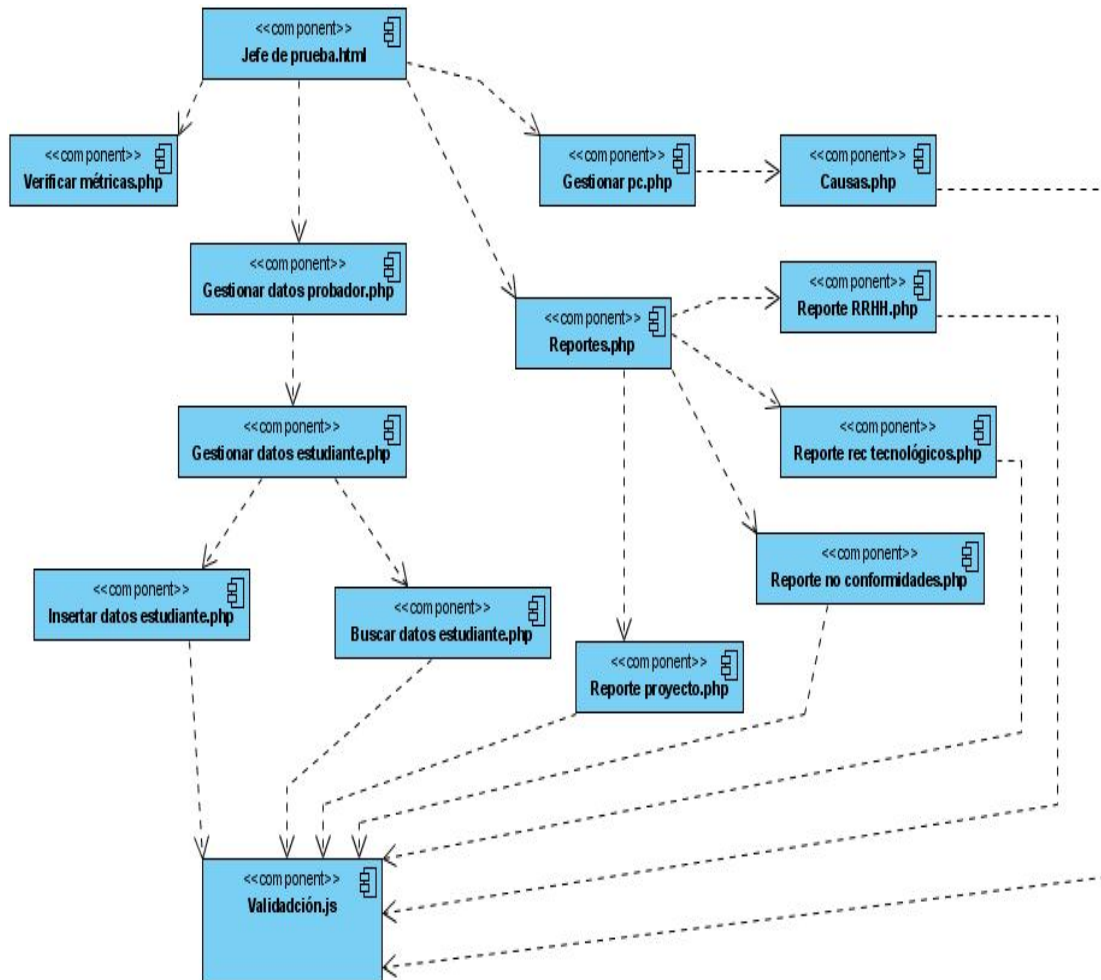


**Figura 4.4. Paquete Seguimiento de los recursos. Vista Administrador.**

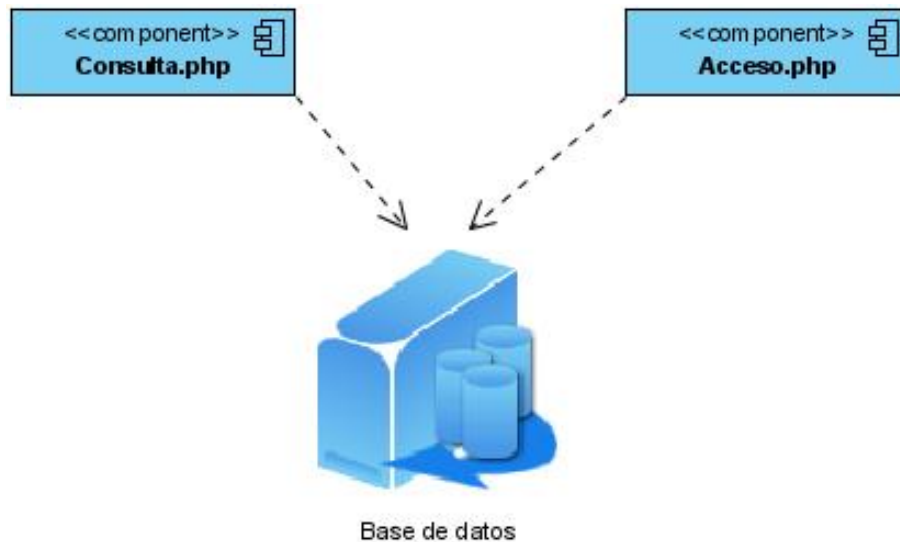


**Figura 4.5. Paquete Seguimiento de los recursos. Vista Especialista de calidad.**





**Figura 4.6. Paquete Seguimiento de los recursos. Vista Jefe de prueba.**



**Figura 4.7. Diagrama de componentes. Paquete Acceso a datos.**

#### **4.2 Prueba.**

La prueba es la última fase del proceso de calidad de software y representa una revisión final de las especificaciones, del diseño y de la codificación.

El proceso de pruebas se divide en etapas o fases, y está compuesto por un conjunto de técnicas, que permiten obtener una serie de casos de pruebas, con el objetivo de descubrir los defectos del software. Dicho de otra forma, es el arte de diseñar casos de pruebas, que sean capaces de detectar los errores existentes en el software con la mínima cantidad de esfuerzo y tiempo, dependiendo de la habilidad en el uso de las técnicas de prueba, y de la estrategia trazada.

Las pruebas de integración constituyen una técnica sistemática para construir la estructura del programa, mientras que al mismo tiempo se llevan a cabo pruebas para detectar errores asociados con la interacción. El objetivo es coger los módulos probados durante la Prueba de Unidad y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño.

A continuación se muestra el diseño del caso de prueba de integración realizado al "Sistema Automatizado para los procesos en el Laboratorio de Calidad"

## Módulo Planificación.

Entrada	Resultados	Condiciones
<ul style="list-style-type: none"> <li>• Se escoge el Jefe de prueba.</li> <li>• Se escoge el proyecto.</li> </ul>	<ul style="list-style-type: none"> <li>• Se muestra una interfaz con el nombre y apellidos del jefe de prueba seleccionados y muestra el listado de los módulos del proyecto.</li> </ul>	<ul style="list-style-type: none"> <li>• En el módulo de Seguimiento de debió haber insertado con anterioridad el jefe de prueba.</li> <li>• En el módulo de Planificación se debió haber insertado con anterioridad el proyecto.</li> </ul>
<ul style="list-style-type: none"> <li>• Se selecciona la prueba a planificar.</li> </ul>	<ul style="list-style-type: none"> <li>• Aparece una interfaz donde se insertan los recursos a planificar.</li> </ul>	<ul style="list-style-type: none"> <li>• En el módulo de Planificación se debió haber insertado con anterioridad la prueba a planificar.</li> </ul>
<ul style="list-style-type: none"> <li>• Se selecciona insertar una prueba.</li> </ul>	<ul style="list-style-type: none"> <li>• El sistema inserta nuevas pruebas.</li> </ul>	<ul style="list-style-type: none"> <li>• Se debe introducir en el módulo de planificación: <ul style="list-style-type: none"> <li>○ Nombre de la prueba</li> <li>○ Clasificación</li> <li>○ Métricas asociadas a la prueba.</li> </ul> </li> </ul>
<ul style="list-style-type: none"> <li>• Se selecciona insertar un proyecto.</li> </ul>	<ul style="list-style-type: none"> <li>• El sistema inserta nuevos proyectos y muestra la interfaz para insertar sus módulos.</li> </ul>	<ul style="list-style-type: none"> <li>• Se debe introducir en el módulo de planificación: <ul style="list-style-type: none"> <li>○ Nombre del proyecto</li> <li>○ Iteración del proyecto.</li> <li>○ Fecha</li> <li>○ Jefe de Prueba.</li> </ul> </li> </ul>
<ul style="list-style-type: none"> <li>• Un usuario se autentica por el Rol de Especialista.</li> </ul>	<ul style="list-style-type: none"> <li>• Se muestra la interfaz del especialista con sus permisos.</li> </ul>	<ul style="list-style-type: none"> <li>• El administrador debió haber asignado el rol de especialista al usuario.</li> </ul>

	<p><i>Planificación:</i></p> <ul style="list-style-type: none"> <li>○ Gestionar Datos del Proyecto.</li> <li>○ Gestionar Métricas.</li> <li>○ Gestionar Planificaciones.</li> <li>○ Gestionar Pruebas</li> <li>○ Insertar Pruebas.</li> </ul> <p><i>No conformidades:</i></p> <ul style="list-style-type: none"> <li>○ Mostrar estados de las pruebas.</li> <li>○ Gestionar las No Conformidades.</li> <li>○ Datos de los elementos.</li> <li>○ Reporte de todos los proyectos.</li> </ul> <p><i>Seguimiento:</i></p> <ul style="list-style-type: none"> <li>○ Gestionar datos de Jefe de prueba.</li> <li>○ Generar reportes.</li> <li>○ Gestionar Cursos optativos.</li> <li>○ Verificar cumplimiento de métricas.</li> </ul>	
<ul style="list-style-type: none"> <li>• Un usuario se autentica por</li> </ul>	<ul style="list-style-type: none"> <li>• Se muestra la interfaz del</li> </ul>	<ul style="list-style-type: none"> <li>• El administrador o el Especialista</li> </ul>

