



**Universidad de las Ciencias Informáticas**

**Facultad 6**

**Herramienta Informática para automatizar los procesos en el  
Laboratorio de Calidad: Módulo Planificación.**



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

**Autores:**

**Roig Calzadilla Díaz.**

**Delvis Echeverría Perez.**

**Tutores:**

**Ing. Irina Napal Torres.**

**Ing. Michael González Jorrín.**

**Ciudad de La Habana, Cuba.**

**Julio de 2007.**

## DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Dirección de Calidad 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 \_\_\_\_\_.

\_\_\_\_\_  
Firma del Autor

**Delvis Echeverría Perez**

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

**Irina Napal Torres**

\_\_\_\_\_  
Firma del Autor

**Roig Calzadilla Díaz.**

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

**Michael Gonzáles Jorrín.**

La calidad nunca es un accidente; siempre es el resultado de un esfuerzo de la inteligencia.

John Ruskin.

**Agradecimientos.**

Agradecemos especialmente a nuestros padres Julita, Xiomara y Roger, sin su dedicación este momento no hubiera sido posible.

A nuestras familias por brindarnos su apoyo.

A nuestra querida amiga Ana Margarita Abreu por siempre estar presente.

A nuestros compañeros de tesis por compartir buenos momentos.

A nuestros amigos de la Universidad Yetel, Dailien, Asnier, Martha, Milena por soportarnos y querernos.

A Michael e Irina por enseñarnos, gran parte de nuestros conocimientos se lo debemos a ellos.

A Aylin Febles por permitirnos formar parte de este proyecto.

A nuestra Decana Yanet por brindarnos su apoyo y amistad.

A nuestra revolución y a nuestro comandante por formarnos como hombres de ciencia y de futuro.

Y a nuestra Universidad por ser nuestra casa durante estos 5 años.

**Dedicatoria.**

Roig Calzadilla Díaz:

Dedico este trabajo a mis padres Xiomara Díaz Ricardo y Juan Roger Calzadilla Martínez, las personas más especiales en mi vida, porque desde el 4 de agosto de 1983 dejaron de vivir para ellos para comenzar a vivir para mí, doy gracias a Dios por tenerlos.

A mis abuelas Rosalía Martínez y Ramona Ricardo en especial mis abuelos Isidro Díaz y a Rogelio Calzadilla por su preocupación y cariño.

A toda mi familia en general pero de manera especial a mi tía Irma, mis tíos José y Ramón, mi prima Arianna, mi primo Manolito, mi gorda Yadira y a mi hermano mayor Jorge.

Delvis Echeverría Perez:

A mi mamá por su amor, dedicación, por ser la persona más importante en mi vida, por demostrarme que no hace falta un padre para formar un hombre.

A mis abuelos Cuca y Julio, por ser este momento, uno de los más importante en sus vidas, y por estar siempre presentes.

A mi tía Maritza que ha sabido hacer de sus dos sobrinos, dos hijos.

A Andy y Meiby por ser “los locos” que más quiero en el mundo.

## Resumen

La investigación está asociada al surgimiento del Laboratorio Central de Certificación de Software en la Universidad de las Ciencias Informáticas, que se espera sea el Centro Nacional de Calidad de Software Cubano. Se pretende dar respuesta a la siguiente interrogante: ¿Cómo definir y automatizar el proceso de planificación y estimación de los recursos en el Laboratorio de Calidad de software de la UCI? Para el logro de este objetivo se implementó una herramienta informática que automatice el proceso de Planificación y Estimación, diseñando una aplicación Web que sea capaz de planificar con los recursos introducidos por los jefes de pruebas. Esta aplicación permitirá, interactuar a través de la red con la información almacenada en la aplicación. Para garantizar la portabilidad de la aplicación se desarrollará bajo ambientes multiplataforma, fundamentalmente con herramientas de software libre.

## Índice

Introducción .....	1
Capítulo 1 Fundamentación Teórica.....	4
1.1 Calidad de Software.....	4
1.2 Planificación y estimación de recursos en un laboratorio de calidad.....	5
1.2.1 Definición del proceso de Planificación.....	6
1.3 Herramientas afines .....	14
1.3.1 Swag .....	14
1.3.2 Gestor de proyectos .....	14
1.3.3 Project KickStart.....	15
1.4 Tendencias y Tecnologías Actuales.....	15
1.4.1 Proceso de desarrollo de software.....	15
1.4.2 Lenguaje de modelado.....	18
1.4.3 Visual Paradigm .....	18
1.4.4 La Arquitectura Cliente-Servidor .....	19
1.4.5 Apache: Servidor Web .....	20
1.4.6 MySQL .....	21
1.4.7 SQL Server.....	22
1.4.8 ¿ Por qué MySQL?.....	22
1.4.9 PHP 5 (Hypertext Preprocessor).....	23
1.4.10 Java Script.....	23
1.4.11 HTML.....	24
1.4.12 ASP.net .....	24
1.4.13 ¿Por qué PHP? .....	25
1.4.14 DreamWeaver 8 .....	25
1.4.15 WAMP 5 .....	25
Capítulo 2 Características del sistema. ....	27
2.1 Objeto de Estudio.....	27
2.1.1 Problema y situación problemática.....	27
2.1.2 Objeto de automatización.....	28
2.1.3 Información que se maneja.....	28
2.1.4 Propuesta de sistema.....	30
2.2 Modelo de negocio.....	31
2.2.1 Descripción del Negocio.....	31
2.2.2 Actores y trabajadores del negocio.....	31
2.2.3 Diagrama Casos de Uso del Negocio.....	32
2.2.4 Diagrama de actividades. Caso de Uso Planificar Pruebas.....	34
2.3 Diagrama de Clases del Modelo de Objetos.....	36
2.4 Especificación de los requisitos de software.....	37
2.4.1 Requerimientos Funcionales.....	37
2.5.2 Requerimientos No Funcionales.....	45
2.6 Definición de los Casos de Uso .....	46

---

2.6.1	Definición de los actores. ....	46
2.6.2	Listado de Casos de Uso. ....	46
2.7	Diagrama de Casos de Uso del Sistema. ....	48
2.8	Casos de Uso por ciclo. ....	48
2.9	Descripción de CUS. ....	50
Capítulo 3	Análisis y Diseño del sistema. ....	67
3.1	Análisis. ....	67
3.1.1	Modelo de análisis. ....	67
3.1.2	Diagrama de clases de análisis. ....	67
3.2	Diseño. ....	69
3.2.1	Arquitectura Web de 3 niveles. ....	70
3.2.2	Diagrama de secuencia del diseño. ....	71
3.2.3	Diagrama de clases del diseño. ....	77
3.3	Diseño de la Base de Datos. ....	80
3.3.1	Diagrama Entidad Relación. ....	80
3.4	Definiciones de diseño que se aplican. ....	81
3.5	Tratamiento de errores. ....	81
3.6	Seguridad. ....	82
3.7	Concepción de la ayuda. ....	83
Capítulo 4	Implementación y Prueba. ....	85
4.1	Implementación. ....	85
4.1.1	Diagrama de despliegue. ....	86
4.1.2	Diagramas de componentes. ....	87
4.2	Modelo de Prueba. ....	90
4.2.1	Descripción caso de prueba de integración. ....	90
Conclusiones	.....	100
Recomendaciones	.....	101
Referencia Bibliográfica	.....	102
Bibliografía	.....	104
Glosario de términos	.....	106
Anexos	.....	107
Anexo 1	Definición de los Casos de Uso del Sistema. ....	107
Anexo 2	Descripción de los Casos de uso Gestionar Usuario y Autenticar Usuario. ....	111
Anexo 3	Diagrama de clases persistentes. ....	117
Anexo 4	Descripción de las tablas de la base de datos. ....	118



## Índice de Tablas.

Tabla 1.1 Relación de actividades en el proceso.....	7
Tabla 2.1 Actores del negocio. ....	32
Tabla 2.2 Trabajadores del negocio.....	32
Tabla 2.3 Definición de Actores. ....	46
Tabla 2.4 Listado de Casos de Uso del Sistema. ....	47
Tabla 2.5 Casos de Uso por ciclo. ....	50
Tabla 2.6 Descripción Caso de Uso Gestionar Prueba. ....	53
Tabla 2.7 Descripción Caso de Uso Insertar datos del proyecto. ....	55
Tabla 2.8 Descripción Caso de Uso Gestionar datos del proyecto.....	57
Tabla 2.9 Descripción Caso de Uso Gestionar Métricas. ....	59
Tabla 2.10 Descripción Caso de Uso Planificar Pruebas. ....	62
Tabla 2.11 Descripción Caso de Uso Gestionar Planificaciones. ....	66
Tabla 4.1 Caso de Prueba de Integración para el módulo Planificación. ....	93
Tabla 4.2 Caso de Prueba de Integración para el módulo No Conformidades.....	95
Tabla 4.3 Caso de Prueba de Integración para el módulo Seguimiento.....	99

## Índice de Figuras.

Figura 1.1 Esquema del proceso. ....	8
Figura 2.1 Diagrama Caso de Uso del negocio. ....	33
Figura 2.2 Diagrama de Actividades Caso de Uso Planificar Pruebas. ....	35
Figura 2.3 Diagrama de Clases del modelo de Objetos ....	36
Figura 2.4 Diagrama de Casos de Uso del sistema.....	48
Figura 3.1 Diagrama de clases del análisis del Caso de Uso Gestionar Prueba.....	68
Figura 3.2 Diagrama de clases del análisis del Caso de Uso Planificar Pruebas.....	68
Figura 3.3 Diagrama de clases del análisis del Caso de Uso Gestiona Planificaciones.....	69
Figura 3.4 Diagrama de secuencia del diseño, escenario del Caso de Uso Gestionar Prueba (Insertar Prueba). ....	71
Figura 3.5 Diagrama de secuencia del diseño, escenario del Caso de Uso Gestionar Prueba (Modificar Prueba). ....	72
Figura 3.6 Diagrama de secuencia del diseño, escenario del Caso de Uso Gestionar Prueba (Eliminar Prueba). ....	72
Figura 3.7 Diagrama de secuencia del diseño, Caso de Uso Planificar Prueba.....	73
Figura 3.8 Diagrama de secuencia del diseño, escenario del Caso de Uso Gestionar Planificaciones (Exportar e imprimir planificaciones). ....	74
Figura 3.9 Diagrama de secuencia del diseño, escenario del Caso de Uso Gestionar Planificaciones (Modificar planificaciones). ....	75
Figura 3.10 Diagrama de secuencia del diseño, escenario del Caso de Uso Gestionar Planificaciones (Eliminar planificaciones). ....	76
Figura 3.11 Diagrama de clases del diseño Caso de Uso Gestionar Prueba.....	77
Figura 3.12 Diagrama de clases del diseño Caso de Uso Planificar Prueba.....	78
Figura 3.13 Diagrama de clases del diseño Caso de Uso Gestionar Planificación. ....	79
Figura 3.14 Diagrama de Entidad Relación. ....	80
Figura 3.15 Iconos eliminar y modificar del sistema. ....	81
Figura 3.16 Mensaje de verificación. Eliminar Proyecto. ....	82
Figura 3.17 Mensaje de validación. Campo Nombre al modificar una Prueba. ....	82
Figura 3.18 Mensaje de validación. Campo Métricas al modificar el valor. ....	82
Figura 3.19 Mensaje de verificación. Campo Contraseñas al crear un usuario.....	83
Figura 3.20 Interfaz de autenticación. Al introducir usuario y contraseña incorrecta.....	83
Figura 4.1 Diagrama de despliegue. ....	86
Figura 4.2 Diagrama de componente Caso de Uso Gestionar Prueba.....	87
Figura 4.3 Diagrama de componente Caso de Uso Planificar Prueba. ....	88
Figura 4.4 Diagrama de componente Caso de Uso Gestionar Planificación. ....	89

## Introducción

Uno de los problemas que afronta actualmente la industria del software es la calidad de sus productos. Desde la década del 70, este tema ha sido motivo de preocupación para especialistas, ingenieros, investigadores y comercializadores de software alrededor del mundo, los cuales han realizado investigaciones entorno a estos objetivos fundamentales:

¿Cómo lograr un proceso de desarrollo de software con calidad? ¿Cómo evaluar la calidad del proceso de desarrollo del software?

La calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad.

Existen diferentes definiciones que han dado varias organizaciones y personalidades. Según la IEEE: “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”.

El problema de la calidad del software ha tomado dos grandes vertientes: Una ve al software como un producto terminado que debe cumplir con una serie de especificaciones claras, prácticamente dictadas desde su concepción, por ejemplo la norma ISO 9126 señala seis categorías de criterios de calidad que el producto terminado debe poseer. La otra vertiente es garantizar la calidad del software vigilando su proceso de fabricación, lo que presta especial atención a las normas, procedimientos, metodologías y habilidades necesarias, por ejemplo el CMMI señala cinco categorías en que es posible clasificar a un área de desarrollo de software, en función del grado de madurez que tiene en el manejo de sus procesos de manufactura.

Con el crecimiento de la industria de software cubana, el tema de la calidad se hace obligatorio en nuestras empresas que comercializan aplicaciones de software, algunas de ellas tienen sus propios sistemas de garantizar la calidad de sus productos internamente. Al surgir la Universidad de las Ciencias Informáticas aumenta en un gran por ciento la producción de software a nivel nacional y en medio de este desarrollo se observó que la calidad de sus productos no era la mejor, por lo que se llegó a la conclusión que era necesario la

creación de laboratorios de calidad de software, en cada una de sus facultades, dentro de la principales actividades en los laboratorios de calidad de software está el proceso de pruebas a las aplicaciones de software, además actualmente se trabaja en la definición de estos procesos y en su automatización. Algunas de las deficiencias que se manifiestan en los laboratorios de calidad de software son:

- Los recursos humanos y tecnológicos en el laboratorio de calidad de software se planifican y se estiman de forma manual.
- No existe un control total de los recursos humanos y tecnológicos que se encuentran disponibles.
- La asignación de los recursos humanos y tecnológicos, no se realiza de forma eficiente.
- La planificación actual trae consigo pérdida de tiempo.

Por lo anteriormente expuesto se plantea el **problema científico**: ¿Cómo contribuir al desarrollo de los procesos de calidad de software?

Por tanto el **objeto de estudio** es: El desarrollo de los procesos de calidad de software y el **campo de acción** está determinado por: El proceso de estimación y planificación de los recursos para las actividades de un laboratorio de pruebas y certificación de calidad en la UCI. Por lo que se plantea como **Objetivo general** de la Investigación: Definir el proceso de estimación y planificación de los recursos humanos y tecnológicos, y desarrollar una aplicación Web que permita elevar la eficiencia del proceso de pruebas del laboratorio de certificación de la UCI. Para cumplir el objetivo general del trabajo se trazaron los **objetivos específicos**:

- Definir el proceso de estimación y planificación para un Laboratorio de Calidad.
- Implementar una aplicación web del proceso definido.

Las **tareas** a desarrollar son:

- Realización de entrevistas a especialistas de Calidad para definir los procesos a partir de la experiencia acumulada.
- Definición del proceso de planificación en un laboratorio de calidad.
- Modelado del negocio de la herramienta informática.
- Levantamiento de requisitos funcionales.

- Realización del flujo de trabajo de análisis y diseño.
- Implementación de la herramienta informática que se propone.

La investigación está estructurada por: Introducción, cuatro capítulos, conclusiones, recomendaciones, referencia bibliográfica, bibliografía, glosario de términos y anexos.

**Capítulo 1** Fundamentación Teórica: se abordan los objetivos estratégicos de la organización, una breve valoración del flujo actual de los procesos y la definición de los mismos, su análisis crítico, así como las tendencias y tecnologías actuales relacionadas con el objeto de estudio.

**Capítulo 2** Características del Sistema: se refleja un análisis detallado del flujo actual de los procesos, los trabajadores que desarrollan dichos procesos y otras personas involucradas. Se define el objeto de automatización, la información que se maneja y la propuesta del sistema. Se identifican los requisitos funcionales y no funcionales, los casos de uso con sus descripciones, los diagramas de actividades y el modelo de objeto.

**Capítulo 3** Análisis y Diseño del Sistema: se analizan los casos de uso del sistema para diseñar las clases que se implementarán, se representan los diagramas de secuencia del diseño, el diagrama de las clases diseñadas con sus relaciones, los principios utilizados para el diseño de dichas clases, el diagrama de clases persistentes, el diagrama entidad relación y las descripciones de las tablas de la base de datos.

**Capítulo 4** Implementación y Prueba: se reflejan los diferentes diagramas correspondientes a este flujo de trabajo, donde se implementará la propuesta realizada en el análisis y diseño, se representa el diagrama de despliegue, los de componentes y el modelo de prueba correspondiente a la aplicación desarrollada.

## **Capítulo 1 Fundamentación Teórica.**

### **Introducción.**

En este capítulo se abordan los objetivos estratégicos de la organización, una breve valoración del flujo actual de los procesos y la definición de los mismos, su análisis crítico, así como las tendencias y tecnologías actuales relacionadas con el objeto de estudio.

### **1.1 Calidad de Software.**

El desarrollo de aplicaciones de software ha devenido en solución para el creciente adelanto de las tecnologías de la información. En el mundo actual hay varias empresas comprometidas en este desarrollo y muchas de ellas a veces fallan por ignorar un aspecto tan importante como lo es la calidad de software. Aunque del tema calidad ya se han dado varios pasos a nivel mundial, sigue siendo un tema que duele a varias de estas empresas.

Hacer las cosas con calidad significa hacer las cosas bien con el costo previsto, y preocuparse de hacer las cosas mejor en cada ocasión. ¿Y qué es “hacer las cosas bien”? Precisamente, conseguir que los objetivos se cumplan según los planes establecidos. El camino de la Calidad es largo, pero en cada paso nos da una mejora, un ahorro, una satisfacción.

La calidad de software es un conjunto de cualidades que caracterizan y determinan la utilidad y existencia del software. Es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad. A raíz de todo esto se crean los laboratorios de calidad de software como una variante para comprobar las funcionalidades de los software producidos. Ejemplos de estos laboratorios, en Ottawa, Canadá en la universidad de Carleton, en Barcelona, España el InQlab y uno en la ciudad argentina de Córdoba.

El tema de la calidad en nuestro país es muy joven todavía, en Cuba existen varias empresas que desarrollan aplicaciones de software y tienen sus propios sistemas de garantizar la calidad de sus productos, pero no existía un laboratorio donde se certificaran. Con la creación de la Universidad de las Ciencias Informáticas y por ende el aumento de la producción de software la necesidad de fabricar un producto de calidad creció, esto fue

la base para la creación de un laboratorio de certificación y pruebas, para que el propio proceso de calidad tuviera calidad, es necesario realizar una buena planificación de los recursos involucrados en el proceso.

Existen algunas herramientas informáticas para la planificación de recursos, humanos y tecnológicos pero al analizarlas se concluye que su utilización no es aplicable a nuestro país puesto que tienen precios muy costosos y no cubren todas nuestras necesidades, es por esto que se decide implementar una herramienta para la planificación adaptada a nuestro entorno y necesidades, vale mencionar la visita realizada por parte de los integrantes del proyecto al ministerio de informática donde se facilitó información sobre la creación de gestores de proyectos pero llevados a un modelo cubano. Ya que muchos de los que existen en el mundo, la mayoría son propietarios y no se enmarcan con las características del país.

## **1.2 Planificación y estimación de recursos en un laboratorio de calidad.**

El éxito de cualquier actividad en la vida humana depende en gran medida de la planificación que se haga. Además una estimación acertada de una actividad ayuda a valorar, evaluar y prever el tiempo que se necesita para la misma.

Planificación: Acción global o conjunto de medidas pertenecientes a un plan establecido y concreto, realizado a la consecución de un fin. [1]

Estimación: Conjetura sobre el comportamiento futuro de una variable bajo ciertas condiciones. Determinación del valor o valores del o los parámetros de un cierto modelo con base en la observación de los resultados de un experimento. [2]

Estas definiciones de Planificación y Estimación ayudan a entender lo importante que pueden ser para el futuro desarrollo de cualquier tarea que emprendamos, para nuestro caso específico en las pruebas de calidad de un software. En la realización de estas pruebas entran a jugar un conjunto de recursos: Computadoras, Personas y Tiempo. Un buen aprovechamiento de estos recursos permitirá terminar un trabajo, en menor tiempo y de una forma más eficaz.

### 1.2.1 Definición del proceso de Planificación.

La definición del proceso de Planificación que a continuación se muestra, es el resultado de año y medio de trabajo en un laboratorio de pruebas de software de los autores de este trabajo y la experiencia de varios especialistas de calidad, en el se reflejan las principales actividades que involucra el proceso, así como los responsables y participantes del mismo, este proceso forma parte de las actividades de un laboratorio de calidad en la UCI.

Descripción del proceso.	
Código	01
Proceso	Planificación de los recursos en un laboratorio de Calidad de Software.
Responsable	Jefe de prueba del laboratorio.
Participantes	Especialistas de Calidad, Jefes de prueba.
Clientes	Dirección de Calidad

#### I- Misión

Garantizar una buena planificación de los recursos de un laboratorio de calidad.

#### II- Resultados a obtener

La planificación de todas las pruebas a realizar a un software en la Universidad de Ciencias Informáticas.

#### III- Funciones

##### 01-Se deciden las pruebas que se le aplicarán a un software.

Decisión por parte de la dirección de calidad de la universidad de las pruebas que se le aplicarán a un software.

##### 02- Se determinan las métricas que se le aplicarán a la prueba.

En dependencia de la prueba a planificar se decide aplicar una métrica para estimar un posible tiempo de duración, estas métricas son por cada tipo de prueba y complejidad de la misma.



**03- Se determinan los recursos involucrados en la prueba.**

Se determinan todos los recursos necesarios o con los que se cuenten para la realización de las pruebas ya sean humanos o tecnológicos.

**04- Se realiza la planificación de la prueba.**

Una vez que se tengan todos los recursos y las métricas que se van a aplicar se procede a la planificación de la prueba donde se determina por turnos de trabajo la fuerza necesaria a intervenir y los recursos tecnológicos así como una posible fecha de comienzo y fin de la prueba.

<b>IV- Relación de actividades del proceso</b>				
<b>Código</b>	<b>Actividad</b>	<b>Responsable</b>	<b>Participante</b>	<b>Resultados a obtener</b>
01.01	Se decide las pruebas que se le aplicarán al software.	Especialista de Calidad.	Especialista de Calidad.	Listado de las pruebas que se el aplicarán a un software.
01.02	Se determinan las métricas que se le aplicarán a las pruebas.	Especialista de Calidad.	Especialista de Calidad.	Se obtienen las métricas a aplicar a cada tipo de prueba.
01.03	Se determinan los recursos involucrados en la prueba.	Jefe de prueba	Jefe de prueba	Cantidad de estudiantes con que se dispone así como cantidad de computadoras u otros hardwares.
01.04	Se realiza la planificación de la prueba.	Jefe de prueba	Jefe de prueba	Se obtiene una planificación con todos los recursos involucrados en las pruebas y una estimación de la duración de la prueba.

