

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS  
Facultad 9



**TÍTULO:**

# Herramienta para la Gestión de la Calidad en el Departamento de Geoinformática.

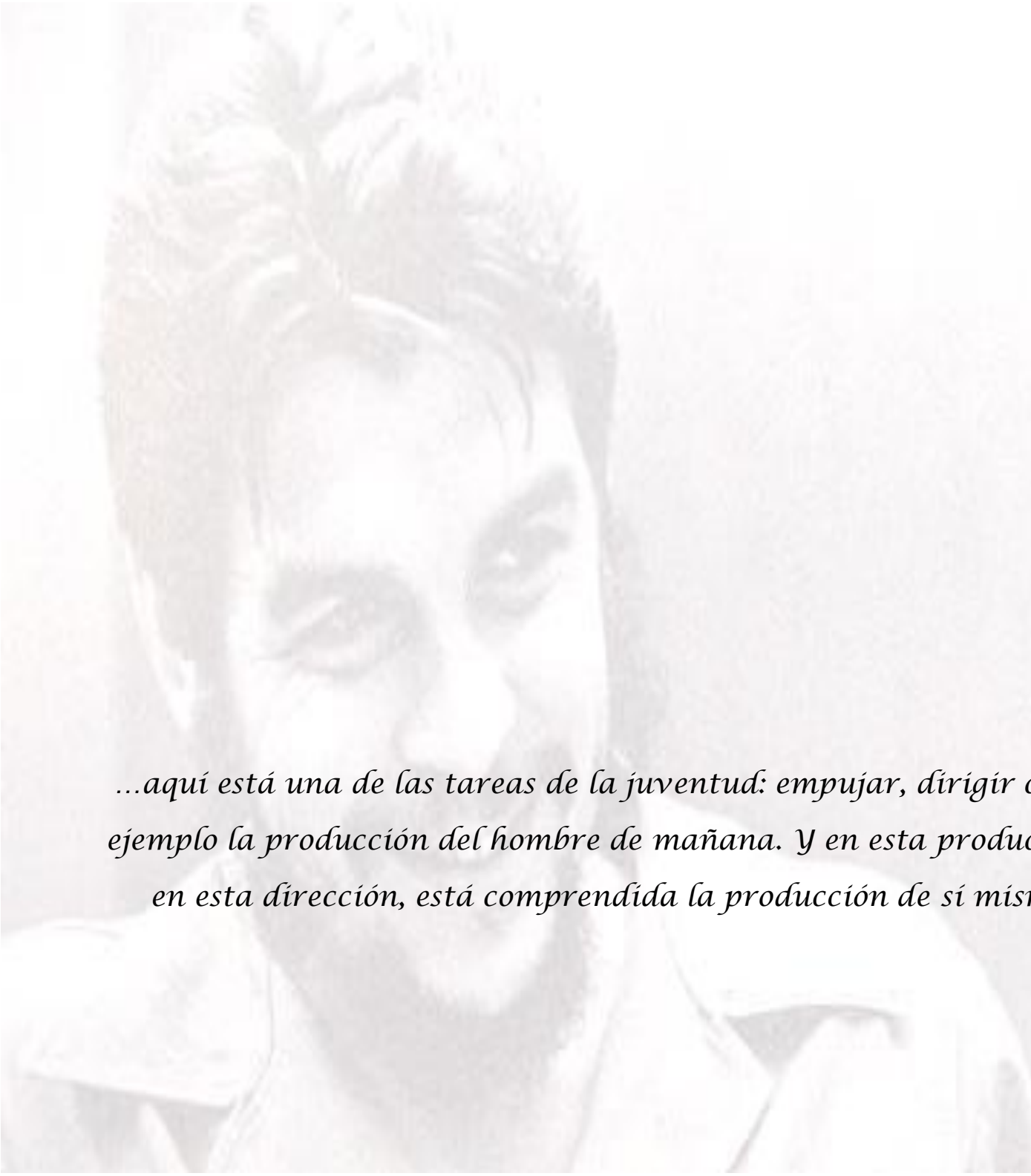
**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS  
INFORMÁTICA**

**AUTORES:** Darién Echenique Carrillo.  
Grétel Marilín Calás Vargas.

**TUTOR:** Ing. Dania Mora Valdés.

**CO-TUTOR:** Ing. Dayris Espinosa Ronquillo.

Ciudad de La Habana, Diciembre 2009  
“Año del 50 Aniversario del triunfo de la Revolución”



*...aquí está una de las tareas de la juventud: empujar, dirigir con el ejemplo la producción del hombre de mañana. Y en esta producción, en esta dirección, está comprendida la producción de sí mismos...*

*Che*

### *De Darién:*

*Dedico este trabajo de diploma a mis padres, a mi hermano, a mis compañeros de estudios y a la Revolución Cubana.*

### *De Grétef:*

*A toda mi familia por el cariño y apoyo que me brindaron, en especial a mis adorables padres Mimi y Pipo, mi razón de ser, por su amor incondicional, por el sacrificio a que se sometieron todos estos años para complacerme en todo y para que no me faltara nada, por su grandioso ejemplo, por el apoyo moral, la confianza depositada en mí, para que mi sueño se hiciera realidad.*

*A mi querido amor Nelson, por ser mi fuente de inspiración, amor, cariño y confianza.*

*A mi querida abuelita Ángela, por apoyarme y quererme en todo momento.*

*A mi precioso sobrino José Alejandro, por llenar mi vida de felicidad.*

## Agradecimientos

*A nuestra tutora por su incansable ayuda en todo momento.*

*A la Revolución Cubana por darnos la posibilidad de formarnos en esta universidad y como personas.*

### **De Darién:**

*Una vez concluidos estos cinco años de estudios y llegado el momento de decir adiós a esta etapa maravillosa que ha contribuido tanto a mi formación como persona, quiero agradecer a las personas que más me han ayudado en esta etapa.*

*En primer lugar agradecer a toda mi familia, a mi madre, a mi padre y a mi hermano por el gran esfuerzo y la paciencia para enfrentar numerosos obstáculos, por apoyarme en cada momento, por darme las fuerzas para seguir adelante, gracias por todo, gracias por no exigir, por no presionar y sobre todo gracias por confiar porque sin su apoyo nada de esto hubiera sido posible.*

*Agradecer a todos mis compañeros, que día a día durante estos cinco compartieron cada instante, cada momento bueno, brindando su apoyo incondicional en los malos.*

*A todos... gracias*

### **De Grétef:**

*A toda mi familia por el apoyo brindado, en especial a Mimi y a Pipo, mis padres queridos, por apoyarme incondicionalmente en todo momento, por estar presentes siempre que los necesité, por darme cariño, amor, los quiero mucho.*

## Agradecimientos

*Al amor de mi vida, Nelson, por llenarme de felicidad, por amarme, respetarme, apoyarme, por su paciencia, te amo con la vida mi amor.*

*A mi abuelita linda Ángela, por estar siempre pendiente de mí, ayudándome en todo lo que podía.*

*A mis hermano Michel y Maikel, por apoyarme y ayudarme en todo momento.*

*A mis tíos Nilva, José, La Negra, Leo, El Negro, Neyda, Irsef, Zory, Beto, Noly, Edel, por su preocupación y ayuda.*

*A mis primos Maidelys, Maikel, Cely, Legna, Luis Carlos, por ayudarme en todo lo que necesité.*

*A mis amigos, Raulf, Zoe, Grissel, Alberto, que han sabido brindarme apoyo, cariño, hospitalidad y quererme como una sobrina más.*

*A mi suegro Wilnelson, por acogerme en su casa como una hija y apoyarme en todo momento, así como a mi suegra Teresa y mi abuela suegra Mercedes por quererme y tenerme siempre presente.*

*A mis amigos Lilitiana, Yanet, Jordan, Maylé, Reinaldo, Raúl, Yannia, Ale, Zuleira, Daílyn, Samara por apoyarme y motivarme en todo momento, y hacer que los días que pasé en la universidad fueran más alegres.*

*A todos...gracias*

## Declaración de Autoría

Declaro que soy el único autor de este trabajo y autorizo al <nombre área> de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

"[Insertar nombre(s) de autor(es)]"    "[Insertar nombre(s) de tutor(es)]"

---

## Opinión del Tutor

<En este acápite se incluye la opinión del tutor del trabajo de diploma>

En la presente investigación se propone el desarrollo de una aplicación que permite elevar la eficiencia en la gestión de la calidad. La propuesta de solución se aplicará en el Departamento de Geoinformática de la facultad 9, perteneciente a la Universidad de las Ciencias Informáticas. Los procesos de revisiones, auditorías y pruebas generan un flujo de información bastante grande, entre revisores, analistas, probadores, desarrolladores y otros roles de un equipo de proyecto. Para minimizar ese flujo de información se propone centralizar los artefactos generados como las listas de chequeo, las no conformidades, las respuestas a las no conformidades, el plan de aseguramiento de la calidad, entre otros en una aplicación que posibilite el tratamiento de esa información de una manera dinámica. El sistema permite: búsqueda de documentos, gestionar el plan de aseguramiento de la calidad, proporcionar información actualizada sobre temas específicos de gestión de calidad y verificar el cumplimiento de las tareas de corrección, entre otras funcionalidades. Propiciará una mayor organización en el trabajo del equipo de proyecto, la información se encontrará al alcance de todos y los diferentes roles que intervienen en los procesos de gestión de la calidad, podrán darle un mejor seguimiento a los estados de la información generada por cada proceso.

## **PALABRAS CLAVES**

Gestión de la calidad, Aplicación, Eficiencia.



In this research proposes the development of an application to increase efficiency in the management of quality. The proposed solution is implemented in the Department of Geoinformatics of the 9 faculty, belonging to the University of Informatics Sciences. The processes of reviews, audits and tests generate a large enough flow of information between reviewers, analysts, testers, developers and other roles of a project team. To minimize the flow of information is proposed to centralize the generated artifacts such as checklists, nonconformities, the responses to nonconformities, the plan for quality assurance, including in an application that makes possible the treatment of such information a dynamic fashion. The system allows:

search for documents, manage the plan of quality assurance, provide updated information on specific topics of quality management and check the fulfillment of the tasks of correction. Further increase in the work organization of the project team, the information will be available to all and the different roles involved in the processes of quality management, can give better tracking of the states of the information generated by each process.

### KEYWORDS

Quality

Management,

Application,

Efficiency.

## Índice de Figuras

Figura 1. DCD Gestionar Documentos de Proyecto. ....	63
Figura 2. DCD Gestionar No Conformidades.....	64
Figura 3. DCD Gestionar Listas de Chequeo.....	65
Figura 4. DCD Gestionar Seguimiento de Tareas.....	65
Figura 5. El patrón MVC .....	68
Figura 6. Diagrama Entidad Relación. ....	70
Figura 7. Diagrama de Clases Persistente .....	70
Figura 8. DC Gestionar Documentos de Proyecto.....	72
Figura 9. DC Gestionar No Conformidades. ....	72
Figura 10. DC Gestionar Listas de Chequeo. ....	73
Figura 11. DC Gestionar Tareas de Corrección.....	73

# Índice de Contenidos

Introducción.....	1
1.1 Conceptos asociados al dominio del problema.....	5
1.1.1 Gestión:.....	5
1.1.2 Sistema de Gestión: .....	5
1.1.3 Calidad: .....	6
1.1.4 Calidad de Software: .....	6
1.1.5 Sistemas de Gestión de la Calidad: .....	7
1.2 Proceso de Gestión de la Calidad del Departamento Geoinformática. ....	7
1.2.1 Descripción General.....	7
1.2.2 Descripción actual del dominio del problema. ....	8
1.2.3 Situación Problemática.....	9
1.3 Análisis de otras soluciones para la Gestión de la Calidad existentes.....	10
1.3.1 KMKey Quality:.....	10
1.3.2 TQC-STI:.....	11
1.3.3 Simandoc: .....	11
1.4 Conclusiones .....	11
2.1 Las Tecnologías de la Información y las Comunicaciones (TIC) .....	12
2.1.1 Metodologías tradicionales:.....	12
2.1.1.1 Rational Unified Process (RUP).....	12
2.1.2 Metodologías ágiles .....	13
2.1.2.1 Agile RUP .....	13
2.1.2.2 Extreme Programming (XP) .....	13
2.1.2.3 Lenguaje Unificado de Modelado (UML).....	13
2.1.3 Herramientas Case .....	14
2.1.3.1 Visual Paradigm.....	14
2.1.3.2 Rational Rose .....	14
2.1.4 Lenguajes de Programación.....	15

# Índice de Contenidos

2.1.4.1 PHP .....	15
2.1.4.2 Java .....	15
2.1.5 Entorno Integrado de Desarrollo (IDE) .....	16
2.1.5.1 Eclipse .....	16
2.1.5.2 Zend Studio for Eclipse.....	16
2.1.5.3 AJAX.....	16
2.1.6 Frameworks.....	17
2.1.6.1 Symfony.....	17
2.1.6.2 Kumbia.....	18
2.1.7 Sistema Gestor de Base de Datos (SGDB).....	19
2.1.7.1 PostgreSQL .....	19
2.1.7.2 MySQL.....	19
2.1.8 Servidor Web.....	20
2.1.8.1 AppServ.....	20
2.2 Justificación de las tecnologías a usar.....	21
2.3 Conclusiones .....	21
Introducción .....	22
3.1 Modelo de Negocio .....	22
3.1.1 Actores y trabajadores del negocio.....	22
3.1.2 Procesos de negocio .....	23
3.1.3 Reglas del Negocio.....	23
3.1.4 Diagrama de Casos de Uso del Negocio .....	24
3.1.5 Descripción textual de los Casos de Uso de Negocio .....	25
3.1.6 Modelo de Objetos.....	29
3.2 Requerimientos del Sistema .....	29
3.2.1 Funcionalidad .....	29
3.2.2 Usabilidad.....	32
3.2.3 Fiabilidad .....	32

# Índice de Contenidos

3.2.4 Eficiencia .....	33
3.2.5 Soporte .....	33
3.2.6 Restricciones de Diseño .....	33
3.2.7 Requisitos para la documentación de usuarios en línea y ayuda del sistema. 34	
3.2.8 Interfaz.....	34
3.2.9 Requisitos legales, de copyright y otros .....	35
3.3 Descripción del Sistema Propuesto.....	35
3.3.1 Descripción de los actores.....	35
3.3.2 Diagrama de Casos de Uso del Sistema .....	36
3.4 Conclusiones.....	61
Introducción .....	62
4.1 Diagramas de Clases .....	62
4.1.1 Subsistema Gestión de Información.....	62
4.1.2 Subsistema Gestión de Calidad.....	63
4.2 Principios de diseño .....	66
4.2.1 Estándares de la interfaz de la aplicación .....	66
4.2.2 Patrón arquitectónico Modelo Vista Controlador (MVC). .....	67
4.2.3 Concepción general de la ayuda .....	68
4.2.4 Seguridad del Sistema.....	68
4.3 Diseño de la Base de Datos.....	69
4.4 Modelo de Despliegue.....	71
4.5 Modelo de Implementación .....	71
4.6 Prueba del sistema propuesto.....	73
4.7 Conclusiones.....	74
Conclusiones.....	75
Referencias Bibliográficas.....	76
Bibliografía .....	80

## Introducción

Desde a mediados del siglo XX, cuando comenzaron a surgir las primeras computadoras, el desarrollo de la informática se ha ido acelerando, posibilitando la creación de ordenadores capaces de realizar operaciones cada vez más complejas y de presentar interfaces cada vez más interactivas. Estos ordenadores son accesibles para todas las organizaciones y empresas, debido a su reducción en tamaño y costo. Unido a esto, la creación de Sistemas Operativos cada vez más especializados y configurables ha hecho posible la simulación de una gran cantidad de actividades de la vida cotidiana de las personas.

Las organizaciones y empresas se volcaron a la utilización de herramientas de gestión, que les facilitarían un mejor control sobre todos los procesos que se desarrollan dentro de su marco de trabajo. Esto trajo consigo que los procesos de desarrollo de software se hayan tornado cada vez más complejos y extensos, generando una gran cantidad de documentación y aplicaciones, que deben ser revisadas y auditadas para garantizar la calidad de los mismos. Es por eso que es necesaria la aplicación de sistemas informáticos, que ayuden a controlar todos los procesos que genera la Calidad del Software y contribuir a la reducción del tiempo de desarrollo de los proyectos.

Cuba se encuentra dando los primeros pasos en la construcción de una industria de software, que le permita insertarse en el mercado mundial y sustituir importaciones de software extranjeros. El uso de aplicaciones que ayuden a la Gestión de la Calidad de Software es de vital importancia en aras de elevar la calidad de productos y garantizar el cumplimiento de los proyectos en el tiempo planificado. Entre los centros en los que se desarrollan y se hace uso de dichas aplicaciones se encuentra la Universidad de la Ciencias Informáticas (UCI), la cual se creó con el fin de aumentar la producción de software en Cuba. Desde sus inicios se pensó en estrategias de desarrollo rápido pero a la vez sólido, haciendo un estudio detallado para el aprovechamiento óptimo de los recursos con que contaba el país.

A esto se le han sumado nuevas tácticas de trabajo que han perfeccionado la calidad de las actividades. Una de las estrategias, propias de la universidad, que ha tenido un resultado apreciable, es la conformación de los Centros Productivos, en los que se llevan a cabo proyectos de desarrollo de software. La mayoría de los proyectos suelen presentar atrasos en sus cronogramas debido a malas planificaciones, lo que dificulta la realización de revisiones y auditorías de manera organizada.