<p>el Rol de Jefe de Prueba.</p>	<p>Jefe de prueba con sus permisos.</p> <p><i>Planificación:</i></p> <ul style="list-style-type: none"> <li>○ Gestionar Datos del Proyecto.</li> <li>○ Gestionar Planificaciones.</li> <li>○ Insertar datos del proyecto.</li> <li>○ Planificar Pruebas.</li> </ul> <p><i>No conformidades:</i></p> <ul style="list-style-type: none"> <li>○ Asignar trabajo.</li> <li>○ Mostrar estado de las pruebas.</li> <li>○ Gestionar las No Conformidades.</li> <li>○ Insertar nuevo elemento de prueba.</li> <li>○ Datos de los elementos de prueba.</li> <li>○ Reportes de los proyectos en los que ha trabajado.</li> </ul> <p><i>Seguimiento:</i></p> <ul style="list-style-type: none"> <li>○ Gestionar datos de Probador.</li> <li>○ Evaluar Probador.</li> <li>○ Gestionar PC</li> </ul>	<p>debieron haber asignado el rol de Jefe de prueba al usuario.</p>
----------------------------------	---	---

Módulo Gestión de las No conformidades.

Entrada	Resultados	Condiciones
<ul style="list-style-type: none"> <li>• Seleccionar proyecto.</li> </ul>	<ul style="list-style-type: none"> <li>• Se muestra la lista de módulos correspondientes al proyecto.</li> </ul>	<ul style="list-style-type: none"> <li>• Se debe de haber insertado el proyecto en el modulo de planificación.</li> </ul>
<ul style="list-style-type: none"> <li>• Se selecciona el módulo al cual se le van a hacer las pruebas.</li> </ul>	<ul style="list-style-type: none"> <li>• Se muestra la interfaz para insertar las no conformidades del modulo probado.</li> </ul>	<ul style="list-style-type: none"> <li>• Se debe de haber insertado los elementos de prueba en el modulo de No Conformidades.</li> </ul>
<ul style="list-style-type: none"> <li>• Se selecciona insertar elementos de prueba.</li> </ul>	<ul style="list-style-type: none"> <li>• Se muestra una interfaz para escoger el modulo al cual se le van a insertar los elementos de prueba.</li> </ul>	<ul style="list-style-type: none"> <li>• Se debe de haber insertado el proyecto en el módulo de planificación.</li> </ul>
<ul style="list-style-type: none"> <li>• Se selecciona asignar trabajo.</li> </ul>	<ul style="list-style-type: none"> <li>• Se muestra una interfaz con los probadores y elementos de prueba.</li> </ul>	<ul style="list-style-type: none"> <li>• Se deben de haber insertado los probadores en el módulo de Seguimiento y los elementos de prueba en el módulo No Conformidades.</li> </ul>
<ul style="list-style-type: none"> <li>• Se selecciona insertar No Conformidades.</li> </ul>	<ul style="list-style-type: none"> <li>• Se muestra la interfaz para insertar las no conformidades.</li> </ul>	<ul style="list-style-type: none"> <li>• Se debe de haber insertado el proyecto en el módulo de planificación y los elementos de prueba en el módulo de No Conformidades.</li> </ul>
<ul style="list-style-type: none"> <li>• Se selecciona insertar respuestas de las no conformidades.</li> </ul>	<ul style="list-style-type: none"> <li>• Se muestra la interfaz para insertar respuesta a las no conformidades.</li> </ul>	<ul style="list-style-type: none"> <li>• Se debe de haber insertado no conformidades al sistema en el módulo No Conformidades.</li> </ul>
<ul style="list-style-type: none"> <li>• Un usuario se autentica por el Rol de Especialista.</li> </ul>	<ul style="list-style-type: none"> <li>• Se muestra la interfaz del especialista con sus permisos.</li> </ul> <p><i>Planificación:</i></p> <ul style="list-style-type: none"> <li>○ Gestionar Datos del Proyecto.</li> </ul>	<ul style="list-style-type: none"> <li>• El administrador debió haber asignado el rol de especialista al usuario.</li> </ul>

	<ul style="list-style-type: none"> <li>○ Gestionar Métricas.</li> <li>○ Gestionar Planificaciones.</li> <li>○ Gestionar Pruebas</li> <li>○ Insertar Pruebas.</li> </ul> <p><i>No conformidades:</i></p> <ul style="list-style-type: none"> <li>○ Mostrar estados de las pruebas.</li> <li>○ Gestionar las No Conformidades.</li> <li>○ Datos de los elementos.</li> <li>○ Reporte de todos los proyectos.</li> </ul> <p><i>Seguimiento:</i></p> <ul style="list-style-type: none"> <li>○ Gestionar datos de Jefe de prueba.</li> <li>○ Generar reportes.</li> <li>○ Gestionar Cursos optativos.</li> <li>○ Verificar cumplimiento de métricas.</li> </ul>	
<ul style="list-style-type: none"> <li>• Un usuario se autentica por el Rol de Jefe de Prueba.</li> </ul>	<ul style="list-style-type: none"> <li>• Se muestra la interfaz del Jefe de prueba con sus permisos.</li> </ul> <p><i>Planificación:</i></p> <ul style="list-style-type: none"> <li>○ Gestionar Datos del Proyecto.</li> <li>○ Gestionar Planificaciones.</li> <li>○ Insertar datos del</li> </ul>	<ul style="list-style-type: none"> <li>• El administrador o el Especialista debieron haber asignado el rol de Jefe de prueba al usuario.</li> </ul>

	<p>proyecto.</p> <ul style="list-style-type: none"> <li>○ Planificar Pruebas.</li> </ul> <p><i>No conformidades:</i></p> <ul style="list-style-type: none"> <li>○ Asignar trabajo.</li> <li>○ Mostrar estado de las pruebas.</li> <li>○ Gestionar las No Conformidades.</li> <li>○ Insertar nuevo elemento de prueba.</li> <li>○ Datos de los elementos de prueba.</li> <li>○ Reportes de los proyectos en los que ha trabajado.</li> </ul> <p><i>Seguimiento:</i></p> <ul style="list-style-type: none"> <li>○ Gestionar datos de Probador.</li> <li>○ Evaluar Probador.</li> <li>○ Gestionar PC.</li> </ul>	
--	---	--

Módulo Seguimiento.

Entrada	Resultados	Condiciones
<ul style="list-style-type: none"> <li>• Se selecciona insertar Probadores.</li> </ul>	<ul style="list-style-type: none"> <li>• El sistema inserta nuevos probadores.</li> </ul>	<ul style="list-style-type: none"> <li>• Se debe introducir en el módulo de Seguimiento los siguientes datos:               <ul style="list-style-type: none"> <li>○ Se debe introducir en el módulo de Seguimiento.</li> <li>○ Nombre</li> </ul> </li> </ul>



		<ul style="list-style-type: none"> <li>○ Apellidos</li> <li>○ Sexo</li> <li>○ Teléfono</li> <li>○ Correo</li> <li>○ Apartamento</li> <li>○ Facultad</li> <li>○ Grupo</li> <li>○ Solapín</li> </ul>
<ul style="list-style-type: none"> <li>• Se selecciona insertar un Jefe de prueba.</li> </ul>	<ul style="list-style-type: none"> <li>• El sistema inserta nuevos Jefes de prueba.</li> </ul>	<ul style="list-style-type: none"> <li>• Se debe introducir en el módulo de Seguimiento los siguientes datos: <ul style="list-style-type: none"> <li>○ Nombre</li> <li>○ Apellidos</li> <li>○ Departamento</li> <li>○ Título</li> <li>○ Grado Científico</li> <li>○ Facultad</li> <li>○ Experiencia</li> <li>○ Dirección</li> <li>○ Teléfono</li> <li>○ Correo</li> <li>○ Desempeño laboral</li> <li>○ Solapín</li> </ul> </li> </ul>
<ul style="list-style-type: none"> <li>• Se selecciona verificar la veracidad de las métricas aplicadas a una prueba.</li> </ul>	<ul style="list-style-type: none"> <li>• Muestra las métricas aplicadas y el tiempo real en que se realizaron las pruebas.</li> </ul>	<ul style="list-style-type: none"> <li>• Se debe de haber insertado en el modulo planificación las pruebas con sus métricas y el sistema debió de haber registrado el tiempo real de las pruebas en el módulo No Conformidades.</li> </ul>
<ul style="list-style-type: none"> <li>• Se selecciona evaluar probador.</li> </ul>	<ul style="list-style-type: none"> <li>• Muestra la interfaz para evaluar el probador.</li> </ul>	<ul style="list-style-type: none"> <li>• Se debe de haber insertado el probador en el modulo de Seguimiento y revisar el trabajo que</li> </ul>

		hizo en el modulo No Conformidades.
<ul style="list-style-type: none"> <li>• Se selecciona gestionar PC.</li> </ul>	<ul style="list-style-type: none"> <li>• Muestra la interfaz para seleccionar la planificación.</li> </ul>	<ul style="list-style-type: none"> <li>• Se debe de haber realizado planificación en el modulo Planificación.</li> </ul>
<ul style="list-style-type: none"> <li>• Un usuario se autentica por el Rol de Especialista.</li> </ul>	<ul style="list-style-type: none"> <li>• Se muestra la interfaz del especialista con sus permisos.</li> </ul> <p><i>Planificación:</i></p> <ul style="list-style-type: none"> <li>○ Gestionar Datos del Proyecto.</li> <li>○ Gestionar Métricas.</li> <li>○ Gestionar Planificaciones.</li> <li>○ Gestionar Pruebas</li> <li>○ Insertar Pruebas.</li> </ul> <p><i>No Conformidades:</i></p> <ul style="list-style-type: none"> <li>○ Mostrar estados de las pruebas.</li> <li>○ Gestionar las No Conformidades.</li> <li>○ Datos de los elementos.</li> <li>○ Reporte de todos los proyectos.</li> </ul> <p><i>Seguimiento:</i></p> <ul style="list-style-type: none"> <li>○ Gestionar datos de</li> </ul>	<ul style="list-style-type: none"> <li>• El administrador debió haber asignado el rol de especialista al usuario.</li> </ul>