**Tabla 1.1 Relación de actividades en el proceso.**

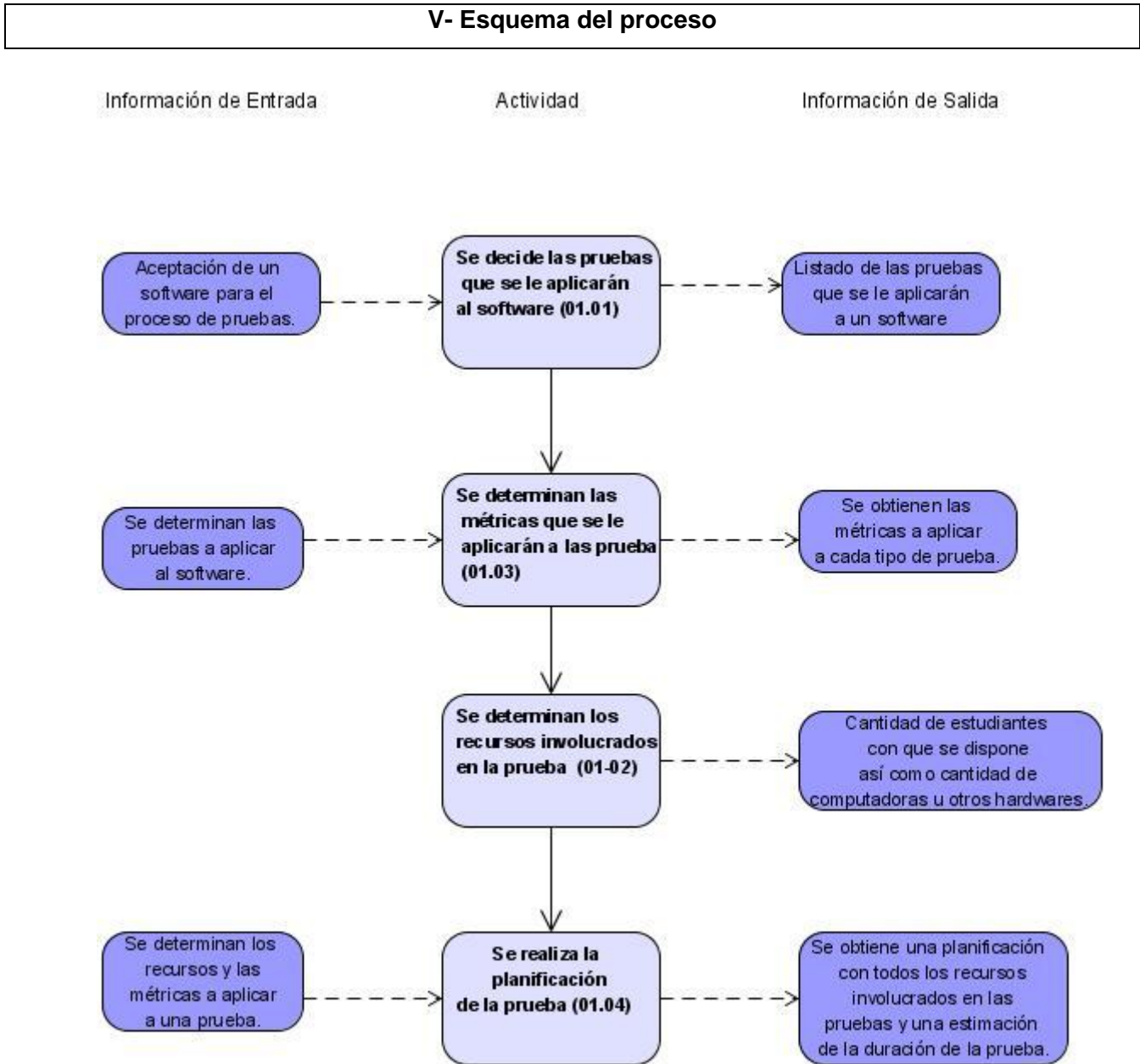


Figura 1.1 Esquema del proceso.

<b>Procedimiento de ejecución de las actividades</b>	
Código	01.01.01
Actividad	Se decide las pruebas que se le aplicarán al software.
Proceso	Planificación de los recursos en un laboratorio de Calidad de Software.
Responsable	Especialista de Calidad.
Participantes	Especialista de Calidad.

**I- Resultados a obtener**

Listado de las pruebas que se aplicarán a un software.

**II- Descripción del procedimiento**

Los especialistas de calidad de la dirección de calidad de la universidad aceptan las revisiones a un software y en dependencia de lo que se quiera probar del software, se decide cuales tipos de prueba se le realizarán.

<b>Procedimiento de ejecución de las actividades</b>	
Código	01.01.02
Actividad	Se determinan las métricas que se le aplicarán a las pruebas.
Proceso	Planificación de los recursos en un laboratorio de Calidad de Software.
Responsable	Especialista de Calidad.
Participantes	Especialista de Calidad.

**I- Resultados a obtener**

Se obtienen las métricas a aplicar a cada tipo de prueba.

**II- Descripción del procedimiento**

El especialista una vez que decida las pruebas a aplicar a un software decide que métricas aplicar a las mismas, basándose en las complejidades que puede presentar. Estas complejidades varían según el tipo de prueba en: Complejidad de Casos de Uso, Complejidad de Capítulos, Complejidad de Prueba.

**Complejidad de Casos de Uso:**

Este tipo de complejidad está presente en las pruebas que para su realización es indispensable trabajar con los casos de uso del sistema.

**Complejidad de Capítulos:**

Este tipo de complejidad está presente en las pruebas que se realizan a algún tipo de documentación.

**Complejidad de Prueba:**

Este tipo de complejidad está presente en las pruebas que para su realización no es necesario trabajar con ningún tipo de documentación en específico.

Cada tipo de complejidad tiene dentro de él 3 clasificaciones (Alta, Media, Baja) que varían en dependencia de la prueba. Las métricas aplicadas a estas complejidades se utilizarán a la hora de la planificación de las pruebas.

<b>Procedimiento de ejecución de las actividades</b>	
Código	01.01.03
Actividad	Se determinan los recursos involucrados en la prueba.
Proceso	Planificación de los recursos en un laboratorio de Calidad de Software.
Responsable	Jefe de prueba.
Participantes	Jefe de prueba.

**I- Resultados a obtener**

Se obtiene la cantidad de estudiantes con que se dispone para realizar las pruebas así como la cantidad de computadoras u otros hardwares necesarios para llevar a cabo las pruebas que se realizarán.

**II- Descripción del procedimiento**

El jefe de prueba evalúa los recursos con que cuenta para la realización de la prueba, ya sean recursos humanos o tecnológicos, o sea determina la cantidad de estudiantes que tiene disponible, así como la cantidad de computadoras con que cuenta en ese momento, además de tener en cuenta otros tipos de hardware que necesita como scanner o impresora y determina si tiene alguno de ellos a su disposición.

<b>Procedimiento de ejecución de las actividades</b>	
Código	01.01.04
Actividad	Se realiza la planificación de la prueba.
Proceso	Planificación de los recursos en un laboratorio de Calidad de Software.
Responsable	Jefe de prueba.
Participantes	Jefe de prueba.

**I- Resultados a obtener**

Se obtiene una planificación de todos los recursos involucrados en las pruebas y una estimación de la duración de la prueba.

**II- Descripción del procedimiento**

El jefe de prueba una vez que tenga las pruebas con sus métricas y los recursos que va a utilizar definidos, procede a realizar la planificación, en ella tiene que constar los turnos de trabajo que se necesitarán para realizar el trabajo teniendo en cuenta la cantidad de estudiantes y de PC, además debe aplicar las métricas antes seleccionadas para estimar una posible fecha de realización del trabajo.

<b>Descripción de contenidos de trabajo por áreas</b>	
Código	A01
Área	Dirección de Calidad
Subordinada a:	Dirección de la IP.

**I- Resultados por los que responde**

Selección de las pruebas que se le aplicarán a un software  
 Selección de las métricas que se utilizarán en la planificación.

<b>II- Relación de actividades que ejecuta</b>	
Código	Actividad
01.01	Se decide las pruebas que se le aplicaran al software.
01.02	Se determinan las métricas que se el aplicaran a las pruebas.

<b>Descripción de contenidos de trabajo por áreas</b>	
Código	A02
Área	Laboratorio de Calidad de las facultades.
Subordinada a:	Vice-decanato de producción de las facultades.

**I- Resultados por los que responde**

Determinación de los recursos humanos y tecnológicos.  
 Realización de la planificación.

<b>II- Relación de actividades que ejecuta</b>	
Código	Actividad
01.03	Se determinan los recursos involucrados en la prueba
01.04	Se realiza la planificación de la prueba

<b>Descripción de puestos de trabajo</b>	
Código	PT01
Denominación	Especialista de Calidad
Área	A01 (Dirección de Calidad)

**I- Resultados por los que responde**

Selección de las pruebas que se le aplicarán a un software

Selección de las métricas que se utilizarán en la planificación.

**II- Relación de actividades que ejecuta**

Código	Actividad
01.01	Se decide las pruebas que se le aplicarán al software.
01.02	Se determinan las métricas que se le aplicarán a las pruebas.

**Descripción de puestos de trabajo**

Código	PT02
Denominación	Jefe de prueba.
Área	A02 (Laboratorio Calidad de la Facultad)

**I- Resultados por los que responde**

Determinación de los recursos humanos y tecnológicos.

Realización de la planificación.

**II- Relación de actividades que ejecuta**

Código	Actividad
01.03	Se determinan los recursos involucrados en la prueba
01.04	Se realiza la planificación de la prueba

## 1.3 Herramientas afines

### 1.3.1 Swag

Es una empresa que ofrece soluciones integrales en Tecnologías de la Información. Utilizando los mayores estándares de calidad para obtener la mayor seguridad y fiabilidad en los servicios y productos. Es una organización que pretende aportar soluciones imaginativas y dinámicas a las problemáticas de las empresas. Su modelo de trabajo es responder a las necesidades concretas de cada cliente. Dando un importante valor añadido a los negocios tradicionales mediante la incorporación de la tecnología más moderna y de la imaginación. [3]

### 1.3.2 Gestor de proyectos

El Gestor de Proyectos permite planificar y administrar los proyectos empresariales, organizando y realizando un seguimiento de las tareas de forma eficaz para evitar retrasos y ajustarse al presupuesto.

El programa permite diseñar, gestionar y evaluar los proyectos de la organización de forma transversal, facilitando la distribución de la información entre las diversas áreas de la empresa y auditando que las tareas asignadas a cada trabajador se realizan de forma correcta y conforme a los plazos establecidos.

Mediante el uso del gestor se puede calcular previsiones precisas de gastos y hacer un seguimiento de los tiempos y la facturación, así como de las personas que ejecutan las operaciones.

Algunas de las funcionalidades del Gestor de Proyectos son:

- Jerarquización en la planificación del proyecto: procesos, tareas y operaciones.
- Foros de discusiones y encuestas para los proyectos.
- Envío de alertas automático vía e-mail de modificaciones en el proyecto.
- Sistema de búsqueda basado en palabras clave.
- Posibilidad de vincular todo tipo de documentos a las tareas.
- Módulos de anuncios para informar de novedades y eventos.



- Asignación de roles de seguridad para cada tipo de usuario.
- Completo sistema de estadísticas, informes y gráficos del proyecto.
- Muestra de la diferencia entre el proyecto estimado y el curso actual de éste. [4]

### **1.3.3 Project KickStart**

Es una herramienta poderosa y fácil de usar que le ayudará a diseñar, organizar y programar cualquier proyecto.

El proceso de 8 pasos de Project KickStart ayuda a concentrar la atención en la estructura del proyecto, los objetivos, los recursos, los riesgos y las cuestiones estratégicas que son vitales para el éxito del proyecto..

Programa un proyecto utilizando el calendario del menú descendente y la gráfica de Gantt. Imprime su listado de tareas o uno de los siete informes preestablecidos. Agrega versatilidad a su proyecto. Establece un "enlace dinámico" de su información con Microsoft Project, Word, Outlook, PowerPoint ó Excel.

Características y ventajas

- Se puede utilizar en proyectos de cualquier tamaño - hasta 1000 tareas y 100 recursos.
- Se puede iniciar con Ejemplos de Proyectos- ya contienen información y están listos para utilizarse.
- Listado de sugerencias en los archivos de Objetivos, Fases y Obstáculos, Gráfica de Gantt para programación "completa".
- Siete tipos de informes preestablecidos.
- Archívelo como HTML- planeación post proyecto para su red interna.
- Enlace dinámico con Word, Outlook, PowerPoint y Excel para incluir proyectos en la planeación de sus propuestas y planes de negocio. [5]

## **1.4 Tendencias y Tecnologías Actuales.**

### **1.4.1 Proceso de desarrollo de software**

**RUP**

Es uno de los procesos más generales de los existentes actualmente,

Un proyecto realizado siguiendo RUP se divide en cuatro fases:

- Inicio.
- Elaboración.
- Construcción.
- Transición.

En cada fase se ejecutarán una o varias iteraciones (de tamaño variable según el proyecto), y dentro de cada una de ellas seguirá un modelo de cascada para los flujos de trabajo que requieren las nuevas actividades anteriormente citadas.

RUP define nueve actividades a realizar en cada fase del proyecto

1. Modelado del negocio
2. Requerimiento
3. Análisis y diseño
4. Implementación
5. Prueba
6. Distribución
7. Gestión de configuración y cambios
8. Gestión del proyecto
9. Ambiente

RUP está caracterizado por ser dirigido por Casos de Usos, centrado en la Arquitectura, iterativo e incremental. [6]

## **XP**

Mientras que el RUP intenta reducir la complejidad del software por medio de estructura y la preparación de las tareas pendientes en función de los objetivos de la fase y actividad actual, XP como toda metodología ágil, lo intenta por medio de un trabajo orientado directamente al objetivo, basado en las relaciones Interpersonales y la velocidad de reacción.

XP intenta minimizar el riesgo de fallo del proceso por medio de la disposición permanente de un representante competente del cliente a disposición del equipo de desarrollo. Este representante debería

estar en condiciones de contestar rápida y correctamente a cualquier pregunta del equipo de desarrollo de forma que no se retrase la toma de decisiones, de ahí lo de competente.

La codificación del software en XP se produce siempre en parejas (dos programadores, un ordenador), por lo que se espera que la calidad del mismo suba en el mismo momento de escribirlo. Al contrario que muchos otros métodos, el código pertenece al equipo en completo, no a un programador o pareja, de forma que cada programador puede cambiar cualquier parte del código en cualquier momento si así lo necesita, dejándose en todo caso las mejoras orientadas al rendimiento para el final. Las parejas no se mantienen para todo el proyecto sino que rotan cíclicamente a lo largo del mismo, tanto en cuanto a los componentes de la misma como en las partes del software que desarrollan, así cada componente del equipo aprende como trabaja el resto. El objetivo ideal sería que cada componente del equipo trabaje al menos una vez con cada uno de los demás integrantes y con cada componente software, de forma que el conocimiento de la aplicación completa lo posea el equipo entero y no unos pocos miembros.

### **¿Por qué RUP?**

- Dividido en cuatro fases
- Las fases se dividen en iteraciones
- El discurrir del proyecto se define en Workflows
- Los artefactos son el objetivo de cada actividad
- Se basa en roles
- UML
- Muy organizativo
- Mucha documentación

Por lo expuesto anteriormente se llega a la conclusión de que este proceso de desarrollo es el correcto para llevar a cabo la descripción de este software, debido a que es un proceso muy organizativo, el cual se basa en roles, viendo así de una forma más descriptiva el papel que juega cada objeto en el sistema.

[7]

### 1.4.2 Lenguaje de modelado.

#### Uml

El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación.

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. UML ofrece nueve diagramas en los cuales modelar sistemas.

- Diagramas de Casos de Uso para modelar los procesos 'business'.
- Diagramas de Secuencia para modelar el paso de mensajes entre objetos.
- Diagramas de Colaboración para modelar interacciones entre objetos.
- Diagramas de Estado para modelar el comportamiento de los objetos en el sistema.
- Diagramas de Actividad para modelar el comportamiento de los Casos de Uso, objetos u operaciones.
- Diagramas de Clases para modelar la estructura estática de las clases en el sistema.
- Diagramas de Objetos para modelar la estructura estática de los objetos en el sistema.
- Diagramas de Componentes para modelar componentes.
- Diagramas de Implementación para modelar la distribución del sistema.

UML es una consolidación de muchas de las notaciones y conceptos más usados orientados a objetos. Empezó como una consolidación del trabajo de Grade Booch, James Rumbaugh, e Ivar Jacobson, creadores de tres de las metodologías orientadas a objetos más populares. [8]

### 1.4.3 Visual Paradigm

Visual Paradigm ofrece:

- Entorno de creación de diagramas para UML 2.0
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.

- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones.
- Disponibilidad en múltiples plataformas. [9]

#### **1.4.4 La Arquitectura Cliente-Servidor**

Con la proliferación de ordenadores personales de bajo coste en el mercado, los recursos de sistemas de información existentes en cualquier organización se pueden distribuir entre ordenadores de diferentes tipos: ordenadores personales de gama baja, media y alta, estaciones de trabajo, mini ordenadores o incluso grandes ordenadores.

El concepto de cliente/servidor proporciona una forma eficiente de utilizar todos estos recursos de máquina de tal forma que la seguridad y fiabilidad que proporcionan los entornos mainframe se traspasa a la red de área local. A esto hay que añadir la ventaja de la potencia y simplicidad de los ordenadores personales.

La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes.

En este modelo las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario.

Los clientes realizan generalmente funciones como:

- Manejo de la interfaz de usuario.
- Captura y validación de los datos de entrada.
- Generación de consultas e informes sobre las bases de datos.

Por su parte los servidores realizan, entre otras, las siguientes funciones:

- Gestión de periféricos compartidos.
- Control de accesos concurrentes a bases de datos compartidas.
- Enlaces de comunicaciones con otras redes de área local o extensa.

Entre las principales características de la arquitectura cliente/servidor se pueden destacar las siguientes:

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

## **Ventajas**

- Aumento de la productividad:
- Los usuarios pueden utilizar herramientas que le son familiares, como hojas de cálculo y herramientas de acceso a bases de datos.
- Una interfaz gráfica de usuario consistente reduce el tiempo de aprendizaje de las aplicaciones.
- Menores costes de operación:
- Permiten un mejor aprovechamiento de los sistemas existentes.[10]

### **1.4.5 Apache: Servidor Web**

Apache es un servidor Web desarrollado por el grupo Apache.

Según estudios realizados por diferentes empresas el servidor Web más utilizado en Internet es Apache.

Apache está disponible para una gran multitud de plataformas:

GNU/Linux

Mac OS y Mac OS X Server

Netware

UNIX comerciales como AIX (R), Digital UNIX (R), HP-UX (R), IRIX (R), SCO (R), Solaris (R), SunOS (R),

Windows (R)

Apache funciona en casi todas las plataformas actuales. Debido a esto se puede escoger la plataforma que más se adapte a nuestras características, y también se podrá cambiar de plataforma si en un momento determinado una plataforma ofrece más ventajas que la que se está utilizando.

Gracias a esto se produce una independencia tecnológica del fabricante de hardware lo que hace que el fabricante esté en continua evolución y ofreciendo productos de calidad a sus clientes ya que en caso de inconformidad por parte de los clientes estos siempre podrían elegir otra plataforma hardware.

Se puede restringir el acceso a determinadas páginas de un sitio Web de una forma sencilla y de fácil mantenimiento. Además puede personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto.

#### **1.4.6 MySQL**

MySQL es la base de datos open source más popular y, posiblemente, mejor del mundo. Su continuo desarrollo y su creciente popularidad está haciendo de MySQL un competidor cada vez más directo de gigantes en la materia de las bases de datos como Oracle. Es un sistema de administración de bases de datos (Database Management System, DBMS) para bases de datos relacionales. Así, MySQL no es más que una aplicación que permite gestionar archivos llamados de bases de datos.

MySQL fue escrito en C y C++ y destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos.

También es muy destacable, la condición de open source de MySQL, que hace que su utilización sea gratuita e incluso se pueda modificar con total libertad, pudiendo descargar su código fuente. Esto ha favorecido muy positivamente en su desarrollo y continuas actualizaciones, para hacer de MySQL una de las herramientas más utilizadas por los programadores orientados a Internet.[12]

### 1.4.7 SQL Server

Microsoft SQL Server es un sistema de gestión de bases de datos relacionales (SGBD) basada en el lenguaje SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. Así de tener unas ventajas que a continuación se pueden describir.

Entre sus características figuran:

Soporte de transacciones.

Gran estabilidad.

Gran seguridad.

Escalabilidad.

Soporta procedimientos almacenados.

Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo accedan a la información.

Además permite administrar información de otros servidores de datos.

Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle o Sybase ASE.

Es común desarrollar completos proyectos complementando Microsoft SQL Server y Microsoft Access a través de los llamados ADP (Access Data Project). De esta forma se completa una potente base de datos (Microsoft SQL Server) con un entorno de desarrollo cómodo y de alto rendimiento (VBA Access) a través de la implementación de aplicaciones de dos capas mediante el uso de formularios Windows. Para el desarrollo de aplicaciones más complejas (tres o más capas), Microsoft SQL Server incluye interfaces de acceso para la mayoría de las plataformas de desarrollo, incluyendo .NET. Microsoft SQL Server, al contrario de su más cercana competencia, no es multiplataforma, ya que sólo está disponible en Sistemas Operativos de Microsoft. [13]

### 1.4.8 ¿ Por qué MySQL?

Luego de analizadas las características y facilidades de los SGBD presentados, y las de la herramienta a desarrollar se decide 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.

#### **1.4.9 PHP 5 (Hypertext Preprocessor)**

PHP (Hypertext Preprocessor) es un lenguaje script , (no se compila para conseguir códigos máquina si no que existe un intérprete que lee el código y se encarga de ejecutar las instrucciones que contiene éste código), para el desarrollo de páginas web dinámicas del lado del servidor, cuyos fragmentos de código se intercalan fácilmente en páginas HTML, debido a esto, y a que es de Open Source (código abierto), es el más popular y extendido en la web. (14)

El PHP es un lenguaje de script incrustado dentro del HTML. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de si mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas.

PHP ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación. Su diseño elegante lo hace perceptiblemente más fácil de mantener y ponerse al día que el código comparables en otros lenguajes.

Debido a su amplia distribución PHP esta perfectamente soportado por una gran comunidad de desarrolladores. Como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparan rápidamente. El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.[15]

#### **1.4.10 Java Script**

JavaScript es un lenguaje de scripts compacto basado en objetos (y no orientado a objetos). JavaScript permite la realización de aplicaciones de propósito general a través de la WWW y aunque no está diseñado para el desarrollo de grandes aplicaciones, es suficiente para la implementación de aplicaciones WWW completas o interfaces WWW hacia otras más complejas.

Por ejemplo, una aplicación escrita en JavaScript puede ser utilizada en un documento HTML proporcionando un mecanismo para el tratamiento de eventos, como un clic del ratón o bien la validación de entrada de datos en un formulario.

Sin que exista una comunicación a través de la red una página HTML con JavaScript incrustado puede interpretar, y alertar al usuario con una ventana de diálogo, de que las entradas de los formularios no son válidas. O bien realizar algún tipo de acción como ejecutar un fichero de sonido o un applet de Java. [16]

#### **1.4.11 HTML**

El HTML es un lenguaje de marcas hipertextuales, un lenguaje diseñado para estructurar textos para generar páginas web. Gracias a Internet y a los navegadores web, el HTML se ha convertido en el formato más fácil para la creación de páginas web debido a su sencillez. La mayoría de las etiquetas del lenguaje html son semánticas. La interpretación de las etiquetas es realizada por el navegador web. El lenguaje HTML es extensible, se le pueden añadir características, etiquetas y funciones adicionales para el diseño de páginas web, generando un producto vistoso, rápido y sencillo. [17]

#### **1.4.12 ASP.net**

- Flexibilidad y extensibilidad. Muchas características de ASP.NET son extensibles para que pueda incorporar características personalizadas con facilidad a las aplicaciones. Por ejemplo, el modelo de proveedor ASP.NET proporciona compatibilidad de conexión para diferentes orígenes de datos.
- Rendimiento. Las características como la compilación previa, el almacenamiento en caché configurable y la invalidación de la caché de SQL le permiten optimizar el rendimiento de sus aplicaciones Web.
- Seguridad. Es fácil lograr una autenticación y autorización de aplicaciones Web.
- Alojamiento. ASP.NET incluye nuevas características que hacen más fácil administrar un entorno de host y crear más oportunidades para los anfitriones de dar más valor.
- Integridad. Las características nuevas y las existentes funcionan conjuntamente para permitirle crear escenarios de un extremo a otro que afrontan retos reales de desarrollo Web. [18]

#### 1.4.13 ¿Por qué PHP?

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad.
- Leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML.
- Posee una amplia documentación en su página oficial.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Nos permite crear los formularios para la Web.

#### 1.4.14 DreamWeaver 8

Dreamweaver es la herramienta de diseño de páginas web más avanzada, tal como se ha afirmado en muchos medios. Aunque sea un experto programador de HTML el usuario que lo maneje, siempre se encontrarán en este programa razones para utilizarlo, sobretodo en lo que a productividad se refiere.

Cumple perfectamente el objetivo de diseñar páginas con aspecto profesional, y soporta gran cantidad de tecnologías, además muy fáciles de usar:

- Hojas de estilo y capas
- Java script para crear efectos e interactividades
- Inserción de archivos multimedia.

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

Es compatible con las principales tecnologías de servidor como por ejemplo, ColdFusion, PHP, ASP, ASP.NET y JSP, para que los desarrolladores, más o menos expertos, puedan dar vida a sus diseños [19]

#### 1.4.15 WAMP 5

W.A.M.P. son las siglas de Windows + Apache + MySQL + PHP, es decir, un instalador paso a paso que incluye en un solo paquete estas tres tecnologías para montar tu propio servidor bajo Windows.