En la Facultad 9 se encuentra el Centro Productivo de Geoinformática y Señales Digitales (GEySED), el cual está conformado por el Departamento de Geoinformática que cuenta con 2 proyectos activos que generan una gran cantidad de documentación y aplicaciones que deben ser revisadas individualmente por parte del Grupo de Calidad del Departamento. Para realizar esta actividad solamente se cuenta con un servidor Subversión por proyecto, donde se almacena toda la documentación y todas las aplicaciones. Para realizar una búsqueda de información, si no se conoce la ubicación exacta del documento que se desea obtener, se desperdicia demasiado tiempo. Otro aspecto fundamental es que las notificaciones y verificaciones a los integrantes de los proyectos, para corregir los defectos encontrados, se realizan vía correo electrónico, personalmente o a través del jefe de proyecto, incidiendo en un mayor tiempo de corrección de los defectos detectados.

Otro aspecto de gran incidencia es que no existe un espacio para publicar la información relacionada con la gestión de la calidad que permita a los grupos de desarrollo mantenerse actualizados y regir el desarrollo de los productos por las normas y estándares establecidos, que ayudarían a evitar errores en la codificación y en la elaboración de las plantillas del expediente de proyecto. En el mundo existen aplicaciones que realizan estas funciones pero a pesar de estar desarrolladas sobre software libre, requieren de gastos en divisas para su aplicación y las circunstancias actuales del país no lo permiten.

La situación problemática planteada anteriormente, permite identificar como **problema científico**: ¿Cómo agilizar los procesos que se ejecutan dentro de la Gestión de la Calidad del Departamento de Geoinformática?

Estableciendo como **objeto de estudio**: Procesos de Gestión de la Calidad en el desarrollo de software y **campo de acción**: Automatización del proceso de Gestión de la Calidad en el Departamento de Geoinformática.

El **objetivo general** que persigue este trabajo es: Desarrollar un Sistema informático para la Gestión de la Calidad en el Departamento de Geoinformática.

La **idea a defender** es: Con el desarrollo de un sistema para la Gestión de la Calidad en el Departamento de Geoinformática se eleva la organización, el seguimiento y control de los procesos que ocurren en esta área.

Para cumplir este propósito se deben realizar las siguientes tareas de investigación:

- Caracterizar las herramientas para la Gestión de la Calidad existentes en el mundo, las metodologías, tendencias y tecnologías que se utilicen para la creación de herramientas.
- Modelar el Negocio.
- Realizar el levantamiento de requisitos del sistema a desarrollar.
- Analizar el sistema a desarrollar.
- Diseñar el sistema a desarrollar.
- Implementar el Sistema.
- Diseñar las pruebas de caja blanca y caja negra al sistema.
- Implementar las pruebas caja negra al sistema.

Los métodos científicos de la investigación son:

Teóricos:

- Analítico-Sintético: Para realizar un análisis con la bibliografía y la documentación relacionada con los estándares de calidad de software y las herramientas de gestión con el objetivo de mejorar el conocimiento del objeto de estudio.
- Análisis Histórico-Lógico: Para realizar un análisis histórico-lógico de las aplicaciones existentes en el mundo, para poder realizar comparaciones entre ellas y posibilitar el entendimiento del sistema.
- Modelación: Para lograr un mayor entendimiento del funcionamiento de los procesos de la Gestión de la Calidad en los proyectos del Departamento de Geoinformática.

Empíricos:

- Observación: Para observar cómo se realiza el proceso de gestión de la calidad con el objetivo de lograr un mejor entendimiento de este proceso.

Posibles resultados:



- Herramienta para la Gestión de la Calidad.
- La documentación técnica asociada al desarrollo de la herramienta.

Este trabajo de diploma consta de 4 capítulos en total. El primer capítulo, aborda el tema de la Fundamentación Teórica de la investigación realizada, mostrando los principales conceptos asociados a la gestión de la calidad de software así como un estudio sobre las aplicaciones existentes en el mundo.

En el segundo capítulo, Tendencias y Tecnologías actuales, se realiza un estudio de las principales tecnologías y metodologías factibles para el desarrollo del sistema, definiéndose cuáles serán usadas en el proceso de desarrollo.

El tercer capítulo, Presentación de la solución propuesta contiene el modelamiento de los procesos del negocio, la representación de los casos de uso del sistema y los requerimientos funcionales y no funcionales del sistema.

El cuarto capítulo, Construcción de la solución propuesta muestra los modelos de Análisis, Diseño, Implementación, así como los estándares de codificación y las pruebas del sistema resultante.

## Introducción

Debido a los grandes avances de la informática, las aplicaciones de gestión han ocupado un lugar muy importante dentro de las actividades de cualquier entidad, favoreciendo el control y la eficiencia de la misma. En el caso específico de las aplicaciones de gestión de la calidad de software son de gran importancia puesto que permiten una mejor organización del desarrollo de software así como un mejor aprovechamiento de los recursos. El presente capítulo se acerca a los principales conceptos y definiciones asociados con los sistemas de gestión y la calidad de software además de ampliar el estudio de la situación problemática y el objeto de estudio.

### 1.1 Conceptos asociados al dominio del problema.

#### 1.1.1 Gestión:

Es coordinar todos los recursos disponibles para conseguir determinados objetivos, implica amplias y fuertes interacciones fundamentalmente entre el entorno, las estructuras, el proceso y los productos que se deseen obtener. Del latín gestio, el concepto de gestión hace referencia a la acción y al efecto de gestionar o de administrar. Gestionar es realizar diligencias conducentes al logro de un negocio o de un deseo cualquiera. Administrar, por otra parte, consiste en gobernar, dirigir, ordenar, disponer u organizar. **(Definición, 2009)**

#### 1.1.2 Sistema de Gestión:

Un sistema de gestión ayuda a lograr los objetivos de la organización mediante una serie de estrategias, que incluyen la optimización de procesos, el enfoque centrado en la gestión y el pensamiento disciplinado.

La implementación de un sistema de gestión eficaz puede ayudar a:

- Gestionar los riesgos sociales, medioambientales y financieros.
- Mejorar la efectividad operativa.
- Reducir costos.

**(BSI, 2009)**

# Capítulo 1: Fundamentación Teórica

## 1.1.3 Calidad:

La Calidad es una herramienta básica para una propiedad inherente de cualquier cosa que permite que esta sea comparada con cualquier otra de su misma especie. La palabra calidad tiene múltiples significados. Es un conjunto de propiedades inherentes a un objeto que le confieren capacidad para satisfacer necesidades implícitas o explícitas.

En la práctica, hay dos tipos de calidad:

- **Calidad externa**, los beneficiarios de la calidad externa son los clientes y los socios externos de una compañía.
- **Calidad interna**, los beneficiarios de la calidad interna son la administración y los empleados de la compañía. **(Kioskea.net, 2008)**

## 1.1.4 Calidad de Software:

“Grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario”. **(IEEE, Std. 610-1990 - Standard Glossary of Software Engineering Terminology)**

Es la concordancia con los requerimientos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se esperan de todo software desarrollado profesionalmente.

## Control de Calidad

El costo de corregir y detectar errores producidos en las primeras fases de desarrollo de software es mayor a medida que se encuentran más alejados de éstas. A causa de esto, la propuesta de control de calidad es empujar las tareas relacionadas con la calidad desde las primeras fases del proyecto. Esto permite encontrar los errores en forma temprana sin que se sigan propagando en las siguientes fases. La garantía de la calidad de software engloba métodos y herramientas de análisis, diseño, codificación y prueba; revisiones y técnicas formales que se aplican en cada fase de la ingeniería de software, así como una estrategia de prueba multiescalada. **(Universidad de Belgrano, 2006)**

# Capítulo 1: Fundamentación Teórica

## 1.1.5 Sistemas de Gestión de la Calidad:

"Estructura de la organización, responsabilidades, procedimientos, procesos y recursos que se establecen para llevar a término la gestión de calidad".

**Las partes que componen el sistema de gestión son:**

- Estructura organizativa: departamento de calidad o responsable de la dirección de la empresa.
- Cómo se planifica la calidad.
- Los procesos de la organización.

Que una empresa tenga implantado un sistema de gestión de la calidad, sólo quiere decir que esa empresa gestiona la calidad de sus productos y servicios de una forma ordenada, planificada y controlada.

**(Buscarportal, 2009)**

## 1.2 Proceso de Gestión de la Calidad del Departamento Geoinformática.

### 1.2.1 Descripción General.

El proceso de gestión de la calidad de software en el Departamento de Geoinformática de la Facultad 9 se encuentra regido por los Lineamientos de Calidad de Software (IPL-3100:2008), para la producción de software en la Universidad de las Ciencias Informáticas (UCI).

El primer paso para asegurar la calidad de software es la elaboración del Plan de Aseguramiento de la Calidad por parte del Jefe de Calidad. En él se establecen en qué fechas y durante qué fases de desarrollo se aplicarán las revisiones y auditorías, siendo de estricto cumplimiento por parte de todos los integrantes de los proyectos.

**Se realizarán auditorías con el objetivo de:** Acumular y evaluar la evidencia, determinar e informar sobre el grado de correspondencia existente entre la información cuantificable y los criterios de evaluación establecidos.

**Se realizarán revisiones con el objetivo de:** Descubrir defectos e inconsistencias en el sistema, indicar la necesidad de mejoras en un producto, las partes que no es necesario mejorar y evaluar conformidad con estándares y especificaciones técnicas.

# Capítulo 1: Fundamentación Teórica

Para realizar las revisiones y auditorías se confeccionan las Listas de Chequeos, que son los documentos donde se establecen las condiciones con las que deben cumplir los artefactos generados por el proceso de desarrollo de software y constituyen una guía para la detección de no conformidades.

Después de realizarse una auditoría o revisión se elabora un documento de No Conformidades, en el cual se recogen todas las no conformidades detectadas, detallando cada una de ellas y señalando la fecha de cumplimiento para su corrección por parte del equipo de desarrollo. Las no conformidades pueden ser de tres tipos: alta, media y baja, dependiendo de su incidencia sobre artefacto que está siendo revisado o auditado.

**En el caso de las auditorías se obtendrá una evaluación que puede ser:**

- **Deficiente:** En el caso de que se detecten demasiadas no conformidades de alta incidencia en la calidad de los procesos y el producto.
- **Aceptable:** Se detectaron un conjunto de no conformidades que tienen una incidencia media o baja en la calidad de los procesos y el producto.
- **Satisfactorio:** Se detectan ninguna o pocas no conformidades de baja incidencia en la calidad de los procesos y el producto.

Además de la gestión interna de la calidad de software por parte del Grupo de Calidad del Departamento, los proyectos productivos pueden ser objeto de revisiones o auditorías llevadas a cabo por los miembros de los equipos de Calidad de la facultad y de Calisoft.

## **1.2.2 Descripción actual del dominio del problema.**

El Departamento de Geoinformática cuenta actualmente con 2 proyectos productivos activos que se encuentran en fase de desarrollo y cuentan con una fuerza productiva heterogénea de profesores y estudiantes de distintos años académicos desde 2do a 5to, que trae como consecuencia que no exista un dominio uniforme de los lineamientos de calidad que rigen el desarrollo de software en la Universidad de las Ciencias Informáticas (UCI). Además el Grupo de Calidad no cuenta con un personal fijo que garantice un seguimiento constante del desarrollo de los proyectos.

# Capítulo 1: Fundamentación Teórica

## 1.2.3 Situación Problemática.

Después de realizada una investigación sobre los procesos de la gestión de la Calidad en el Departamento de Geoinformática de la Facultad 9, se detectaron un número de problemas que afectan los procesos y que influyen directamente sobre el tiempo de duración de la corrección de los defectos encontrados y la organización del trabajo.

### Las principales dificultades encontradas fueron:

- **Gran flujo de información, que hace engorroso el trabajo:** Cada proyecto genera constantemente gran cantidad de documentación y artefactos que deben ser revisados individualmente.
- **El repositorio es el lugar donde se almacena toda la documentación de cada proyecto:** Para almacenar toda la información generada por cada proyecto se utiliza un repositorio Subversión para cada proyecto. Debido a la estructura de este sistema, para acceder a la información que se necesita es necesario conocer su ubicación exacta dentro del repositorio. Esto dificulta y retrasa el trabajo al tratar de localizar determinado documento.
- **Necesidad de publicar información referente a la gestión de la calidad:** No existe un espacio para publicar información que posibilite mantener al personal del proyecto actualizado y registrar el desarrollo del producto por las normas y estándares establecidos para la codificación y la elaboración de las plantillas del expediente de proyecto.
- **No existe un mecanismo de actualización del estado de las tareas:** El Jefe de Proyecto siempre es el intermediario entre el equipo de desarrollo y el equipo de calidad, retrasando de esta manera la solución en tiempo de los problemas encontrados.
- **No existe un mecanismo de notificaciones a los integrantes de los proyectos:** Los integrantes de los proyectos solamente son informados de las tareas a cumplir vía correo electrónico o personalmente por el Jefe de Proyecto dando lugar a que existan retrasos en el tiempo de realización de las tareas encomendadas.
- **Existencia de aplicaciones foráneas:** Internacionalmente existen herramientas de gestión que permiten solucionar todas estas dificultades, pero a pesar de ser aplicaciones realizadas en código

# Capítulo 1: Fundamentación Teórica

abierto, estas requieren de inversiones para su aplicación, y la universidad no se encuentra en condiciones de afrontarlas.

## 1.3 Análisis de otras soluciones para la Gestión de la Calidad existentes.

### 1.3.1 KMKey Quality:

Es un software de gestión de calidad ideal para la implantación y mantenimiento de un sistema de gestión de calidad de cualquier tipo: ISO 9000 (ISO 9001:2000, ISO 14001, OHSAS 18001). Entre sus características se destacan:

- Entorno web: Aplicación 100% programada para ser utilizada vía Internet. Se trabaja siempre mediante un navegador (Internet Explorer o Firefox) desde cualquier lugar con acceso a Internet.
- Información accesible mediante permisos: Acceso a los expedientes donde el usuario ha sido asignado y con la posibilidad de realizar las acciones correspondientes a su perfil.
- Multi-entorno: El usuario puede escoger si trabaja con herramientas Microsoft (MS Office, Internet Explorer) u otras (Open Office, Firefox). La gestión documental acepta ficheros de todo tipo (.doc, .xls, .odt, .pdf).
- Software libre: Todos los sistemas operativos, bases de datos y programas necesarios para la instalación de KMKey son Código Abierto (licencia GPL).

Las principales funciones que realiza son:

#### **Realiza la Gestión documental:**

Basado en un entorno web que le proporcionará acceso universal desde cualquier ordenador, KMKey Quality incorpora un Gestor Documental donde se almacenan las distintas revisiones de los documentos propios de un sistema de calidad, tales como el Manual de Calidad, la Política y los Objetivos de Calidad, el Organigrama, el Mapa de Procesos, los Procedimientos, los Formatos o las Instrucciones de Trabajo, permitiendo siempre un acceso directo a la revisión vigente.

#### **No conformidades y acciones correctivas/preventivas:**

# Capítulo 1: Fundamentación Teórica

Desde el mismo gestor de expedientes cualquier usuario autorizado puede introducir No Conformidades, así como planificar las correspondientes Acciones Correctivas o Preventivas, repartiendo las tareas a cada uno de los responsables, controlando plazos y acciones realizadas.

## **Otros procedimientos: auditorías, proveedores, indicadores:**

De igual modo se pueden gestionar expedientes específicos para Planes de Formación, Planes de Auditorías o Planes de Mejora Continua, así como los Dossiers de Revisión del Sistema, el Seguimiento y Evaluación de Proveedores o la Gestión y Control de Indicadores. **(KMKey, 2008)**

### **1.3.2 TQC-STI:**

Es una aplicación de software para la administración y control electrónico de tareas dentro de un sistema de calidad, garantizando que las mismas sean realizadas correctamente y en los tiempos pautados. TQC-STI se convierte en la herramienta de software que facilita la administración eficiente de todas las anomalías y no conformidades relacionadas a su gestión de calidad. **(World Workers Argentina, 2009)**

### **1.3.3 Simandoc:**

Es un nuevo producto para la gestión de la calidad, medio ambiente, salud ocupacional, seguridad y manejo de documentos, desarrollado íntegramente en la República Argentina y basado en tecnologías web.

Se trata de una aplicación web cliente/servidor de bajo costo y gran flexibilidad, con mínimos requerimientos de instalación en el servidor y en los puestos clientes. Conformando totalmente los requisitos de las normativas ISO 9001 y 14001, OHSAS 18001 e IRAM, con respecto a sistemas de gestión de calidad, medio ambiente, seguridad y salud ocupacional, constituyendo una base de software ideal para implementar una política y un procedimiento consistente para la gestión de procesos y el control de la documentación[3]. **(Simandoc, 2008)**

## **1.4 Conclusiones**

Durante el desarrollo de este capítulo se ha podido ampliar el conocimiento sobre aquellos sistemas existentes para la gestión de la calidad, haciendo comparaciones entre ellas, que sirve de guía para el desarrollo del sistema. Se logró un mayor entendimiento de los procesos de gestión de la calidad dentro del Departamento de Geoinformática así como de las principales problemáticas que tiene este proceso.



# Capítulo 2: Tendencias y tecnologías actuales a desarrollar

## Introducción

El gran desarrollo alcanzado por las Tecnologías de la Información y las Comunicaciones (TIC) ha traído consigo grandes avances en cuanto a qué tecnología usar para realizar un determinado proyecto, contando además con varios tipos de fabricantes que cada día presentan nuevas versiones o productos. En el presente capítulo se hace referencia a una parte de las tecnologías y metodologías que brindan aceptables opciones para desarrollar el sistema y definir cuáles utilizar al final del mismo.