	<p>Jefe de prueba.</p> <ul style="list-style-type: none"> <li>○ Generar reportes.</li> <li>○ Gestionar Cursos optativos.</li> <li>○ Verificar cumplimiento de métricas.</li> </ul>	
<ul style="list-style-type: none"> <li>• Un usuario se autentica por el Rol de Jefe de Prueba.</li> </ul>	<ul style="list-style-type: none"> <li>• Se muestra la interfaz del Jefe de prueba con sus permisos.</li> </ul> <p><i>Planificación:</i></p> <ul style="list-style-type: none"> <li>○ Gestionar Datos del Proyecto.</li> <li>○ Gestionar Planificaciones.</li> <li>○ Insertar datos del proyecto.</li> <li>○ Planificar Pruebas.</li> </ul> <p><i>No conformidades:</i></p> <ul style="list-style-type: none"> <li>○ Asignar trabajo.</li> <li>○ Mostrar estado de las pruebas.</li> <li>○ Gestionar las No Conformidades.</li> <li>○ Insertar nuevo elemento de prueba.</li> </ul>	<ul style="list-style-type: none"> <li>• El administrador o el Especialista debieron haber asignado el rol de Jefe de prueba al usuario.</li> </ul>

	<ul style="list-style-type: none"> <li>○ Datos de los elementos de prueba.</li> <li>○ Reportes de los proyectos en los que ha trabajado.</li> </ul> <p><i>Seguimiento:</i></p> <ul style="list-style-type: none"> <li>○ Gestionar datos de Probador.</li> <li>○ Evaluar Probador.</li> <li>○ Gestionar PC.</li> </ul>	
--	---	--

**Tabla 4.1. Descripción del caso de prueba de integración.**

**Conclusiones.**

En este capítulo se abordaron los conceptos fundamentales asociados con las fases de Implementación y Prueba en el desarrollo del software. Se realizó el modelo de despliegue y los distintos modelos de componentes que dan cumplimiento a los requisitos planteados en el sistema, así como el diseño del caso de prueba de integración que demuestra la funcionalidad del sistema como un todo y la calidad de este.



## Conclusiones.



Para el desarrollo del presente trabajo de diploma se realizó un estudio del proceso de seguimiento de recursos que se efectúa en el Laboratorio de Calidad Central de la UCI a través del intercambio con especialistas de esta entidad, así como la observación y la práctica diaria, conjuntamente con la búsqueda de información que de este campo aparece en diferentes bibliografías.

Se realizó una exhaustiva búsqueda bibliográfica acerca de las últimas tendencias y tecnologías que a nivel internacional se están utilizando en el mundo de la informática, para definir aquellas que mejor respuesta darían al problema al cual se quería dar solución. Como conclusión, se optó por la utilización del PHP como lenguaje de programación, MySQL como gestor de base de datos, RUP como metodología de desarrollo de software y el Visual Paradigm como herramienta de modelado por formar parte de la amplia gama de software libre, solución de gran importancia para Cuba actualmente.

La elaboración y aplicación de una herramienta, creada a partir de la necesidad de automatizar los procesos que se llevan a cabo en el Laboratorio Central de Calidad de Software de la UCI, permitirá elevar la eficiencia en el proceso de revisión de los productos. La implementación de esta propuesta, organizada en tres módulos: Planificación, Gestión de No Conformidades y Seguimiento de los Recursos, posibilitará dar respuesta a las solicitudes de los clientes en el menor tiempo posible.

En el laboratorio Central de la UCI es de vital importancia contar con un sistema automatizado que permita administrar todos los recursos que se emplean en la realización de las diferentes pruebas.

Por tanto, se considera que se han cumplido cada uno de los objetivos trazados al comenzar este trabajo, demostrándose además con el mismo que con la creación de una plataforma uniforme y consistente es posible la gestión automatizada de los procesos que intervienen en el seguimiento de los recursos en el Laboratorio Central de Calidad de Software de la UCI.

Este Trabajo de Diploma fue presentado en la Jornada Científica Estudiantil de este año obteniendo la condición de Relevante.



## *Recomendaciones.*



1. Continuar desarrollando la línea de investigación relacionada con la calidad de software; para ampliar y actualizar los conocimientos acordes con las últimas tendencias existentes a nivel mundial y poder así hacer más eficiente este proceso en la Universidad.
2. Profundizar en el estudio de las tendencias y tecnologías actuales para determinar con mayor exactitud el basamento teórico de las nuevas herramientas a desarrollar.
3. Aplicar esta herramienta en otros laboratorios de calidad de software con el objetivo de generalizar la propuesta para elevar la eficiencia de estos procesos y evaluar las mejoras posibles para una nueva versión.
4. Diseñar a partir del sistema desarrollado y utilizando los datos que se almacenen durante un tiempo, un sistema inteligente que permita la generación automática de algunos datos y ayude a la toma de decisiones.



## Bibliografía



9001, I. (2000). "ISO 9001 Generalidades." 15/10/06, from

[http://www.usm.edu.ec/informatica/sw\\_eng/calsoft/iso%209001/iso9001.htm](http://www.usm.edu.ec/informatica/sw_eng/calsoft/iso%209001/iso9001.htm).

Aiteco, C. (2006). "Concepto de calidad." 15/12/06, from <http://www.aiteco.com/calidad>.

Albert, E. (2005). "¿Que es PHP?" 4 de noviembre de 2006, from

<http://www.ucc.queque.com/TecnologiaWeb/LenguajesWeb/PHP.htm>.

Date, C. J. (2003). Introducción a los Sistemas de Bases de Datos. Ciudad de La Habana, Editorial Félix Varela.

Desarrollo, W. (2003). "¿Qué es una tecnología?" 4 de noviembre de 2006, from

<http://www.desarrolloweb.com/manuales/15>.

Enrique., P. R. (2005). Herramienta para la estructuración y la gestión de contenidos de aplicaciones multimedia educativas. Facultad de ingeniería Industrial. Ciudad de La Habana, Instituto Superior Politécnico "José Antonio Echeverría": 128.

Héctor, G. L. A. (2005). Aplicación Web para Gestionar Antecedentes Penales. Facultad de Ingeniería Industrial. Ciudad de La Habana, Instituto Superior Politécnico José Antonio Echeverría: 188.

Henry, S. (2006). "Historia de PHP." 10 de octubre de 2006, from

<http://www.mailxmail.com/curso/informatica/phpprogramacionweb/capitulo4.htm>.

IEEE. (1990). "IEEE 610-1990." 12/12/06, from <http://dmi.uib.es/~bbuades/calidad/sld061.htm>.

ISO. (84). "Norma ISO 8402." 05/11/06, from <http://www.exit109.com/~leebee/>.

ISO. (2000). "Certificación de calidad, normas ISO." from

<http://www.ucbcba.edu.bo/orgaest/ceind/interesante.html>.

ISO. (2000). "Norma ISO 9000: 2000." 12/10/06, from <http://eprints.rclis.org/>.

James, G. G. (2005). "Taller de sistemas informáticos." 2de noviembre del 2006, from

[http://www.pcm.gob.pe/portal\\_ongei/publicaciones/cultura/Lib5038](http://www.pcm.gob.pe/portal_ongei/publicaciones/cultura/Lib5038).

Java, S. (2005). "Biblioteca de aplicaciones web: Java Script." 15 de octubre del 2006, from

[http://www.webviva.com/biblioteca/modules.php?name=News&new\\_topic=3](http://www.webviva.com/biblioteca/modules.php?name=News&new_topic=3).

