

Universidad de las Ciencias Informáticas

Facultad 7



## Título: Sistema para la gestión de revisiones

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

Autores:

Yailin Vizcaino Amador

Ramsés Ramos Miranda

Tutores:

Ing. Anet Conde Almeida

Ing. Yislenys Suarez Hernández

Ciudad de la Habana, junio 2010

“Año 52 de la Revolución”

# Declaración de Autoría

## Declaración de Autoría

Declaramos ser autores de la presente tesis y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio.

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

\_\_\_\_\_

\_\_\_\_\_

Firma del Autor

Firma del Autor

Yailín Vizcaíno Amador

Ramsés Ramos Miranda

\_\_\_\_\_

\_\_\_\_\_

Firma del Tutor

Firma del Tutor

Ing: Anet Conde Almeida

Ing: Yislenys Suárez Hernández

## Resumen

Las revisiones son procesos que se realizan encaminadas a garantizar la calidad del Proceso de Desarrollo de Software. El presente trabajo surge por la necesidad de crear un sistema que facilite la gestión del proceso de revisiones que se lleva a cabo en el Grupo de Calidad del Centro de Informática Médica (CESIM).

Para materializar la propuesta planteada se trazó como objetivo, desarrollar un sistema que permita informatizar los procesos de revisiones realizados en el Grupo de Calidad del CESIM mediante el uso de Python como lenguaje de programación, Django como framework y Wing IDE como entorno de desarrollo integrado. Se usará PostgreSQL como gestor de Base de Datos y UML como lenguaje de modelado para cada disciplina propuesta por la metodología RUP con la cual se llevará a cabo el desarrollo de la herramienta.

El Sistema para la gestión de revisiones (SIGRE) brinda una gama de beneficios tanto para los revisores como para los usuarios que serán sometidos a este proceso. Su uso permite conocer la fecha de realización de las revisiones a los distintos proyectos pertenecientes al centro, el estado actual de las deficiencias detectadas, los plazos de corrección de las mismas, así como las listas de chequeo utilizadas en cada caso. Además la informatización de este proceso permite realizar de forma sencilla y rápida acciones de modificación, inserción y eliminación de datos generados como resultado de las revisiones realizadas.

**Contenido**

Declaración de Autoría ..... II

Resumen ..... III

Introducción ..... 1

**Capítulo 1 Fundamentación Teórica ..... 5**

**Introducción..... 5**

**1.1 Conceptos Fundamentales ..... 5**

        1.1.1 Calidad..... 5

        1.1.2 Aseguramiento de la calidad ..... 6

        1.1.3 Revisiones ..... 6

**1.2 Otros conceptos ..... 8**

        1.2.1 Proyecto ..... 8

        1.2.2 Expediente de Proyecto..... 9

        1.2.3 Gestión de la Información ..... 9

        1.2.4 Entrevista ..... 9

        1.2.5 Lineamientos de Calidad..... 10

        1.2.6 Lista de Chequeo ..... 10

**1.3 Herramientas actuales para la planificación y el desarrollo de revisiones..... 11**

        1.3.1 Gantt PV ..... 11

        1.3.2 Dotproject ..... 11

        1.3.3 Agile Track ..... 12

        1.3.4 Project Server ..... 13

        1.3.5 RedMine ..... 14

        1.3.6 MGRSoft ..... 15

**1.4 Tecnologías, Herramientas y Metodologías ..... 15**

        1.4.1 Lenguaje de Programación, Python ..... 16

1

# Tabla de Contenidos

1.4.2 Framework, Django Framework .....	18
1.4.3 Entorno de desarrollo, Wing IDE.....	19
1.4.4 Metodología de Desarrollo, RUP .....	19
1.4.5 Notación de Modelado, BPMN .....	20
1.4.6 Lengua de Modelado, UML.....	20
1.4.7 Herramienta de Modelado, Visual Paradigm.....	21
1.4.8 Gestor de Base de Datos, PostgreSQL.....	21
1.4.9 Tipos de Aplicaciones .....	22
1.4.9.1 Aplicaciones web .....	22
1.4.9.2 Aplicaciones de Escritorio .....	23
1.4.9.3 Ventajas de las Aplicaciones web sobre las de Escritorio.....	23
Capítulo 2. Características del Sistema .....	25
2.1 Descripción General del Objeto de Estudio .....	25
2.2 Situación Problemática.....	26
2.3 Descripción del flujo de trabajo en la realización de un proceso de revisión.....	27
2.4 Procesos del Negocio.....	27
2.5 Descripción de los procesos del negocio .....	27
2.6 Objeto de Informatización.....	28
2.7 Propuesta de Sistema.....	28
2.7.1 Descripción General de la Propuesta del Sistema.....	28
2.8 Modelo de Negocio .....	28
2.8.1 Identificación de los roles del entorno del Negocio .....	29
2.8.2 Diagramas de procesos del negocio.....	31
2.8.3 Descripción de los procesos del negocio.....	34
2.9 Requisitos Funcionales .....	43
2.9.1 Requisitos Funcionales .....	43
2.10 Requisitos no Funcionales .....	45
2.11 Modelo de Casos de Usos del Sistema.....	47

# Tabla de Contenidos

2.11.1 Actores del Sistema .....	47
2.12 Modelo de Casos de Usos del Sistema .....	48
Capítulo 3. Análisis y Diseño del Sistema .....	50
3.1 Descripción de la Arquitectura .....	50
3.2 Patrones .....	51
3.3 Modelo de Análisis .....	52
3.3.1 Diagrama de Clases del Análisis .....	54
3.4 Diagramas de Clases del Diseño .....	54
3.4.1 Diagramas de Interacción en el Diseño .....	55
Capítulo 4. Implementación .....	57
4.1 Modelo de Datos .....	57
4.1.1 Descripción de las Tablas .....	58
4.2 Implementación .....	58
4.2.1 Modelo de Implementación .....	58
4.2.2 Diagrama de Despliegue .....	58
4.2.3 Diagrama de Componentes .....	59
4.3 Seguridad .....	61
Conclusiones .....	62
Recomendaciones .....	63
Referencias Bibliográficas .....	64
Bibliografía Consultada .....	67
Glosario .....	69

## Introducción

La Universidad de las Ciencias Informáticas (UCI), surgida al calor de la Batalla de Ideas en el año 2002, fue concebida desde sus inicios como una universidad de excelencia; sus estudiantes estarían vinculados al proceso docente y productivo mediante programas de estudios flexibles y acorde a las necesidades del centro. En pocos años, este centro se ha insertado en el campo económico cubano, convirtiéndose en una de las empresas del software más importantes que posee el país actualmente. La universidad se encuentra avalada por la calidad de los productos que la misma genera y el grado de aceptación que posee tanto a nivel nacional como internacional.

Para lograr un alto nivel de aceptación en el mercado se hace indispensable, como premisa fundamental, comprobar que los productos elaborados en la UCI cumplan con las normas de calidad establecidas para tales fines. Es por ello que el aseguramiento de la calidad es un factor determinante en el proceso de desarrollo de software, como elemento rector en la organización, dirección y control de las actividades productivas, que permitan brindar productos cuyos servicios cumplan satisfactoriamente las expectativas del cliente.

Las revisiones como procesos sistemáticos se ejecutan para lograr la fortaleza durante el desarrollo del software a lo largo de todas las fases por las que transita a lo largo del ciclo de desarrollo del software según la metodología seleccionada. Para darle cumplimiento, se realizan acciones para corregir los defectos que se detectan en las entidades en la cual se ejecutan, por lo que son fundamentales para obtener seguridad, eficiencia y calidad en los productos generados.

El proceso de revisiones se ha realizado de forma histórica por personal calificado para asumir esta tarea, los cuales son conocidos comúnmente como revisores. Estos se encargan de forma personal de informar con antelación las fechas en que se realizarán las revisiones a los departamentos o empresas mediante la comunicación verbal, reunión personal con las partes involucradas, mediante el envío de una carta o cita informativa.

El Grupo de Calidad perteneciente al CESIM, posee varias tareas para asegurar la calidad de todos los productos que se generan como parte del proceso productivo. El mismo tiene creado un Grupo de Auditoría y Revisiones el cual se encarga de revisar los proyectos productivos del CESIM. La gestión de la información referente a los cronogramas de ejecución de las revisiones, así como de los elementos a revisar, se tramitan mediante el uso de correo

## Introducción

electrónico, medios de mensajería instantánea (jabber) o de forma personal entre las partes involucradas en el proceso, por lo que la información en ocasiones no fluye coherentemente provocando atrasos en los cronogramas y las actividades a desarrollar.

Las revisiones son ejecutadas por un grupo de revisores el cual está formado por un revisor líder y dos miembros del grupo. El trabajo es realizado por cada uno de los revisores sobre cada uno de los elementos a revisar, mediante la aplicación de Listas de Chequeos o el uso del Registro de Dificultades y Acciones Correctivas (RDAC), en dependencia del tipo de revisión que se realiza, los involucrados detectan las dificultades del proyecto. Como resultado de este proceso se elabora un solo documento final denominado Registro de Dificultades y Acciones Correctivas donde son expuestas cada una de las deficiencias encontradas con sus respectivos plazos para ser corregidas, por lo que es engorrosa la integración de la información recopilada expuesta a errores, omisiones y duplicaciones.

Toda la documentación generada en cada una de las revisiones es archivada en una computadora la cual tiene destinada un recurso de hardware para dicho fin, con el objetivo de poder darle un posterior uso en futuras revisiones. Dada la magnitud de esta información se hace difícil tener control del lugar exacto donde se tienen registrados documentos específicos a utilizar en un momento dado, por lo que se dificulta la búsqueda de estos, además de estar expuestos al peligro de ser perdidos ante fallas inesperadas de dicha computadora.

Atendiendo a las posibilidades tecnológicas que existen en la universidad y la gran cantidad de información que se genera como resultado del proceso productivo por todas las partes involucradas se formula el siguiente **problema científico** ¿Cómo facilitar la gestión de la información en los procesos de revisiones desarrollados en el Grupo de Calidad del CESIM?

Para dar solución a la problemática planteada se define como **objeto de estudio** el proceso de aseguramiento de la calidad en el Grupo de Calidad del CESIM, delimitando como **campo de acción** el proceso de de revisiones que se realiza en el Grupo de Calidad del CESIM.

Para dar cumplimiento al problema planteado se define como **objetivo general** desarrollar un sistema que permita informatizar la gestión de los procesos de revisiones en el Grupo de Calidad del CESIM.

Para darle cumplimiento al objetivo propuesto se definieron las siguientes **tareas de investigación**:

# Introducción

- Analizar los diferentes sistemas que existen para la gestión de revisiones.
- Evaluar las tecnologías y librerías existentes que puedan ser usadas en el trabajo.
- Aplicar la metodología, lenguaje de programación, plataforma y entorno integrado de desarrollo (IDE) propuesto para elaborar la herramienta.
- Realizar un análisis sobre las revisiones que se realizan en el área.
- Elaborar la documentación correspondiente a los Flujos de Trabajo propuestos por la metodología definida a usar.
- Implementar las funcionalidades requeridas para llevar a cabo el desarrollo de la aplicación.
- Desarrollar pruebas de caja negra al sistema obtenido.

Entre los **principales aportes** que brindará el sistema una vez que se encuentre a disposición de los usuarios pueden ser citados los siguientes:

- Planificar y consultar de forma automatizada las fechas de realización de las revisiones.
- Controlar y verificar de forma permanente cada uno de los resultados arrojados en las revisiones realizadas.
- Consultar y llenar las Listas de Chequeo necesarias en cada uno de los productos o documentos revisados.
- Introducir datos y estados actuales del proyecto.
- Emitir reportes.
- Introducir, almacenar y consultar las dificultades detectadas en las revisiones y las acciones correctivas a desarrollar así como el estado de cumplimiento de las mismas.
- Seleccionar la documentación para consultar y evaluar los resultados de las revisiones realizadas.
- Informar de forma detallada el tipo de revisión a la cual pertenece la documentación seleccionada.
- Actualizar las diferentes fases en que se encuentra el desarrollo de los proyectos productivos asociados a los diferentes departamentos que atiende el Grupo de Calidad del CESIM.
- Actualizar las fases, roles y responsabilidades en el proceso de desarrollo del software atendiendo a los procesos de mejoras.

# Introducción

El presente trabajo de diploma se encuentra estructurado por cuatro capítulos, distribuidos de la siguiente forma:

**Capítulo1.** Fundamentación Teórica: Este capítulo recoge toda la fundamentación teórica que sustenta el desarrollo de la investigación, para el posterior desarrollo del sistema propuesto. En el mismo se hace un estudio del estado del arte de sistemas existentes con características similares en cuanto a objetivos de uso y de funcionamiento. Se hace además referencia a las tendencias actuales así como a las tecnologías y las metodologías que se aplican durante el proceso de investigación y desarrollo del sistema.

**Capítulo2.** Características del Sistema: Se hace énfasis en las características de los procesos que serán desarrollados. Se hace un análisis de cada uno de los flujos de trabajo, los modelos necesarios para realizar el desarrollo del sistema y se definen los requisitos funcionales y no funcionales que el mismo tendrá.

**Capítulo3.** Análisis y Diseño del Sistema: Está dedicado al flujo de Análisis y Diseño propuesto por el Proceso Unificado de Rational (*Rational Unified Process*; por sus siglas en inglés RUP), el mismo proporciona las clases del diseño y del sistema así como los diagramas correspondientes a cada una de ellas. Brinda los recursos para la base de datos además de brindar las propuestas de solución necesarias.

**Capítulo4.** Implementación y Pruebas: Aquí se muestra el modelo de implementación que se utiliza, los correspondientes diagramas de despliegue y componentes así como la organización que los mismos poseen. Se especifican los planes de ejecución de pruebas para comprobar los requisitos de las funcionalidades desarrolladas

## Capítulo 1 Fundamentación Teórica

### Introducción

En este capítulo se hace una valoración general del tema que se está abordando. Se realiza una exposición de los conceptos más relevantes relacionados con el tema de las revisiones que son aspectos históricos referentes al mismo, haciéndose un bosquejo de la situación actual en el mundo de la problemática planteada. Se analiza además el ambiente de desarrollo en el que se encuentra para determinar el uso de las herramientas acorde a las tendencias actuales en Cuba y el mundo.

### 1.1 Conceptos Fundamentales

#### 1.1.1 Calidad

La calidad data de cientos de años atrás desde que las primeras culturas buscaban la perfección en el acabado de sus obras de arte, construcciones u otras tareas que demandaran resultados tangibles. La misma cobra fuerza a medida que el proceso productivo desarrollado por el hombre avanzó a lo largo de la historia, teniendo un auge significativo con la Revolución Industrial Inglesa del siglo XVIII. Sin embargo, no es hasta mediados de la década de los 20 del siglo XX que comienza a cobrar importancia significativa avalada por la avalancha tecnológica que ha devenido en el desarrollo de la humanidad desde entonces.

Dado el notable avance y la gran importancia que la misma ha acaparado en los últimos años, organismos internacionales han trazado varias definiciones acorde a sus propias necesidades dentro de sus procesos de desarrollo.

La Real Academia de la Lengua Española define a la calidad como propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar a su valor.

La Organización Internacional de Estándares (ISO por sus siglas en inglés) define la calidad en su norma ISO 8402:1994 como totalidad de propiedades y características de un producto, proceso o servicio que le confiere su aptitud para satisfacer unas necesidades expresadas o implícitas. [1]

## 1.1.2 Aseguramiento de la calidad

Para alcanzar la calidad de un proceso o producto de forma óptima y que cumpla con todos los requisitos que se desea, es indispensable tener una serie de actividades que estén encaminadas a asegurar que la misma sea cumplida de forma adecuada.

Para ello el proceso de aseguramiento de la calidad juega un papel decisivo para obtener este objetivo.

El aseguramiento de la calidad, se puede definir como el esfuerzo total para plantear, organizar, dirigir y controlar la calidad en un sistema de producción con el objetivo de dar al cliente productos con la calidad adecuada. Es simplemente asegurar que la calidad sea lo que debe ser. [2]

La Oficina Internacional de Estandarización (ISO por sus siglas en inglés) en su norma ISO 9000 la define como la actividad que permite proporcionar a los clientes o usuarios la seguridad de que el producto o los servicios tienen la calidad deseada, concertada, pactada o contratada. [3]

Existen varios mecanismos y actividades que manejan el aseguramiento de calidad en las empresas y organizaciones. Las pruebas, auditorías y revisiones son las actividades más comunes empleadas para llevar a término exitoso la calidad en los procesos y actividades que en ellas se desarrollan.