## 2.1 Las Tecnologías de la Información y las Comunicaciones (TIC)

### 2.1.1 Metodologías tradicionales:

#### 2.1.1.1 Rational Unified Process (RUP)

El Proceso Unificado Racional (*Rational Unified Process* en inglés, habitualmente resumido como RUP) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

#### Principales características:

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo)
- Pretende implementar las mejores prácticas en Ingeniería de Software
- Desarrollo iterativo
- Administración de requisitos

El RUP es un producto de Rational (IBM). Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso) (**Grupo Soluciones Innova, 2007**)

## Capítulo 2: Tendencias y tecnologías actuales a desarrollar

### 2.1.2 Metodologías ágiles

#### 2.1.2.1 Agile RUP

Es una nueva metodología, nacida de una aplicación de la clásica RUP siendo una buena opción para los proyectos pequeños, facilitando que las personas adaptadas a trabajar en RUP no tengan que cambiar la metodología para desarrollar pequeños proyectos. Reduce considerablemente la documentación generada y los flujos de trabajo con respecto a la tradicional RUP. **(Hirsch, 2008)**

#### 2.1.2.2 Extreme Programming (XP)

La programación extrema o *eXtreme Programming* (XP) es un enfoque de la ingeniería de software formulado por Kent Beck, autor del primer libro sobre la materia, *Extreme Programming Explained: Embrace Change* (1999). Es el más destacado de los procesos ágiles de desarrollo de software.

**Las características fundamentales del método son:**

- Desarrollo iterativo e incremental.
- Corrección de todos los errores antes de añadir nueva funcionalidad.
- Propiedad del código compartida.

**(Sourceforge, 2004)**

#### 2.1.2.3 Lenguaje Unificado de Modelado (UML)

UML, por sus siglas en inglés, *Unified Modeling Language*; es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Solamente es factible para proyectos que utilicen el paradigma de la Programación Orientada a Objetos (POO). **(Iycos, 2009)**

## Capítulo 2: Tendencias y tecnologías actuales a desarrollar

### 2.1.3 Herramientas Case

#### 2.1.3.1 Visual Paradigm

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML.

También proporciona características tales como generación del código, ingeniería inversa y generación de informes. Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de clases. Permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML al instante, creando de manera simple toda la documentación. Cabe destacar igualmente su robustez, usabilidad y portabilidad. **(Visual Paradigm, 2009)**

#### 2.1.3.2 Rational Rose

Herramienta de desarrollo basada en modelos que se integra con las bases de datos y los IDE de las principales plataformas del sector. Rational Rose Enterprise es un entorno de modelado que permite generar código a partir de modelos Ada, ANSI C++, C++, CORBA, Java/J2EE, Visual C++ y Visual Basic. Al igual que todos los productos de Rational Rose, ofrece un lenguaje de modelado común que agiliza la creación del software. Brinda soporte para el Lenguaje Unificado de Modelado (UML).

Dentro de sus principales características se destacan, los componentes del modelo se pueden controlar independientemente, lo que permite una gestión y un uso de modelos más granular. Complemento de modelado Web que incluye funciones de visualización, modelado y herramientas para desarrollar aplicaciones Web. Modelado en UML para diseñar bases de datos, que integra los requisitos de datos y aplicaciones mediante diseños lógicos y analíticos. **(IBM,2009)**

## Capítulo 2: Tendencias y tecnologías actuales a desarrollar

### 2.1.4 Lenguajes de Programación

#### 2.1.4.1 PHP

PHP es el acrónimo de Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación.

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

Entre sus características principales se destacan la de ser un lenguaje multiplataforma, completamente orientado a la web, presenta una capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL, además de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones). Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda. Es open source, por lo que se presenta como una alternativa de fácil acceso para todos. Permite aplicar técnicas de programación orientada a objetos, así como manejo de excepciones (desde PHP5). Posee una biblioteca nativa de funciones sumamente amplia e incluida. No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución. **(Desarrollo Web.com, 2005)**

#### 2.1.4.2 Java

Java es un lenguaje de programación con el que se puede realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Está desarrollado por la compañía Sun Microsystems con gran dedicación y siempre enfocado a cubrir las necesidades tecnológicas más punteras. Una de las principales características por las que Java se ha hecho muy famoso es que es un lenguaje independiente

## Capítulo 2: Tendencias y tecnologías actuales a desarrollar

de la plataforma. Además de ser multiplataforma, enfocado hacia la Programación Orientada a Objetos y presenta un alto nivel de seguridad en su código. **(Desarrollo Web.com, 2008)**

### 2.1.5 Entorno Integrado de Desarrollo (IDE)

#### 2.1.5.1 Eclipse

Es una plataforma universal para integrar herramientas de desarrollo, con una arquitectura abierta y basada en pluggins. Eclipse da soporte a todo tipo de proyectos que abarcan desde el ciclo de vida del desarrollo de aplicaciones, incluyendo soporte para modelado. El entorno integrado de desarrollo (IDE) de Eclipse emplea módulos (en inglés plug-in) para proporcionar toda su funcionalidad al frente de la plataforma de cliente, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Presenta características principales como:

- Editor visual con sintaxis coloreada.
- Compilación incremental de código.
- Depura código que resida en una máquina remota.

#### 2.1.5.2 Zend Studio for Eclipse

Zend Studio o Zend Development Environment es un completo entorno de desarrollo integrado para el lenguaje de programación PHP. Está escrito en Java, y está disponible para las plataformas Microsoft Windows, Mac OS X y GNU/Linux.

No requiere la instalación previa de PHP ni del entorno de ejecución de Java, tiene PhpDoc y Cliente FTP integrado, plegado de código (comentarios, bloques de phpDoc, cuerpo de funciones y métodos e implementación de clases), soporte para PHP 4, PHP 5, y para control de versiones usando CVS o Subversion (a elección del desarrollador). Detecta errores de sintaxis en tiempo real y tiene una Instalación de barras de herramientas para Internet Explorer y Mozilla Firefox (opcional). **(Desarrollo Wev.com, 2009)**

#### 2.1.5.3 AJAX

## Capítulo 2: Tendencias y tecnologías actuales a desarrollar

AJAX, acrónimo de *Asynchronous JavaScript And XML* (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

**Librería JQuery:** jQuery es la librería JavaScript que ha irrumpido con más fuerza como alternativa a Prototype. Su autor original es John Resig, aunque como sucede con todas las librerías exitosas, actualmente recibe contribuciones de decenas de programadores. JQuery también ha sido programada de forma muy eficiente y su versión comprimida apenas ocupa 20 KB.

JQuery comparte con Prototype muchas ideas e incluso dispone de funciones con el mismo nombre. Sin embargo, su diseño interno tiene algunas diferencias drásticas respecto a Prototype, sobre todo el "encadenamiento" de llamadas a métodos.

Las características que más destacan a jQuery de otras librerías o frameworks son el soporte de CSS1, 2, 3 y XPath básico, además de que sólo pesa 10kb (Lo que le otorga velocidad) y es 100% compatible con Internet Explorer, Firefox, Safari, y Opera. La librería es tan sólida y difundida que la utilizan Mozilla, NBC, Wordpress, Amazon, Intel, y AOL, entre muchas otras grandes empresas. **(Libros web, 2009)**

### 2.1.6 Frameworks

#### 2.1.6.1 Symfony

Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web.

## Capítulo 2: Tendencias y tecnologías actuales a desarrollar

Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas \*nix (Unix, Linux, etc.) como en plataformas Windows. A continuación se muestran algunas de sus características.

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y \*nix estándares).
- Independiente del sistema gestor de bases de datos. Su capa de abstracción y el uso de Propel, permiten cambiar con facilidad de SGBD en cualquier fase del proyecto.
- Utiliza programación orientada a objetos, de ahí que sea imprescindible PHP 5.
- Sencillo de usar en la mayoría de casos, está más indicado para grandes aplicaciones Web que para pequeños proyectos.
- Aunque utiliza MVC (Modelo vista controlador), tiene su propia forma de trabajo en este punto, con variantes del MVC clásico como la capa de abstracción de base de datos, el controlador frontal y las acciones.
- Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador sólo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con las bibliotecas de otros fabricantes.

**(Fabien Potencier, 2008)**

### 2.1.6.2 Kumbia

KumbiaPHP es un framework para aplicaciones web libre escrito en PHP5. Basado en las prácticas de desarrollo web como DRY y el Principio KISS para software comercial y educativo. Kumbia fomenta la

## Capítulo 2: Tendencias y tecnologías actuales a desarrollar

velocidad y eficiencia en la creación y mantenimiento de aplicaciones web, reemplazando tareas de codificación repetitivas por poder, control y placer. Presenta características tales como:

- Sistema de plantillas sencillo.
- Mapeo Objeto Relacional (ORM) y Separación MVC.
- Soporte para AJAX.
- Generación de Formularios.

**(kumbiaphp, 2009)**

### 2.1.7 Sistema Gestor de Base de Datos (SGDB)

#### 2.1.7.1 PostgreSQL

El gestor PostgreSQL 8.2 es un gestor potente para la utilización del PostgreSQL, presenta una interfaz amigable, sobre todo elaborado en software libre y multiplataforma, con esta se obtiene 20% más de performance (*Rendimiento*), incluye ordenamientos más rápidos en disco y en memoria, los usuarios pueden ganar más eficiencia en sistemas de análisis de datos y sobre todo resulta menos costoso en relación con grandes gestores de base de datos.

Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP), cadenas de bits, etc. También permite la creación de tipos propios, la declaración de funciones propias, así como la definición de disparadores, la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos. Incorpora una estructura de datos array y funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc. Soporta el uso de índices, reglas y vistas e Implementación del estándar SQL92/SQL99. **(netpecos, 2007)**

#### 2.1.7.2 MySQL

El servidor de bases de datos MySQL es la base de datos de fuente abierta más popular en el mundo. Su arquitectura lo hace extremadamente rápido y fácil de adaptar. Este servidor de bases de datos potente es



## Capítulo 2: Tendencias y tecnologías actuales a desarrollar

usado por muchos programas de origen abierto incluyendo phpBB, osCommerce, y phpNuke. Posee un para la interacción con la programación en PHP[5]. Entre sus características se destacan:

- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc.).
- Gran portabilidad entre sistemas.
- Gestión de usuarios y contraseñas, manteniendo un muy buen nivel de seguridad en los datos.

(netpecos, 2007)

### 2.1.8 Servidor Web

#### 2.1.8.1 AppServ

AppServ es un programa que une una serie de aplicaciones y utilidades, de manera que al instalarlo, no sólo se tiene instalado todo esto, sino que, estarán perfectamente configurados y listos para usarlos en cuestión de minutos, mientras que instalando por separado estos programas, hay que dedicarle mucho más tiempo a su correcta configuración.

#### Aplicaciones que instala:

- Apache
- PHP
- MySQL
- PhpMyAdmin

**Apache** es un servidor HTTP de código abierto disponible para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras. Esta aplicación es necesaria para montar un host local en el cual se apoyarán los demás programas que se instalan.

**PhpMyAdmin** es una utilidad adicional, en absoluto es necesaria para el desarrollo de un sitio web dinámico que se sustente sobre una base de datos, pero facilita mucho la tarea, ya que permite administrar una base de datos sin necesidad de tener que recurrir a la escritura de líneas de comandos

## Capítulo 2: Tendencias y tecnologías actuales a desarrollar

sobre la consola del equipo, proporcionando un entorno gráfico para ello, que además es bastante intuitivo. (dipler.org, 2009)

### 2.2 Justificación de las tecnologías a usar

Después de un profundo análisis de todas las tecnologías y metodologías actuales se utiliza como framework de desarrollo a Symfony, puesto que brinda una gran fortaleza para realizar mapeos objeto-relacional de las bases de datos. Tiene integración con AJAX, así como una amplia documentación existente y es de dominio por parte del equipo de desarrollo. Como lenguaje de programación a PHP, porque es el lenguaje en que está desarrollado Symfony y que brinda grandes opciones para la programación web además de ser multiplataforma, fácil de usar y que cuenta con una amplia documentación. Como sistema gestor de base de datos se selecciona a PostgreSQL, por ser un sistema multiplataforma, de código abierto y de gran integración con PHP. Además de utilizar la tecnología AJAX para lograr una interacción más amigable con la aplicación y ganar en rapidez de la misma. Como servidor web se selecciona el AppServ puesto que es un sistema multiplataforma, de código abierto, altamente confiable y seguro, y que no requiere de grandes requerimientos para su utilización. Como metodología de desarrollo se escoge Agile RUP puesto que es una versión eficiente para proyectos pequeños y que el personal de desarrollo ya está familiarizado con la metodología RUP, por lo que no hay atraso al tener que estudiar nuevas metodologías.

### 2.3 Conclusiones

Durante el desarrollo de este capítulo se ha podido conocer la existencia de tecnologías y metodologías que permiten desarrollar un sistema de gestión eficaz. Al final se realiza la selección idónea, teniendo en cuenta las características del sistema, las políticas de desarrollo de la Universidad y el conocimiento del equipo de desarrollo.

## Capítulo 3: Presentación de la solución propuesta

### Introducción

En el siguiente capítulo se abordan todos los aspectos relacionados con el modelado de los procesos del negocio, que se desarrollan dentro de la Gestión de la Calidad, permitiendo un mejor entendimiento de estos. En base a ellos se crean los Requerimientos del Sistema que permiten la estructuración del sistema, creando las bases para su posterior diseño e implementación.

### 3.1 Modelo de Negocio

Se realiza el Modelo de Negocio para comprender la estructura y la dinámica de la organización, sus problemas actuales y en base a ellos determinar las mejoras potenciales que se podrán desarrollar. Además garantiza un entendimiento común entre los usuarios finales, consumidores y desarrolladores. De él se derivaran los Requerimientos del Sistema que se desea desarrollar.

#### 3.1.1 Actores y trabajadores del negocio

##### Actores del Negocio

Actor	Descripción
Integrante de Proyecto	Son todos los integrantes de los proyectos productivos.
Grupo de Calidad	Son todos los integrantes del los equipos de calidad de cada proyecto.

##### Trabajadores del Negocio

Trabajador	Descripción
Miembro de Grupo de Calidad	Es el encargado de realizar las revisiones, auditorías y de controlar las tareas de corrección. Sus principales actividades son: revisar los artefactos generados por los integrantes de los proyectos, elaborar las no conformidades, elaborar las listas de chequeo, actualizar la Información Técnica y enviar las notificaciones de tareas de corrección a los integrantes de los proyectos y darles seguimiento a las mismas.

## Capítulo 3: Presentación de la solución propuesta

Jefe de Calidad	Es el encargado de elaborar el Plan de Aseguramiento de la Calidad y gestionar los Permisos de acceso.
-----------------	--

### 3.1.2 Procesos de negocio

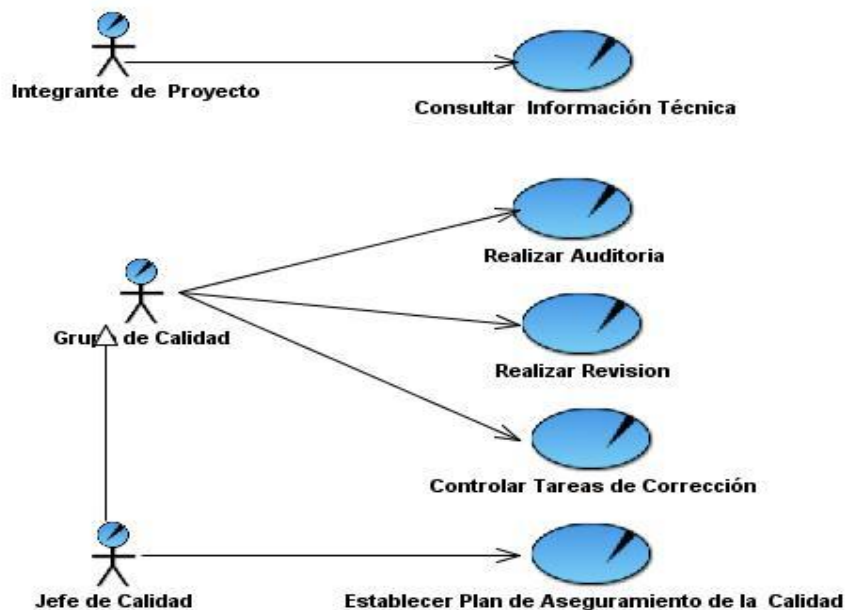
- **Realizar Revisión:** Este proceso consiste en la revisión de los artefactos generados por los integrantes de los proyectos productivos, en las fases definidas. Los encargados de realizar las revisiones, son los Miembros del Grupo de Calidad, ellos elaboran las no conformidades de acuerdo a los errores detectados en la revisión de los artefactos.
- **Realizar Auditoria:** Este proceso consiste en realizar la auditoría de los artefactos generados por los integrantes de los proyectos productivos. Los miembros del Grupo de Calidad de cada proyecto, elaboran las no conformidades de acuerdo a los errores detectados en la auditoría de los artefactos, definiendo una evaluación.
- **Controlar tareas asignadas:** Este proceso consiste en dar seguimiento a las tareas asignadas a los integrantes de los proyectos para corregir las no conformidades detectadas en las revisiones o auditorías realizadas en las diferentes fases de los proyectos.
- **Establecer el Plan de Aseguramiento de la Calidad:** Este proceso consiste en la elaboración por parte del Jefe de Calidad de cada proyecto del Plan de Aseguramiento de la Calidad. En este plan es donde se planifica cada una de las revisiones o auditorías que serán realizadas al proyecto, además de especificar en qué fase de desarrollo se realizará y quiénes serán los encargados de efectuarla.
- **Consultar Información Técnica:** Este proceso consiste en consultar información técnica relacionada con los procesos de calidad del departamento, por parte de los Integrantes de Proyecto para mantenerse actualizados sobre los procesos de calidad y evitar errores en la codificación o en la elaboración de los expedientes de proyectos.