- Java, S. (2005). "Tutorial de Java Script." 5 de octubre del 2006, from <http://www.unav.es/cti/manuales/TutorialJavaScript/indices/>.
- Jim, J. (2003) Historia de PHP. **Volume**, [www.desarrolloweb.com/articulos/392.php](http://www.desarrolloweb.com/articulos/392.php) DOI:
- Joaquin, G. (2005). "CMM-CMMI." 05/01/07, from <http://www.ingenierossoftware.com/calidad/cmm-cmmi-nivel-2.php>.
- José, G. V. (2005). "Tecnología Informática." 2 de noviembre del 2006, from <http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml>.
- José, G. V. A. (2003). Desarrollo Web con PHP y MySQL. Madrid.
- Luke, W. and T. Laura (2003). PHP and MySQL Web Development.
- Martín, R. M. (2004). Programación PHP. Madrid.
- Melchor, G. M. (2007). UCI, Calidad de Software.
- Patricio, S. C. and H. K. Nancy. (2005). "Tutorial de UML." 20 de octubre del 2006, from <http://www.dcc.uchile.cl/~psalinas/uml/introduccion.html>.
- Patterson, F. J., B. Brian, et al. (2003). La biblia de C#. Madrid.
- Paúl, Z. P. and O. C. T. Lilia (2006). Sistema de control de acceso a la Universidad de las Ciencias Informáticas. Ciudad de la Habana, Universidad de las Ciencias Informáticas: 106.
- PHP. (2005). "Introducción al PHP." 17 de octubre del 2006, from <http://www.ciberteca.net/webmaster/php>.
- S., P. R. (2005). Ingeniería de Software. Un enfoque práctico. Ciudad de La Habana.
- Web, D. (2005). "¿Qué es JSP?" 21 de octubre del 2006, from <http://www.desarrolloweb.com/articulos/831.php>.
- Enterprise, P. M. (2006). "Tour en línea de EPM." 17/12/06, from <http://www.projecthosts.com/tourla.asp?gclid=CNTsIKDg9YkCFTE1JAodfnlqVw>.
- Magdalena, A. U. and H. P. A. Jorge (2006). "Generación, control y seguimiento de recursos propios." 18.
- Microsoft, C. (2007). "La solución de Microsoft Office para la administración de proyectos." 10/01/07, from <http://www.microsoft.com/spain/office/products/project/default.aspx>.
- Pressman, R. S. (1998). "Ingeniería de Software." 15/11/06, from <http://www.exit109.com/~leebee/>.
- Software, G. (2006). "Sistemas de seguridad." 22/12/06, from <http://www.gscssoftware.com/default.htm>.
- Varis. (2006). "Comunicaciones subterráneas inteligentes." 21/12/06, from <http://www.varismine.com>.



Date, C. J. (2003). Introducción a los Sistemas de Bases de Datos. Ciudad de La Habana, Editorial Félix Varela.

## Anexos.

**Anexo I: Cuaderno de trabajo del probador: documento donde se registran los datos de los trabajos realizados por el probador.**

Cuaderno de Trabajo												
Trabajo	Fecha	Proceso	Estimado		Real			Hasta la fecha				
			Tiempo	Unid	Tiempo	Unid	Velocidad	Tiempo	Unid	Velocidad	Max	Min
1												
	Descripción:											
2												
	Descripción:											
3												
	Descripción:											
4												
	Descripción:											

**Legenda:**

1. Trabajo: Actividad planificada, enumera comenzado por el 1.
2. Fecha: Fecha de comienzo del trabajo.
3. Procesos: Tipo de tarea a realizar.
4. Tiempo estimado: Tiempo estimado para realizar la tarea.
5. Unidades estimadas: Unidades estimadas para el trabajo acabado.
6. Tiempo real: Tiempo total real de unidades totales.
7. Unidades reales: Número real de unidades totales.
8. Velocidad real: Tiempo real dividido por las unidades reales.
9. Tiempo hasta la fecha: Sumar el valor de Tiempo Hasta la Fecha del último trabajo realizado de este tipo.
10. Unidades hasta la fecha: Unidades Hasta la Fecha + Unidades Reales del trabajo mas reciente de este tipo.

11. Velocidad hasta la fecha: Tiempo Hasta le Fecha / Unidades Hasta la Fecha (se obtendrá minutos por unidad para todos los trabajos terminados hasta le fecha).
12. Max: velocidad máxima para todos los trabajos de cada tipo que hayan acabado.
13. Min: velocidad mínima para todos los trabajos de cada tipo que hayan acabado
14. Descripción: Descripción clara y pequeña del trabajo que se ha realizado.

**Anexo II: Planilla de control de pc: documento donde queda registrado el uso de cada pc.**

<b>Plantilla control de pc</b>			
Número de pc	Nombre de probador	Tiempo de uso	
		Hora inicio	Hora fin

Leyenda:

1. Número de pc: Especificación del número de la pc donde trabajará el probador.
2. Nombre de probador: Se especifica el nombre del probador.
3. Hora inicio: Hora en que se sentó en la pc.
4. Hora fin: Hora en que terminó de trabajar.

**Anexo III: Descripción del caso de uso “Gestionar datos de Jefe de Prueba”**

<b>Caso de uso</b>	
<b>CU 1</b>	Gestionar datos de Jefe de prueba.
<b>Propósito</b>	Insertar datos de un nuevo Jefe de prueba, buscar y visualizar los datos de un Jefe de prueba existente, modificar los datos de un Jefe de prueba existente, eliminar un Jefe de prueba, exportar los datos de un Jefe de prueba e imprimir los datos de un Jefe de prueba.
<b>Actores</b>	Especialista de calidad

<b>Resumen</b>	<p>Este caso de uso inicia cuando el especialista de calidad va a realizar alguna de las siguientes operaciones relacionadas con el Jefe de prueba:</p> <ul style="list-style-type: none"> <li>• Insertar datos de un nuevo Jefe de prueba.</li> <li>• Buscar y visualizar datos de un Jefe de prueba.</li> <li>• Modificar datos de un Jefe de prueba.</li> <li>• Eliminar datos de Jefe de prueba.</li> <li>• Exportar datos de Jefe de prueba.</li> <li>• Imprimir datos de Jefe de prueba.</li> </ul> <p>El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se emite el resultado de la operación solicitada.</p>
<b>Precondiciones</b>	El usuario debe haberse autenticado como Especialista de calidad.
<b>Poscondiciones</b>	El sistema inserta un nuevo Jefe de prueba, busca y visualiza los datos de un Jefe de prueba existente, modifica datos de un Jefe de prueba, elimina datos de un Jefe de prueba, exporta o imprime datos de un Jefe de prueba.
<b>Referencias</b>	RF 1.1, RF 1.2, RF 1.3, RF 1.4, RF 1.5, RF 1.6
<b>Flujo normal de los eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
<p>1. El Especialista de calidad quiere realizar alguna de las siguientes operaciones:</p> <ul style="list-style-type: none"> <li>• Insertar datos de un nuevo Jefe de prueba.</li> <li>• Buscar y visualizar datos de un Jefe de prueba.</li> <li>• Modificar datos de un Jefe de prueba.</li> <li>• Eliminar datos de Jefe de prueba.</li> <li>• Exportar datos de un Jefe de prueba.</li> <li>• Imprimir datos de un Jefe de prueba.</li> </ul>	<p>2. El sistema, en dependencia de la operación que solicita realizar, hace lo siguiente:</p> <ul style="list-style-type: none"> <li>• Si decide Insertar datos de un nuevo Jefe de prueba, ir a Sección <b>“Insertar datos de un nuevo Jefe de prueba”</b>.</li> <li>• Si decide Buscar y visualizar datos de un Jefe de prueba, ir a la Sección <b>“Buscar y visualizar datos de un Jefe de prueba”</b>.</li> <li>• Si decide Modificar datos de un Jefe de prueba, ir a Sección <b>“Modificar datos de un Jefe de prueba”</b>.</li> </ul>

	<ul style="list-style-type: none"> <li>• Si decide Eliminar datos de Jefe de prueba, ir a la Sección <b>“Eliminar datos de Jefe de prueba”</b>.</li> <li>• Si decide Exportar datos de Jefe de prueba, ir a la Sección <b>“Exportar datos de Jefe de prueba”</b>.</li> <li>• Si decide Imprimir datos de Jefe de prueba, ir a la Sección <b>“Imprimir datos de Jefe de prueba”</b>.</li> </ul>
<b>Sección “Insertar datos de un nuevo Jefe de prueba”.</b>	
	<p>1. El sistema muestra la interfaz correspondiente para la inserción de un nuevo Jefe de prueba.</p>
<p>2. Provee los datos necesarios para insertar un nuevo Jefe de prueba:</p> <ul style="list-style-type: none"> <li>• Nombre</li> <li>• Apellidos</li> <li>• Departamento</li> <li>• Título</li> <li>• Grado Científico</li> <li>• Facultad</li> <li>• Experiencia</li> <li>• Dirección</li> <li>• Teléfono</li> <li>• Correo</li> <li>• Desempeño laboral</li> <li>• Solapín</li> </ul>	<p>3. Verifica que los datos que introdujo fueron los necesarios para proceder a insertar un nuevo Jefe de prueba.</p>
	<p>4. Inserta los datos de un nuevo Jefe de prueba.</p>
	<p>5. Da la posibilidad de insertar un nuevo usuario.</p>
<b>Flujos alternos Sección “Insertar un nuevo Jefe de prueba”.</b>	

Acción del actor	Respuesta del sistema
	3.1 Si falta algún dato necesario para la inserción de un nuevo Jefe de prueba, emite un mensaje de error y no inserta un nuevo Jefe de prueba.
<b>Sección “Buscar y visualizar datos de un Jefe de prueba”.</b>	
	1. Muestra la interfaz correspondiente a la búsqueda de usuarios.
2. Selecciona la opción Jefe de prueba y llena el campo “Texto”.	3. Verifica que al menos un campo tenga valor.
	4. Busca los Jefes de prueba que cumplan con los parámetros introducidos.
	5. Muestra una interfaz con el resultado de la búsqueda.
	6. Da la posibilidad de realizar otra búsqueda.
<b>Flujos alternos Sección “Buscar y visualizar datos de un Jefe de prueba”</b>	
	4.1 Si no encuentra ningún Jefe de prueba que cumpla con los parámetros indicados muestra la interfaz sin ningún Jefe de prueba.
<b>Sección “Modificar datos de un Jefe de prueba”.</b>	
	1. Ejecutar los pasos 1-4 de la <b>sección “Buscar y Visualizar datos de un Jefe de prueba”</b> .
2. Oprime la opción “Modificar los datos personales del usuario”.	3. Muestra la interfaz para modificar los datos del Jefe de prueba.
4. Modifica los datos del Jefe de prueba.	5. Verifica que no se haya quedado ningún campo de información vacío.
	6. Registra la modificación en la base de datos.
<b>Flujos alternos Sección “Modificar datos de un Jefe de prueba”.</b>	