## 1.1.3 Revisiones

Las revisiones según la Real Academia de la Lengua Española se definen como el proceso de someter a algo a un nuevo examen para corregirlo, enmendarlo o repararlo. Este término generalmente es asociado con la corrección de errores. Estas se aplican en distintos momentos del desarrollo del software y de esta forma se detectan errores y defectos que puedan ser eliminados. Se considera que las revisiones son un “filtro” para el proceso de ingeniería de software y sirven para purificar las diferentes actividades que suceden como resultado del análisis, el diseño y la codificación. Los principales objetivos que se buscan con el desarrollo de las revisiones son:

- Evaluar objetivamente los procesos y productos de trabajo del ciclo de vida del software.
- Descubrimiento de defectos e inconsistencias en el software.

## Fundamentación Teórica

- Registrar, comunicar y asegurar la resolución de los defectos detectados en la revisión.

Existen dos tipos fundamentales de revisiones que se realizan en dependencia de la forma en que se desarrollan las mismas. La diferencia principal que existentes entre ambas están enmarcadas en la forma en que se desarrolla la reunión de revisión. Estas son:

- Inspecciones: En este tipo de revisiones los participantes leen el documento, paso a paso, guiados por el autor del mismo, y comprobando en cada uno de los pasos que se realizan el cumplimiento de los criterios de una lista de comprobación.
- Visita guiada: En estas revisiones se demuestra la funcionalidad del objeto revisado mediante la simulación de su funcionamiento con casos de prueba y ejemplos. Se introducen los casos de prueba y se van registrando los resultados obtenidos.

Revisiones Informales:

Este tipo de revisiones en ocasiones son llamadas inadecuadamente sólo con el término de revisiones, por lo cual se crea ambigüedad con respecto a su nombre genérico. Se basan en el intercambio de opiniones entre las personas que participan, que pueden presentar o no una guía de pasos para su realización. Entre sus características principales se encuentran:

- Usualmente no son planeadas.
- No se ha definido un proceso para su realización.
- No existen roles definidos.

Revisiones Formales:

Es más recomendable utilizar este tipo de revisiones ya que las personas que participan en ellas son responsables de la fiabilidad de la evolución y generan un informe en el que reflejan el acto de revisión que llevan a cabo. Con esto se llega a la conclusión de que las revisiones informales se pueden considerar un antepasado poco evolucionado de esta técnica. En las revisiones formales los informes que son generados sirven como hitos del proyecto y de esta forma promueven la preparación por parte de los que participan. Este tipo de revisiones se caracteriza por:

- Tener objetivos definidos y procesos documentados.
- Tener roles definidos.
- Su realización incluye métodos o técnicas para la detección de defectos.
- Se recolectan datos para el control del proceso.
- Todos los participantes son responsables de la calidad de la revisión

## Revisiones Técnicas:

Este tipo de revisiones permiten verificar el trabajo que realizan los desarrolladores, atendiendo a la facilidad con la que pueden ser introducidos los errores y ningún proyecto está exento de ellos. Las revisiones técnicas permiten que sea más factible examinar el trabajo técnico desde otra perspectiva que no sea la de los desarrolladores, ya que algunos errores son más fáciles de detectar por el autor que por otra persona. Entre las revisiones técnicas más comunes se encuentran:

- Revisión de la especificación de requisitos.
- Revisión del diseño.
- Revisión del código.
- Revisión de las pruebas.
- Revisión del manual de usuario.

## Revisiones de Gestión de Proyecto:

Se efectúan con el objetivo de controlar que el proyecto progrese. Se pueden evaluar los riesgos asociados con relación al costo, escala de tiempo, recursos utilizados y la calidad del producto. Para desarrollar las revisiones de gestión de proyecto es necesario que exista un Plan de Desarrollo que esté bien estructurado, que permita evaluar la progresión del proyecto, que los hitos estén bien definidos, que los resultados del proyecto estén bien documentados y que estos hayan sido examinados en una revisión técnica.

## **1.2 Otros conceptos**

### **1.2.1 Proyecto**

Un proyecto es una planificación que consiste en un conjunto de actividades que se encuentran interrelacionadas y coordinadas; la razón de un proyecto es alcanzar objetivos específicos dentro de los límites que imponen un presupuesto y un lapso de tiempo previamente definido.

Un proyecto es un emprendimiento que tiene lugar durante un tiempo limitado, y que apunta a lograr un resultado único. Surge en base a una necesidad, acorde con la visión de la organización, aunque esta puede desviarse en función del interés. El proyecto finaliza cuando se obtiene el resultado deseado, desaparece la necesidad inicial, o se agotan los recursos disponibles.

## 1.2.2 Expediente de Proyecto

El Expediente de Proyecto de la UCI fue realizado con el fin de estandarizar y organizar los artefactos generados durante el proceso de desarrollo de software, generalizar la documentación, facilitar la formación y adiestramiento de proyecto en el uso de modelos y estándares propios y para elevar la calidad del producto entregado al cliente. Debe ser implantado en cada uno de los proyectos vigentes de la universidad.

La documentación asociada a los proyectos de software y sistemas debe cumplir con algunos requisitos como son:

- Servir como medio de comunicación entre los miembros del equipo.
- Servir de repositorio de información que pueda ser utilizado por los ingenieros de sistemas.
- Proveer información para el control de los planes, cronogramas e hitos en el proceso de desarrollo de software.
- Definir quién hace y cómo hace las actividades específicas del desarrollo.

## 1.2.3 Gestión de la Información

Son varias las definiciones que se le atribuyen a este proceso, fundamentado esencialmente por encontrarse presente en las principales esferas de la vida sobre todo en el marco social, político y económico. Históricamente se ha visto muy asociado a la definición de gestión del conocimiento y aunque los mismos tienen similitudes cada uno por separado tiene características particulares que lo diferencian.

La gestión de la información se puede definir como el proceso que se encarga de llevar a término actividades encaminadas a suministrar, controlar y almacenar de forma minuciosa los recursos necesarios para la toma de decisiones para posteriores procesos de mejora en las organizaciones donde es llevado a cabo.

## 1.2.4 Entrevista

La definición de entrevista es amplia y se puede enmarcar según el ámbito en la que es desarrollada. Varias organizaciones hacen sus definiciones acorde a sus propias necesidades, la Real Academia de la Lengua Española la define como “Mantener una conversación con una o varias personas acerca de ciertos extremos, para informar al público de sus respuestas.” [4]

Otros conceptos para la misma la definen como “una conversación planificada para obtener información. Su uso constituye un medio para el conocimiento cualitativo de los fenómenos o sobre características personales del entrevistado y puede influir en determinados aspectos de la conducta humana por lo que es importante una buena comunicación. ” [5]

### 1.2.5 Lineamientos de Calidad

Dentro del proceso de aseguramiento de la calidad se suscriben los Lineamientos de Calidad que constituyen una guía a seguir por los proyectos de desarrollo de software atendiendo a cuáles son los artefactos que deben generar y procesos que deben realizar para aspirar a un producto final con calidad. [6]

Existen algunas normas generales para aplicar los mismos, ellas son:

- El cumplimiento de los lineamientos será chequeado en las auditorías y revisiones que se le practiquen al proyecto.
- El cumplimiento de estos lineamientos es obligatorio para todos los proyectos de desarrollo de software de la UCI.
- En caso de que algunos de los lineamientos no sea aplicable a algún proyecto debido a la naturaleza del mismo, deberá constar en la documentación los motivos de la decisión firmada por el Director General de Producción.
- Para la construcción de los artefactos del proyecto es obligatorio hacer uso del Expediente de Proyecto con las siguientes consideraciones.
  - Si una plantilla para un determinado artefacto no se encuentra presente, el proyecto envía una propuesta al asesor de calidad de su facultad que es el encargado de aprobarla.
  - El formato de las plantillas no es estricto, pueden ser adaptadas al ámbito del proyecto siempre y cuando no se viole el contenido.

### 1.2.6 Lista de Chequeo

Es una lista de preguntas en forma de cuestionario que se utiliza para determinar el grado de cumplimiento de las reglas que son establecidas previamente. Constituye una herramienta de muy fácil utilización para las revisiones y debe estar disponible para la ejecución de sus actividades. Permiten mejorar la planificación y da la oportunidad de hacer una recopilación de los datos que surgen durante este proceso.

## 1.3 Herramientas actuales para la planificación y el desarrollo de revisiones

Varias herramientas actualmente se dedican a la confección de cronogramas y a la planificación de actividades, algunas se han convertido en los últimos años en fuentes muy utilizadas en empresas que planifican sus actividades para lograr sus objetivos en los tiempos pactados. Ellas han ganado gran aceptación en las empresas que trabajan directamente con clientes que demandan de servicios y productos en plazos acordados.

### 1.3.1 Gantt PV

Gantt PV es un programa gratuito, de apariencia sencilla y sin grandes complicaciones, para planificación de proyectos, descomposición, representación y seguimiento de tareas sobre diagrama de Gantt. [7]

#### Ventajas de Gantt PPV

- Gantt PV es un programa Open Source.
- Es de simple manejo.
- Permite la planificación de proyectos, descomposición, representación y seguimiento de tareas sobre diagrama de Gantt.
- Está disponible para los sistemas Operativos, Windows, Linux y Mac OSX.

#### Desventajas de Gantt PV

- Al programa le falta calidad y robustés ya que no siempre corre de manera estable y reporta errores en varios momentos de su corrida.
- La comunidad que lo soporta presenta baja actividad tanto en desarrollo como en reporte y arreglo de bugs, no es muy grande y no tiene mayores fondos para invertir por lo que carece de seguridad el soporte a sus distintas versiones.

### 1.3.2 Dotproject

Algo más veterana esta solución en entorno web ofrece un marco completo para la planificación, gestión y seguimiento de múltiples proyectos para clientes diferentes, quienes pueden disponer también de acceso para monitorizar la evolución del desarrollo. [8]

Permite gestionar las tareas y fases de un proyecto, ofrece para ello una gran cantidad de módulos que permiten realizar la correcta planificación y gestión de las múltiples actividades que surgen.

## Ventajas de DotProject

- El usuario está acostumbrado a un entorno web, esto lo hace fácil de usar.
- Interfaz de usuario simple, clara y consistente.
- Múltiples proyectos para clientes diferentes.
- Posee varios idiomas y soporta distintos niveles de permisos de uso de módulos.

## Desventajas de DotProject

- Las diversas actualizaciones de versiones no garantizan la compatibilidad con las anteriores.
- Los eventos no asocian esfuerzos en el informe de horas asignadas.
- La forma en la que se lleva a cabo la imputación de las actividades no es amigable para el usuario.
- No se incluye ningún tipo de soporte para llevar a cabo la gestión de los riesgos del proyecto.
- DotProject es una herramienta orientada a la Gestión de Proyectos, por lo que se orienta a la administración de recursos para desarrollar un producto, cuya producción requiera de un conjunto de actividades o tareas que se desarrollen entre ellas en forma paralela o independiente.

### **1.3.3 Agile Track**

Herramienta para planificación y seguimiento de proyectos, de interfaz sencilla. Usada en el desarrollo de software en equipos reducidos con metodologías ágiles, especialmente eXtreme Programming. Gestiona ciclos de desarrollo basados en iteraciones, con seguimiento de historias de usuario, tareas y bugs. Multiplataforma para Windows y Linux, consta de dos módulos: servidor que trabaja con MySQL, y el cliente para el seguimiento de los proyectos. Es un desarrollo Open Source, de uso gratuito con licencia AFL (*Academic Free License*). [9]

## Ventajas de Agile Track

- Se centra en simplicidad, utilidad, flexibilidad, y sentido práctico.
- Agile Track permite a todos los equipos utilizar sus propios procesos ágiles y tener éxito.
- Es multiplataforma, puede ser utilizado en Windows ó Linux sin ningún problema.

## Desventajas de Agile Track

- Tiene carácter multiusuario, pero si se quiere añadir más personal se debe adquirir una licencia única por usuario.
- Es usada fundamentalmente en el desarrollo mediante metodologías ágiles.

Pocos son los sistemas que existen en la actualidad para la gestión de la información en los procesos de revisiones, lo que demuestra que es un área que no ha sido explotada con tanta fuerza como otras de la vida cotidiana. En la actualidad, una gran cantidad de proyectos se trazan objetivos para crear sistemas que permitan la gestión de las revisiones de forma automática o semiautomática, mediante el empleo de software que permitan el manejo de esta actividad. Sus principios de funcionamiento están encaminados fundamentalmente en las bases de conocimientos existentes de las áreas donde serán desplegados y puestos en funcionamiento.

Estos sistemas en plena explotación deben permitir de forma general:

- Brindar a los grupos de desarrollo de una vía permanente y segura para obtener cronogramas de ejecución y realización de las revisiones a los módulos en desarrollo.
- Brindar una guía de seguimiento del proceso de corrección a defectos detectados y mantener informados de los pendientes a solucionar.
- Establecer mecanismos que permitan el trabajo simultáneo por parte de equipos de revisores que arrojen un resultado común.

### **1.3.4 Project Server**

Es un programa complementario de Project Professional para la planificación y seguimiento de proyectos, el cual permite la cooperación en línea entre jefes de proyecto, integrantes de grupo, personas y organizaciones interesadas, involucradas de forma activa en el proyecto o sus intereses se ven afectados por el mismo. Permite además a la organización compartir estándares entre proyectos, mejorar la seguridad de éstos, ver la disponibilidad y el resto de la información relacionada con los recursos entre proyectos, administrar y crear informes sobre carteras de proyectos.

Project Server funciona con Project Web Access, la interfaz de usuario basada en web que se utiliza para obtener acceso a la información de Project Server. Ofrece una solución eficaz para la colaboración entre grupos, una gran flexibilidad y diversas ventajas para la comunicación.

Desde Project Web Access, los jefes de equipo pueden conectarse a Project Server para:

## Fundamentación Teórica

- Asignar tareas a integrantes del grupo y realizar un seguimiento del trabajo completado.
- Solicitar y recibir informes de estado.
- Realizar análisis del tipo "¿qué pasaría si?".
- Ver la disponibilidad de los recursos en toda la organización y crear el equipo.

Los integrantes de grupo pueden obtener acceso a la información de Project Server mediante Project Web Access para:

- Revisar y actualizar asignaciones de tareas.
- Ver tareas en el formato de diagrama de Gantt.
- Integración con Microsoft Outlook.

Otras personas interesadas también puede tener acceso a la información para:

- Revisar la información sobre el proyecto, las tareas y los recursos.

### Ventajas de Project Server

- Colaboración y coordinación de actividades con facilidad.
- Administración eficaz de los recursos.

### Desventajas de Project Server

- Sólo funciona en máquinas de 64 bit.
- Poco conocido.

### **1.3.5 RedMine**

Red Mine es una herramienta de fuente abierta, permite la gestión de proyectos basado en web y seguimiento de errores. Incluye calendario y diagramas de Gantt para ayudar a la representación visual de los proyectos y sus plazos. Es compatible con varios proyectos. RedMine es libre y software de código abierto que proporciona características integradas de gestión de proyectos, seguimiento de problemas y soporte para múltiples opciones de control de versiones.

Una vez instalada, el administrador da de alta los proyectos a través de la interface web, puede dar de alta a los desarrolladores y jefes de proyecto (o pueden darse de alta ellos mismos a través de la interface web). [10]

## 1.3.6 MGRSoft

El Modelo consta de un *Sistema de Procesos* que contempla procedimientos, roles y métricas y que se basa en la utilización de la experiencia acumulada por la empresa en materia de revisiones que es almacenada en una *Base de Conocimientos* que posee toda la información referente al proceso de revisiones. Además, se incluye un conjunto de *Herramientas Automatizadas* integradas en un paquete que permite gestionar dicho conocimiento en los dos procesos definidos dentro del modelo. La Base de Conocimientos contiene la información referente a las revisiones que permite contar con la experiencia acumulada en este proceso para que pueda ser utilizada en el desarrollo y revisión de nuevos proyectos. MGRSoft incluye una propuesta de las métricas particulares para el proceso de revisión tanto a nivel estratégico como a nivel de proyectos que permiten analizar los resultados de la ejecución de las revisiones. [11]

Estas herramientas de planificación son poco factibles, las mismas planifican las tareas mediante un flujo de actividades de forma continua donde generalmente una actividad depende del fin de otra para poder dar inicio, funcionando como un diagrama de secuencia de actividades de forma lineal carente de un sistema de planificación dinámico. Dichas herramientas carecen además de sistemas de búsquedas que respalden las necesidades a satisfacer para el proceso de planificación de las revisiones que se realizan en el CESIM, basado en criterios de búsquedas según las estrategias particulares de trabajo asumidas por el Grupo de Calidad pertenecientes al CESIM.