### 3.1.3 Reglas del Negocio

## Capítulo 3: Presentación de la solución propuesta

- Un Usuario puede ser: Un Integrante de Proyecto, Jefe de Proyecto, un Miembro del Grupo de Calidad o el Jefe de Calidad de Proyecto.
- El Jefe de Proyecto es un stakeholder del negocio.
- Si el Usuario es un Miembro del Grupo de Calidad no puede ser de otro tipo.
- Si el Usuario es el Jefe de Calidad de Proyecto no podrá ser de otro tipo.
- Sólo el Jefe de Calidad de Proyecto puede realizar el Plan de Aseguramiento de la Calidad.
- Cualquier Usuario podrá consultar la Información Técnica.
- Sólo los Miembros del Grupo de Calidad podrán realizar las revisiones y auditorías, y controlar las tareas de corrección.
- El Jefe de Calidad se puede comportar como un actor del negocio y también como un trabajador del negocio.

### 3.1.4 Diagrama de Casos de Uso del Negocio



## Capítulo 3: Presentación de la solución propuesta

### 3.1.5 Descripción textual de los Casos de Uso de Negocio

#### CUN Realizar Revisión

Caso de uso del negocio	Realizar Revisión
<b>Actores</b>	Integrante de Proyecto
<b>Resumen</b>	El caso de uso se inicia cuando los integrantes de proyectos entregan los artefactos que van a ser revisados por los Miembros del Grupo de Calidad. El Miembro del Grupo de Calidad realiza la revisión, elabora las no conformidades del artefacto si se encuentran defectos y se crea una nueva tarea de corrección.
<b>Casos de uso asociados</b>	----
<b>Acción del actor</b>	<b>Respuesta del proceso de negocio</b>
1. Los Integrantes de Proyectos entregan los artefacto(s).	2. El Miembro Grupo de Calidad recibe los artefactos. Se le asigna a un Miembro del Grupo de Calidad el artefacto a revisar.
	3. Se revisa el artefacto guiándose por los parámetros establecidos en las Listas de Chequeo definidas para esta revisión.
	4. Si existen defectos en los artefactos se elabora una no conformidad por cada defecto encontrado.
	5. Se establece la fecha límite para la corrección de los defectos encontrados y se asigna el responsable de realizar la corrección.
	6. Se crea una nueva tarea de corrección y se termina el caso de uso.
<b>Flujo Alternos de los eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del proceso del Negocio</b>
	4. Si el artefacto no presenta defectos se guarda como que ya pasó la revisión y termina el caso de uso.
<b>Poscondiciones</b>	Se ha realizado la revisión del artefacto.

## Capítulo 3: Presentación de la solución propuesta

Mejoras Propuestas	-
--------------------	---

### CUN Realizar Auditoría

Caso de uso del negocio	Realizar Auditoría
<b>Actores</b>	Integrantes de Proyecto
<b>Resumen</b>	El caso de uso se inicia cuando los integrantes de proyectos entregan los artefactos que van a ser auditados por los Miembros del Grupo de Calidad. El Miembro del Grupo de Calidad realiza la auditoría, elabora las no conformidades del artefacto si se encuentran defectos, se crea una nueva tarea de corrección y se emite la evaluación de la auditoría.
<b>Casos de uso asociados</b>	----
<b>Acción del actor</b>	<b>Respuesta del proceso de negocio</b>
1. Los Integrantes de Proyec -tos entregan los artefacto(s).	2. El Miembro del Grupo de Calidad recibe el artefacto(s) a auditar.
	3. Se audita el artefacto guiándose por los parámetros establecidos en las Listas de Chequeo definidas para esta auditoría.
	4. Si existen defectos en los artefactos se elabora una no conformidad por cada defecto encontrado.
	5. Se establece la fecha límite para la corrección de los defectos encontrados y se asigna el responsable de realizar la corrección.
	6. Se crea el Informe General de la auditoría que contiene la evaluación de la misma y se le entrega al Jefe de Proyecto.
7. El Jefe de Proyecto recibe el Informe general de la auditoría y termina el	

## Capítulo 3: Presentación de la solución propuesta

caso de uso.	
Flujo Alternos de los eventos	
Acción del Actor	Respuesta del proceso del Negocio
	4. Si el artefacto no presenta defectos se guarda como que ya paso la auditoría.
	5. Se crea el Informe General de la auditoría que contiene la evaluación de la misma y se le entrega a los Integrantes de Proyecto
6. El Jefe de Proyecto reciben el Informe general de la auditoria y termina el caso de uso.	
Poscondiciones	Se ha realizado la auditoría del artefacto.
Mejoras Propuestas	-

### CUN Controlar Tareas de Corrección.

Caso de uso del negocio	Controlar Tareas de Corrección
<b>Actores</b>	Grupo de Calidad, Integrantes de Proyecto
<b>Resumen</b>	El Grupo de Calidad le entrega las no conformidades a los integrantes de proyecto para que definan los responsables de las correcciones
<b>Casos de uso asociados</b>	----
Acción del actor	Respuesta del proceso de negocio
1. El Grupo de Calidad entrega el Listado de No Conformidades.	2. Los Integrantes de Proyecto, reciben el Listado de No Conformidades.
	3. Definen los responsables de las Tareas de Corrección.
	4. Informa a los Integrantes de Proyecto de la Tarea de Corrección
5. Los Integrantes de Proyecto reciben la Tarea de Corrección.	
6. Los Integrantes de Proyecto entregan el artefacto con la corrección realizada.	



## Capítulo 3: Presentación de la solución propuesta

	7. El Miembro del Grupo de Calidad revisa la corrección.
	8. Si la corrección fue efectiva, actualiza el listado de No Conformidades y termina el caso de uso.
Flujo Alternos de los eventos	
Acción del Actor	Respuesta del proceso del Negocio
	8. Si todavía persisten los defectos le envía una nueva Tarea de Corrección a los Integrantes de Proyecto para realizar la corrección nuevamente.
Poscondiciones	Se controló la corrección de los defectos encontrados.
Mejoras Propuestas	-

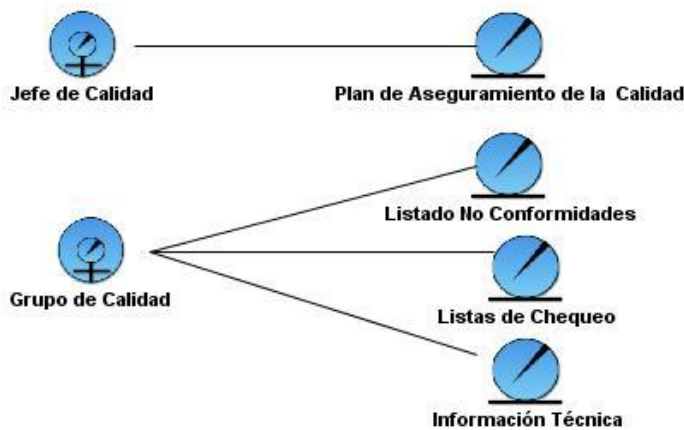
### CUN Establecer Plan de Aseguramiento de la Calidad.

Caso de uso del negocio	Establecer Plan de Aseguramiento de la Calidad.
<b>Actores</b>	Jefe de Calidad
<b>Resumen</b>	El Jefe de Proyecto entrega el cronograma de trabajo. El Jefe de Calidad recibe el cronograma de trabajo y establece en cuáles etapas serán auditadas o revisadas y el responsable de realizarlas. Notifica al Jefe de Proyecto de las etapas a ser auditadas o revisadas.
<b>Casos de uso asociados</b>	----
Acción del actor	Respuesta del proceso de negocio
1.El Jefe de Calidad recibe el Cronograma de Proyecto del Jefe de Proyecto	2 El Jefe de Calidad establece las etapas en las que se realizarán las revisiones o auditorías.
	3. El Jefe de Calidad establece los responsables de efectuar las revisiones o auditorías.
	4. El Jefe de Calidad informa al Jefe de Proyecto sobre las etapas en las que el proyecto será revisado o auditado.

## Capítulo 3: Presentación de la solución propuesta

5. El Jefe de Proyecto recibe la notificación sobre cuáles etapas serán auditadas o revisadas y termina el caso de uso	
Flujo Alternos de los eventos	
Acción del Actor	Respuesta del proceso del Negocio
Poscondiciones	Se ha elaborado el Plan de Aseguramiento de la Calidad. para el cronograma de trabajo del proyecto
Mejoras Propuestas	-

### 3.1.6 Modelo de Objetos



## 3.2 Requerimientos del Sistema

### 3.2.1 Funcionalidad

#### 3.2.1.1 Realizar Búsquedas de Documentos

## Capítulo 3: Presentación de la solución propuesta

Este requerimiento permite realizar la acción de buscar documentos dentro del expediente de proyecto, para agilizar los procesos de gestión de la calidad.

### 3.2.1.2 Sistema de Reportes

Este requerimiento permite la elaboración de reportes sobre las tareas de corrección y no conformidades, permitiendo una mejor visión de los procesos de calidad. Además permite que los reportes sean exportados fuera del sistema en distintos formatos de documentos.

### 3.2.1.3 Mostrar Información Técnica

Este requerimiento permite el acceso a información técnica relacionada con la gestión de la calidad que permita a los grupos de desarrollo mantenerse actualizados y registrar el desarrollo de los productos por las normas y estándares establecidos, que ayudan a evitar errores en la codificación y en la elaboración de las plantillas del expediente de proyecto.

### 3.2.1.4 Mostrar Noticias

Este requerimiento permite crear un sistema de noticias para los grupos de desarrollo sobre temas de interés para la gestión de la calidad.

### 3.2.1.5 Gestionar Información Técnica

Este requerimiento permite realizar las operaciones de Adicionar, Modificar, Eliminar una Información Técnica. Para adicionar una Información Técnica se necesitan llenar los campos: nombre documento, autor, fecha de edición, tipo de documento, dirección del archivo. Cuando se adiciona una Información Técnica debe mostrarse el mensaje: “Se ha insertado una Información Técnica”.

### 3.2.1.6 Gestionar Noticias

Este requerimiento permite realizar las operaciones de Adicionar, Modificar, Eliminar una Noticia. Para adicionar una Noticia se necesitan llenar los campos: nombre noticia, tipo, cuerpo noticia. Cuando se adiciona una Noticia debe mostrarse el mensaje: “Se ha insertado una Noticia”.

### 3.2.1.7 Gestionar Documentos de Proyecto

Este requerimiento permite realizar las operaciones de Adicionar, Modificar, Eliminar un Documento de Proyecto. Para adicionar un Documento de Proyecto se necesitan llenar los campos: nombre documento, fecha de creación, autor, tipo, dirección, descripción. Cuando se adiciona una Noticia debe mostrarse el mensaje: “Se ha insertado un Documento de Proyecto”.

### 3.2.1.8 Gestionar No Conformidades

## Capítulo 3: Presentación de la solución propuesta

Este requerimiento permite realizar las operaciones de Adicionar, Modificar, Eliminar una No Conformidad. Para adicionar una No Conformidad se necesitan llenar los campos: no conformidad, elemento, clasificación, estado, respuesta equipo de desarrollo, proyecto. Cuando se adiciona una No Conformidad debe mostrarse el mensaje: “Se ha insertado una No Conformidad”.

### **3.2.1.9 Gestionar Listas de Chequeo**

Este requerimiento permite realizar las operaciones de Adicionar, Modificar, Eliminar una Lista de Chequeo. Para adicionar una Lista de Chequeo se necesitan llenar los campos: nombre lista, proyecto, descripción, listado de elementos. Cuando se adiciona una Lista de Chequeo debe mostrarse el mensaje: “Se ha insertado una Lista de Chequeo”.

### **3.2.1.10 Gestionar Tareas de Corrección**

Este requerimiento permite realizar las operaciones de Adicionar, Modificar, Eliminar una Tarea de Corrección. Para adicionar una Tarea de Corrección se necesitan llenar los campos: nombre tarea corrección, descripción de tarea, fecha inicio, fecha límite, responsables corrección, nombre no conformidad. Cuando se adiciona una No Conformidad debe mostrarse el mensaje: “Se ha insertado una Tarea de Corrección”.

### **3.2.1.11 Gestionar Plan de Aseguramiento de la Calidad.**

Este requerimiento permite realizar las operaciones de Adicionar, Modificar, Eliminar un Plan de Aseguramiento de la Calidad. Para adicionar un Plan de Aseguramiento de la Calidad se necesitan llenar los campos: nombre Plan de Aseguramiento de la Calidad, proyecto, descripción, listado de actividades. Cuando se adiciona un Plan de Aseguramiento de la Calidad debe mostrarse el mensaje: “Se ha insertado un Plan de Aseguramiento de la Calidad.”.

### **3.2.1.12 Gestionar Proyectos**

Este requerimiento permite realizar las operaciones de Adicionar, Modificar, Eliminar un Proyecto. Para adicionar un Proyecto se necesitan llenar los campos: nombre proyecto, descripción, facultad, fase de desarrollo. Cuando se adiciona un Proyecto debe mostrarse el mensaje: “Se ha insertado un Proyecto”.

### **3.2.1.13 Gestionar Notificaciones**

Este requerimiento permite realizar las operaciones de Enviar y Eliminar una Notificación. Para enviar una Notificación se necesitan llenar los campos: nombre notificación, asunto, destinatarios,

## Capítulo 3: Presentación de la solución propuesta

cuerpo notificación. Cuando se envía una Notificación debe mostrarse el mensaje: “Se ha enviado una Notificación”.

### 3.2.1.14 Gestionar Permisos de Acceso

Este requerimiento permite realizar las operaciones de Adicionar, Modificar, Eliminar un Permiso de Acceso. Para adicionar un Permiso de Acceso se necesitan llenar los campos: usuario, nivel de acceso, rol, proyecto. Cuando se adiciona un Permiso de Acceso debe mostrarse el mensaje: “Se ha insertado un Permiso de Acceso”.

### 3.2.1.15 Autenticarse

Este requerimiento permite realizar la operación de autenticarse en el sistema para todos los usuarios del mismo. Para autenticarse se deben llenar los campos: usuario, contraseña. Al autenticarse se muestra el mensaje: “Autenticación completada, Bienvenido al sistema”

### 3.2.1.16 Listado de Notificaciones

Este requerimiento permite realizar las operaciones Leer y Eliminar Notificación del Perfil de Usuario. Este requerimiento permite mantener actualizado a los usuarios sobre qué tareas deberán cumplir.

### 3.2.1.17 Generar Documentos

Con este requerimiento se genera un documento en cualquiera de los siguientes formatos (Word, pdf, Excel). El documento generado contiene los datos resultantes de los reportes, no conformidades, tareas de corrección, listas de chequeo o Plan de Aseguramiento de la Calidad.

## 3.2.2 Usabilidad

**3.2.2.1** El sistema debe poder ser usado por cualquier persona que tenga conocimientos básicos de computación.

**3.2.2.2** Debe poseer una interfaz amigable, que contenga los colores característicos de la UCI.

**3.2.2.3** La información deberá estar disponible en todo momento, limitada solamente por las restricciones de acuerdo a las políticas de seguridad definidas.

## 3.2.3 Fiabilidad

## Capítulo 3: Presentación de la solución propuesta

**3.2.3.1** Al sistema se accederá a través de la autenticación convencional: usuario y contraseña. La autenticación se realizará utilizando el controlador de dominio (LDAP) de la universidad, que será el encargado de verificar el usuario y la contraseña. Además se utilizarán los servicios web que se brindan en la UCI para conocer los datos personales de cada usuario.

**3.2.3.2** Cada usuario debe tener sólo los permisos necesarios para realizar las operaciones que le sean permitidas en el sistema, impidiendo que pueda acceder a áreas para las cuales no posee permisos de acceso.

**3.2.3.3** Debe mantenerse la consistencia de los datos en correspondencia con la realidad.

### 3.2.4 Eficiencia

**3.2.4.1** El tiempo de respuesta promedio por transacción de las peticiones realizadas al sistema será de 20 segundos y como máximo de 2 minutos.

**3.2.4.1** El número de clientes o transacciones que el sistema puede alojar al mismo tiempo es de 80.

### 3.2.5 Soporte

**3.2.5.1** La aplicación recibirá mantenimiento cada un tiempo determinado por el equipo de desarrollo y los clientes.

### 3.2.6 Restricciones de Diseño

## Capítulo 3: Presentación de la solución propuesta

**3.2.6.1** El sistema debe tener una apariencia profesional y un diseño gráfico sencillo, con la utilización de las tonalidades de los colores azul oscuro, azul claro y gris fundamentalmente, pues son los colores representativos de la Universidad.