El software que se instala con WAMP5 contiene los siguientes servidores y programas:

Apache: El servidor de páginas web más extendido del mercado.

PHP5. El motor renovado del lenguaje.

MySQL. La base de datos más extendida para utilizar con PHP.

PHPmyadmin. Un software que permite administrar una base de datos a través de una interfaz web.

SQLitemanager. Un sistema para administrar una base de datos a partir de sentencias SQL. [20]

### **Conclusiones.**

- Se desea desarrollar una Herramienta Software para estimación y planificación de las pruebas de software.
- Se ha escogido el lenguaje de programación PHP por las posibilidades multiplataforma que este brinda, además por su posibilidad de uso gratuito.
- El editor seleccionado para la programación de este fue DreamWeaver, por la facilidad en su uso.
- El lenguaje utilizado para la metodología fue el de modelación UML y la herramienta CASE empleada fue Visual Paradigm

## Capítulo 2 Características del sistema.

### Introducción

En este capítulo se refleja un análisis detallado del flujo actual de los procesos, los trabajadores que desarrollan dichos procesos y otras personas involucradas. Se define el objeto de automatización, la información que se maneja y la propuesta del sistema. Se identifican los requisitos funcionales y no funcionales, los casos de uso con sus descripciones, los diagramas de actividades y el modelo de objeto.

### 2.1 Objeto de Estudio.

#### 2.1.1 Problema y situación problemática.

La investigación se enmarca dentro de la Calidad de software, los objetivos que propone la asignatura Gestión de Software, encargada de impartir este tema en la Universidad y estrechamente relacionada con la Dirección de Calidad de dicha universidad, son:

- Desarrollar una conciencia económica y una cultura de la calidad, sin perder de vista la satisfacción del cliente como meta.
- Aplicar técnicas para asegurar la calidad de los productos informáticos en la empresa.
- Aplicar los procedimientos para controlar y planificar proyectos.

Para dar cumplimiento a estos objetivos la Dirección de Calidad de la Universidad de las Ciencias Informáticas realiza varios procesos donde se destacan el proceso de aseguramiento de la calidad, las pruebas de aceptación, liberación, revisión y auditoría. Dentro de estos procesos, la planificación de los recursos utilizados juega un papel importante, pues es el encargado de mantener un uso racional de los recursos. El proceso que se describirá en esta investigación será la planificación y estimación de los recursos humanos y tecnológicos en un laboratorio de calidad, este es iniciado por el especialista de calidad, que decide que pruebas aplicar a cada producto y envía la solicitud de planificación al Jefe de

Prueba, este recibe esta solicitud y determina los recursos necesarios para la planificación, posteriormente realiza la planificación y confecciona un informe de esta, lo envía al especialista de calidad, con el cual se podrán conocer todos los recursos involucrados para realizar una prueba.

Este proceso se realiza de forma manual y no siempre la persona encargada de la planificación es quién lleva a cabo esta tarea. Además la comunicación entre las partes involucradas en este proceso no siempre se realiza la manera indicada. No se tiene un conocimiento exacto de los recursos disponibles, por lo que se hace una mala gestión de los mismos, todo esto conlleva a la pérdida de tiempo y que la planificación no se realiza de forma exacta y satisfactoria.

### **2.1.2 Objeto de automatización.**

La automatización del recursos, comienza con Insertar Nueva Prueba, aquí se inserta el nombre de la prueba y se le asigna una clasificación, (Casos de uso, Capítulos, Complejidad de prueba), además se le insertan las métricas asociadas, una vez insertadas las pruebas se procede a Gestionar Pruebas que incluye: Buscar Pruebas, Mostrar Pruebas, Seleccionar Pruebas, Modificar Pruebas y Eliminar Pruebas.

Si las métricas utilizadas para realizar las planificaciones no son las más eficientes se procede a Gestionar Métricas, estas métricas se pueden buscar, mostrar, seleccionar, modificar. Una actividad necesaria antes de la planificación es Insertar los datos del proyecto, estos datos, al igual que las métricas se pueden gestionar. Al proceder a la planificación y estimación, se deben insertar los recursos y los atributos necesarios para cada prueba, una vez terminada esta, se envía una confirmación de concluida la planificación. El sistema debe dar la opción de Gestionar las Planificaciones para emitir respuesta de esta forma a las solicitudes de reportes que se den por partes de los personas interesadas en ellas.

### **2.1.3 Información que se maneja.**

1. Documento de las planificaciones realizadas
  - Fecha.
  - Nombre del que realiza la planificación.
  - Cantidad de estudiantes.

- Cantidad de computadoras.
  - Casos de usos o Capítulos.
  - Turnos empleados.
  - Nombre de la prueba a planificar.
  - Iteración.
  - Nombre del proyecto.
  - Nombre del módulo.
2. Documentos con los Datos de los proyectos.
- Nombre del Proyecto.
  - Cantidad de módulo.
  - Nombres de los módulos.
  - Iteraciones.
  - Pruebas realizadas.
3. Actas de Entregas
- Documentación requerida para la prueba de un software.
4. Listado de métricas aplicadas.
- Valores de métricas según complejidades.
5. Especificación de Casos de Uso.
- Listado de casos de usos involucrados en el proyecto con sus requisitos asociados.
6. Manuales de Instalación.
- Información por pasos de la instalación.
7. Manuales de Usuarios.

- Información detallada de las interfaces del software.
- 8. Documento de Integración.
  - Información de integración de los módulos de un proyecto.
- 9. Documento de Respuesta a las No Conformidades.
  - Respuesta de las no conformidades por parte de los desarrolladores.

#### **2.1.4 Propuesta de sistema.**

La herramienta que se propone desarrollar (SoftlaQ) automatizará 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.**

El principal objetivo de este módulo es la planificación y estimación de los recursos, para poder realizar esta actividad anteriormente en el sistema deben estar insertadas las pruebas y sus métricas correspondientes. Estas planificaciones serán adicionadas en el sistema, y se podrán acceder a ellas mediante reportes que serán solicitados por personas interesadas. Estos reportes además de exportarlos e imprimirlos se pueden gestionar: buscar, mostrar, seleccionar, modificar y eliminar. Las pruebas y las métricas también podrán ser gestionadas.

##### **El módulo de No conformidades.**

En este módulo se gestionarán todas las no conformidades encontradas durante el proceso de revisión. 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. Además el sistema permitirá exportar e imprimir los diferentes reportes de No Conformidades.

##### **El módulo de seguimiento.**

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 construirá utilizando como lenguaje de programación PHP5, aprovechando las amplias ventajas que él mismo ofrece y que fueron descritas en el capítulo 1, además 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. Existirán 5 tipos de usuarios: administradores, especialista de calidad, jefes de prueba, probadores y desarrolladores. Cada usuario que entra al sistema debe autenticarse antes de realizar alguna acción.

## 2.2 Modelo de negocio.

### 2.2.1 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 un laboratorio de calidad de software en la UCI. Inicialmente el solicitante (vicedecanos de producción, asesor de calidad de cada facultad) 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 incluye tres actividades fundamentales, la revisión de documentación, planificación de las pruebas y ejecución de pruebas.

En la revisión de la documentación se verifica que el proyecto cumpla con los lineamientos básicos que exige la dirección de calidad, de no cumplir, se le informa la negación de la revisión del software, en caso contrario se comienza con la planificación y estimación de las pruebas, luego se realiza la ejecución de las mismas incluyendo la realización 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 su interés.

### 2.2.2 Actores y trabajadores del negocio.

Actores del negocio	Justificación
Solicitante	Asesor de calidad de la facultad o vicedecano de producción que soliciten 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 Actores del negocio.**

Trabajadores del negocio	Justificación
Jefe de prueba	Encargado en cada facultad de controlar los recursos en el laboratorio de calidad. Planifica los recursos del laboratorio, 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	Consulta las no conformidades emitidas y las responde.

**Tabla 2.2 Trabajadores del negocio.**

### 2.2.3 Diagrama Casos de Uso del Negocio.

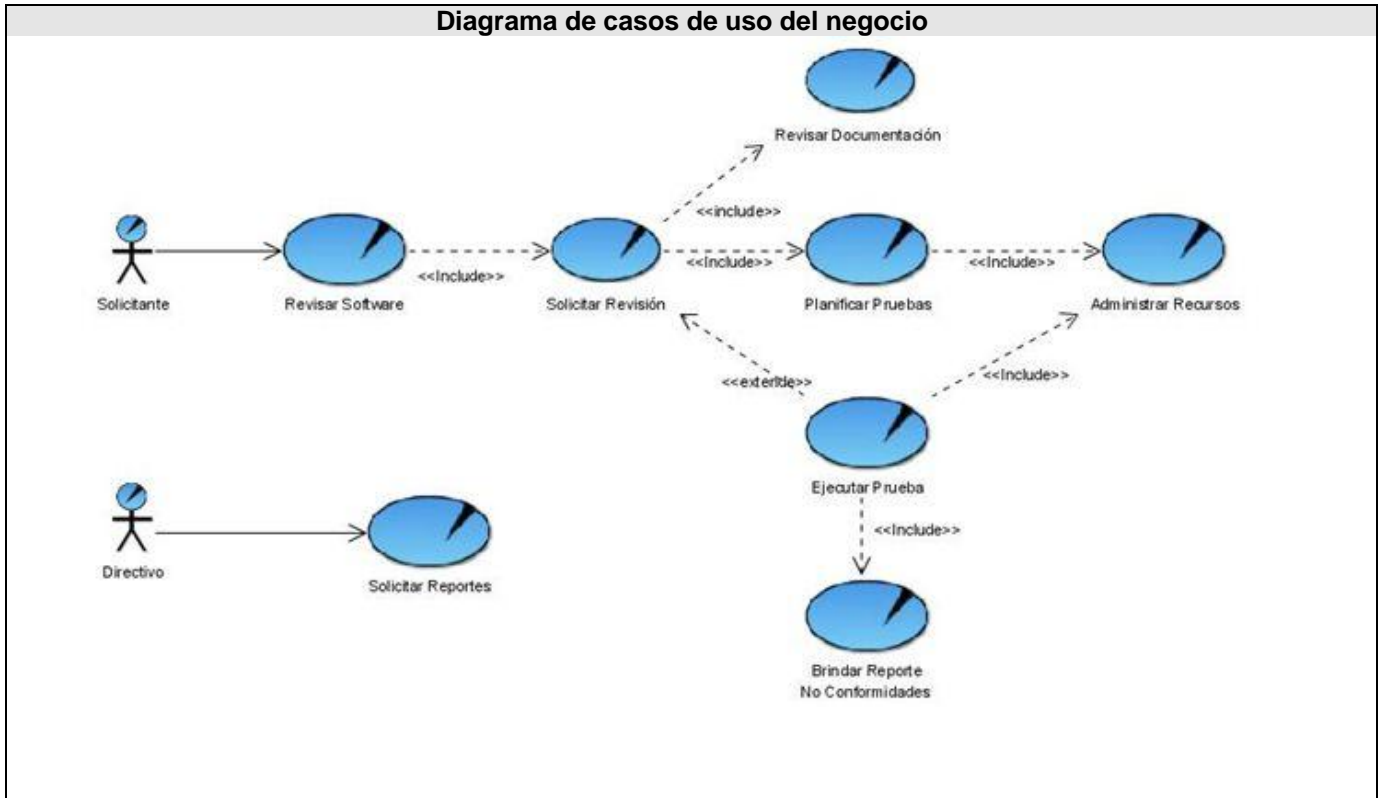
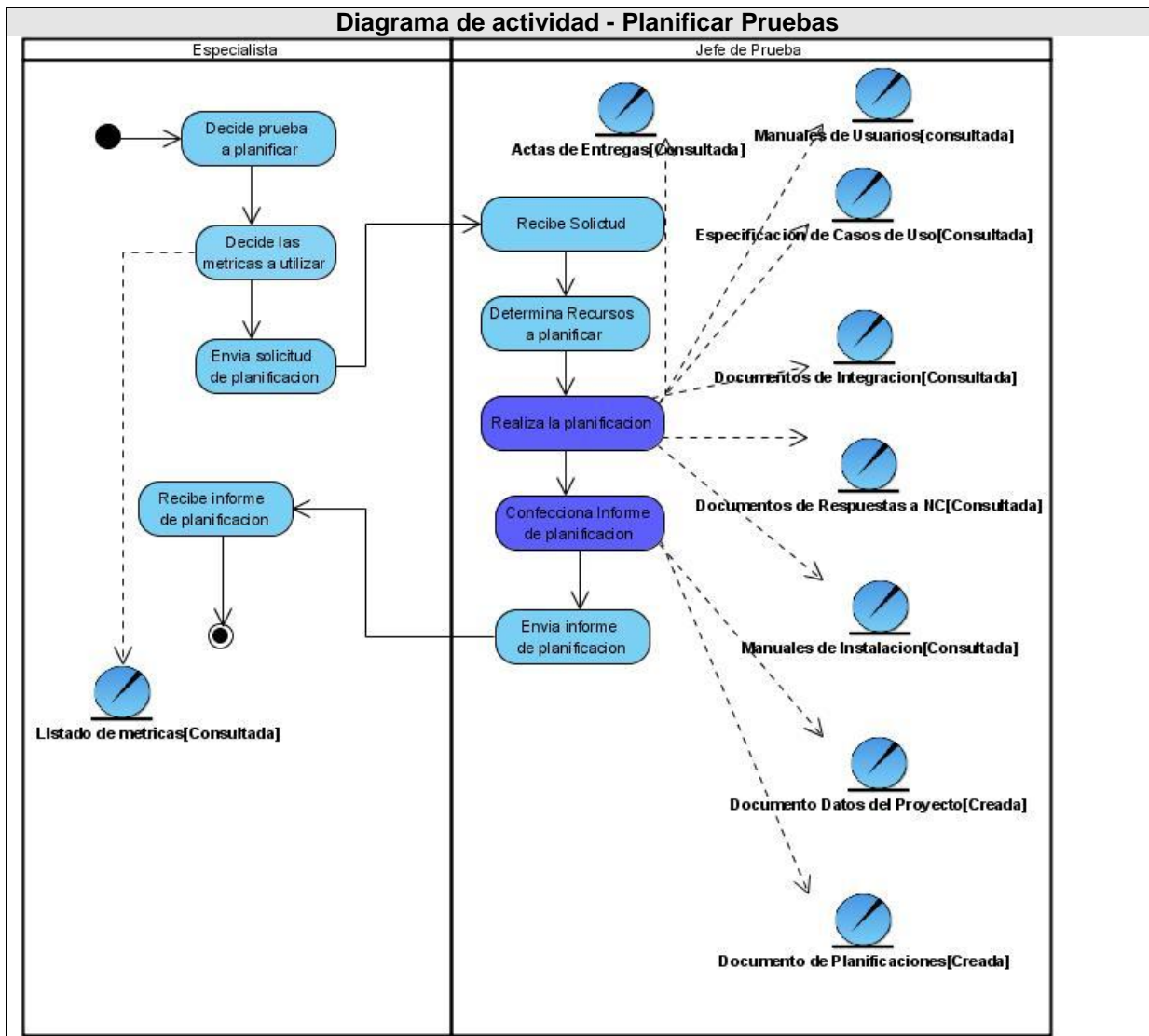


Figura 2.1 Diagrama Caso de Uso del negocio.

Listado de casos de uso del negocio, con la descripción y actores o trabajadores que participan.

- Solicitar Revisión
- Revisar Software.
- Revisar Documento.
- Planificar Prueba.
- Ejecutar Prueba
- Brindar Reporte de No conformidades.
- Administrar recursos.
- Solicitar Reportes.

#### **2.2.4 Diagrama de actividades. Caso de Uso Planificar Pruebas**



**Figura 2.2 Diagrama de Actividades Caso de Uso Planificar Pruebas**

El especialista decide la prueba a planificar y selecciona las métricas a utilizar, y este le envía la solicitud de planificación al jefe de prueba, una vez que este recibe la solicitud determina los recursos con que cuenta, una vez que tenga los recursos planifica y elabora el informe de planificación, este informe es enviado al especialista.

### 2.3 Diagrama de Clases del Modelo de Objetos.

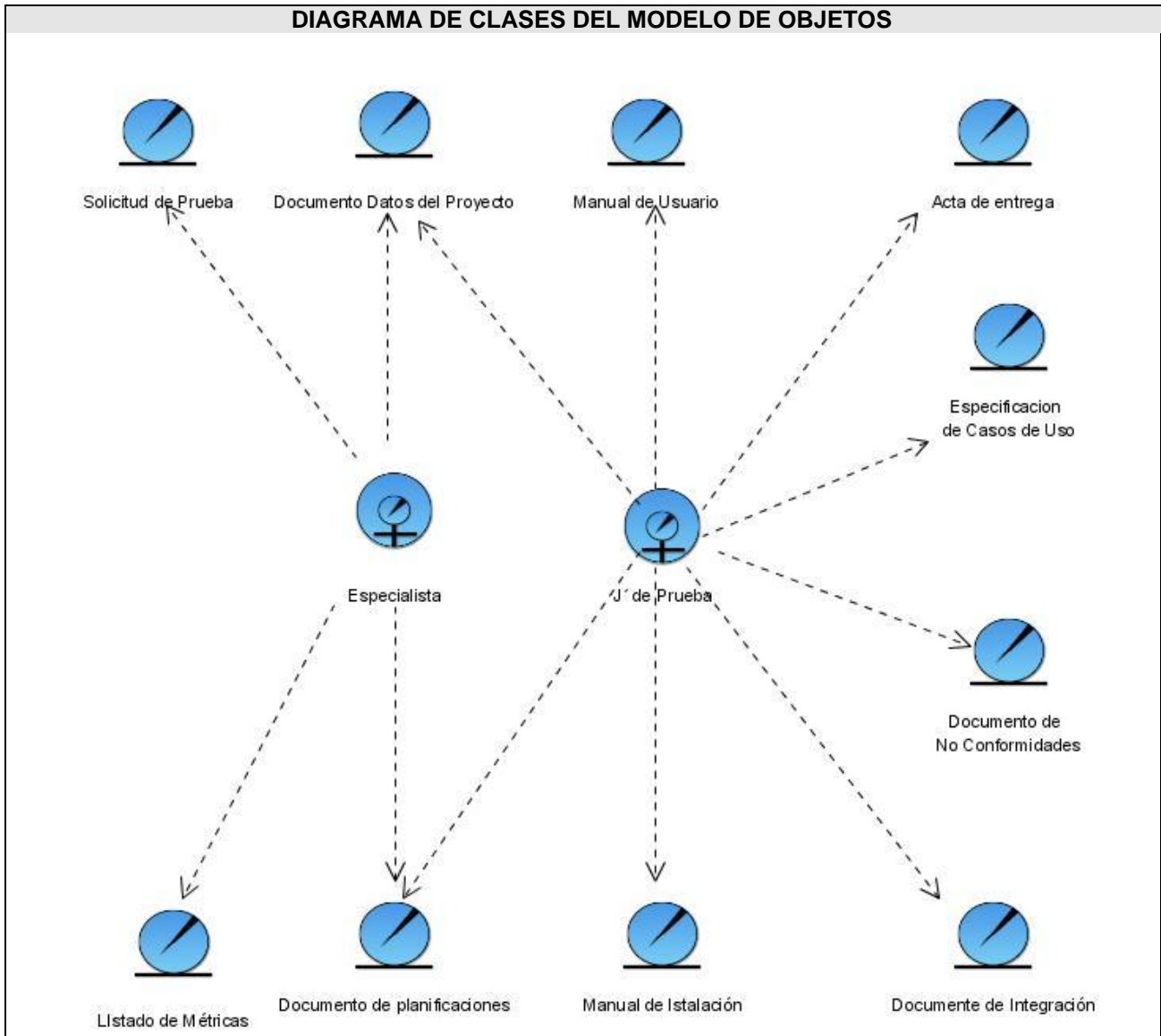


Figura 2.3 Diagrama de Clases del modelo de Objetos

## **2.4 Especificación de los requisitos de software.**

### **2.4.1 Requerimientos Funcionales.**

#### **1. Gestionar Pruebas**

##### **1.1. Insertar Nueva Prueba**

1.1.1. Insertar Nombre de Prueba.

1.1.2. Seleccionar clasificación.

1.1.1.1. Capítulos.

1.1.1.2. Casos de Usos.

1.1.1.3. Prueba.

##### **1.2. Buscar pruebas**

1.2.1. Nombre de la prueba

1.2.2. Clasificación.

##### **1.3. Mostrar pruebas.**

1.3.1. Nombre de la prueba

##### **1.4. Seleccionar Pruebas para gestionarlas.**

##### **1.5. Modificar pruebas.**

1.5.1. Nombre de la prueba.

1.5.2. Clasificación.

##### **1.6. Eliminar Pruebas**

#### **2. Gestionar Métrica.**

## **2.1 Insertar métricas.**

2.1.1 Valores de la métrica para cada complejidad (Alta, Media, Baja) para cada tipo de prueba.

## **2.2 Buscar Métricas.**

2.2.1 Nombre de prueba.

2.2.2 Clasificación.

## **2.3 Mostrar métricas.**

2.3.1 Nombre de la prueba.

2.3.2 Valores de la métrica asociado a esa prueba.

## **2.4 Seleccionar métricas para modificarlas.**

## **2.5 Modificar métricas.**

2.5.1 Valores de la métrica.

## **3. Insertar datos del proyecto.**

### **3.1. Insertar datos del proyecto.**

3.1.1. Nombre del proyecto

3.1.2. Iteración del proyecto.

3.1.3. Fecha

3.1.4. Jefe de Prueba.



### **3.2. Insertar datos de los módulos.**

- 3.2.1. Nombre del proyecto
- 3.2.2. Jefe de Prueba
- 3.2.3. Nombre del módulo
- 3.2.4. Iteración del módulo.
- 3.2.5. Fecha

## **4. Gestionar datos del proyecto.**

### **4.1. Buscar datos del proyecto.**

- 4.1.1. Nombre del proyecto
- 4.1.2. Nombre del módulo.
- 4.1.3. Iteración.
- 4.1.4. Jefe de Prueba.

### **4.2. Mostrar datos del proyecto.**

- 4.2.1. Nombre del proyecto
- 4.2.2. Nombre del módulo.
- 4.2.3. Iteración.
- 4.2.4. Fecha
- 4.2.5. Jefe de Prueba.

### **4.3. Seleccionar datos del proyecto para gestionarlos.**

### **4.4. Modificar datos del proyecto.**

4.4.1. Nombre del proyecto

4.4.2. Iteración.

4.4.3. Fecha

4.4.4. Jefe de Prueba.

**4.5. Eliminar proyecto.**

## **5. Planificar pruebas.**

### **5.1. Seleccionar datos para la planificación.**

5.1.1. Mostrar listado con los nombres de los jefes de prueba.

5.1.2. Seleccionar el jefe de prueba que va a planificar.

5.1.3. Mostrar listado con los nombres de los proyectos.

5.1.4. Seleccionar el proyecto que se va a planificar.

5.1.5. Mostrar los módulos del proyecto que se va a planificar.

5.1.6. Seleccionar el módulo que se va a planificar.

5.1.7. Mostrar listado con los nombres de las pruebas a planificar.

5.1.8. Seleccionar la prueba que se va a planificar.

**5.2. Planificar pruebas con combinaciones de:**

- 5.2.1. Cantidad de máquinas
- 5.2.2. Cantidad de estudiantes
- 5.2.3. Turnos de trabajo.
- 5.2.4. Capítulos con complejidad alta
- 5.2.5. Capítulos con complejidad media
- 5.2.6. Capítulos con complejidad baja
- 5.2.7. Casos de Usos con complejidad alta
- 5.2.8. Casos de Usos con complejidad media
- 5.2.9. Casos de Usos con complejidad baja
- 5.2.10. Complejidad de prueba alta
- 5.2.11. Complejidad de prueba media
- 5.2.12. Complejidad de prueba baja.

**5.3. Enviar confirmación.**

- 5.3.1. Enviar confirmación de la planificación terminada a los interesados vía correo.