	5.1 Si quedó algún campo de información necesaria vacío, emite un mensaje de error de completitud de datos.
<b>Sección “Eliminar datos de un Jefe de prueba”.</b>	
	1. Ejecutar los pasos 1-4 de la <b>sección “Buscar y Visualizar datos de un Jefe de prueba”.</b>
2. Oprime la opción “Eliminar el usuario y todos los registros asociados a él”.	3. Muestra un mensaje de ratificación.
4. Acepta el mensaje.	5. Elimina el Jefe de prueba.
	6. Permite buscar otro Jefe de prueba para eliminarlo.
<b>Flujos alternos Sección “Eliminar datos de un Jefe de prueba”.</b>	
4.1 Cancela en mensaje. Retornar al 1.	
<b>Sección “Exportar datos de un Jefe de prueba”.</b>	
	1. Ejecutar los pasos 1-4 de la <b>sección “Buscar y Visualizar datos de un Jefe de prueba”.</b>
2. Oprime la opción “Exportar e imprimir datos de Jefe de prueba”	3. Muestra la interfaz con los datos a exportar.
4. Oprime el botón “Exportar”.	5. Muestra la interfaz del documento.
6. Selecciona el lugar donde va a guardar el documento.	7. Exporta el documento.
<b>Flujos alternos Sección “Exportar datos de un Jefe de prueba”</b>	
1.1 No desea exportar los datos de un Jefe de prueba.	
<b>Sección “Imprimir datos de un Jefe de prueba”</b>	
	1. Ejecutar los pasos 1-5 de la <b>sección “Buscar y Visualizar datos de Jefe de prueba”.</b>

2. Oprime la opción "Exportar e imprimir datos de Jefe de prueba"	3. Muestra la interfaz con los datos a imprimir.
4. Oprime la opción "Imprimir".	5. Muestra la interfaz para imprimir.
	6. Imprime los datos.
<b>Flujos alternos Sección "Imprimir datos de un Jefe de prueba"</b>	
1.1 No desea imprimir los datos del Probador y cierra dicha interfaz.	

**Anexo IV: Descripción del caso de uso "Gestionar datos del Probador".**

<b>Caso de uso</b>	
<b>CU 2</b>	Gestionar datos de Probador.
<b>Propósito</b>	Insertar datos de un nuevo Probador, buscar, visualizar, modificar y eliminar los datos de un Probador existente, además exportar e imprimir datos de un Probador.
<b>Actores</b>	Jefe de prueba.
<b>Resumen</b>	<p>Este caso de uso inicia cuando el Jefe de prueba va a realizar alguna de las siguientes operaciones relacionadas con el Probador:</p> <ul style="list-style-type: none"> <li>• Insertar datos de un nuevo Probador.</li> <li>• Buscar y visualizar datos de un Probador.</li> <li>• Modificar datos de un Probador.</li> <li>• Eliminar datos de Probador.</li> <li>• Exportar datos de Probador.</li> <li>• Imprimir datos de Probador.</li> </ul> <p>El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se emite el resultado de la operación solicitada.</p>
<b>Precondiciones</b>	El usuario debe haberse autenticado como Jefe de prueba.



<b>Poscondiciones</b>	Inserta datos de un nuevo Probador, busca, visualiza, modifica y elimina los datos de un Probador existente, además exporta e imprime datos de un Probador.
<b>Referencias</b>	RF 2.1, RF 2.2, RF 2.3, RF 2.4, RF 2.5, RF 2.6
<b>Flujo normal de los eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	<p>1. El sistema muestra la interfaz para seleccionar el Probador a gestionar:</p> <ul style="list-style-type: none"> <li>• Estudiante.</li> <li>• Trabajador.</li> </ul>
<p>1. El Jefe de prueba quiere realizar alguna de las siguientes operaciones:</p> <ul style="list-style-type: none"> <li>• Insertar datos de un nuevo probador.</li> <li>• Buscar y visualizar datos de un probador.</li> <li>• Modificar datos de un probador.</li> <li>• Eliminar datos de probador.</li> <li>• Exportar datos de un probador.</li> <li>• Imprimir datos de un probador.</li> </ul>	<p>2. El sistema, en dependencia de la operación que solicita realizar, hace lo siguiente:</p> <ul style="list-style-type: none"> <li>• Si decide Insertar datos de un nuevo probador, ir a Sección <b>“Insertar datos de un nuevo Probador”</b>.</li> <li>• Si decide Buscar y visualizar datos de un probador, ir a la Sección <b>“Buscar y visualizar datos de un Probador”</b>.</li> <li>• Si decide Modificar datos de un probador, ir a Sección <b>“Modificar datos de un Probador”</b>.</li> <li>• Si decide Eliminar datos de probador, ir a la Sección <b>“Eliminar datos de Probador”</b>.</li> <li>• Si decide Exportar datos de un probador, ir a la Sección <b>“Exportar datos de un Probador”</b>.</li> <li>• Si decide Imprimir datos de un probador, ir a la Sección <b>“Imprimir datos de un Probador”</b>.</li> </ul>
<b>Sección “Insertar datos de un nuevo Probador”.</b>	

	1. El sistema muestra la interfaz correspondiente para la inserción de los datos de un nuevo usuario.
2. Escoge una de las siguientes opciones: <ul style="list-style-type: none"> <li>• Estudiante.</li> <li>• Trabajador.</li> </ul>	3. Si escogió la opción de “Estudiante” el sistema muestra la interfaz correspondiente para insertar un estudiante.
4. Provee los datos necesarios para insertar un nuevo Probador: <ul style="list-style-type: none"> <li>• Nombre</li> <li>• Apellidos</li> <li>• Sexo</li> <li>• Teléfono</li> <li>• Correo</li> <li>• Apartamento</li> <li>• Facultad</li> <li>• Grupo</li> <li>• Solapín</li> </ul>	5. Verifica que los datos que introdujo fueron los necesarios para proceder a insertar un nuevo Probador.
	6. Inserta un nuevo Probador.
	7. Da la posibilidad de insertar un nuevo usuario.
<b>Flujos alternos Sección “Insertar datos de un nuevo Probador”.</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	3.1 Si escogió la opción de “Trabajador” el sistema muestra la interfaz correspondiente para insertar un trabajador

<p>4.1 Provee los datos necesarios para insertar un nuevo probador trabajador:</p> <ul style="list-style-type: none"> <li>• Nombre</li> <li>• Apellidos</li> <li>• Sexo</li> <li>• Teléfono</li> <li>• Correo</li> <li>• Organización</li> </ul>	<p>4.2 Va a la Acción 5 de la Sección “Insertar datos de un nuevo probador”.</p>
	<p>5.1 Si falta algún dato necesario para la inserción del nuevo Probador, emite un mensaje de error y no inserta los datos del Probador.</p>
<p><b>Sección “Buscar y visualizar datos de Probador”.</b></p>	
	<p>1. Muestra la interfaz correspondiente a la búsqueda usuarios.</p>
<p>2. Selecciona la opción “Estudiante” o “Trabajador” y llena el campo “Texto”.</p>	<p>3. Verifica que al menos un campo tenga valor.</p>
	<p>4. Busca los probadores que cumplan con los parámetros introducidos.</p>
	<p>5. Muestra el resultado de la búsqueda.</p>
	<p>6. Da la posibilidad de realizar otra búsqueda.</p>
<p><b>Flujos alternos Sección “Buscar y visualizar datos de Probador”</b></p>	
	<p>4.1 Si no encuentra ningún Probador que cumpla con los parámetros indicados muestra la interfaz sin ningún Probador.</p>
<p><b>Sección “Modificar datos de Probador”.</b></p>	
	<p>1. Ejecutar los pasos 1-5 de la <b>sección “Buscar y Visualizar datos de Probador”</b>.</p>
<p>2. Oprime la opción “Modificar datos personales del usuario”.</p>	<p>3. Muestra la interfaz para modificar los datos del Probador.</p>

4. Modifica los datos del Probador.	5. Verifica que no se haya quedado ningún campo de información obligatoria vacío.
	6. Registra la actualización en la base de datos.
<b>Flujos alternos Sección “Modificar datos de un Probador”</b>	
	5.1 Si quedó algún campo de información necesaria vacío, emite un mensaje de error de completitud de datos.
<b>Sección “Eliminar datos de Probador”.</b>	
	1. Ejecutar los pasos 1-5 de la <b>sección “Buscar y Visualizar datos de Probador”</b> .
2. Oprime la opción “Eliminar usuario y todos los registros asociados a él”.	3. Muestra un mensaje de ratificación.
4. Acepta el mensaje.	5. Elimina el probador.
	6. Permite buscar otro Probador para eliminarlo.
<b>Flujos alternos Sección “Eliminar datos de Probador”</b>	
4.1 Cancela el mensaje. Retornar al 1.	
<b>Sección “Exportar datos de Probador”.</b>	
	1. Ejecutar los pasos 1-5 de la <b>sección “Buscar y Visualizar datos de Probador”</b> .
2. Oprime la opción “Exportar e imprimir datos de Probador”	3. Muestra la interfaz con los datos a exportar.
4. Oprime el botón “Exportar”.	5. Muestra la interfaz del documento.
6. Selecciona el lugar donde va a guardar el documento.	7. Exporta el documento.
<b>Flujos alternos Sección “Exportar datos de Probador”</b>	
1.1 No desea exportar los datos del Probador cierra dicha interfaz.	
<b>Sección “Imprimir datos de Probador”</b>	

	1. Ejecutar los pasos 1-5 de la <b>sección “Buscar y Visualizar datos de Probador”</b> .
2. Oprime la opción “Exportar e imprimir datos de Probador”	3. Muestra la interfaz con los datos a imprimir.
4. Oprime la opción “Imprimir”.	5. Muestra la interfaz para imprimir.
	6. Imprime los datos.
<b>Flujos alternos Sección “Imprimir datos de probador”</b>	
1.1 No desea imprimir los datos del probador y cierra dicha interfaz.	