**3.2.6.2** La aplicación debe de ser compatible con los siguientes navegadores: Internet Explorer y Mozilla.

**3.2.6.3** La aplicación debe ser compatible con los Sistemas Operativos: Windows 2000 NT, Windows XP y GNU/Linux.

### **3.2.7 Requisitos para la documentación de usuarios en línea y ayuda del sistema.**

**3.2.7.1** La aplicación contará con un mapa del sitio para la orientación del usuario que esté navegando.

### **3.2.8 Interfaz**

#### **3.2.8.1 Interfaces de Usuario**

**3.2.8.1.1** El sistema debe tener una apariencia profesional y un diseño gráfico sencillo, con la utilización de las tonalidades de los colores azul oscuro, azul claro y gris fundamentalmente, pues son los colores representativos de la Universidad.

#### **3.2.8.2 Interfaces de Hardware**

**3.2.8.2.1** Las computadoras que utilizarán el software a desarrollar deberán tener 128 MB de Memoria tipo RAM como mínimo.

#### **3.2.8.3 Interfaces de Software**

**3.2.9.3.1** Las computadoras que utilizarán el software deben tener instalado:

- Windows 2000 NT, Windows XP Profesional ó GNU/Linux en cualquier distribución.
- Navegador Web compatible con IE 4.0 o superior, Mozilla, Opera.

## Capítulo 3: Presentación de la solución propuesta

**3.2.8.3.2** El servidor (PC) que alojará la aplicación y la Base de Datos deberá tener instalado un servidor Apache con versión 5 de PHP, framework Symfony y PostgreSQL versión 8.3.

### 3.2.8.4 Interfaces de Comunicación

Se tendrá acceso a todos los servicios y aplicaciones brindados por la Universidad.

### 3.2.9 Requisitos legales, de copyright y otros

- El sistema debe ajustarse y regirse por la ley, decretos leyes, decretos, resoluciones y manuales (órdenes) establecidos, que norman los procesos que serán automatizados.
- Las herramientas de desarrollo son libres.
- Como producto, Geoinformática se distribuye amparado bajo las normativas legales establecidas en el registro comercial emitido por las entidades jurídicas de la Universidad de las Ciencias Informáticas.

## 3.3 Descripción del Sistema Propuesto

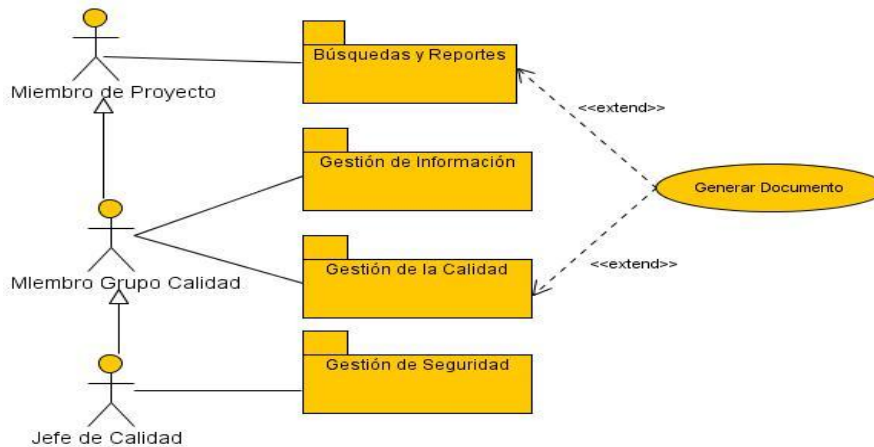
### 3.3.1 Descripción de los actores

Actor	Descripción
<b>Miembro de Proyecto</b>	Sólo puede consultar informaciones, documentos y sus notificaciones.
<b>Miembro de Grupo de Calidad</b>	Es el encargado de gestionar las No Conformidades, Listas de Chequeo, Tareas de Corrección, Notificaciones y los Documentos de Proyecto.
<b>Jefe de Calidad</b>	Es el encargado de Gestionar el Plan de Aseguramiento de la Calidad., Proyectos y los Permisos de Acceso.



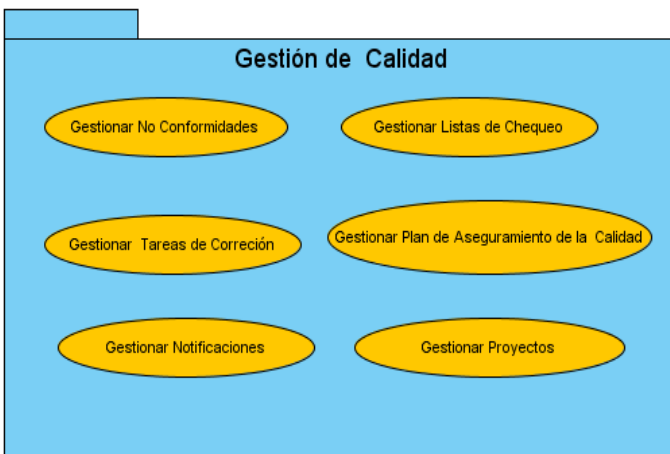
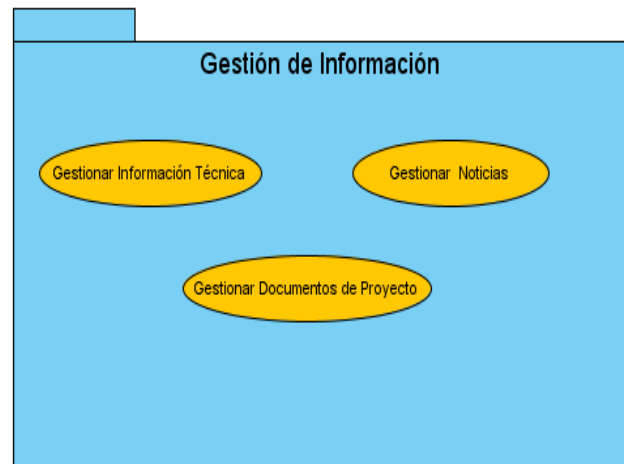
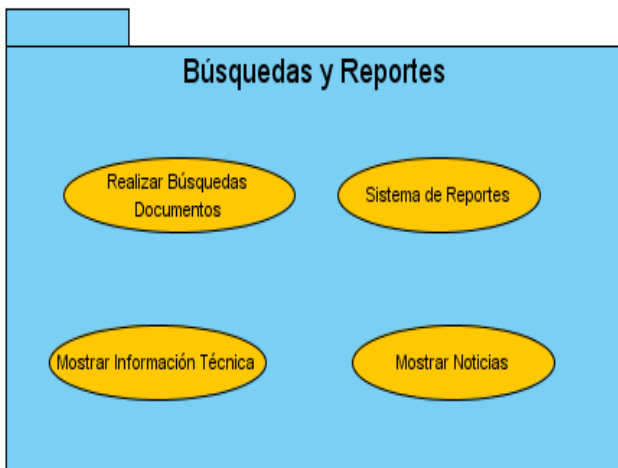
# Capítulo 3: Presentación de la solución propuesta

## 3.3.2 Diagrama de Casos de Uso del Sistema



### 3.3.2.1 Subsistemas

#### Diagramas de Paquetes



## Capítulo 3: Presentación de la solución propuesta

### 3.3.2.2 Especificación de los Casos de Uso

#### CU Gestionar Documentos de Proyecto.

Caso de Uso:	Gestionar Documentos de Proyecto
Actores:	Miembro de Grupo de Calidad
Resumen:	El caso de uso se inicia cuando el Miembro de Grupo de Calidad selecciona la opción “Gestionar Documentos de Proyecto” del menú “Gestión de Información y Noticias”. El sistema le muestra las acciones a realizar y en dependencia de las mismas muestra el formulario donde puede introducir los datos necesarios para llevar a cabo la acción deseada. El caso de uso termina una vez realizada la acción deseada y el Miembro de Grupo de Calidad sale de la aplicación.
Precondiciones:	. El usuario debe estar autenticado como Miembro de Grupo de Calidad
Referencias	
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Miembro de Grupo de Calidad selecciona la opción “Gestionar Documentos de Proyecto” del menú “Gestión de Información y Noticias”.	2. El sistema muestra una página con las opciones “Listado Documentos de Proyecto” y “Adicionar” en forma de pestañas.

## Capítulo 3: Presentación de la solución propuesta

<p>3. El Miembro del Grupo de Calidad selecciona la opción:</p> <ul style="list-style-type: none"> <li>• Listado Documentos de Proyecto (Ir a la sección Listado Documentos de Proyecto)</li> <li>• Adicionar (Ir a la sección Adicionar)</li> </ul>	
Flujo Normal de Eventos	
Sección “Adicionar Documento de Proyecto”	
Acción del Actor	Respuesta del Sistema
	<p>1. El sistema muestra un formulario que permite adicionar un nuevo documento de proyecto, con los siguientes campos: nombre documento, fecha de creación, autor, tipo, dirección, descripción.</p>
<p>2. El Miembro de Grupo de Calidad llena los datos del formulario</p>	
<p>3. El Miembro de Grupo de Calidad da clic en el botón “Adicionar”</p>	<p>4. El sistema comprueba que no exista otro Documento de Proyecto con el mismo nombre.</p>
	<p>5. El sistema adiciona el nuevo Documento de Proyecto y muestra un mensaje de</p>

## Capítulo 3: Presentación de la solución propuesta

	confirmación: “Se ha insertado el Documento de Proyecto”.
<i>Prototipo de Interfaz</i>	
Flujos Alternos	
2. El Miembro de Grupo de Calidad deja algún campo vacío.	
3. El Miembro de Grupo de Calidad da clic en el botón “Adicionar”	4. El sistema selecciona en rojo los campos que no pueden quedar vacíos y muestra un mensaje: “Existen campos vacíos”
<i>Prototipo de Interfaz</i>	
	5. El sistema muestra un mensaje de error porque existe otro Documento de Proyecto con el mismo nombre, emitiendo un mensaje “Ya existe un Documento de Proyecto con ese nombre”
Flujo Normal de Eventos	
Sección “Listado Información Técnica”	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra un listado de todos los Documentos de Proyecto que contiene, los datos de la lista serán: nombre documento, fecha de creación, descripción, autor, tipo, dirección.
2. El Miembro de Grupo de Calidad selecciona la opción que desea realizar.	

## Capítulo 3: Presentación de la solución propuesta

Si selecciona:	
<ul style="list-style-type: none"> <li>• Modificar (Ir a la sección “Modificar Documento de Proyecto”).</li> <li>• Eliminar (Ir a la sección “Eliminar Documento de Proyecto”).</li> <li>• Ver Detalles (Ir a la sección “Ver Detalles Documento de Proyecto”)</li> </ul>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema.
Flujo Normal de Eventos	
Sección “Modificar Información Técnica”	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra un formulario con los mismos campos que el de adicionar llenados con los datos de la Información Técnica que se desea modificar.
2. El Miembro de Grupo de Calidad realiza las actualizaciones que desea y envía los datos.	3. El sistema modifica los datos de la Información Técnica y muestra el mensaje: “Información Técnica actualizada”.
<i>Prototipo de Interfaz</i>	
Flujo Normal de los Eventos.	
Sección “Eliminar Información Técnica”	

## Capítulo 3: Presentación de la solución propuesta

Acción del Actor	Respuesta del Sistema
	1. El sistema muestra un mensaje de confirmación: “Esta seguro que desea eliminar este Documento de Proyecto”.
2. El Miembro de Grupo de Calidad selecciona “Aceptar”	3. El sistema elimina el Documento de Proyecto, actualiza el listado y muestra el mensaje: “Documento de Proyecto eliminado satisfactoriamente”.
<i>Prototipo de Interfaz</i>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
2. El Miembro de Grupo de Calidad selecciona “Cancelar”	3. El sistema no elimina el Documento de Proyecto
<i>Prototipo de Interfaz</i>	
Flujo Normal de los Eventos.	
Sección “Mostrar Detalles de Información Técnica”	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra en una ventana en forma de tabla toda la información del Documento de Proyecto seleccionado.
<i>Prototipo de Interfaz</i>	
Poscondiciones	El sistema queda actualizado después de efectuada cualquier acción descrita anteriormente

### CU Gestionar No Conformidades

Caso de Uso:	Gestionar No Conformidades
--------------	----------------------------

## Capítulo 3: Presentación de la solución propuesta

Actores:	Miembro de Grupo de Calidad
Resumen:	El caso de uso se inicia cuando el Miembro del Grupo de Calidad selecciona la opción “Gestionar No Conformidades” del menú “Gestión de la Calidad”. El sistema le muestra las acciones a realizar y en dependencia de las mismas, muestra el formulario donde debe introducir los datos necesarios para realizar la acción deseada. El caso de uso termina una vez que se ha realizado la acción deseada y el Miembro del Grupo de Calidad sale de la aplicación.
Precondiciones :	El usuario debe estar autenticado y tener los permisos de Miembro de Grupo de Calidad.
Referencias	
Prioridad	Crítico
<b>Flujo Normal de Eventos</b>	
Acción del Actor	Respuesta del Sistema
1. El Miembro del Grupo de Calidad selecciona la opción “Gestionar No Conformidades” del menú “Gestión de la Calidad”.	2. El sistema muestra una página con las opciones “Listado No Conformidades”, “Adicionar” y “Generar Documento” en forma de pestañas.
3. El Miembro del Grupo de Calidad selecciona la opción: <ul style="list-style-type: none"> <li>• Listado No Conformidades (Ir a la sección Listado de No conformidades)</li> <li>• Adicionar (Ir a la</li> </ul>	

## Capítulo 3: Presentación de la solución propuesta

sección Adicionar) <ul style="list-style-type: none"> <li>• Generar Documento (Ir a la sección Generar Documento).</li> </ul>	
<b>Flujo Normal de Eventos</b>	
<b>Sección “Adicionar No Conformidad”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. El sistema muestra el formulario con los campos a llenar correspondiente a esta opción: no conformidad, elemento, clasificación, estado, fecha detección, proyecto y acciones.
2. El Miembro de Grupo de Calidad llena los datos del formulario	
3. El Miembro de Grupo de Calidad da clic en el botón “Adicionar”	4. El sistema comprueba que no exista otra No Conformidad con el mismo nombre para el mismo proyecto.
	5. El sistema adiciona la No Conformidad y muestra un mensaje de confirmación: “No Conformidad adicionada satisfactoriamente”.
<i>Prototipo de Interfaz</i>	
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
2. El Miembro de Grupo de Calidad deja algún campo vacío.	



## Capítulo 3: Presentación de la solución propuesta

3. El Miembro de Grupo de Calidad da clic en el botón “Adicionar”	4. El sistema selecciona en rojo los campos que no pueden quedar vacíos y muestra un mensaje: “Existen campos vacíos”
<i>Prototipo de Interfaz</i>	
	5. El sistema muestra un mensaje de error porque existe otra No Conformidad con el mismo nombre para el mismo proyecto, emitiendo un mensaje “Ya existe una No Conformidad con ese nombre para este proyecto”.
<i>Prototipo de Interfaz</i>	
Flujo Normal de Eventos	
Sección “Listado No Conformidades”	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra un listado de todas las No Conformidades que contiene, los datos de la lista serán: no conformidad, elemento, clasificación, estado, respuesta equipo de desarrollo, creador, fecha de detección, proyecto. Además el listado presentará un filtrado de información por: proyecto, creador, estado.
2. El Miembro de Grupo de Calidad selecciona la opción que desea realizar. Si selecciona: <ul style="list-style-type: none"> <li>• Modificar (Ir a la sección “Modificar</li> </ul>	

## Capítulo 3: Presentación de la solución propuesta

<p>No Conformidad”).</p> <ul style="list-style-type: none"> <li>• Eliminar (Ir a la sección “Eliminar No Conformidad”).</li> <li>• Ver Detalles (Ir a la sección “Ver Detalles de No Conformidad”)</li> </ul>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema.
Flujo Normal de Eventos	
Sección “Generar Documento”	
Acción del Actor	Respuesta del Sistema
	<p>1. El sistema muestra un listado de todas las No Conformidades que contiene, los datos de la lista serán: no conformidad, elemento, clasificación, estado, respuesta equipo de desarrollo, creador, fecha de detección, proyecto. Además el listado presentará un filtrado de información por: proyecto, creador, estado.</p>
<p>2. El Miembro de Proyecto selecciona las No conformidades que desea incluir en el documento</p>	
<p>3. El Miembro de</p>	

## Capítulo 3: Presentación de la solución propuesta

Proyecto pulsa “Generar Documento” (Se invoca el Caso de Uso extendido Generar Documento).	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Flujo Normal de Eventos	
Sección “Modificar No Conformidad”	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra un formulario con los mismos campos que el de adicionar llenados con los datos de la No Conformidad que se desea modificar.
2. El Miembro de Grupo de Calidad realiza las actualizaciones que desea y envía los datos.	3. El sistema modifica los datos de la No Conformidad y muestra el mensaje: “No Conformidad actualizada”.
<i>Prototipo de Interfaz</i>	
Flujo Normal de los Eventos.	
Sección “Eliminar No Conformidad”	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra un mensaje de confirmación: “Está seguro que desea eliminar esta No Conformidad”.
2. El Miembro de Grupo de Calidad selecciona “Aceptar”	3. El sistema elimina la No Conformidad, actualiza el listado y muestra el mensaje: “No Conformidad

## Capítulo 3: Presentación de la solución propuesta

	eliminada satisfactoriamente”.
<i>Prototipo de Interfaz</i>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
2. El Miembro de Grupo de Calidad selecciona “Cancelar”	3. El sistema no elimina la No Conformidad
<i>Prototipo de Interfaz</i>	
Flujo Normal de los Eventos.	
Sección “Mostrar Detalles No Conformidad”	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra en una ventana en forma de tabla toda la información de la No Conformidad seleccionada.
<i>Prototipo de Interfaz</i>	
Poscondiciones	El sistema queda actualizado después de efectuada cualquier acción descrita anteriormente