**6. Gestionar planificación.**

**6.1. Buscar planificación.**

6.1.1. Nombre del proyecto

6.1.2. Nombre del módulo.

6.1.3. Iteración.

6.1.4. Intervalos de Fecha

6.1.5. Jefe de Prueba.

6.1.6. Tipo de Prueba.

**6.2. Mostrar planificación.**

6.2.1. Nombre del proyecto

6.2.2. Nombre del módulo.

6.2.3. Intervalos de Fecha

6.2.4. Jefe de Prueba.

6.2.5. Tipo de Prueba.

6.2.6. Cantidad de Estudiantes.

6.2.7. Cantidad de máquinas.

6.2.8. Cantidad de Turnos.

**6.3. Seleccionar planificación para gestionarla.****6.4. Modificar planificación.**

- 6.4.1. Tipo de Prueba.
- 6.4.2. Cantidad de máquinas.
- 6.4.3. Cantidad de estudiantes.
- 6.4.4. Turnos de trabajo.
- 6.4.5. Capítulos con Complejidad Alta.
- 6.4.6. Capítulos con Complejidad Media.
- 6.4.7. Capítulos con Complejidad Baja.
- 6.4.8. Casos de Usos con Complejidad Alta.
- 6.4.9. Casos de Usos con Complejidad Media.
- 6.4.10. Casos de Usos con Complejidad Baja.
- 6.4.11. Complejidad de Prueba Alta.
- 6.4.12. Complejidad de Prueba Media.
- 6.4.13. Complejidad de Prueba Baja.

**6.5. Exportar planificación.**

- 6.5.1. La planificación se exporta en la parte de la PC que se desee en formato pdf.

**6.6. Imprimir Planificación.****6.7. Eliminar Planificación.****7. Gestionar Usuario.**

**7.1. Crear usuario del sistema.**

7.1.1. Usuario

7.1.2. Contraseña

**7.2. Eliminar usuario del sistema.****7.3. Modificar datos del usuario.**

7.3.1. Usuario

7.3.2. Contraseña

**7.4. Asignar a los usuarios los roles:**

7.4.1. Jefe de Prueba

7.4.1.1. Insertar datos del proyecto.

7.4.1.2. Gestionar datos del proyecto.

7.4.1.3. Planificar pruebas.

7.4.1.4. Gestionar Planificación.

7.4.1.5. Autenticar Usuario.

7.4.2. Especialista

7.4.2.1. Gestiona Pruebas.

7.4.2.2. Gestiona Métricas.

7.4.2.3. Gestionar Datos del Proyecto.

7.4.2.4. Gestionar Planificación.

7.4.2.5. Autenticar Usuario.

### 7.4.3. Administrador

#### 7.4.3.1. Gestionar Usuario.

#### 7.4.3.2. Autenticar Usuario

## 8. Autenticar Usuario.

### 8.1. Autenticar Usuario.

#### 8.1.1. Usuario.

#### 8.1.2. Contraseña.

### 2.5.2 Requerimientos 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 fácil de comprender.

#### 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, así 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 JavaScript y tener instalado el PDF para los reportes. 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.

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 imprimir los diferentes tipos de reportes. Pentium IV, 239 GHz, 249 MB de RAM.

## 2.6 Definición de los Casos de Uso

### 2.6.1 Definición de los actores.

Actores	Justificación
Especialista	Este actor en el sistema inserta pruebas con sus métricas y las gestiona. Además puede gestionar los datos del proyecto y las planificaciones.
Jefe de prueba	Este actor selecciona las pruebas a planificar, inserta los datos del proyecto, planifica las pruebas y envía la confirmación de planificación terminada. También puede gestionar las planificaciones y los datos del proyecto.
Gestor	Este actor gestiona la planificación y los datos del proyecto.
Usuario	Este actor es para al autenticación en el sistema.
Administrador	Este actor gestiona todos los usuarios del sistema.

Tabla 2.3 Definición de Actores.

### 2.6.2 Listado de Casos de Uso.

Gestionar Pruebas	CU-1
Gestionar Métrica	CU-2
Insertar Datos del Proyecto	CU-3



Planificar Pruebas	CU-4
Gestionar Datos del Proyecto	CU-5
Gestionar Planificación	CU-6
Gestionar Usuario	CU-7
Autenticarse	CU-8

**Tabla 2.4 Listado de Casos de Uso del Sistema.**

Para ver la definición de los casos de uso del sistema consultar [Anexo1](#).

## 2.7 Diagrama de Casos de Uso del Sistema.

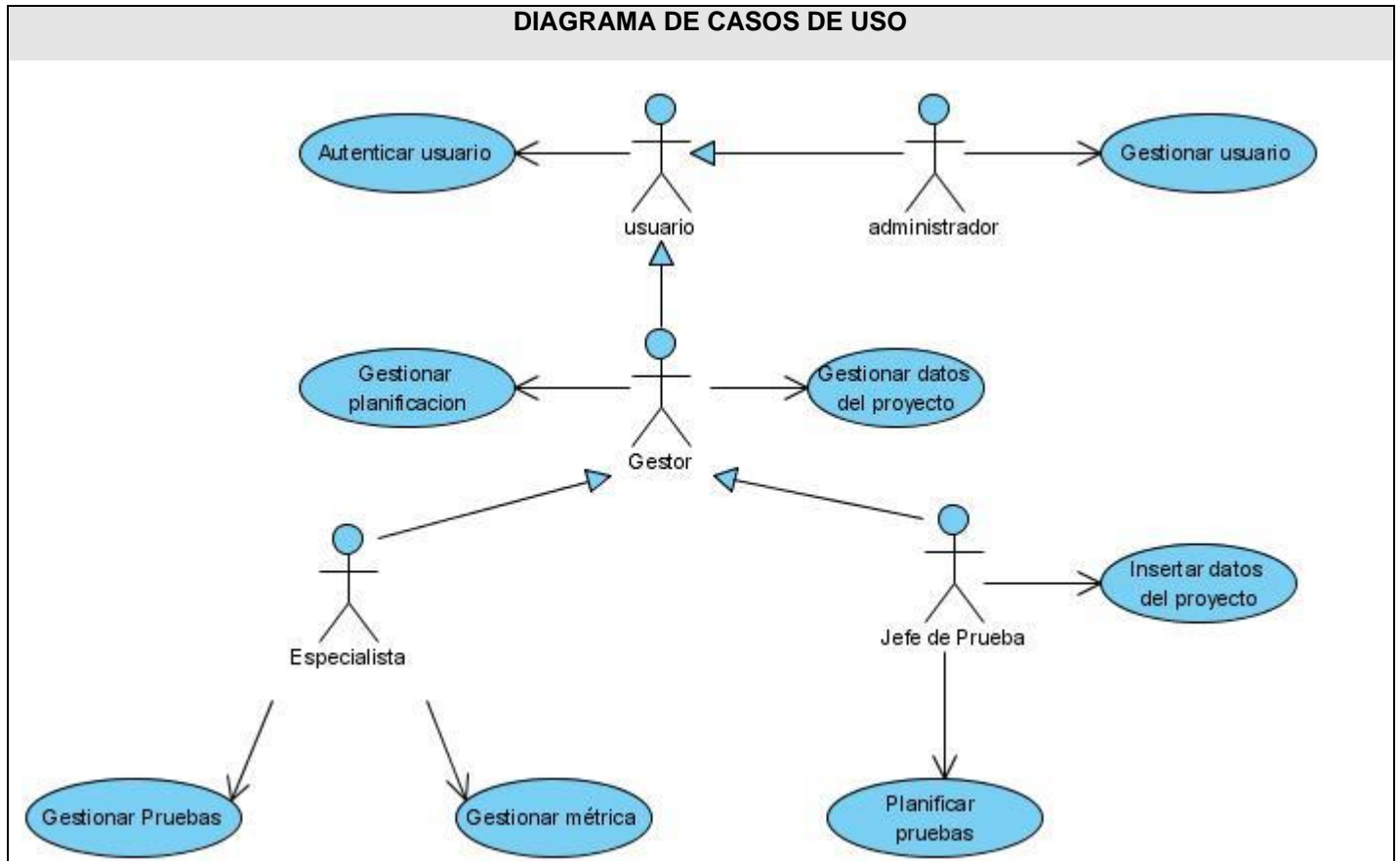


Figura 2.4 Diagrama de Casos de Uso del sistema

## 2.8 Casos de Uso por ciclo.

Cód	Nombre de caso de uso	Paquete	Justificación de la selección.
CU1	Gestionar Pruebas	1	Este caso de uso tiene gran importancia, pues para realizar una planificación es necesario tener pruebas insertadas, y estas pruebas se insertan en este caso de uso,

			además después de insertar alguna prueba se puede modificar algún atributo que se inserte mal y este caso de uso es el encargado de esta operación.
<b>CU3</b>	<b>Insertar datos del proyecto</b>	<b>1</b>	Este caso de uso es importante, pues para realizar una planificación se necesita tener el proyecto al cual se va a probar, insertado, y este proyecto se inserta en este caso de uso.
<b>CU5</b>	<b>Gestionar datos del Proyecto</b>	<b>1</b>	Cuando se desea modificar o eliminar los datos de un proyecto insertado este caso de uso se encarga de hacerlo. Estas funciones se pueden realizar antes de planificar.
<b>CU4</b>	<b>Planificar</b>	<b>2</b>	Este caso de uso es el encargado de realizar la planificación de una prueba insertada a un proyecto, planifica y estima los recursos con que se cuenta para realizar las pruebas.
<b>CU2</b>	<b>Gestionar Métrica</b>	<b>3</b>	Una vez que se realice la planificación con una métrica, y esta métrica no resulte, se puede modificar mediante este caso de uso.
<b>CU6</b>	<b>Gestionar Planificaciones</b>	<b>3</b>	Una vez que se tenga planificaciones realizadas se puede ejecutar este caso de uso, que es el encargado de gestionar estas planificaciones realizando operaciones sobre ellas como modificar, eliminar, imprimir o exportar.
<b>CU8</b>	<b>Gestionar Usuario</b>	<b>4</b>	Este caso de uso es el encargado de crear

			los usuarios y asignar los roles que deben tener cada uno de ellos.
<b>CU7</b>	<b>Autenticar Usuario</b>	<b>4</b>	Después que es creado el usuario, este se podrá autenticar en el sistema.

**Tabla 2.5 Casos de Uso por ciclo.**

### 2.9 Descripción de CUS.

<b>Caso de uso: Gestionar Pruebas</b>	
<b>Actores:</b> Especialista de Calidad	
<b>Propósito</b>	
Realizar operaciones sobre una prueba, tales como : realizar búsquedas, eliminarla etc.	
<b>Resumen</b>	
Se podrán insertar nuevas pruebas al sistema además la posibilidad de realizar búsquedas, se seleccionaran las pruebas mostradas, para modificarlas o eliminarlas.	
<b>Precondiciones</b>	
Las pruebas deben estar aceptadas por la Dirección de calidad.	
<b>Referencias:</b> R 1.1, R 1.2 , R 1.3, R 1.4 , R 1.5 , R 1.6,R 2.1.	
<b>Casos de uso asociados:</b>	
<b>Postcondiciones</b>	
La prueba queda actualizada.	
<b>Requerimientos especiales:</b>	
<b>Prototipo:</b>	
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>

<p>1. El especialista selecciona insertar una nueva prueba.</p>	<p>2. El Sistema muestra la interfaz para insertar pruebas con los siguientes campos.</p> <ul style="list-style-type: none"> <li>• Nombre de la prueba.</li> </ul> <p>Seleccionar un tipo de complejidad</p> <ul style="list-style-type: none"> <li>• Capítulos</li> <li>• Casos de Usos</li> <li>• Complejidad de prueba</li> </ul> <p>Insertar valores para las métricas de esa prueba.</p> <ul style="list-style-type: none"> <li>• Complejidad Alta</li> <li>• Complejidad Media</li> <li>• Complejidad Baja</li> </ul>
<p>3. El especialista introduce el nombre de la prueba, escoge el tipo de complejidad que esta prueba posee e inserta los valores de métrica por complejidades.</p>	<p>4. El sistema inserta toda la información de la prueba.</p>
<p>5. El especialista realiza búsquedas de pruebas insertadas.</p>	<p>6. Ver sección “Buscar prueba”.</p>
<p><b>Sección “Buscar Prueba ”</b></p>	
<p><b>Acción del Actor</b></p>	<p><b>Respuesta del Sistema</b></p>

<p>1. El especialista selecciona buscar una prueba insertada.</p>	<p>2. El sistema muestra la interfaz para buscar pruebas con los siguientes criterios de búsqueda.</p> <ul style="list-style-type: none"> <li>• Nombre de la prueba</li> <li>• Clasificación.</li> </ul>
<p>3. El especialista escoge los criterios para su búsqueda.</p>	<p>4. En caso de existir pruebas que cumplan con esos criterios el sistema las muestra. En caso contrario ir al flujo alterno 4.1</p>
<p>5. El especialista selecciona la prueba para realizar algunas de las operaciones siguientes.</p> <ul style="list-style-type: none"> <li>• Modificar Prueba.</li> <li>• Eliminar prueba.</li> </ul>	<p>6. En dependencia de la selección del actor se realizan las siguientes operaciones:</p> <ul style="list-style-type: none"> <li>• Si desea modificar la prueba ver la sección “Modificar Prueba”.</li> <li>• Si desea eliminar la prueba, el sistema elimina la prueba.</li> </ul>
<p><b>Curso Alternativo</b></p>	
	<p>4.1 Si no existen pruebas por los criterios de búsqueda introducidos se le informa al usuario. Proveniente del flujo básico 4.</p>
<p><b>Sección “Modificar Prueba ”</b></p>	
<p><b>Acción del Actor</b></p>	<p><b>Respuesta del Sistema</b></p>
<p>1. El especialista selecciona modificar una prueba.</p>	<p>2. El sistema muestra la interfaz modificar pruebas de acuerdo a los siguientes criterios.</p> <ul style="list-style-type: none"> <li>• Nombre de la prueba.</li> <li>• Clasificación.</li> </ul>

3. El especialista modifica los datos que desee y selecciona guardar.	4. El sistema guarda la prueba con los nuevos cambios.
<b>Sección “Eliminar Prueba ”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El especialista selecciona eliminar una prueba.	2. El sistema muestra un mensaje de confirmación si desea eliminar la prueba.
3. Si el especialista desea eliminar la prueba acepta el mensaje sino ver flujo alterno 3.1	4. El sistema elimina la planificación.
<b>Curso Alternativo</b>	
3.1 Si el especialista no desea eliminar la planificación cancela el mensaje. Proveniente del flujo 3.	3.2 El sistema muestra la interfaz de búsqueda.

**Tabla 2.6 Descripción Caso de Uso Gestionar Prueba.**

<b>Caso de uso: Insertar datos del proyecto</b>
<b>Actores:</b> Jefe de Prueba
<b>Propósito</b>
Este caso de uso se realiza para conocer los datos del proyecto involucrado en la planificación.
<b>Resumen</b>
Se insertan datos de interés del proyecto como: Nombre del proyecto, Nombre del módulo así como datos del modulo, Iteración, Fecha, Asesor.

<b>Precondiciones</b>	
El proyecto es seleccionado para ser probado.	
<b>Referencias: R 3.1, R 3.2</b>	
<b>Casos de uso asociados:</b>	
<b>Postcondiciones</b>	
Se insertan los datos del proyecto.	
<b>Requerimientos especiales:</b>	
<b>Prototipo:</b>	
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1. El jefe de prueba selecciona insertar los datos de un proyecto.	2. El Sistema muestra la interfaz para insertar los datos del proyecto. <ul style="list-style-type: none"> <li>• Nombre del proyecto</li> <li>• Iteración.</li> <li>• Fecha</li> <li>• Nombre del Jefe de Prueba.</li> </ul>
3. El jefe de prueba introduce los datos.	4. El sistema inserta los datos y muestra la interfaz de insertar datos de los módulos. Ver sección “Insertar datos del módulo”
<b>Sección “Insertar datos del módulo ”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1 El sistema muestra la interfaz para insertar los datos del módulo. <ul style="list-style-type: none"> <li>• Seleccionar Proyecto al que pertenece.</li> <li>• Nombre de Jefe de prueba</li> <li>• Nombre del módulo</li> <li>• Iteración</li> </ul>



	<ul style="list-style-type: none"> <li>• Fecha.</li> </ul>
2. El jefe de prueba introduce los datos.	3. El sistema inserta los datos.

**Tabla 2.7 Descripción Caso de Uso Insertar datos del proyecto.**

<b>Caso de uso: Gestionar datos del proyecto.</b>
<b>Actores:</b> Gestor
<b>Propósito</b>
Realizar operaciones sobre los datos del proyecto como: búsquedas, modificaciones, en caso de que estos datos sufran alguna variación.
<b>Resumen</b>
Con los datos del proyecto insertados se podrá realizar búsquedas, se seleccionaran los datos mostrados, para modificarlos o eliminarlos.
<b>Precondiciones</b>
-Se hallan insertado datos del proyecto.
<b>Referencias:</b> R 4.1, R 4.2, R 4.3 , R 4.4 , R 4.5
<b>Casos de uso asociados:</b>
<b>Postcondiciones</b>
Los datos del proyecto quedan actualizados.
<b>Requerimientos especiales:</b>
<b>Prototipo:</b>
<b>Curso Normal de los Eventos</b>

Acciones del Actor	Respuesta del Sistema
<p>1. El gestor selecciona realizar una búsqueda de los datos del proyecto insertados.</p>	<p>2. El sistema muestra la interfaz de búsqueda de los datos según los siguientes criterios.</p> <ul style="list-style-type: none"> <li>• Nombre del proyecto</li> <li>• Nombre del módulo.</li> <li>• Iteración.</li> <li>• Jefe de Prueba.</li> </ul>
<p>3. El gestor introduce los criterios que necesita.</p>	<p>4. En caso de que existan datos que cumpla con los criterios introducidos el sistema muestra el resultado. En caso contrario ver flujo alterno 4.1</p>
<p>5. El gestor selecciona los datos y realiza algunas de las operaciones:</p> <ul style="list-style-type: none"> <li>• Modificar los datos.</li> <li>• Eliminar un proyecto.</li> </ul>	<p>6. En dependencia de la selección del actor se realizan las operaciones:</p> <ul style="list-style-type: none"> <li>• Si desea modificar los datos ver la sección “Modificar Datos”.</li> <li>• Si desea eliminar los datos, ver la sección “Eliminar proyecto”.</li> </ul>
<b>Curso Alternativo</b>	
	<p>4.1 Si no existen datos que cumpla con los criterios el sistema informa al usuario. Proveniente del flujo básico 4.</p>
<b>Sección “Modificar Datos ”</b>	

Acción del Actor	Respuesta del Sistema
1. El gestor selecciona modificar los datos.	2. El sistema muestra la interfaz de modificar datos con todos los campos posibles a modificar. <ul style="list-style-type: none"> <li>• Nombre del proyecto</li> <li>• Nombre del módulo</li> <li>• Iteración</li> <li>• Fecha</li> <li>• Asesor</li> </ul>
3. El gestor modifica los datos.	4. El sistema guarda los nuevos datos.
<b>Sección “Eliminar Datos ”</b>	
Acción del Actor	Respuesta del Sistema
1. El gestor selecciona eliminar un proyecto.	2. El sistema muestra un mensaje de confirmación si desea eliminar el proyecto.
3. Si el gestor desea eliminar el proyecto acepta el mensaje, sino ir a flujo alterno 3.1	4. El sistema elimina el proyecto.
<b>Curso Alternativo</b>	
3.1 Si no desea eliminar el proyecto cancela el mensaje. Proveniente del flujo básico 3.	3.2 El sistema muestra la interfaz de búsqueda.

**Tabla 2.8 Descripción Caso de Uso Gestionar datos del proyecto.**

<b>Caso de uso: Gestionar Métricas.</b>	
<b>Actores:</b> Especialista de Calidad	
<b>Propósito</b>	
Realizar operaciones sobre las métricas como: búsquedas y modificaciones cuando se quiera alterar el valor de una de ellas.	
<b>Resumen</b>	
Con los datos de las métricas se podrá realizar búsquedas, se seleccionaran los datos mostrados para ser modificados.	
<b>Precondiciones</b>	
Se encuentren pruebas insertadas.	
<b>Referencias:</b> R 2.1, R 2.2, R 2.3, R 2.4, R 2.5	
<b>Casos de uso asociados:</b>	
<b>Postcondiciones</b>	
Se actualizan las métricas que están siendo utilizadas.	
<b>Requerimientos especiales:</b>	
<b>Prototipo:</b>	
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1. El especialista selecciona realizar una búsqueda de las métricas insertadas.	2. El sistema muestra la interfaz de búsqueda de las métricas según los siguientes criterios. <ul style="list-style-type: none"> <li>• Nombre de Prueba</li> <li>• Clasificación.</li> </ul>

<p>3. El especialista introduce los criterios que necesita.</p>	<p>4. En caso de que existan métricas que cumpla con los criterios introducidos el sistema muestra el resultado. En caso contrario ir a flujo alternativo 4.1</p>
<p>2 El especialista selecciona las métricas para modificarla.</p>	<p>2.1 Si desea modificar las métricas ver la sección “Modificar métricas”.</p>
<p><b>Curso Alternativo</b></p>	
	<p>4.1 Si no existen métricas que cumplan con los criterios el sistema informa al usuario. Proveniente del flujo 4.</p>
<p><b>Sección “Modificar Métricas ”</b></p>	
<p><b>Acción del Actor</b></p>	<p><b>Respuesta del Sistema</b></p>
<p>1. El especialista selecciona modificar una métrica.</p>	<p>2. El sistema muestra la interfaz de modificar Métricas con el campo posible a modificar.</p> <ul style="list-style-type: none"> <li>• Valores de la métrica.</li> </ul>
<p>2.4 El especialista modifica los valores de esa métrica.</p>	<p>2.5 El sistema guarda las métricas con los nuevos valores.</p>

**Tabla 2.9 Descripción Caso de Uso Gestionar Métricas.**

<b>Caso de uso: Planificar pruebas.</b>	
<b>Actores:</b> Jefe de prueba	
<b>Propósito</b>	
Lograr una planificación de los recursos involucrados en las pruebas así como una estimación del tiempo.	
<b>Resumen</b>	
El Jefe de prueba realiza la planificación teniendo en cuenta la complejidad de la prueba y los recursos en disposición en ese momento, una vez terminada la planificación se enviara una confirmación de planificación terminada.	
<b>Precondiciones</b>	
Se encuentren insertadas las pruebas a planificar así como los proyectos.	
<b>Referencias:</b> R 5.1, R 5.2, R 5.3	
<b>Casos de uso asociados:</b>	
<b>Postcondiciones</b>	
Se planifican los recursos involucrados en las pruebas y se notifique a los interesados en la planificación.	
<b>Requerimientos especiales:</b>	
<b>Prototipo:</b>	
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1. El jefe de prueba selecciona planificar.	2. El sistema le muestra una interfaz donde se escoge el proyecto que va a ser probado y el Jefe de prueba que realizara la planificación.
3. El jefe de prueba selecciona el proyecto que va a ser probado y su nombre del listado de los jefes prueba.	4. Posteriormente el sistema muestra una interfaz con los módulos de ese proyecto.

<p>5. El jefe de prueba selecciona el modulo que va a ser probado en caso de realizar una prueba unitaria, en caso de ser una prueba de integración seleccionaría el nombre del proyecto.</p>	<p>6. El sistema muestra una interfaz donde se muestra el nombre del jefe de prueba, del proyecto y modulo y selecciona el tipo de prueba que desea planificar.</p>
<p>7. El jefe de prueba selecciona la prueba que va a planificar.</p>	<p>6. El sistema muestra la interfaz de planificación, donde se escogen los siguientes datos:</p> <ul style="list-style-type: none"><li>• Cantidad de máquinas</li><li>• Cantidad de estudiantes</li><li>• Turnos de trabajo.</li></ul> <p>Además de mostrar las complejidades para que el asesor seleccione el tipo de complejidad de la prueba a planificar.</p> <ul style="list-style-type: none"><li>• Capítulos con complejidad alta</li><li>• Capítulos con complejidad media</li><li>• Capítulos con complejidad baja</li><li>• Casos de Usos con complejidad alta</li><li>• Casos de Usos con complejidad media</li><li>• Casos de Usos con complejidad baja</li><li>• Complejidad de prueba alta</li><li>• Complejidad de prueba media</li><li>• Complejidad de prueba baja</li></ul>

7. El jefe de prueba introduce los datos en el sistema.	8. El sistema realiza la planificación de los recursos y muestra la misma.
9. Cuando se termina la planificación el jefe de prueba selecciona la opción de enviar confirmación.	10. Ver sección “Enviar Confirmación”
<b>Sección “Enviar Confirmación ”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El jefe de prueba selecciona la opción de enviar confirmación.	2. El sistema muestra la opción de enviar la confirmación.
3. El jefe de prueba acepta el envío de confirmación.	4. El sistema envía confirmación.