## 1.4 Tecnologías, Herramientas y Metodologías

Teniendo en cuenta que la herramienta a implementar constituirá un módulo de la plataforma a desarrollarse por parte del Grupo de Calidad del CESIM y para la cual se definió una arquitectura previa donde se determinaron las herramientas, metodologías y lenguaje de programación, la realización del módulo de Revisiones se ajusta a la definición realizada de forma tal que sea compatible en el momento de la integración tanto para la plataforma del proyecto como para la facultad, las cuales serán creadas bajo los mismos principios de arquitectura, herramientas, metodologías y lenguaje de programación.

Para el desarrollo de la herramienta se define como lenguaje de programación Python, como framework Django y como Entorno Integrado de Desarrollo (IDE) WingIDE. Como herramienta

de modelado se define el uso de Visual Paradigm utilizando para los procesos del negocio la notación BPMN y para el resto de las disciplinas UML. Como metodología para desarrollar la herramienta se define el uso de RUP y PostgreSQL como Gestor de Base.

## 1.4.1 Lenguaje de Programación, Python

Python fue creado a finales de la década de los 80 por el holandés Guido van Rossum el cual lanzó en 1991 el código versión 0.9.0. Varias han sido desde entonces las versiones de este lenguaje hasta llegar a la actualidad en su versión 3.1.1 lanzada en junio de 2009 por la Python Software Foundation.

Este lenguaje se puede ejecutar sobre Windows, Linux/Unix, Mac OS X y ha sido portado a Java y .Net máquinas virtuales. Es libre de usar, incluso para productos comerciales, debido a su licencia OSI (Iniciativa de Fuente Abierta) de código abierto. [12]

Entre las principales bondades que brinda este lenguaje de programación se encuentra el ser un lenguaje multiparadigma lo que se puede interpretar como que el mismo permite optar por varios estilos ya sea programación orientada a objetos, estructurada o funcional.

Características del lenguaje Python:

### ➤ Multiplataforma

Hay versiones disponibles de Python en muchos sistemas informáticos distintos. Originalmente se desarrolló para Unix, aunque cualquier sistema es compatible con el lenguaje siempre y cuando exista un intérprete programado para él.

### ➤ Interpretado

Quiere decir que no se debe compilar el código antes de su ejecución. En realidad sí que se realiza una compilación, pero esta se realiza de manera transparente para el programador. En ciertos casos, cuando se ejecuta por primera vez un código, se producen unos bytecodes que se guardan en el sistema y que sirven para acelerar la compilación implícita que realiza el intérprete cada vez que se ejecuta el mismo código.

### ➤ Interactivo

Python dispone de un intérprete por línea de comandos en el que se pueden introducir sentencias. Cada sentencia se ejecuta y produce un resultado visible, que puede ayudarnos a entender mejor el lenguaje y probar los resultados de la ejecución de porciones de código rápidamente.

# Fundamentación Teórica

## ➤ Orientada a Objetos

La programación orientada a objetos está soportada en Python y ofrece en muchos casos una manera sencilla de crear programas con componentes reutilizables.

## ➤ Funciones y Librerías

Dispone de muchas funciones incorporadas en el propio lenguaje, para el tratamiento de strings, números, archivos, etc. Además, existen muchas librerías que podemos importar en los programas para tratar temas específicos como la programación de ventanas o sistemas en red o cosas tan interesantes como crear archivos comprimidos en .zip.

## ➤ Sintaxis clara

Por último, destacar que Python tiene una sintaxis muy visual, gracias a una notación indentada (con márgenes) de obligado cumplimiento. En muchos lenguajes, para separar porciones de código, se utilizan elementos como las llaves o las palabras clave *begin* y *end*. Para separar las porciones de código en Python se debe tabular hacia dentro, colocando un margen al código que iría dentro de una función o un bucle. Esto ayuda a que todos los programadores adopten unas mismas notaciones y que los programas de cualquier persona tengan un aspecto muy similar.

## Ventajas de Python

Entre las características que distinguen a este lenguaje de programación se encuentran varias de sus principales ventajas, resaltando entre ellas:

- Sintaxis simple, clara y sencilla.
- Tipado dinámico.
- Gran cantidad de librerías disponibles.
- Lenguaje de alta potencia.

Por tanto se hace óptimo el uso de Python dada las grandes ventajas del lenguaje ante el fácil aprendizaje y uso del mismo para cualquier nivel de conocimiento de los desarrolladores. Sus librerías y bibliotecas lo convierten en un poderoso lenguaje. Se ajusta todo tipo de programación (Programación Orientada a Objeto, Programación Estructurada o Programación Funcional). Trabaja con entornos de desarrollos con bajo consumo de máquina lo que aumenta el rendimiento de las estaciones de trabajo. Actualmente gana aceptación en el mercado mundial de la industria del software por los beneficios que brinda, lo que lo convierte en un

fuerte rival para otros lenguajes, asegurando de esta forma su continuidad de uso y actualización con los consiguientes aumentos en soportes y garantías.

### 1.4.2 Framework, Django Framework

Es un *framework* de desarrollo para la web de código abierto desarrollado escrito sobre Python, y cumple con el paradigma del Modelo Vista Controlador, su primera versión fue en julio de 2005 hasta llegar a la actual versión 1.1 lanzada en julio de 2009.

Inicialmente Django fue desarrollado para gestionar aplicaciones web de páginas orientadas a noticias de *World Online*, más tarde se liberó bajo licencia BSD. [13]

La meta fundamental de Django es facilitar la creación de sitios web complejos. Django pone énfasis en el re-uso, la conectividad y extensibilidad de componentes, del desarrollo rápido y del principio de DRY (del inglés *Don't Repeat Yourself*). Python es usado en todas las partes del framework, incluso en configuraciones, archivos, y en los modelos de datos. [14]

Facilidades que ofrece Django:

- Mapeador objeto-relacional.

Define modelos de datos completamente en Python y accede a los datos mediante una potente API de acceso dinámico a la base de datos.

- Interfaz de administración automático.

Ahorra los tediosos trabajos de crear interfaces para añadir o actualizar contenidos. Django genera un sistema de administración automáticamente listo para ser usado en producción.

- Diseño de URLs elegantes.

Django permite diseñar bellas URLs sin limitaciones específicas del framework.

- Sistema de plantillas.

Posee un potente y extensible lenguaje de plantillas para separar diseño, contenido y código Python, de una forma amigable para diseñadores.

- Sistema de cache.

Utiliza el *memcached* u otro framework de cache para obtener un súper-rendimiento. El cacheo es tan granular como se desee.

- Internacionalización.

Django soporta aplicaciones multilinguaje permitiendo especificar strings de traducción y proporcionando herramientas para funcionalidades específicas del lenguaje.

### 1.4.3 Entorno de desarrollo, Wing IDE

Wing IDE es un entorno integrado de desarrollo especialmente diseñado para el lenguaje de programación Python, disponible en Linux. Permite el desarrollo rápido de aplicaciones de plataforma cruzada para escritorio, web y empresariales. Wing IDE se enfoca en incrementar la productividad y la calidad del código, especialmente en proyectos complejos con requerimientos cambiantes. [15]

Wing IDE permite el desarrollo de plataformas de escritorios y de aplicaciones web, un manejo fácil en la integración de aplicaciones y un manejo sencillo de las pruebas de software.

Se propone el uso de este IDE para el desarrollo del sistema propuesto por las ventajas que ofrece en el desarrollo de aplicaciones web con lenguaje Python, lenguaje para el cual ha sido totalmente creado con su consiguiente ahorro en recursos económicos y de máquina. Además su uso presupone la realización eficiente de pruebas al sistema a desarrollar y un manejo fácil para posibles integraciones con posibles plataformas que le puedan dar soporte.

### 1.4.4 Metodología de Desarrollo, RUP

Es el acrónimo Rational Unified Process (Proceso Unificado de Rational). Este es una metodología de la Ingeniería de Software el cual logra disciplinar el trabajo en un equipo de desarrollo pues el mismo define quien debe hacer qué, cómo y cuándo.

RUP es una metodología 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. [16]

No es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. RUP es un producto de Rational de la IBM (*International Business Machines*). Se caracteriza por ser iterativo e incremental,

estar centrado en la arquitectura y guiado por los casos de uso, la administración de requisitos, el uso de arquitectura basada en componentes, el control de cambios, el modelado visual del software y especial énfasis en la verificación de la calidad del software.[17]

### 1.4.5 Notación de Modelado, BPMN

BPMN (*Business Process Modeling Notation*, en español Notación para el Modelado de Procesos de Negocio) es una notación gráfica estandarizada que permite el modelado de procesos de negocio, en un formato de flujo de trabajo (*workflow*). Inicialmente desarrollada por la organización BPMP (*Business Process Management Initiative*), es actualmente mantenida por el OMG (*Object Management Group*), luego de la fusión de las dos organizaciones en el año 2005.

El principal objetivo de BPMN es proveer una notación estándar que sea fácilmente leíble y entendible por parte de todos los involucrados e interesados del negocio (*stakeholders*). Entre estos interesados están los analistas de negocio (quienes definen y redefinen los procesos), los desarrolladores técnicos (responsables de implementar los procesos) y los gerentes y administradores del negocio (quienes monitorizan y gestionan los procesos).

Actualmente hay una amplia variedad de lenguajes, herramientas y metodologías para el modelado de procesos de negocio. La cada vez mayor adopción de la notación BPMN como estándar ayudará a unificar la expresión de conceptos básicos de procesos de negocio (por ejemplo: procesos públicos y privados, orquestación, coreografía, etc.) así como conceptos avanzados de modelado (por ejemplo manejo de excepciones, compensación de transacciones, entre otros).

### 1.4.6 Lengua 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.[18]

## Fundamentación Teórica

UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir.

UML intenta solucionar el problema de propiedad de código que se da con los desarrolladores, al implementar un lenguaje de modelado común para todos los desarrollos, se crea una documentación también común, que cualquier desarrollador con conocimientos de UML será capaz de entender, independientemente del lenguaje utilizado para el desarrollo. [19]

### **1.4.7 Herramienta de Modelado, 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. [20]

Posibilita la representación gráfica de los diagramas permitiendo ver el sistema desde diferentes perspectivas, como el de componentes, despliegue, secuencia, casos de uso, clase, actividad, estado, entre otros. Además, identifica requisitos y comunica información, se centra en cómo los componentes del sistema interactúan entre ellos, sin entrar en detalles excesivos, además, permite ver las relaciones entre los componentes del diseño y mejora la comunicación entre los miembros del equipo usando un lenguaje gráfico.

Tiene disponible distintas versiones: Enterprise, Professional, Standard, Modeler, Personal y Community. Facilita licencias especiales para fines académicos.

Entre sus características más significativas se puede resaltar que su licencia es gratuita, posee varios idiomas, sus ediciones son compatibles y es de fácil uso para la creación de aplicaciones web.

### **1.4.8 Gestor de Base de Datos, PostgreSQL**

Es un motor de base de datos surgido en 1986 con el lanzamiento de su primera versión. Este gestor es altamente potente y posee prestaciones y funcionalidades equivalentes a otros de

# Fundamentación Teórica

carácter comercial. Es más completo que MySQL ya que permite métodos almacenados, restricciones de integridad y vistas. Actualmente su última versión, la 8.4 fue lanzada en julio del 2009.

Es un sistema de gestión de base de datos relacional orientada a objetos de software libre, publicado bajo la licencia BSD (*Berkeley Software Distribution*).

PostgreSQL es un sistema de bases de datos objeto-relacional con características de los mejores sistemas de bases de datos comerciales. PostgreSQL es libre y su código fuente completo está disponible [21].

Algunas de sus características principales:

- Alta concurrencia.
- Amplia variedad de tipos nativos.
- Uso de disparadores.
- Funciones de Ventanas.
- Expresiones de tablas comunes y consultas recursivas.
- Instalaciones ilimitadas.
- Estabilidad y confiabilidad.
- Gran soporte a proveedores.
- Extensible.
- Multiplataforma.
- Diseño para ambientes de amplio volumen.

Realizado el estudio de las ventajas de los anteriores gestores de bases de datos se hace recomendable el uso de PostgreSQL dada sus ventajas fundamentales de uso libre y soportar varias plataformas, lo que aumenta su potencia de uso y aplicación en distintos sistemas operativos.

## **1.4.9 Tipos de Aplicaciones**

### **1.4.9.1 Aplicaciones web**

Una aplicación web es un programa de computadora que los usuarios utilizan accediendo a un servidor web a través de Internet o de una intranet. Las aplicaciones web utilizan las tecnologías existentes para generar contenidos dinámicos y permitir a los usuarios del Sistema

# Fundamentación Teórica

modificar la lógica del negocio en el servidor. Esta aplicación informática es popular debido a la practicidad del navegador web como cliente ligero. [22]

Es importante mencionar que una página web puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones, como por ejemplo rellenar y enviar formularios, participar en juegos diversos y acceder a gestores de base de datos de todo tipo.[23]

## 1.4.9.2 Aplicaciones de Escritorio

Es la aplicación creada para ejecutarse en un ordenador de escritorio, sobre un sistema operativo de interfaz visual como Windows o Linux. [24]

## 1.4.9.3 Ventajas de las Aplicaciones web sobre las de Escritorio

Aplicaciones web	Aplicaciones Escritorio
<ul style="list-style-type: none"><li>➤ No necesita instalar ningún tipo de programa o software.</li><li>➤ Se accede desde cualquier ordenador conectado a una red en la cual se encuentre la aplicación.</li><li>➤ Brinda privacidad para el acceso a sus datos. Desglosa las responsabilidades acorde a los niveles de usuarios.</li><li>➤ No depende su uso del sistema operativo, puede ser usada con tan sólo tener un navegador web.</li><li>➤ Facilidad de recuperación de contraseñas.</li><li>➤ Brinda un sistema avanzado para la consulta, altas, bajas y modificaciones de datos provenientes de cualquier área. Permite mantener actualizado los</li></ul>	<ul style="list-style-type: none"><li>➤ Mayor calidad gráfica visual.</li><li>➤ Menor tiempo de respuesta.</li><li>➤ Mayor personalización.</li></ul>

datos de información.	
-----------------------	--

Realizado el análisis de las ventajas que ofrecen cada uno de los distintos tipos de aplicación informáticas existentes en la actualidad, es evidente el óptimo uso de las aplicaciones web para el desarrollo de aplicaciones informáticas. Su selección es avalada también por la necesidad de lograr un alto nivel de seguridad, un acceso fácil y rápido desde distintos puestos de trabajo y la facilitación del uso de la misma ajustado a la dinámica de trabajo actual.

### **Conclusiones Parciales**

El presente capítulo ha posibilitado introducir y conocer aspectos fundamentales que nos permite adentrarnos en el proceso de revisiones que se realiza en el CESIM, así como a las principales funcionalidades que debe de ser creadas para la informatización del proceso. Permitió acercarse a los principales conceptos asociados al entorno y objetivos de trabajo y al alcance del mismo. Se valoran las principales herramientas a utilizar así como las tendencias actuales en el campo de estudio. Se establecieron comparaciones con elementos similares para corroborar la eficacia de la selección propuesta y se validaron los posibles resultados finales a obtener.

## Capítulo 2. Características del Sistema

En el siguiente capítulo se describen las principales características del sistema a desarrollar. Como resultado del desarrollo de este proceso se hace una descripción de cada uno de los trabajadores que intervienen, los actores y los casos de usos del negocio. Se definen y documentan los requisitos funcionales y no funcionales a tener en cuenta así como los actores y los casos de usos del sistema.