### CU Gestionar Listas de Chequeo

Caso de Uso:	Gestionar Listas de Chequeo
Actores:	Miembro de Grupo de Calidad
Resumen:	El caso de uso se inicia cuando el Miembro de Grupo de Calidad selecciona la opción “Gestionar Listas de Chequeo” del menú “Gestión de Calidad”. El sistema le muestra las acciones a realizar y en dependencia de las mismas, muestra el formulario donde debe introducir los datos necesarios para realizar la acción deseada. El caso de uso termina una vez que se ha realizado la

## Capítulo 3: Presentación de la solución propuesta

	acción deseada y el Miembro de Grupo de Calidad sale de la aplicación.
Precondiciones:	El usuario debe estar autenticado y tener los permisos de Miembro de Grupo de Calidad.
Referencias	
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Miembro de Grupo de Calidad selecciona la opción “Gestionar Listas de Chequeo” del menú “Gestión de Calidad”.	2. El sistema muestra una ventana con las opciones “Listado de Listas de Chequeo” y “Adicionar” en forma de pestañas.
3. 3. El Miembro de Grupo de Calidad selecciona la opción: <ul style="list-style-type: none"> <li>• Listado Listas de Chequeo (Ir a la sección Listado Listas de Chequeo )</li> <li>• Adicionar (Ir a la sección Adicionar)</li> </ul>	
Flujo Normal de Eventos	
Sección “Adicionar Lista de Chequeo”	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra un formulario que permitirá adicionar una nueva Lista de Chequeo, cuyos campos serán: nombre lista,

## Capítulo 3: Presentación de la solución propuesta

	proyecto, Creador, fase de desarrollo, fecha de creación, acciones.
2. El Miembro de Grupo de Calidad llena los campos del formulario	
3. El Miembro de Grupo de Calidad pulsa "Adicionar"	4. El sistema verifica que no exista una Lista de Chequeo para el proyecto seleccionado.
	5. El sistema adiciona la Lista de Chequeo y muestra el mensaje: "Lista de Chequeo adicionada correctamente".
<i>Prototipo de Interfaz</i>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
2. El Miembro de Grupo de Calidad deja campos del formulario en blanco.	
3. El Miembro de Grupo de Calidad pulsa "Adicionar"	4. El sistema selecciona en rojo los campos que no pueden quedar vacíos y muestra un mensaje: "Existen campos vacíos"
<i>Prototipo de Interfaz</i>	
	4. El sistema muestra un mensaje de error puesto que ya existe una Lista de Chequeo para el proyecto seleccionado con el mensaje: "Ya existe una Lista de Chequeo para el proyecto seleccionado"
<i>Prototipo de Interfaz</i>	
Flujo Normal de Eventos	

## Capítulo 3: Presentación de la solución propuesta

Sección "Listado Listas de Chequeo"	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra un listado de todas las Listas de Chequeo que contiene en una lista cuyos datos serán: nombre lista, proyecto, creador, fecha creación, fecha actualización.
<p>2. El Miembro de Grupo selecciona la opción que desea realizar. Si selecciona:</p> <ul style="list-style-type: none"> <li>• Modificar (Ir a la sección "Modificar Lista de Chequeo").</li> <li>• Eliminar (Ir a la sección "Eliminar Lista de Chequeo").</li> <li>• Ver Detalles (Ir a la sección "Ver Detalles Lista de Chequeo").</li> <li>• Generar Documento (Se invoca el CU extendido Generar Documento)</li> </ul>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema.
Flujo Normal de Eventos	
Sección "Modificar Listas de Chequeo"	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra un formulario con los

## Capítulo 3: Presentación de la solución propuesta

	mismos campos que el de adicionar llenados con los datos de la Lista de Chequeo que se desea modificar.
2. El Miembro de Grupo de Calidad realiza las actualizaciones que desea y envía los datos.	3. El sistema modifica los datos de la Lista de Chequeo y muestra el mensaje: “Lista de Chequeo actualizada”.
<i>Prototipo de Interfaz</i>	
Flujo Normal de los Eventos.	
Sección “Eliminar Listas de Chequeo”	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra un mensaje de confirmación: “Está seguro que desea eliminar esta Lista de Chequeo”.
2. El Miembro de Grupo de Calidad selecciona “Aceptar”	3. El sistema elimina la Lista de Chequeo, actualiza el listado y muestra el mensaje: “Lista de Chequeo eliminada satisfactoriamente”.
<i>Prototipo de Interfaz</i>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
2. El Miembro de Grupo de Calidad selecciona “Cancelar”	3. El sistema no elimina la Lista de Chequeo.
<i>Prototipo de Interfaz</i>	
Flujo Normal de los Eventos.	
Sección “Mostrar Detalles Lista de Chequeo”	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra todos los detalles de



## Capítulo 3: Presentación de la solución propuesta

	la Lista de Chequeo. Además muestra un listado de todos los que contiene, cuyos datos serán: nombre elemento, descripción.
<p>2. El Miembro de Grupo de Calidad selecciona la opción que desea realizar. Si selecciona:</p> <ul style="list-style-type: none"> <li>• Adicionar (Ir a la sección “Adicionar Elemento de Lista de Chequeo”).</li> <li>• Modificar (Ir a la sección “Modificar Elemento de Lista de Chequeo”).</li> <li>• Eliminar (Ir a la sección “Eliminar Elemento de Lista de Chequeo”).</li> <li>• Ver Detalles (Ir a la sección “Ver Detalles Elemento de Lista de Chequeo”).</li> </ul>	
Flujo Normal de los Eventos.	
Sección “Adicionar Elemento”	
Acción del Actor	Respuesta del Sistema
	<p>1. El sistema muestra un formulario que permitirá adicionar un nuevo Elemento de Lista de Chequeo a la Lista de Chequeo. Los campos serán: nombre elemento, descripción, listado de condiciones.</p>

## Capítulo 3: Presentación de la solución propuesta

2 El Miembro de Grupo de Calidad llena los campos del formulario	
3. El Miembro de Grupo de Calidad pulsa “Adicionar”	4. El sistema verifica que no exista un Elemento de Lista de Chequeo con el mismo nombre dentro de la Lista de Chequeo.
	5. El sistema adiciona el elemento y muestra el mensaje “Elemento de Lista de Chequeo adicionado correctamente”.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
2. El Miembro de Grupo de Calidad deja campos del formulario en blanco.	
3. El Miembro de Grupo de Calidad pulsa “Adicionar”	4. El sistema selecciona en rojo los campos que no pueden quedar vacíos y muestra un mensaje: “Existen campos vacíos”
<i>Prototipo de Interfaz</i>	
	4.El sistema muestra un mensaje de error puesto que ya existe un elemento con ese nombre para la lista de chequeo con el mensaje: “Ya existe un elemento con ese nombre para esta Lista de Chequeo”
<b>Flujo Normal de los Eventos.</b>	
<b>Sección “Modificar Elemento”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. El sistema muestra un formulario con los mismos campos que el de adicionar

## Capítulo 3: Presentación de la solución propuesta

	llenados con los datos del elemento que se desea modificar.
2. El Miembro de Grupo de Calidad realiza las actualizaciones que desea y envía los datos.	3. El sistema modifica los datos del elemento y muestra el mensaje: "Elemento actualizado".
<b>Flujos Alternos</b>	
Acción del Actor	Respuesta del Sistema
<i>Prototipo de Interfaz</i>	
Flujo Normal de los Eventos.	
Sección "Eliminar Elemento"	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra un mensaje de confirmación: "Está seguro que desea eliminar este Elemento".
2. El Miembro de Grupo de Calidad selecciona "Aceptar"	3. El sistema elimina el Elemento, actualiza el listado y muestra el mensaje: "Elemento eliminado satisfactoriamente".
<b>Flujos Alternos</b>	
Flujos Alternos	Respuesta del Sistema
2. El Miembro de Grupo de Calidad selecciona "Cancelar"	3. El sistema no elimina el Elemento.
<i>Prototipo de Interfaz</i>	
Flujo Normal de los Eventos.	
Sección "Detalles Elemento"	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra todos los detalles del

## Capítulo 3: Presentación de la solución propuesta

	Elemento seleccionado.
<i>Prototipo de Interfaz</i>	
Poscondiciones	El sistema queda actualizado después de efectuada cualquier acción descrita anteriormente

### CU Gestionar Tareas de Corrección.

Caso de Uso:	Gestionar Tareas de Corrección
Actores:	Miembro de Grupo de Calidad
Resumen:	El caso de uso se inicia cuando el Miembro de Grupo de Calidad selecciona la opción “Gestionar Tareas de Corrección” del menú “Gestión de Calidad”. El sistema le muestra las acciones a realizar y en dependencia de las mismas, muestra el formulario donde debe introducir los datos necesarios para realizar la acción deseada. El caso de uso termina una vez que se ha realizado la acción deseada y el Miembro del Grupo de Calidad sale de la aplicación.
Precondiciones:	El usuario debe estar autenticado y tener los permisos de Miembro de Grupo de Calidad.
Referencias	
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Miembro de Grupo de Calidad selecciona la opción “Gestionar Tareas de Corrección” del menú “Gestión de Calidad”.	2. El sistema muestra una página con las opciones “Listado Tareas de Corrección”, “Adicionar” y “Generar Documento” en forma de pestañas.
3. El Miembro del Grupo de	

## Capítulo 3: Presentación de la solución propuesta

<p>Calidad selecciona la opción:</p> <ul style="list-style-type: none"> <li>Listado Tareas de Corrección (Ir a la sección Listado Tareas de Corrección)</li> <li>Adicionar (Ir a la sección Adicionar)</li> </ul>	
Flujo Normal de Eventos	
Sección “Adicionar Tarea de Corrección”	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra un listado de las no conformidades que están sin corregir en el sistema. Los datos de la lista serán: nombre no conformidad, tipo no conformidad, tipo actividad, nombre de artefacto, proyecto, fecha inicio, fecha de fin, estado. Además el listado presentara un filtrado de información por: proyecto, creador, estado.
2. El Miembro de Grupo de Calidad selecciona la no conformidad que desea para crear la tarea de corrección.	3. El sistema muestra un formulario que permitirá crear una nueva tarea de corrección con los siguientes campos: nombre tarea corrección, descripción de tarea, fecha inicio, fecha límite, responsables corrección.
4. El Miembro de Grupo de Calidad llena los campos del formulario	
5. El Miembro de Grupo de	6. El sistema verifica que no exista una tarea

## Capítulo 3: Presentación de la solución propuesta

Calidad pulsa “Crear Tarea”.	de corrección para la no conformidad seleccionada.
	7. El sistema adiciona la tarea de corrección y muestra el mensaje: “Tarea de Corrección creada correctamente”.
<i>Prototipo de Interfaz</i>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
2. El Miembro de Grupo de Calidad deja algún campo vacío.	
3. El Miembro de Grupo de Calidad da clic en el botón “Crear Tarea”	4. El sistema selecciona en rojo los campos que no pueden quedar vacíos y muestra un mensaje: “Existen campos vacíos”
<i>Prototipo de Interfaz</i>	
	5. El sistema muestra un mensaje de error porque existe otra tarea de corrección para la misma no conformidad con el mensaje: “Ya existe una tarea de corrección para la no conformidad seleccionada”.
<i>Prototipo de Interfaz</i>	
Flujo Normal de Eventos	
Sección “Listado Tareas de Corrección”	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra un listado de todas las tareas de corrección que contiene, los datos de la lista serán: nombre tarea de corrección, revisor, nombre no conformidad, fecha inicio, fecha límite,

## Capítulo 3: Presentación de la solución propuesta

	estado, proyecto. Además el listado presentara un filtrado de información por: proyecto, creador, estado.
<p>2. El Miembro de Grupo de Calidad selecciona la opción que desea realizar. Si selecciona:</p> <ul style="list-style-type: none"> <li>• Modificar (Ir a la sección “Modificar tarea de corrección”).</li> <li>• Eliminar (Ir a la sección “Eliminar tarea de corrección”).</li> <li>• Ver Detalles (Ir a la sección “Ver Detalles de tarea de corrección ”)</li> </ul>	
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema.</b>
<b>Flujo Normal de Eventos</b>	
<b>Sección “Generar Documento”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema.</b>
	<p>1. El sistema muestra un listado de todas las tareas de corrección que contiene, los datos de la lista serán: nombre tarea de corrección, revisor, responsables, nombre no conformidad, fecha inicio, fecha límite, estado, proyecto. Además el listado presentara un filtrado de información por:</p>

## Capítulo 3: Presentación de la solución propuesta

	proyecto, creador, estado.
2. El Miembro de Grupo de Calidad selecciona las tareas de corrección que incluirá en el documento.	
3. El Miembro de Proyecto pulsa “Generar Documento” (Se invoca el Caso de Uso extendido Generar Documento).	
<b>Flujos Alternos</b>	
Acción del Actor	Respuesta del Sistema.
<b>Flujo Normal de Eventos</b>	
<b>Sección “Modificar Tarea de Corrección”</b>	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra un formulario con los mismos campos que el de adicionar llenados con los datos de la Tarea de Corrección que se desea modificar.
2. El Miembro de Grupo de Calidad realiza las actualizaciones que desea y envía los datos.	3. El sistema modifica los datos de la Tarea de Corrección y muestra el mensaje: “Tarea de Corrección actualizada”.
<i>Prototipo de Interfaz</i>	
<b>Flujo Normal de los Eventos.</b>	
<b>Sección “Eliminar Tarea de Corrección”</b>	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra un mensaje de



## Capítulo 3: Presentación de la solución propuesta

	confirmación: “Esta seguro que desea eliminar esta Tarea de Corrección”.
2. El Miembro de Grupo de Calidad selecciona “Aceptar”	3. El sistema elimina la Tarea de Corrección, actualiza el listado y muestra el mensaje: “Tarea de Corrección eliminada satisfactoriamente”.
<i>Prototipo de Interfaz</i>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
2. El Miembro de Grupo de Calidad selecciona “Cancelar”	3. El sistema no elimina la Tarea de Corrección
<i>Prototipo de Interfaz</i>	
Flujo Normal de los Eventos.	
Sección “Mostrar Detalles Tarea de Corrección”	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra en una ventana en forma de tabla toda la información de la Tarea de Corrección seleccionada.
<i>Prototipo de Interfaz</i>	
Poscondiciones	El sistema queda actualizado después de efectuada cualquier acción descrita anteriormente

Para ver el resto de las descripciones consultar Anexo 1.

## Capítulo 3: Presentación de la solución propuesta

### 3.4 Conclusiones

Durante el desarrollo del presente capítulo, se ha profundizado el estudio de los procesos de negocio que rigen el desarrollo de las actividades de la Gestión de la Calidad, dentro del Departamento de Geoinformática. Se han identificado los actores y trabajadores que intervienen en los procesos y con cuáles entidades y actividades interactúan estos, permitiendo de esta manera, determinar las problemáticas actuales y en base a ellas determinar las posibles mejoras. Se identificaron los requerimientos funcionales y no funcionales que debe tener el sistema a desarrollar, haciendo una profunda descripción de todos sus casos de usos, que servirán de base para el diseño e implementación de la futura aplicación.

### Introducción

En el presente capítulo se realizan los modelos de Diseño e Implementación, así como los estándares de codificación y las pruebas del sistema resultante. El Diseño presenta cómo está construido el sistema, lo cual se realiza a través de los diagramas de clases web, que tienen la finalidad de describir la interacción entre las distintas páginas de la aplicación. También se muestra el diagrama de clases persistente de la base de datos del sistema, al igual que los principios de diseño utilizados. Se explican las generalidades de la implementación, se muestra el Modelo de Despliegue, que contiene el Diagrama de Despliegue y la explicación de sus componentes físicos, el Modelo de Implementación y la Prueba del sistema propuesto, en el cual se presenta el plan de pruebas y los casos de prueba diseñados para probar la aplicación.

### 4.1 Diagramas de Clases

#### 4.1.1 Subsistema Gestión de Información.

**DCD Gestionar Documentos de Proyecto.**

## Capítulo 4: Construcción de la solución propuesta

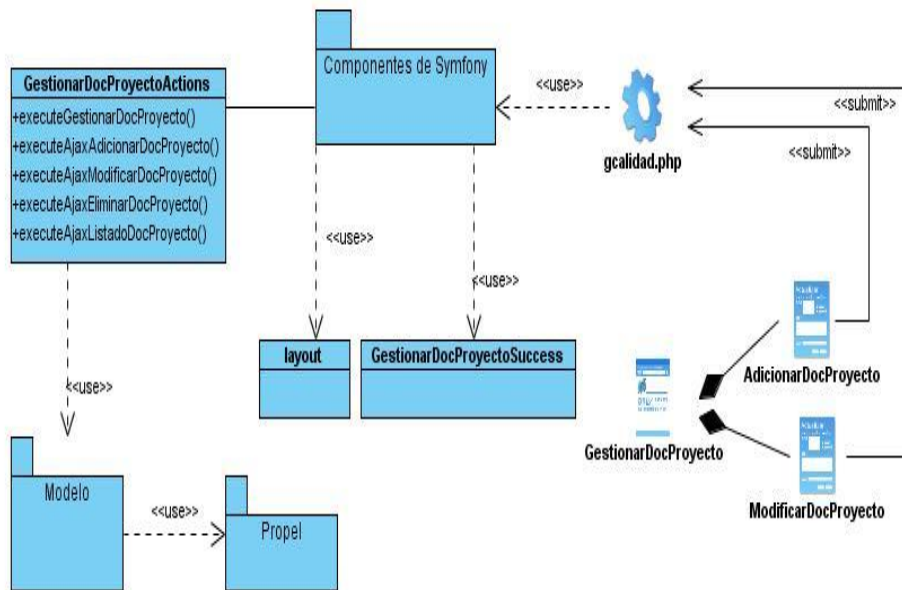


Figura 1. DCD Gestionar Documentos de Proyecto.