**Tabla 2.10 Descripción Caso de Uso Planificar Pruebas.**

<b>Caso de uso: Gestionar planificaciones.</b>
<b>Actores:</b> Gestor
<b>Propósito</b>
Realizar operaciones sobre las planificaciones tales como: búsquedas , modificaciones etc.
<b>Resumen</b>
Con las planificaciones insertadas se podrán realizar búsquedas, se seleccionaran las planificaciones mostradas, para modificarlas, eliminarlas, exportarlas o imprimirlas.
<b>Precondiciones</b>



Que existan planificaciones.	
<b>Referencias:</b> R 6.1, R 6.2, R 6.3, R 6.4, R 6.5, R 6.6, R 6.7.I	
<b>Casos de uso asociados:</b>	
<b>Postcondiciones</b>	
Las planificaciones quedan actualizadas.	
<b>Requerimientos especiales:</b>	
<b>Prototipo:</b>	
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1. El gestor selecciona realizar una búsqueda de las planificaciones hechas.	2. El sistema muestra la interfaz de búsqueda de planificaciones según los siguientes criterios. <ul style="list-style-type: none"> <li>• Nombre del proyecto</li> <li>• Nombre del modulo.</li> <li>• Iteración.</li> <li>• Fecha</li> <li>• Asesor.</li> <li>• Tipo de Prueba</li> </ul>
3. El gestor introduce los criterios que necesita.	4. En caso de que existan planificaciones que cumplan con los criterios introducidos el sistema muestra el resultado. En caso contrario ver flujo alterno 4.1
2. El gestor selecciona una planificación para realizar algunas de las operaciones <ul style="list-style-type: none"> <li>• Modificar planificaciones.</li> <li>• Exportar planificaciones</li> <li>• Imprimir planificaciones</li> </ul>	5. En dependencia de la selección del gestor se realizan las operaciones: <ul style="list-style-type: none"> <li>• Si desea modificar las planificaciones ver la sección “Modificar planificaciones”.</li> <li>• Si se desea exportar las planificaciones ver la sección “Exportar planificaciones”.</li> </ul>

<ul style="list-style-type: none"> <li>• Eliminar planificación.</li> </ul>	<ul style="list-style-type: none"> <li>• Si se desea imprimir las planificaciones ver la sección “Imprimir planificaciones”.</li> <li>• Si se desea eliminar planificaciones, ver la sección “Eliminar planificaciones”.</li> </ul>
<p><b>Curso Alternativo</b></p>	
	<p>4.1 Si no existen planificaciones que cumpla con los criterios el sistema informa al usuario. Proveniente del flujo básico 4.</p>
<p><b>Sección “Modificar planificaciones ”</b></p>	
<p><b>Acción del Actor</b></p>	<p><b>Respuesta del Sistema</b></p>
<p>1. El gestor selecciona modificar una planificación.</p>	<p>2. El sistema muestra la interfaz de modificar las planificaciones con todos los datos posibles a editar.</p> <ul style="list-style-type: none"> <li>• Cantidad de maquinas</li> <li>• Cantidad de estudiantes</li> <li>• Turnos de trabajo.</li> <li>• Capítulos con complejidad alta</li> <li>• Capítulos con complejidad media</li> <li>• Capítulos con complejidad baja</li> <li>• Casos de Usos con complejidad alta</li> <li>• Casos de Usos con complejidad media</li> <li>• Casos de Usos con complejidad baja</li> <li>• Complejidad de prueba alta</li> <li>• Complejidad de prueba media</li> <li>• Complejidad de prueba baja</li> </ul>

3. El gestor modifica los datos y actualiza la planificación.	4. El sistema muestra la nueva planificación.
<b>Sección “Exportar planificaciones ”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El gestor selecciona ver la vista de impresión.	2. El sistema muestra una ventana con la planificación en formato pdf.
3. El gestor selecciona el lugar donde desea exportar y presiona aceptar.	4. El sistema exporta la planificación.
<b>Sección “Imprimir planificación ”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El gestor selecciona ver la vista de impresión.	2. El sistema muestra una ventana con la planificación en formato pdf.
3. El gestor selecciona imprimir.	4. El sistema imprime la planificación.
<b>Sección “Eliminar planificación ”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El gestor selecciona eliminar una planificación.	2. El sistema muestra un mensaje de confirmación si desea eliminar la planificación.
3. Si el gestor desea eliminar la planificación acepta el mensaje sino ver flujo alterno 3.1	4. El sistema elimina la planificación.
<b>Curso Alternativo</b>	
3.1 Si el gestor no desea eliminar la planificación cancela el mensaje. Proveniente del flujo 3.	3.2 El sistema muestra la interfaz de búsqueda.

**Tabla 2.11 Descripción Caso de Uso Gestionar Planificaciones.**

Para consultar las descripciones de los casos de uso Gestionar Usuario y Autenticar usuario ver [Anexo 2](#).

**Conclusiones**

Como se puede apreciar en este capítulo se da la idea del sistema que se propone. Con una explicación clara de sus requisitos y descripciones de sus casos de uso. Además de conocer los actores y trabajadores involucrados en el sistema.

## Capítulo 3 Análisis y Diseño del sistema.

### Introducción

En este capítulo se analizan los casos de uso del sistema para diseñar las clases que se implementarán, se representan los diagramas de secuencia del diseño, el diagrama de las clases diseñadas con sus relaciones, los principios utilizados para el diseño de dichas clases, el diagrama de clases persistentes, el diagrama entidad relación y las descripciones de las tablas de la base de datos.

### 3.1 Análisis.

Durante el análisis, analizamos 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, es importante hacer notar que el análisis hace abstracciones y evita resolver algunos problemas y tratar algunos requisitos que serán pospuestos al diseño y la implementación. [21]

#### 3.1.1 Modelo de análisis

El Modelo de análisis es una jerarquía de paquetes del análisis que contienen clases del análisis y realizaciones de casos de uso, además ofrece una especificación más precisa de los requisitos, sin dejar de mencionar que en el modelo de análisis se describe utilizando el lenguaje de los desarrolladores, puede por tanto introducir un mayor formalismo y ser utilizado para razonar sobre los funcionamientos internos del sistema.

El modelo de análisis estructura los requisitos de un modo que facilita su comprensión, su preparación, su modificación y en general su mantenimiento; además se puede considerar como la primera aproximación al modelo de diseño. [22]

#### 3.1.2 Diagrama de clases de análisis.

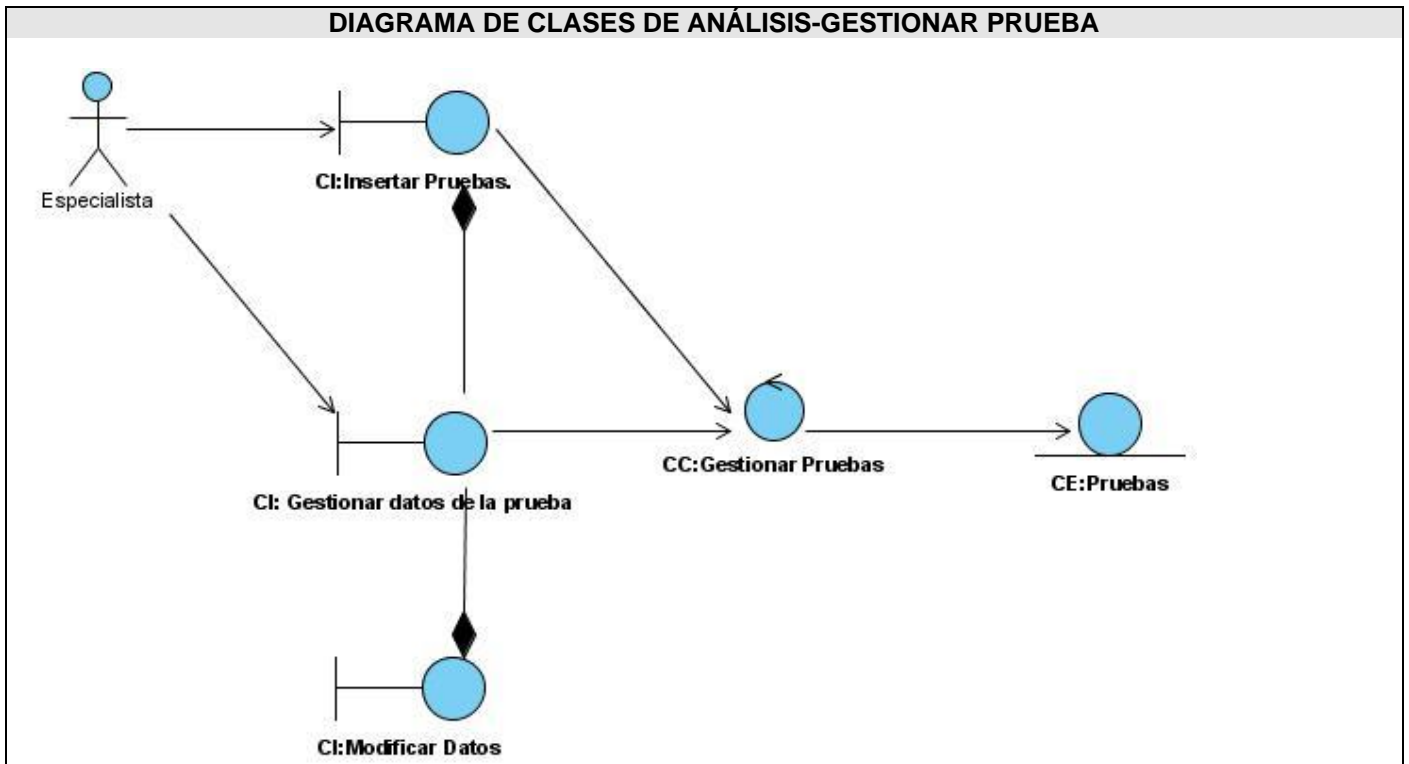


Figura 3.1 Diagrama de clases del análisis del Caso de Uso Gestionar Prueba.

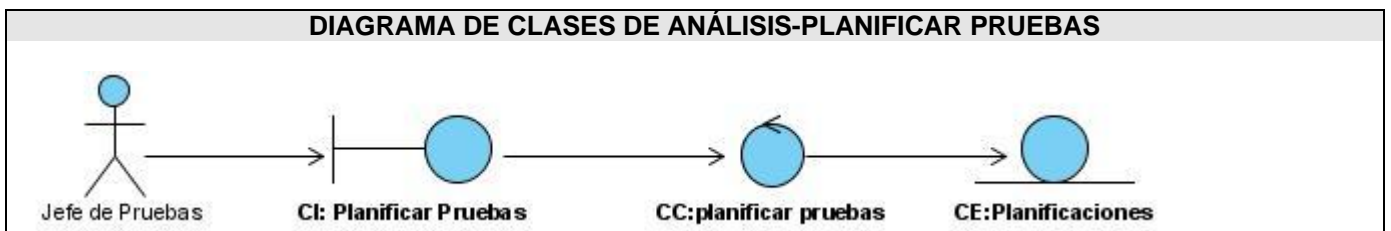


Figura 3.2 Diagrama de clases del análisis del Caso de Uso Planificar Pruebas.

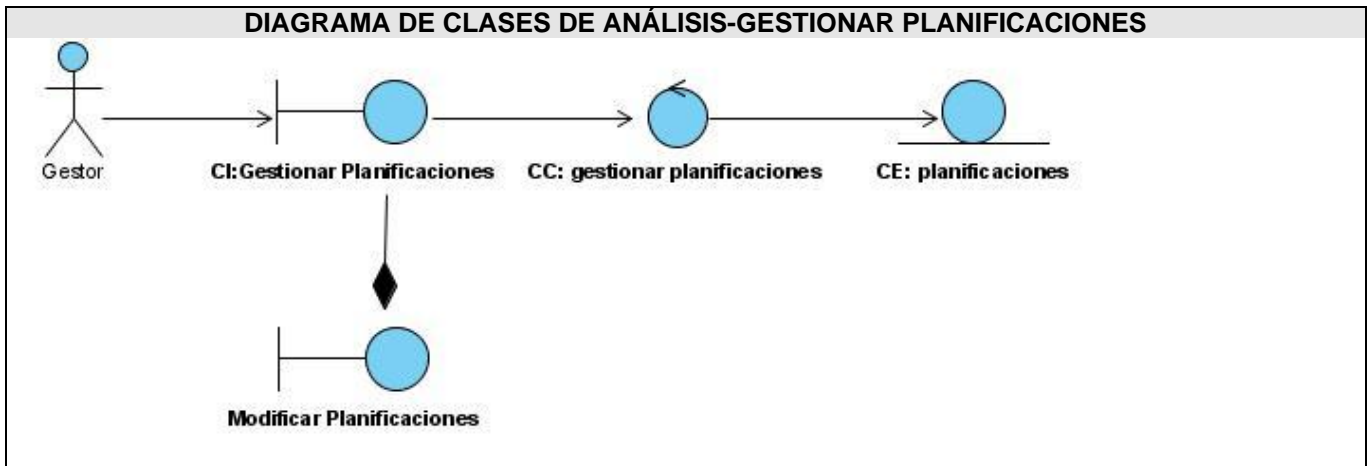


Figura 3.3 Diagrama de clases del análisis del Caso de Uso Gestionar Planificaciones.

### 3.2 Diseño.

En el diseño se modela el sistema y se encuentra su forma (incluida su arquitectura) para que soporte todos los requisitos, incluyendo los requisitos no funcionales y otras restricciones que se le suponen. Una entrada esencial en el diseño es el resultado del análisis. El propósito principal es: adquirir una comprensión en profundidad de los aspectos relacionado con los requisitos funcionales y no funcionales y restricciones asociadas a la programación, sistemas operativos, y tecnologías de interfaz de usuarios. [23] UML posee una extensión para el modelado de aplicaciones Web, dicha extensión es usada para el diseño de las clases. Los estereotipos que usa esta extensión son:



<<Server page>>Representa la página Web que tiene código que se ejecuta en el servidor. Este código interactúa con recursos en el servidor. Las operaciones representan las funciones del código y los atributos las variables visibles dentro del alcance de la página



<<Client Page>> Una instancia de Página Cliente es una página Web, con formato HTML; mezcla de datos, presentación y lógica. Son interpretadas por el browser. Cada página cliente solo puede ser construida por una página servidor.



<<Form>> Grupo de elementos de entrada que son parte de una página cliente. Se relaciona directamente con la etiqueta de igual nombre del HTML. Sus atributos son los elementos de entrada del formulario (Text Field, Text Area, Button, Label, Radio Button, Radio Group, Select, Check Box y HiddenFields).

### 3.2.1 Arquitectura Web de 3 niveles.

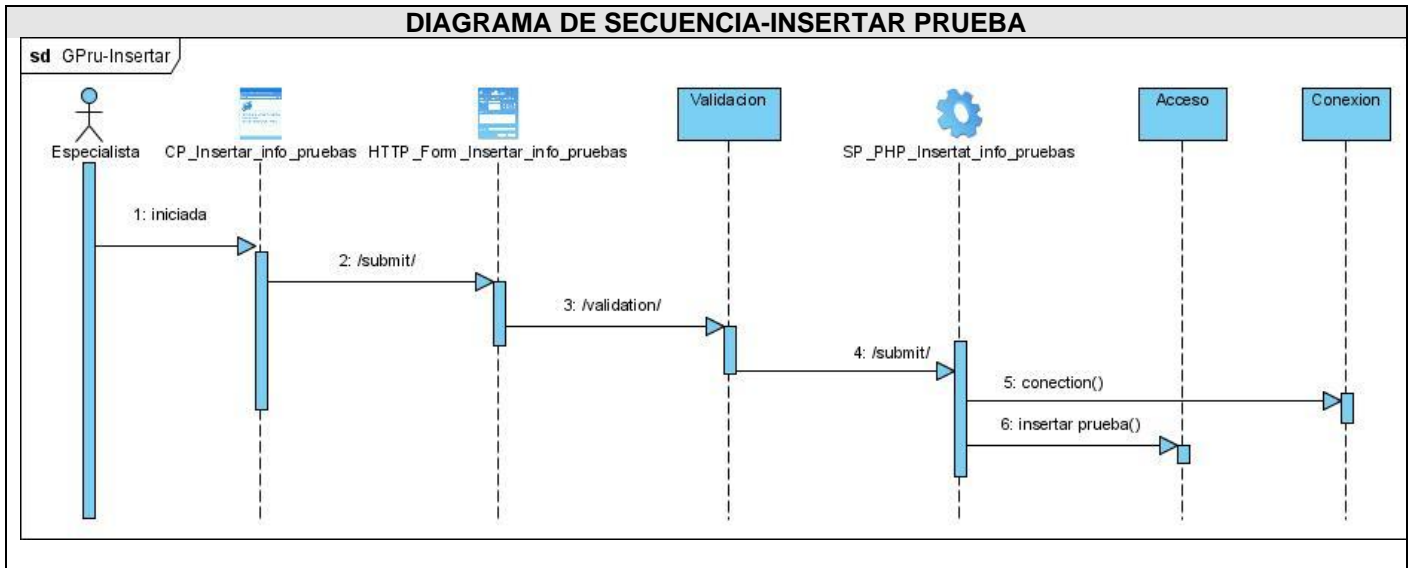
La arquitectura de las aplicaciones Web suelen presentar un esquema de tres niveles. El primer nivel consiste en la capa de presentación que incluye no solo el navegador, sino también el servidor Web que es el responsable de dar a los datos un formato adecuado. El segundo nivel está referido habitualmente a algún tipo de programa o script. Finalmente, el tercer nivel proporciona al segundo los datos necesarios para su ejecución.

Una aplicación Web típica recogerá datos del usuario (primer nivel), los enviará al servidor, que ejecutará un programa (segundo y tercer nivel) y cuyo resultado será formateado y presentado al usuario en el navegador (primer nivel otra vez).





### 3.2.2 Diagrama de secuencia del diseño.



**Figura 3.4 Diagrama de secuencia del diseño, escenario del Caso de Uso Gestionar Prueba (Insertar Prueba).**

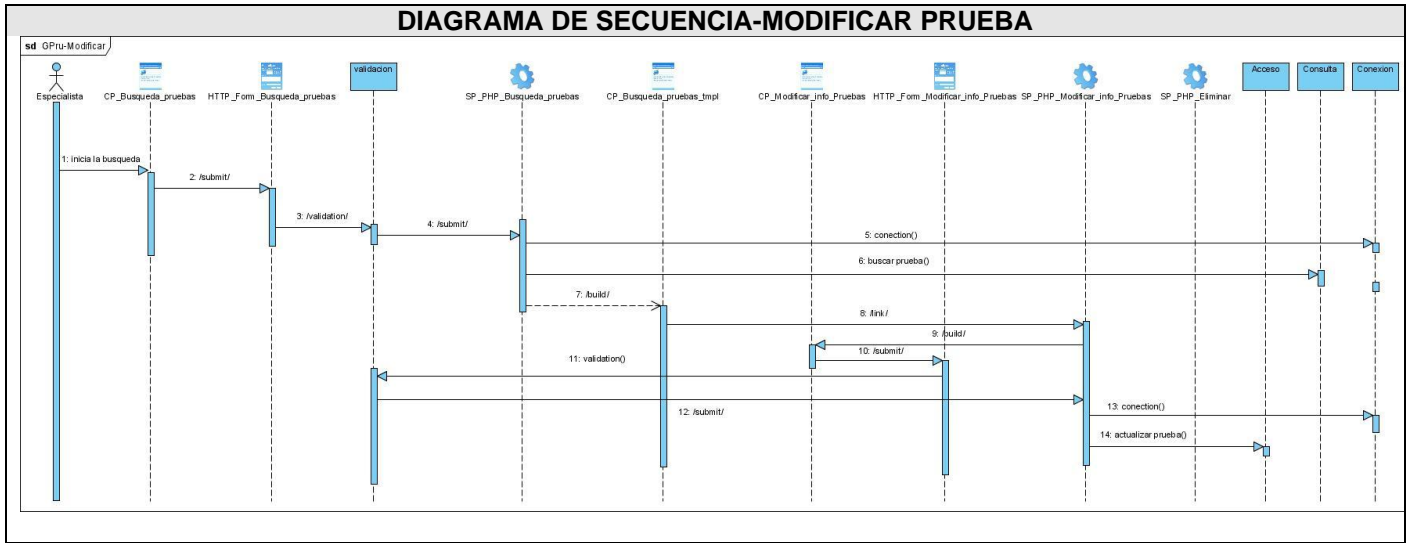


Figura 3.5 Diagrama de secuencia del diseño, escenario del Caso de Uso Gestionar Prueba (Modificar Prueba).

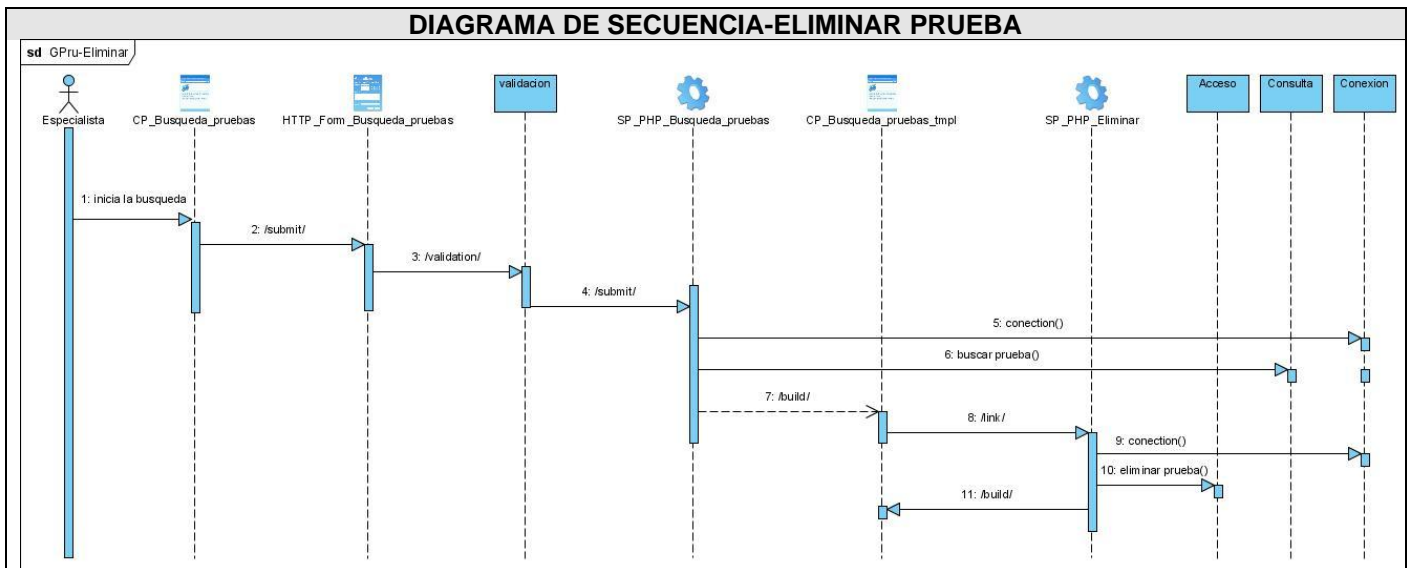


Figura 3.6 Diagrama de secuencia del diseño, escenario del Caso de Uso Gestionar Prueba (Eliminar Prueba).