**Anexo V: Descripción del caso de uso “Evaluar Probador”.**

<b>Caso de uso</b>	
<b>CU 3</b>	Evaluar probador.
<b>Propósito</b>	Evaluar un probador según el trabajo su desempeñado.
<b>Actores</b>	Jefe de prueba
<b>Resumen</b>	<p>Este caso de uso se inicia cuando el Jefe de prueba revisa el trabajo realizado por el Probador, según la calidad del trabajo desempeñado por el Probador el Jefe de prueba lo evalúa de:</p> <ul style="list-style-type: none"> <li>• 0</li> <li>• 1</li> <li>• 2</li> <li>• 3</li> <li>• 4</li> <li>• 5</li> </ul> <p>El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se emite el resultado de la operación solicitada.</p>
<b>Precondiciones</b>	Que el Jefe de prueba se haya autenticado correctamente

<b>Poscondiciones</b>	El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se emite el resultado de la operación solicitada.	
<b>Referencias</b>	RF 3	
<b>Flujo normal de los eventos</b>		
<b>Acción del actor</b>	<b>Respuesta del sistema</b>	
1. El Jefe de prueba oprime la opción "Evaluar probador".	2. El sistema muestra la interfaz para buscar al Probador a evaluar.	
3. Buscar al Probador.	4. Selecciona el Probador que va evaluar.	
5. Muestra la interfaz de evaluación.	4. Evalúa al Probador de: <ul style="list-style-type: none"> <li>• 0</li> <li>• 1</li> <li>• 2</li> <li>• 3</li> <li>• 4</li> <li>• 5</li> </ul>	
5. Oprime la opción "Insertar"	6. Inserta la evaluación.	

**Anexo VI: Descripción del caso de uso "Gestionar PC".**

<b>Caso de uso</b>	
CU 4	Gestionar pc.
<b>Propósito</b>	Saber la cantidad de pc que no se utilizaron y la causa por la que no se utilizó.
<b>Actores</b>	Jefe de prueba
<b>Resumen</b>	Brindar la posibilidad al Jefe de prueba insertar el numero de pc que nos se utilizaron y la causa por la que cada pc no se utilizó.
<b>Precondiciones</b>	Que el Jefe de prueba se haya autenticado correctamente.

<b>Poscondiciones</b>	El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se emite el resultado de la operación solicitada.	
<b>Referencias</b>	RF 4.1, RF 4.2, RF 4.3	
<b>Flujo normal de los eventos</b>		
<b>Acción del actor</b>	<b>Respuesta del sistema</b>	
	1. El sistema muestra la interfaz correspondiente para gestionar las pc no utilizadas.	
2. Inserta la fecha de planificación	3. Muestra la interfaz con los datos de la planificación	
4. Inserta la cantidad de pc que no se usaron y sus causas.	5. Inserta los datos.	
<b>Flujos alternos</b>		
4.1 Si la causa por la que la pc no se puede utilizar no está predefinida selecciona "Adicionar causa".	4.2 Muestra la interfaz para adiciona la nueva causa.	
4.3 Adiciona la causa y da guardar.	4.4 Guarda la causa.	

**Anexo VII: Descripción del caso de uso "Generar Reporte".**

<b>Caso de uso</b>	
<b>CU 5</b>	Generar reportes.
<b>Propósito</b>	Generar reportes, exportar los reportes e imprimir los reportes.
<b>Actores</b>	Gestor de reportes.
<b>Resumen</b>	5
<b>Precondiciones</b>	Gestor de reportes se hayan autenticado en la aplicación como Especialista de calidad y el Jefe de prueba.
<b>Poscondiciones</b>	El sistema genera reporte, exporta o imprime reporte.
<b>Referencias</b>	RF 5.1, RF 5.2, RF 5.3

<b>Flujo normal de los eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	1. El sistema muestra la interfaz correspondiente para generar reporte de seguimiento con las opciones: <ul style="list-style-type: none"> <li>• Recursos humanos.</li> <li>• Recursos tecnológicos.</li> <li>• Reporte No conformidades.</li> <li>• Reporte general.</li> </ul>
2. El actor escoge la acción a realizar.	3. El sistema, en dependencia de la operación que solicita realizar, hace lo siguiente: <ul style="list-style-type: none"> <li>• Si decide Generar reporte de seguimiento de recursos humanos, ir a Sección <b>“Recursos humanos”</b>.</li> <li>• Si decide Generar reporte de seguimiento de recursos tecnológicos, ir a Sección <b>“Recursos tecnológicos”</b>.</li> <li>• Si decide Generar reporte general de proyecto, ir a la Sección <b>“Reporte de no conformidades”</b>.</li> <li>• Si decide Generar reporte general de proyecto, ir a la Sección <b>“Reporte general”</b>.</li> </ul>
<b>Sección “Recursos humanos”.</b>	
	1. Muestra la interfaz correspondiente para Generar reporte de seguimiento de recursos humanos.
2. Introduce el o los parámetros por los cuales desea realizar el reporte de recursos humanos: <ul style="list-style-type: none"> <li>• Nombre del proyecto</li> <li>• Fecha inicio</li> </ul>	3. Muestra la interfaz con las personas según los introducidos en el criterio de búsqueda y con las opciones de “Exportar” e “Imprimir”.



<ul style="list-style-type: none"> <li>• Fecha fin</li> <li>• Iteración</li> </ul>	
<p>4. Selecciona la actividad a realizar:</p> <ul style="list-style-type: none"> <li>• Exportar.</li> <li>• Imprimirlo.</li> </ul>	<p>5. El sistema, en dependencia de la operación que solicita realizar, hace lo siguiente:</p> <ul style="list-style-type: none"> <li>• Si se escoge la opción “Exportar”, el sistema exporta el reporte.</li> <li>• Si escoge “Imprimir” el sistema imprime el reporte.</li> </ul>
<b>Flujos alternos Sección “Recursos humanos”.</b>	
<p>4.1 Si no desea imprimir, ni exportar el reporte sale de la sección o puede generar otro reporte.</p>	
<b>Sección “Recursos tecnológicos”.</b>	
	<p>1. Muestra la interfaz correspondiente para Generar reporte de recursos tecnológicos.</p>
<p>2. Introduce el o los parámetros por los cuales desea realizar el reporte de recursos tecnológicos:</p> <ul style="list-style-type: none"> <li>• Nombre Proyecto.</li> <li>• Tipo de prueba.</li> <li>• Iteración</li> </ul>	<p>3. Muestra el reporte de acuerdo al criterio de búsqueda introducido y da la posibilidad de “Exportar” e “Imprimir” en reporte.</p>
<p>4. Selecciona la actividad a realizar:</p> <ul style="list-style-type: none"> <li>• Exportar</li> <li>• Imprimir</li> </ul>	<p>5. El sistema, en dependencia de la operación que solicita realizar, hace lo siguiente:</p> <ul style="list-style-type: none"> <li>• Si se escoge la opción “Exportar”, el sistema exporta el reporte.</li> <li>• Si escoge “Imprimir” el sistema imprime el reporte.</li> </ul>
<b>Flujos alternos Sección “Recursos tecnológicos”.</b>	
<p>4.1 Si no desea exportar o imprimir el reporte sale de la sección o puede generar otro reporte.</p>	
<b>Sección “Reporte de no conformidades”.</b>	

	1. Muestra la interfaz correspondiente para Generar reporte de no conformidades.
2. Introduce el o los parámetros por los cuales desea realizar el reporte de no conformidades: <ul style="list-style-type: none"> <li>• Fecha inicio</li> <li>• Fecha fin</li> <li>• Nombre Proyecto</li> </ul>	3. Muestra el reporte de acuerdo al criterio de búsqueda introducido y da la posibilidad de “Exportar” e “Imprimir” en reporte.
4. Selecciona la actividad a realizar: <ul style="list-style-type: none"> <li>• Exportar</li> <li>• Imprimir</li> </ul>	5. El sistema, en dependencia de la operación que solicita realizar, hace lo siguiente: <ul style="list-style-type: none"> <li>• Si se escoge la opción “Exportar”, el sistema exporta el reporte.</li> <li>• Si escoge “Imprimir” el sistema imprime el reporte.</li> </ul>
<b>Flujos alternos Sección “Reporte de no conformidades”.</b>	
4.1 Si no desea exportar o imprimir el reporte sale de la sección o puede generar otro reporte.	
<b>Sección “Reporte general”.</b>	
	1. Muestra la interfaz correspondiente para seleccionar los datos.
2. Introduce el o los parámetros por los cuales desea realizar el reporte general de proyecto: <ul style="list-style-type: none"> <li>• Nombre del Proyecto</li> <li>• Tipo de prueba</li> <li>• Iteración</li> </ul>	3. Muestra la interfaz con el reporte según los datos seleccionados y con las opciones de “Exportar” e “Imprimir”.
4. Selecciona la actividad a realizar: <ul style="list-style-type: none"> <li>• Exportar.</li> <li>• Imprimirlo.</li> </ul>	5. El sistema, en dependencia de la operación que solicita realizar, hace lo siguiente: <ul style="list-style-type: none"> <li>• Si se escoge la opción “Exportar”, el sistema exporta el reporte.</li> <li>• Si escoge “Imprimir” el sistema imprime el</li> </ul>

	reporte.
<b>Flujos alternos Sección “Reporte general”.</b>	
4.1 Si no desea exportar o imprimir el reporte sale de la sección o puede generar otro reporte.	