### 4.1.2 Subsistema Gestión de Calidad.

#### DCD Gestionar No Conformidades.

# Capítulo 4: Construcción de la solución propuesta

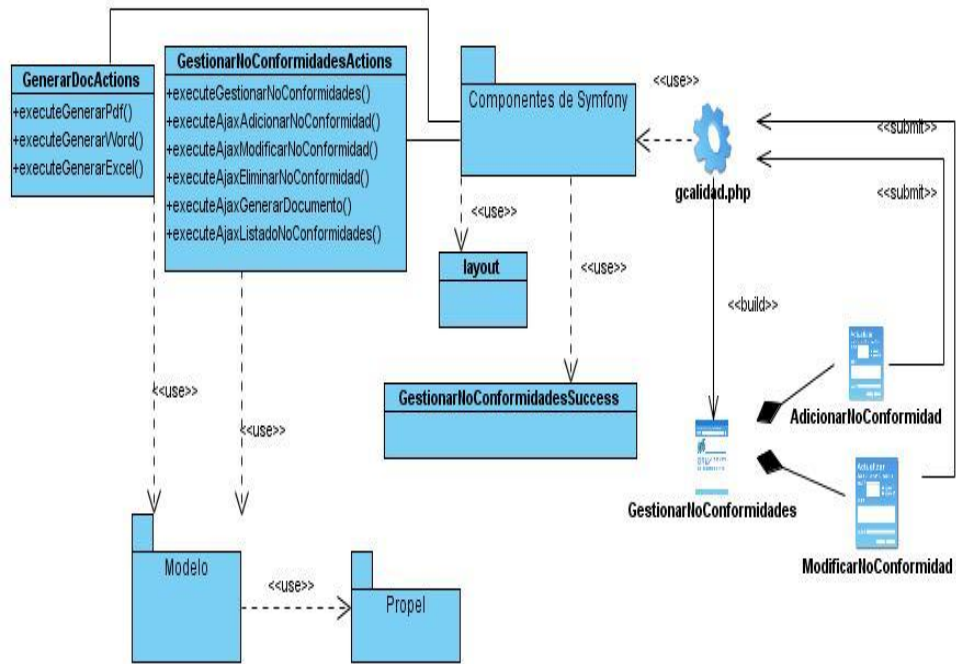


Figura 2. DCD Gestionar No Conformidades.

## DCD Gestionar Listas de Chequeo.

# Capítulo 4: Construcción de la solución propuesta

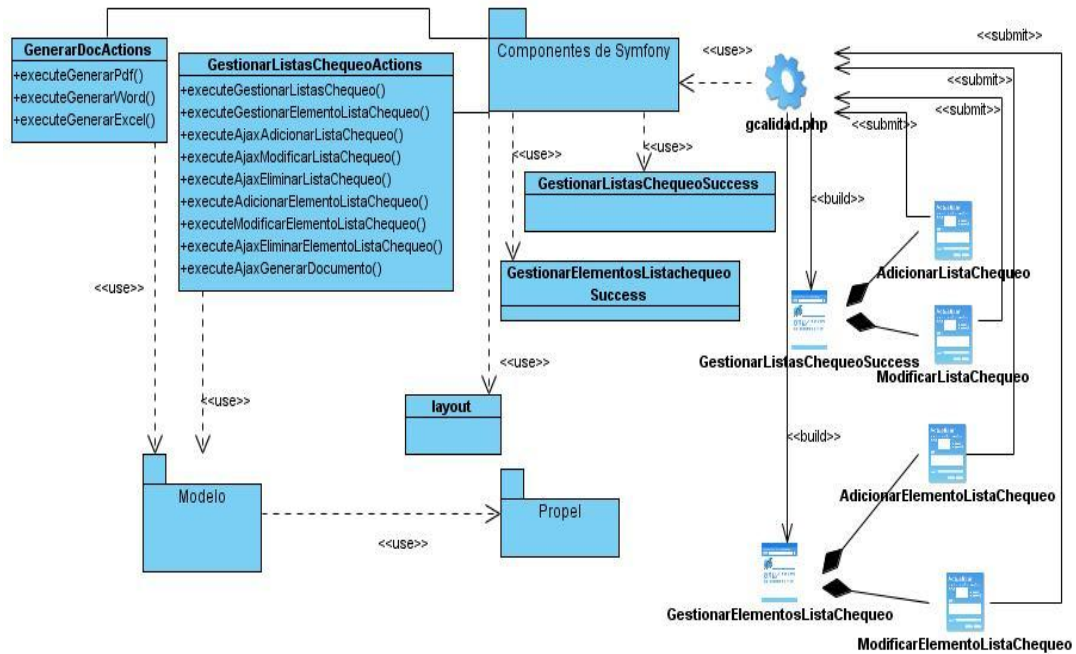


Figura 3. DCD Gestionar Listas de Chequeo.

## DCD Gestionar Tareas de Corrección.

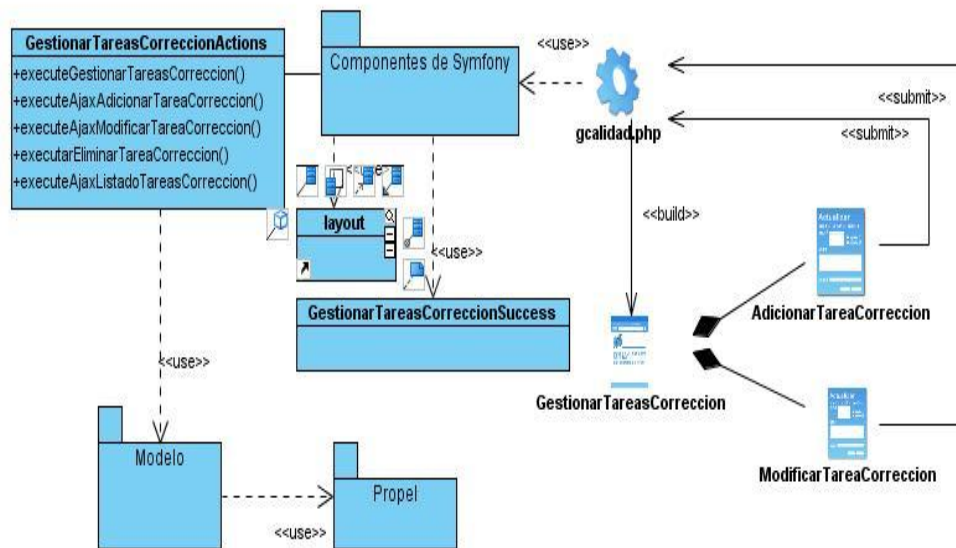


Figura 4. DCD Gestionar Seguimiento de Tareas.

## Capítulo 4: Construcción de la solución propuesta

### 4.2 Principios de diseño

Definir principios a la hora de diseñar una aplicación permite que la misma se convierta en una herramienta atractiva para el usuario, que no sólo tiene incidencia sobre la interfaz gráfica de la aplicación, sino también sobre la forma en que se publica la información necesaria para el usuario, para lo cual es fundamental un contenido bien distribuido. Una interfaz bien diseñada trae consigo un buen diseño gráfico, una buena navegabilidad, así como una adecuada usabilidad.

Con el objetivo de realizar un adecuado diseño de la interfaz de la aplicación Web, el sistema debe estar compuesto por una plantilla que contenga un menú lateral izquierdo, separando por bloques las principales funcionalidades que el sistema brinda, y a la derecha de este menú, se encuentra el área para los formularios que permiten la interacción con el usuario. Para el diseño de la interfaz gráfica de la aplicación se utilizan los colores identificativos de la Universidad de las Ciencias Informáticas (azul oscuro, azul claro y blanco).

#### 4.2.1 Estándares de la interfaz de la aplicación

Para lograr un adecuado diseño de la aplicación Web se proponen una serie de estándares a tener en cuenta para garantizar la consistencia del diseño. Esos estándares son los siguientes:

- Los usuarios sólo pueden visualizar el contenido estrictamente necesario de acuerdo al rol que desempeñan dentro de la aplicación.
- Elaborar una interfaz sencilla que permita su fácil utilización por parte de una persona con conocimientos mínimos de computación.
- Garantizar que los contrastes de colores no afecten la legibilidad del texto y que la fuente brinde una fácil lectura.
- Mostrar al usuario un mensaje de advertencia cada vez que vaya a realizar una acción que cambie los datos del sistema para evitar errores indeseados.
- Garantizar que el uso de servicios externos a la aplicación no afecten la interfaz gráfica en caso de presentar fallos o estar fuera de servicio.

## Capítulo 4: Construcción de la solución propuesta

### 4.2.2 Patrón arquitectónico Modelo Vista Controlador (MVC).

El patrón MVC, es un patrón clásico del diseño web, conocido como arquitectura MVC, que está formado por tres niveles:

- El Modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La Vista transforma el modelo en una página web que permite al usuario interactuar con ella.
- El Controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. Si por ejemplo una misma aplicación debe ejecutarse tanto en un navegador estándar como un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, correo electrónico). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación.



## Capítulo 4: Construcción de la solución propuesta

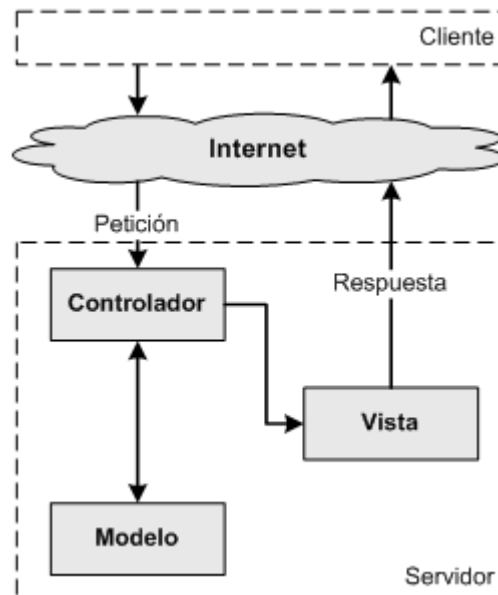


Figura 5. El patrón MVC

(Fabien Potencier, 2008)

### 4.2.3 Concepción general de la ayuda

Se realiza un Manual de Usuario en formato PDF que está adjunto al sistema donde se incluye una explicación detallada de cómo acceder y trabajar con el mismo, las restricciones para los diferentes roles de usuario y cada una de las funcionalidades a las que tienen acceso. Las principales secciones del manual son:

- Descripción general del sistema.
- Roles de Usuarios.
- Navegabilidad del sitio.
- Formularios de Gestión.
- Posibles errores.

### 4.2.4 Seguridad del Sistema

## Capítulo 4: Construcción de la solución propuesta

La autenticación es un mecanismo que aplican muchos sistemas, de software o no, para identificar a sus usuarios. Se trata de probar que cada usuario es quien dice ser. En la actualidad son muchas las tecnologías que se utilizan para los procesos de autenticación. En el caso del software sigue siendo el nombre de usuario y la contraseña la vía fundamental.

Si el nombre de usuario y la contraseña cifrada coinciden con el par almacenado en la base de datos de usuarios, entonces el interesado podrá utilizar las facilidades del sistema que son permitidas de acuerdo a su identidad y privilegios.

La UCI, se rige por las políticas de seguridad contenidas en el Plan de Seguridad Informática de la UCI, las cuales deben conocer todos los usuarios de las tecnologías informáticas y de comunicaciones de la Universidad.

En el sistema se define como política de seguridad para la cuenta de administración, que la contraseña tendrá una longitud de 7 caracteres como mínimo, y debe cumplir con varios requerimientos de complejidad como empleo de números, caracteres especiales, letras mayúsculas y minúsculas, que hacen que una contraseña sea fuerte. Además de que el tiempo de caducidad tiene que ser de dos meses, con período de tres días para cambiar de una contraseña a otra, y no puede coincidir con las 24 contraseñas anteriores.

El método de encriptación que se utiliza para almacenar la contraseña es SHA -1(Secure Hash Algorithm, Algoritmo de Hash Seguro), fue desarrollado por la NSA (Agencia de Seguridad Nacional), para ser incluido en el estándar DSS (Digital Signature Standard). SHA-1 se considera seguro y libre de puertas traseras, ya que el hecho de que el algoritmo sea realmente seguro favorece a los propios intereses de la NSA. Produce firmas de 160 bits, a partir de bloques de 512 bits del mensaje original.

### 4.3 Diseño de la Base de Datos

**Diagrama Entidad Relación:**

# Capítulo 4: Construcción de la solución propuesta

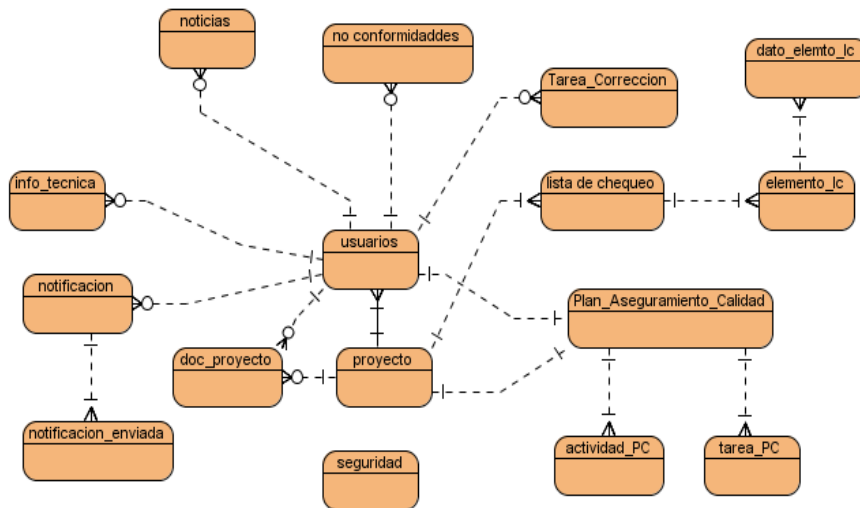


Figura 6. Diagrama Entidad Relación.

## Diagrama de Clases Persistente:

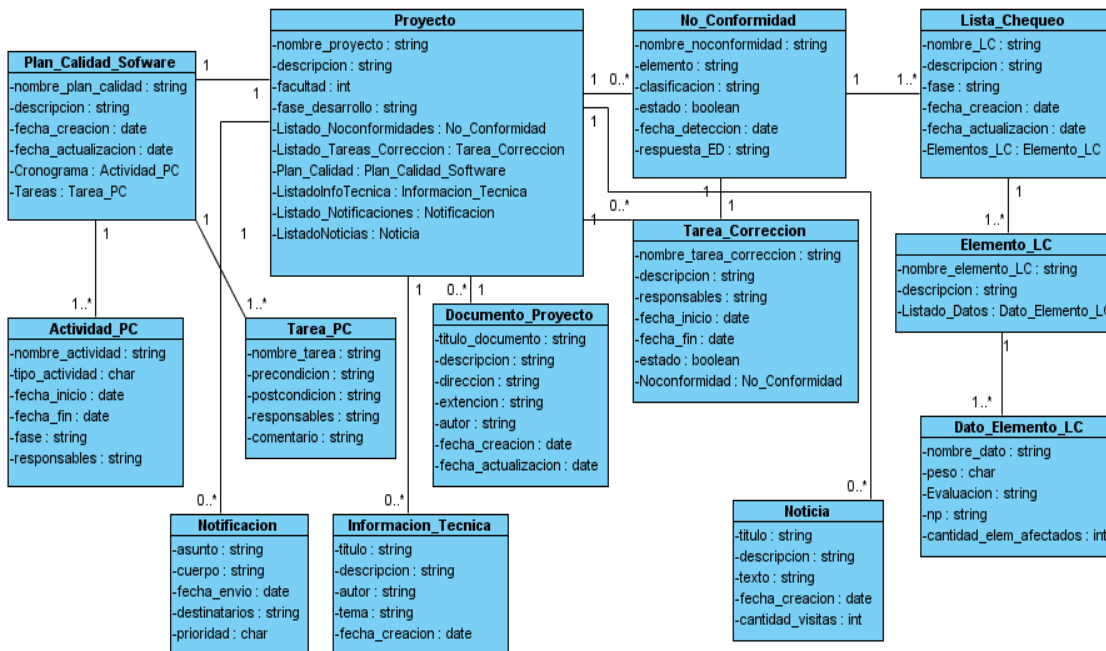


Figura 7. Diagrama de Clases Persistente

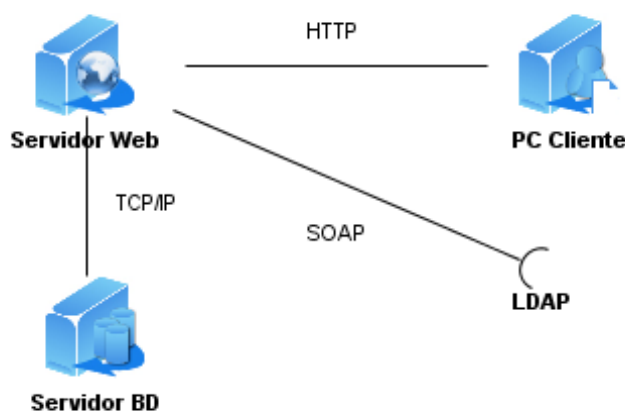
## Capítulo 4: Construcción de la solución propuesta

Para ver el Diagrama de las Tablas de la Base de Datos consultar Anexo 3.