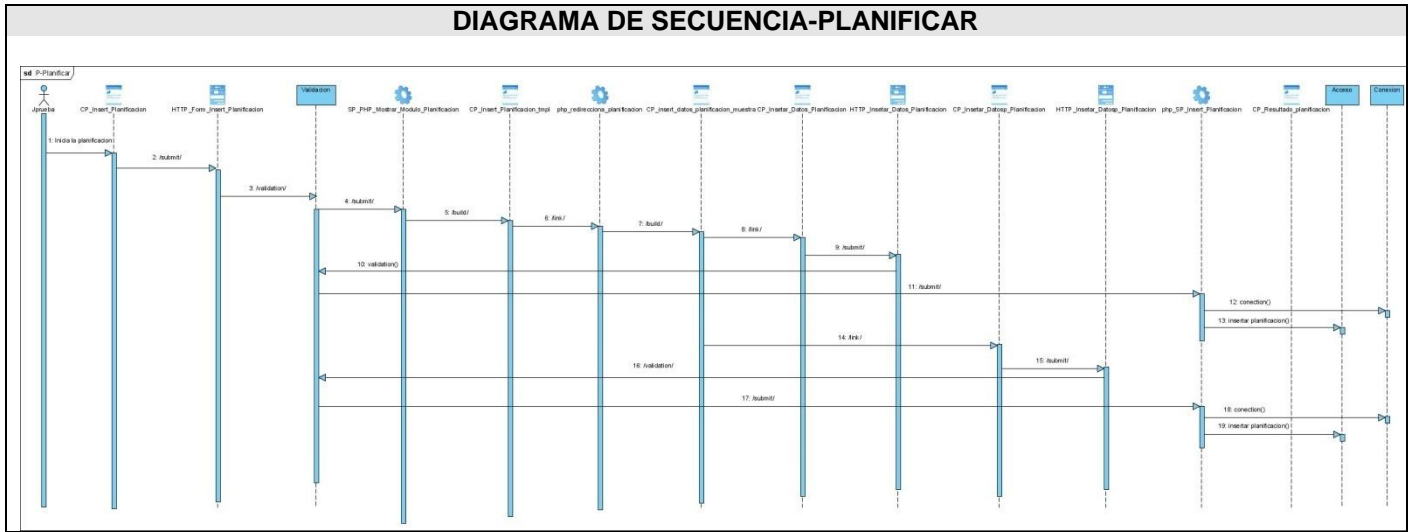
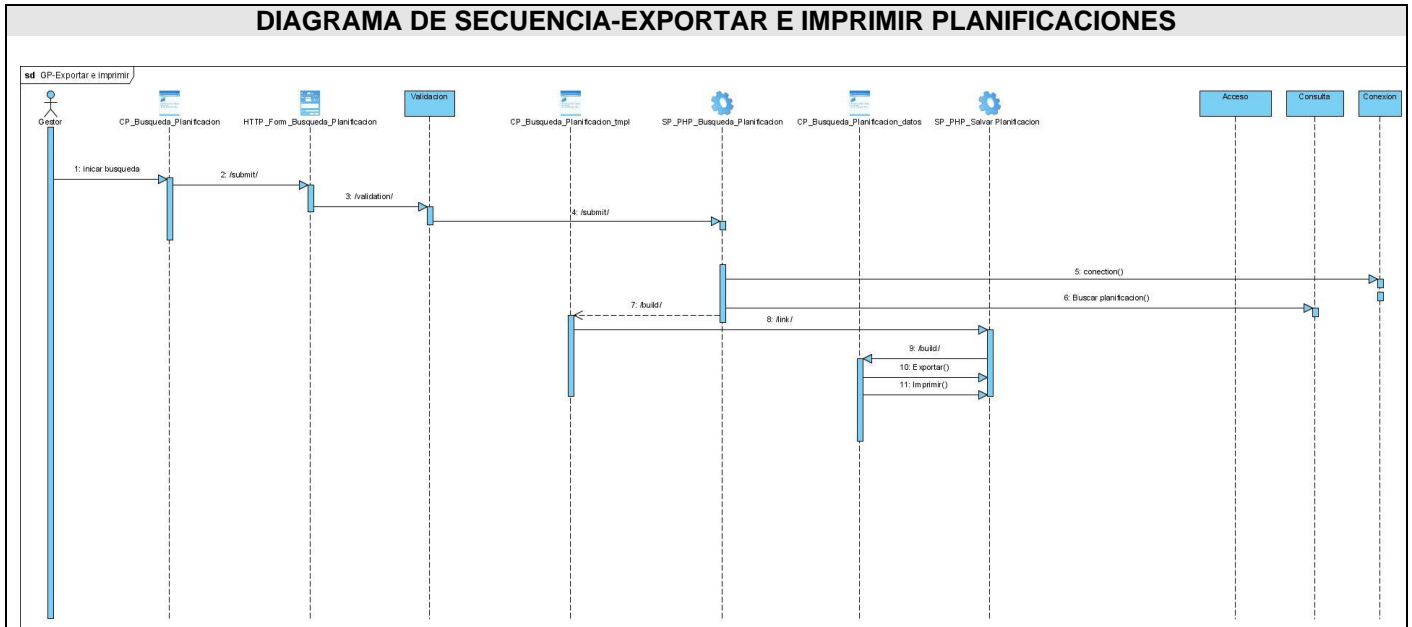


Figura 3.7 Diagrama de secuencia del diseño, Caso de Uso Planificar Prueba.



**Figura 3.8 Diagrama de secuencia del diseño, escenario del Caso de Uso Gestionar Planificaciones (Exportar e imprimir planificaciones).**

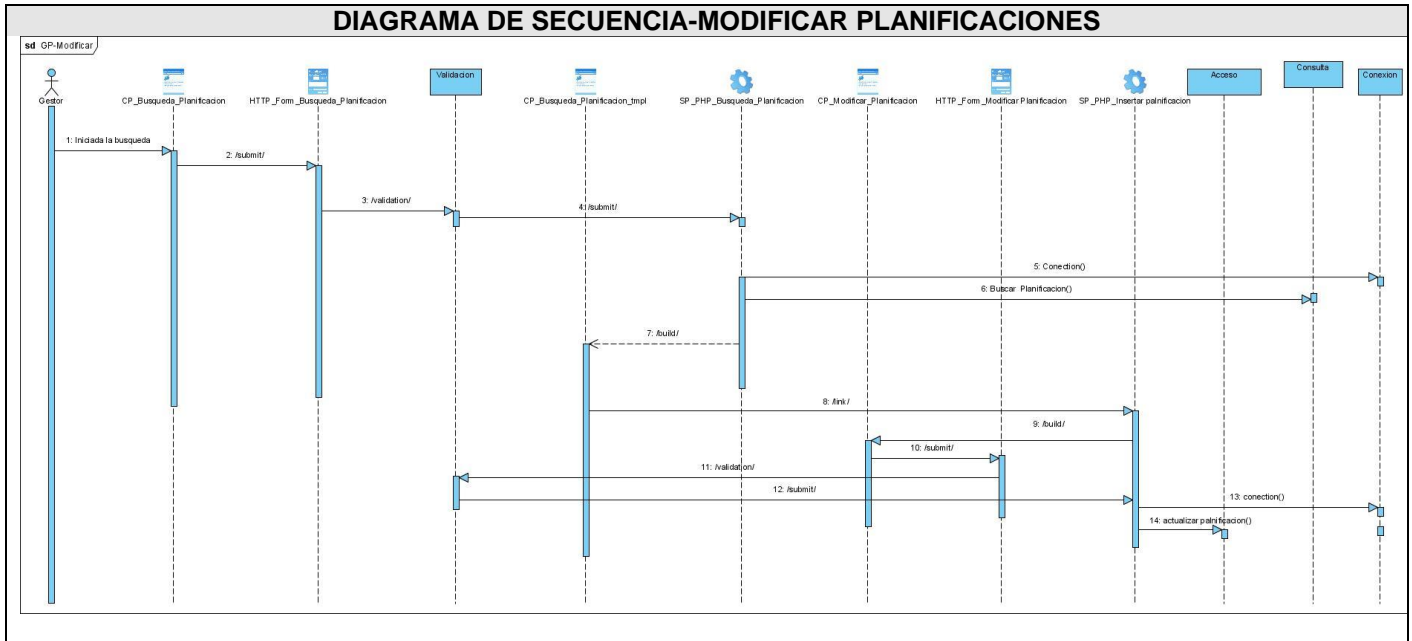


Figura 3.9 Diagrama de secuencia del diseño, escenario del Caso de Uso Gestionar Planificaciones (Modificar planificaciones).

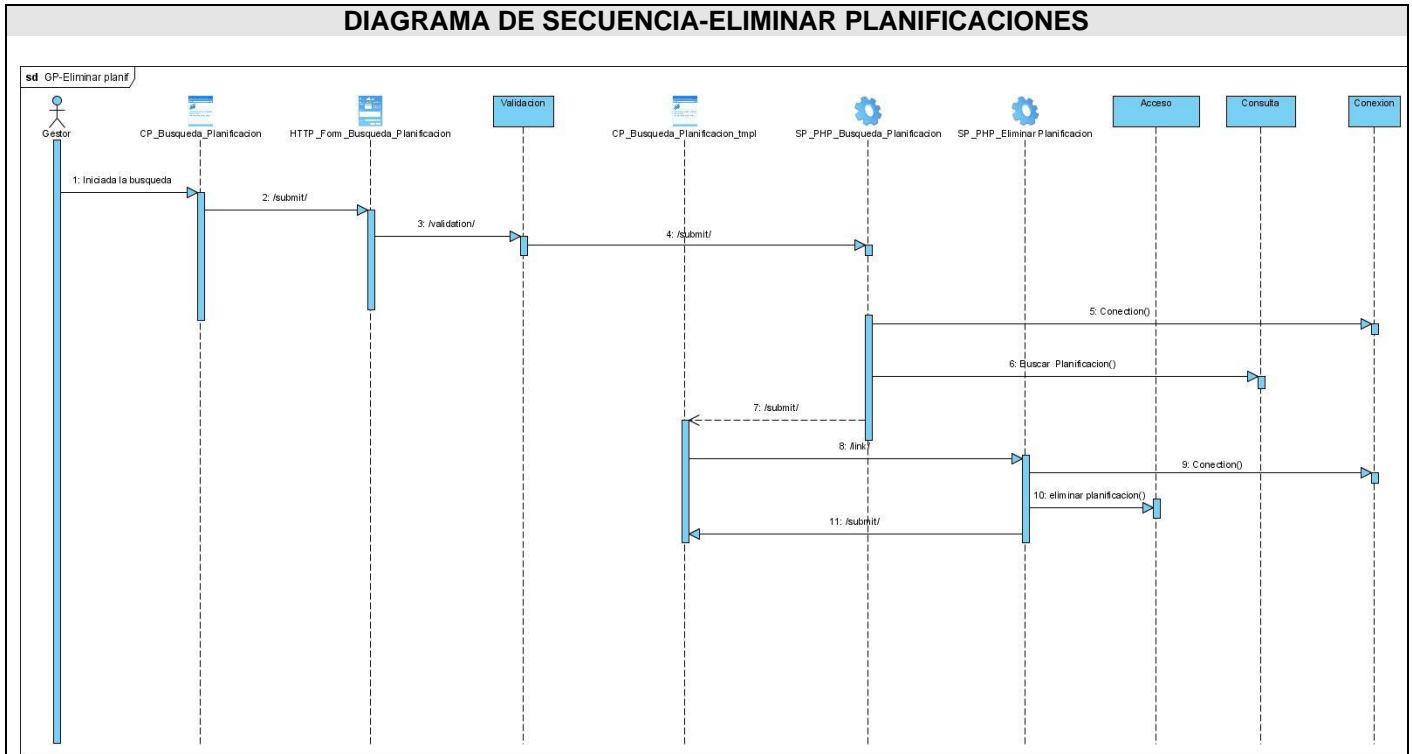


Figura 3.10 Diagrama de secuencia del diseño, escenario del Caso de Uso Gestionar Planificaciones (Eliminar planificaciones).

### 3.2.3 Diagrama de clases del diseño.

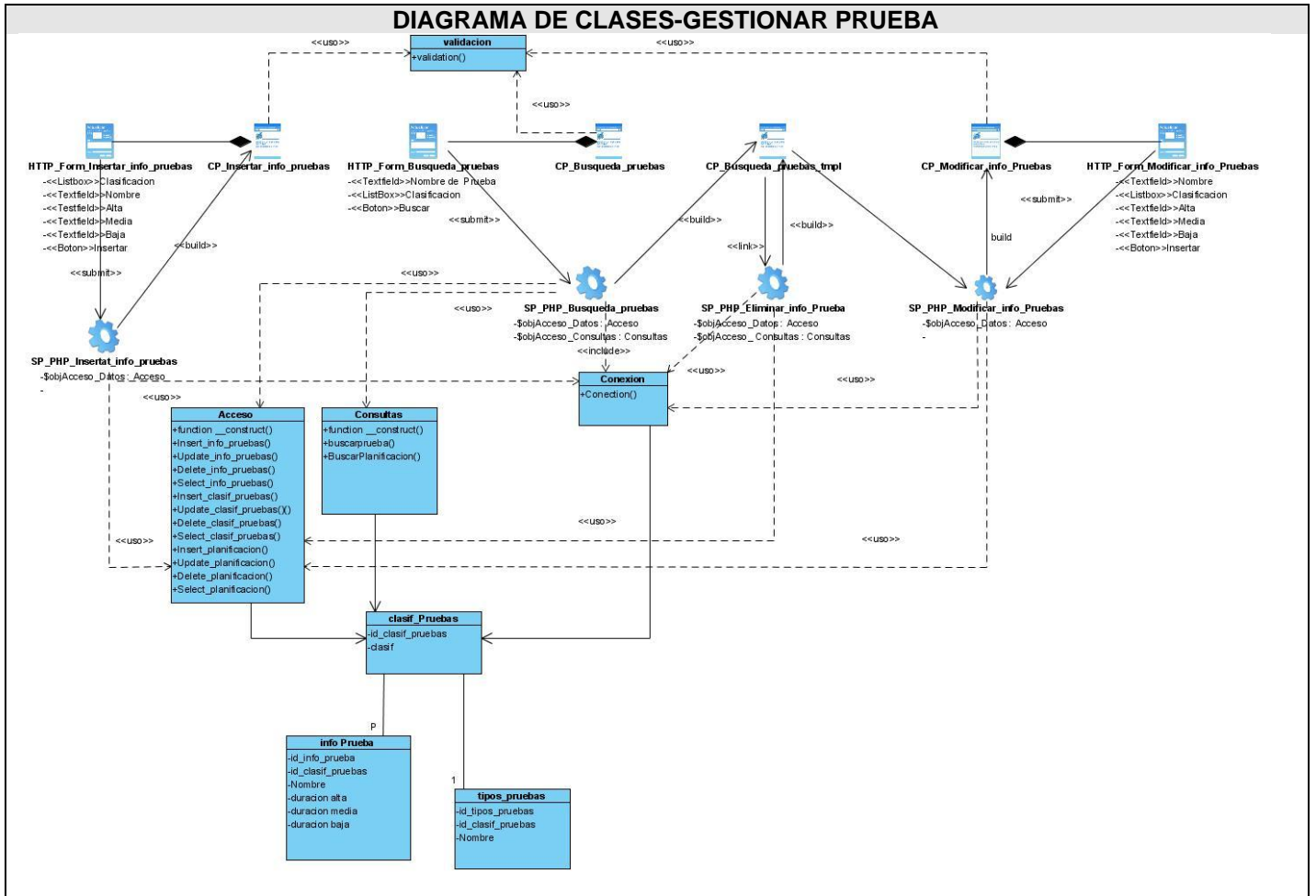


Figura 3.11 Diagrama de clases del diseño Caso de Uso Gestionar Prueba.

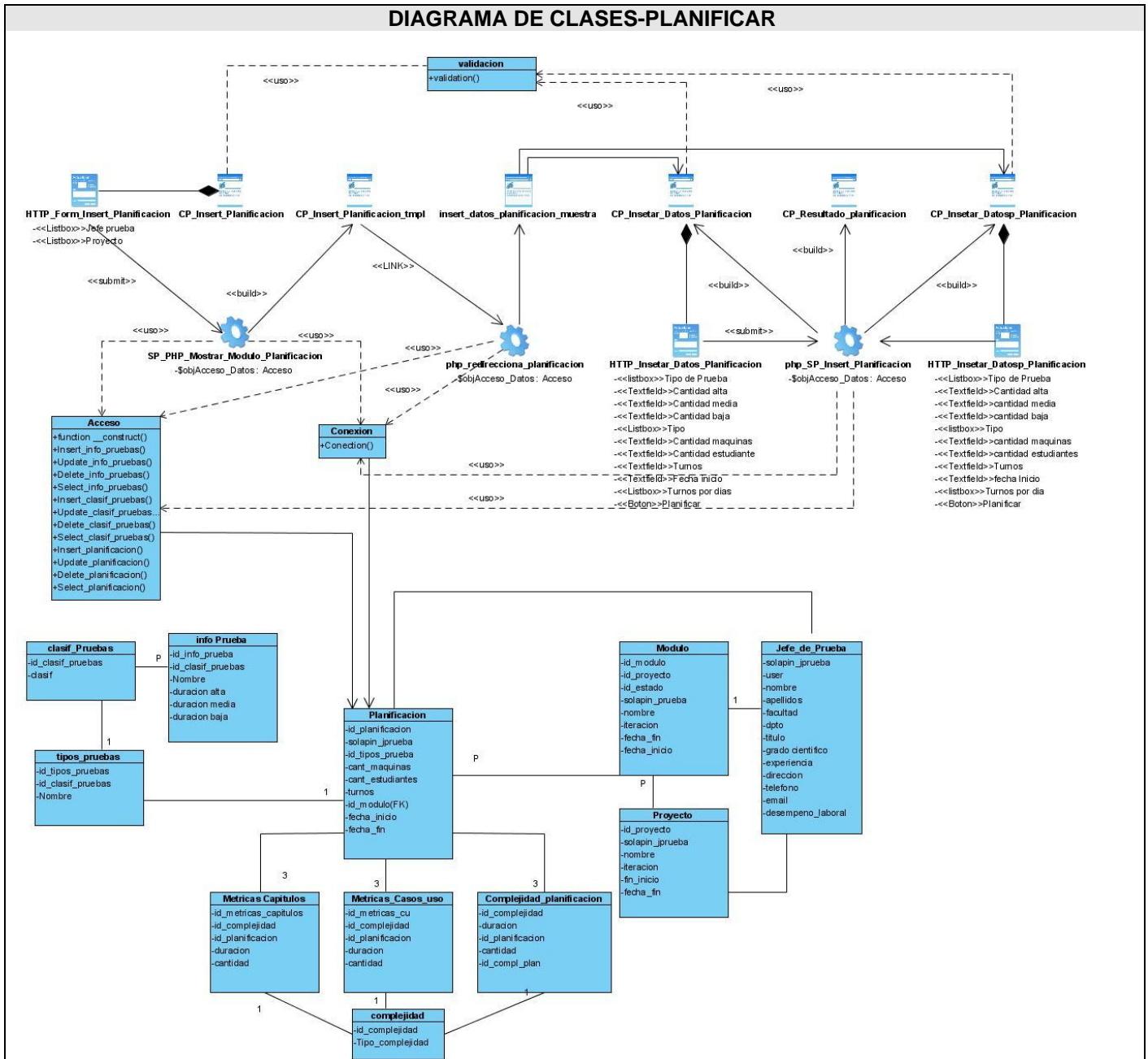


Figura 3.12 Diagrama de clases del diseño Caso de Uso Planificar Prueba.



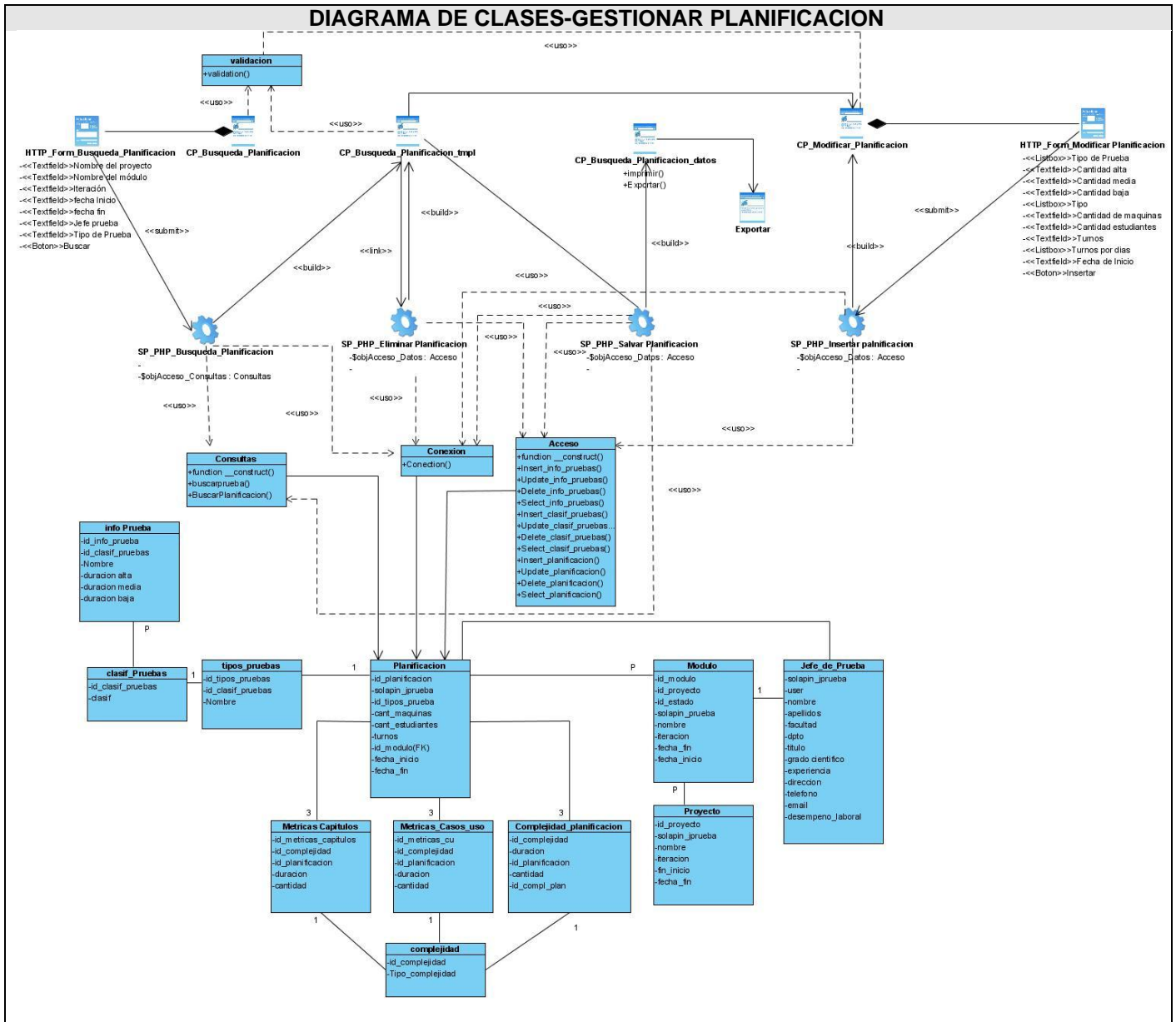


Figura 3.13 Diagrama de clases del diseño Caso de Uso Gestionar Planificación.

### 3.3 Diseño de la Base de Datos.

Para ver el diagrama de clases persistentes ver [Anexo 3](#).

#### 3.3.1 Diagrama Entidad Relación.

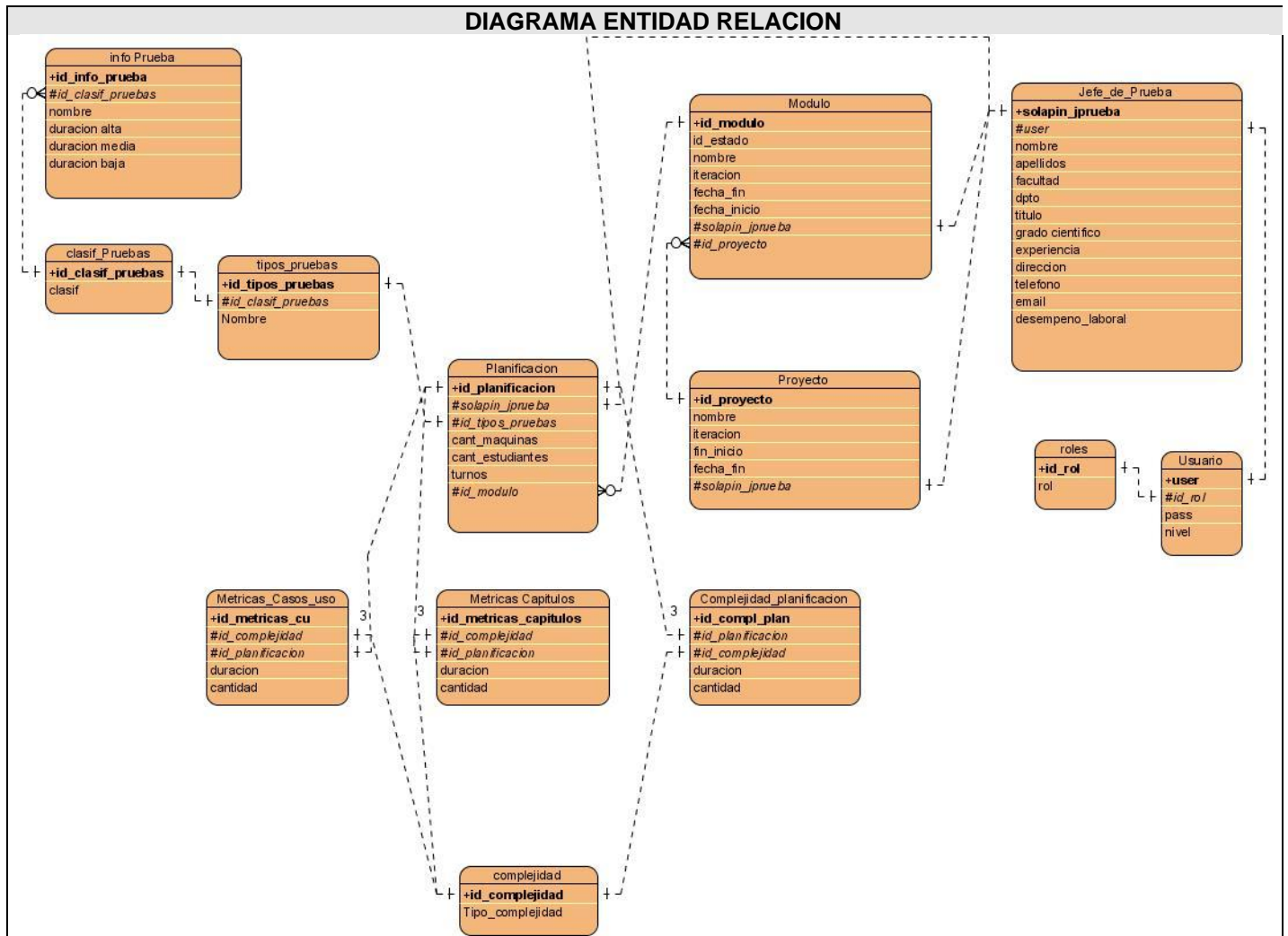


Figura 3.14 Diagrama de Entidad Relación.