**Anexo VIII: Descripción del caso de uso “Verificar cumplimiento de Métrica”.**

Caso de uso	
CU 6	Verificar cumplimiento de métrica.
<b>Propósito</b>	Verificar cumplimiento de las métricas planificadas.
<b>Actores</b>	Gestor de reportes.
<b>Resumen</b>	Este caso de uso se inicia cuando el Jefe de prueba o el Especialista de calidad desean ver el cumplimiento de las métricas planificadas. El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se emite el resultado de la operación solicitada.
<b>Precondiciones</b>	Que el Jefe de prueba y el Especialista de calidad se hayan autenticado correctamente.
<b>Poscondiciones</b>	El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se emite el resultado de la operación solicitada.
<b>Referencias</b>	RF 6.1, RF 6.2
Flujo normal de los eventos	
Acción del actor	Respuesta del sistema
	1. El sistema muestra la interfaz para verificar el cumplimiento de las métricas.
2. Selecciona el proyecto al cual quiere ver las métricas.	3. Muestra una interfaz con el Id del módulo y el nombre del módulo del proyecto escogido en forma de vínculo.

4. Oprime unos de estos 2 vínculos.	5. Muestra la interfaz con las métricas correspondientes.
<b>Flujos alternos</b>	
	5.1 Muestra la interfaz en blanco si no se encontró nada.

**Anexo IX: Descripción del caso de uso “Gestionar cursos optativos”.**

Caso de uso	
<b>CU 7</b>	Gestionar cursos optativos.
<b>Propósito</b>	Adicionar Curso optativo, asignar Curso optativo.
<b>Actores</b>	Gestor de reportes.
<b>Resumen</b>	Este caso de usos comienza cuando un Especialista de calidad o un Jefe de prueba desean adicionar o asignar un curso optativo. El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se emite el resultado de la operación solicitada.
<b>Precondiciones</b>	Gestor de reportes se hayan autenticado en la aplicación como Especialista de calidad y el Jefe de prueba.
<b>Poscondiciones</b>	El sistema adiciona curso optativo y asigna curso optativo.
<b>Referencias</b>	RF 7.1, RF 7.2
Flujo normal de los eventos	
Acción del actor	Respuesta del sistema
1. El gestor de reportes quiere realizar una de las siguientes operaciones: <ul style="list-style-type: none"> <li>• Crear Curso optativo.</li> <li>• Asignar curso optativo.</li> </ul>	2. El sistema, en dependencia de la operación que solicita realizar, hace lo siguiente: <ul style="list-style-type: none"> <li>• Si decide adicionar curso optativo, ir a Sección “<b>Crear Curso optativo</b>”.</li> <li>• Si decide asignar curso optativo, ir a Sección “<b>Asignar Curso optativo</b>”.</li> </ul>

<b>Sección “Crear Curso optativo”.</b>	
	1. Muestra la interfaz correspondiente para Adicionar un curso optativo.
2. Introduce los datos del curso optativo: <ul style="list-style-type: none"> <li>• Nombre.</li> <li>• Descripción.</li> </ul>	3. Verifica que los datos que introdujo fueron los necesarios para proceder a adicionar un curso optativo.
	4. Adiciona un nuevo curso optativo.
	5. Da la posibilidad de insertar un nuevo curso optativo.
<b>Flujos alternos Sección “Adicionar curso optativo”.</b>	
	3.1 Si falta algún dato necesario para la inserción de un nuevo curso optativo, emite un mensaje de error y no inserta un nuevo curso optativo.
<b>Sección “Asignar curso optativo”.</b>	
	1. Muestra la interfaz correspondiente para seleccionar a quien se le va a asignar un curso optativo.
2. Escoge a quien le va asignar un curso optativo según los criterios de búsqueda: <ul style="list-style-type: none"> <li>• Categoría</li> <li>• Nombre</li> <li>• Facultad</li> <li>• Apartamento</li> <li>• Organización</li> </ul>	3. Muestra el resultado de la búsqueda.
4. Selecciona la persona al que le va a agregar el curso optativo.	5. Muestra la interfaz para asignar los cursos disponibles.
6. Selecciona el curso a asignar y la nota correspondiente y oprime la opción “Insertar”.	7. Inserta el curso y la nota.

	8. Da la posibilidad de asignar otro curso a otra persona.
<b>Flujo alterno sección “Asignar curso optativo”.</b>	
	3.1 Si no hay resultados de la búsqueda muestra la interfaz sin datos.

**Anexo X: Descripción del caso de uso “Gestionar Usuarios”**

<b>Caso de uso</b>	
<b>CU 8</b>	Gestionar usuarios.
<b>Propósito</b>	Crear un nuevo usuario, editar datos de usuario, asignar roles a usuarios, restringir roles a usuarios y eliminar usuarios.
<b>Actores</b>	Administrador
<b>Resumen</b>	<p>El caso de uso comienza cuando el Administrador va a realizar alguna de las siguientes operaciones:</p> <ul style="list-style-type: none"> <li>• Crear un nuevo usuario.</li> <li>• Editar datos de usuario.</li> <li>• Buscar y visualizar datos de usuario.</li> <li>• Asignar roles a usuarios.</li> <li>• Restringir roles a usuarios.</li> <li>• Eliminar usuario.</li> </ul> <p>El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se emite el resultado de la operación solicitada.</p>
<b>Precondiciones</b>	Que el Administrador se haya autenticado en el sistema.
<b>Poscondiciones</b>	El sistema crea un nuevo usuario, edita datos de usuario, asigna roles a usuarios, restringe roles a usuarios y elimina usuarios
<b>Referencias</b>	RF 8.1, RF 8.2, RF 8.3, RF 8.4, RF 8.5
<b>Flujo normal de los eventos</b>	

Acción del actor	Respuesta del sistema
<p>1. El Administrador quiere realizar alguna de las operaciones siguientes:</p> <ul style="list-style-type: none"> <li>• Crear un nuevo usuario.</li> <li>• Editar datos de usuario.</li> <li>• Asignar roles a usuarios.</li> <li>• Restringir roles a usuarios.</li> <li>• Eliminar usuario.</li> </ul>	<p>2. El sistema en dependencia de la operación que solicita el Administrador hace lo siguiente:</p> <ul style="list-style-type: none"> <li>• Si desea Crear un nuevo usuario ir a la Sección <b>“Crear un nuevo usuario”</b>.</li> <li>• Si desea Editar datos de usuario ir a la Sección <b>“Editar datos de usuario”</b>.</li> <li>• Si desea Buscar datos de un usuario ir a la sección <b>“Buscar y visualizar datos de un usuario”</b>.</li> <li>• Si desea Asignar roles a usuarios ir a la Sección <b>“Asignar roles a usuarios”</b>.</li> <li>• Si desea Restringir roles a usuarios ir a la Sección <b>“Restringir roles a usuarios”</b>.</li> <li>• Si desea Eliminar usuario ir a la Sección <b>“Eliminar usuario”</b>.</li> </ul>
<b>Sección “Crear un nuevo usuario”.</b>	
	<p>1. Muestra la interfaz correspondiente para la creación de un nuevo usuario.</p>
<p>2. Provee los datos necesarios para crear un nuevo usuario:</p> <ul style="list-style-type: none"> <li>• Usuario</li> <li>• Contraseña</li> <li>• Rol</li> </ul>	<p>3. Verifica que los datos que introdujo fueron los necesarios para proceder a crear el nuevo usuario:</p> <ul style="list-style-type: none"> <li>• Usuario</li> <li>• Contraseña</li> <li>• Rol</li> </ul>
	<p>4. Crea el nuevo</p>
	<p>5. Da la posibilidad de crear otro usuario.</p>
<b>Flujos alternos Sección “Crear un nuevo usuario”.</b>	
	<p>3.1 Si falta algún dato necesario para la creación del nuevo usuario, emite un mensaje de error y no crea</p>

	el usuario.
<b>Sección “Buscar y visualizar datos de usuarios”.</b>	
	1. Muestra la interfaz correspondiente a la búsqueda de usuarios.
2. Introduce los parámetros necesarios para la búsqueda de un usuario. <ul style="list-style-type: none"> <li>• Tipo de usuario</li> <li>• Texto</li> </ul>	3. Verifica que al menos un campo tenga valor.
	4. Busca los usuarios que cumplan con los parámetros introducidos.
	5. Muestra los resultados de la búsqueda.
<b>Flujos alternos Sección “Buscar y visualizar datos de usuarios”.</b>	
	3.1 Si no existe al menos un parámetro con valor entonces emite un mensaje de error, “Introducir al menos un parámetro”.
	4.1 Si no encuentra ningún usuario que cumpla con los parámetros indicados entonces muestra un mensaje, “No existe el usuario solicitado”.
<b>Sección “Editar datos de usuario”.</b>	
	1. Realiza los pasos del 1 al 5 de la <b>Sección “Buscar y visualizar datos de usuarios”.</b>
2. Oprime el botón “Editar usuario”.	3. Muestra la interfaz correspondiente para la edición de datos de un usuario.
4. Provee los datos necesarios para editar datos de usuario: <ul style="list-style-type: none"> <li>• Usuario</li> <li>• Contraseña</li> </ul>	5. Verifica que los datos que introdujo fueron los necesarios para proceder a editar los datos de usuario: <ul style="list-style-type: none"> <li>• Usuario</li> <li>• Contraseña</li> </ul>
	6. Edita los datos de usuario y emite un mensaje de