### 2.1 Descripción General del Objeto de Estudio

Como resultado del acelerado proceso de desarrollo de la industria del software en el mundo, Cuba ha tenido que desarrollar de forma acelerada su propia industria nacional para el desarrollo y fomento de la misma. Varias son las instituciones que a nivel nacional se encargan del desarrollo de esta novedosa industria en aras de brindar soluciones tecnológicas que respondan a los intereses de desarrollo económico trazado por el país.

El Proceso de Aseguramiento de la calidad, está contemplado como el proceso más importante para lograr un producto altamente competitivo y funcional que permita insertarse de forma satisfactoria en el mercado con las exigencias requeridas. Este define todo un conjunto de actividades a desarrollar para garantizar que todos los artefactos resultantes del desarrollo del software estén a la altura de las normas y exigencias internacionales. Una de las actividades definidas para asegurar que la calidad sea óptima en el software desarrollado y que todos sus artefactos generados estén en condiciones de aceptación por parte de los clientes son las revisiones.

El objeto de estudio del siguiente trabajo, agrupa los procesos para la gestión de las revisiones en el Grupo de Calidad del Centro, identificando las actividades a desarrollar para la realización de las revisiones a los proyectos asociados al CESIM, además agrupa los procesos para el uso de las Listas de Chequeo que se utilizan en la realización de cada una de las revisiones y los reportes a emitir con los resultados arrojados; así como viabilizar el seguimiento de cada revisión realizada con el manejo de una base de datos donde estará archivada toda la documentación correspondiente para posterior uso, teniendo en cuenta la actualización de las diferentes fases en que se encuentra el desarrollo del software de acuerdo a las nuevas exigencias propuestas por los procesos de mejoras que se desarrollan.

## 2.2 Situación Problemática

Como resultado de la investigación realizada para obtener información detallada de como se realiza el proceso de revisiones en los proyectos productivos pertenecientes al CESIM, se detectaron un conjunto de problemas que son relacionados a continuación, lo que justifica el por qué la realización de una herramienta que permita la humanización y la optimización de las revisiones dentro del Grupo de Calidad del CESIM.

Todo el trabajo que hoy se realiza dentro del Grupo de Calidad del CESIM para el manejo y realización de revisiones se tramita de forma informal mediante el uso de correo electrónico, sistemas de mensajería instantánea (jabber) o mediante comunicación personal entre las partes involucradas en el proceso. Esto evidencia la posibilidad de la no materialización de la revisión de acuerdo al cronograma previsto, justificado por la alta probabilidad de olvido de la misma o la no consulta en tiempo de los medios utilizados para comunicar la realización de la actividad.

Una de las características de las revisiones es que son realizadas por un grupo de revisores los cuales trabajan en los distintos elementos a revisar. Las dificultades detectadas por cada uno de los revisores involucrados al aplicar las Listas de Chequeo son registradas en el Registro de Dificultades y Acciones Correctivas. Todo el proceso descrito anteriormente es realizado por uno o más revisores por lo que puede contribuir a la aparición de errores, omisiones y duplicaciones en el momento de elaborar el RDAC y el Informe Final de Resultados.

Actualmente toda la documentación obtenida de cada revisión es guardada en una computadora dentro del Grupo de Calidad del CESIM. En la misma se destina un espacio en uno de los compartimientos de su disco de almacenamiento de información. En dicho espacio queda guardada toda esta información la cual crece mensualmente dado que la actividad de revisión es de forma escalonada y periódica hasta obtener el resultado esperado en la documentación de los proyectos. El cúmulo de documentos generados por cada revisión es considerablemente alto, de ahí que la búsqueda de los mismos para posteriores seguimientos se hace engorrosa y en ocasiones provoca atrasos en los cronogramas de realización. Esta información está expuesta además al peligro de ser perdida de forma total ante una falla o daño que sufra dicha computadora pues la misma carece de salvvas.

## **2.3 Descripción del flujo de trabajo en la realización de un proceso de revisión**

El análisis del flujo de trabajo entre el Grupo de Auditoría y Revisiones del Grupo de Calidad del CESIM; encargados de realizar las revisiones, y los miembros de los proyectos productivos del CESIM permiten tener una noción lo más certera posible de como se lleva a cabo el proceso de revisiones a los proyectos.

Todo comienza cuando el revisor se presenta en la fecha establecida en el cronograma de revisiones ante el proyecto a revisar. El revisor en conjunto con los miembros del proyecto ejecuta la revisión acorde a las actividades establecidas. Terminada la revisión se procederá a realizar la reunión de cierre donde se da por terminado el proceso.

## **2.4 Procesos del Negocio**

Thompson, R.J y Redstone, L (1997) definen un proceso como una serie de actividades interconectadas en busca de un propósito. En un contexto de negocios, el propósito de los procesos será proveer a sus clientes internos con sus requerimientos de una manera oportuna. [25]

En la Ingeniería de Software un proceso del negocio define: “quién, qué, cuándo y cómo” hay que realizar las cosas para lograr un determinado producto de software. Un Proceso de Desarrollo de Software es la definición del conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto, a modo de plantilla que explica los pasos necesarios para terminar el proyecto. [26]

## **2.5 Descripción de los procesos del negocio**

El revisor se presenta en la fecha establecida en el cronograma de revisiones ante el proyecto a revisar. El revisor y los miembros del proyecto ejecutan en conjunto la reunión de apertura donde son especificados los objetivos, los cuales están sujetos al tipo de revisión que se realiza. Una vez culminada la reunión el revisor procederá a la revisión de cada uno de los artefactos generados como resultado de la culminación de la fase, aplicando para ello las Listas de Chequeo previamente establecidas con los parámetros a tener en cuenta. Durante el proceso además se realizarán encuestas a los distintos miembros del proyecto acorde al rol que desempeñan. Una vez terminada la revisión a los artefactos se crea el Registro de Dificultades y Acciones Correctivas y se debate entre los involucrados las dificultades detectadas. Terminado todo lo antes descrito se procederá a elaborar el Informe Final de

# Características del Sistema

Resultados el cual se presenta en la reunión de cierre que se ejecuta para dar por terminado el proceso.

## **2.6 Objeto de Informatización**

Analizando los problemas que actualmente existen en el Grupo de Calidad del CESIM para realizar las revisiones a los proyectos productivos se hace la propuesta de desarrollar funcionalidades que faciliten la planificación de las revisiones y la revisión en sí a los proyectos asociados al CESIM. De esta forma se garantizarán funcionalidades que eliminarán trabajos que actualmente se realizan por los involucrados en el proceso, lo cual es una garantía para minimizar los errores, las omisiones y las duplicaciones de datos, contribuyendo de esta forma a un mejor desarrollo del proceso que se realiza actualmente.

## **2.7 Propuesta de Sistema**

### **2.7.1 Descripción General de la Propuesta del Sistema**

Después del análisis de la situación problemática existente en el Grupo de Calidad del CESIM en el proceso de revisiones a los proyectos productivos, se identificó la necesidad de desarrollar una herramienta semi automatizada que permita la gestión de todos los procesos involucrados en las revisiones que se realizan en la actualidad. Entre sus principales beneficios se encuentran la planificación de un cronograma de revisiones que permite la consulta del mismo de forma permanente por las partes involucradas, el trabajo con las Listas de Chequeos, así como la emisión de reportes con los resultados finales de cada revisión realizada. Permitirá además contar con una base de datos con toda la documentación de cada revisión, lo cual permitirá un fácil seguimiento acorde a la fase en que se materializa la misma. Constará además con facilidades de configuración que permitan adaptarla a los nuevos procesos de mejoras.

## **2.8 Modelo de Negocio**

El modelo de negocio es una técnica que se utiliza dentro del entorno del negocio para llevar a cabo la ejecución de la descripción de los procesos de la organización. El mismo permite especificar los requisitos más importantes del futuro sistema mediante el negocio.

Su objetivo final es describir cada uno de los procesos del negocio describiendo los actores y trabajadores que intervienen en cada uno así como las actividades, flujo de datos y las reglas del negocio establecidas.

## Características del Sistema

Este modelo proporciona el entendimiento entre las partes involucradas, dígame clientes y desarrolladores permitiendo que entre ambos se logre la comprensión de la estructura dinámica de la organización en la cual va a ser desarrollado el sistema, además de permitir identificar los riesgos y las posibles mejoras.

### 2.8.1 Identificación de los roles del entorno del Negocio

Una vez que se ha terminado la identificación de los procesos del negocio que intervienen se hace necesario encontrar y definir cada uno de los involucrados en su realización así como el rol que cada uno desempeñará.

**Actor del negocio:** Cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos con los que el negocio interactúa.

Actores del negocio	Descripción
<b>Proyecto</b>	Es el principal beneficiado con los resultados que se obtienen del negocio. Es la entidad a la que se le realizan las revisiones.

**Trabajadores del negocio:** Representa a personas o sistemas dentro del negocio que son las que realizan las actividades que están comprendidas dentro de un caso de uso, permaneciendo dentro de la frontera del negocio.

Trabajadores del negocio	Descripción
<b>Planificador</b>	Se encarga de planificar las revisiones, para ello, debe verificar los cronogramas establecidos para cada fase y la fecha en que culmina cada una de ellas en el proyecto, o revisar el Registro de Dificultades y Acciones Correctivas actualizado, para planificar la revisión en dependencia de los plazos

## Características del Sistema

	establecidos en los cronogramas.
<b>Revisor</b>	Es el encargado realizar las revisiones y elaborar el Registro de Dificultades y Acciones Correctivas.
<b>Asesor de Calidad</b>	Se encarga de monitorear y escalar las dificultades detectadas en el proceso de revisiones.
<b>Nivel Competente</b>	Se encargará de aplicar la medida correspondiente u otorgar determinados permisos para la corrección de las dificultades.