Para ver la descripción de las tablas de la base de datos consultar [Anexo 4](#).

### 3.4 Definiciones de diseño que se aplican.

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:



**Eliminar.**



**Modificar.**

**Figura 3.15 Iconos eliminar y modificar del sistema.**

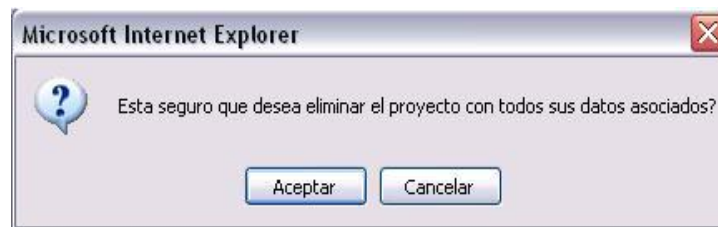
### 3.5 Tratamiento de errores.

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 JavaScripts.

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 3.16 Mensaje de verificación. Eliminar Proyecto.**



**Figura 3.17 Mensaje de validación. Campo Nombre al modificar una Prueba.**



**Figura 3.18 Mensaje de validación. Campo Métricas al modificar el valor.**

### 3.6 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 sólo para aquellos que pueden tener acceso a ellas.

A continuación se muestran algunos de estos mensajes:



**Figura 3.19 Mensaje de verificación. Campo Contraseñas al crear un usuario.**



**Figura 3.20 Interfaz de autenticación. Al introducir usuario y contraseña incorrecta.**

### 3.7 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 este capítulo realizamos todo el análisis y el diseño del sistema donde dejamos todo listo para el próximo flujo de implementación. Se diseñó una base de datos donde se garantiza el cumplimiento de todos los requisitos funcionales, además de abordar en temas importantes como el de la seguridad.

## **Capítulo 4 Implementación y Prueba.**

### **Introducción**

En este capítulo se reflejan los diferentes diagramas correspondientes a este flujo de trabajo, donde se implementará la propuesta realizada en el análisis y diseño, se representa el diagrama de despliegue, los de componentes y el modelo de prueba correspondiente a la aplicación desarrollada.

### **4.1 Implementación.**

En la implementación se empieza con el resultado del análisis y diseño, y se implementa el sistema en términos de componentes, es decir: ficheros de código fuente, scripts, ficheros de códigos binarios y ejecutables. Dentro de los principales objetivos se encuentran: Distribuir el sistema asignando componentes ejecutables a nodos en el diagrama de despliegue e Implementar las clases y subsistemas encontrados en el diseño, entre otros. [24]

### 4.1.1 Diagrama de despliegue.

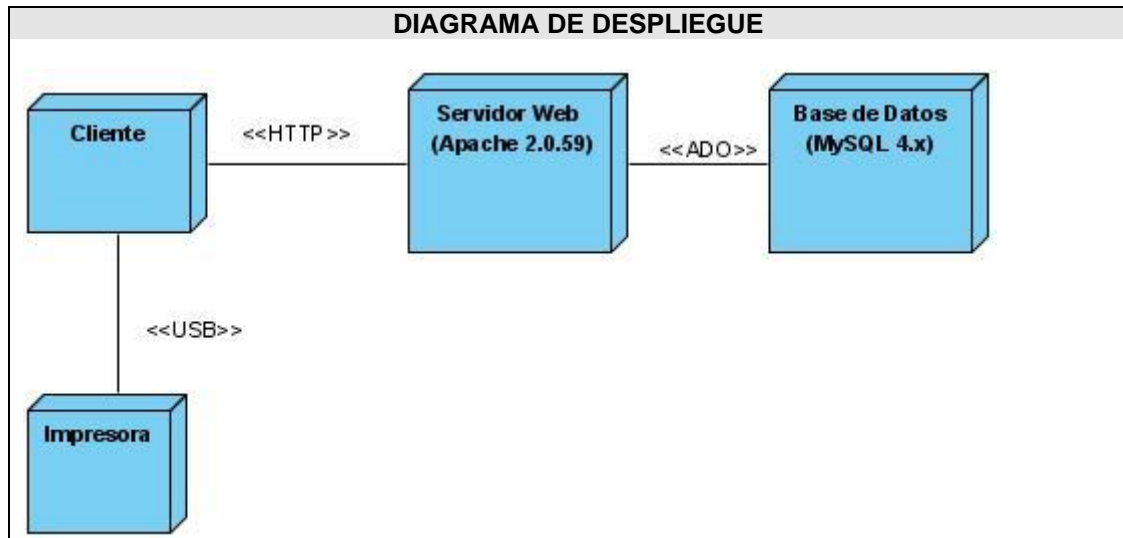


Figura 4.1 Diagrama de despliegue.



### 4.1.2 Diagramas de componentes.

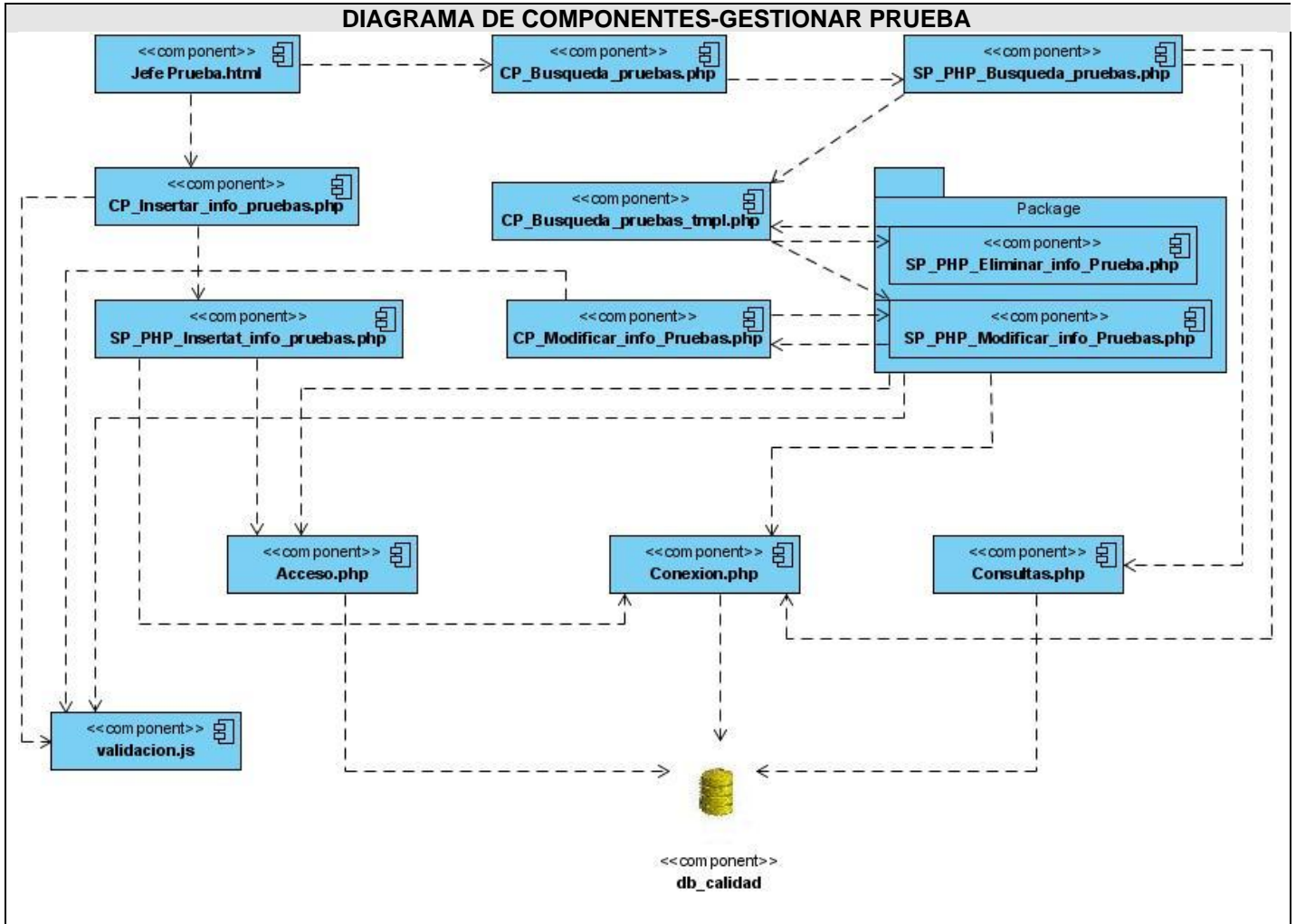


Figura 4.2 Diagrama de componente Caso de Uso Gestionar Prueba.

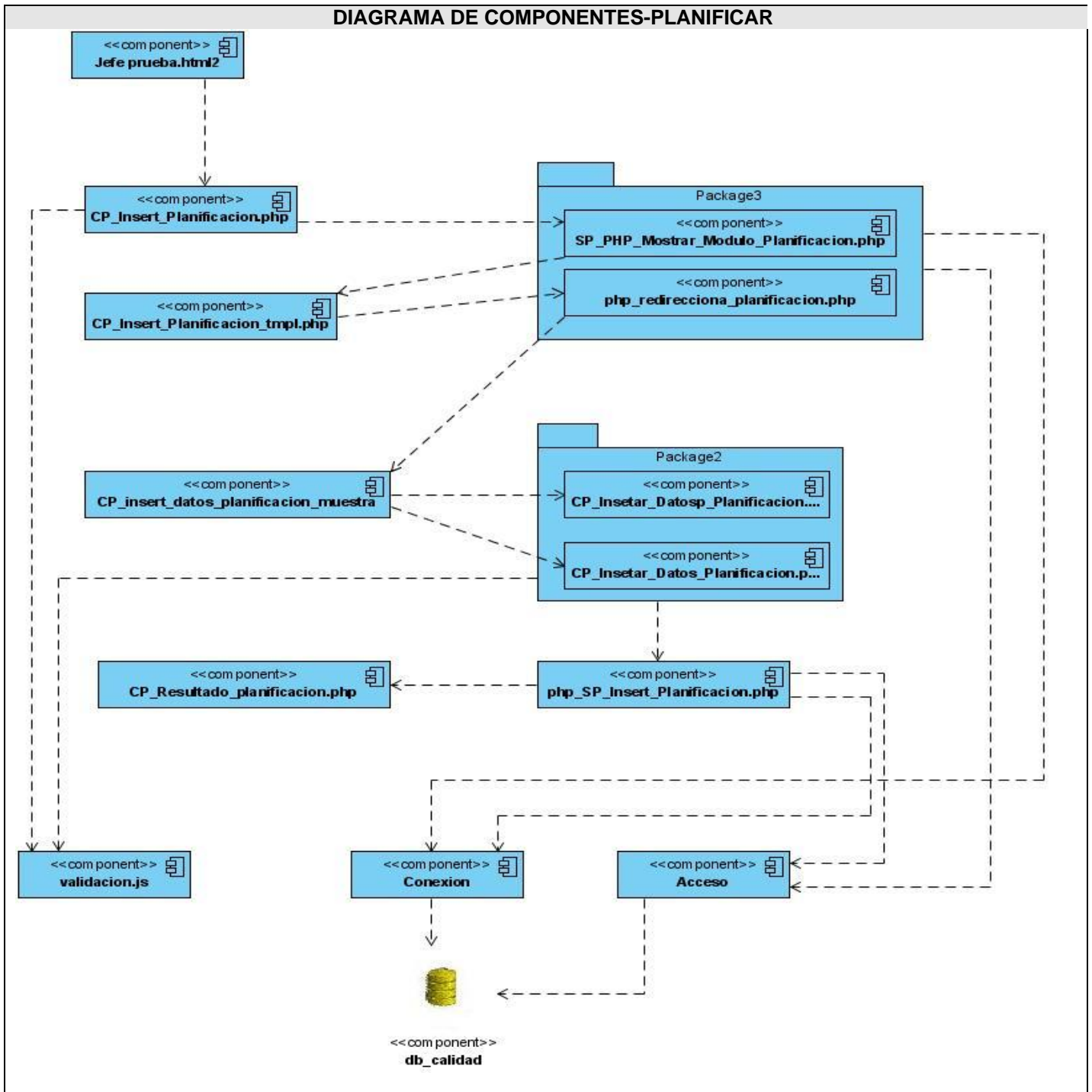


Figura 4.3 Diagrama de componente Caso de Uso Planificar Prueba.

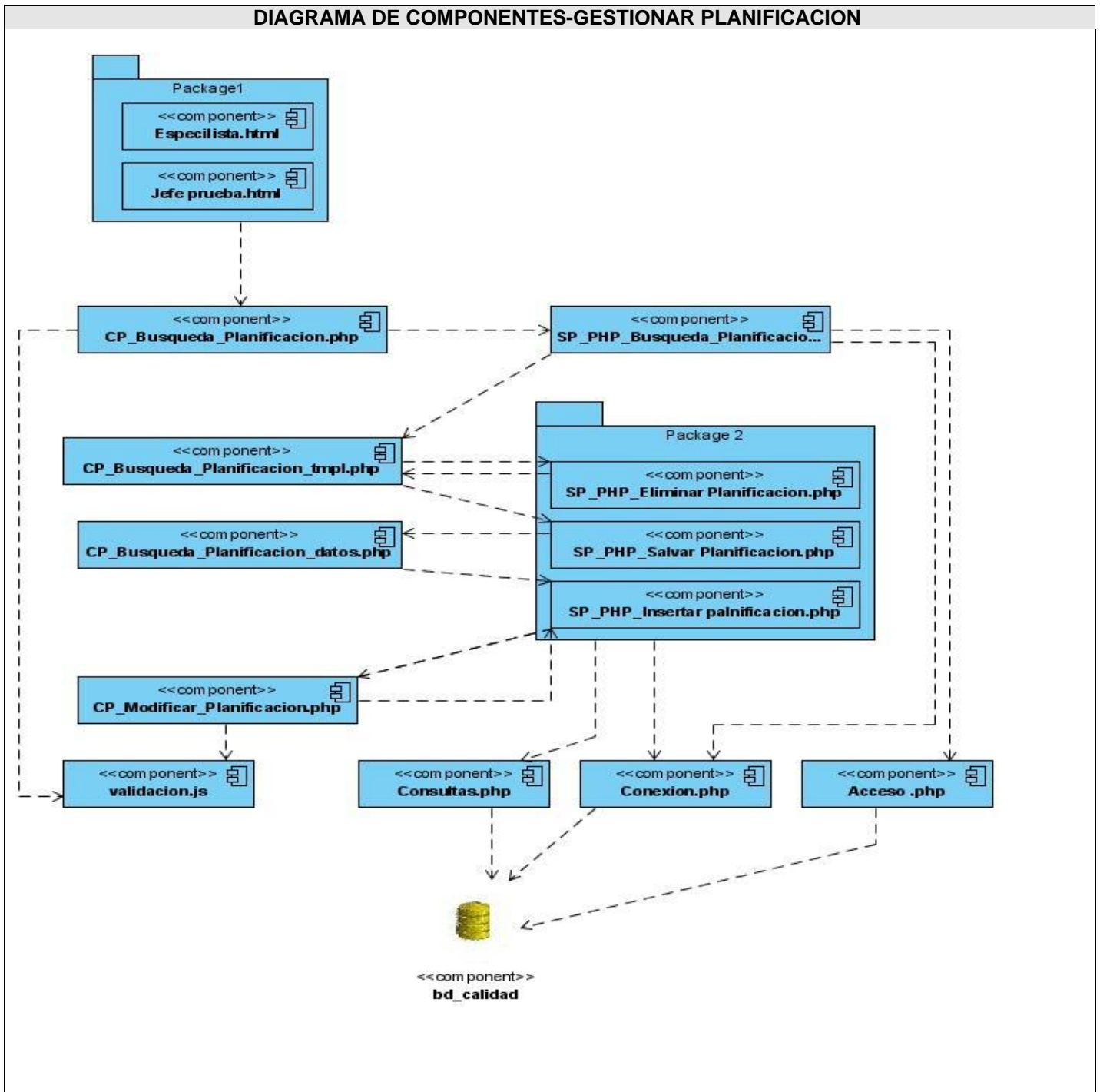


Figura 4.4 Diagrama de componente Caso de Uso Gestionar Planificación.

## 4.2 Modelo de Prueba.

### 4.2.1 Descripción caso de prueba de integración.

#### 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</li> </ul>	<ul style="list-style-type: none"> <li>• El administrador debió haber asignado el rol de especialista al usuario.</li> </ul>

	<p>permisos.</p> <p>Planificación:</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>No conformidades:</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>Seguimiento:</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>Planificación:</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>No conformidades:</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>Seguimiento:</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>
----------------------------------	--	---

**Tabla 4.1 Caso de Prueba de Integración para el módulo Planificación.**

**No conformidades.**

<b>Entrada</b>	<b>Resultados</b>	<b>Condiciones</b>
<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>Planificación:</p>	<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 Datos del Proyecto.</li> <li>○ Gestionar Métricas.</li> <li>○ Gestionar Planificaciones.</li> <li>○ Gestionar Pruebas</li> <li>○ Insertar Pruebas.</li> </ul> <p>No conformidades:</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>Seguimiento:</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>Planificación:</p>	<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>○ Gestionar Datos del Proyecto.</li> <li>○ Gestionar Planificaciones.</li> <li>○ Insertar datos del proyecto.</li> <li>○ Planificar Pruebas.</li> </ul> <p>No conformidades:</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>Seguimiento:</p> <ul style="list-style-type: none"> <li>○ Gestionar datos de Probador.</li> <li>○ Evaluar Probador.</li> <li>○ Gestionar PC.</li> </ul>	
--	---	--

**Tabla 4.2 Caso de Prueba de Integración para el módulo No Conformidades.**

**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> <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> </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 hizo en el modulo No Conformidades.</li> </ul>
<p>Se selecciona gestionar PC.</p>	<p>Muestra la interfaz para seleccionar la planificación.</p>	<p>Se debe de haber realizado planificación en el modulo Planificación.</p>
<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>Planificación:</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>No Conformidades:</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</li> </ul>	<ul style="list-style-type: none"> <li>•El administrador debió haber asignado el rol de especialista al usuario.</li> </ul>

	<p>proyectos.</p> <p>Seguimiento:</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>	
<p>•Un usuario se autentica por el Rol de Jefe de Prueba.</p>	<p>•Se muestra la interfaz del Jefe de prueba con sus permisos.</p> <p>Planificación:</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>No conformidades:</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</li> </ul>	<p>•El administrador o el Especialista debieron haber asignado el rol de Jefe de prueba al usuario.</p>

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

**Tabla 4.3 Caso de Prueba de Integración para el módulo Seguimiento.**

### **Conclusiones**

En este capítulo se realizó toda la implementación del sistema y se mostraron los diagramas de despliegue y componentes, con los casos de pruebas elaborados se comprobó la integración de los módulos del sistema.

## Conclusiones

- Con la definición del proceso anteriormente expuesto se garantizará una buena planificación de los recursos de un laboratorio de calidad, además tendrá como resultado la planificación de todas las pruebas a realizar a un software en la Universidad de Ciencias Informáticas.
- El desarrollo de la herramienta informática, 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, posibilitará dar respuesta a las solicitudes de los clientes en el menor tiempo posible.
- El presente Trabajo de Diploma se presentó en la Jornada Científica Estudiantil de este curso a nivel de facultad y universidad obteniendo la condición de Relevante y Destacado respectivamente.

## Recomendaciones

- 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.
- Profundizar en el estudio de las tendencias y tecnologías actuales para determinar con mayor exactitud el basamento teórico de las nuevas herramientas a desarrollar.
- 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.
- 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.

## Referencia Bibliográfica

- [1]. Definición.org. [En línea] febrero de 2007. <http://www.definicion.org/planificacion>
- [2]. Definición.org. [En línea] febrero de 2007. <http://www.definicion.org/estimacion>
- [3]. Swag. [En línea] noviembre de 2006. [http://www.labqa.com/100\\_quienes.html](http://www.labqa.com/100_quienes.html)
- [4]. Swag. [En línea] noviembre de 2006. [http://www.labqa.com/312a\\_gestorProyectos.html](http://www.labqa.com/312a_gestorProyectos.html)
- [5]. Projectkickstart. [En línea] noviembre de 2006. <http://www.projectkickstart.com/html/spanish.htm>
- [6]. Microsoft. [Online] enero, 2007.  
<http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art20.asp>
- [7]. Molpeceres, Alberto. Proceso de Desarrollo RUP, XP, FDD. [En línea] noviembre de 2006.  
<http://www.willydev.net/descargas/articulos/general/cualxpfddrup.PDF>
- [8]. UML. [En línea] enero de 2007. <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/x40.html>
- [9]. Grupo Alarcos. [En línea] enero de 2007. [http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/LabTr1\\_VP.pdf](http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/LabTr1_VP.pdf)
- [10]. CSI. [En línea] noviembre de 2006. <http://www.csi.map.es/csi/silice/Global71.html>
- [11]. Pérez, José Angel Bustos. [En línea] noviembre de 2006.  
[http://www.augcyl.org/glol/old/N\\_1/apache.html](http://www.augcyl.org/glol/old/N_1/apache.html)
- [12]. Esepestudio. [En línea] noviembre de 2006. <http://www.espestudio.com/articulo/desarrollo-web/bases-de-datos-mysql/Que-es-MySQL.htm>
- [13]. Wikipedia. [En línea] enero de 2007. [http://es.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](http://es.wikipedia.org/wiki/Microsoft_SQL_Server)
- [14]. Ciberaula. [En línea] enero de 2007. [http://php.ciberaula.com/articulo/introduccion\\_php](http://php.ciberaula.com/articulo/introduccion_php)
- [15]. Maestro de la Web. [En línea] Noviembre de 2006. <http://www.maestrosdelweb.com/editorial/phpintro>
- [16]. Telyse. [En línea] diciembre de 2007.  
<http://www.telyse.net/telyse/areatecnica/manuales/js/capitulo1.html>
- [17]. Hooping. [En línea] diciembre de 2007. <http://www.hooping.net/faq/html.htm>



- [18]. Msdn. [En línea] enero 2007. [http://msdn2.microsoft.com/es-es/library/fa1h9d0d\(VS.80\).aspx](http://msdn2.microsoft.com/es-es/library/fa1h9d0d(VS.80).aspx)
- [19]. Desarrollo Web. [En línea] noviembre de 2006. <http://www.desarrolloweb.com/articulos/332.php>
- [20]. WAMP. [En línea] febrero de 2007. <http://www.desarrolloweb.com/articulos/1598.php>
- [21]. El proceso Unificado de Desarrollo de Software. [aut. libro] Grady Booch, James Rumbaugh Ivar Jacobson. s.l. : Addison Wesley Longman, 1999, Vol. I, págs. 165-166.
- [22]. El proceso Unificado de Desarrollo de Software. [aut. libro] Grady Booch, James Rumbaugh Ivar Jacobson. s.l. : Addison Wesley Longman, 1999, Vol. I, pág. 173.
- [23]. El Proceso Unificado de Desarrollo de Software. [aut. libro] Grady Booch, James Rumbaugh. Ivar Jacobson. s.l. : Addison Wesley Longman, 199, Vol. I, pág. 205.
- [24]. El proceso unificado de desarrollo de software. [aut. libro] Grady Booch, James Rumbaugh. Ivar Jacobson. s.l. : Addison Wesley Longman, 1999, Vol. I.