### 4.4 Modelo de Despliegue

El Modelo de Despliegue agrupa los objetos que describen la distribución física del Sistema en términos de funcionalidad entre los nodos. Cada nodo representa un recurso que interviene en el despliegue del Sistema. Describe la arquitectura física del Sistema durante la ejecución en términos de procesadores, dispositivos y componentes de software. Además, describen la topología del Sistema, es decir, la estructura de los elementos de hardware y software que ejecuta cada uno de ellos.

**Diagrama de Despliegue:**



### 4.5 Modelo de Implementación

El Modelo de Implementación describe cómo los elementos del modelo de diseño y las clases, se implementan en términos de componentes, ficheros de código fuente, ejecutables. Los Diagramas de despliegue y componentes conforman lo que se conoce como un Modelo de Implementación al describir los componentes y construir su organización y dependencia entre los nodos físicos con los que funcionará la aplicación.

Un Diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean éstos componentes de código fuente, binarios o ejecutables. Se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, las

# Capítulo 4: Construcción de la solución propuesta

restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. Los elementos de modelado dentro de un Diagrama de componentes serán componentes y paquetes.

## Diagrama de Componentes:

### Caso de Uso Gestionar Documentos de Proyecto:

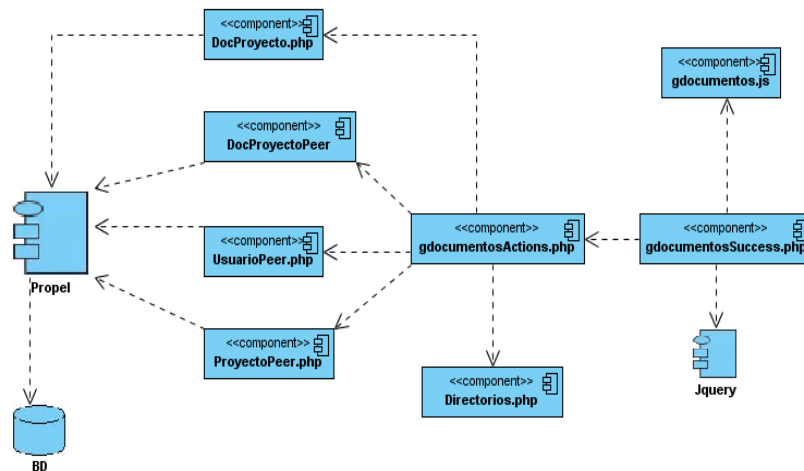


Figura 8. DC Gestionar Documentos de Proyecto.

### Caso de Uso Gestionar No Conformidades:

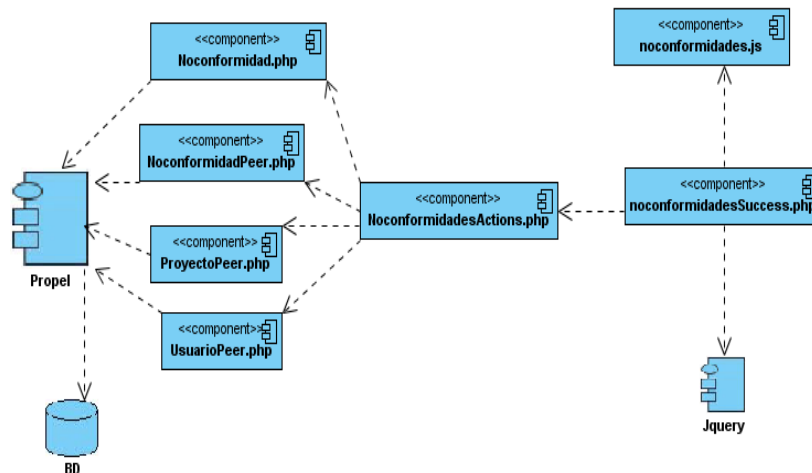


Figura 9. DC Gestionar No Conformidades.

# Capítulo 4: Construcción de la solución propuesta

## Caso de Uso Gestionar Listas de Chequeo:

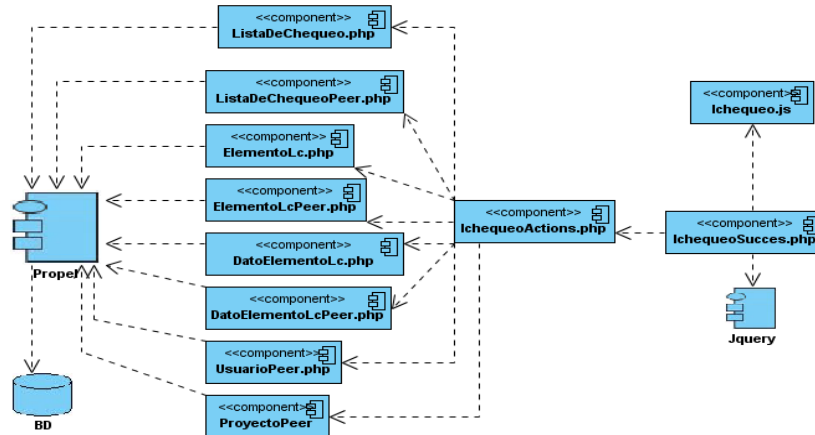


Figura 10. DC Gestionar Listas de Chequeo.

## Caso de Uso Gestionar Tareas de Corrección:

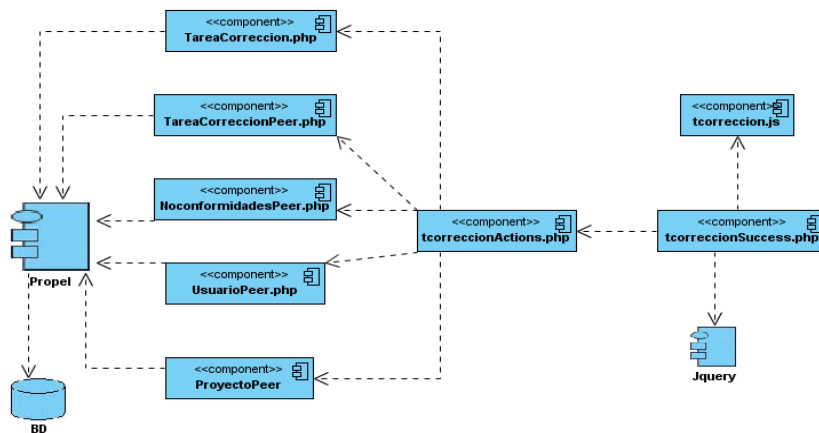


Figura 11. DC Gestionar Tareas de Corrección.

Para ver el resto de los Diagramas de Componentes Consultar Anexo 4.

## 4.6 Prueba del sistema propuesto

Las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados, y una evaluación es hecha de

## Capítulo 4: Construcción de la solución propuesta

algún aspecto del sistema o componente. La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación.

Entre los varios métodos de pruebas que existen, se encuentran las conocidas como **Pruebas de Caja Negra**, la cual se centra principalmente en los requisitos funcionales del software. Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos. Este tipo de pruebas permiten encontrar funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos o en accesos a las Bases de Datos externas, errores de rendimiento, errores de inicialización y terminación.

Para preparar los casos de pruebas hacen falta un número de datos que ayuden a la ejecución de estos casos y que permitan que el sistema se ejecute en todas sus variantes, pueden ser datos válidos o inválidos para el programa, si lo que se desea es hallar un error o probar una funcionalidad. Los datos se escogen atendiendo a las especificaciones del problema, sin importar los detalles internos del programa, a fin de verificar que el programa corra bien.

### 4.7 Conclusiones

En el capítulo se mostraron los resultados obtenidos durante el flujo de trabajo de implementación. Se realizó el diagrama de componentes con el objetivo de brindar una idea de cómo se implementó el software en término de componentes, dando una visión de cómo quedó construida y distribuida la aplicación. Mediante los casos de prueba se validó el software, verificando el correcto funcionamiento del sistema, comprobándose que no existían errores en las funciones operativas.

## Conclusiones

Con la finalización del presente trabajo, se cumplió con el objetivo general trazado y con las tareas a cumplir, al inicio de la investigación. Se realizó un estudio profundo del estado del arte, de las herramientas existentes en el mundo del software, para la gestión de la calidad, seleccionándose aquellas que podrían utilizarse como referencia de la solución. La evaluación de varios lenguajes para la programación permitió seleccionar el más adecuado de acuerdo a los objetivos y requerimientos de la aplicación.

Se cumplió con el objetivo general de desarrollar un Sistema informático para la Gestión de la Calidad en el Departamento de Geoinformática, el cual fue programado usando PHP como lenguaje para la implementación, Visual Paradigm como herramienta CASE, y Zend Studio for Eclipse como Entorno Integrado de Desarrollo(IDE).

Se concluye que el sistema brinda una solución factible a la problemática inicial, así como las siguientes ventajas:

- Control exhaustivo de documentos, cronogramas y registros.
- Actualización fácil y segura de la documentación.
- Acceso fácil a cualquier documento o registro.
- Trazabilidad y seguimiento a las no conformidades y defectos detectados.
- Ahorro de tiempo.
- Aseguramiento de los conocimientos para la capacitación del equipo de proyecto en aspectos de la gestión de la calidad.



### Referencias Bibliográficas

Como resultado del proceso de investigación y realización de la aplicación, que se utiliza para mejorar los procesos de gestión de la calidad, han surgido ideas que serían recomendables tener en cuenta para un futuro perfeccionamiento del sistema, a continuación se listan las mismas:

- Agregar nuevas funcionalidades al sistema que le permita ser más interactivo, de acuerdo con las expectativas y las necesidades existentes dentro del Departamento de Geoinformática.
- Integración con sistemas de correo electrónico y con un repositorio subversión.

## Referencias Bibliográficas

BSI. 2009. **BSI, The British Standards Institution. *BSI, The British Standards Institution.*** [En línea] 5 de 10 de 2009. [Citado el: 5 de 12 de 2009.] <http://www.bsigroup.com.mx/es-mx/Auditoria-y-Certificacion/Sistemas-de-Gestion/De-un-vistazo/Que-son-los-sistemas-de-gestion/>.

Buscarportal. 2009. **Buscarportal. *Buscarportal.*** [En línea] 15 de 9 de 2009. [Citado el: 5 de 12 de 2009.] [http://www.buscarportal.com/articulos/iso\\_9001\\_2000\\_gestion\\_calidad.html](http://www.buscarportal.com/articulos/iso_9001_2000_gestion_calidad.html).

Definicion. 2009. **Definicion.de. *Definicion.de.*** [En línea] 10 de 9 de 2009. [Citado el: 5 de 12 de 2009.] <http://definicion.de/gestion/>.

Desarrollo Web.com. 2005. **Desarrollo Web.com. *Desarrollo Web.com.*** [En línea] 10 de 5 de 2005. [Citado el: 5 de 12 de 2009.] <http://www.desarrolloweb.com/articulos/392.php>.

—. 2008. **Desarrollo Web.com. *Desarrollo Web.com.*** [En línea] 9 de 10 de 2008. [Citado el: 5 de 12 de 2009.] <http://www.desarrolloweb.com/articulos/497.php>.

Desarrollo Wev.com. 2009. **Desarrollo Wev.com. *Desarrollo Wev.com.*** [En línea] 20 de 11 de 2009. [Citado el: 5 de 12 de 2009.] <http://www.desarrolloweb.com/articulos/1178.php>.

dipler.org. 2009. **dipler.org. *dipler.org.*** [En línea] 20 de 11 de 2009. [Citado el: 5 de 12 de 2009.] <http://www.dipler.org/2008/10/que-es-appserv/>.

Fabien Potencier, François Zaninotto. 2008. ***Symfony, la guía definitiva.*** 2008.

Grupo Soluciones Innova. 2007. **Grupo Soluciones Innova. *Grupo Soluciones Innova.*** [En línea] 5 de 7 de 2007. [Citado el: 5 de 12 de 2009.] <http://www.rational.com.ar/herramientas/rup.html>.

## Referencias Bibliográficas

Hirsch, Michael. 2008. **Agilealliance. Agilealliance.** [En línea] 10 de 10 de 2008. [Citado el: 5 de 12 de 2009.] <http://www.agilealliance.org/system/article/file/1135/file.pdf>.

IBM. 2009. **IBM. IBM.** [En línea] 5 de 11 de 2009. [Citado el: 5 de 12 de 2009.] [ftp://ftp.software.ibm.com/software/rational/web/datasheets/rose\\_ds.pdf](ftp://ftp.software.ibm.com/software/rational/web/datasheets/rose_ds.pdf).

Kioskea.net. 2008. **Kioskea.net. Kioskea.net.** [En línea] 16 de 10 de 2008. [Citado el: 5 de 12 de 2009.] <http://es.kioskea.net/contents/qualite/qualite-introduction.php3>.

KMKey. 2008. **KMKey knowledgee manager. KMKey knowledgee manager.** [En línea] 20 de 6 de 2008. [Citado el: 5 de 12 de 2009.] [http://www.kmkey.com/productos/software\\_gestion\\_calidad](http://www.kmkey.com/productos/software_gestion_calidad).

kumbiaphp. 2009. **kumbiaphp. kumbiaphp.** [En línea] 1 de 12 de 2009. [Citado el: 5 de 12 de 2009.] <http://www.kumbiaphp.com/blog/about/>.

Librosweb. 2009. **Librosweb. Librosweb.** [En línea] 10 de 8 de 2009. [Citado el: 5 de 12 de 2009.] <http://librosweb.es/ajax/capitulo1.html>.

lycos. 2009. **lycos. lycos.** [En línea] 10 de 5 de 2009. [Citado el: 5 de 12 de 2009.] <http://usuarios.lycos.es/oopere/uml.htm>.

netpecos. 2007. **netpecos. netpecos.** [En línea] 20 de 3 de 2007. [Citado el: 5 de 12 de 2009.] [http://www.netpecos.org/docs/mysql\\_postgres/x15.html](http://www.netpecos.org/docs/mysql_postgres/x15.html).

—. 2007. **netpecos. netpecos.** [En línea] 20 de 3 de 2007. [Citado el: 5 de 12 de 2009.] [http://www.netpecos.org/docs/mysql\\_postgres/x57.html](http://www.netpecos.org/docs/mysql_postgres/x57.html).

Simandoc. 2008. **Simandoc, Sistema de Gestion de la Calidad y Manejo de Documentos. Simandoc, Sistema de Gestion de la Calidad y Manejo de Documentos.** [En línea] 9 de 12 de 2008. [Citado el: 5 de 12 de 2009.] <http://www.simandoc.com.ar/>.

## Referencias Bibliográficas

Sourceforge. 2004. **Sourceforge. Sourceforge.** [En línea] 29 de 9 de 2004. [Citado el: 5 de 12 de 2009.] <http://oness.sourceforge.net/proyecto/html/ch05.html>.

Universidad de Belgrano. 2006. **Universidad de Belgrano. Universidad de Belgrano.** [En línea] 9 de 3 de 2006. [Citado el: 5 de 12 de 2009.] [http://www.ub.edu.ar/catedras/ingenieria/ing\\_software/ubftecwwwdfd/calidadsw/calidad.htm](http://www.ub.edu.ar/catedras/ingenieria/ing_software/ubftecwwwdfd/calidadsw/calidad.htm).

Visual Paradigm. 2009. **Visual Paradigm. Visual Paradigm.** [En línea] 20 de 10 de 2009. [Citado el: 5 de 12 de 2009.] <http://www.visual-paradigm.com/product/vpuml/editions/standard.jsp>.

World Workers Argentina. 2009. **World Workers Argentina. World Workers Argentina.** [En línea] 28 de 11 de 2009. [Citado el: 5 de 12 de 2009.] [http://www.worldworkers.com.ar/paq\\_alianzas\\_estrategicas.htm](http://www.worldworkers.com.ar/paq_alianzas_estrategicas.htm).

## Bibliografía

ISO 9000 Sistemas de gestión de la calidad — Conceptos y vocabulario. (Traducción Certificada). Ginebra, Suiza, 2000.

IEEE Std 730-1998: Standard for Software Quality Assurance Plans. New York: IEEE, 1998.

Simandoc. Sistema de Gestión de la Calidad y Manejo de Documentos.

<http://www.simandoc.com.ar/Funcionalidad.htm#>

KEMP, S. and PMP. QUALITY MANAGEMENT DEMYSTIFIED, McGRAW HILL, 2006.

PRESSMAN, R. S. Ingeniería de Software un enfoque práctico. 6. Madrid, Mc Graw-Hill, 2005.

KMKey. Knowledge Manager Key.

<http://www.kmkey.com/descripcion-kmkey>

SCHULMEYER, G. G. Handbook of Software Quality Assurance Fourth Edition, Artech house, 2008.

Fabien Potencier, François Zaninotto. 2008. *Symfony, la guía definitiva*. 2008.

TQC STI. Total Quality Control.

<http://www.xtrategus.com.ar/Soluciones/Overview/tabid/167/Default.aspx>

<https://seguridad.uci.cu/documentacion/Contraseñas/Compañera%20inseparable,%20la%20contraseña.pdf>

<https://seguridad.uci.cu/documentacion/Contraseñas/Compañera%20inseparable,%20la%20contraseña.pdf>