## 2.8.2 Diagramas de procesos del negocio

### Diagrama de proceso del negocio Planificar Revisión

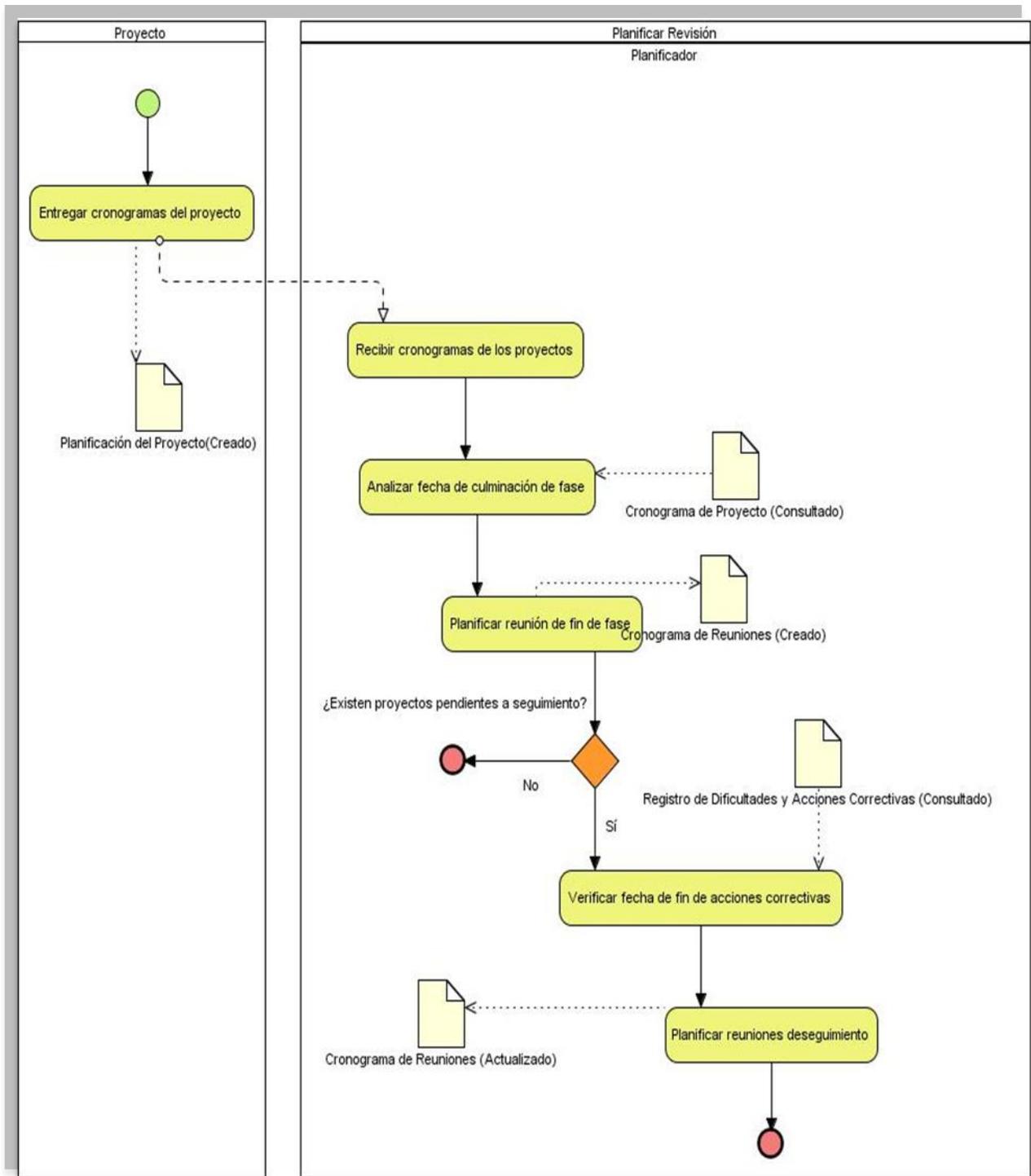


Figura 1 Diagrama de proceso del negocio Planificar Revisión

Diagrama de proceso del negocio Realizar Revisión de Fin de Fase

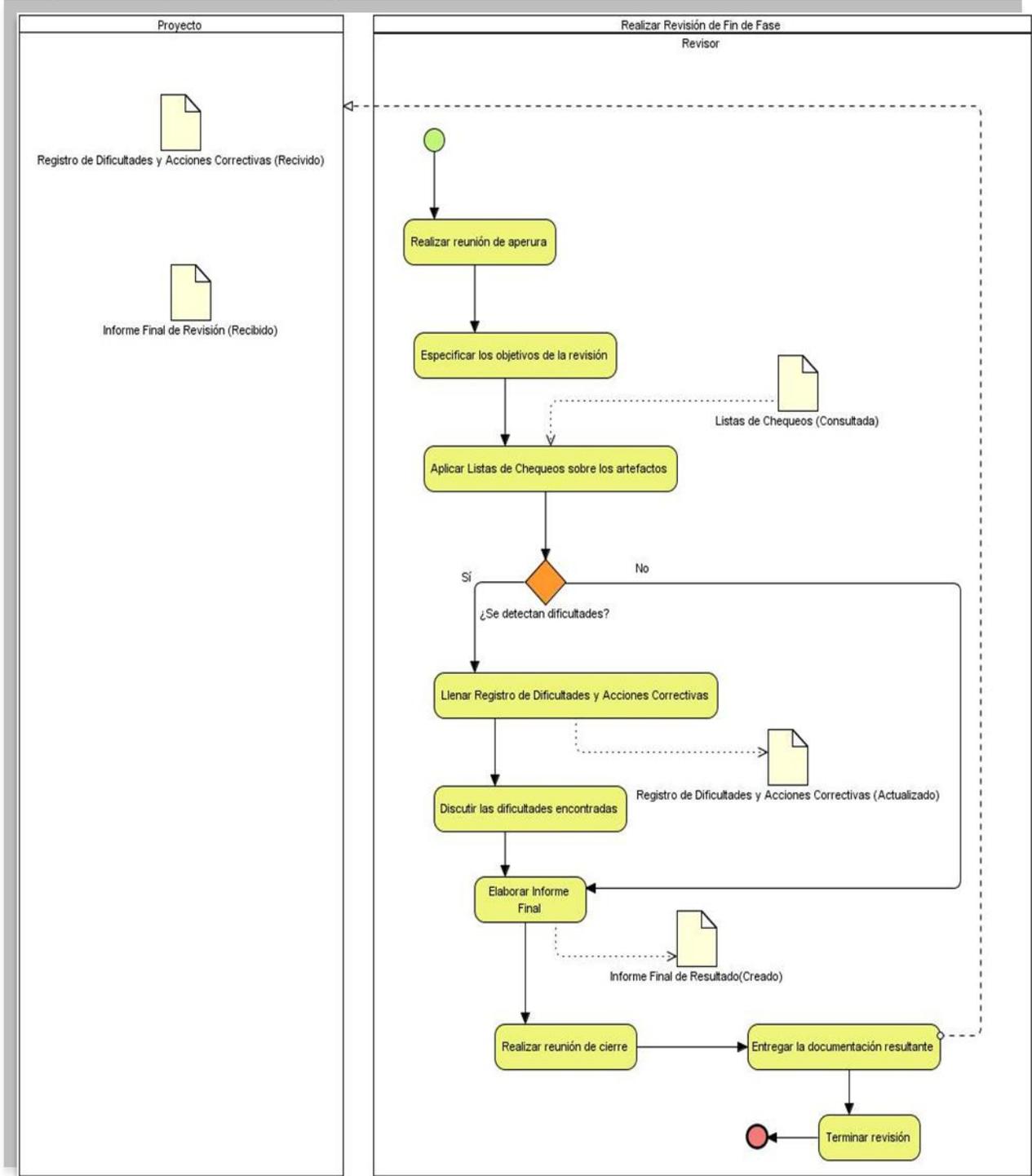


Figura 2 Diagrama de proceso del negocio Realizar Revisión de Fin de Fase

## Diagrama de proceso del negocio Realizar Revisión de Seguimiento

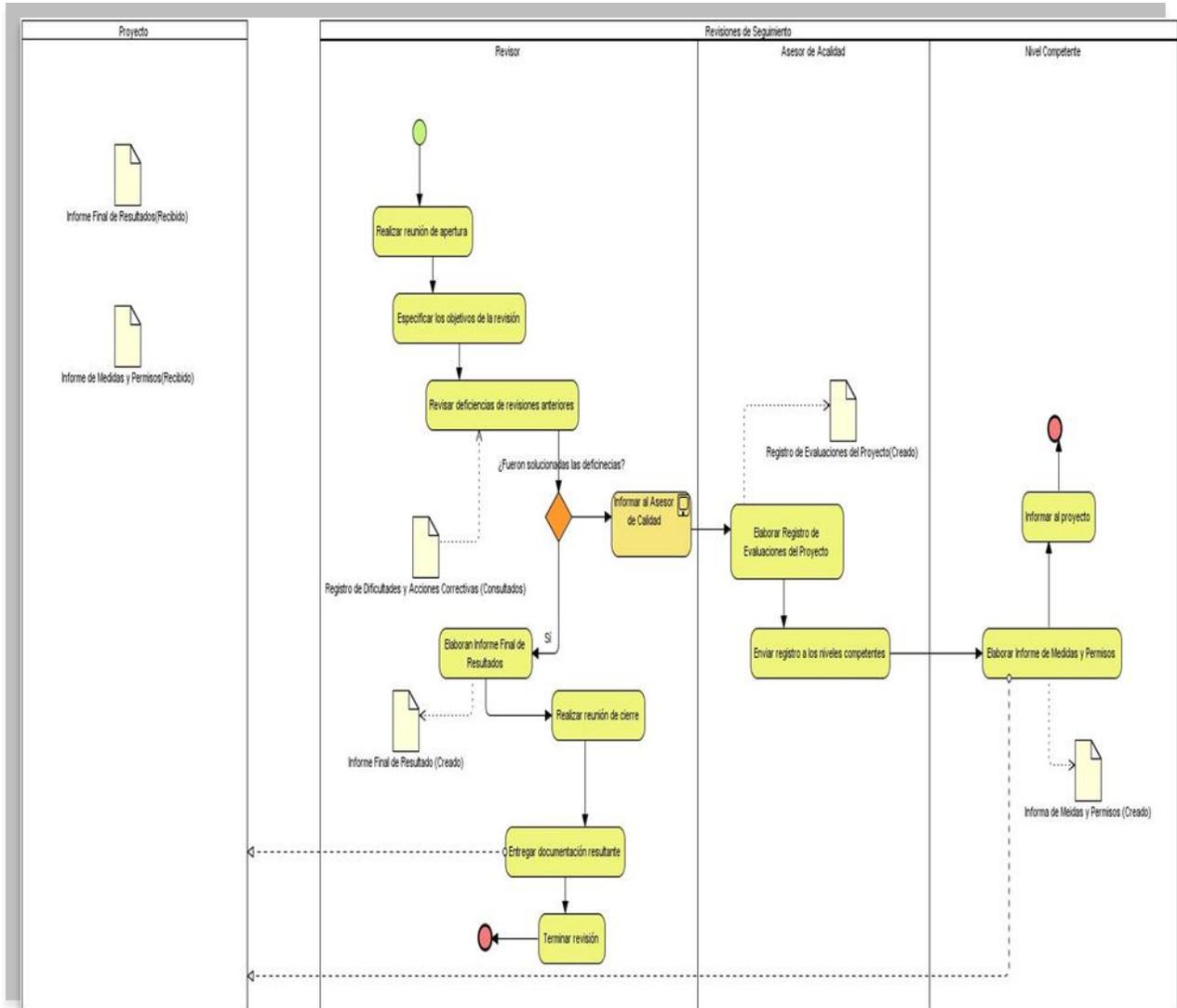


Figura 3 Diagrama de proceso del negocio Realizar Revisión de Seguimiento

## 2.8.3 Descripción de los procesos del negocio

<b>PROCESO:</b>	Planificar Revisión	
<b>Misión:</b>	Se realiza la planificación de las revisiones asignándole una fecha la cual está ajustada a los cronogramas entregados por el proyecto o a los plazos establecidos para las correcciones a realizar.	
<b>Responsable:</b>	Planificador	
<b>ACTORES INVOLUCRADOS</b>		
	<b>Rol</b>	<b>Funciones</b>
	Planificador:	Planifica las fechas en que se realizarán las revisiones a los proyectos.
	Proyecto:	Entrega los cronogramas de ejecución del proyecto y los Registros de Dificultades y Acciones Correctivas actualizados.
<b>ACTIVIDADES</b>		

## Características del Sistema

<b>Actividad:</b>	Planificar revisiones			
<b>Flujo de información</b>				
Cuando el proyecto entrega los cronogramas se le asigna una fecha por parte del planificador para llevar a cabo la ejecución de la revisión.				
<b>Artefacto</b>	<b>Emisor</b>	<b>Receptor</b>	<b>Formato</b>	<b>Frecuencia</b>
Cronograma de revisiones	Planificador	Proyecto	Digital	Mensual
<b>REGLAS DEL NEGOCIO</b>				
<b>Regla</b>		<b>Actividad</b>		
Un proyecto tendrá una sola Revisión de Fin de Fase por cada una de las que transite.		No existe		
<b>REFERENCIAS A PROCEDIMIENTOS</b>				
<b>Procedimiento</b>		<b>Descripción</b>		
No existe		No existe		

<b>PROCESO:</b>	Realizar Revisión de Fin de Fase
	Realizar la revisión previamente planificada con el objetivo de detectar

## Características del Sistema

<b>Misión:</b>	errores y deficiencias en el proceso de desarrollo del software y llevar a cabo la documentación de las mismas para posterior corrección.			
<b>Responsable:</b>	Revisor			
<b>ACTORES INVOLUCRADOS</b>				
<b>Rol</b>		<b>Funciones</b>		
Revisor		Realiza las revisiones a los proyectos para detectar errores y dificultades.		
Proyecto		Es sometido al proceso de revisiones previamente planificadas.		
<b>ACTIVIDADES</b>				
<b>Actividad:</b>	Aplicar Listas de Chequeos			
<b>Flujo de información</b>				
El revisor aplica los parámetros establecidos en la Lista de Chequeo en cada uno de los elementos a revisar.				
<b>Artefacto</b>	<b>Emisor</b>	<b>Receptor</b>	<b>Formato</b>	<b>Frecuencia</b>

## Características del Sistema

Lista de Chequeo.	Departamentos a los distintos niveles que se encuentren involucrados en las revisiones.	Revisores del Grupo de Calidad del CESIM.	Digital	Cuando los departamentos ajustan todos los parámetros a tener en cuenta en las revisiones, atendiendo a las nuevas prácticas y exigencias de calidad establecidas y a las metodologías de trabajo.
<b>Actividad:</b>	Llenar el Registro de Dificultades y Acciones Correctivas			
<b>Flujo de información</b>				
Al concluir la aplicación de las Listas de Chequeo se confecciona el Registro de Dificultades y Acciones Correctivas con las deficiencias detectadas en la revisión.				
<b>Artefacto</b>	<b>Emisor</b>	<b>Receptor</b>	<b>Formato</b>	<b>Frecuencia</b>
	Revisor	Proyecto	Digital	Se realiza uno

## Características del Sistema

Registro de Dificultades y Acciones Correctivas.				por cada revisión que se aplica al proyecto.
<b>Actividad:</b>	Elaborar Informe Final de Resultados.			
<b>Flujo de información</b>				
Al concluir cada revisión se elabora un informe donde queda reflejado los resultados obtenidos en la misma.				
<b>Artefacto</b>	<b>Emisor</b>	<b>Receptor</b>	<b>Formato</b>	<b>Frecuencia</b>
Informe Final de Resultados.	Revisor	Proyecto	Digital	Se realiza uno por cada revisión que se aplica al proyecto.
<b>REGLAS DEL NEGOCIO</b>				
<b>Regla</b>		<b>Actividad</b>		
No existe		No existe		
<b>REFERENCIAS A PROCEDIMIENTOS</b>				
<b>Procedimiento</b>		<b>Descripción</b>		
No existe		No existe		

## Características del Sistema

<b>PROCESO:</b>	Realizar Revisión de Seguimiento	
<b>Misión:</b>	Realizar la revisión previamente planificada con el objetivo de verificar el cumplimiento en la corrección a las dificultades encontradas en revisiones anteriores y realizar los documentos correspondientes a dicha actividad.	
<b>Responsable:</b>	Revisor	
<b>ACTORES INVOLUCRADOS</b>		
	<b>Rol</b>	<b>Funciones</b>
	Revisor	Verificar el cumplimiento en la corrección de las dificultades encontradas.
	Asesor de Calidad	Escalar las dificultades encontradas en el proceso de desarrollo a niveles superiores.
	Nivel competente	Aplicar las medidas pertinentes a los proyectos.
	Proyecto	Es sometido al proceso de revisiones previamente planificado.
<b>ACTIVIDADES</b>		

## Características del Sistema

<b>Actividad:</b> Actualizar el Registro de Dificultades y Acciones Correctivas				
<b>Flujo de información</b>				
El revisor se encarga de verificar si se le dio solución a cada una de las dificultades encontradas durante la revisión de fin de fase y actualiza el Registro de Dificultades y Acciones Correctivas				
Artefacto	Emisor	Receptor	Formato	Frecuencia
Registro de Dificultades y Acciones Correctivas.	Proyecto	Revisor	Digital	Siempre que se realiza una revisión de seguimiento a un proyectos.
<b>Actividad:</b> Elaborar el Informe Final de Resultados.				
<b>Flujo de información</b>				
Al concluir cada revisión se elabora un informe donde queda reflejado los resultados obtenidos en la misma.				
Artefacto	Emisor	Receptor	Formato	Frecuencia
Informe Final de	Revisor	Proyecto	Digital	Se realiza uno

## Características del Sistema

Resultados.				por cada revisión que se aplica al proyecto.
<b>SUBPROCESOS</b>				
<b>Subproceso:</b>	Escalar deficiencias			
<b>Misión:</b>	Informar a niveles superiores el no cumplimiento de las acciones correctivas a las dificultades encontradas en las revisiones realizadas con anterioridad.			
<b>Responsable:</b>	Revisor			
<b>ACTORES INVOLUCRADOS</b>				
<b>Rol</b>		<b>Funciones</b>		
Revisor		Verificar el cumplimiento en la corrección de las dificultades encontradas.		
Asesor de Calidad		Elegir las dificultades encontradas en el proceso de desarrollo a niveles superiores.		
Nivel Competente		Aplicar las medidas pertinentes a los proyectos.		
Proyecto		Es sometido al proceso de revisiones previamente planificado.		

## Características del Sistema

ACTIVIDADES				
<b>Actividad:</b>	Escalar las dificultades no corregidas.			
<b>Precondiciones</b>	Deben de detectarse acciones correctivas no resultas en los plazos establecidos.			
Flujo de información				
Al detectar deficiencias que no han sido corregidas en el plazo establecido se procede a escalar la situación a niveles superiores.				
Artefacto	Emisor	Receptor	Formato	Frecuencia
Registro de Evaluaciones del Proyecto	Asesor de Calidad	Nivel Competente	Digital	Se realiza uno por cada revisión de seguimiento en la que se detecten dificultades no corregidas en el plazo establecido.
REGLAS DEL NEGOCIO				
Regla		Actividad		
No existe		No existe		
REFERENCIAS A PROCEDIMIENTOS				
Procedimiento		Descripción		
No existe		No existe		

## 2.9 Requisitos Funcionales

Los requisitos funcionales son capacidades o condiciones con las que debe cumplir el producto a elaborar, no alteran la funcionalidad del software y se mantienen invariables sin importar con que cualidades o propiedades se relacionen. [27]

A partir del estudio realizado en los procesos del negocio que intervienen se definen una gama de requerimientos funcionales que permiten el adecuado funcionamiento del sistema propuesto.

### 2.9.1 Requisitos Funcionales

#### 1. Gestionar Expediente de Revisiones

1.1 Gestionar Expediente de Revisiones de Fin de Fase.

1.1.1 Adicionar Expediente de Revisiones de Fin de Fase.

1.1.2 Buscar Expediente de Revisiones de Fin de Fase.

1.1.3 Modificar Expediente de Revisiones de Fin de Fase.

1.1.4 Mostrar Expediente de Revisiones de Fin de Fase.

1.1.5 Eliminar Expediente de Revisiones de Fin de Fase.

1.2 Gestionar Expediente de Revisiones de Seguimiento.

1.2.1 Adicionar Expediente de Revisiones de Seguimiento.

1.2.2 Buscar Expediente de Revisiones de Seguimiento.

1.2.3 Modificar Expediente de Revisiones de Seguimiento.

1.2.4 Mostrar Expediente de Revisiones de Seguimiento.

1.2.5 Eliminar Expediente de Revisiones de Seguimiento.

#### 2. Gestionar Documentación

2.1 Gestionar Acta de Apertura

2.1.1 Adicionar Acta de Apertura

2.1.2 Buscar Acta de Apertura

2.1.3 Modificar Acta de Apertura

2.1.4 Mostrar Acta de Apertura

2.1.5 Eliminar Acta de Apertura

2.2 Gestionar Acta de Cierre

2.2.1 Adicionar Acta de Cierre

2.2.2 Buscar Acta de Cierre

2.2.3 Modificar Acta de Cierre

2.2.4 Mostrar Acta de Cierre

2.2.5 Eliminar Acta de Cierre

2.3 Gestionar Informe Final de Resultados

2.3.1 Adicionar Informe Final de Resultados

2.3.2 Buscar Informe Final de Resultados

2.3.3 Modificar Informe Final de Resultados

2.3.4 Mostrar Informe Final de Resultados

2.3.5 Eliminar Informe Final de Resultados

### **3. Generar Información del RDAC**

#### **4. Gestionar RDAC**

4.1. Buscar RDAC

4.2 Modificar RDAC

4.3 Mostrar RDAC

4.4 Eliminar RDAC

#### **5. Modificar Información del RDAC**

#### **6. Gestionar Cronograma de Revisiones**

6.1 Gestionar Cronograma de Revisiones de Fin de Fase

6.1.1 Adicionar Cronograma de Revisiones de Fin de Fase

6.1.2 Buscar Cronograma de Revisiones de Fin de Fase

6.1.3 Modificar Cronograma de Revisiones de Fin de Fase

6.1.4 Mostrar Cronograma de Revisiones de Fin de Fase

6.1.5 Eliminar Cronograma de Revisiones de Fin de Fase

6.2 Gestionar Cronograma de Revisiones de Seguimiento

6.2.1 Adicionar Cronograma de Revisiones de Seguimiento

6.2.2 Buscar Cronograma de Revisiones de Seguimiento

6.2.3 Modificar Cronograma de Revisiones de Seguimiento

6.2.4 Mostrar Cronograma de Revisiones de Seguimiento

6.2.5 Eliminar Cronograma de Revisiones de Seguimiento

## **7. Realizar Revisión de Fin de Fase**

## **8. Realizar Revisión de Seguimiento**

## **9. Gestionar Listas de Chequeo**

9.1 Adicionar Listas de Chequeo

9.2 Buscar Listas de Chequeo

9.3 Modificar Listas de Chequeo

9.4 Mostrar Listas de Chequeo

9.5 Eliminar Listas de Chequeo

## **2.10 Requisitos no Funcionales**

Los requisitos no funcionales son cualidades o propiedades que el producto debe tener. Se debe de pensar en cómo el producto puede ser amigable a los usuarios, fácil de usar, confiable y seguro para cualquier entorno en el que sea desplegado.