## Bibliografía

1. Ciberaula. [En línea] enero de 2007. [http://php.ciberaula.com/articulo/introduccion\\_php](http://php.ciberaula.com/articulo/introduccion_php)
2. CSI. [En línea] noviembre de 2006. <http://www.csi.map.es/csi/silice/Global71.html>
3. Definición.org. [En línea] febrero de 2007. <http://www.definicion.org/planificacion>
4. Definición.org. [En línea] febrero de 2007. <http://www.definicion.org/estimacion>
5. Desarrollo Web. [En línea] noviembre de 2006. <http://www.desarrolloweb.com/articulos/332.php>
6. El proceso Unificado de Desarrollo de Software. [aut. libro] Grady Booch, James Rumbaugh Ivar Jacobson. s.l. : Addison Wesley Longman, 1999, Vol. I, págs. 165-166.
7. El proceso Unificado de Desarrollo de Software. [aut. libro] Grady Booch, James Rumbaugh Ivar Jacobson. s.l. : Addison Wesley Longman, 1999, Vol. I, pág. 173.
8. El Proceso Unificado de Desarrollo de Software. [aut. libro] Grady Booch, James Rumbaugh. Ivar Jacobson. s.l. : Addison Wesley Longman, 199, Vol. I, pág. 205.
9. El proceso unificado de desarrollo de software. [aut. libro] Grady Booch, James Rumbaugh. Ivar Jacobson. s.l. : Addison Wesley Longman, 1999, Vol. I. pág. 260.
10. Esepestudio. [En línea] noviembre de 2006. <http://www.espestudio.com/articulo/desarrollo-web/bases-de-datos-mysql/Que-es-MySQL.htm>
11. Grupo Alarcos. [En línea] enero de 2007. [http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/LabTr1\\_VP.pdf](http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/LabTr1_VP.pdf)
12. Hooping. [En línea] diciembre de 2007. <http://www.hooping.net/faq/html.htm>
13. Maestro de la Web. [En línea] Noviembre de 2006. <http://www.maestrosdelweb.com/editorial/phpintro>
14. Microsoft. [Online] enero, 2007. <http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art20.asp>
15. Msdn. [En línea] enero 2007. [http://msdn2.microsoft.com/es-es/library/fa1h9d0d\(VS.80\).aspx](http://msdn2.microsoft.com/es-es/library/fa1h9d0d(VS.80).aspx)
16. Molpeceres, Alberto. Proceso de Desarrollo RUP, XP, FDD. [En línea] noviembre de 2006. <http://www.willydev.net/descargas/articulos/general/cualxpfdrrup.PDF>

17. Pérez, José Angel Bustos. [En línea] noviembre de 2006.  
[http://www.augcyl.org/glol/old/N\\_1/apache.html](http://www.augcyl.org/glol/old/N_1/apache.html)
18. Proyectkickstart. [En línea] noviembre de 2006. <http://www.projectkickstart.com/html/spanish.htm>
19. Swag. [En línea] noviembre de 2006. [http://www.labqa.com/100\\_quienes.html](http://www.labqa.com/100_quienes.html)
20. Swag. [En línea] noviembre de 2006. [http://www.labqa.com/312a\\_gestorProyectos.html](http://www.labqa.com/312a_gestorProyectos.html)
21. Telyse. [En línea] diciembre de 2007.  
<http://www.telyse.net/telyse/areatecnica/manuales/js/capitulo1.html>
22. UML. [En línea] enero de 2007. <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/x40.html>
23. WAMP. [En línea] febrero de 2007. <http://www.desarrolloweb.com/articulos/1598.php>
24. Wikipedia. [En línea] enero de 2007. [http://es.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](http://es.wikipedia.org/wiki/Microsoft_SQL_Server)

## Glosario de términos

**CMMI:** Capability Maturity Model Integration. Modelo para la mejora o evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software. Fue desarrollado por el Instituto de Ingeniería del Software de la Universidad Carnegie Mellon.

**CUN:** Casos de uso del Negocio.

**CUS:** Casos de uso del sistema

**IEEE:** Corresponde a las siglas de The Institute of Electrical and Electronics Engineers, Instituto de Ingenieros Eléctricos y Electrónicos. Es una asociación internacional sin fines de lucro formada por profesionales de las nuevas tecnologías, como ingenieros de telecomunicaciones, ingenieros electrónicos, Ingenieros en informática.

**ISO:** Organización Internacional para la Estandarización (International Organization for Standardization). Es una organización que ha definido un conjunto de protocolos diferentes, llamados protocolos ISO/OSI. Esta organización de carácter voluntario fue fundada en 1946 y es responsable de la creación de estándares internacionales en muchas áreas, incluyendo la informática y las comunicaciones.

**RF:** Requisitos Funcionales

**RNF:** Requisitos No Funcionales

**SoftlaQ:** Software para la automatización de los procesos de Laboratorios de Calidad de Software:

- Estimación y Planificación de los recursos humanos y tecnológicos.
- Gestion de las No Conformidades.
- Seguimiento de los recursos humanos y tecnológicos.

Desarrollado por estudiantes de la Universidad de las Ciencias Informáticas.

**WWW:** World Wide Web, la Web o WWW, es un sistema de hipertexto que funciona sobre Internet.

## Anexos

### Anexo 1 Definición de los Casos de Uso del Sistema.

CU-1	<b>Gestionar Pruebas</b>
<b>Actor</b>	Especialista
<b>Descripción</b>	Este caso de uso es para cuando se desee insertar una nueva prueba al sistema, se inserta su nombre y se selecciona el tipo de complejidad que posee y se le insertan las métricas, después se realizan búsquedas a una prueba insertada para modificarla o eliminarla.
<b>Referencia</b>	R 1.1 ,R 1.1.1, R 1.2 , R 1.3, R 1.4 , R 1.5 , R 1.6

CU-2	<b>Gestionar Métrica</b>
<b>Actor</b>	Especialista
<b>Descripción</b>	Este caso de uso es el encargado de realizar búsquedas a una métrica insertada para

	modificarla o eliminarla.
<b>Referencia</b>	R 2.1, R 2.2 , R 2.3 , R 2.4 , R 2.5 , R 2.6

CU-3	<b>Insertar datos del proyecto</b>
<b>Actor</b>	Jefe de prueba
<b>Descripción</b>	Este caso de uso es el encargado de insertar los datos del proyecto que va a ser probado así como los datos de los módulos del proyecto.
<b>Referencia</b>	R 3.1, R 3.2

CU-4	<b>Planificar pruebas.</b>
<b>Actor</b>	Jefe de prueba.
<b>Descripción</b>	Este caso de uso se encarga de escoger las pruebas que se van a planificar. Después se encarga de realizar las planificaciones a las pruebas una vez que el jefe de prueba termine la planificación envía una conformación al especialista.

<b>Referencia</b>	<b>R 5.1, R 5.2, R 5.3</b>
-------------------	----------------------------

<b>CU-5</b>	<b>Gestionar datos del proyecto</b>
<b>Actor</b>	Gestor Jefe de prueba Especialista
<b>Descripción</b>	Este caso de uso es el encargado de realizar las búsquedas a datos de proyectos insertados para modificarlos o eliminarlos.
<b>Referencia</b>	<b>R 4.1 , R 4.2 , R 4.3 , R 4.4 , R 4.5</b>

<b>CU-6</b>	<b>Gestionar planificación.</b>
<b>Actor</b>	Gestor Jefe de Prueba Especialista
<b>Descripción</b>	Este caso de uso se utiliza cuando se desea realizar búsquedas a planificaciones realizadas para modificarlas, exportarlas, imprimirlas o eliminarlas.
<b>Referencia</b>	<b>R 6.1 , R 6.2 , R 6.3 , R</b>

	6.4 , R 6.5 , R 6.6 , R 6.7
--	-----------------------------

<b>CU-7</b>	<b>Autenticar Usuario</b>
<b>Actor</b>	Especialista Jefe de Prueba Gestor Usuario Administrador
<b>Descripción</b>	Este caso de uso permite que los usuarios se autenticuen en el sistema.
<b>Referencia</b>	R 8.1

<b>CU-8</b>	<b>Gestionar Usuarios</b>
<b>Actor</b>	Administrador
<b>Descripción</b>	Este caso de uso le permite al administrador crear los usuarios y modificarlos, asignar los roles, restringir roles, eliminar usuarios, usuarios.
<b>Referencia</b>	R 7.1, R 7.2, R 7.3, R 7.4, R 7.5



**Anexo 2 Descripción de los Casos de uso Gestionar Usuario y Autenticar Usuario.**

<b>Caso de uso:</b> Gestionar Usuario	
<b>Actores:</b> Administrador	
<b>Propósito</b>	
Crear un nuevo usuario, editar datos de usuario, asignar roles a usuarios, restringir roles a usuarios y eliminar usuarios	
<b>Resumen</b>	
<p>El caso de uso se inicia 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>• 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>	
El usuario debe ser administrador.	
<b>Referencias:</b> R 7.1, R 7.2, R 7.3, R 7.4, R 7.5	
<b>Casos de uso asociados:</b>	
<b>Postcondiciones</b>	
Queda creado, editado los datos, asignado el rol ó eliminado un usuario.	
<b>Requerimientos especiales:</b>	
<b>Prototipo:</b>	
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>

<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>• 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 Eliminar usuario ir a la Sección <b>“Eliminar usuario”</b>.</li> </ul>
	<p>3. El sistema finaliza la ejecución del caso de uso</p>
<p><b>Sección 1: Crear un nuevo usuario.</b></p>	
	<p>1. Muestra la interfaz correspondiente para la creación de un nuevo usuario.</p>
	<p>2. Brinda la posibilidad de crear un nuevo usuario con los siguientes datos:</p> <ul style="list-style-type: none"> <li>• Usuario</li> <li>• Contraseña</li> <li>• Asignar rol al usuario</li> </ul>

<p>3. Provee los datos necesarios para crear un nuevo usuario.</p>	<p>4. Crea un nuevo usuario.</p>
	<p>5. Da la posibilidad de crear otro usuario.</p>
<p><b>Sección 2: Editar datos de usuario.</b></p>	
	<p>1. Muestra la interfaz correspondiente para la edición de datos de un usuario.</p>
	<p>2. El sistema brinda la posibilidad de buscar un usuario según:</p> <ul style="list-style-type: none"> <li>• Usuario.</li> <li>• Contraseña.</li> </ul>
<p>3. El administrador introduce uno de los criterios de búsqueda.</p>	<p>4. El sistema busca y muestra los usuarios con los criterios de búsquedas especificados.</p>
	<p>5. El sistema brinda posibilidad de editar los datos del usuario:</p> <ul style="list-style-type: none"> <li>• Usuario</li> <li>• Contraseña</li> </ul>
<p>6. EL usuario edita los datos del usuario:</p> <ul style="list-style-type: none"> <li>• Usuario</li> <li>• Contraseña</li> </ul>	<p>7. El sistema cambia los datos del usuario.</p>

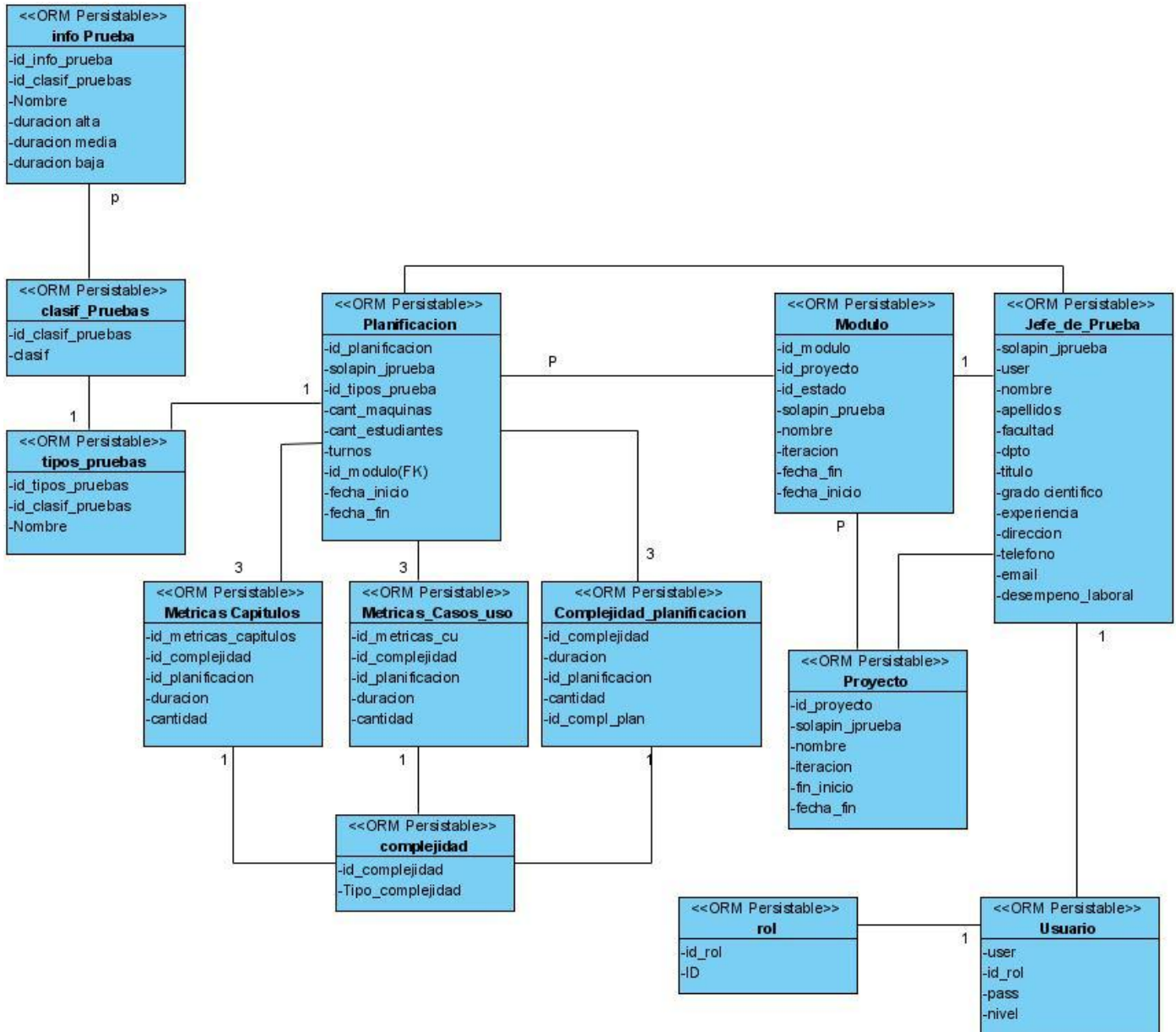
	8. Da la posibilidad de editar los datos de otro usuario.
<b>Sección 3: Asignar roles a usuarios.</b>	
	1. Ejecuta los pasos del 1 al 4 de la <b>sección “Editar datos de usuario”</b> .
	2. El sistema brinda la posibilidad de asignar los siguientes roles a los usuarios: <ul style="list-style-type: none"> <li>• Jefe de Prueba</li> <li>• Especialista</li> <li>• Desarrollador</li> <li>• Probador</li> <li>• Administrador</li> </ul>
3. El Administrador selecciona el rol que va asignar.	4. El sistema asigna el rol especificado.
<b>Sección 4: Eliminar usuarios.</b>	
	1. Ejecuta los pasos del 1 al 4 de la <b>sección “Asignar roles a usuarios”</b> .
2. El administrador selecciona el usuario que desea borrar.	3. Muestra un mensaje de ratificación “Está seguro que desea eliminar el usuario”
4. Acepta el mensaje.	5. El sistema elimina el usuario seleccionado.

<b>Cursos alternos</b>
<p><b>Acción 4 Sección Crear un nuevo usuario.</b></p> <p>1. Si falta algún dato necesario para la creación del nuevo usuario, emite un mensaje de error “Debe de llenar todos los datos del usuario” y no crea el usuario.</p> <p><b>Acción 4 Sección Editar datos de usuario</b></p> <p>1. En caso de no existir el usuario especificado se muestra un mensaje de información “No se encontró el usuario”</p> <p><b>Acción 4 Sección Eliminar usuario</b></p> <p>1. Cancelar Mensaje</p>

<b>Caso de uso:</b> Autenticar Usuario
<b>Actores:</b> Usuario del sistema
<b>Propósito</b>
Permitir autenticarse.
<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>Referencias:</b> R 8.1
<b>Casos de uso asociados:</b>
<b>Postcondiciones</b>
Se habilitan las funcionalidades según los privilegios.
<b>Requerimientos especiales:</b>
<b>Prototipo:</b>

<b>Curso Normal de los Eventos</b>	
<b>Acciones 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.
	6. Termina el caso de uso Autenticarse.
<b>Cursos alternos</b>	
<b>Acción 5 de la sección Respuesta del sistema</b>	
1. En caso de no coincidir la contraseña se le envía un mensaje de: Contraseña incorrecta.	

### Anexo 3 Diagrama de clases persistentes.



**Anexo 4 Descripción de las tablas de la base de datos.**

<b>Nombre: Info_Pruebas</b>		
<b>Descripción:</b> Esta tabla almacena las duraciones por complejidad de los tipos de pruebas.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_info_pruebas	Integer	Este campo es llave primaria y almacena el identificador que distingue la información de la duración de las pruebas
id_clasif_pruebas	Integer	Este campo es una llave primaria y almacena el identificador que distingue la clasificación de la pruebas.
nombre	String	Este campo almacena los nombres de las pruebas.
duración _ alta	Integer	Este campo almacena la duración para complejidad alta de una prueba.
duración _ media	Integer	Este campo almacena la duración para complejidad media de una prueba.
duración _ baja	Integer	Este campo almacena la duración para complejidad baja de una prueba.

<b>Nombre: clasif_pruebas</b>		
<b>Descripción:</b> En esta tabla se almacenan las clasificaciones de pruebas.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_clasif_pruebas	Integer	Este campo es una llave primaria y almacena el identificador que distingue la clasificación de la pruebas.



clasif	String	Este campo almacena los nombres de las clasificaciones de pruebas.
--------	--------	--

<b>Nombre: tipos _ pruebas</b>		
<b>Descripción:</b> En esta tabla se almacenan todos los tipos de prueba.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_tipos_pruebas	Integer	Este campo es llave primaria y almacena el identificador que distingue a los tipos de prueba.
id_clasif_pruebas	Integer	Este campo es llave extranjera y almacena el identificador que distingue la clasificación de la pruebas.
nombre	String	Este campo almacena los nombres de los tipos de pruebas.

<b>Nombre: Proyecto</b>		
<b>Descripción:</b> En esta tabla se almacena los datos del proyecto.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id _ proyecto	String	Este campo es llave primaria y almacena el identificador que distingue a los proyectos.
solapin_jprueba	Integer	Este campo es llave extranjera y almacena el número de solapin del jefe de prueba.
nombre	String	Este campo almacena los nombres de los proyectos.
iteración	Integer	Este campo almacena las iteraciones de pruebas de un proyecto.
fecha _ inicio	String	Este campo almacena la fecha en que el proyecto entra en revisión.
fecha _ fin	String	Este campo almacena la fecha en que el

		proyecto termino de ser revisado.
--	--	-----------------------------------

<b>Nombre: modulo</b>		
<b>Descripción:</b> En esta tabla se almacenan los datos de los módulos de un proyecto.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id _ modulo	Integer	Este campo es llave primaria y almacena el identificador que distingue a los módulos.
id _ proyecto	String	Este campo es llave extranjera y almacena el identificador que distingue a los proyectos.
solapin _ prueba	Integer	Este campo es llave extranjera y almacena el número de solapin del jefe de prueba.
id _ estado	Integer	Este campo es llave extranjera y almacena el identificador que distingue el estado en el que se encuentra un módulo.
nombre	String	Este campo almacena los nombres de los módulos.
iteración	Integer	Este campo almacena las iteraciones de pruebas de un módulo.
fecha _ inicio	String	Este campo almacena la fecha en que el modulo entra en revisión.
fecha _ fin	String	Este campo almacena la fecha en que el modulo termino de ser revisado.

<b>Nombre: planificación</b>		
<b>Descripción:</b> En esta tabla se almacena las planificaciones realizadas.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id _ planificación	Integer	Este campo es una llave primaria y almacena

		el identificador que distingue las planificaciones.
solapin_jprueba	Integer	Este campo es llave extranjera y almacena el número de solapin del jefe de prueba.
id_tipos_pruebas	Integer	Este campo es llave extranjera y almacena el identificador que distingue a los tipos de prueba.
id _ modulo	Integer	Este campo es llave extranjera y almacena el identificador que distingue a los módulos.
cant_maquinas	Integer	Este campo almacena la cantidad de máquinas que intervienen en una planificación.
cant_estudiantes	Integer	Este campo almacena la cantidad de estudiantes que intervienen en una planificación.
fecha _ inicio	String	Este campo almacena la fecha en que debe comenzar el trabajo.
fecha _ fin	String	Este campo almacena la fecha en que debe terminar el trabajo.
turnos	Integer	Este campo almacena los turnos de trabajo en una planificación.

<b>Nombre: métricas _ capítulos</b>		
<b>Descripción:</b> En esta tabla se almacenan las métricas de las pruebas que utilizan capítulos.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_metricas_capitulos	Integer	Este campo es una llave primaria y almacena el identificador que distingue a las métricas de los capítulos.
id _ complejidad	Integer	Este campo es una llave extranjera y almacena el identificador que distingue a las

		complejidades de pruebas.
id _ planificación	Integer	Este campo es una llave extranjera y almacena el identificador que distingue las planificaciones.
duración	Integer	Este campo almacena la duración de las métricas por capítulo.
cantidad	Integer	Este campo me almacena la cantidad de capítulos que utilizaran esa métrica.

<b>Nombre: métricas_casos_uso</b>		
<b>Descripción:</b> En esta tabla se almacenan las métricas de las pruebas que utilizan casos de uso.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_metricas_cu	Integer	Este campo es una llave primaria y almacena el identificador que distingue a las métricas de los casos de uso.
id _ complejidad	Integer	Este campo es una llave extranjera y almacena el identificador que distingue a las complejidades de pruebas.
id _ planificación	Integer	Este campo es una llave extranjera y almacena el identificador que distingue las planificaciones.
duración	Integer	Este campo almacena la duración de las métricas por casos de uso.
cantidad	Integer	Este campo me almacena la cantidad de capítulos que utilizaran esa métrica.

<b>Nombre: complejidad _ planificación</b>		
<b>Descripción:</b> En esta tabla se almacenan las métricas de las pruebas que utilizan casos de uso.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>

id_compl_plan	Integer	Este campo es una llave primaria y almacena el identificador que distingue a las métricas de la complejidad de prueba.
id _ complejidad	Integer	Este campo es una llave extranjera y almacena el identificador que distingue a las complejidades de pruebas.
id _ planificación	Integer	Este campo es una llave extranjera y almacena el identificador que distingue las planificaciones.
duración	Integer	Este campo almacena la duración de las métricas por complejidad de pruebas.

<b>Nombre: Complejidad</b>		
<b>Descripción:</b> En esta tabla se almacena las complejidades.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id _ complejidad	Integer	Este campo es una llave primaria y almacena el identificador que distingue a las complejidades de pruebas.
tipo _ complejidad	String	Este campo almacena los nombres de los tipos de complejidad.

<b>Nombre: Usuario</b>		
<b>Descripción:</b> Se almacena los datos de los usuarios del sistema.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
User	String	Es la llave primaria y almacena el nombre de usuario relacionado con los probadores y jefes de pruebas.
Id _ roles	Integer	Es una llave extranjera y almacena el identificador del rol que desempeña el usuario.

Pass	String	Almacena la contraseña que tiene el usuario.
Nivel	Integer	Almacena

Nombre: roles		
Descripción: Se almacena el rol que desempeñan los usuarios del sistema.		
Atributo	Tipo	Descripción
Id_rol	Integer	Es la llave primaria y almacena el identificador de un rol que puede desempeñar un usuario.
rol	String	Almacena un rol que puede desempeñar un usuario del sistema.