	operación satisfactoria.
	7. Da la posibilidad de editar los datos de otro usuario.
<b>Flujos alternos Sección “Editar datos de usuario”.</b>	
	1. 1 En caso de no existir el usuario especificado se muestra un mensaje de información “No se encontró el usuario”
	5.1 Si falta algún dato necesario para editar los datos de usuario, emite un mensaje de error y no edita los datos de usuario.
<b>Sección “Asignar roles a usuarios”.</b>	
	1. Ejecuta los pasos del 1 al 4 de la <b>sección “Editar datos de usuario”.</b>
	2. El sistema brinda la posibilidad de asignar los siguientes roles a los usuarios: <ul style="list-style-type: none"> <li>• Jefe de Prueba</li> <li>• Especialista de calidad</li> <li>• Desarrollador</li> <li>• Probador</li> <li>• Administrador</li> </ul>
2. Selecciona el rol que desempeñara el usuario. <ul style="list-style-type: none"> <li>• Jefe de Prueba</li> <li>• Especialista de calidad</li> <li>• Desarrollador</li> <li>• Probador</li> <li>• Administrador</li> </ul>	3. El sistema asigna el rol seleccionado.
<b>Sección “Restringir roles a usuarios”.</b>	
	1. Ejecuta los pasos del 1 al 3 de la <b>sección</b>

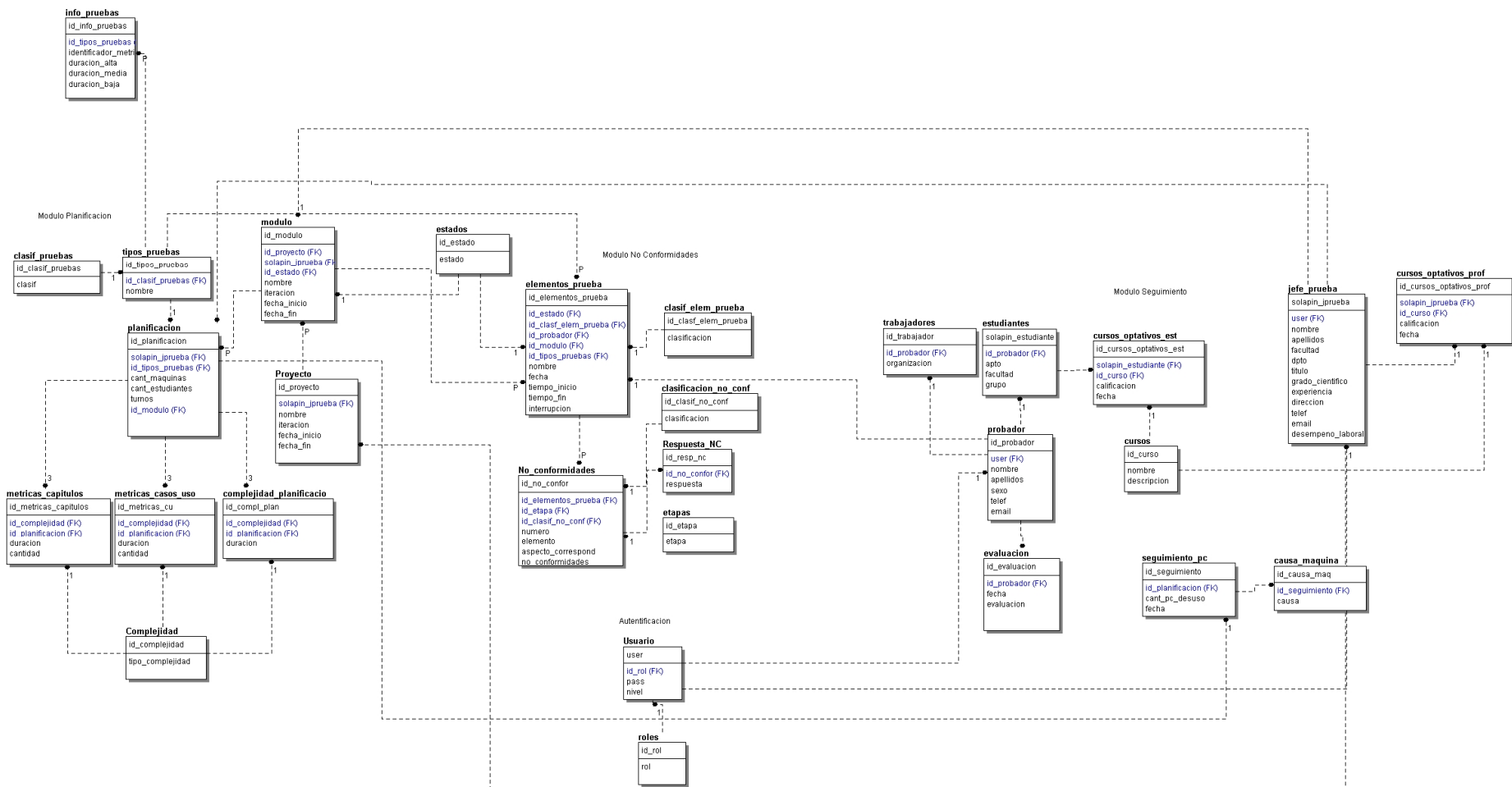
	<b>“Asignar roles a usuarios”.</b>
	2. El sistema da la posibilidad de asignara los roles: <ul style="list-style-type: none"> <li>• Jefe de Prueba</li> <li>• Especialista de calidad</li> <li>• Desarrollador</li> <li>• Probador</li> <li>• Administrador</li> </ul>
3. Restringe el rol al usuario de: <ul style="list-style-type: none"> <li>• Jefe de Prueba</li> <li>• Especialista de calidad</li> <li>• Desarrollador</li> <li>• Probador</li> <li>• Administrador</li> </ul>	4. Guarda la información.
<b>Sección “Eliminar usuarios”.</b>	
	1. Ejecuta los pasos del 1 al 7 de la <b>sección “Buscar y visualizar datos de usuario”.</b>
2. El administrador selecciona el usuario que desea borrar.	3. Muestra un mensaje de ratificación “Esta seguro que desea eliminar el usuario”
4. Acepta el mensaje.	5. El sistema elimina el usuario seleccionado.
<b>Flujos alternos sección “Eliminar usuarios”.</b>	
4.1 Cancela el mensaje.	

**Anexo XI: Descripción del caso de uso “Autenticarse”**

<b>Caso de uso</b>	
CU 8	Autenticarse.
<b>Propósito</b>	Permitir autenticarse.
<b>Actores</b>	Usuario

<b>Resumen</b>	El Caso de Uso se inicia cuando el usuario introduce los datos que se le piden para acceder a la aplicación, estos se verifican y finaliza dándole los permisos y habilitándole la entrada.	
<b>Precondiciones</b>	El usuario debe estar registrado.	
<b>Poscondiciones</b>	Se habilitan las funcionalidades según los privilegios.	
<b>Referencias</b>	RF 8	
<b>Flujo normal de los eventos</b>		
<b>Acción del actor</b>	<b>Respuesta del sistema</b>	
1. El usuario introduce nombre de usuario y contraseña.	2. El sistema encripta la contraseña.	
	3. El sistema desencripta la contraseña.	
	4. Busca el usuario y compara la contraseña.	
	5. En caso de ser correcta la contraseña, se le asignan los permisos al usuario y este entra al sistema.	
<b>Flujos alternos</b>		
5.1 En caso de no coincidir la contraseña se le envía un mensaje de: "Contraseña incorrecta".		

## Anexo XII: Base de datos General.





## Glosario de Términos.



**Arquitectura de software:** Es la estructuración del software en capas o módulos dentro de una computadora.

**Componente:** Es un conjunto o bloque de software que proporciona por si mismo o en conjunción con otros componentes una función o servicio único y puede ser re-utilizado para construir diversos sistemas.

**Desarrollo de software basado en componentes:** Es el paradigma de ensamblar componentes y escribir código para hacer que estos componentes funcionen.

**URI (Uniform Resource Identifier):** Es una cadena de caracteres compacta utilizada para identificar o nombrar un recurso. El principal propósito de esta identificación es habilitar la interacción con representaciones del recurso sobre una red, típicamente la WWW, utilizando protocolos específicos.

**Servicio:** Es una unidad de trabajo que desempeña un proveedor de servicios para satisfacer las necesidades (resultado esperado) de un consumidor de servicios.

**Servicio Web (WS):** Es un sistema de software identificado por una URI, cuyas interfaces y enlaces (bindings) son definidas y descritas usando XML. Este sistema puede ser “descubierto” y usado por otros sistemas de software. Esa interacción debe darse según la manera descrita en la definición, usando mensajes de acuerdo a protocolos de Internet.

**SOAP (Simple Object Access Protocol):** Es un protocolo estándar que define como dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. SOAP es uno de los protocolos utilizados en los servicios Web.

**PHP:** (Hypertext PreProcessor) Lenguaje de programación para el desarrollo de web dinámicas, con sintaxis parecida a la C. Originalmente se conocía como *Personal Hombre Page tools*, herramientas para páginas personales (en Internet).

**HTML: HyperText Markup Language** (lenguaje de marcas hipertextuales), lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.

**Herramientas CASE:** Las Herramientas CASE (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas ayudan en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, calculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.