Atendiendo a las características de la aplicación propuesta y el fin de uso que se propone para la misma, se definen los siguientes requisitos no funcionales.

- **Usabilidad:** El sistema de ser diseñado de forma tal, que los usuarios que harán uso del mismo obtengan los conocimientos necesarios en el menor tiempo posible para la explotación de sus funcionalidades.

## Características del Sistema

- **Seguridad:** Es el requerimiento más complejo y difícil de garantizar, a su vez permitirá garantizar que la aplicación será utilizada correctamente por cada usuario según sus niveles permitidos.
  - **Confidencialidad:** Se debe tratar el manejo de permisos de forma que solamente se acceda a la información autorizada de acuerdo a los niveles de permisos que debe tener cada usuario del sistema. De esta forma se garantiza que la información no sea expuesta a personal indebido.
  - **Integridad:** Se debe de tratar el manejo de la información de forma tal, que la información no sea modificada por personal ajeno, evitando de esta forma alteraciones en los resultados planteados en la documentación.
  - **Disponibilidad:** Se deberá garantizar el acceso pleno de cada usuario con facultades para el uso de la aplicación las 24 horas del día.
- **Respaldo y recuperación de base de datos:** El sistema debe permitir la realización de copias de respaldo hacia otros dispositivos de almacenamiento para prevenir pérdida de información. Esta base de datos debe ser posible de recuperar mediante las copias de respaldo previamente creadas.
- **Documentación en línea y ayuda a los usuarios:** El sistema debe estar provisto de ayudas dinámicas y documentos (manuales de usuarios o tutoriales) que permitan el uso eficiente del sistema y el respaldo a las dudas en su uso.
- **Soporte:** Debe presentar un diseño que sea capaz de la realización de pruebas que permitan la explotación de la aplicación de forma eficiente. Se incluye la posibilidad de brindar asistencia técnica que permita la solución de problemas en tiempo real de ejecución que garantice la solución de fallas que pueda presentar. Debe ser garantizada su adaptabilidad y compatibilidad con los distintos Sistemas Operativos.

Como garantía del producto se especificará los requerimientos de legalidad que permitan la autoría del propietario, el derecho de autor y todos los requerimientos legales que se establezcan entre las partes involucradas.

- **Interfaz:** El diseño de la interfaz visual debe ser amigable para los usuarios que interactúan con la aplicación, permitiendo el fácil entendimiento de las funcionalidades

## Características del Sistema

que el mismo brinda, además de poseer colores amigables y refrescantes para una mejor interacción entre usuario y la aplicación.

La interfaz constará de un grupo de botones, editores para textos, desplegables e informaciones que permitirán al usuario un fácil manejo de la aplicación y una realización rápida de las actividades. Toda la información referente a los textos, mensajes y opciones serán mostrados en idioma español.

El diseño de la interfaz permitirá mostrar mensajes para la guía de usuarios en caso de errores en entradas inválidas de los datos o de confirmación de realización de actividades. La interfaz permitirá además la posibilidad de obtener informaciones de salidas en diferentes formatos con posibilidad de ser impresos.

- **Software:** Para la elaboración de la aplicación se utilizará como sistema operativo Windows XP con Service Pack 2.
- **Hardware:** Para el desarrollo de la aplicación propuesta se demanda una estación de trabajo con 80 Gb de disco duro como mínimo, CPU a 3.0 GHz y 513 MB de RAM.

### 2.11 Modelo de Casos de Usos del Sistema

Dado la magnitud y lo complejo que se hace la construcción de la aplicación que se propone desarrollar, se hace necesario modelar sus principales funcionalidades en aras de obtener una noción lo más acertada posible de la interacción entre los actores que intervendrán en el uso del mismo y cada una de las funcionalidades de las que podrán hacer uso.

#### 2.11.1 Actores del Sistema

Actores	Descripción
Revisor	Es el encargado de efectuar las revisiones a los proyectos productivos ya sea de fin de fase o de seguimiento. Es además el encargado de llenar cada uno de los documentos que se generan como resultado de una revisión. Se encarga de actualizar cada documento, fases o información

# Características del Sistema

	referente a los proyectos.
<b>Planificador</b>	Es el encargado de hacer toda la planificación referente a las fechas en que se efectuarán las revisiones a cada uno de los proyectos. Es el único que puede hacer cambios en los cronogramas acorde a las necesidades específicas que pueden ocurrir.

## 2.12 Modelo de Casos de Usos del Sistema

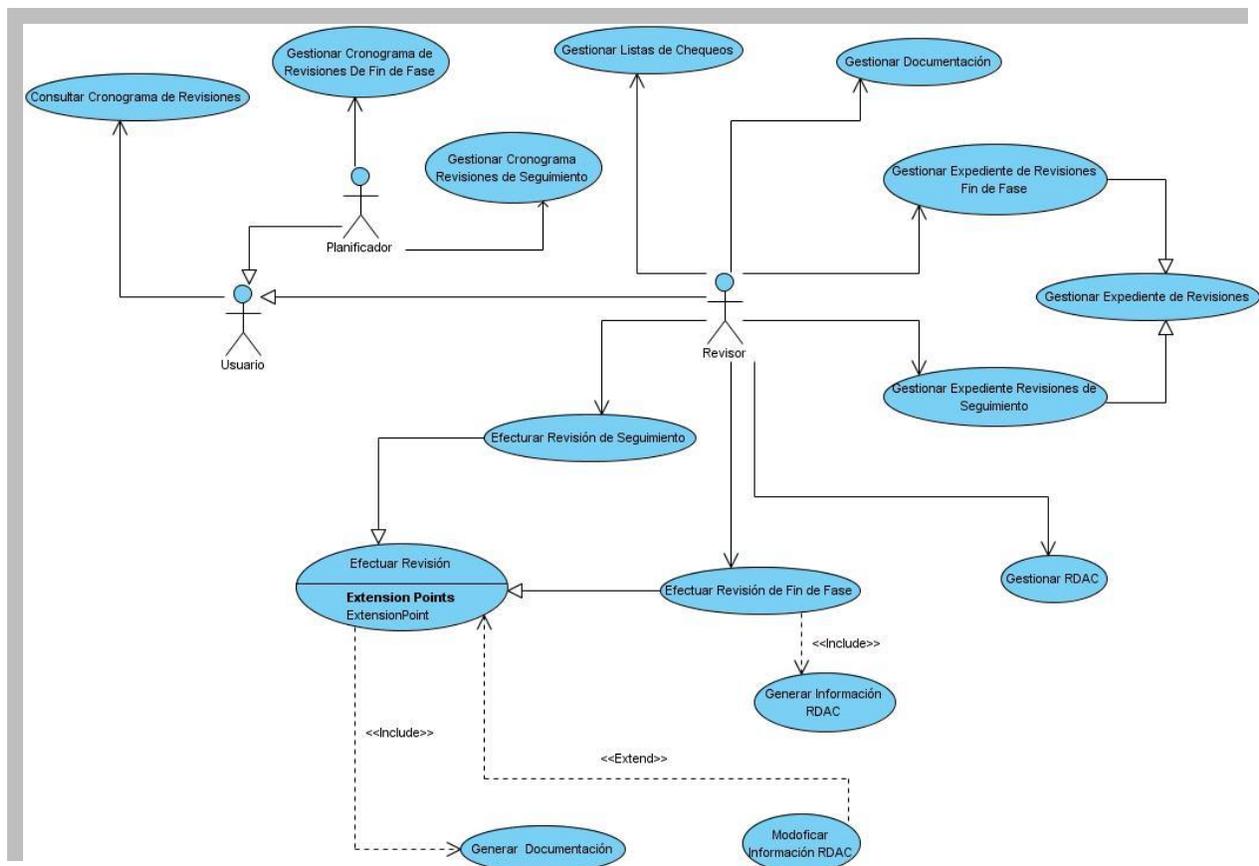


Figura 4 Diagrama de Casos de Usos del Sistema

### **Conclusiones Parciales**

En el presente capítulo se ha realizado un estudio acerca de los procesos fundamentales que intervienen en las revisiones que se realizan en el CESIM y la lógica del negocio de dichos procesos. Se identificaron los requerimientos a cumplir por parte del sistema, tanto funcional como no funcional. Se realizó la definición de los Casos de Usos del Sistema donde se encuentran las principales funcionalidades a implementar, además se llevo a cabo la descripción de cómo es la interacción entre los usuarios y los Casos de Usos del Sistema.

## Capítulo 3. Análisis y Diseño del Sistema

En este capítulo se realiza la descripción del diseño de la aplicación haciendo uso de los distintos diagramas afines para dicha actividad. Se realiza el análisis de cada uno de los casos de usos y se describe mediante el uso de estereotipos web la relación entre las distintas páginas que conforman la aplicación que se pretende desarrollar. Se obtiene además el diagrama de clases persistentes para la generación de la base de datos, se definen los principios de diseño y los patrones y estándares a utilizar.

### 3.1 Descripción de la Arquitectura

En los inicios de la informática, la programación se consideraba un arte y se desarrollaba como tal, debido a la dificultad que entrañaba para la mayoría de las personas, con el tiempo se han ido introduciendo guías generales con las cuales se pueden resolver los problemas que se generan en la programación y creación del software; a estas guías se les ha denominado Arquitectura de Software.

La utilización del Modelo Vista Controlador (MVC) como estilo dentro de la Arquitectura de Software permite la separación de los datos de la aplicación, la interfaz de usuario y la lógica del control en tres componentes distintos.

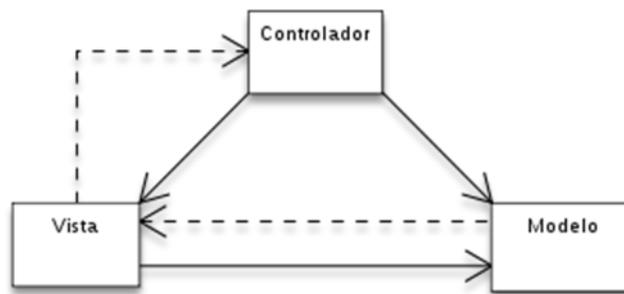


Figura 5 Modelo Vista Controlador (MVC)

- **Modelo:** Administra el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista y responde a instrucciones de cambiar el estado habitualmente desde el controlador). Mantiene el conocimiento del sistema. No depende de ninguna vista y de ningún controlador.
- **Vista:** Maneja la visualización de la información.

- Controlador: Interpreta las acciones del ratón y el teclado, informando al modelo y a la vista para que cambien según resulte apropiado. Tiene tres variantes principales: Activa, Pasiva y Documento-Vista.

El código usado por Django en su conjunto con el uso de las CGI hace un acercamiento total al uso del MVC. Entre las principales ventajas que hacen el uso de MVC con Django se puede mencionar que cada pieza de la aplicación web que funciona sobre Django tiene un único propósito clave, que puede ser modificado independientemente sin afectar las otras piezas. Por ejemplo, un desarrollador puede cambiar la URL de cierta parte de la aplicación sin afectar la implementación subyacente. Un diseñador puede cambiar el HTML de una página sin tener que tocar el código Python que la renderiza. Un administrador de base de datos y especificar el cambio en único lugar, en lugar de tener que buscar y reemplazar en varios archivos. [28]

Django sigue el patrón MVC tan al pie de la letra que puede ser llamado un framework MVC. Someramente la M, V y C se separan en Django de la siguiente manera:

- M, la porción de acceso a la base de datos, es manejada por la capa de base de datos de Django.
- V, la porción que selecciona que datos mostrar y cómo mostrarlos, es manejado por la vista y las plantillas.
- C, la porción que delega a la vista dependiendo de la entrada del usuario, es manejada por el framework mismo.

Django se rige por una arquitectura MVC (Modelo-Vista-Controlador), aunque se distingue en la vista de MVC los datos que se presentan (lo que ellos llaman vista) y la manera de mostrarlo, la plantilla. Por ello, ellos denominan a su arquitectura MTV (Model-Template-View, Modelo-Plantilla-Vista). El controlador en este caso sería el propio *framework* mismo.

## 3.2 Patrones

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo del software y otros ámbitos referentes al diseño de interacción de interfaces.

# Análisis y Diseño del Sistema

Actualmente el uso de patrones se ha generalizado entre los desarrolladores de aplicaciones en todo el mundo como una manera de estandarizar la creación de los nuevos sistemas informáticos.

Un patrón de diseño es una solución a un problema propio del diseño. Para que esta solución sea considerada como un patrón debe poseer ciertas características siendo la más importante, el estar comprobada su efectividad en la solución de problemas similares en ocasiones anteriores.

Los patrones GRASP (*General Responsibility Assignment Software Patterns*) son usados para la asignación de responsabilidades. Estos, más que considerarse como patrones son considerados como buenas prácticas recomendables en el diseño del software.

- **Experto:** La responsabilidad siempre recae sobre la clase que conoce toda la información necesaria para poder crear un objeto.
- **Creador:** Ayuda a identificar quien debe ser el responsable de la creación o instanciación de nuevos objetos o clases.
- **Controlador:** Sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, por lo que recibe las peticiones de los usuarios y se encarga de enviarlos a las distintas clases según el método llamado.
- **Alta cohesión:** La información que sea almacenada en la clase debe ser coherente y debe poseer alta relación con la misma.
- **Bajo acoplamiento:** La idea fundamental es tener las clases lo menos relacionadas posibles, lo que permite que la modificación de una de las clases repercuta lo menos posible en cada una de las restantes.

## 3.3 Modelo de Análisis

Este modelo es usado para representar la estructura global del sistema, describe la realización de casos de uso, sirve como una abstracción del modelo de diseño y se centra en los requerimientos no funcionales. [29]

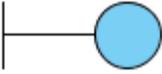
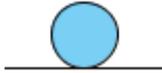
Durante el análisis, se analizan los requisitos que fueron descritos en la captura de requisitos, refinándolos y estructurándolos. El objetivo de hacerlo es conseguir una comprensión más

## Análisis y Diseño del Sistema

precisa de los requisitos y una descripción de los mismos que sea fácil de mantener y que ayude a estructurar todo el sistema, incluyendo su arquitectura. [30]

En la construcción de un modelo de análisis se tiene que identificar las clases que describen la realización de los casos de usos, los atributos y las relaciones entre ellas. Con esta información se construye el Diagrama de Clases del Análisis, que por lo general se descompone para agrupar las clases en paquetes. Esta descomposición tiene impacto por lo general en el diseño e implementación de la solución. [31]

Para realizar el modelado del análisis se definen tres clases;

Clases	Descripción	Representación
Interfaz	Las clases de interfaz se utilizan para modelar la interacción entre el sistema y sus actores.	 Interfaz
Entidad	Las clases entidad se utilizan para modelar información que poseen larga vida y que es a menudo persistente.	 Entidad
Control	Las clases control representan coordinación, secuencia, transacciones y control de objetos y son utilizadas para encapsular el control de un caso de uso.	 Control

Para llevar a cabo la realización del modelo de análisis descrito se usa varios diagramas definidos por UML, ellos son;

Diagrama	Representación
Diagrama de Clases	 Diagrama de Clases

Diagramas de Colaboración	 Diagrama de Comunicación
Diagramas de Secuencia	 Diagrama de Secuencia

### 3.3.1 Diagrama de Clases del Análisis.

Un diagrama de clases nos es más que un tipo de diagrama estático que está encaminado a la descripción de la estructura que presenta el sistema, mostrando sus clases, atributos y las relaciones que existen entre ellas. Son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de toda la información que manejará el sistema y los componentes que se encargan del funcionamiento y la relación entre ellos.

En estos diagramas de clases de análisis se hace uso de las tres tipos de clases definidos para el análisis según la ingeniería de Software; clases interfaz, entidad y controladoras.

El modelamiento del diseño es la antesala a la implementación del sistema. Este permite verificar el cumplimiento de los requisitos del sistema aun sin necesidad de un prototipo funcional de este. La especificación del diseño puede ser una de las etapas más largas dentro del desarrollo de un software, creando los modelos de diseños de la aplicación a construir y la refinación de cada uno de los elementos que la conforman a lo largo del ciclo de desarrollo.

### 3.4 Diagramas de Clases del Diseño

En el diagrama de clases del diseño se hace especificación a cada uno de los atributos y métodos que debe de tener cada una de las clases, se muestra la estructura estática del sistema representado de forma simplificada las responsabilidades y la colaboración entre cada clase dentro del sistema. Permite además especificar, documentar y visualizar modelos estructurales, siendo así base para la elaboración de ejecutables mediante el uso de la ingeniería directa e inversa.

# Análisis y Diseño del Sistema

En los diagramas de clases del diseño se hace uso de estereotipos web para entender la lógica del funcionamiento del sistema de una forma más dinámica y acercada a la realidad, estas son:

Estereotipos	Descripción	Representación
Server Page	Representa la página web que tiene el código que se ejecuta en el servidor.	
Client Page	Es una página web con formato HTML, donde se mezclan datos representación y lógica.	
HTML Form	Es una colección de elementos de entrada que son parte de una página cliente.	

En estos diagramas son usados además las clases controladoras, entidades y acceso a datos que modelan el funcionamiento del sistema con todos los atributos que el mismo posee.

## 3.4.1 Diagramas de Interacción en el Diseño

Los Diagramas de Interacción son un grupo de diagramas que pertenecen a los Diagramas de Comportamiento y se dividen en Diagramas de Colaboración y en Diagramas de Secuencias.

### ➤ Diagramas de Secuencia

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada método de la clase.

### ➤ Diagramas de Colaboración

Un diagrama de colaboración es esencialmente un diagrama que muestra interacciones organizadas alrededor de los roles.

## Análisis y Diseño del Sistema

Los diagramas de interacción son de gran importancia para el sistema pues permite observar la interacción que se produce entre los distintos objetos que participan en un escenario determinado.

### **Conclusiones Parciales**

En este capítulo se logra representar gráficamente la descripción del diseño mediante los distintos diagramas de clases del diseño referentes a la aplicación. Para la realización de estos diagramas se tuvo en cuenta la utilización de los estereotipos web que describen las distintas relaciones existentes. Además se definieron los principios de diseño, de arquitectura así como los principales patrones a tener en cuenta en la implementación de la herramienta web.

## Capítulo 4. Implementación

En el presente capítulo se muestra los resultados alcanzados en la implementación del sistema tal y como lo define RUP para esta disciplina dentro de la fase de Construcción. En esta etapa se muestra el modelo de datos obtenido y la descripción de sus tablas. Se especifican además los principales artefactos como el diagrama de despliegue y de componentes.

### 4.1 Modelo de Datos

El modelo de datos es un lenguaje orientado a describir las bases de datos. Un modelo de datos permite describir la estructura de datos de la base y su relación, las restricciones de integridad y las operaciones de manipulación de datos.

Los modelos de datos aportan la base conceptual para diseñar aplicaciones que hacen un uso intensivo de datos, así como la base formal para las herramientas y técnicas empleadas en el desarrollo y uso de sistemas de información.[32]

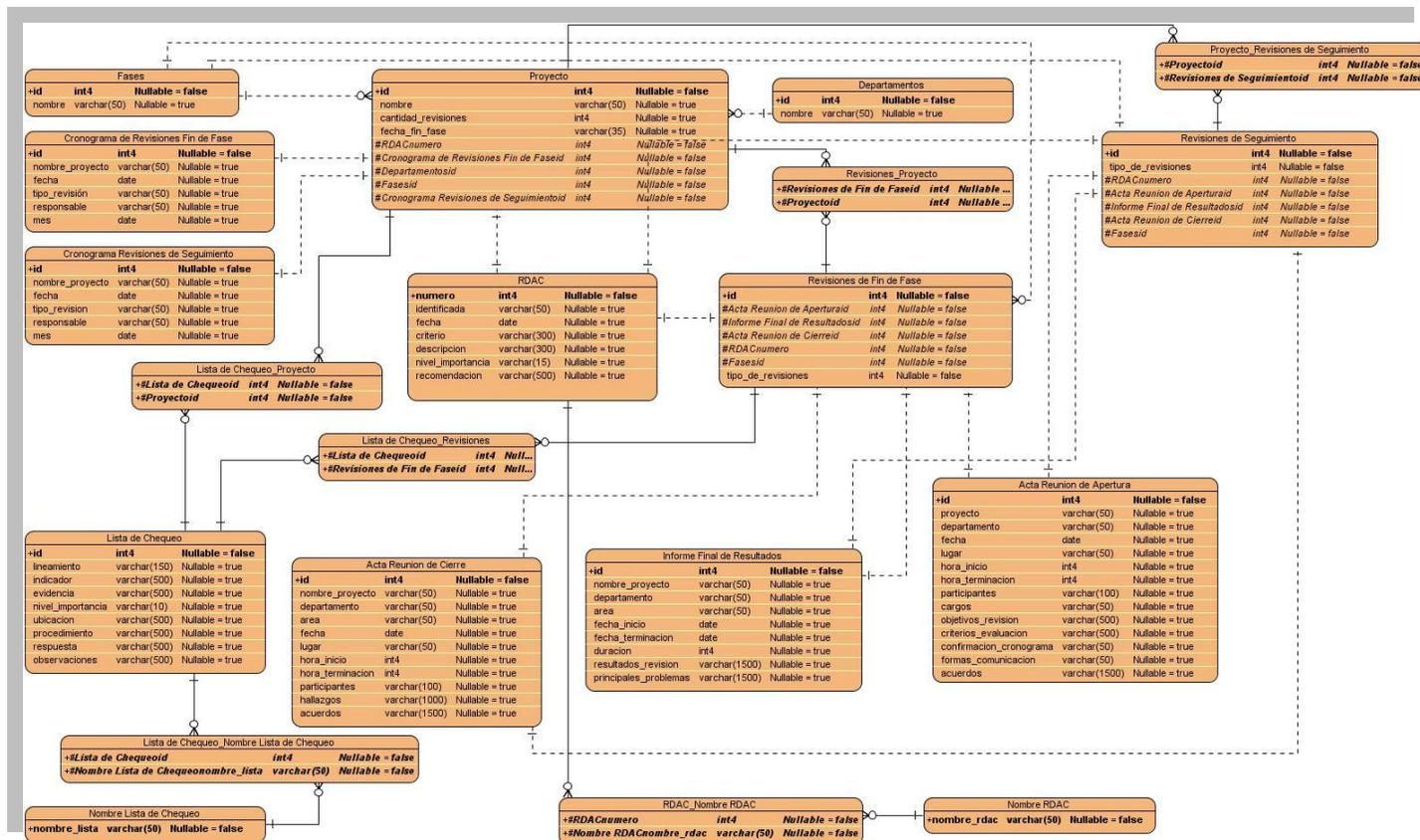


Figura 21 Modelo de Datos

## 4.1.1 Descripción de las Tablas

Las tablas forman parte del modelo de datos, permiten comprender los diagramas resultantes de la base de datos de la aplicación. La especificación de la información correspondiente con las clases y atributos que conforman la misma se muestra a continuación:

## 4.2 Implementación

Uno de los flujos de trabajo que define RUP como metodología de desarrollo, es el de Implementación. En este se comienza con el resultado del diseño y se implementa el sistema en término de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares. Su objetivo principal es desarrollar la arquitectura del sistema como un todo. Los propósitos de la implementación son:

- Definir la organización del código.
- Planificar las integraciones de sistema necesarias en cada iteración.
- Implementar las clases y subsistemas encontrados durante el Diseño. [33]

### 4.2.1 Modelo de Implementación

Un diagrama de implementación se asocia típicamente con un caso de uso para documentar que elementos de diseño (por ejemplo, componentes y clases) implementará la funcionalidad del caso de uso en el nuevo sistema. Muestra las dependencias entre las partes de código del sistema (diagrama de componentes) o la estructura del sistema en ejecución (diagrama de despliegue): los diagramas de componentes se utilizan para modelar la vista de implementación estática de un sistema, mientras que los diagramas de despliegue se utilizan para modelar la vista de despliegue estática. [34]

### 4.2.2 Diagrama de Despliegue

Un Diagrama de Despliegue modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de hardware (nodos) y muestra como los elementos y artefactos del software se trazan en esos nodos. [35]

Se utiliza en el diseño y en la implementación (como el diagrama de componentes) y nodos, así como las relaciones entre todos estos. Es más limitado que el diagrama de componentes, en el sentido que representa la estructura del sistema solo en tiempo de ejecución, pero no en

# Implementación

tiempo de desarrollo o compilación, por lo que resulta más amplio en el sentido de que puede contener más clases de elementos.

Para el despliegue se requiere de un Servidor de Base de Datos, el mismo debe ser Pentium 4, Service Pack 2 o superior para su uso sobre Windows, con 1Gb de memoria RAM como mínimo, 120G de disco duro para el almacenamiento de datos y un Sistema Gestor de Base de Datos en PostgreSQL montado sobre plataforma Windows o Linux, preferentemente software libre. Para el Servidor web se requiere una PC Pentium 4, Service Pack 2 o superior para su uso en Windows, 1Gb de memoria RAM como mínimo, 80G de disco duro y que permita su uso desde Windows o Linux, preferiblemente sobre software libre.

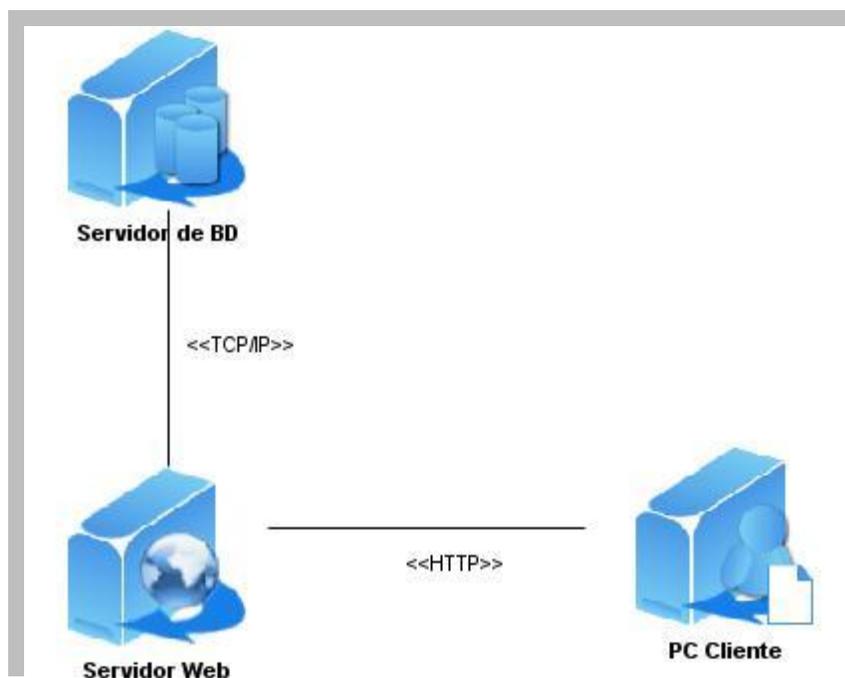


Figura 22 Diagrama de Despliegue

## 4.2.3 Diagrama de Componentes

El diagrama de componentes ilustra los componentes del software que serán usados para construir el sistema. Estos pueden ser construidos para el modelo de clase y escritos para satisfacer los requisitos del nuevo sistema, puede ser dada para otros proyectos o vendedores de tercera persona. El Diagrama de Componente muestra la relación entre los componentes del software, sus dependencias, comunicaciones, localización y otras condiciones. Los Diagramas de Componentes son usados para estructurar los componentes en los sistemas del software. Ellos examinan y controlan las dependencias entre componentes o interfaces de los

# Implementación

componentes. Un componente representa una parte modular, desplegable y reutilizable de un sistema.

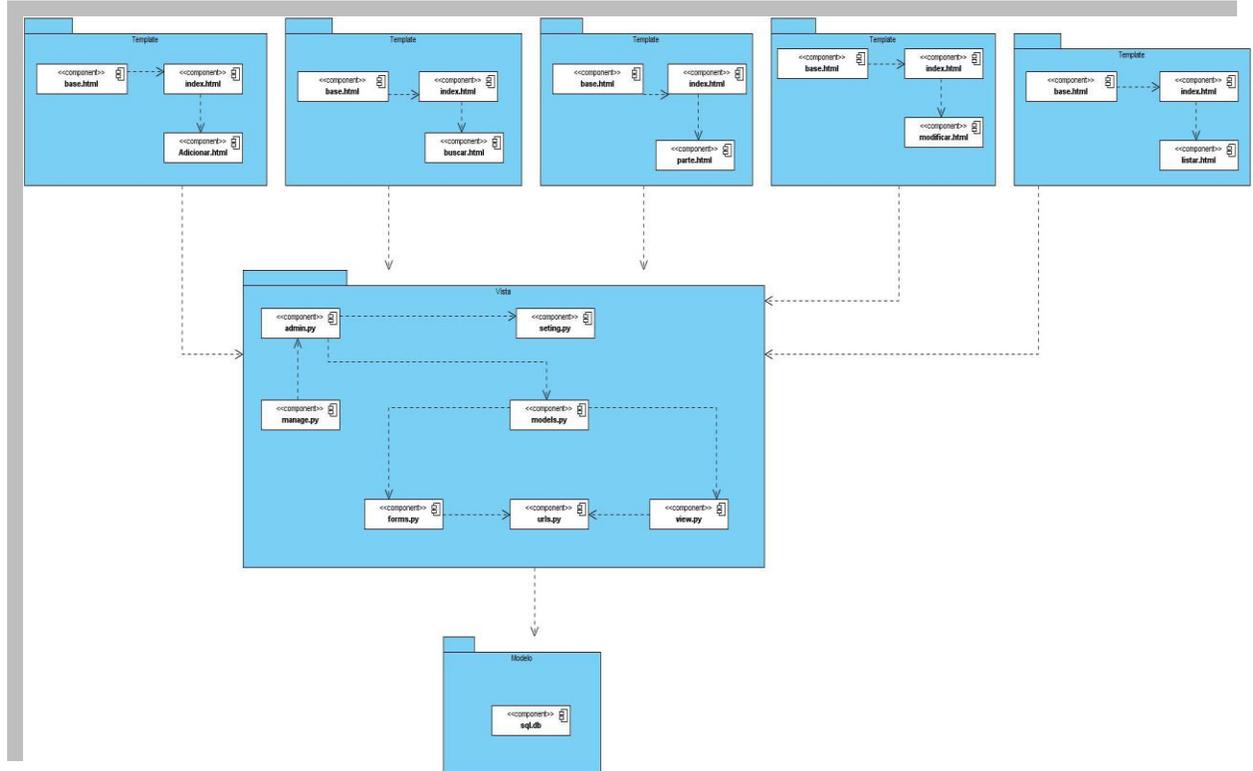


Figura 23 Diagrama de Componentes

A continuación se hace una representación del contenido de cada uno de los paquetes en los Templates:

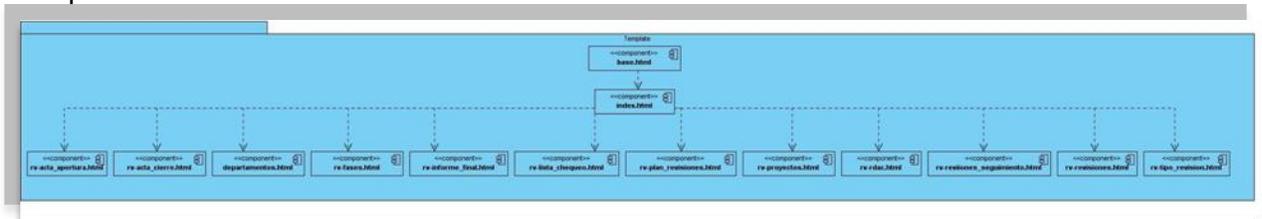


Figura 23.1 Paquete Adicionar

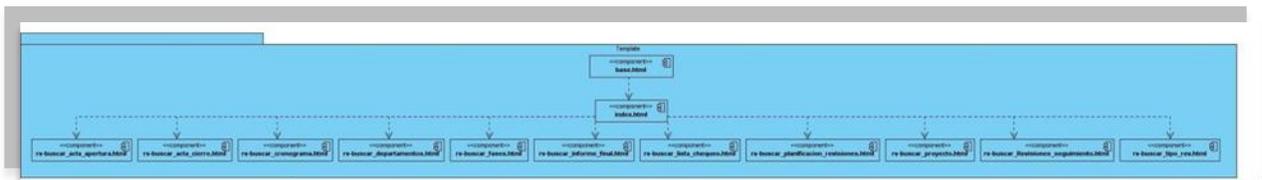


Figura 23.2 Paquete Buscar

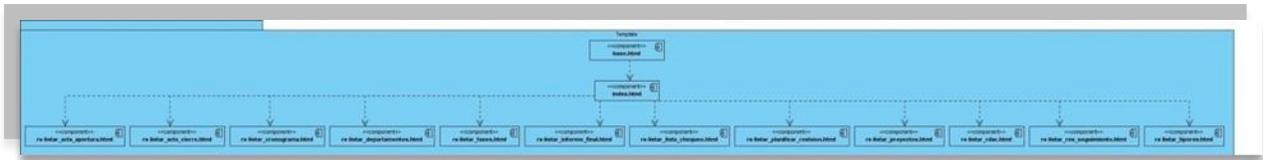


Figura 23.3 Paquete Listar

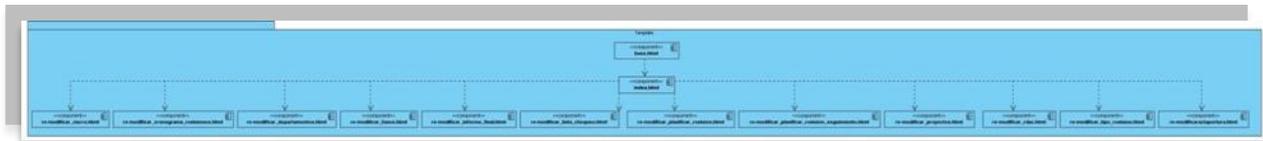


Figura 23.4 Paquete Mostrar

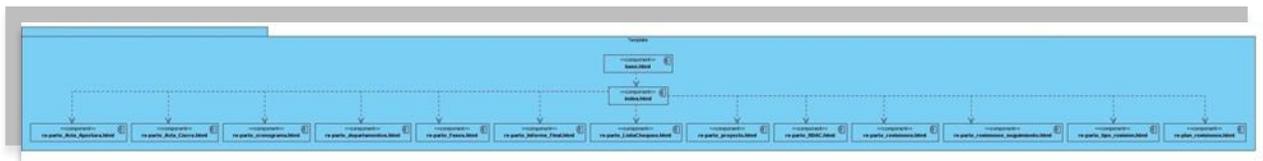


Figura 23.5 Paquete Parte

## 4.3 Seguridad

La seguridad es un elemento esencial para lograr la integridad, confidencialidad y autenticidad de la información almacenada y manejada por el sistema. Para ello el sistema implementa mecanismos que permitan asegurar la misma.

Al sistema sólo podrán acceder los usuarios con los permisos previamente establecidos y a los módulos a los cuales tendrán acceso acorde al nivel de accesibilidad de cada uno de los usuarios.

### Conclusiones Parciales

En el presente capítulo se desarrolló el flujo de trabajo de Implementación propuesto por RUP. En el mismo se desarrolló el Modelo de Datos como base fundamental para la generación de la Base de Datos de la herramienta, se muestran los principales diagramas referentes al flujo de trabajo además de implementar la propuesta de solución mediante una herramienta web.

## Conclusiones

La investigación realizada incluyó todo el proceso de revisiones que se realizan en el centro. Se realizó un estudio en el marco teórico, conceptual y tecnológico para la creación de las bases para llevar el diseño e implementación de la herramienta obtenida, la cual le permite al Grupo de Calidad del CESIM agilizar el trabajo en los procesos de revisiones que se le realizan a los proyectos productivos asociados al centro.

Del estudio realizado se pudo obtener las siguientes conclusiones:

- Los sistemas informáticos para la realización de revisiones en la actualidad son escasos y poco usados a nivel nacional e internacional.
- Los procesos de negocio que intervienen en las revisiones que se realizan actualmente en el Grupo de Calidad del CESIM pueden ser optimizado mediante el empleo de medios informáticos.
- La realización de los distintos modelos y diagramas propuestos por la metodología RUP son eficientes para solucionar el problema planteado.
- Con la utilización de la herramienta propuesta se optimizará el trabajo referente a toda la información manejada en los procesos de revisiones que se realizan actualmente en el CESIM.

Por lo resultados obtenidos tras la elaboración de la solución se puede concluir de forma tal que los objetivos planteados fueron cumplidos satisfactoriamente.

## Recomendaciones

A partir de los resultados del presente trabajo se plantean las siguientes recomendaciones:

- Incluir funcionalidades que permitan la gestión de los eventos relacionados con las auditorías informáticas.
- Trabajar en conjunto con los especialistas en el tema para futuros procesos de mejoras que le incorporen nuevas funcionalidades.
- Implementar las funcionalidades o permisos necesarias para que el proyecto modifique las dificultades detectadas y pueda interactuar con el RDAC

# Referencias Bibliográficas

## Referencias Bibliográficas

1. Universidad Tecnológica de Pereira. [En línea] [Citado el: 12 de noviembre de 2009.] <http://www.utp.edu.co/php/revistas/ScientiaEtTechnica/docsFTP/111233326-331.pdf>.
2. Portal PostgreSQL. [En línea] [Citado el: 16 de noviembre de 2009.] [http://www.postgresql.cl/acerca\\_de](http://www.postgresql.cl/acerca_de).
3. Comunidad Virtual de Conocimientos en Gestión de Negocios. [En línea] [Citado el: 12 de noviembre de 2009.] <http://www.gestiopolis.com/canales/gerencial/articulos/27/ISO.htm>.
4. Real Academia de la Lengua Española. [En línea] [Citado el: 2 de febrero de 2010.] <http://buscon.rae.es/drae/>.
5. Entrono Virtual de Aprendizaje. [Online] [Cited: febrero 2, 2010.] <http://eva.uci.cu/mod/resource/view.php?id=2756..>
6. Portal Calisoft. [En línea] [Citado el: 22 de noviembre de 2009.] [http://calisoft.uci.cu/attachments/040\\_Lineamientos%20de%20Calidad%20%28IPL%203101-2008%29%20v1.01.pdf](http://calisoft.uci.cu/attachments/040_Lineamientos%20de%20Calidad%20%28IPL%203101-2008%29%20v1.01.pdf).
7. Navegapolis. [En línea] [Citado el: 25 de febrero de 2010.] <http://www.navegapolis.net/content/view/314/78/>.
8. Aplicaciones Empresariales. [En línea] [Citado el: 27 de febrero de 2010.] <http://www.aplicacionesempresariales.com/dotproject-gestion-de-proyectos.html>.
9. Navegapolis. [En línea] [Citado el: 27 de febrero de 2010.] <http://www.navegapolis.net/content/blogcategory/0/85/10/80>.
10. Chu Wiki. [En línea] [Citado el: 9 de junio de 2010.] <http://chuwiki.chuidiang.org/index.php?title=Redmine>.
11. **Autores, Colectivo de.** *Una propuesta para la Gestión de Revisiones.*
12. Portal Python. [En línea] [Citado el: 16 de noviembre de 2009.] <http://www.python.org>.

## Referencias Bibliográficas

13. Comunidad Virtual de Cocimientos en Gestión de Negocios. [En línea] [Citado el: 12 de noviembre de 2009.] <http://www.gestiopolis.com/canales/gerencial/articulos/27/asesis.htm>.
14. Wikipedia. [En línea] [Citado el: 9 de junio de 2010.]  
<http://es.wikipedia.org/wiki/Django>
15. Linperial International Systems. [En línea] [Citado el: 9 de junio de 2010.]  
<http://www.linperial.com/communities/forums/developers/?q=node/62>.
16. **Hospitalaria, Consulta Especializada del Servicio de Rehabilitación para el Sistema de Informatización.** Ciudad de la Habana : s.n., 2009.
17. Idem 16. [En línea]
18. Wikipedia de Producción. [En línea] [Citado el: 2 de febrero de 2010.]  
<http://ucipedia.uci.cu/index.php/UM>.
19. UML. [En línea] [Citado el: 9 de junio de 2010.]  
<http://usuarios.multimania.es/oopere/uml.htm>
20. Sitio de Descarga de softwares. [En línea] [Citado el: 9 de junio de 2010.]  
[http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_%28M%C3%8D%29\\_14720\\_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/).
21. Idem 5. [En línea]
22. Wikipedia de Producción. [En línea] [Citado el: 8 de febrero de 2010.]  
[http://ucipedia.uci.cu/index.php/Aplicaci%C3%B3n\\_web](http://ucipedia.uci.cu/index.php/Aplicaci%C3%B3n_web).
23. Wikipedia. [En línea] [Citado el: 9 de junio de 2010.]  
[http://es.wikipedia.org/wiki/Aplicaci%C3%B3n\\_web](http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web)
24. Comunidad de desarrollo web. [En línea] [Citado el: 8 de febrero de 2010.]  
<http://www.desarrolloweb.com/wiki/aplicacion-de-escritorio.html>.

## Referencias Bibliográficas

25. Gestio Polis. [En línea] [Citado el: 9 de junio de 2010.]  
<http://www.gestiopolis.com/operaciones/reingenieria-de-procesos-de-negocios.htm>.
26. **Jacobson, Ivar, Booch, Grady y Runbaugh, James.** *El Proceso Unificado de Desarrollo de Software.* Madrid : s.n., 2000.
27. Idem a 26. [Online]
28. **Holovaty, Adrian y Kaplan-Moss, Jacob.** *El libro de Django.*
29. **Seguí, Arian y Reyes, Dariel F.** *Herramienta para la gestión de la información del Expediente de Proyecto.* Ciudad de la Habana : s.n., 2009.
30. **Informáticas, Universidad de las Ciencias.** Conferencia 7 Ingeniería de Software. Ciudad de la Habana : s.n., 2010.
31. **Informáticas, Universidad de las Ciencias.** *Conferencia 6 Ingeniería de Software.* Ciudad de la Habana : s.n.
32. Universidad Autónoma de Barcelona. [En línea] [Citado el: 9 de junio de 2010]  
<http://elies.rediris.es/elies9/4-2.htm>.
33. Comunidad de SOFTWARE factory. [En línea] [Citado el: 8 de marzo de 2010.]  
<http://www.newsoftwarefactory.com/implementacion.htm>.
34. Comunidad SPARK SYSTEMS. [En línea] [Citado el: 8 de marzo de 2010.]  
[http://www.sparxsystems.com.ar/resources/tutorial/use\\_case\\_model.html](http://www.sparxsystems.com.ar/resources/tutorial/use_case_model.html).
35. Comunidad SPARK SYSTEMS. [En línea] [Citado el: 8 de marzo de 2010.]  
[http://www.sparxsystems.com.ar/resources/tutorial/uml2\\_deploymentdiagram.html..](http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html..)

## Bibliografía Consultada

- Aplicaciones empresariales[En línea] [Citado el: 2010 de junio de 5] <http://www.aplicacionesempresariales.com/sqlite-el-motor-de-base-de-datos-agil-y-robusto.html>
- **Autores, Colectivo de.** *Una propuesta para la Gestión de Revisiones.*
- **Canonical Ltd. Ubuntu.** Ubuntu. [En línea] [Citado el: 1 de noviembre de 2009.] <http://www.ubuntu.com>.
- Compute-rs.com[En línea] [Citado el: 2010 de junio de 5] <http://www.compute-rs.com/es/consejos-1571565.htm>
- Comunidad SPARK SYSTEMS. [En línea] [Citado el: 8 de marzo de 2010.] [http://www.sparxsystems.com.ar/resources/tutorial/uml2\\_deploymentdiagram.html..](http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html..)
- **Esencia Humana SC.** Esencia Humana. [En línea] [Citado el: 2009 de diciembre de 6.] <http://www.esenciahumana.com.mx>.
- **GNU Consultores.** GNU Consultores. [En línea] [Citado el: 2010 de febrero de 3.] <http://www.gnuconsultores.com>.
- **Holovaty, Adrian y Kaplan-Moss, Jacob.** *El libro de Django.*
- **Hospitalaria, Consulta Especializada del Servicio de Rehabilitación para el Sistema de Informatización.** Ciudad de la Habana : s.n., 2009.
- **Jacobson, Ivar, Booch, Grady y Runbaugh, James.** *El Proceso Unificado de Desarrollo de Software.* Madrid : s.n., 2000.
- Libro Ingeniería del software. [En línea] [Citado el: 2010 de marzo de 8]<http://books.google.com.cu/>
- Libros Web. [En línea] [Citado el: 2009 de noviembre de 10.] <http://www.librosweb.es>.
- Portal Calisoft. [En línea] [Citado el: 22 de noviembre de 2009.] [http://calisoft.uci.cu/attachments/040\\_Lineamientos%20de%20Calidad%20%28IPL%203101-2008%29%20v1.01.pdf](http://calisoft.uci.cu/attachments/040_Lineamientos%20de%20Calidad%20%28IPL%203101-2008%29%20v1.01.pdf).
- Portal PostgreSQL. [En línea] [Citado el: 16 de noviembre de 2009.] [http://www.postgresql.cl/acerca\\_de](http://www.postgresql.cl/acerca_de).
- Portal Python. [En línea] [Citado el: 16 de noviembre de 2009.] <http://www.python.org>.

## Bibliografía Consultada

- Real Academia de la Lengua Española. [En línea] [Citado el: 2 de febrero de 2010.] <http://buscon.rae.es/drae/>.
- **Seguí, Arian y Reyes, Dariel F.** *Herramienta para la gestión de la información del Expediente de Proyecto*. Ciudad de la Habana : s.n., 2009.
- Tienda Linux. [En línea] [Citado el: 2010 de febrero de 3.] <http://www.soporte.tiendalinux.com>.
- **Universidad de las Ciencias Infomáticas.** Calisoft. [En línea] [Citado el: 2009 de diciembre de 4.] <http://calisoft.uci.cu>.
- Universidad Tecnológica de Pereira. [En línea] [Citado el: 12 de noviembre de 2009.] <http://www.utp.edu.co/php/revistas/ScientiaEtTechnica/docsFTP/111233326-331.pdf>.
- Wikipedia. [En línea] [Citado el: 2009 de diciembre de 5.] <http://es.wikipedia.org>.
- Winware. [En línea] [Citado el: 12 de febrero de 2010.] <http://www.wingide.com>.

## Glosario

**Framework:** Es una estructura de soporte definida en la cual otro proyecto puede ser organizado y trabajado. Un framework puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

**Herramienta Case:** Aplicación informática destinada a aumentar la productividad en el desarrollo de software, reduciendo el coste de las mismas en tiempo y dinero, se utiliza para el modelado del sistema.

**IDE:** Entorno Integrado de Desarrollo, es un programa compuesto por un conjunto de herramientas para un programador. Puede ser exclusivo de un lenguaje o puede ser utilizado en más de uno.

**MVC:** Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica del control en tres componentes distintos. Es ampliamente utilizado en aplicaciones web

**RDAC:** Registro de Dificultades y Acciones Correctivas.

**Revisión de Fin de Fase:** Revisión que es realizada a los artefactos generados por un proyecto al concluir cada una de las fases definidas por la metodología en uso durante el proceso de desarrollo de software.

**Revisión de Seguimiento:** Revisión que se realiza para darle seguimiento al cumplimiento de los defectos a corregir que fueron detectados en la Revisión de Fin de Fase o una Revisión de Seguimiento realizada con anterioridad.